Selected Inductive Biases in Neural Networks To Generalize Beyond the Training Domain

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät

der Eberhard Karls Universität Tübingen

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

(Dr. rer. nat.)

vorgelegt von

M.Sc. Lukas Schott

aus München

Tübingen

2021

Tag der mündlichen Qualifikation:	30.03.2022
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatter:	Prof. Dr. Matthias Bethge
2. Berichterstatter:	Prof. Dr. Ullrich Köthe

In loving memory of my father
Max Schott.

Abstract

Artificial neural networks in computer vision have yet to approach the broad performance of human vision. Unlike humans, artificial networks can be derailed by almost imperceptible perturbations, lack strong generalization capabilities beyond the training data and still mostly require enormous amounts of data to learn novel tasks. Thus, current applications based on neural networks are often limited to a narrow range of controlled environments and do not transfer well across tasks.

This thesis presents four publications that address these limitations and advance visual representation learning algorithms.
In the first publication, we aim to push the field of disentangled representation learning towards more realistic settings. We observe that natural factors of variation describing scenes, e.g., the position of pedestrians, have temporally sparse transitions in videos. We leverage this sparseness as a weak form of learning signal to train neural networks for provable disentangled visual representation learning. We achieve competitive results on the disentanglement_lib benchmark datasets and our own contributed datasets, which include natural transitions.
The second publication investigates whether various visual representation learning approaches generalize along partially observed factors of variation. In contrast to prior robustness benchmarks that add unseen types of perturbations during test time, we compose, interpolate, or extrapolate the factors observed during training. We find that the tested models mostly struggle to generalize to our proposed benchmark. Instead of predicting the correct factors, models tend to predict values in previously observed ranges. This behavior is quite common across models. Despite their limited out-of-distribution performances, the models can be fairly modular as, even though some factors are out-of-distribution, other in-distribution factors are still mostly inferred correctly.
The third publication presents an adversarial noise training method for neural networks inspired by the local correlation structure of common corruptions caused by rain, blur, or noise. On the ImageNet-C classification benchmark, we show that networks trained with our method are less susceptible to common corruptions than those trained with existing methods.
Finally, the fourth publication introduces a generative approach that outperforms existing approaches according to multiple robustness metrics on the MNIST digit classification benchmark. Perceptually, our generative model is more aligned with human vision compared to previous approaches, as images of digits at our model's decision boundary can

also appear ambiguous to humans.
In a nutshell, this work investigates ways of improving adversarial and corruption robustness, and disentanglement in visual representation learning algorithms. Thus, we alleviate some limitations in machine learning and narrow the gap towards human capabilities.

Kurzfassung

Die künstlichen neuronalen Netze des computergesteuerten Sehens können mit den vielfältigen Fähigkeiten des menschlichen Sehens noch lange nicht mithalten. Im Gegensatz zum Menschen können künstliche neuronale Netze durch kaum wahrnehmbare Störungen durcheinandergebracht werden, es mangelt ihnen an Generalisierungsfähigkeiten über ihre Trainingsdaten hinaus und sie benötigen meist noch enorme Datenmengen für das Erlernen neuer Aufgaben. Somit sind auf neuronalen Netzen basierende Anwendungen häufig auf kleine Bereiche oder kontrollierte Umgebungen beschränkt und lassen sich schlecht auf andere Aufgaben übertragen.

In dieser Dissertation, werden vier Veröffentlichungen besprochen, die sich mit diesen Einschränkungen auseinandersetzen und Algorithmen im Bereich des visuellen Repräsentationslernens weiterentwickeln.
In der ersten Veröffentlichung befassen wir uns mit dem Erlernen der unabhängigen Faktoren, die zum Beispiel eine Szenerie beschreiben. Im Gegensatz zu vorherigen Arbeiten in diesem Forschungsfeld verwenden wir hierbei jedoch weniger künstliche, sondern natürlichere Datensätze. Dabei beobachten wir, dass die zeitlichen Änderungen von Szenerien beschreibenden, natürlichen Faktoren (z.B. die Positionen von Personen in einer Fußgängerzone) einer verallgemeinerten Laplace-Verteilung folgen. Wir nutzen die verallgemeinerte Laplace-Verteilung als schwaches Lernsignal, um neuronale Netze für mathematisch beweisbares Repräsentationslernen unabhängiger Faktoren zu trainieren. Wir erzielen in den disentanglement_lib Wettbewerbsdatensätzen vergleichbare oder bessere Ergebnisse als vorherige Arbeiten – dies gilt auch für die von uns beigesteuerten Datensätze, welche natürliche Faktoren beinhalten.
Die zweite Veröffentlichung untersucht, ob verschiedene neuronale Netze bereits beobachtete, eine Szenerie beschreibende Faktoren generalisieren können. In den meisten bisherigen Generalisierungswettbewerben werden erst während der Testphase neue Störungsfaktoren hinzugefügt - wir hingegen garantieren, dass die für die Testphase relevanten Variationsfaktoren bereits während der Trainingsphase teilweise vorkommen. Wir stellen fest, dass die getesteten neuronalen Netze meist Schwierigkeiten haben, die beschreibenden Faktoren zu generalisieren. Anstatt die richtigen Werte der Faktoren zu bestimmen, neigen die Netze dazu, Werte in zuvor beobachteten Bereichen vorherzusagen. Dieses Verhalten ist bei allen untersuchten neuronalen Netzen recht ähnlich. Trotz ihrer begrenzten Generalisierungsfähigkeiten, können die Modelle jedoch modular sein: Obwohl sich einige Faktoren während der Trainingsphase in einem zuvor ungesehenen

Wertebereich befinden, können andere Faktoren aus einem bereits bekannten Wertebereich größtenteils dennoch korrekt bestimmt werden.
Die dritte Veröffentlichung präsentiert ein adversielles Trainingsverfahren für neuronale Netze. Das Verfahren ist inspiriert durch lokale Korrelationsstrukturen häufiger Bildartefakte, die z.B. durch Regen, Unschärfe oder Rauschen entstehen können. Im Klassifizierungswettbewerb ImageNet-C zeigen wir, dass mit unserer Methode trainierte Netzwerke weniger anfällig für häufige Störungen sind als einige, die mit bestehenden Methoden trainiert wurden.
Schließlich stellt die vierte Veröffentlichung einen generativen Ansatz vor, der bestehende Ansätze gemäß mehrerer Robustheitsmetriken beim MNIST Ziffernklassifizierungswettbewerb übertrifft. Perzeptiv scheint unser generatives Modell im Vergleich zu früheren Ansätzen stärker auf das menschliche Sehen abgestimmt zu sein, da Bilder von Ziffern, die für unser generatives Modell mehrdeutig sind, auch für den Menschen mehrdeutig erscheinen können.
Diese Arbeit liefert also Möglichkeiten zur Verbesserung der adversiellen Robustheit und der Störungstoleranz sowie Erweiterungen im Bereich des visuellen Repräsentationslernens. Somit nähern wir uns im Bereich des maschinellen Lernens weiter der Vielfalt menschlicher Fähigkeiten an.

Contents

Chapter 1

Introduction

1.1 Limited generalization in machine learning

> *Clever Hans was a horse owned by Wilhelm von Osten in the early 20th century. It was claimed that it could solve mathematical problems such as subtraction, addition, and others by tapping its hoof on the ground to demonstrate the solution. For instance, "four plus two" would result in six taps. However, after raising scientific interest, it was later shown in several ablation studies that the horse does not solve math problems. It merely relied on the unconscious but anticipative facial expression of its questioner to determine when to stop tapping its hoof (Johnson, 1911).*

The story of the Clever Hans horse reveals the difficulty of assuring that even a straightforward task such as performing simple additions is learned as intended. It further demonstrates the limitations of *shortcut* solutions: As the horse relied on the unintentional expressions of its questioner, it was unable to answer the mathematical questions on its own or if the questioner did not know the solution. Thus, the applicability of the horse's feigned mathematical skills was limited to the demonstration with the visible presence of its questioner (Prinz, 2006). Nonetheless, von Osten happily continued showcasing the abilities of his horse in defiance of being scientifically disproven.

Machine learning has also been showcased to demonstrate astonishing abilities in various tasks such as playing the combinatorially challenging board game Go (Silver *et al.*, 2016), protein folding (Jumper *et al.*, 2021), competing in the question answering Show *Jeopardy!* (Ferrucci *et al.*, 2013), in the strategy game StarCraft II (Vinyals *et al.*, 2019), and visual pattern recognition (Ciresan *et al.*, 2011). With the increased abilities, our lives become more reliant on machine learning systems such as virtual assistants in phones (e.g., Siri, Alexa, Google Assistant), image processing in autonomous driving, traffic light recognition apps for the blind or to track herds of livestock. Thus, a more thorough understanding of machine learning and its possible downsides is crucial.

With the showcasing of revolutionary capabilities of neural networks, several limitations come to light. Failure cases or unintended solutions of machine learning have been observed in various disciplines. In reinforcement learning, *reward hacking* describes an agent following an unintended strategy but still achieving high reward. For instance, a robot rewarded to achieve an environment free of messes might simply disable its vision to seemingly remove the mess (Amodei *et al.*, 2016). In medical imaging, unplanned behavior has been shown by models picking up unintended signals in the data. For example, a neural network designed to aid diagnosing dermoscopic images strongly relies on surgical ink markings present in the data (Winkler *et al.*, 2019) . Also, neural networks lack robustness. In image-based tasks, they are easily confused by common corruptions such as rain, blur, or compression artifacts. In a more extreme case, adversarially crafted perturbations that are humanly almost imperceptible, can derail a neural network (Szegedy *et al.*, 2014; Biggio *et al.*, 2013). For instance, an image of a "pig" can be slightly perturbed to be classified as an "airliner" by a neural network (Madry and Schmidt, 2018). For an undefended neural network, such adversarial perturbations exist for practically every input. Even though such adversarial perturbations are unlikely to occur in nature, methods have been developed to apply them in the physical world. A universal sticker can be put on or next to objects such that they are recognized as a toaster (Brown *et al.*, 2017). Notably, this sticker only remotely resembles a toaster. Thus, despite their remarkable capabilities, common machine learning algorithms show unintended behavior. Small shifts in the application domain can reveal such shortcomings.

It has been proposed that these failure cases can be related to the fact that neural networks do not learn the intended solutions, but rather rely on spurious features (Ilyas *et al.*, 2019) or shortcut learning (Geirhos *et al.*, 2020). To provide an intuitive example, we consider a cow in front of an atypical scene such as a sandy beach instead of a usual green pasture. Here, it can happen that the cow on the beach is no longer recognized by a neural network (Beery *et al.*, 2018). It seems the green pasture background itself is usually already quite predictive and might be sufficient to recognize cows in certain tasks. Thus, the network might actually never learn the underlying concept of a cow, but still be able to recognize it in images with suiting backgrounds. Surprisingly, such shortcuts are almost omnipresent in high dimensional problems: They are quite predictive in various settings and learning algorithms often seem to prefer shortcuts over a principled solution (Brendel and Bethge, 2019; Ilyas *et al.*, 2019; Wilson *et al.*, 2017; Bartlett *et al.*, 2020; Arjovsky *et al.*, 2020; Bruna *et al.*, 2015; Geirhos *et al.*, 2019).

Similarly to Geirhos *et al.* (2020) and given the previous examples, we define **shortcut solutions** as solutions that perform well on domains that are identically distributed to their training domain but do not generalize to other scenarios. We define the **intended solution** for neural networks as solutions that not only perform well on tasks similar to their training domain, but also generalize to various out-of-distribution scenarios similar to humans. We rely on humans as a proxy for generalization because a) they generalize much better than neural networks (Geirhos *et al.*, 2018), and b) it is desired that neural

networks are aligned with our intuitions as we want them to be useful for humans.[1]

Another possible explanation why neural networks often do not learn the intended solution is that they are often highly underspecified. Well-performing models on images have tens or hundreds of millions of tuneable parameters. Here, the number of tuneable parameters often even surpasses the number of images. When combining this high flexibility of neural networks with the ubiquitous presence of shortcuts, we can get a multitude of different models that perform well on the training and similar test data (Barker and Achinstein, 1955; Choromanska *et al.*, 2015; Draxler *et al.*, 2018). However, on out-of-distribution tasks, these models vary in performance, even if they only differ by the random seed during the initialization (D'Amour *et al.*, 2020). Thus, the common strategy of simply providing a training domain with labelled images is often insufficient to properly specify a network. A possible remedy could be provided by inductive biases that specify our models and guide them towards learning the intended solution.

Avoiding shortcuts and achieving generalization is imperative for further improvement of machine learning systems acting in our complex world. In image-based learning tasks, generalizations beyond the training domain are omnipresent. For instance, weather changes like rain, snow, hail, or fog alone can lead to small domain shifts. Backgrounds can change due to different seasons, e.g., leaves can turn from green over yellow to brown. Moreover, lighting conditions of an image can change due to clouds or even in laboratory settings due to a different light bulb or a different camera angle. This myriad of possible variations in the world results in a "heavy-tailed distribution" and it is almost infeasible to account for all possible scenarios during the development of a learning algorithm. Thus, we require additional tools to assure safer behavior and foster generalization.

All in all, we highlight the necessity for generalization in machine learning. We propose *shortcuts* and *underspecification* to be core underlying problems of the limited generalization capabilities of machine learning algorithms observed in many tasks. For more trustworthy machine learning solutions and a broader set of applications, it is desired that networks actually learn concepts that transfer more reliably across tasks and domains. To learn a solution closer to the intended one, we propose to incorporate additional inductive biases. The overall reasoning of this section is also depicted in Fig. 1.1. In the next section, we provide an overview of common inductive biases to further specify neural networks to foster generalization.

[1] We refer to Sholarin *et al.* (2015) for a more philosophical discussion of relying on humans as a measure.

Figure 1.1: **Necessity for inductive biases for generalization in machine learning.**

1.2 Inductive biases for generalization[2]

In the previous section, we argue for the necessity of additional specifications (inductive biases) for machine learning models to achieve intended generalizations. Here, we delve into various proposed inductive biases from the research literature. The section is structured as follows: First, we provide a formal definition of inductive biases and give a prototypical example. Second, we present a brief historical account of structure in deep learning. Third, we introduce a coarse categorization of inductive biases into groups. Lastly, we discuss inductive biases in the context of human-like generalization.

1.2.1 Definition and examples of inductive biases

We informally define **inductive biases** as characteristics of learning algorithms that influence their generalization behavior beyond the training domain (Mitchell, 1997; Abnar *et al.*, 2020).[3] Our broad definition of inductive biases covers all explicit and implicit assumptions while implementing a learning algorithm similar to the definition of Hüllermeier *et al.* (2013). In contrast to statistical learning theory that focuses on generalizations to test data that stems from the same distribution as the training data (Vapnik, 2013; Vapnik and Chervonenkis, 1982), we concentrate on generalizations beyond the training domain (also referred to as *out-of-distribution generalization*) and use humans as a proxy to determine what types of generalization should be achievable. To limit the scope, we further focus on inductive biases that have been previously considered in the context of generalization in the computer vision literature.

A prototypical example of an inductive bias in vision are *convolutional* neural networks (CNNs) (LeCun *et al.*, 1999; Fukushima, 1988). They leverage translational symmetries in our world. Especially, low-level image features, like edges or textures, can appear in

[2] This section is partially adapted from our paper (Schott *et al.*, 2021).

[3] In contrast to Abnar *et al.* (2020), who explicitly define inductive biases independent of the data, we also consider data used to initialize an algorithm as a possible inductive bias as, e.g., pretraining on large image datasets is a common approach in out of distribution generalization.

arbitrary positions in images. CNNs share learned parameters and use them to compute abstractions across the image. For instance, if the concept of edges is learned at a specific location of the image, it can also be extracted at other positions. When combining this translational symmetry with a pooling operation, we gain a spatial invariance. E.g., for simple object recognition in front of a plain background, the exact position of an object is not relevant for a classification. Compared to fully connected neural networks, convolutional networks drastically reduce the number of model parameters, are more data efficient, and outperform previous approaches on various benchmarks (Yin *et al.*, 2013; Stallkamp *et al.*, 2011; Arganda-Carreras *et al.*, 2015).

1.2.2 A brief history of structure in deep learning

Historically, incorporating structure in algorithms versus flexibility varied. In the first wave of deep learning in around 1950, low computational power and small datasets limited learning to small models such as the perceptron (McCulloch and Pitts, 1943; Rosenblatt, 1958). No training algorithms were available to efficiently train stacked perceptrons in practice, and limiting them to problems that are linearly separable (Minsky and Papert, 1969). Therefore, most systems and research focussed on rule-based algorithms and domain experts. The second wave of deep learning around 1980/90, laid out the algorithmic foundation for later achievements. Here, tools like backpropagation enable the training of deep and flexible network architectures consisting of multiple, stacked perceptrons (Rumelhart *et al.*, 1986). Also, more data efficient algorithms for time series or images like shift invariant neural networks were developed (LeCun *et al.*, 1989; Fukushima, 1988).

In the early 21st century, powerful GPUs allowed for more computations and the rise of a digital age led to abundant amounts of data. Larger models could be trained mostly by leveraging the techniques proposed in the second wave. These models significantly outperformed previous handcrafted approaches (e.g., (Stallkamp *et al.*, 2011; Yin *et al.*, 2013; Kümmerer *et al.*, 2014)). Thus, a shift from hand-crafted feature engineering to highly flexible architectures occurred. With the mantra of "end-to-end learning", as little as possible should be specified beforehand and everything inferred from the data. This can be summarized in an exaggerated way by an alleged quote of Frederick Jelinek who worked on automated speech recognition: "Every time I fire a linguist, the performance of the speech recognizer goes up."[4] Also, limitations of too specific structures were pointed out, e.g., the thought of recognizing cats in images by fitting geometric forms like a triangle to the nose and ears has complex dependencies on the viewpoint, occlusions and lighting (Li, 2015). Thus, most rule-based systems are not suited to cover the vast range of variations present in the world.

Paradoxically, with the rising success of large networks, also voices raised the point that

[4]The specific phrasing and time of the quote is unclear (Wikiquote, 2021)

from the perspective of theoretical non-convex optimization, the success of such over-parameterized models should not be possible (Sejnowski, 2020). Later work partially explained the unreasonable success of deep learning and attributed it to implicit properties of the optimizer (Roberts, 2021), high dimensional spaces and of the architectures (Kawaguchi, 2016). Those could be seen as revealing implicit biases in the deep learning pipeline.

From today's perspective, in a recent debate in 2018, key figures Yann LeCun and Christopher Manning discussed "What innate priors should we build into the architecture of deep learning systems" (LeCun and Manning, 2018). Here, LeCun called structure a "necessary evil" but hopes to relax structures by leveraging more data and more evolved unsupervised learning techniques in the future. He underlines this by stating that even the structural benefits of CNNs should not be necessary if enough data is available. Chris Manning, in contrast, stated incorporating structure as a "necessary good". He used humans as role models of good learners with high sample efficiency that he attributes to priors. He further stated that massive amounts of compute and data have sent the research field "off track", as one "can do a lot of stuff with a very simple learning device". However, we should strive for "good learners" similar to humans that, compared to machines, need little data to learn certain tasks.

In the context of this thesis, we also acknowledge the necessity of inductive biases and try to find a middle ground. We aim to develop inductive biases that incorporate structure to help on certain out-of-distribution scenarios, but are yet flexible enough to not lower the performance. We closely investigate inductive biases and their effect on other out-of-distribution scenarios to check whether there are possible trade-offs.

1.2.3 Common inductive biases

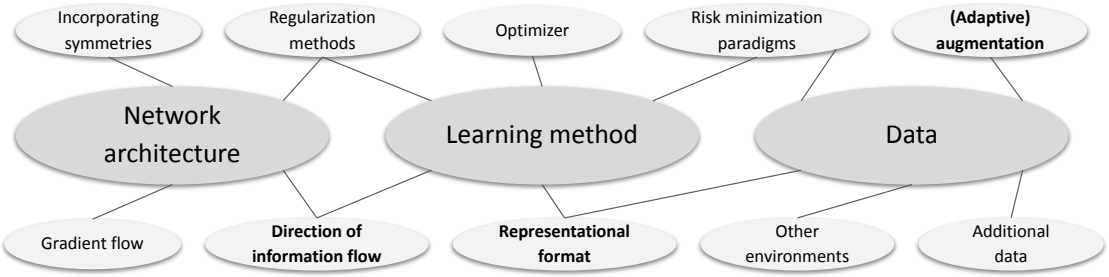


Figure 1.2: **Examples of inductive biases.** A list of common inductive biases considered for generalization in machine learning. Bold inductive biases correspond to biases studied in this thesis.

We coarsely categorize inductive biases of neural networks in visual representation learning into *network architecture*, *learning method*, and *data*. We focus on promising direc-

tions for generalizations beyond the training domain, and on literature relevant in this thesis. For a broader spectrum of inductive biases, we refer to Battaglia *et al.* (2018) and Craven (1996).

Network architecture

Incorporating symmetries: Our physical world is governed by symmetries that are connected to conservation laws (Noether, 1915). Discovering and leveraging such symmetries has lead to tremendous advances in physics. E.g., spatial translational symmetry leads to the conservation of momentum or the time translational symmetry leads to a conservation of energy. In the previous example of the CNN, we also observed benefits of leveraging spatial symmetries in neural networks. Among others, this has further been generalized to rotations (Marcos *et al.*, 2016; Fasel and Gatica-Perez, 2006), scales (Xu *et al.*, 2014) and group operations by so-called G-Convolutions (Cohen and Welling, 2016). Other methods also introduce invariances in the processing on the level of individual pixels by leveraging set based and coordinate-based representations (Achlioptas *et al.*, 2018; Zhang *et al.*, 2019b). To implement these symmetries, the individual network layers leverage a high degree of weight sharing, which facilitates the generalization along the considered symmetry.

Biases of common network architectures: Implicitly, the network architecture also has a large influence on how information is processed. CNNs, for instance, often focus on local relations common in textures (Brendel and Bethge, 2019; Geirhos *et al.*, 2019). In contrast, Long-Short-Term-Memory (LSTM) or Gated Recurrent Units (GRU) networks introduce a specific cell structure that allows for non-vanishing gradient updates during learning to input points with arbitrary distances (Hochreiter and Schmidhuber, 1997; Cho *et al.*, 2014). Thus, very distant inputs can be related more easily. Similarly, transformers (Vaswani *et al.*, 2017), which treat the input as a set and rely on an attention mechanism, have been shown to leverage distant relations in images (Carion *et al.*, 2020). Thus, in contrast to convolutional networks that rely on texture for object classifications, transformers can be trained to rely on shape similar to humans and show promising out-of-distribution generalization results (Geirhos *et al.*, 2021). While transformers, LSTMs and GRUs allow for distant spatial dependencies, Residual Networks (ResNets) (He *et al.*, 2016), DenseNets (Huang *et al.*, 2017) and UNets (Ronneberger *et al.*, 2015) facilitate the learning across different hierarchies of information processing in neural networks. They enable direct relations of low-level concepts like edges extracted in the shallow layer of neural networks with more abstract object properties that can be found in deeper layers. Respectively, these networks have shown high accuracies on image-based benchmarks like ImageNet, CIFAR, and segmentation tasks. Their high reliability and flexibility renders these networks as a common backbone for further developments in generalization methods such as Geirhos *et al.* (2019); Rusak *et al.* (2020) or Hendrycks *et al.* (2020b).

Learning method

In this section, we use the term learning method as an umbrella for design choices regarding different types of risk minimization objectives, representational format, regularization methods, and the direction in which information is processed.

Risk minimization paradigms: Normally, we do not have access to the full data distribution for a certain task, but instead have to rely on a limited set of training samples with a corresponding label. Additionally, we often rely on a proxy loss function that defines how close our model predictions for given samples are to the true label. Given a loss function and a set of labelled examples from a training distribution, *empirical risk* is defined as the average loss over all training samples. *Empirical risk minimization*, finds a hypothesis from the hypothesis class that minimizes the average loss. However, often there exists a multitude of different strategies to achieve low or even zero empirical risk. To further narrow the space of possible solutions, Arjovsky *et al.* (2020) introduce *invariant risk minimization*. They assume additional access to multiple environments and aim for a solution that extract features that are invariant across environments. For instance, to distinguish cows from camels it is often quite predictive whether the background is a green pasture or a sandy desert. Assuming additional access to a few images of cows on a sandy background and leveraging invariant risk minimization, shifts the focus to the desired object (cow or camel). Note that assuming additional environments is also leveraged by Hyvärinen and Morioka (2016) for provably identifiable representation learning.

Direction of information flow: In visual representation learning, the direction in which we compute representations, e.g., from images to labels or, vice versa, from labels to images can have a high influence on the generalization properties of a model. In the standard deep learning setup in visual representation learning tasks such as object recognition, we train a network end-to-end with full supervision to learn a mapping from images to labels (feedforward) on the training data, which consists of pairs of images and their corresponding labels. It has been shown that this approach can rely on single predictive features that are sufficient to classify an object (Brendel and Bethge, 2019; Ilyas *et al.*, 2019; Jacobsen *et al.*, 2019; Geirhos *et al.*, 2019). For instance, to distinguish cats from ships, looking at local texture patterns can be sufficient. In contrast, in the principle of analysis by synthesis (de Cordemoy, 1973; Von Humboldt *et al.*, 1999), we learn a mapping from labels to images. For a classification of images, we synthesize images from a model to find the likelihood for each sample $p(\mathbf{x}|c)$ for a class c. Next, based on maximum likelihood, we can infer the class (Schott *et al.*, 2019). Thus, due to this reconstruction, all features have to be matched correctly for a likely classification. However, this method is computationally much more demanding compared to feedforward network architectures. For a more in-depth evaluation, we refer to Mackowiak *et al.* (2021).

Regularization methods: A common philosophical principle to choose one hypothesis over another about the same prediction is Occam's razor. Analogously, in neural networks, the concept of regularization is used to incentive solutions with lower capacity. Common candidates for regularization are weight decay, L1 regularization, or early stopping. This can also be seen in the context of the bias-variance trade-off (Kohavi *et al.*, 1996). Here, a lower regularization can lead to more flexible models, but they are more prone to overfitting and can have high variances.

Representational format: The representation of high-dimensional data has a significant impact on the generalization performance of the downstream model (Bengio *et al.*, 2013). A principled method in visual representation learning is based on (nonlinear) independent component analysis (ICA) (Comon, 1994; Bell and Sejnowski, 1995), also referred to as disentanglement. Corresponding methods assume a set of independent latent variables that give rise to the observed data. Common methods to recover the latent factors, such as the Variational Autoencoder (VAE) (Kingma and Welling, 2014) and variants thereof (Kumar *et al.*, 2018; Chen *et al.*, 2018), rely on a bottleneck and maximize a lower bound to the marginal likelihood of the data (also referred to as evidence). However, in most cases, the unknown latent variables cannot be guaranteed to be identified correctly (Hyvärinen and Pajunen, 1999; Locatello *et al.*, 2019a). Given some additional weak assumptions about these latent factors, it was shown that they can be inferred up to some unavoidable transformations (Hyvärinen and Morioka, 2016; Locatello *et al.*, 2020b; Klindt *et al.*, 2021). Subsequent studies on downstream tasks have shown that this method can facilitate the generalization capabilities of a neural network. Furthermore, this allows mimicking the ground truth generative model behind a scene on the training data. This property could be helpful to learn principled models.

As these disentanglement models only require weak or no explicit supervision, they can be used to harness large quantities of unlabeled data, which is available in abundance. Here, models pretrained using contrastive learning, a method highly connected to disentanglement (Zimmermann *et al.*, 2021), are currently among the state-of-the-art approaches on the ImageNet classification benchmark (Chen *et al.*, 2020a).

Activity matching: Another approach relies on mimicking brain activities of animals or humans for certain inputs, and thus guiding models to learn similar strategies (Fong *et al.*, 2018). Given similar processing patterns, one could expect to overcome limitations in machine learning, such as limited out-of-distribution generalization. Empirically, by matching the neural activities in mouse brains, the robustness of artificial neural networks towards Gaussian noise and adversarial perturbations could be increased (Li *et al.*, 2019b) . However, a gap in robustness remains between the true and artificial neural models.

Data

Augmenting the input data: Another way to boost the out-of-distribution performance of a neural network is to augment in the input data. For instance, simply adding Gaussian noise to the input of ImageNet classifiers can substantially improve their out of distribution performance (Rusak *et al.*, 2020) and is related to regularization (Bishop, 1995). Other approaches artificially create input variations, e.g., shearing or rotating the input, or by adding synthesized common corruptions such as rain, blur or compression artifacts. More adaptive methods like adversarial training iteratively compute worst-case perturbations for a given classifier and add them to its training data (Madry *et al.*, 2018). Combined with a carefully constrained search space of allowed perturbations, such methods show state-of-the-art results on common out-of-distribution tasks (Kireev *et al.*, 2021; Calian *et al.*, 2021).

Leveraging additional data: Powerful pre-trained methods based on contrastive learning or other un-/ weakly supervised representation learning methods (Xie *et al.*, 2020) allow harvesting enormous amounts of (weakly annotated) datasets (Sun *et al.*, 2017; Kolesnikov *et al.*, 2020). Surprisingly, often using more data and relying on an end-to-end learnable architecture, can outperform sophisticated baselines, which is also referred to as *Sutton's bitter lesson* (Sutton, 2019).

1.2.4 Evaluation of inductive biases in the context of generalization

Naively incorporating inductive biases into a model should be treated with care. The *no free lunch theorem*, states that the introduction of a certain inductive bias to improve the performance on one task is guaranteed to worsen the performance on another task (Wolpert and Macready, 1997). However, as defined in the previous section, we strive for a robustness similar to humans. Thus, we postulate the existence of a solution with human-like robustness and accept trade-offs in other domains. We think this is a reasonable goal, as there is still a large gap in terms of out-of-distribution performance between humans and machines.

To quantify our progress, we consider various out-of-distribution scenarios that seem easy for humans. Here, previously introduced inductive biases often help neural networks to generalize and have enabled machine learning applications on previously unreachable domains. Nonetheless, a large gap towards human like generalization remains. Thus, finding better generalizing inductive biases is an open research question.

1.3 Goal of this thesis

The goal of this thesis is to provide a condensed overview of our published work on inductive biases for visual representation learning. For each contribution, we highlight the previous state as well as its impact. Our contributions are the following:

- We first introduce a principled method for unsupervised disentanglement in natural videos. Based on temporal properties of natural transitions and other weak assumptions, we can provably identify the factors of variation behind a dataset. We also provide two practical implementations based on a VAE (named SlowVAE) and based on flows. On common disentanglement datasets and two introduced natural datasets, we further show competitive or superior results on various disentanglement metrics. Thus, we advance the field of provable disentanglement towards more natural data.
- Second, we further test whether not only the corresponding underlying factors of datasets are learned, but also the generative model behind each factor. This test relies on introducing systematic out-of-distribution data splits along known factors of variation, such as size. E.g., if small and medium-sized objects are recognized during training, the model is required to recognize large objects during testing. Based on a large-scaled benchmark of 17 representation learning algorithms on four different datasets with various out-of-distribution scenarios, we conclude that the generalization towards novel configurations of present factors from the data is limited.
- Third, we introduce *adversarial noise training* (ANT). This is a model adaptive data augmentation method that mimics certain properties of common corruptions. We show that our proposed training scheme can increase the corruption robustness on ImageNet-C and MNIST-C. We further show that additive Gaussian noise is already a competitive baseline if it is properly scaled.
- In the fourth and last contribution, we introduce a model that relies on the principle of *analysis by synthesis* (ABS). Our model achieves high or even state-of-the-art adversarial robustness on various L_p-norms on the MNIST digit classification task. It further leads to adversarials that appear to be at the human decision boundary. Furthermore, it shows robustness towards distal adversarials, meaning that confidently predicted images resemble humanly plausible images of digits.

Next, we discuss the research question *do our investigated inductive biases learn the intended solution?* To answer this question, we test our proposed inductive biases across *multiple* out-of-distribution scenarios. This is in contrast to the individual contributions sections, in which we always consider a specific inductive bias for *one* out-of-distribution

scenario and not across *multiple*[5]. We propose this procedure to estimate our progress towards human-like robustness. The more out-of-distribution settings a proposed inductive bias successfully transfers to, the smaller the gap towards human-like robustness and the intended solution. More concretely, the out-of-distribution scenarios we focus on are common corruptions, adversarial examples, and novel configurations of known factors of variation. Our considered inductive biases are implemented by the models ABS, ANT and SlowVAE. We find that our inductive biases are mostly orthogonal. While they increase the performance on the individually considered out-of-distribution scenarios, they have nearly no effect on other out-of-distribution scenarios.

The second research question investigates *how well our proposed inductive biases can be combined?* To evaluate, whether our inductive biases can be combined in a symbiotic manner, we consider preliminary results and other literature. We find that the combination of our inductive biases is often mutually beneficial. Thus, it points out promising future research directions.

1.4 Outline

In the introduction in Section 1.1, we highlight the necessity of inductive biases and in Section 1.2 introduce related candidates relevant for this thesis.

In the subsequent contribution sections in Chapter 2, we introduce inductive biases. Here, the individual contributions are ordered by the intricacy of the considered generalization scenario and provide a high-level description of a corresponding publication in the Appendices A to D. Furthermore, each contribution section is coarsely structured by introducing the problem, shortly describing our proposed inductive bias and discussing the contribution in the context of the relevant literature.

In *Transfer and combination of our inductive biases* in Chapter 3, we provide a bigger picture and discuss our proposed inductive biases across all individually considered generalization scenarios. In the second part of this discussion, we examine the combination of our proposed inductive biases.

In the outlook in Chapter 4, we hypothesize about future directions in the research of inductive biases for generalization in machine learning.

In the appendix, we add our publications on which this thesis is based on. The overall outline is depicted in Fig. 1.3

[5]This separation of inductive biases and a corresponding out-of-distribution scenario is performed to give a clear structure to this thesis. The results to support this discussion among multiple scenarios are also mostly taken from the publications in the appendix and complement by results from the literature. When no suiting results were available, novel experiments were performed.

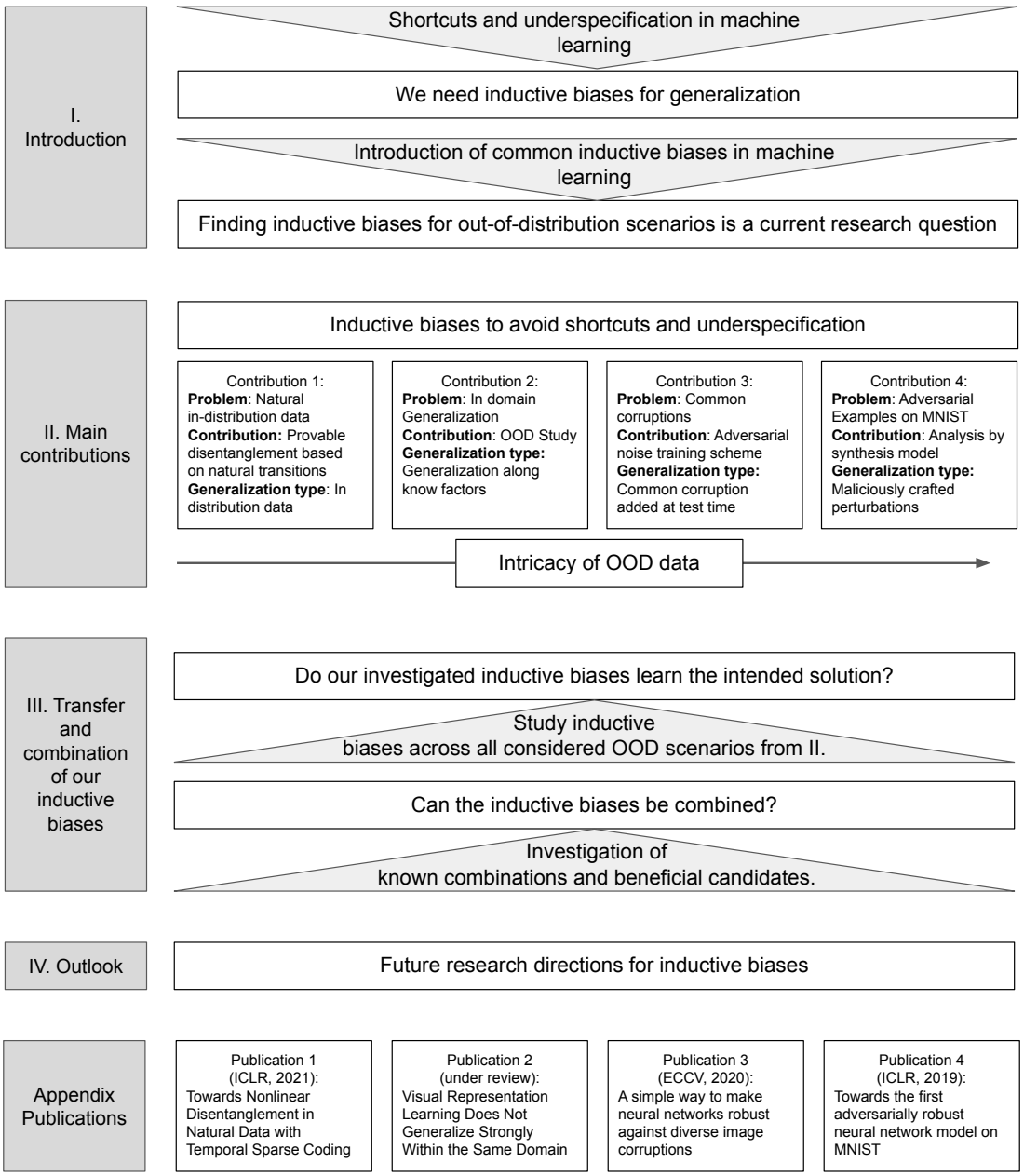


Figure 1.3: **Outline of this thesis.**

Chapter 2

Main contributions

In this chapter, we provide an intuitive and high-level description of our contributions. Each contribution consists of providing an example of limited generalization and, subsequently, presenting approaches to overcome those limitations. We arrange our contributions by the intricacy of the examined generalization types. We start with a simple scenario in which the training data distribution is equivalent to the test distribution, but no direct supervision signal is given. In the subsequent chapters, we gradually move to more intricate out-of-distribution scenarios and allow for full supervision. Here, we first consider a combinatorial generalization along factors of variation present in the data, such as having small and medium-sized objects during training but requiring the model to recognize large objects during testing. Next, we consider common corruptions such as rain, blur, or compression artifacts. In contrast to the previous section, these artifacts are introduced at test time and not necessarily present during training. Lastly, we consider adversarial examples, which are maliciously crafted samples to maximally change the classification of a learning model.

Each section also corresponds to an individual paper presented in the Appendix. The individual papers contain an encapsulated and much more detailed description, with a focus on the specific contribution.

2.1 Towards nonlinear disentanglement in natural data with temporal sparse coding

David Klindt*, Lukas Schott*, Yash Sharma*, Ivan Ustyuzhaninov, Wieland Brendel, Matthias Bethge°, Dylan Paiton°.
Published as a conference paper and as an oral at the ICLR 2021.

*Joint first authors / equal contribution, °joint senior authors

Author contributions M.B. proposed the idea of temporal sparse coding with deep networks repeatedly to the lab; D.A.K. conceived the idea of the model with input from **L.S.** and D.P.; D.A.K., **L.S.** and Y.S. performed the main experiments; **L.S.** and Y.S. respectively designed the simplified natural dataset KITTI-Masks and NaturalSprites with input from D.A.K., D.P. and W.B.; D.A.K., Y.S. and **L.S.** analyzed the statistics of the natural datasets; **L.S.** performed an in-depth assessment of how latents are encoded in figures 4., 5. and multiple appendix figures with input from D.A.K., Y.S. and D.P.; M.B. structured and supervised the theoretical analysis of the paper.; I.U., D.A.K. and W.B. proved theorem 1; D.A.K. derived the objective function for the SlowVAE with input from I.U.; **L.S.** derived the objective function for the SlowFlow with input from D.P.; **L.S.** and Y.S. performed an in-depth comparison to PCL in appendix B; D.P. compared the metrics in appendix B; Y.S. performed the permutation experiments in appendix G; D.A.K., D.P., Y.S. and **L.S.** wrote the manuscript with input from W.B., I.U. and M.B.
All section and figure references are w.r.t. to the publication in Appendix A.

We provide a model to learn a disentangled representation of the world based on video data. Here, we alleviate current limitations in unsupervised representation learning approaches, such as the restriction to artificial or heavily controlled environments. In contrast to previous methods that introduce artificial inductive biases to learn a provable disentangled latent representation, we rely on properties observed in temporal data as a first principle to develop our model.

2.1.1 Problem: disentanglement in toy data

A principled approach for visual representation learning can be derived from the independent component analysis (ICA) and disentanglement literature (Jutten and Herault,

1991; Comon, 1994; Hyvärinen and Morioka, 2017). Here, intuitively, we assume that observations in the world are rendered by a generative model that receives certain factors of variation as input. It is assumed that these factors of variation are not observed directly and provide an abstract description of a scene. They could, for instance, specify an object and its properties such as position in space, color, or others. A *generator* model or computer graphics engine then draws an observable image corresponding to the specified factors. The goal of disentanglement is to recover the latent factors of variation from observations.

Disentangled representations can be useful for generalization to novel scenarios (Higgins *et al.*, 2017b), fairness (Locatello *et al.*, 2019b), increased interpretability (Adel *et al.*, 2018; Higgins *et al.*, 2017a), predictive performance (Locatello *et al.*, 2019a) and sample efficiency (van Steenkiste *et al.*, 2019; Locatello *et al.*, 2020a).

One limitation in disentanglement learning approaches is that we do not have guarantees on identifying non-linearly mixed latent factors solely based on independent and identically distributed observations without additional assumptions (Hyvärinen and Pajunen, 1999; Locatello *et al.*, 2019a). For instance, a factor could be learned such that it simultaneously controls the shape and size when varied, but according to the true model, those should be two separate factors. Locatello *et al.* (2019a) showed that this empirically affects the commonly used latent variables models, such as the variational autoencoder (Kingma and Welling, 2014) and its extensions (Kim and Mnih, 2018; Higgins *et al.*, 2017a; Chen *et al.*, 2018; Kumar *et al.*, 2018). To overcome this limitation, methods rely on additional types of supervision signals that enable a provable identification of the latent distribution. Minimal but sufficient types of supervision rely on receiving pairs of images as input that only differ in a few factors (Locatello *et al.*, 2020b; Hyvärinen and Morioka, 2017), or by conditioning on additional variables (Hyvärinen and Morioka, 2016; Khemakhem *et al.*, 2020b,a). However, so far, these types of weak supervision are often created artificially and might not exist in nature. Thus, current methods for disentanglement mostly rely on artificial datasets (Kim and Mnih, 2018; Matthey *et al.*, 2017; LeCun *et al.*, 2004) or highly controlled environments (Gondal *et al.*, 2019). It remains unclear whether these required additional assumptions for identifiability are present in unstructured, real-world environments. In the next section, we address this limitation and present a possible remedy to help to transfer to more natural settings.

Another limiting assumption required for provable identifiability is that all possible data points are observed. This might be practically infeasible in most cases, as the number of data points scales exponentially with the number of factors, or to infinity for continuous factors. This is further discussed in Section 2.2.

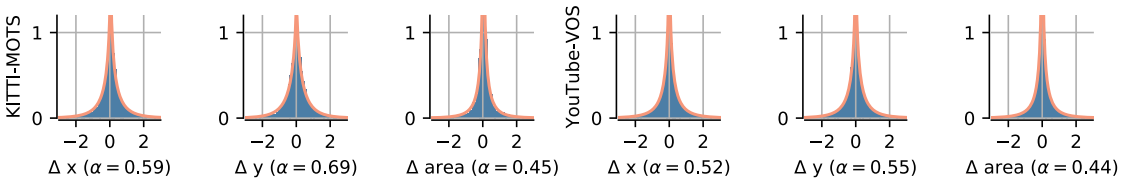


Figure 2.1: **Sparse natural transitions.** We depict the histogram of differences of factors of variation from neighboring time frames in videos. The orange lines correspond to a fit of generalized Laplacian distribution. On the left side, we consider masked pedestrians from KITTI-MOTS. On the right side, we show the same for various objects from YouTube-VOS. Figure adapted from our paper (Klindt *et al.*, 2021) / Appendix A.

2.1.2 Approach: temporal sparse coding

As highlighted in the previous section, to provably identify the latent factors given observations from natural data, we require additional assumptions or inductive biases to inform our model.

To find a suitable inductive bias for identifying latent factors, we analyze fully annotated data from natural videos in YouTube-VOS (Xu *et al.*, 2018) and of pedestrians from an autonomous driving dataset (Geiger *et al.*, 2012; Voigtlaender *et al.*, 2019a; Milan *et al.*, 2016). We extract object masks and analyze their behavior over time. E.g., the x-position, y-position, and scale of a walking pedestrian, as shown in Fig. 2.3. We find that the transitions of these measured factors all follow a sparse generalized Laplace distribution, as shown in Fig. 2.1 (Sinz *et al.*, 2009). This observation is motivated by and in accordance with previous measured transition behaviors (Simoncelli and Olshausen, 2001; Olshausen, 2003; Hyvärinen *et al.*, 2003).

We leverage the observed sparse temporal transitions to inform a model with neighboring time frames of videos and, under mild assumptions, provide a proof for identifiability. Our proof allows us to identify factors of variation of up to just permutations and sign flips, which is a more extensive type of identification compared to previous approaches that additionally require linear transformations (Hyvärinen and Morioka, 2016) or pointwise nonlinearities (Locatello *et al.*, 2020b; Hyvärinen and Morioka, 2017). To the best of the authors' knowledge, this is the first approach connecting the presence of sparse transitions in natural data to provable disentanglement[1].

Empirically, we compare our model to previous methods on various datasets and metrics. As baselines, we consider the identifiable models Ada-GVAE (Locatello *et al.*, 2020b) and PCL (Hyvärinen and Morioka, 2017). For brevity, we here disregard the inferior

[1]Similar to our work, Hyvärinen and Morioka (2017) assume a generalized exponential distribution, which is theoretically not directly applicable to our observed leptokurtic sparseness, but empirically performs close to our model. For more details, we refer to a direct comparison in our paper (Klindt *et al.*, 2021) in Appendix A.

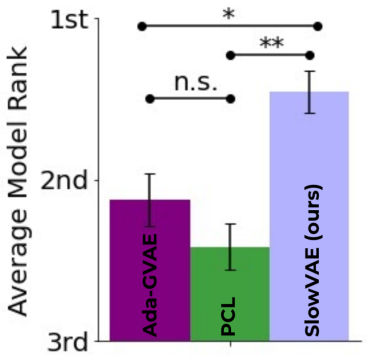


Figure 2.2: **Disentanglement performances.** We show average ranks over all metrics and datasets including dSprites (LAP & UNI), Natural Sprites (discrete & continuous), KITTI-Masks ($\Delta t = 0.05$ & $\Delta t = 0.15$). For more, see Appendix B. Significance tests based on 100k sample permutation tests, * $p < 0.05$, ** $p < 0.001$, non-significant (n.s.), i.e., $p > 0.05$.

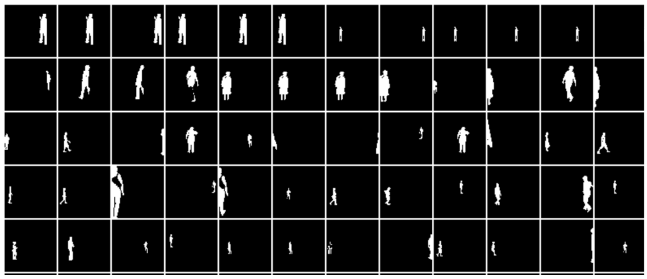


Figure 2.3: **KITTI-Masks dataset.** We show samples from masked pedestrians from the KITTI-Masks dataset. Each image corresponds to a time frame from KITTI. We further cropped individual pedestrians. The measured factors of variation are size (number of pixels in the mask) and x/y-position (center of mass of the mask). Figure adapted from our paper (Klindt *et al.*, 2021) / Appendix A.

performing non-identifiable models, such as the β-VAE and its variants. To briefly compare models, we here aggregate across several metrics (Higgins *et al.*, 2017a; Kim and Mnih, 2018; Ridgeway and Mozer, 2018; Chen *et al.*, 2018; Kim and Mnih, 2018; Kumar *et al.*, 2018), This is done for brevity and as evaluating disentanglement is still an unsolved research problem. Individual results are shown in Appendix A.

We evaluate our model on the datasets from the DisLib, such as dSprites (Matthey *et al.*, 2017), Shapes3D (Kim and Mnih, 2018), SmallNorb (LeCun *et al.*, 2004) and MPI3D (Gondal *et al.*, 2019). Here, to train the models, we sample pairs of images corresponding to Laplacian (LAP) transitions that match our model assumptions and with uniform (UNI) transitions as an ablation. In both cases, our model shows superior or competitive results compared to previous baselines. Furthermore, we evaluate our model on our contributed datasets with natural transitions, namely, Natural Sprites (based on YouTube-VOS (Xu *et al.*, 2018)) and KITTI-Masks (based on KITTI (Geiger *et al.*, 2012; Voigtlaender *et al.*, 2019b)). Again, our model performs better on average. The overall aggregated results including various ablations are depicted in Fig. 2.2.

For a qualitative assessment, we show that our models learn latent representations, which demonstrate a clear correspondence between the learned and ground truth latent representation. This can, for instance, be seen clearly for our model on the factors x- and

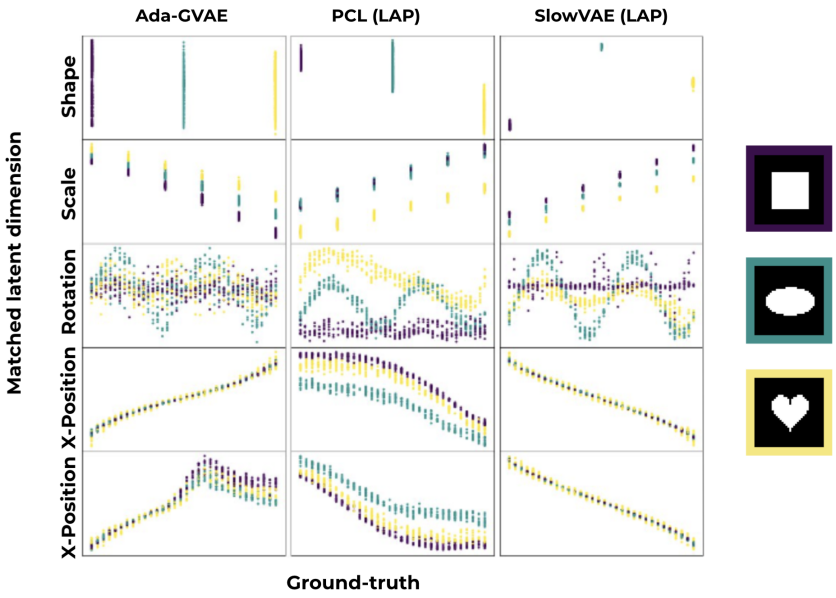


Figure 2.4: **Corresponding latents over ground truth.** For the dSprites dataset (3 samples on the right), we pick the best performing models and visualize the latent spaces over the ground truth. To match latents embeddings and the ground truth, we calculate the Mathews correlation coefficient (MCC) between all latents and the ground truth. Subsequently, we perform Kuhn-Munkres algorithm for a non-greedy matching of latent variables to ground truth factors. Next, we scatter-plot the embedded latent values (y-axis) over the corresponding ground truth values (x-axis) for various samples. We further color-code by the ground truth shape variable, as depicted on the right. In case of an optimal embedding up to permutations and sign flips, all plots should be diagonal. Figure adapted from our paper (Klindt *et al.*, 2021) / Appendix A.

y-position in Fig. 2.4. We also see some limitations for implementing categorical variables in the top row, as no diagonal structure is visible. However, for categorical variables such as shape, there is no order. Thus, a monotonic embedding cannot be expected. Additionally, the shape variable is not fully disentangled from the scale. Lastly, rotations seem to be very difficult for the models. We discuss this further in the next section.

In a nutshell, we advance the field of disentanglement towards more natural data by relying on sparse transitions. We would like to highlight that the sparse transitions are naturally present in the considered video datasets. Thus, our model can be considered *unsupervised* as no annotation is required to infer the underlying latent factors (according to definition of unsupervised learning in Goodfellow *et al.* (2016), p. 105).

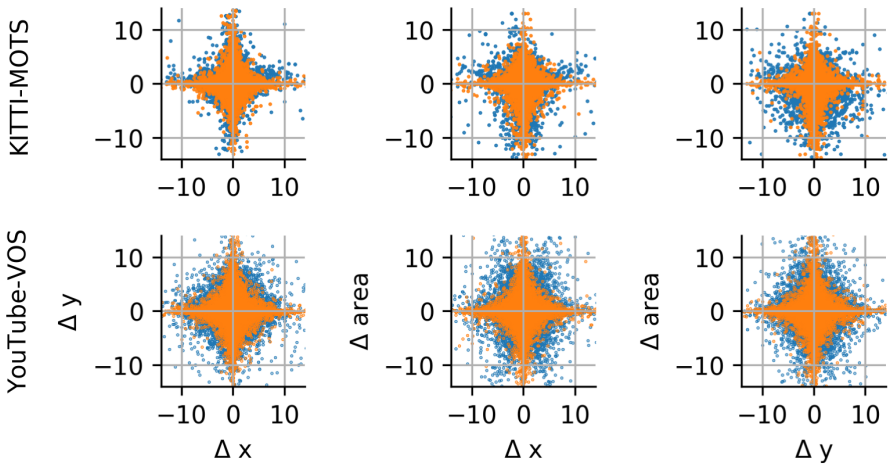


Figure 2.5: **Permuted and natural transitions.** In blue, we show differences in factors for transitions (neighboring frames in video) for pairs of factors. The top row corresponds to KITTI-MOTS and the bottom row to YouTube-VOS. In orange, we randomly permute the pairs to enforce independence. Figure adapted from our paper (Klindt *et al.*, 2021) / Appendix A.

2.1.3 Discussion and outlook

We propose a novel method for provable disentanglement of the underlying latent representation, up to just permutations and sign flips. Here, our method is built on sparse temporal transitions which are present in the data. However, some underlying assumptions are not necessarily true in nature. Here, we discuss those assumptions and propose possible alleviations.

One common assumption in disentanglement that is most probably not true, is that the underlying factors are assumed to be independent. To investigate this, we empirically compare the naturally joint distributions over the transitions in KITTI-Masks and Natural Sprites across the measured factors. As a control with enforced independent factors, we also randomly permute time pairs for individual factors (e.g., combining the y-transitions with x-transitions from another random frame). When comparing the scatter plots of the natural and controlled (=independent) distribution in Fig. 2.5, we clearly see a dependence of the measured natural factors. Surprisingly, in a control experiment on Natural Sprites, our SlowVAE model has a higher performance on the correlated data. All in all, this shows some empirical evidence that our model might not be too reliant upon this independence assumption. On the other hand, a dependence of factors has been shown to be reflected in the latent structure (Träuble *et al.*, 2021). Models that relax the independence assumptions are considered by Khemakhem *et al.* (2020a); Yang *et al.* (2021).

Second, we assume an injective mapping from factors to images and that factors have

non-periodic boundary conditions. These assumptions are, for instance, violated in dSprites (Matthey *et al.*, 2017). Here, three objects are rotated around 360° and have different rotational symmetries: a heart with no rotation symmetry, an ellipse with a 2-fold symmetry, and a square with a 4-fold symmetry, e.g., Fig. 2.4 on the right. Thus, for the square and ellipsis, there is a one-to-many (non-injective) mapping. To relax the injectivity, one way to extend the theory of disentanglement is to allow for such one-to-many mappings is to use probabilistic models. Another approach relies on introducing different definitions of disentanglement. The common definition of disentanglement is based on a one-to-one correspondence of latent factor dimensions to model factor dimensions (Ridgeway and Mozer, 2018; Eastwood and Williams, 2018), is violated in dSprites as the factor of rotation is connected to the object shape through different symmetries. Thus, alternative definitions rely on group representation theory (Bouchacourt *et al.*, 2021; Higgins *et al.*, 2018). We propose that this problem is less relevant in practice, as for more realistic datasets such perfect symmetries are rare. For instance, a small spot or dent on a round object would remove the rotational symmetry. Nonetheless, our SlowVAE still achieves comparably high disentanglement scores and visually shows clear disentanglement for the other factors.

The third assumption that is not covered by our method are factors that do not change. For instance, categorical variables such as the shape of a rigid object like a car are unlikely to change (unless one is watching a transformer movie). Thus, we cannot expect to observe the Laplace shaped transitions that our theory requires for such variables. Nevertheless, we see that our SlowVAE model can separate categorical factors that do not change over time, such as different object shapes in dSprites. However, future work could further inspect this property of fixed factor values from a more theoretical perspective.

Lastly, the common assumption of observing the whole data distribution during training (test=train) is most likely not true for more realistic datasets, as the number of possible combinations of factors scales exponentially with the number of factors. For continuous factors, this would require an infinitive number of samples and is practically infeasible. However, we observe that our SlowVAE shows good empirical results on disentangling almost continuous[2] latent factors on KITTI-Masks. We further investigate more systematic test/train splits in the next section.

In summary, even though some properties of natural data are not covered by our theory, we show that our model performs well empirically. In future work, one could try to expand our theory to cover the violated assumptions described above. From an empirical perspective, we could move towards more unstructured datasets like YouTube-VOS. Here, one could also directly input the raw data instead of the object masks. Moreover, the commonly used neural network architecture in disentanglement is too simplistic. Here, pioneer works by Dittadi *et al.* (2021) showed improvements by simply

[2] We are ignoring pixel artifacts, which make the factors discrete but with $\sim 4000+$ steps s.t. it is almost continuous.

using deeper network architectures. Likewise, with the success of contrastive learning approaches, works combining the scalable contrastive learning-based methods with disentanglement seem promising (Zimmermann *et al.*, 2021).

2.2 Visual representation learning does not generalize strongly within the same domain

Lukas Schott, Julius von Kügelgen, Frederik Träuble, Peter Gehler, Chris Russell, Matthias Bethge, Bernhard Schölkopf, Francesco Locatello, Wieland Brendel.
Published as a conference paper and as a poster at the ICLR 2022.

Author contributions L.S. conceived the idea for the benchmark with inputs from W.B., J.v.K., M.B., B.S., P.G. and C.R.; **L.S.** designed, implemented and performed all experiments with input from J.v.K., F.T., W.B., M.B. and B.S.; F.T. and **L.S.** wrote the publicly released evaluation code; W.B. had the idea for a novel dataset based on a generative model, based on this **L.S.** designed and created the dataset; J.v.K. and **L.S.** wrote the problem setting (section 3) and inductive bias section; W.B., F.L. and **L.S.** wrote the introduction; F.T. and **L.S.** wrote the experimental setup section (section 4); **L.S.** wrote the experiments and results (section 5), related work (section 6), and conclusion and discussion (section 7) with input from F.L. and W.B.; W.B. and F.L. helped with the overall story and revised the paper; F.T. and P.G. performed extensive code reviews.
All section and figure references are w.r.t. to the publication in Appendix B.

2.2.1 Problem: in-domain generalization

In the previous section, we considered a setup in which a world is defined by a few, not directly observed latent factors of variation, which are rendered into observable images by a generative process. Some examples of considered factors of variation are scale, color, rotation, or object type. To provably recover these latent factors, we further assumed that all possible combinations of factors of variation are present in the data and could be inferred from rendered images. In this section, we also view the world as being created by a fixed generative model and aim to recover the latent factors but systematically violate the assumption that all factors are present in the training data by explicitly testing on unseen configurations of factors of variation.

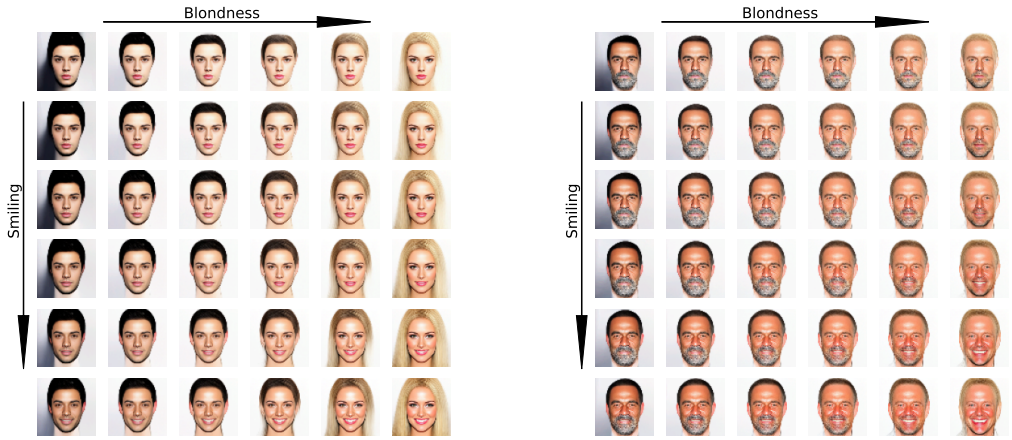
In contrast to other previous out-of-distribution benchmarks, we assure that the corresponding factors are present in the data. For instance, common corruption benchmarks introduce novel corruptions such as snow, blur, or compression artifacts at test time (Hen-

drycks and Dietterich, 2019; Michaelis *et al.*, 2019; Mu and Gilmer, 2019). However, we assure that each factor is partially observed during training. For instance, for an extrapolation of *sizes*, hearts from *small* to *medium* are observed during training, and *large* hearts during test time. Thus, in our benchmark, the model is only required to generalize along the axis of present factors of variation. Furthermore, despite having disjoint test and train data splits along factors, we assume a fixed generative process behind our data. In other words, the generative process is the same during training and test time just with different inputs. Given this assured partial presence of factors in the data and a fixed generative process, we refer to our setup as *in-domain* generalization benchmark.

Our setup is motivated by the fact that it is generally unfeasible to assume that all configurations of factors are realized in the data, as the number of possible combinations scales exponentially with the number of factors. Furthermore, humans can seemingly recognize or imagine novel configurations such as a pink elephant, even though they have never observed it in the world. Moreover, in children's books or mythologies, it is common to stimulate such imaginations by tales of dwarfs and giants (extrapolation), mermaids (interpolation/ composition of fish + human) or a Minotaur (head and tail of a bull and the body of a man). Thus, humans can sometimes make "infinite use of finite means" by abstracting individual concepts and mechanism from the world and reapply them in novel settings (Von Humboldt *et al.*, 1999; Chomsky, 2014). We hypothesize that the procedure of learning to mimic the true mechanism of a factor can greatly help in terms of efficiency and generalization.

Closest to our work, Montero *et al.* (2021) study very similar dataset splits. However, they focus on reconstructions (the decoder), whereas we focus on representation learning (the encoder). Furthermore, they only consider non-identifiable models and one supervised decoder. We extend the model classes to identifiable models, a wide variety of network architectures, and different learning objectives such as un-, weakly-, fully- supervised and transfer learning approaches. Overall, Montero *et al.* (2021) show that the model generalization of the unsupervised generators is limited. We extend this to discriminative models listed above and provide an in-depth analysis of *how* models behave on out-of-distribution data.

In the context of shortcuts and similarly to Funke *et al.* (2021); Zhang *et al.* (2019a, 2018), we propose a correct generalization to our benchmark as a necessary condition towards ensuring that our model has learned the generative mechanism behind the data. For example, simple memorization or patter matching, e.g., a nearest neighbor model, is not feasible and would also not scale gently with the number of factors in most cases. However, a possible solution could rely on learning the true generative mechanism behind the data, as the *same* underlying model is used to generate the test and training data. In general, our setup should favor models that learn *individual mechanisms* behind each factor of variation. On the one hand, this would enable models to extract the factors in a modular fashion. Thus, even if some factors are out of distribution, other factors could



(a) Age 1/6 (young) (b) Age 6/6 (old)

Figure 2.6: **CelebGlow Dataset**. The CelebGlow dataset is created by performing traversals along annotated directions in the latent space of a pretrained Glow network and rendering all possible combinations (Kingma and Dhariwal, 2018). As directions, we consider the common factors blondness, smiling, and age from the CelebA dataset (Liu *et al.*, 2018b). As starting points, we consider random samples from a Gaussian with low standard deviation ($\sigma = 0.3$) to avoid too much prior variation in the factors. Figure adapted from our paper (Schott *et al.*, 2021) / Appendix B.

still be inferred correctly. On the other hand, this also enables models to generalize beyond the current values for a given factor.

Despite the seeming ease for humans, our proposed benchmark is theoretically not solvable only based on the training data. There exist infinitive possible models that correctly fit the training data but would behave differently on the test domain. Thus, we propose certain inductive biases to choose a model and facilitate a correct generalization.

2.2.2 Benchmark of visual representation learning approaches

In this section, we introduce our in-domain generalization benchmark and evaluate various proposed visual representation learning approaches. In particular, we are interested in certain inductive biases which facilitate a generalization beyond the training data. We briefly re-iterate the considered inductive biases from the literature and present their performance on our proposed benchmark.

To create our benchmark, we consider datasets in which all possible factors are available, such as dSprites (Matthey *et al.*, 2017), Shapes3D (Kim and Mnih, 2018), MPI3D

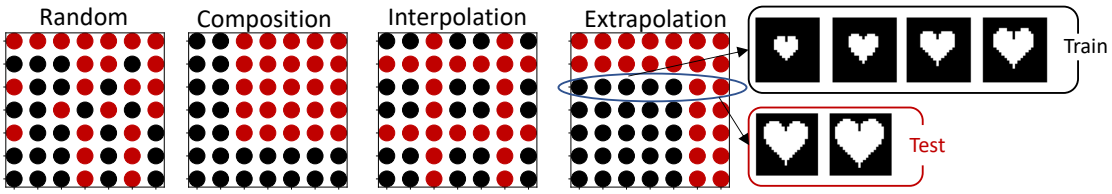


Figure 2.7: **Dataset splits.** We depict the systematic splits random, composition, interpolation and extrapolation for two factors of variations (x-and and y-axis). Red dots correspond to test samples and black dots to training samples. Examples of corresponding observations are shown on the right. Figure adapted from our paper (Schott *et al.*, 2021) / Appendix B.

(Geirhos *et al.*, 2018) and our contributed dataset CelebGlow, which is based on CelebA (Liu *et al.*, 2018b) and the Glow network (Kingma and Dhariwal, 2018). The dataset is depicted and described in Fig. 2.6. The goal is to infer all factors of variation given an image, even though they only have been partially observed during training. For all datasets, we consider four different types of splits along the factors of variation behind the data. The first type consists of a random test-train split, which is frequently used in various computer vision datasets. The other types are more systematic and referred to as *interpolation* (e.g. *train*= small and large objects $\rightarrow$ *test*= medium-sized objects), *extrapolation* (small and medium-sized objects $\rightarrow$ large objects), and *composition* (big hearts and small squares $\rightarrow$ big squares and small hearts). This procedure is depicted in Fig. 2.7.

The first considered inductive bias is different representational formats from disentanglement algorithms. For instance, certain weakly supervised algorithms rely on pairs of images with a certain conditional structure and provably allow for an identification of latent factors up to some unavoidable ambiguities. Depending on the model, the identification can range from pointwise nonlinearities or linear transformation up to just permutations and sign flips. However, those guarantees only hold for the training set, and it remains unclear whether they can generalize to a disjoint test set. More concretely, we test β-VAE (Higgins *et al.*, 2017a), PCL (Hyvärinen and Morioka, 2017), Ada-GVAE (Locatello *et al.*, 2020b) and our SlowVAE model (Klindt *et al.*, 2021). Here, each network is trained as proposed by the authors to match the corresponding assumptions.

The second inductive bias we investigate are various architectural designs, which mostly rely on incorporating certain symmetries that are present in the world (Noether, 1915; Higgins *et al.*, 2018). In machine learning, for instance, convolutional neural networks (CNNs) naturally incorporate equivariance to shifts in the image. Here, relying on local and reusable filters to extract reoccurring patterns such as edges has tremendously boosted neural network performances on image-based tasks such as classification performances (LeCun *et al.*, 1999). Similar implementations can be found for rotation and scale equivariance (Fasel and Gatica-Perez, 2006; Xu *et al.*, 2014) or pixel/coordinate

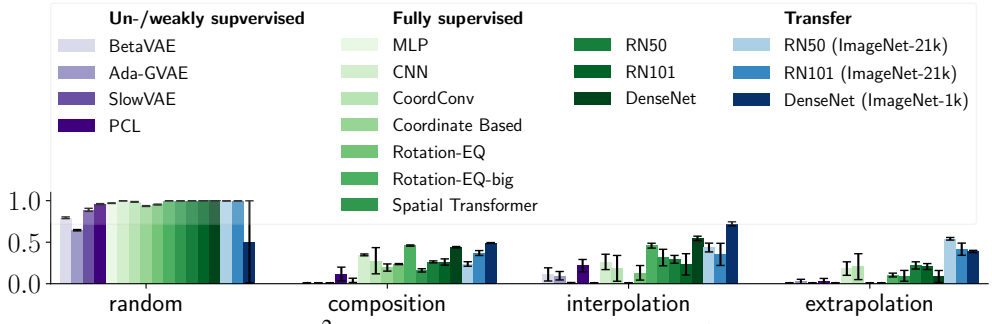
permutations (Achlioptas *et al.*, 2018; Zhang *et al.*, 2019b). Additionally, the network architecture can be used to facilitate the learning of abstract and low-level concepts in images (Huang *et al.*, 2017; He *et al.*, 2016). While powerful theoretically, in practice these invariances often have to be known beforehand to be incorporated into the network architecture. Furthermore, for certain invariances such as a rotation in 3D projected onto 2D, it is not possible to "hard-code" the transformation before seeing the data, i.e., as the back of an object is unknown. In our study, we consider MLPs, CNNs, CoordConv (Liu *et al.*, 2018a), Rotationally-Equivariant (Rotation-EQ) CNNs (Cohen and Welling, 2016), Spatial Transformers (STN) (Jaderberg *et al.*, 2015), ResNet (RN) 50 and 101 (He *et al.*, 2016), and DenseNet (Huang *et al.*, 2017). All networks are trained to directly predict the FoVs $y = f(x)$ in a fully supervised fashion.

The third and last inductive bias that we consider simply relies on leveraging transferable structures from other tasks. In computer vision, large image corpora, consisting of more than 14 million (Deng *et al.*, 2009) or even more than 300 million (Sun *et al.*, 2017) labelled images, are used to pretrain neural networks. Subsequently, the neural networks are then "fine-tuned" on a particular task. This procedure has turned out to be quite effective in various settings (Chen *et al.*, 2020a; Xie *et al.*, 2020), often even outperforming task-specific designed architectures (Sutton, 2019). Here, we fine-tune a DenseNet pretrained on ImageNet-1k, and a Renset50 and 101 pretrained on ImageNet-21k.

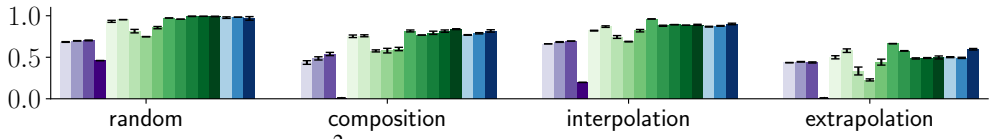
We train multiple random seeds for each architecture described above on each test-train split (random, composition, interpolation, and extrapolation). On the random test-train split, we see that almost all approaches perform well. The R^2-scores are depicted in Fig. 2.8. One exception are smaller models on the complex CelebGlow dataset that has 1000 categorical variables and might require large capacity networks like the RN50 and bigger. For systematic splits, in which a model is required to interpolate, extrapolate, or compose factors of variation, we observe large drops in performances. Overall, no tested model can achieve high scores on all systematic splits, regardless of the architecture and supervision signal. Especially, for extrapolation, we observe the lowest performances. We conclude that models struggle to learn the underlying mechanisms behind the data and seem to rely on mechanisms that do not generalize well. This effect is most pronounced on MPI3D.

One notable exception are the good model performances on the Shapes3D interpolation and composition setting. In Shapes3D, the factors of variation mostly consist of colors for different objects. Thus, for interpolation, we hypothesize that most models generalize well, as they use ReLU activation functions that are linear for positive inputs. Thus, might help to correctly interpolate colors. Also, the dataset is fairly modular in the sense that specific factors can be inferred from fixed positions in the images. E.g., the factor "wall color" could be inferred from the same background pixels across different images.

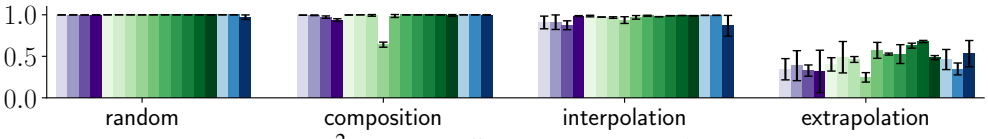
We also tested the modularity of the models by measuring the performances on in-distribution (seen during training) factors while other factors are out-of-distribution. For



(a) R^2-score on MPI3D-Real test splits.



(b) R^2-score on CelebGlow test splits.

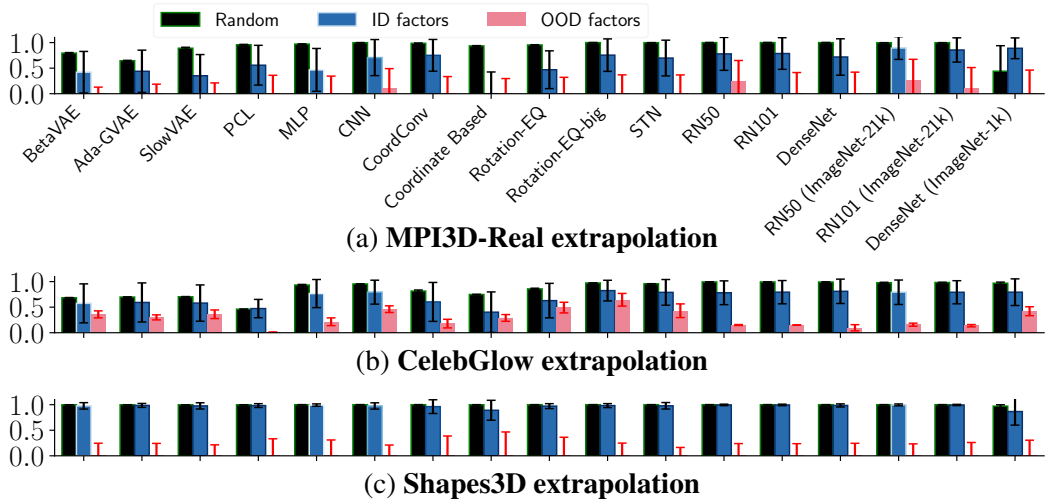


(c) R^2-score on Shapes3D test splits.

Figure 2.8: **R^2-score on various test splits.** On the OOD splits composition, interpolation, and extrapolation, we observe large drops in performance compared to the in-distribution random splits. Figure adapted from our paper (Schott *et al.*, 2021) / Appendix B.

instance, we require models to infer previously observed object orientations at unseen scales. In contrast to common criticisms of deep neural networks (Csordás *et al.*, 2021; Greff *et al.*, 2020; Lake and Baroni, 2018), we find that the models are fairly modular, as in-distribution factors are still inferred well even if other factors are out-of-distribution. This can be seen by comparing the random performances with the ID factors in Fig. 2.9. By the procedure of exclusion and by observing the low out-of-distribution (OOD factors) performances in Fig. 2.9, we can conclude that the errors are mostly due to incorrectly inferred OOD factors. Thus, we further investigate the behavior of the models on the OOD factors.

Despite using variously different inductive biases and observing limited generalization capabilities, we found that the models still make surprisingly similar mistakes. To show this, we measured the Pearson correlations between different models on out-of-distribution factor predictions. We find that the models have high positive correlations on MPI3D, Shapes3D, and dSprites (all Pearson $\rho \geq 0.57$) but negatively correlate with the ground-truth (all Pearson $\rho \leq -0.48$). One exception is the CelebGlow dataset. Here some positive correlations with the ground truth model are observed (all Pearson $\rho \leq 0.50$), but are nonetheless still far from extrapolating correctly. To further quantify the generalization behavior on the out-of-distribution factors, we aggregated the predic-



(a) **MPI3D-Real extrapolation**

(b) **CelebGlow extrapolation**

(c) **Shapes3D extrapolation**

Figure 2.9: **Extrapolation and modularity, R^2-score on subsets.** To investigate the root of the extrapolation errors in Fig. 2.8, we split the performance along individual factors and distinguish between factors that have been observed during training (ID factors) and the ones that have not been observed (OOD factors). Note that because this is still based on the test set, the joint of different factors have not been observed during training. We refer to the text for further details. Figure adapted from our paper (Schott *et al.*, 2021) / Appendix B.

tions on the test set and found that models tend to predict values in previously observed ranges. For more details, we refer to the paper in Appendix B.

We conclude that our proposed out-of-distribution task is still mostly unsolved, and current models pursue strategies which do not generalize systematically. Finding inductive biases suited to this type of generalization remain, to the best knowledge of the author, still an open research question.

2.2.3 Discussion and outlook

In this section, we first discuss our benchmark in a broader context and subsequently the implications of our benchmark results on future model development.

Limitations and positioning of our Benchmark

We introduce a computer vision benchmark that requires neural networks to recombine, interpolate, or extrapolate existing factors from the training data. This contrasts with previous works that introduce novel factors such as artificial rain or blur at test time (Hendrycks and Dietterich, 2019; Mu and Gilmer, 2019; Michaelis *et al.*, 2019). Thus,

our work could be seen as an intermediate and more principled milestone on the path towards universal out-of-distribution generalization in visual representation learning.

Our benchmark requires models to learn all possible configurations of factors. However, in the real world, this might not necessarily be a desired property. In safety critical environments (Leike *et al.*, 2017) such as traffic datasets (Cordts *et al.*, 2016), not all factor combinations are valid. For instance, in a one-way street, a car pointing in the wrong direction should not necessarily be allowed. We argue, however, that for humans it is possible to imagine driving in the wrong direction or even thinking of a pink elephant on the road. Thus, such a scenario could be present in a model, but one should definitely be aware of the fact that such a scenario is most likely imaginary or not allowed.

As our benchmark relies on simple datasets (dSprites, Shapes3D, MPI3D, CelebGlow), a possible extension could introduce more unstructured real-world scenarios. Here, we require a larger labelled dataset where multiple factors are annotated along each axis.

Furthermore, on an abstract level, it remains unclear which factors can and should be extrapolated. Even for humans, certain concepts are very difficult to grasp and extrapolate. For instance, the concept of dimensionality is simple to visualize for 1D with a stick, for 2D with a sheet of paper, and 3D with our world. However, 4D and higher dimensions are very difficult to visualize. Even, the concept of numbers is not necessarily reflected in languages. For instance, 139 Aboriginal Australian languages have an upper limit at "three" or "four" and "several" or "many" to refer to higher quantities (Barras, 2021; Bowern and Zentz, 2012). It has been further claimed that the Pirahã people of the Brazilian Amazon might not use numbers at all (Everett, 2005).

Implications of our benchmarks for further model research

We reveal strong limitations in various visual representation learning networks to generalize factors even if they are present in the data. On the other hand, we observe that most of our tested models are quite modular in the sense that even if one factor is out-of-distribution, other in-distribution factors are still inferred correctly.

Our benchmark points in the research direction of independent (causal) mechanisms. We hypothesize that learning the underlying mechanism of a dataset should allow for a scalable generalization to our benchmark. Thus, the true *mechanism* should be able to extrapolate certain factors and the *independence* should allow for a scalable recombination of arbitrary factors. We note that such an independence can only be achieved approximately. For instance, the scaling of an object also depends on the object itself. E.g., when considering an elephant that should be scaled up, more specific fine-grained structures such as hairs or skin structures are revealed and have to be known by the model.

Another possible approach to perform well in our benchmark is to engineer network architectures that by design encode certain transformations such as rotation, shifts, or

scales. However, as the number of factors of variation is almost infinitive in the real world, approaches that hard-code the type of transformation might not scale to more realistic scenarios in which more factors, such as varying lighting conditions, are present.

2.3 A simple way to make neural networks robust against diverse image corruptions

Evgenia Rusak*, Lukas Schott*, Roland S. Zimmermann*, Julian Bitterwolf, Oliver Bringmann°, Matthias Bethge°, Wieland Brendel°.
Published as a conference paper and as an oral at the ECCV 2020.

*Joint first authors / equal contribution, °joint senior authors

Author contributions W.B. and M.B. developed the idea of the adversarial noise training; E.R. and W.B. conceived a realization with input from **L.S.** and R.S.Z.; E.R., **L.S.**, R.S.Z. and W.B. designed and performed the experiments on ImageNet-C with input from W.B. and J.B.; **L.S.**, E.R. R.S.Z. and W.B. designed and performed the experiments on MNIST-C; **L.S.**, R.S.Z., E.R. and W.B. designed and performed adversarial robustness evaluation; E.R., **L.S.**, R.S.Z. and W.B. wrote the manuscript with input from J.B., O.B. and M.B.; E.R. designed figures 1-3 with input from **L.S.**, R.S.Z. and W.B.; R.S.Z. designed figure 4 with input from E.R., **L.S.**, and W.B..
All section and figure references are w.r.t. to the publication in Appendix C.

2.3.1 Problem: common corruptions

So far, we considered settings that assume all factors of variation are fully (Section 2.1) or partially (section 2.2) presented in the training data. In this section, we allow for variations that have not necessarily been observed during training. More concretely, we consider *common corruptions* that are introduced during test time.

Common corruptions in computer vision often refer to digital artifacts or weather conditions. Typical weather conditions are rain, snow, or fog. Usual artifacts in digital systems are blur, Gaussian noise, or compression artifacts such as pixelating. Current machine learning algorithms are quite brittle and not robust w.r.t. corruptions added at test time. This has been shown for common datasets with corruptions (denoted by -C) such as ImageNet-C, CIFAR-C, MNIST-C, Cityscapes-C, ... (Hendrycks and Dietterich, 2019; Mu and Gilmer, 2019; Michaelis *et al.*, 2019). This can have tremendous implications on the reliability of many camera-based systems in the real world based on machine learning, especially for safety-critical systems. For example, as those corruptions are difficult to account for during development, lacking robustness might be one of the reasons why Waymo, one of the leading companies in autonomous driving, launched their ser-

vices in the weather-friendly Phoenix, Arizona (Davies, 2017). In general, it is desired to develop machine learning algorithms that are robust to common corruptions. Thus, we are trying to find suitable inductive biases to foster the generalization capabilities of neural networks.

Here, we study the effect of common corruptions on machine learning systems in the context of simple image classification tasks. The goal is to learn a mapping from an image to a corresponding label on clean data which consists of many images and their corresponding labels. After training, a classifier should be able to classify objects, i.e., a bird, despite common corruptions added to the image. In contrast to the previous sections, here, the classifier has not necessarily seen these corruptions during training. It should be robust and reliable *despite* never having seen such a corruption. This is not the case for standard neural network classifiers, which can drop to half of their original accuracy (Hendrycks and Dietterich, 2019).

In terms of shortcuts, it has been hypothesized that standard neural networks rely on spurious features or correlations in high dimensional data (Ilyas *et al.*, 2019; Geirhos *et al.*, 2020). Here, we further conjecture that these shortcuts are quite brittle and easily broken by common corruptions. Next, we are trying to develop a principled method that iteratively detects and removes such shortcuts in neural networks until it ends up relying on more robust strategies.

Common previous approaches rely on augmenting the input data with pre-computed images that should help the network transfer to common corruptions. For instance, Geirhos *et al.* (2019) stylize images to remove texture cues, pushing a network to rely more on shape cues. Similarly, Hendrycks *et al.* (2020b) rely on a set of pre-computed augmentations to train a more robust neural network. Both methods have been shown to increase common corruption robustness. Adversarial training methods, on the other hand, adaptively compute the worst-case perturbations for a given classifier (Madry *et al.*, 2018). Here, the considered types of perturbations are much more complex and usually much smaller in magnitude. Our approach is in between these listed approaches: We constrain the search space of possible corruptions and compute our perturbations adaptively.

2.3.2 Approach: adversarial noise training

One way to tackle common corruptions is by simply adding them to the training data. However, this might not scale well, as a network trained on one corruption might not necessarily be robust to other types of corruptions. For instance, training on rainy images does not generally improve the robustness to snow, fog, or very light rain. Surprisingly, this transfer might not even work for two types of corruptions, which are visually almost indistinguishable (Geirhos *et al.*, 2018). As the list of possible artifacts is almost limitless, we try to find a more principled approach.

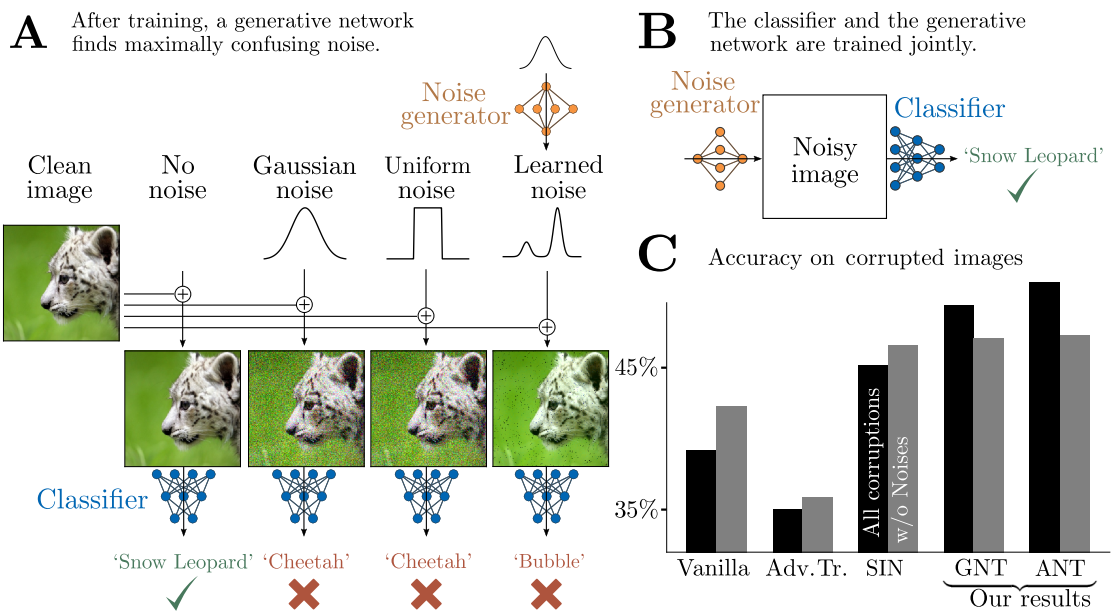


Figure 2.10: **Overview of our approach.** In **A**, we show how various noises added onto an image can derail a classifier. Notably, our learned noise is much smaller in magnitude. In **B**, we depict our architectural setup, in which we jointly train a noise generator and a classifier. After training the classifier should be robust w.r.t. the generated noises. Finally, in **C**, we see an improvement over previous methods even on non-noise categories on the ImageNet-C dataset. Figure adapted from our paper (Rusak *et al.*, 2020) / Appendix C.

To introduce our approach, we investigate properties of common corruptions. Common corruptions can, in general, be fairly structured. Snowflakes, for instance, are individually quite unique but share many common attributes in structure across snowflakes. Roughly spoken, they are only locally dependent, e.g., as snowflakes are quite symmetric, knowing one half of a snowflake could be quite informative about the second half. However, one snowflake might not be too informative for the position or exact structure of other snowflakes. Thus, abstractly, snowflakes could be viewed as different samples from a local, shared distribution. We mimic this structure by augmenting images in the training data with noise sampled from a learned distribution that is constrained to only allow for local correlations.

Practically, we implement this learned local noise distribution by transforming a Gaussian distribution with a convolutional neural network architecture that only has small or 1×1 kernel sizes. We refer to this network as the noise generator. In the next step, this noise is added to an image. To make this procedure more adaptive to the classifier, we adversarially train the noise generator, s.t. the noise is most severe for a given classifier. We further constrain the noise to be small in magnitude to avoid trivial strategies such as simply masking the whole input. We refer to this noise as *adversarial noise*. Subse-

quently, in *adversarial noise training* (ANT), we alternate between training the classifier and training the noise generator as inspired by Goodfellow *et al.* (2014); Schmidhuber (1992). To stabilize this min-max optimization procedure, we further leverage common training procedures like experience replay (Mnih *et al.*, 2015). The overall procedure is visualized in Fig. 2.10.

Intuitively, from the perspective of shortcuts, the noise generator detects and masks the current shortcuts used by the neural network by learning the most adversarial noise. In the subsequent network update step, the network tends towards leveraging a different strategy. Upon convergence of this alternating training scheme, the network should no longer rely on shortcuts that can be masked by small locally correlated noise. Subsequently, it should be more robust w.r.t. to common corruptions that are also locally correlated and small in magnitude.

We carefully evaluate our proposed adversarial noise training. First, we show that our learned noise is significantly more severe compared to other noise distributions. When added to an image, it can change a ResNet50 classifier decision by adding noise that is significantly smaller (1/2 or less) in magnitude than Gaussian or uniform noise. Second, we evaluate on 15 common corruptions with five different severities from the MNIST-C and ImageNet-C dataset. As a baseline, we additionally train on images augmented with Gaussian noise with several magnitudes. We find that this simple baseline is already quite effective and outperforms previous methods relying on patch-based noise training. Lastly, we find that our proposed adversarial noise training scheme further increases the performance on the noise as well as on the non-noise corruptions. As our method solely relies on adaptive augmenting of the input, it can be combined with other methods. When combining our method with training on stylized images (Geirhos *et al.*, 2019), which makes neural networks more reliant on shape queues and less reliant on textures, we achieve at-the-time state-of-the-art performance.

2.3.3 Discussion and outlook

We propose a principled method to make an image classification neural network robust w.r.t. to locally correlated noise distributions. We further show a successful transfer to the common corruption dataset of ImageNet-C. Here, we discuss possible extensions and limitations of our approach.

A possible extension of our approach is to further empower our noise generator. So far, we only considered locally correlated additive noise that is independent of the image. We observe that our proposed approach is not state-of-the art on image-dependent corruptions such as defocus, motion, and blur corruptions. Thus, we could also input local patches of the images to the noise generator. This could enable the noise generator to also mimic the image-dependent corruptions.

Another extension is to further allow the generator to produce perturbations in the frequency space. This could enable it to better model low-frequency perturbations that have been shown to be very effective in derailing various neural networks (Sharma *et al.*, 2019). An alternative way to allow for more general perturbations introduced by Calian *et al.* (2021), is to use pertained image-to-Image transfer networks to generate adversarial perturbations. To avoid a trivial solution that sets all pixels to zero, they bound the parameters of their image-to-image transfer network. However, so far, a direct comparison with our approach is not possible as they only consider a downscaled version of ImageNet.

A general limitation of common corruption benchmarks like ImageNet-C is that they only provide results for a subset of corruptions. Hence, it remains unclear whether the measured accuracies on the artificially corrupted images are representative for other corruptions. Even for the rain corruption, it is not guaranteed that the accuracies on artificial rain from ImageNet-C are predictive for the network performance for real rain. Supporting evidence for such doubts shows that neural networks can have highly varying performances on corruptions that look almost indistinguishable to humans (Geirhos *et al.*, 2018). Here, further research on validating the ImageNet-C results as an indicator of a network's out-of-distribution performance is required.

2.4 Towards the first adversarially robust neural network model on MNIST

Lukas Schott*, Jonas Rauber*, Matthias Bethge°, Wieland Brendel°.
Published as a conference paper and as a poster at the ICLR 2019.

*Joint first authors / equal contribution, °joint senior authors

Author contributions M.B. and W.B. had the idea for analysis-by-synthesis for adversarial robustness; **L.S.** and W.B. conceived a VAE based realization with input from J.R. and M.B.; **L.S.** W.B. and J.R. designed the confidence calibrated softmax; J.R. performed an extensive robustness evaluation of the model with input from **L.S.** and W.B.; J.R. contributed the Pointwise Attack; **L.S.** contributed the Latent descent Attack; **L.S.** performed the distal adversarial attacks; W.B. and **L.S.** conceived and implemented the lower bound for the robustness; W.B., **L.S.** and J.R. wrote the manuscript with input from M.B.
All section and figure references are w.r.t. to the publication in Appendix D.

2.4.1 Problem: adversarial examples

So far, we always restricted the types of generalizations along factors that partially present in the training data, or to common corruptions. Now, we consider a broader scenario and focus on maliciously crafted perturbations, also called *adversarial examples*, that change the predictions of a neural network.

In computer vision, adversarial examples are images with small, but almost imperceptible perturbations that can severely sabotage the output of a neural network (Szegedy *et al.*, 2014). For instance, only a few pixels need to be changed such that a classifier can no longer correctly detect handwritten digits (Schott *et al.*, 2019). Finding the most malicious perturbations for a classifier can be quite difficult and is a research field of its own (Carlini *et al.*, 2019; Wiyatno *et al.*, 2019). Nevertheless, for an undefended network, such examples are almost omnipresent: For nearly every input image, there exists a perceptually close image that is misclassified. As neural networks have often been compared to the human visual system, which seems to be robust w.r.t. adversarial examples[3], the vulnerability of artificial neural networks undermines our current understanding.

[3]In a recent publication Elsayed *et al.* (2018) find adversarial examples that fool neural networks and *time-constrained* humans. However, they also conclude that adversarial examples have bigger effects on artificial neural networks than humans.

Despite tremendous research efforts and various publications, so far, there is still a large gap in robustness. For instance, on the more complex ImageNet dataset, the robustness guarantees are still weak and mostly restricted to predefined perturbations bounds such as specific norms. Here, we argue that even the simple MNIST digit classification benchmark is not solved from the viewpoint of human visual robustness, even for state-of-the-art robustness classifiers.

It has been shown by Ilyas *et al.* (2019) that adversarial examples leverage spurious features (shortcuts) used by neural networks and can actually represent predictive features. To show this, the Ilyas et al. propose a setup with three stages. First, they create a dataset that only consists of adversarial images that fool a pretrained network into predicting another class. For each adversarial image, they store the corresponding network predictions as new labels, which –as adversarial examples are almost imperceptible– are different from what a human observer would predict. In a second step, they train a second neural network from scratch, only on adversarial examples that fooled the previous network with the seemingly wrong labels. For instance, an adversarial image of a dog that is perturbed s.t. a network labels it as a cat is now passed into the second network but, counterintuitively, labeled as cat. In the last step, this newly trained classifier is tested on new clean, unperturbed images. Now, despite only being trained on seemingly mislabeled images, it can correctly classify clean images with high accuracy, mostly agreeing with human annotators. Thus, the authors literally conclude "adversarial examples are not bugs, they are features" meaning that neural networks rely on non-robust but highly predictive features that are present in the data.

In the next section, we try to develop a method that avoids such shortcuts and learns a more robust solution.

2.4.2 Approach: analysis by synthesis

One suggested inductive bias to stimulate a model to avoid shortcuts and learn a diverse set of features is based on the principle of ABS. Here, novel images are classified if they are synthesized –a mapping from labels to images– correctly and matched. By requiring the model to generate whole images, shortcuts should be avoided. For instance, to recognize an image of a cat, multiple visible properties such as the eyes, ears, and tail have to be synthesized for a likely match. This contrasts with typical feedforward neural network architectures that learn a mapping from images to labels, which could rely on shortcuts. For instance, in the example with the cat, it could be sufficient to recognize a cat solely by considering the eyes and ignoring other features like the ears. Thus, such a network would not be robust if the cat eyes are not visible anymore during test time. In this scenario, the ABS-based classifier could still rely on other features such as the ears as it is trained to synthesize whole images.

To implement and evaluate the proposed inductive bias of ABS (de Cordemoy, 1973;

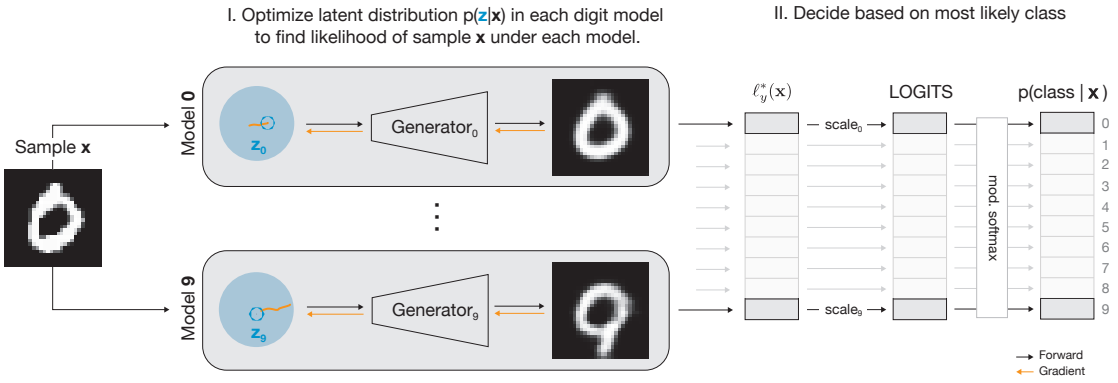


Figure 2.11: **Overview of our approach.** We first train one VAE for MNIST digit class and only keep the generators. To classify a sample digit, we perform a gradient guided search in the latent of each space to find the most likely match, as shown in **I**. Next, in **II**, we scale the likelihoods with a scalar to account for class imbalances to get logits similar to a standard classifier. Lastly, to determine the label and the posterior probability, we pass the logits through a confidence calibrated softmax function and take the argmax. Figure adapted from our paper (Schott *et al.*, 2019) / Appendix D.

Von Humboldt *et al.*, 1999), we consider the simple and commonly used MNIST digit classification task. As a model, we rely on VAE (Kingma and Welling, 2014), which allow us to do approximate posterior inference by estimating the evidence lower bound (ELBO). We split the training dataset into images corresponding to each class and train one VAE per class c to approximate the likelihood under each model $p(x|c)$. After training, we disregard the encoder and only keep the ten generators, one per digit class from 0-9. To classify a new test image, we perform a gradient guided search in the latent space of each generator to find the closest match (in terms of likelihood). Thus, given a test image, for each class, we get one proposal for the likeliest corresponding image and its likelihood. Additionally, we discriminatively learn a scalar to scale the class conditional likelihood of each model. We pass each weighted likelihood to a calibrated softmax function (we refer to our paper for details, Appendix D) to receive an approximate posterior $p(c|x)$. Now, the digit model corresponding to the highest probability refers to the class. This is also depicted in Fig. 2.11

We consider various baselines to compare our approach to. Other established methods designed to be robust to adversarial examples mostly rely on randomization (Cohen *et al.*, 2019), using the Lipschitz constant (Hein and Andriushchenko, 2017; Lecuyer *et al.*, 2019) or adversarial training (Madry *et al.*, 2018). Some of these methods allow for mathematically provable claims about the robustness of a model in a certain vicinity of its input. Currently, adversarial training is one of the most prominent methods (Xie *et al.*, 2019). Here, we alternate between estimating adversarial examples for a classifier and training the classifier to be robust w.r.t. the adversarial images. If carefully tuned,

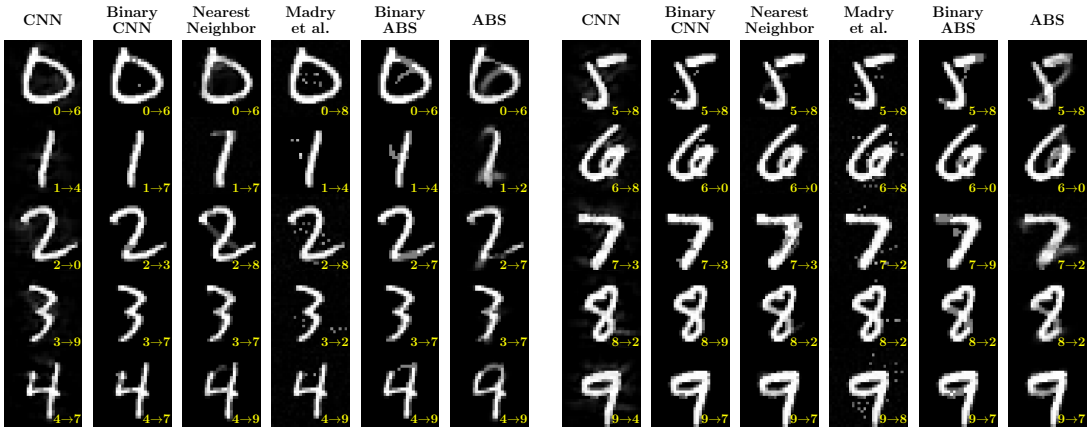


Figure 2.12: **Adversarial examples**. Here, we show the lowest found L_2 perturbation for random samples for each model. The adversarial examples for our ABS model can appear close to the perceptual boundary of humans. More random samples on different L_p norms and sampling strategies are shown in Appendix D. Figure adapted from our paper (Schott *et al.*, 2019) / Appendix D.

this procedure can be quite effective and provides high robustness on the considered norm for the attacks. We use the model of Madry et al.[4] trained on the L_∞-norm as a baseline to compare our model with. Other baselines we consider are a standard (vanilla) convolutional neural network with and without input binarization. The input binarization simply thresholds input pixels above 0.5 to 1 and the rest to 0 (we preprocess images to be in the range $[0, 1]$). Thresholding does not change the image too much, as MNIST is almost binary. Lastly, to provide an elementary baseline, we also provide a nearest neighbor classifier.

We show empirical robustness results on MNIST on the L_0, L_2 and L_∞ norm. We go to great lengths in using multiple attacks to provide a precise estimate of the model robustness (more details regarding the attacks are provided in the next section). Our ABS model performs favorably over a vanilla CNN on all considered L_p norms, and even provides state-of-the-art results on the L_2 norm, despite never having seen any perturbed images during training (see Figs. 2.12 and 2.14 for quantitative and qualitative results). Our results show that the at-the-time state-of-the-art defense of Madry *et al.* (2018) does not solve MNIST from a *general* viewpoint of robustness.[5] Despite having high robustness in terms of L_∞, their model performs worse compared to a vanilla CNN in terms of L_0 robustness, as shown in Fig. 2.14 on the right. Furthermore, we observe that the L_∞ results of Madry et al. can partially be reproduced by simply binarizing the input of a

[4] At the time of publication, this was the acknowledged state-of-the-art adversarially robust neural network on MNIST

[5] We would like to emphasize that Madry et al. do not make any claims on the robustness beyond L_∞. We evaluate it on other norms to investigate the overall robustness on MNIST and explore side effects.

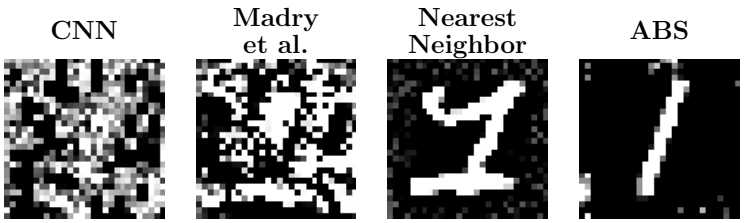
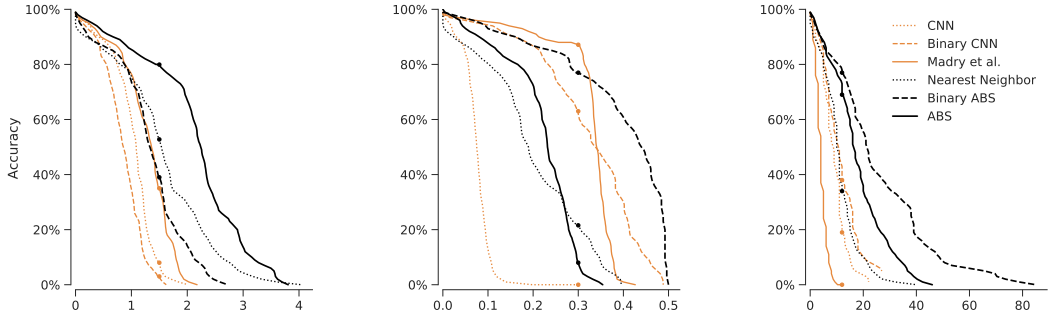


Figure 2.13: **Distal adversarials.** We perform gradient ascent on the pixels of a random noise input image until $p(c = 1|\mathbf{x}) \geq 0.9$ is true for each model. For the CNN and the adversarially trained model of Madry et al., these images are not recognizable. The nearest neighbor, simply walks to the nearest digit, whereas our model shows a prototypical representative of the class. Figure adapted from our paper (Schott *et al.*, 2019) / Appendix D.



(a) L_2 distance (b) L_∞ distance (c) L_0 distance

Figure 2.14: **Adversarial distortion curves**. Accuracy over the allowed perturbation budget for each L_p-norm. Figure adapted from our paper (Schott *et al.*, 2019) / Appendix D.

vanilla convolutional neural network.[6] Lastly, humanly unrecognizable images, referred to as *distal adversarials*, are classified as a certain digit with high confidence by the tested feedforward architectures, as demonstrated in Fig. 2.13 on the left.

To motivate our results on lower-bounding[7] the robustness, we would like to highlight that our likelihood function in pixel space given the latents z of a model $p(x|z)$ is a Gaussian, which in our case is the most predominant term in the evidence lower bound (ELBO) of Kingma and Welling (2014). Crucially, this Gaussian likelihood only changes

[6] Madry et al., concurrently updated their paper and show that their model learns a threshold operation in the first layer of their model.

[7] Our conservative estimate includes an optimization problem in a low-dimensional space. We sample extensively and provide multiple steps of gradient descent to solve this low-dimensional optimization problem. We also do multiple random restarts and show that this procedure works very reliably. Nevertheless, we have not guaranteed for the exactness of the result, and, thus, avoid the term *provable*.

slowly if we have small (in terms of L2 distance) changes in the image. This is a desired property of our model, as the class of a digit is also unlikely to change if we minimally change the image. We follow this principle and mathematically derive a computable lower bound for the robustness which, at-the-time, was comparable to several provable robustness claims (Hein and Andriushchenko, 2017). Now, such provable claims are close to empirical claims on MNIST (Li *et al.*, 2019a).

In a nutshell, we show on MNIST that L_∞ robust models can have severe drawbacks. We show that our proposed ABS model is robust to multiple threat scenarios compared to a vanilla feedforward network and leads to state-of-the-art robustness on the L_2 norm. Also, compared with other tested methods, ABS has plausible distal adversarials.

2.4.3 Evaluation of adversarial robustness

As evaluating robustness empirically has many pitfalls (Athalye *et al.*, 2018; Carlini *et al.*, 2019; Croce and Hein, 2020), we consider a large variety of proposed attacks and introduce novel attacks to fill gaps in the attack literature. Common pitfalls are masked or obfuscated gradients, or to simple attack models. Hence, to properly evaluate a model, a diverse set of attacks should be considered. From the literature, we consider gradient-based, gradient-estimating, score-based and decision-based attacks to evaluate the adversarial robustness of our models. Furthermore, as we strive for models that are robust across multiple threat models, we examine a variety of different norms, namely L_0, L_2, and L_∞. As, at the time, no reliable L_0 attacks existed and few specific attacks for class-conditional generative models were available, we develop two novel attacks: the *Pointwise Attack* and the *Latent Descent Attack*.

Pointwise Attack: Due to the discrete nature of L_0, most previous attacks are not applicable to this norm. Thus, we introduce a novel decision-based attack that greedily minimizes the L_0 norm. Our attack starts by adding salt-and-pepper noise to an image until it is misclassified. Subsequently, it simply iterates over all pixels and tries to reset the perturbed values to the original values. If the image is still misclassified, the pixel is set to the original value, otherwise the pixel is kept perturbed. Finally, if no more pixels can be reset, the attack ends and returns the adversarial image. To achieve optimal results, we re-run our attack with 10 random initializations, which can be computed in parallel. Compared to simply adding salt-and-pepper noise, our attack is $3x - 10x$ more effective in achieving a minimal median L_0-perturbations in our experiments (for details, we refer to Table 1 in Appendix D). Now, there are more effective attacks available (Croce *et al.*, 2020). However, as they rely on gradients, the Pointwise Attack remains useful as it is applicable to models that only output decisions, and it is also not affected by obfuscated (misleading) gradients.

Latent Descent Attack: Class-conditional generative models often do not provide analytic gradients. As an alternative, we exploit the structure of the ABS architecture, which

provides an image for the most likely candidate under each model for a given image. We select the generated image corresponding to the most likely wrong class. Given this image and the input image, we perform a binary search by linearly interpolating between the two images to find an adversarial image. As the ELBO is dominated by the reconstruction loss, this procedure works reliably in practice. We show in our paper that our latent descent attack performs almost on par with the most successful attacks for our model, namely the Boundary Attack (Brendel *et al.*, 2018) and the DeepFool attack (Moosavi-Dezfooli *et al.*, 2016) with gradient estimation. Especially, our attack performs preferably on different samples than other attacks. Hence, we can combine both attacks to find a tighter upper bound on the true robustness of a model.

2.4.4 Discussion and outlook

Here, we highlight and discuss our main contributions in a broader context. Thereby, we focus on the importance of our used Gaussian likelihood and the difficulty of properly evaluating the robustness of a model.

We claim that MNIST might not be solved from a point of human-like robustness. We support this claim by showing limitations in transferring the L_∞-robust model of Madry *et al.* (2018), which is a widely accepted defense method, to other norms. Relying on the principle of Analysis by Synthesis (ABS), we provide a model that is more robust than a vanilla network across multiple norms and even shows a widely acknowledged state-of-the-art performance on L_2 (Jacobsen *et al.*, 2019; Ju and Wagner, 2020). We further provide some qualitative evidence that the decision boundaries of our model are perceptually close to the humans. This claim has been substantiated by Golan *et al.* (2020), who quantitatively show that controversial stimuli for our model and humans have higher agreement compared to other models.

Since our publication, further progress on provable robustness and adversarial training led to models that are also robust across multiple norms (Tramèr and Boneh, 2019; Croce and Hein, 2021). Now, developments in provable robustness can even provide provable bounds on MNIST on the robustness that are close to the empirical values we show (Li *et al.*, 2019a). Nevertheless, our model remains a competitive L2 baseline and with other desired properties such as interpretable decision boundaries and reasonable distal adversarials. Furthermore, to the best knowledge of the author, still no model exists that allows for human like robustness on MNIST. For instance, the MNIST-C dataset, which introduces common corruptions on MNIST, still poses a challenge, especially for adversarially trained networks (Mu and Gilmer, 2019; Rusak *et al.*, 2020).

A property that is often overlooked in our ABS model is the importance of the Gaussian likelihood term that we use in the ELBO. Preliminary (unpublished) experiments of replacing this term with a Bernoulli distribution revealed a strong decreased in the robustness. Technically, a Bernoulli-based model can become too certain on background

pixels in MNIST that are mostly 0. Thus, the individual models have exorbitant confidences on the background pixels. Therefore, changing a background pixel can result in a large change in likelihood and flipping the class label. In contrast, as we use a Gaussian likelihood with fixed width, we avoid such overconfident dependencies on single pixels.

In our work, we focus on a model with an information bottleneck by using an eight-dimensional latent space. This is to guarantee that our latent space is trained properly. For a high dimensional latent space, it can happen that not all latents have been trained to map to an image due to the curse of dimensionality and a limited number of training iterations. In an ablation study, Chen *et al.* (2020b) also find lower robustness results for an analysis-by-synthesis approach with a 32-dimensional latent space.

A downside of our conceptual Gaussian and VAE based ABS implementation is that our model does not scale well out of the box. However, an extension by Ju and Wagner (2020) introduces several improvements such as a discriminative loss to scale the inference towards real-world datasets such as SVHN (Netzer *et al.*, 2011) and a traffic sign dataset (Houben *et al.*, 2013) while remaining high robustness. To further scale ABS to the larger ImageNet datasets, we could process the large images in a patch-based fashion with a sliding window approach similar to BagNets (Brendel and Bethge, 2019), who showed that local patches are sufficient to achieve performances better than AlexNet. BagNets could also be used to create a training dataset for the individual patches to train our ABS model.

Our work also greatly substantiates the difficulty of evaluating the empirical robustness of models. We show that it is necessary to consider various attacks and thread models, as the scores do not only vary across different L_p metrics for a model but also across multiple attacks, which influence future evaluations (Lim *et al.*, 2020). We further show that the commonly presented accuracy at a certain max L_p norm threshold ε can be misleading, as could lead to vastly different results. For instance, our binary CNN is worse on L_∞ at $\varepsilon = 0.3$ compared to Madry *et al.* (2018), but performs favorably for $\varepsilon < 0.1$ or $\varepsilon > 0.4$, as shown in Fig. 2.14. Thus, we recommend plotting the model accuracy over the distortion and reporting the median perturbation size required to fool a model.

In a nutshell, we introduce an implementation of ABS with Gaussian likelihood term and show that it is a good inductive bias for L_p norm robustness on MNIST. Our work has been substantial proof of concept to revive promising investigation on leveraging ABS. Lastly, our thorough evaluation procedure has inspired future work.

Chapter 3

Transfer and combination of our inductive biases

In the previous chapter, we considered individual inductive biases in the context of a *specific* out-of-distribution scenario. In this chapter, we discuss the presented inductive biases across *multiple* out-of-distribution scenarios. We use this as a proxy to evaluate whether our proposed inductive biases help us to learn human-like generalization capabilities. Thus, the first question we tackle is: *Do our investigated inductive biases learn the intended solution?* Next, we investigate whether our proposed inductive biases can be combined beneficially or whether they are mutually exclusive.

Here, the discussion is mostly based on the experiments presented in the appendices of our publications and related works. If no suiting results are available, we performed novel experiments.

3.1 Do our investigated inductive biases learn the intended solution?

Given a typical deep learning setup, we intend to learn a solution that not only solves the training set, but also generalizes beyond it. In practice, there are almost countless different out-of-distribution types. For instance, in autonomous driving just for snow alone, there are numerous variations depending on the temperature and humidity, not even mentioning the fact that ordinary snowflakes have hundreds of branches of ribs, which could vary across snowflakes in a combinatorial fashion (Palmer, 2011). Hence, alone for snow, it is impossible to consider all variations during model development. How can we tackle the vast variety of different types of perturbations, artifacts, and other variations present in our world?

One hopeful observation is that humans, in contrast to machines, seem fairly robust. For instance, a person from the Black Forest who is on his first vacation close to the equator on a Caribbean island will most likely be able to drive a car with some sand on his

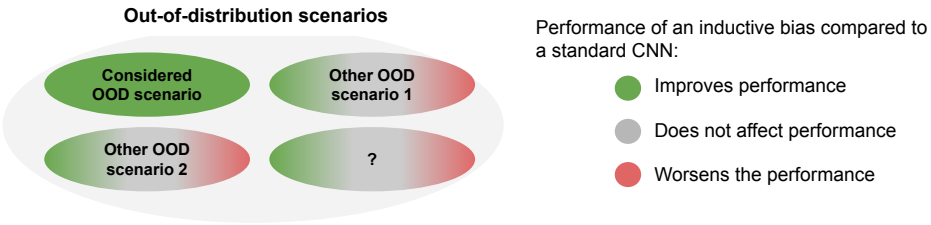


Figure 3.1: **Transfer of inductive biases.** Instead of considering an inductive bias on a specific out-of-distribution scenario, we here consider the effect on multiple scenarios. Here, we depict possible outcomes and distinguish three cases. First, the inductive bias is **transferable** if it increases the performance on all scenarios (all green). Second, it is **specific** if it improves the performance on the scenario it was designed for but has no effect on others (one green, others gray). Lastly, we refer to an inductive bias as **overfitting** if it performs well on a single scenario but worsens the performance on others (one green, others red). In this section, we use these scenarios as a proxy to reason about novel scenarios (depicted by '?').

windshield and palm tree background, despite never having seen this before in real life. Analogously, Neil Armstrong had no apparent visual problems during his first steps on the moon (NASA, 1969). In contrast, such a scenario could completely derail a naively trained machine learning algorithm.

Given the almost infinitive number of out-of-distribution scenarios and the technical limitation of only testing on a finite set, it remains unclear whether this will ever lead towards human-like generalization capabilities.

To investigate whether we are making progress, we consider our proposed inductive biases and test them across multiple other out-of-distribution scenarios. A desired inductive bias should not overfit to a single out-of-distribution scenario, but help to generalize across many scenarios. We suggest the procedure of evaluating a proposed inductive bias across many out-of-distribution scenarios as a proxy for human-like generalization capabilities by selecting scenarios that pose no larger difficulties to humans. This principle is visualized in Fig. 3.1. Here, the intended solution corresponds to inductive biases that would transfer to all out-of-distribution scenarios. In contrast, an overfitting inductive bias learns an unintended solution that reduces the performance on certain out-of-distribution scenarios.

This is also closely related to the no-free lunch theorem (Wolpert and Macready, 1997). In our context, it implies that there is always a fundamental trade-off: an increase in performance on one out-of-distribution scenario will lead to a decrease in another one. However, as human generalization capabilities outmatch machine learning algorithms, we pose human-like generalization as an achievable goal with acceptable trade-offs. Such trade-offs for humans have been shown by Dubey *et al.* (2018), who demonstrate that

the speed of solving games is heavily affected if textures are modified counterintuitively, e.g., switching the textures of a climbable ladder with background textures. However, their proposed modifications are mostly not applicable in the real world. Besides the human generalization capabilities, machine learning algorithms that are more aligned with our understanding should be more intuitive to understand and develop.

An example of a limited inductive bias that can worsen the generalization across multiple out-of-distribution scenarios is L_∞ adversarial training on MNIST (Madry *et al.*, 2018; Schott *et al.*, 2019). As claimed, this method does indeed improve the robustness with respect to L_∞ perturbations. However, this inductive bias has unintended side effects, as it leads to a lower L_0 robustness compared to a vanilla network. To further investigate the shortcomings of L_∞ adversarial training, we show that a good defense strategy for L_∞ robustness is to binarize the input with a thresholding operation and to focus on bright pixels that are not affected by thresholding, as MNIST is almost binary. Now, in terms of L_∞, we need to change the relevant pixels quite a bit in to be flipped past the input binarization. In an updated version of their publication, Madry *et al.* (2018) also analyzed their network weights and found thresholding filters in the first layer. Thus, plain L_∞ adversarial training on MNIST can be more vulnerable to L_0 attacks compared to a cross-entropy trained neural network. In Fig. 3.1, this would correspond to an *overfitting* inductive bias. Note that this overfitting only applies to MNIST and the effect of adversarial training on other datasets and the magnitude of the perturbations has further been studied by Kireev *et al.* (2021), who evaluate of L_∞ adversarial training on common corruptions in natural images.

To evaluate the success of our proposed inductive biases of disentanglement, adversarial noise training, and analysis by synthesis with a Gaussian likelihood, we consider the out-of-distribution scenarios of common corruptions, adversarial examples and generalization along factors of variation present in the data.

3.1.1 Analysis by synthesis with a Gaussian likelihood

In Section 2.4 and Schott *et al.* (2019), we propose an implementation for analysis by synthesis and show that it is fairly robust to various L_p and distal adversarial examples on MNIST. Note that during model development, we chose specifically to tackle adversarial robustness. Now, we study a threat scenario not considered during the development and test the performance of ABS in terms of common corruptions and our *in-domain generalization benchmark*.

On the common corruptions benchmark, MNIST-C, Mu and Gilmer (2019) show that our ABS model performs worse compared to a vanilla CNN. However, when repeating their experiment with the CNN architecture considered by Madry *et al.* (2018) but without adversarial training, we found the opposite to be true. In our experiments, our ABS model performs $\sim$10% better compared to a standard CNN. Similarly, adversarial

training seems to be better than a normally trained architectural twin. This might be due to the fact that Mu and Gilmer (2019) used dropout, different network architectures, and a slightly different optimization procedure during training. We presume that the MNIST-C network performances are dependent on small variations in the network architecture, and a more thorough comparison is required. Given the current observations, we conclude that neither ABS nor adversarial training are transferable inductive biases for out-of-distribution generalization. Both, on average, perform similar to a vanilla neural network. However, an important benefit of our ABS model is that the predictions on MNIST-C samples have low confidence ($p(c|\mathbf{x}) \leq 0.2$ for almost all $\mathbf{x}$). For our trained CNN model and others, we observe overconfident wrong predictions.[1] To explain why ABS is adversarially robust but roughly on par with a vanilla network on common corruptions, we hypothesize that the common corruption samples are too far away from the training data distribution. Thus, our ABS model performs poorly, as it is only trained on clean samples and the Gaussian likelihood might only be helpful for reliably generalizing to samples near the learned distributions. The suggested far distance between corrupted images to the data manifold defined by the VAEs would also be in accordance with the low confidence predictions of our ABS model.

We further measured the performance of our VAE-based ABS model on our proposed in-domain generalization benchmark from section 2.2. This is motivated by the suggested benefits of causal models in terms of generalization (Schölkopf, 2019). To compare the causal ABS inference with an anti-causal feedforward method[2], we used our SlowVAE model. For the feedforward part, we used the encoder. For ABS, we solely relied on the decoder and used gradient descent in the latent space for inference. Preliminary results[3] on disentanglement scores on our systematic test splits show no improvements when using ABS-based inference instead of the encoder. A related observation is made by Montero *et al.* (2021) who trained a generative model on dSprites that mapped factors of variations to images. Similar to our findings on feedforward networks in Section 2.2, they show that the tested generative models are unable to properly reconstruct novel configurations of factors. Even though our focus here is on ABS-based representation learning and theirs lays on image generation, both results point towards similar limitations of generative models.

In conclusion, ABS with a Gaussian likelihood is an effective inductive bias on MNIST covering various L_p norms and often aligning with humans on controversial stimuli (Golan *et al.*, 2020). However, it is *specific* (see Fig. 3.1) to adversarial scenarios, as it does not affect the classification of strongly perturbed digits, nor the generalization to novel combinations of factors. In contrast to Madry *et al.* (2018)'s model, we observe no large drops on L_0 norm (Schott *et al.*, 2019).

[1] results not published.

[2] It is currently an ongoing debate whether the MNIST digit classification task is causal or anti-causal (Arjovsky *et al.*, 2020).

[3] results not published.

3.1.2 Disentanglement in visual representation learning

Disentanglement is a promising direction for unsupervised representation learning and generalization. It has been shown to improve downstream generalization (Locatello *et al.*, 2020b; Peters *et al.*, 2017). Yet, as discussed in the literature (Träuble *et al.*, 2021; Montero *et al.*, 2021) and by us (Schott *et al.*, 2021), we found no increase in robustness along factors of variation present in the data. Analogously to the previous section, we now discuss the effect of disentanglement on adversarial and common corruption robustness.

The connection between adversarial examples in disentanglement has explicitly been tested for the β-VAE and β-TCVAE. Here, Willetts *et al.* (2019) show in their experiments that there is no obvious connection between the degree of disentanglement of a learned representation and the adversarial robustness of a model. However, they find that the objectives used in disentanglement, such as a high regularization in the latent space of a VAE, are connected to robustness. They hypothesize that higher regularization in a VAE leads to a higher overlap in the encoder posterior, leading to a less *lookup-table* like representation and a more structural model.

We do not have any direct results for disentanglement methods and their performance on common corruptions. However, Willetts *et al.* (2019) also investigate Gaussian noise robustness and find that for β-VAEs with a $\beta > 1$ on the chairs dataset can lead to slightly increased robustness. More generally, weakly-supervised pertaining such as contrastive learning[4] enables us to leverage large amounts of unlabeled datasets. The combination of those two inductive biases (large amounts of data and self supervised representation learning), can greatly improve the performance in various settings (Kotar *et al.*, 2021; Geirhos *et al.*, 2021).

We conclude that disentanglement does not automatically lead to robustness or generalization. However, the objectives for disentanglement are closely related to methods that allow to use large amounts of data and increase model generalization capabilities. Thus, disentanglement and generalization seem to be mutually achievable.

3.1.3 Adversarial noise training

We tried to design an adversarial noise training that increases the robustness on common corruptions. To investigate the effect on other out-of-distribution scenarios, we evaluate the adversarial robustness of our adversarial noise training (Rusak *et al.*, 2020).

Based on the evaluation of our proposed ANT model on the L_2 and L_∞ norms, we find that our method slightly increases the robustness compared to a vanilla model. However, the

[4] a connection between contrastive learning methods and disentanglement has been shown by Zimmermann *et al.* (2021)

robustness improvements are only marginal compared to specifically designed methods for adversarial robustness on ImageNet, such as adversarial training (Xie *et al.*, 2019). Surprisingly, the other direction, testing adversarial robust networks on common corruptions can lead to large drops in performance (Rusak *et al.*, 2020; Gilmer *et al.*, 2019). Especially on the fog category, we observed a $\geq 10x$ drop for three tested networks designed for adversarial robustness. Recently, Kireev *et al.* (2021) further investigated this connection and found that relaxing the considered distance metrics used in adversarial training can revert this effect and actually improve the performance of adversarial training on ImageNet-C. However, a significant drop on the fog category remains.

We did not empirically investigate the performance of adversarial training on our in-domain generalization benchmark and leave this to future work.

3.1.4 Summary

We observed that our investigated inductive biases ANT, ABS, and disentanglement are mostly *specific* and do not have high effects on other considered out-of-distribution scenarios (see Fig. 3.1). This contrasts with inductive biases such as L_∞ adversarial training on MNIST that can lead to a decrease in performance on other norms. However, the fact that our inductive biases seem to be specific and have no side effects is not as surprising as it might seem. (Geirhos *et al.*, 2018) show that a network can have state-of-the-art performance when trained on one noise, but simultaneously have chance level performance on similar noise that look visually almost indistinguishable to humans. Also, D'Amour *et al.* (2020) vary random seeds in neural networks and show that there is almost no correlation of the performances on different out-of-distribution scenarios.

Finally, to answer the question *Do our proposed inductive biases help learning the intended solution?*, we defined the intended solution as a generalization to various out-of-distribution scenarios. As we cannot test all possible out-of-distribution scenarios, we consider only a subset as a proxy. Here, on the one hand, our inductive biases show robustness w.r.t. various out-of-distribution scenarios, e.g., ANT improves the accuracy on all ImageNet-C corruptions and ABS is robust w.r.t. multiple L_p norms. On the other hand, when cross-evaluating our methods on ImageNet-C, adversarial examples, or our in-domain generalization benchmark, we found no effect. In terms of the no free lunch theorem, we observed no drops while cross validating on other out-of distribution sets. Thus, we do find inductive biases that help us move forward, but a gap towards the intended solution with human-like robustness on multiple domains still remains.

We acknowledge that the statement of moving forward towards human-like robustness is limited, as we only consider a few out-of-distribution scenarios out of almost infinitely many. This general principle has been paraphrased by Hubert L Dreyfus who stated, "It was like claiming that the first monkey that climbed a tree was making progress towards landing on the moon" (Mitchell, 2021). However, small generalization improvements

could already have an impact on current applications. Moreover, our inductive biases show robustness to broad threat scenarios. ABS allows for out-of-the-box robustness towards multiple lp norms, and ANT increases the robustness towards multiple perturbations.

3.2 On the combination of inductive biases

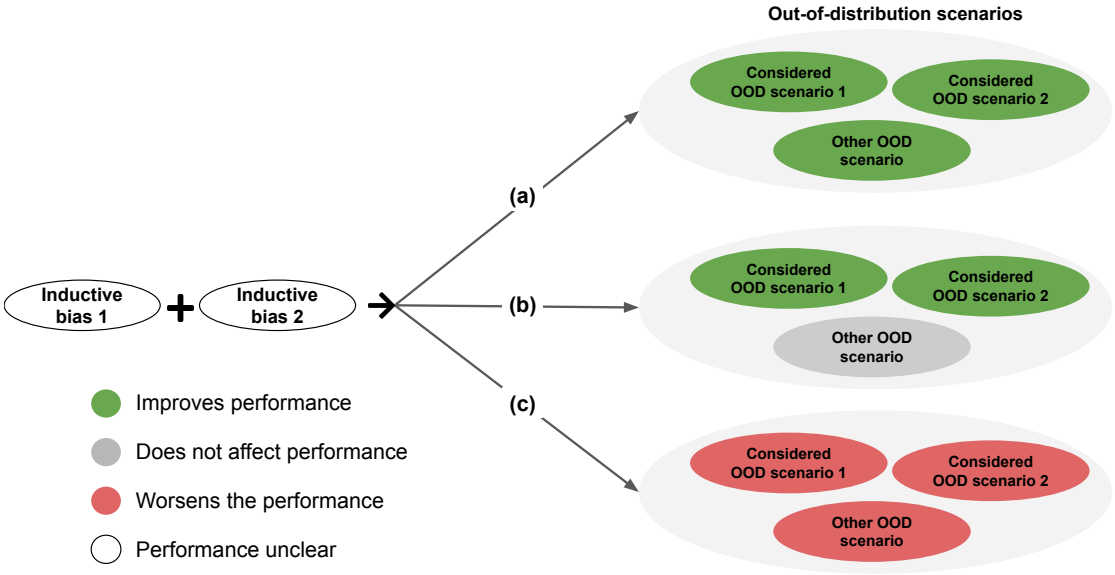


Figure 3.2: **Combination of inductive biases.** We propose three different options for combining inductive biases. **(a)** The combination is **symbiotic** and therefore bigger than its parts. **(b)** The inductive biases are **orthogonal**, and combining them results in a solution that performs well on the individually considered out-of-distribution scenarios. **c)** The inductive biases are **exclusive**, and combining them worsens the performance - even on the out-of-distribution scenarios they were designed for.

In the previous section, we investigated individual inductive biases across one or multiple out-of-distribution scenarios. They can be either transferable across multiple domains, overfit and decrease the performances on other domains, or be specific and have no effect on other domains. We observed that our proposed inductive biases are specific to their out-of-distribution scenario and have mostly no effect on others. Naturally, the question arises whether different inductive biases can be combined to achieve further out-of-distribution generalization in multiple scenarios. Such a combination could either be beneficial or even harmful. We visualize the possible outcomes in Fig. 3.2. Again, we mostly focus on the inductive biases considered in this thesis and the related work. More concretely, we repeat the combination of ABS and a Gaussian likelihood and we discuss

the combination of contrastive learning, our sparse transition prior, and ABS. Lastly, we discuss ANT and data augmentation methods in the context of ABS and disentanglement.

A prototypical example of a symbiotic combination of inductive biases is ABS with a Gaussian likelihood (Schott *et al.*, 2019). Plain ABS as proposed concurrently by Kilbertus *et al.* (2018) could, for instance, rely on a computer graphics model to render objects for recognition. Here, an object needs to be rendered and matched perfectly. Otherwise, no classification is possible, and the sample is rejected. In practice, it is not feasible to assume a rendering machine that has the capacity to render all variations in the world. Furthermore, simply rejecting unknown inputs is insufficient in common applications like autonomous vehicle driving. To generalize beyond the manifold defined by the generative model, e.g., the rendering engine, we leverage the Gaussian likelihood commonly used in VAEs to extend the plain ABS. On MNIST, we show empirically and theoretically that this combination can lead to an adversarially robust classification and properly calibrated predictions in the whole input space. Therefore, we generalized the plain ABS principle to the whole input space.

Analogous to the considered example of ABS with a Gaussian likelihood, we next hypothesize about possible combinations of our considered inductive biases to further improve the generalization capabilities.

3.2.1 Disentanglement, ABS, and more data

In the literature, a notable combination uses contrastive learning as weakly supervised pertaining to leverage large amounts of data (Chen *et al.*, 2020a). Simply using more data has been shown again and again to improve results and beat sophisticated baselines (Sutton, 2019). Intuitively, more data can be used to train large models and avoid overfitting. Methods relying on large datasets are currently state-of-the-art in various robustness benchmarks, and is a reliable go-to option for applications to take a pre-trained network from a large dataset and fine-tune it on a specific task (Kolesnikov *et al.*, 2020). To extend this, we hypothesize that our method to leverage the sparse transition priors observed in YouTube-VOS and KITTI-Masks could be used as a further type of signal to not only leverage image data but also video data similarly to Qian *et al.* (2021). Thus, allowing to use more data could further improve generalization performances.

In principle, it is also possible to further combine representation learning methods with ABS for more trustworthy models. Under certain assumptions, the sparse transition prior enables us to identify the true latent space up to some ambiguities. Thus, after successful training of a feedforward model (e.g., PCL, or our SlowVAE), we should be able to identify all latents corresponding to each image or video frame. Subsequently, we could use these image-latent pairs to train a generative model conditioned on various latents (for SlowVAE, we get this generative model for free as we train a decoder jointly with the encoder). After training the generative model, we could use it to verify whether

the latent representation has been inferred robustly by synthesizing the corresponding latents and checking whether they match with the input image, similarly to Ghosh *et al.* (2019) and Lamb *et al.* (2018). Given a mismatch, to restore the correct class, we could perform gradient descent inference in the latent space of the generative model (Schott *et al.*, 2019). In this way, we would bypass the vulnerable encoder. Lastly, given this trustworthy latent space of model latents often only requires a linear readout to apply to a specific task. Such a latent space could for instance be used in combination with invariant risk minimization (IRM) that has shown to be produce optimal results for a linear classifier in multiple environments (Arjovsky *et al.*, 2020).

All in all, the proposed combination of identifiable representation learning techniques with ABS and an IRM based readout principle could be promising in terms of scalable and trustworthy machine learning. However, this also hinges on other problems, e.g., achieving a scalable and robust perceptual loss for the ABS on natural images.

3.2.2 Data augmentation

In Section 2.3, we show that our proposed ANT could be combined with a model trained on stylized images, which encourages models to focus on shape rather than texture, to further improve the common corruption robustness. This has been combined by Kireev *et al.* (2021) with an adapted version of adversarial training that improved the results even further.

Not only for feedforward methods, we see many benefits of data augmentations for common corruptions, but also for generative classifiers. For a scaled implementation of ABS implemented by a score SDE based generative model, we observe improvements on CIFAR-10-C[5]. Those improvements could be further combined by leveraging data augmentation methods such as AugMix (Hendrycks *et al.*, 2020a).

3.2.3 Summary

All in all, we observed that the inductive biases presented in this thesis are orthogonal to each other and could be combined in a modular fashion. This could help to further close the gap to human-like robustness. In Reinforcement learning, it has been shown that combining inductive biases can lead to large improvements (Hessel *et al.*, 2018). We thus propose a large-scale study implementing and combining strong inductive biases presented in the literature.

[5] ongoing work.

Chapter 4

Outlook

Current deep learning methods have been shown to achieve super-human performance in cases where the training distribution is identical to the test distribution (He *et al.*, 2015). In the next era of deep learning, we hope to see similar achievements on scenarios in which the test distribution differs slightly from the training distribution.

We have a clear set of milestones ahead and various scenarios to quantify our progress. To measure our progress towards on out-of-distribution settings, various benchmarks have been proposed. Some examples are common corruptions and perturbations (Hendrycks and Dietterich, 2019), images without texture cues (Geirhos *et al.*, 2019), our in-domain benchmark (Schott *et al.*, 2021) and many more (Funke *et al.*, 2018; Montero *et al.*, 2021). In an extreme case, one can also use adversarial examples to find the worst case distributional shift. Similarly, we could also tackle more complex goals that automatically require generalization properties, such as simultaneous localization and mapping (SLAM) in the wild.

So far, there exists no general solution to the proposed out-of-distribution benchmarks or "in the wild" applications but we are making progress. Despite an almost exponentially growing number of research publications focussing on this topic and increasing of research funds, novel scenarios still require human engineering work and additional task-specific assumptions. There is no simple plug-and-play solution that works reliably on all proposed benchmarks. While we have not achieved general out-of-distribution robustness, the progress itself has not stagnated. For instance, the ImageNet-C error has deceased from 76.7% mCE (Hendrycks and Dietterich, 2019) of a standard ResNet to 53.6% mCE (Hendrycks *et al.*, 2020b) since the release of the benchmark 2019. Even further, current methods with additional assumptions that allow access to more data and the test images (but not the test labels!), even decreased the error down to 22.0% (Rusak *et al.*, 2021).

In terms of Sutton's bitter lesson, in which simply more data and compute power improves the results (Sutton, 2019), this trend is likely to continue. For instance, despite already having large image copra to train our models, we still observe further advancements simply through larger datasets and bigger models (Sun *et al.*, 2017). Given ter-

abytes of daily data uploads and semi-automatic labeling tools, this trend will most likely continue. Moreover, novel generations of processing units like TPUs, GPUs, and increased dataset center sizes will most likely further advance this trend. At the end, breakthroughs in quantum computing might even increase the pace of current progress that is often already displayed on logarithmic scales.

Similarly to more data and compute, incorporating structure into learning models still regularly increases the performance of learning algorithms in various benchmarks. E.g., architectural improvements from fully-connected neural networks over convolutional networks, and lately visual transformers successively resulted in novel state-of-the-art performances on the ImageNet and other benchmarks (PapersWithCode, 2021). Given the vast structure of our world and our physical understanding of it and that has not yet been incorporated into models, we suspect that further inductive biases will follow.

Overall, the current best performing architectures are often a combination of novel inductive biases, large amounts of data, and models with many parameters. E.g., the current best performing ImageNet model is trained on the largest dataset but also relies on visual transformers and other inductive biases (Zhai *et al.*, 2021; Mao *et al.*, 2021).

Nevertheless, we find the improvements through incorporated structure more revealing. Simply adding more data and larger models does often increase the performance, but does not necessarily help us with our understanding of algorithms and the world. In contrast, novel inductive biases underlined by their success can point out important features of scene understanding. For instance, seminal work of Vaswani *et al.* (2017) highlights the concept of attention in language processing tasks. Building on this, Carion *et al.* (2020) show that visual transformers can combine distant features like an elephant's trunk and tail by treating image representations as a set and subsequently removing the *locality* bias of convolutions and revealing the importance of wide context. Subsequently, such models could also be used to provide the highest shape bias, narrowing the gap towards humans. Similarly, Golan *et al.* (2020) showed that our proposed analysis by synthesis model has the highest alignment with humans on controversial stimuli, thus highlighting the importance of generative classifiers.

Incorporating structure also has other benefits and has probably also not reached its full potential. Implementing the right structure can enable machines to learn tasks much faster, with fewer data samples and fewer parameters. Naively estimating the number of different images a child sees in its first three years[1] results in roughly 50 million images. This is less than current datasets, which have up to 300 million images and more than a billion labels (Sun *et al.*, 2017). Nonetheless, there is still a gap between human and machine vision capabilities (Kühl *et al.*, 2020), that future inductive biases could narrow without requiring more data.

[1] Assuming 3 years of vision and 1 new image per second and a 12hour wake period. Note that we are ignoring effects of evolution here, as they are hard-wired into the brain and could therefore also (debatably) be considered as prior structure.

At the core of incorporating structure and finding novel inductive biases, we expect to build vision models to better leverage video or interactive datasets. In the example from the previous paragraph, we naively compared common image datasets with the naturally available data during childhood, but ignored the temporal and its interactive nature. An example of a first "small step" towards leveraging such structure is using sparse temporal transitions of pairs of images for provable disentanglement (Klindt *et al.*, 2021). Other similar methods rely on predicting future frames of videos or other auxiliary tasks (Jaderberg *et al.*, 2016). Analogously, in simple environments, models have been trained to learn a 3D aware structure of the environments by predicting novel viewpoints of a scene (Kosiorek *et al.*, 2021). Moreover, directly controlling for 3D properties of scenes allows manipulating scenes in a controllable manner (Niemeyer and Geiger, 2021) and provides much more control compared to standard generative adversarial networks with less incorporated physical prior knowledge.

Furthermore, the amount of used data could be further reduced by building curiosity-driven systems with a sense of boredom (Schmidhuber, 1991). For instance, interactive data queries could be used to actively query points that have the highest learning potential for the algorithm. Such a system could make predictions about the back of objects by using certain symmetry assumptions and later using interactive properties of the environment to verify this prediction.

Lastly, more adaptive processing schemes might be crucial. Current neural networks in image processing are fixed during deployment or testing. Moreover, the computational budget for processing each image is usually the same and independent of its complexity. Novel approaches that can adaptively adjust the amount of required compute could be more powerful. For instance, models that are based on differential equation solvers have been shown to improve the likelihood of images over methods with a fixed budged (Song *et al.*, 2020). Analogously, simply adapting the batch-norm parameters on the test set (without accessing the test labels) can improve the performance on ImageNet-C (Schneider *et al.*, 2020). Thus, more adaptive architectures could further drive the next wave of machine learning research.

In a nutshell, we hope to extend towards inductive biases that incorporate structures of our visual world and are adaptive in terms of leveraging data and compute. By quantifying the usefulness, we hopefully also gain a better understanding for the importance of certain ingredients required for a human-like scene understanding.

Acknowledgments

Foremost, I would like to thank Wieland Brendel for constantly guiding me during the past four years. I will miss his honest and sharp-witted feedback that helped me stay on track. Our meetings have not only been scientifically enlightening, but also personally joyful.

I would like to thank Matthias Bethge for creating and pushing our surrounding scientific ecosystem, in which we as students could thrive. This fast-moving and ever-growing wave provided constant opportunities – it was simply propelling to be apart.

A shoutout also goes to my TAC committee Matthias Bethge, Wieland Brendel, Matthias Hein and Georg Martius. I appreciate their guidance, assurance, and feedback on my path towards my PhD.

A big thanks also goes to Heike König and Melanie Palm for their everlasting help, support and patience. I would not have managed my way through the bureaucratic jungle without their support.

I would also like to thank Fred Hamprecht, Ullrich Köthe, and Steffen Wolf for a marvelous time during my master thesis that motivated me to stay in research and pursue a PhD.

I would like to thank David Klindt, Claudio Michaelis for their support in struggling times. I would further like to thank Judy Borowski, Roland Geirhos, Matthias Kümmerer, Evgenia Rusak, Roland Zimmermann, Dominik Zietlow, Dylan Paiton, and all other colleagues from the Bethgelab for making my time there unforgettable and creating many positive memories.

Special thanks go to Ilona Stolpner for her everlasting love and ability to support me even through the most difficult times. Similarly, my friends also helped to stay grounded and not forget about other important aspects of life. Special thanks go to Johann Theisen, Paul Spallinger, Daniel Wuttcke, Mathis Brosowsky, Bastian Bühler, Julian Wahlbrecht, Alexander Koch, Alexander Dombrowski, Saya Rapp and many more.

Last but not least, I would like to thank my family for their love, emotional support, and always putting things into perspective. Thank you to Max & Johanna Schott, Felix & Johanna Wodtke-Schott, Arne & Heidi & Paul & Harriet & Henrik Sand.

Bibliography

Abnar, S., Dehghani, M., and Zuidema, W. (2020). Transferring inductive biases through knowledge distillation. *arXiv preprint arXiv:2006.00555*.

Achlioptas, P., Diamanti, O., Mitliagkas, I., and Guibas, L. J. (2018). Learning representations and generative models for 3d point clouds. In J. G. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 40–49. PMLR.

Adel, T., Ghahramani, Z., and Weller, A. (2018). Discovering interpretable representations for both deep generative and discriminative models. In J. G. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 50–59. PMLR.

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D. (2016). Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*.

Arganda-Carreras, I., Turaga, S. C., Berger, D. R., Cireşan, D., Giusti, A., Gambardella, L. M., Schmidhuber, J., Laptev, D., Dwivedi, S., Buhmann, J. M., *et al.* (2015). Crowdsourcing the creation of image segmentation algorithms for connectomics. *Frontiers in neuroanatomy*, **9**, 142.

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. (2020). Invariant risk minimization.

Athalye, A., Carlini, N., and Wagner, D. A. (2018). Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In J. G. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 274–283. PMLR.

Barker, S. F. and Achinstein, P. (1955). On the new riddle of induction. *The Philosophical Review*, **69**(4), 511–522.

Barras, C. (2021). How did neanderthals and other ancient humans learn to count? *Nature*, **594**(7861), 22–25.

Bartlett, P. L., Long, P. M., Lugosi, G., and Tsigler, A. (2020). Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, **117**(48), 30063–30070.

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., *et al.* (2018). Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*.

Beery, S., Van Horn, G., and Perona, P. (2018). Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pages 456–473.

Bell, A. J. and Sejnowski, T. J. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, **7**(6), 1129–1159.

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, **35**(8), 1798–1828.

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., and Roli, F. (2013).

Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402. Springer.

Bishop, C. M. (1995). Training with noise is equivalent to tikhonov regularization. *Neural computation*, **7**(1), 108–116.

Bouchacourt, D., Ibrahim, M., and Deny, S. (2021). Addressing the topological defects of disentanglement.

Bowern, C. and Zentz, J. (2012). Diversity in the numeral systems of australian languages. *Anthropological Linguistics*, **54**(2), 133–160.

Brendel, W. and Bethge, M. (2019). Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Brendel, W., Rauber, J., and Bethge, M. (2018). Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Brown, T. B., Mané, D., Roy, A., Abadi, M., and Gilmer, J. (2017). Adversarial patch. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Bruna, J., Mallat, S., Bacry, E., and Muzy, J.-F. (2015). Intermittent process analysis with scattering moments. *The Annals of Statistics*, **43**(1), 323–351.

Calian, D. A., Stimberg, F., Wiles, O., Rebuffi, S.-A., Gyorgy, A., Mann, T., and Gowal, S. (2021). Defending against image corruptions through adversarial augmentations. *ArXiv preprint*, **abs/2104.01086**.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer.

Carlini, N., Athalye, A., Papernot, N., Brendel, W., Rauber, J., Tsipras, D., Goodfellow, I., Madry, A., and Kurakin, A. (2019). On evaluating adversarial robustness. *ArXiv preprint*, **abs/1902.06705**.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. E. (2020a). A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Chen, T. Q., Li, X., Grosse, R. B., and Duvenaud, D. (2018). Isolating sources of disentanglement in variational autoencoders. In S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 2615–2625.

Chen, Y., Xie, R., and Zhu, Z. (2020b). On breaking deep generative model-based defenses and beyond. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1736–1745. PMLR.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Chomsky, N. (2014). *Aspects of the Theory of Syntax*, volume 11. MIT press.

Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B., and LeCun, Y. (2015). The loss surfaces of multilayer networks. In G. Lebanon and S. V. N. Vishwanathan, editors, *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2015, San Diego, California, USA, May 9-12, 2015*, volume 38 of *JMLR Workshop and Conference Proceedings*. JMLR.org.

Ciresan, D. C., Meier, U., Masci, J., Gambardella, L. M., and Schmidhuber, J. (2011). Flexible, high performance convolutional neural networks for image classification. In T. Walsh, editor, *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 1237–1242. IJCAI/AAAI.

Cohen, J. M., Rosenfeld, E., and Kolter, J. Z. (2019). Certified adversarial robustness via randomized smoothing. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 1310–1320. PMLR.

Cohen, T. and Welling, M. (2016). Group equivariant convolutional networks. In M. Balcan and K. Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2990–2999. JMLR.org.

Comon, P. (1994). Independent component analysis, a new concept? *Signal processing*, **36**(3), 287–314.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 3213–3223. IEEE Computer Society.

Craven, M. W. (1996). *Extracting comprehensible models from trained neural networks*. Ph.D. thesis, The University of Wisconsin-Madison.

Croce, F. and Hein, M. (2020). Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 2206–2216. PMLR.

Croce, F. and Hein, M. (2021). Adversarial robustness against multiple *l_p*-threat models at the price of one and how to quickly fine-tune robust models to another threat model. *ArXiv preprint*, **abs/2105.12508**.

Croce, F., Andriushchenko, M., Singh, N. D., Flammarion, N., and Hein, M. (2020). Sparse-rs: a versatile framework for query-efficient sparse black-box adversarial attacks. *ArXiv preprint*, **abs/2006.12834**.

Csordás, R., van Steenkiste, S., and Schmidhuber, J. (2021). Are neural nets modular? inspecting functional modularity through differentiable weight masks. In *International Conference on Learning Representations*.

D'Amour, A., Heller, K., Moldovan, D., Adlam, B., Alipanahi, B., Beutel, A., Chen, C., Deaton, J., Eisenstein, J., Hoffman, M. D., *et al.* (2020). Underspecification presents challenges for credibility in modern machine learning. *arXiv preprint arXiv:2011.03395*.

Davies, A. (2017). Self-driving cars flock to arizona, land of good weather and no rules.

de Cordemoy, G. (1973). *Discours physique de la parole (1668)*, volume 90. Slatkine.

Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Li, F. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255. IEEE Computer Society.

Dittadi, A., Träuble, F., Locatello, F., Wuthrich, M., Agrawal, V., Winther, O., Bauer, S., and Schölkopf, B. (2021). On the transfer of disentangled representations in realistic settings. In *International Conference on Learning Representations*.

Draxler, F., Veschgini, K., Salmhofer, M., and Hamprecht, F. A. (2018). Essentially no barriers in neural network energy landscape. In J. G. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1308–1317. PMLR.

Dubey, R., Agrawal, P., Pathak, D., Griffiths, T., and Efros, A. A. (2018). Investigating human priors for playing video games. In J. G. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1348–1356. PMLR.

Eastwood, C. and Williams, C. K. I. (2018). A framework for the quantitative evaluation of disentangled representations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Elsayed, G. F., Shankar, S., Cheung, B., Papernot, N., Kurakin, A., Goodfellow, I. J., and Sohl-Dickstein, J. (2018). Adversarial examples that fool both computer vision and time-limited humans. In S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 3914–3924.

Everett, D. (2005). Cultural constraints on grammar and cognition in pirahã: Another look at the design features of human language. *Current anthropology*, **46**(4), 621–646.

Fasel, B. and Gatica-Perez, D. (2006). Rotation-invariant neoperceptron. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 3, pages 336–339. IEEE.

Ferrucci, D., Levas, A., Bagchi, S., Gondek, D., and Mueller, E. T. (2013). Watson: beyond jeopardy! *Artificial Intelligence*, **199**, 93–105.

Fong, R. C., Scheirer, W. J., and Cox, D. D. (2018). Using human brain activity to guide machine learning. *Scientific reports*, **8**(1), 1–10.

Fukushima, K. (1988). Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural networks*, **1**(2), 119–130.

Funke, C., Borowski, J., Wallis, T., Brendel, W., Ecker, A., and Bethge, M. (2018). Comparing the ability of humans and dnns to recognise closed contours in cluttered images. *Journal of Vision*, **18**(10), 800–800.

Funke, C. M., Borowski, J., Stosio, K., Brendel, W., Wallis, T. S., and Bethge, M. (2021). Five points to check when comparing visual perception in humans and machines. *Journal of Vision*, **21**(3), 16–16.

Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? the KITTI vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, pages 3354–3361. IEEE Computer Society.

Geirhos, R., Temme, C. R. M., Rauber, J., Schütt, H. H., Bethge, M., and Wichmann, F. A. (2018). Generalisation in humans and deep neural networks. In S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 7549–7561.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. (2019). Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. (2020). Shortcut learning in deep neural networks. *Nature Machine Intelligence*, **2**(11), 665–673.

Geirhos, R., Narayanappa, K., Mitzkus, B., Thieringer, T., Bethge, M., Wichmann, F. A., and Brendel, W. (2021). Partial success in closing the gap between human and machine vision. *arXiv preprint arXiv:2106.07411*.

Ghosh, P., Losalka, A., and Black, M. J. (2019). Resisting adversarial attacks using gaussian mixture variational autoencoders. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019,*

The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019, pages 541–548. AAAI Press.

Gilmer, J., Ford, N., Carlini, N., and Cubuk, E. D. (2019). Adversarial examples are a natural consequence of test error in noise. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2280–2289. PMLR.

Golan, T., Raju, P. C., and Kriegeskorte, N. (2020). Controversial stimuli: Pitting neural networks against each other as models of human cognition. *Proceedings of the National Academy of Sciences*, **117**(47), 29330–29337.

Gondal, M. W., Wuthrich, M., Miladinovic, D., Locatello, F., Breidt, M., Volchkov, V., Akpo, J., Bachem, O., Schölkopf, B., and Bauer, S. (2019). On the transfer of inductive bias from simulation to the real world: a new disentanglement dataset. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 15714–15725.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C., and Bengio, Y. (2014). Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680.

Greff, K., van Steenkiste, S., and Schmidhuber, J. (2020). On the binding problem in artificial neural networks. *ArXiv preprint*, **abs/2012.05208**.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 1026–1034. IEEE Computer Society.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society.

Hein, M. and Andriushchenko, M. (2017). Formal guarantees on the robustness of a classifier against adversarial manipulation. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 2266–2276.

Hendrycks, D. and Dietterich, T. G. (2019). Benchmarking neural network robustness to common corruptions and perturbations. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Hendrycks, D., Mu, N., Cubuk, E. D., Zoph, B., Gilmer, J., and Lakshminarayanan, B. (2020a). Augmix: A simple method to improve robustness and uncertainty under data shift. In *International Conference on Learning Representations*.

Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., *et al.* (2020b). The many faces of robustness: A critical analysis of out-of-distribution generalization. *arXiv preprint arXiv:2006.16241*.

Hessel, M., Modayil, J., van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M. G., and Silver, D. (2018). Rainbow: Combining improvements in deep reinforcement learning.

In S. A. McIlraith and K. Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 3215–3222. AAAI Press.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2017a). beta-vae: Learning basic visual concepts with a constrained variational framework. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Higgins, I., Pal, A., Rusu, A. A., Matthey, L., Burgess, C., Pritzel, A., Botvinick, M., Blundell, C., and Lerchner, A. (2017b). DARLA: improving zero-shot transfer in reinforcement learning. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1480–1490. PMLR.

Higgins, I., Amos, D., Pfau, D., Racaniere, S., Matthey, L., Rezende, D., and Lerchner, A. (2018). Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, **9**(8), 1735–1780.

Houben, S., Stallkamp, J., Salmen, J., Schlipsing, M., and Igel, C. (2013). Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In *International Joint Conference on Neural Networks*.

Huang, G., Liu, Z., van der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2261–2269. IEEE Computer Society.

Hüllermeier, E., Fober, T., and Mernberger, M. (2013). Inductive bias. In W. Dubitzky, O. Wolkenhauer, K.-H. Cho, and H. Yokota, editors, *Encyclopedia of Systems Biology*, pages 1018–1018. Springer New York, New York, NY.

Hyvärinen, A. and Morioka, H. (2016). Unsupervised feature extraction by time-contrastive learning and nonlinear ICA. In D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3765–3773.

Hyvärinen, A. and Morioka, H. (2017). Nonlinear ICA of temporally dependent stationary sources. In A. Singh and X. J. Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54 of *Proceedings of Machine Learning Research*, pages 460–469. PMLR.

Hyvärinen, A. and Pajunen, P. (1999). Nonlinear independent component analysis: Existence and uniqueness results. *Neural networks*, **12**(3), 429–439.

Hyvärinen, A., Hurri, J., and Väyrynen, J. (2003). Bubbles: a unifying framework for low-level statistical properties of natural image sequences. *JOSA A*, **20**(7), 1237–1252.

Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. (2019). Adversarial examples are not bugs, they are features. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 125–136.

Jacobsen, J., Behrmann, J., Zemel, R. S., and Bethge, M. (2019). Excessive invariance causes adversarial vulnerability. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Jaderberg, M., Simonyan, K., Zisserman, A., and Kavukcuoglu, K. (2015). Spatial transformer networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2017–2025.

Jaderberg, M., Mnih, V., Czarnecki, W. M., Schaul, T., Leibo, J. Z., Silver, D., and Kavukcuoglu, K. (2016). Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*.

Johnson, H. M. (1911). Clever hans (the horse of mr. von osten): A contribution to experimental, animal, and human psychology. *The Journal of Philosophy, Psychology and Scientific Methods*, **8**(24), 663–666.

Ju, A. and Wagner, D. (2020). E-abs: extending the analysis-by-synthesis robust classification model to more complex image domains. In *Proceedings of the 13th ACM Workshop on Artificial Intelligence and Security*, pages 25–36.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., *et al.* (2021). Highly accurate protein structure prediction with alphafold. *Nature*, pages 1–11.

Jutten, C. and Herault, J. (1991). Blind separation of sources, part i: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, **24**(1), 1–10.

Kawaguchi, K. (2016). Deep learning without poor local minima. In D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 586–594.

Khemakhem, I., Monti, R. P., Kingma, D. P., and Hyvärinen, A. (2020a). Ice-beem: Identifiable conditional energy-based deep models based on nonlinear ICA. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Khemakhem, I., Kingma, D. P., Monti, R. P., and Hyvärinen, A. (2020b). Variational autoencoders and nonlinear ICA: A unifying framework. In S. Chiappa and R. Calandra, editors, *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 2207–2217. PMLR.

Kilbertus, N., Parascandolo, G., and Schölkopf, B. (2018). Generalization in anti-causal learning. *ArXiv preprint*, **abs/1812.00524**.

Kim, H. and Mnih, A. (2018). Disentangling by factorising. In J. G. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2654–2663. PMLR.

Kingma, D. P. and Dhariwal, P. (2018). Glow: Generative flow with invertible 1x1 convolutions. In S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 10236–10245.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In Y. Bengio and Y. LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Kireev, K., Andriushchenko, M., and Flammarion, N. (2021). On the effectiveness of adversarial training against common corruptions. *ArXiv preprint*, **abs/2103.02325**.

Klindt, D. A., Schott, L., Sharma, Y., Ustyuzhaninov, I., Brendel, W., Bethge, M., and Paiton, D. (2021). Towards nonlinear disentanglement in natural data with temporal sparse coding. In *International Conference on Learning Representations*.

Kohavi, R., Wolpert, D. H., *et al.* (1996). Bias plus variance decomposition for zero-one loss functions. In *ICML*, volume 96, pages 275–83.

Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., and Houlsby, N. (2020). Big transfer (bit): General visual representation learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 491–507. Springer.

Kosiorek, A. R., Strathmann, H., Zoran, D., Moreno, P., Schneider, R., Mokrá, S., and Rezende, D. J. (2021). Nerf-vae: A geometry aware 3d scene generative model. *ArXiv preprint*, **abs/2104.00587**.

Kotar, K., Ilharco, G., Schmidt, L., Ehsani, K., and Mottaghi, R. (2021). Contrasting contrastive self-supervised representation learning pipelines. In *ICCV*.

Kühl, N., Goutier, M., Baier, L., Wolff, C., and Martin, D. (2020). Human vs. supervised machine learning: Who learns patterns faster? *ArXiv preprint*, **abs/2012.03661**.

Kumar, A., Sattigeri, P., and Balakrishnan, A. (2018). Variational inference of disentangled latent concepts from unlabeled observations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Kümmerer, M., Theis, L., and Bethge, M. (2014). Deep gaze i: Boosting saliency prediction with feature maps trained on imagenet. *ArXiv preprint*, **abs/1411.1045**.

Lake, B. M. and Baroni, M. (2018). Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In J. G. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2879–2888. PMLR.

Lamb, A., Binas, J., Goyal, A., Serdyuk, D., Subramanian, S., Mitliagkas, I., and Bengio, Y. (2018). Fortified networks: Improving the robustness of deep networks by modeling the manifold of hidden representations. *ArXiv preprint*, **abs/1804.02485**.

LeCun, Y. and Manning, C. (2018). What innate priors should we build into the architecture of deep learning systems? Accessed: 2021-09-09.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, **1**(4), 541–551.

LeCun, Y., Haffner, P., Bottou, L., and Bengio, Y. (1999). Object recognition with gradient-based learning. In *Shape, contour and grouping in computer vision*, pages 319–345. Springer.

LeCun, Y., Huang, F. J., and Bottou, L. (2004). Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pages II–104. IEEE.

Lecuyer, M., Atlidakis, V., Geambasu, R., Hsu, D., and Jana, S. (2019). Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 656–672.

Leike, J., Martic, M., Krakovna, V., Ortega, P. A., Everitt, T., Lefrancq, A., Orseau, L., and Legg, S. (2017). Ai safety gridworlds. *ArXiv preprint*, **abs/1711.09883**.

Li, F.-F. (2015). How we're teaching computers to understand pictures. Accessed: 2021-09-09.

Li, Q., Haque, S., Anil, C., Lucas, J., Grosse, R. B., and Jacobsen, J. (2019a). Preventing gradient attenuation in lipschitz constrained convolutional networks. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 15364–15376.

Li, Z., Brendel, W., Walker, E. Y., Cobos, E., Muhammad, T., Reimer, J., Bethge, M., Sinz, F. H., Pitkow, Z., and Tolias, A. S. (2019b). Learning from brains how to regularize machines. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural*

Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, pages 9525–9535.

Lim, C. H., Urtasun, R., and Yumer, E. (2020). Hierarchical verification for adversarial robustness. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 6072–6082. PMLR.

Liu, R., Lehman, J., Molino, P., Such, F. P., Frank, E., Sergeev, A., and Yosinski, J. (2018a). An intriguing failing of convolutional neural networks and the coordconv solution. In S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 9628–9639.

Liu, Z., Luo, P., Wang, X., and Tang, X. (2018b). Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, **15**(2018), 11.

Locatello, F., Bauer, S., Lucic, M., Rätsch, G., Gelly, S., Schölkopf, B., and Bachem, O. (2019a). Challenging common assumptions in the unsupervised learning of disentangled representations. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 4114–4124. PMLR.

Locatello, F., Abbati, G., Rainforth, T., Bauer, S., Schölkopf, B., and Bachem, O. (2019b). On the fairness of disentangled representations. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 14584–14597.

Locatello, F., Tschannen, M., Bauer, S., Rätsch, G., Schölkopf, B., and Bachem, O. (2020a). Disentangling factors of variations using few labels. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Locatello, F., Poole, B., Rätsch, G., Schölkopf, B., Bachem, O., and Tschannen, M. (2020b). Weakly-supervised disentanglement without compromises. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 6348–6359. PMLR.

Mackowiak, R., Ardizzone, L., Kothe, U., and Rother, C. (2021). Generative classifiers as a basis for trustworthy image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2971–2981.

Madry, A. and Schmidt, L. (2018). A brief introduction to adversarial examples.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Mao, X., Qi, G., Chen, Y., Li, X., Duan, R., Ye, S., He, Y., and Xue, H. (2021). Towards robust vision transformer. *ArXiv preprint*, **abs/2105.07926**.

Marcos, D., Volpi, M., and Tuia, D. (2016). Learning rotation invariant convolutional filters for texture classification. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 2012–2017. IEEE.

Matthey, L., Higgins, I., Hassabis, D., and Lerchner, A. (2017). dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/.

McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, **5**(4), 115–133.

Michaelis, C., Mitzkus, B., Geirhos, R., Rusak, E., Bringmann, O., Ecker, A. S., Bethge, M., and Brendel,

W. (2019). Benchmarking robustness in object detection: Autonomous driving when winter is coming. *ArXiv preprint*, **abs/1907.07484**.

Milan, A., Leal-Taixé, L., Reid, I., Roth, S., and Schindler, K. (2016). MOT16: A benchmark for multi-object tracking. *ArXiv preprint*, **abs/1603.00831**.

Minsky, M. and Papert, S. (1969). *Perceptrons*. MIT Press, Cambridge, MA.

Mitchell, M. (2021). Why ai is harder than we think. *ArXiv preprint*, **abs/2104.12871**.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, New York.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., *et al.* (2015). Human-level control through deep reinforcement learning. *Nature*, **518**(7540), 529.

Montero, M. L., Ludwig, C. J., Costa, R. P., Malhotra, G., and Bowers, J. (2021). The role of disentanglement in generalisation. In *International Conference on Learning Representations*.

Moosavi-Dezfooli, S., Fawzi, A., and Frossard, P. (2016). Deepfool: A simple and accurate method to fool deep neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2574–2582. IEEE Computer Society.

Mu, N. and Gilmer, J. (2019). Mnist-c: A robustness benchmark for computer vision. *ArXiv preprint*, **abs/1906.02337**.

NASA (1969). One giant leap for mankind. `https://www.nasa.gov/mission_pages/apollo/apollo11.html`, Accessed: 2021-09-21.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning.

Niemeyer, M. and Geiger, A. (2021). Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464.

Noether, E. (1915). The finiteness theorem for invariants of finite groups. *Mathematische Annalen*, **77**, 89–92.

Olshausen, B. A. (2003). Learning sparse, overcomplete representations of time-varying natural images. In *Proceedings 2003 International Conference on Image Processing (Cat. No. 03CH37429)*, volume 1, pages I–41. IEEE.

Palmer, B. (2011). Why no two snowflakes are the same.

PapersWithCode (2021). Image classification on imagenet. [Online; accessed 1-October-2021].

Peters, J., Janzing, D., and Schölkopf, B. (2017). *Elements of causal inference: foundations and learning algorithms*. The MIT Press.

Prinz, W. (2006). Messung kontra augenschein. *Psychologische Rundschau*, **57**(2), 106–111.

Qian, R., Meng, T., Gong, B., Yang, M.-H., Wang, H., Belongie, S., and Cui, Y. (2021). Spatiotemporal contrastive video representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6964–6974.

Ridgeway, K. and Mozer, M. C. (2018). Learning deep disentangled embeddings with the f-statistic loss. In S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 185–194.

Roberts, D. A. (2021). Sgd implicitly regularizes generalization error.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, **65**(6), 386.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, **323**(6088), 533–536.

Rusak, E., Schott, L., Zimmermann, R. S., Bitterwolf, J., Bringmann, O., Bethge, M., and Brendel, W. (2020). A simple way to make neural networks robust against diverse image corruptions. In *European Conference on Computer Vision*, pages 53–69. Springer.

Rusak, E., Schneider, S., Gehler, P., Bringmann, O., Brendel, W., and Bethge, M. (2021). Adapting imagenet-scale models to complex distribution shifts with self-learning. *ArXiv preprint*, **abs/2104.12928**.

Schmidhuber, J. (1991). A possibility for implementing curiosity and boredom in model-building neural controllers. In *Proc. of the international conference on simulation of adaptive behavior: From animals to animats*, pages 222–227.

Schmidhuber, J. (1992). Learning factorial codes by predictability minimization. *Neural computation*, **4**(6), 863–879.

Schneider, S., Rusak, E., Eck, L., Bringmann, O., Brendel, W., and Bethge, M. (2020). Improving robustness against common corruptions by covariate shift adaptation. *Advances in Neural Information Processing Systems*, **33**.

Schölkopf, B. (2019). Causality for machine learning. *ArXiv preprint*, **abs/1911.10500**.

Schott, L., Rauber, J., Bethge, M., and Brendel, W. (2019). Towards the first adversarially robust neural network model on MNIST. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Schott, L., von Kügelgen, J., Träuble, F., Gehler, P., Russell, C., Bethge, M., Schölkopf, B., Locatello, F., and Brendel, W. (2021). Visual representation learning does not generalize strongly within the same domain.

Sejnowski, T. J. (2020). The unreasonable effectiveness of deep learning in artificial intelligence. *Proceedings of the National Academy of Sciences*, **117**(48), 30033–30038.

Sharma, Y., Ding, G. W., and Brubaker, M. A. (2019). On the effectiveness of low frequency perturbations. In S. Kraus, editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 3389–3396. ijcai.org.

Sholarin, M., Ayodele, I., Wogu, I. A. P., Omole, F., and Agoha, B. (2015). " man is the measure of all things " : A critical analysis of the sophist conception of man.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., *et al.* (2016). Mastering the game of go with deep neural networks and tree search. *nature*, **529**(7587), 484–489.

Simoncelli, E. P. and Olshausen, B. A. (2001). Natural image statistics and neural representation. *Annual review of neuroscience*, **24**(1), 1193–1216.

Sinz, F., Gerwinn, S., and Bethge, M. (2009). Characterization of the p-generalized normal distribution. *Journal of Multivariate Analysis*, **100**(5), 817–820.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2020). Score-based generative modeling through stochastic differential equations. *ArXiv preprint*, **abs/2011.13456**.

Stallkamp, J., Schlipsing, M., Salmen, J., and Igel, C. (2011). The german traffic sign recognition benchmark: a multi-class classification competition. In *The 2011 international joint conference on neural networks*, pages 1453–1460. IEEE.

Sun, C., Shrivastava, A., Singh, S., and Gupta, A. (2017). Revisiting unreasonable effectiveness of data in deep learning era. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 843–852. IEEE Computer Society.

Sutton, R. (2019). The bitter lesson. `http://incompleteideas.net/IncIdeas/BitterLesson.html`, Accessed: 2021-08-23.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. (2014). Intriguing properties of neural networks. In Y. Bengio and Y. LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Tramèr, F. and Boneh, D. (2019). Adversarial training and robustness for multiple perturbations. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5858–5868.

Träuble, F., Creager, E., Kilbertus, N., Locatello, F., Dittadi, A., Goyal, A., Schölkopf, B., and Bauer, S. (2021). On disentangled representations learned from correlated data. In *International Conference on Machine Learning*, pages 10401–10412. PMLR.

van Steenkiste, S., Locatello, F., Schmidhuber, J., and Bachem, O. (2019). Are disentangled representations helpful for abstract visual reasoning? In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 14222–14235.

Vapnik, V. (2013). *The nature of statistical learning theory*. Springer science & business media.

Vapnik, V. N. and Chervonenkis, A. Y. (1982). Necessary and sufficient conditions for the uniform convergence of means to their expectations. *Theory of Probability & Its Applications*, **26**(3), 532–553.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., *et al.* (2019). Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, **575**(7782), 350–354.

Voigtlaender, P., Krause, M., Osep, A., Luiten, J., Sekar, B. B. G., Geiger, A., and Leibe, B. (2019a). MOTS: multi-object tracking and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 7942–7951. Computer Vision Foundation / IEEE.

Voigtlaender, P., Krause, M., Osep, A., Luiten, J., Sekar, B. B. G., Geiger, A., and Leibe, B. (2019b). MOTS: multi-object tracking and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 7942–7951. Computer Vision Foundation / IEEE.

Von Humboldt, W., von Humboldt, W. F., *et al.* (1999). *Humboldt:'On Language': On the Diversity of Human Language Construction and Its Influence on the Mental Development of the Human Species*. Cambridge University Press.

Wikiquote (2021). Fred jelinek — wikiquote,. [Online; accessed 9-September-2021].

Willetts, M., Camuto, A., Rainforth, T., Roberts, S., and Holmes, C. (2019). Improving vaes' robustness to adversarial attack. *ArXiv preprint*, **abs/1906.00230**.

Wilson, A. C., Roelofs, R., Stern, M., Srebro, N., and Recht, B. (2017). The marginal value of adaptive gradient methods in machine learning. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4148–4158.

Winkler, J. K., Fink, C., Toberer, F., Enk, A., Deinlein, T., Hofmann-Wellenhof, R., Thomas, L., Lallas, A., Blum, A., Stolz, W., *et al.* (2019). Association between surgical skin markings in dermoscopic images and diagnostic performance of a deep learning convolutional neural network for melanoma recognition. *JAMA dermatology*, **155**(10), 1135–1141.

Wiyatno, R. R., Xu, A., Dia, O., and de Berker, A. (2019). Adversarial examples in modern machine learning: A review. *ArXiv preprint*, **abs/1911.05268**.

Wolpert, D. H. and Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, **1**(1), 67–82.

Xie, C., Wu, Y., van der Maaten, L., Yuille, A. L., and He, K. (2019). Feature denoising for improving adversarial robustness. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 501–509. Computer Vision Foundation / IEEE.

Xie, Q., Luong, M., Hovy, E. H., and Le, Q. V. (2020). Self-training with noisy student improves imagenet classification. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 10684–10695. IEEE.

Xu, N., Yang, L., Fan, Y., Yue, D., Liang, Y., Yang, J., and Huang, T. (2018). Youtube-vos: A large-scale video object segmentation benchmark. *ArXiv preprint*, **abs/1809.03327**.

Xu, Y., Xiao, T., Zhang, J., Yang, K., and Zhang, Z. (2014). Scale-invariant convolutional neural networks. *ArXiv preprint*, **abs/1411.6369**.

Yang, M., Liu, F., Chen, Z., Shen, X., Hao, J., and Wang, J. (2021). Causalvae: disentangled representation learning via neural structural causal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9593–9602.

Yin, F., Wang, Q.-F., Zhang, X.-Y., and Liu, C.-L. (2013). Icdar 2013 chinese handwriting recognition competition. In *2013 12th international conference on document analysis and recognition*, pages 1464–1470. IEEE.

Zhai, X., Kolesnikov, A., Houlsby, N., and Beyer, L. (2021). Scaling vision transformers. *ArXiv preprint*, **abs/2106.04560**.

Zhang, X., Watkins, Y., and Kenyon, G. T. (2018). Can deep learning learn the principle of closed contour detection? In *International Symposium on Visual Computing*, pages 455–460. Springer.

Zhang, X., Wu, X., and Du, J. (2019a). Challenge of spatial cognition for deep learning. *ArXiv preprint*, **abs/1908.04396**.

Zhang, Y., Hare, J. S., and Prügel-Bennett, A. (2019b). Deep set prediction networks. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3207–3217.

Zimmermann, R. S., Sharma, Y., Schneider, S., Bethge, M., and Brendel, W. (2021). Contrastive learning inverts the data generating process. *arXiv preprint arXiv:2102.08850*.

Appendix A

Publication 1:
Towards nonlinear disentanglement in natural data with temporal sparse coding

Published as a conference paper and as an oral at the ICLR 2021.

TOWARDS NONLINEAR DISENTANGLEMENT IN NATURAL DATA WITH TEMPORAL SPARSE CODING

David Klindt[*]
University of Tübingen
klindt.david@gmail.com

Lukas Schott[*]
University of Tübingen
lukas.schott@bethgelab.org

Yash Sharma[*]
University of Tübingen
yash.sharma@bethgelab.org

Ivan Ustyuzhaninov
University of Tübingen
ivan.ustyuzhaninov@bethgelab.org

Wieland Brendel
University of Tübingen
wieland.brendel@bethgelab.org

Matthias Bethge[‡]
University of Tübingen
matthias.bethge@bethgelab.org

Dylan M Paiton[‡]
University of Tübingen
dylan.paiton@bethgelab.org

ABSTRACT

Disentangling the underlying generative factors from data has so far been limited to carefully constructed scenarios. We propose a path towards natural data by first showing that the statistics of natural data provide enough structure to enable disentanglement, both theoretically and empirically. Specifically, we provide evidence that objects in natural movies undergo transitions that are typically small in magnitude with occasional large jumps, which is characteristic of a temporally sparse distribution. Leveraging this finding we provide a novel proof that relies on a sparse prior on temporally adjacent observations to recover the true latent variables up to permutations and sign flips, providing a stronger result than previous work. We show that equipping practical estimation methods with our prior often surpasses the current state-of-the-art on several established benchmark datasets without any impractical assumptions, such as knowledge of the number of changing generative factors. Furthermore, we contribute two new benchmarks, Natural Sprites and KITTI Masks, which integrate the measured natural dynamics to enable disentanglement evaluation with more realistic datasets. We test our theory on these benchmarks and demonstrate improved performance. We also identify non-obvious challenges for current methods in scaling to more natural domains. Taken together our work addresses key issues in disentanglement research for moving towards more natural settings.

1 INTRODUCTION

Natural scene understanding can be achieved by decomposing the signal into its underlying factors of variation. An intuitive approach for this problem assumes that a visual representation of the world can be constructed via a generative process that receives factors as input and produces natural signals as output (Bengio et al., 2013). This analogy is justified by the fact that our world is composed of distinct entities that can vary independently, but with regularity imposed by physics. What makes the approach appealing is that it formalizes representation learning by directly comparing representations to underlying ground-truth states, as opposed to the indirect evaluation of benchmarking against heuristic downstream tasks (e.g. object recognition). However, the core issue with this approach is *non-identifiability*, which means a set of possible solutions may all appear equally valid to the model, while only one identifies the true generative factors.

Our work is motivated by the question of whether the statistics of natural data will allow for the formulation of an identifiable model. Our core observation that enables us to make progress in

[*‡]Equal contribution. Code: `https://github.com/bethgelab/slow_disentanglement`

addressing this question is that *generative factors of natural data have sparse transitions*. To estimate these generative factors, we compute statistics on measured transitions of area and position for object masks from large-scale, natural, unstructured videos. Specifically, we extracted over 300,000 object segmentation mask transitions from YouTube-VOS (Xu et al., 2018; Yang et al., 2019) and KITTI-MOTS (Voigtlaender et al., 2019; Geiger et al., 2012; Milan et al., 2016) (discussed in detail in Appendix D). We fit generalized Laplace distributions to the collected data (Eq. 2), which we indicate with orange lines in Fig. 1. We see empirically that all marginal distributions of temporal transitions are highly sparse and that there exist complex dependencies between natural factors (e.g. motion typically affects both position and apparent size). In this study, we focus on the sparse marginals, which we believe constitutes an important advance that sets the stage for solving further issues and eventually applying the technology to real-world problems. With this information at hand, we are able to provide a stronger proof for capturing the underlying generative factors of the data up to permutations and sign flips that is not covered by previous work (Hyvärinen and Morioka, 2016; 2017; Khemakhem et al., 2020a). Thus, we present the first work, to the best of our knowledge, which proposes a theoretically grounded solution that covers the statistics observed in real videos.

Our contributions are: With measurements from unstructured natural video annotations we provide evidence that natural generative factors undergo sparse changes across time. We provide a proof of identifiability that relies on the observed sparse innovations to identify nonlinearly mixed sources up to a permutation and sign-flips, which we then validate with practical estimation methods for empirical comparisons. We leverage the natural scene information to create novel datasets where the latent transitions between frames follow natural statistics. These datasets provide a benchmark to evaluate how well models can uncover the true latent generative factors in the presence of realistic dynamics. We demonstrate improved disentanglement over previous models on existing datasets and our contributed ones with quantitative metrics from both the disentanglement (Locatello et al., 2018) and the nonlinear ICA community (Hyvärinen and Morioka, 2016). We show via numerous visualization techniques that the learned representations for competing models have important differences, even when quantitative metrics suggest that they are performing equally well.

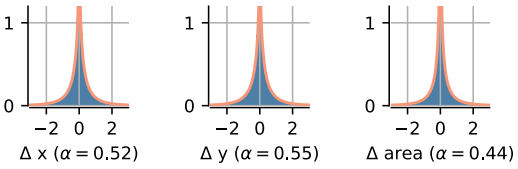


Figure 1: **Statistics of Natural Transitions**. The histograms show distributions over transitions of segmented object masks from natural videos for horizontal and vertical position as well as object size. The red lines indicate fits of generalized Laplace distributions (Eq. 2) with shape value α. Data shown is for object masks extracted from YouTube videos. See Appendix G for 2D marginals and corresponding analysis from the KITTI self-driving car dataset.

2 Related Work – Disentanglement and Nonlinear ICA

Disentangled representation learning has its roots in blind source separation (Cardoso, 1989; Jutten and Herault, 1991) and shares goals with fields such as inverse graphics (Kulkarni et al., 2015; Yildirim et al., 2020; Barron and Malik, 2012) and developing models of invariant neural computation (Hyvärinen and Hoyer, 2000; Wiskott and Sejnowski, 2002; Sohl-Dickstein et al., 2010) (see Bengio et al., 2013, for a review). A disentangled representation would be valuable for a wide variety of machine learning applications, including sample efficiency for downstream tasks (Locatello et al., 2018; Gao et al., 2019), fairness (Locatello et al., 2019; Creager et al., 2019) and interpretability (Bengio et al., 2013; Higgins et al., 2017; Adel et al., 2018). Since there is no agreed upon definition of disentanglement in the literature, we adopt two common measurable criteria: i) each encoding element represents a single generative factor and ii) the values of generative factors are trivially decodable from the encoding (Ridgeway and Mozer, 2018; Eastwood and Williams, 2018).

Uncovering the underlying factors of variation has been a long-standing goal in independent component analysis (ICA) (Comon, 1994; Bell and Sejnowski, 1995), which provides an identifiable solution for disentangling data mixed via an invertible linear generator receiving at most one Gaussian factor as input. Recent unsupervised approaches for nonlinear generators have largely been based on Variational Autoencoders (VAEs) (Kingma and Welling, 2013) and have assumed that the data is independent and identically distributed (*i.i.d.*) (Locatello et al., 2018), even though nonlinear methods that make this *i.i.d.* assumption have been proven to be *non-identifiable* (Hyvärinen and Pajunen,

1999; Locatello et al., 2018). Nonetheless, the bottom-up approach of starting with a nonlinear generator that produces well-controlled data has led to considerable achievements in understanding nonlinear disentanglement in VAEs (Higgins et al., 2017; Burgess et al., 2018; Rolinek et al., 2019; Chen et al., 2018), consolidating ideas from neural computation and machine learning (Khemakhem et al., 2020a), and seeking a principled definition of disentanglement (Ridgeway, 2016; Higgins et al., 2018; Eastwood and Williams, 2018).

Recently, Hyvärinen and colleagues (Hyvärinen and Morioka, 2016; 2017; Hyvärinen et al., 2018) showed that a solution to identifiable nonlinear ICA can be found by assuming that generative factors are conditioned on an additional observed variable, such as past states or the time index itself. This contribution was generalized by Khemakhem et al. (2020a) past the nonlinear ICA domain to any consistent parameter estimation method for deep latent-variable models, including the VAE framework. However, the theoretical assumptions underlying this branch of work do not account for the sparse transitions we observe in the statistics of natural scenes, which we discuss in further detail in appendix F.1.1. Another branch of work requires some form of supervision to demonstrate disentanglement (Szabó et al., 2017; Shu et al., 2019; Locatello et al., 2020). We select two of the above approaches, that are both different in their formulation and state-of-the-art in their respective empirical settings, Hyvärinen and Morioka (2017) and Locatello et al. (2020), for our experiments below. The motivation of our method and dataset contributions is to address the limitations of previous approaches and to enable unsupervised disentanglement learning in more naturalistic scenarios.[1]

The fact that physical processes bind generative factors in temporally adjacent natural video segments has been thoroughly explored for learning in neural networks (Hinton, 1990; Földiák, 1991; Mitchison, 1991; Wiskott and Sejnowski, 2002; Denton and Birodkar, 2017). We propose a method that uses time information in the form of an L_1-*sparse* temporal prior, which is motivated by the natural scene measurements presented above as well as by previous work (Simoncelli and Olshausen, 2001; Olshausen, 2003; Hyvärinen et al., 2003; Cadieu and Olshausen, 2012). Such a prior would intuitively allow for sharp changes in some latent factors, while most other factors remain unchanged between adjacent time-points. Almost all similar methods are variants of slow feature analysis (SFA, Wiskott and Sejnowski, 2002), which measure slowness in terms of the Euclidean (i.e. L_2, or log Gaussian) distance between temporally adjacent encodings. Related to our approach, a probabilistic interpretation of SFA has been previously proposed (Turner and Sahani, 2007), as well as extensions to variational inference (Grathwohl and Wilson, 2016). Additionally, Hashimoto (2003) suggested that a sparse (Cauchy) slowness prior improves correspondence to biological complex cells over the L_2 slowness prior in a two-layer model. However, to the best of our knowledge, an L_1 temporal prior has previously only been used in deep auto-encoder frameworks when applied to semi-supervised tasks (Mobahi et al., 2009; Zou et al., 2012), and was mentioned in Cadieu and Olshausen (2012), who used an L_2 prior, but claimed that an L_1 prior performed similarly on their task. Similar to Hyvärinen et al. (Hyvärinen and Morioka, 2016; Hyvärinen et al., 2018), we only assume that the latent factors are temporally dependent, thus avoiding assuming knowledge of the number of factors where the two observations differ (Shu et al., 2019; Locatello et al., 2020).

Most of the standard datasets for disentanglement (dSprites (Matthey et al., 2017), Cars3D (Reed et al., 2015), SmallNORB (LeCun et al., 2004), Shapes3D (Kim and Mnih, 2018), MPI3D (Gondal et al., 2019)) have been compiled into a disentanglement library (DisLib) by Locatello et al. (2018). However, all of the DisLib datasets are limited in that the data generating process is independent and identically distributed (*i.i.d.*) and all generative factors are assumed to be discrete. In a follow-up study, Locatello et al. (2020) proposed combining pairs of images such that only k factors change, as this matches their modeling assumptions required to prove identifiability. Here, $k \in \mathcal{U}\{1, D-1\}$ and D denotes the number of ground-truth factors, which are then sampled uniformly. We additionally use the measurements from Fig. 1 to construct datasets for evaluating disentanglement that have time transitions which directly correspond to natural dynamics.

[1] As in slow feature analysis, we consider learning from videos without labels as *unsupervised*.

3 THEORY

3.1 GENERATIVE MODEL

We have provided evidence to support the hypothesis that generative factors of natural videos have sparse temporal transitions (see Fig. 1). To model this process, we assume temporally adjacent input pairs $(\mathbf{x}_{t-1}, \mathbf{x}_t)$ coming from a nonlinear generator that maps factors to images $\mathbf{x} = g(\mathbf{z})$, where generative factors are dependent over time:

$$p(\mathbf{z}_t, \mathbf{z}_{t-1}) = p(\mathbf{z}_t|\mathbf{z}_{t-1})p(\mathbf{z}_{t-1}). \tag{1}$$

Assume the observed data $(\mathbf{x}_t, \mathbf{x}_{t-1})$ comes from the following generative process, where different latent factors are assumed to be independent (cf. Appendix F.2):

$$\mathbf{x} = g(\mathbf{z}), \quad p(\mathbf{z}_{t-1}) = \prod_{i=1}^{d} p(z_{t-1,i}), \quad p(\mathbf{z}_t|\mathbf{z}_{t-1}) = \prod_{i=1}^{d} \frac{\alpha\lambda}{2\Gamma(1/\alpha)} \exp -(\lambda|z_{t,i} - z_{t-1,i}|^{\alpha}), \tag{2}$$

where λ is the distribution rate, $p(\mathbf{z}_{t-1})$ is a factorized Gaussian prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$ (as in Kingma and Welling, 2013) and $p(\mathbf{z}_t|\mathbf{z}_{t-1})$ is a factorized generalized Laplace distribution (Subbotin, 1923) with shape parameter α, which determines the shape and especially the kurtosis of the function.[2] Intuitively, smaller α implies larger kurtosis and sparser temporal transitions of the generative factors (special cases are Gaussian, $\alpha = 2$, and Laplacian, $\alpha = 1$). Critically, for our proof we assume $\alpha < 2$ to ensure that temporal transitions are sparse. The novelty of our approach lies in our explicit modeling of sparse transitions that cover the statistics of natural data, which results in a stronger identifiability proof than previously achieved (see Appendix F.1.1 for a more detailed comparison with Hyvärinen and Morioka, 2017; Khemakhem et al., 2020a).

3.2 IDENTIFIABILITY PROOF

Theorem 1 *For a ground-truth $(g^*, \lambda^*, \alpha^*)$ and a learned (g, λ, α) model as defined in Eq. (2), if the functions g^* and g are injective and differentiable almost everywhere, $\lambda^* = \lambda$, $\alpha^* = \alpha < 2$ (i.e. there is no model misspecification) and the distributions of pairs of images generated from the priors $\mathbf{z}^* \sim p^*(\mathbf{z})$ and $\mathbf{z} \sim p(\mathbf{z})$ generated as $(g^*(\mathbf{z}^*_{t-1}), g^*(\mathbf{z}^*_t))$ and $(g(\mathbf{z}_{t-1}), g(\mathbf{z}_t))$, respectively, are matched almost everywhere, then $g = g^* \circ \sigma$, where σ is composed of a permutation and sign flips.*

The formal proof is provided in Appendix A.1. Similar to linear ICA, but in the temporal domain, we have to assume that the transitions of generative factors across time be non-Gaussian. Specifically, if the temporal changes of ground-truth factors are sparse, then the only generator consistent with the observations is the ground-truth one (up to a permutation and sign flips). The main idea behind the proof is to represent g as $g^* \circ h$ and note that if h were not a permutation, then the distributions $((g^* \circ h)(\mathbf{z}_{t-1}), (g^* \circ h)(\mathbf{z}_t))$ and $(g^*(\mathbf{z}^*_{t-1}), g^*(\mathbf{z}^*_t))$ would not match, due to the injectivity of g^*. Whether or not these distributions are the same is equivalent to whether or not the distributions of pairs (z_{t-1}, z_t) and $(h(z_{t-1}), h(z_t))$ are the same. For these distributions to be the same, the function h must preserve the Gaussian marginal for the first time step as well as the joint distribution, implying that it must preserve both the vector lengths and distances in the latent space. As we argue in the extended proof, this can only be the case if h is a composition of permutations and sign flips.

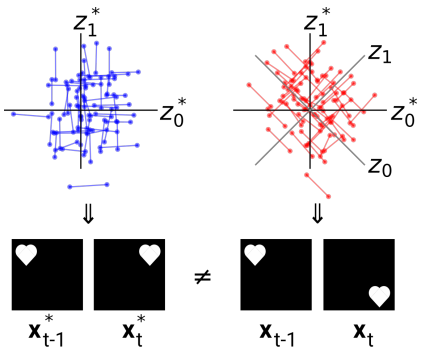


Figure 2: **Proof Intuition**. Latent representation and example generated image pairs for ground-truth (blue) and entangled (red) model. See text below for details.

Intuition Fig. 2 illustrates, by contradiction, why the model defined in Eq. (2) is identifiable. We consider temporal pairs of latents represented by connected points. A sparse transition prior encourages axis-alignment, as can be seen from the Laplace transition prior in the third image of Fig. 3.

[2]For a stationary stochastic process, $p(\mathbf{z}_{t-1})$ represents the instantaneous marginal distribution and $p(\mathbf{z}_t|\mathbf{z}_{t-1})$ the transition distribution. In case of an autoregressive process with non-Gaussian innovations with finite variance, it follows from the central limit theorem that the marginal distribution converges to a Gaussian in the limit of large λ.

This results in lines that are parallel with the axes in both the ground truth (left, blue, $\mathbf{z}^*$) and learned model (right, red, $\mathbf{z}$). In this example, z_0^* corresponds to horizontal position, while z_1^* corresponds to vertical position. The learned model must satisfy two criteria: (1) the latent factors should match the sparse prior (axis-aligned) and (2) the generated image pairs should match the ground-truth image pairs. If the learned latent factors were mismatched, for example by rotation, then the image pair distributions would not be matched. In this example, the ground truth model would produce image pairs with typically vertical or horizontal transitions, while the learned model pairs result in mostly diagonal transitions. Thus, the learned model cannot satisfy both criteria without aligning the latent axes with the ground-truth axes.

3.3 Slow Variational Autoencoder

In order to validate our proof, we must choose a probabilistic latent variable model for estimating the data density. We chose to build upon the framework of VAEs because of their efficiency in estimating a variational approximation to the ground truth posterior of a deep latent variable model (Kingma and Welling, 2013). We will refer to this model as *SlowVAE*. In Appendix B we note shortcomings of such an approach and test an alternative flow-based model.

The standard VAE objective assumes *i.i.d.* data and a standard normal prior with diagonal covariance on the learned latent representations $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. To extend this to sequences, we assume the same functional form for our model prior as in Eq. (1) and Eq. (2). The posterior of our model is independent across time steps. Specifically,

$$q(\mathbf{z}_t, \mathbf{z}_{t-1}|\mathbf{x}_t, \mathbf{x}_{t-1}) = q(\mathbf{z}_t|\mathbf{x}_t)\, q(\mathbf{z}_{t-1}|\mathbf{x}_{t-1}), \quad q(\mathbf{z}|\mathbf{x}) = \prod_{i=1}^{d} \mathcal{N}(\mu_i(\mathbf{x}), \sigma_i^2(\mathbf{x})), \tag{3}$$

where $\mu_i(\mathbf{x})$ and $\sigma_i^2(\mathbf{x})$ are the input-dependent mean and variance of our model's posterior. We visualize this combination of priors and posteriors in Fig. 3. For a given pair of inputs $(\mathbf{x}_t, \mathbf{x}_{t-1})$, the full evidence lower bound (ELBO, which we derive in Appendix A.2) can be written as

$$\begin{aligned}\mathcal{L}(\mathbf{x}_t, \mathbf{x}_{t-1}) = E_{q(\mathbf{z}_t, \mathbf{z}_{t-1}|\mathbf{x}_t, \mathbf{x}_{t-1})}[\log p(\mathbf{x}_t, \mathbf{x}_{t-1}|\mathbf{z}_t, \mathbf{z}_{t-1})] - D_{KL}(q(\mathbf{z}_{t-1}|\mathbf{x}_{t-1})|p(\mathbf{z}_{t-1})) \\ - \gamma\, E_{q(\mathbf{z}_{t-1}|\mathbf{x}_{t-1})}[D_{KL}(q(\mathbf{z}_t|\mathbf{x}_t)|p(\mathbf{z}_t|\mathbf{z}_{t-1}))],\end{aligned} \tag{4}$$

where γ is a regularization term for the sparsity prior, analogous to β in β-VAEs (Higgins et al., 2017) (technically, Eq. 4 is only an ELBO with $\gamma \leq 1$). The first term on the right-hand side is the log-likelihood (i.e. the negative reconstruction error, with $p(\mathbf{x}_t, \mathbf{x}_{t-1}|\mathbf{z}_t, \mathbf{z}_{t-1})$ parameterized by the decoder of the VAE), the second term is the KL to a normal prior as in the standard VAE and the last term is an expectation of the KL between the posterior at time step t and the conditional prior $p(\mathbf{z}_t|\mathbf{z}_{t-1})$. The expectation in the last term is taken over samples from the posterior at the previous time step $q(\mathbf{z}_{t-1}|\mathbf{x}_{t-1})$. We observed empirically that taking the mean, $\mu(\mathbf{x}_{t-1})$, as a single sample produces good results, analogous to the log-likelihood that is typically evaluated at a single sample from the posterior (see Blei et al. (2017) for context).

In practice, we need to choose α, λ, and γ. For the latter two, we can perform a random search for hyperparameters, as we discuss below. For the former, any $\alpha < 2$ would break the general rotation symmetry by having an optimum for axis-aligned representations, which theorem 1 includes as a requirement for identifiability. As can be seen in Figs. 1 and 11, $\alpha \approx 0.5$ provides the best fit to the ground-truth marginals. However, we used $\alpha = 1$ as a parsimonious choice for SlowVAE, since the Laplace is a well-understood distribution that allows us to derive a simple closed-form solution for the ELBO in Eq. 4, which we derive in Appendix A.2.

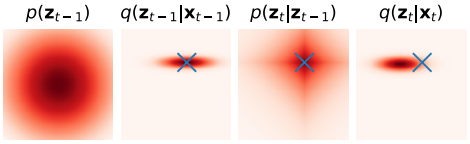


Figure 3: **SlowVAE illustration.** The prior and posterior for a two-dimensional latent space. Left to right: Normal prior for $t-1$, posterior for $t-1$, conditional Laplace prior for t, and posterior for t. The blue cross in the right three plots indicates the mean of the posterior for $t-1$.

3.4 Towards an Approximate Theory of Disentanglement

A number of our theoretical assumptions are violated in practice: After non-convex optimization, on a finite data sample, the distributions $p(\mathbf{x}_t, \mathbf{x}_{t-1})$ and $p^*(\mathbf{x}_t, \mathbf{x}_{t-1})$ are probably not perfectly

matched. In addition, the model assumptions on $p(\mathbf{z}_t, \mathbf{z}_{t-1})$ likely do not fully match the distribution of the ground truth factors. For example, the model may be misspecified such that $\alpha \neq \alpha^*$ or $\lambda \neq \lambda^*$, or the chosen family of distributions may be incorrect altogether. In the following section we will present results on several datasets where the marginal distributions $p(\mathbf{z}_{t-1})$ are drawn from a Uniform (not Normal) distribution, and some of them are over unordered sets (categories) or bounded periodic spaces (rotation). Also, in practice the model latent space is usually chosen to have more dimensions than the ground truth generative model. On real data, factors of variation may be dependent (Träuble et al., 2020; Yang et al., 2020). We show this is the case on YouTube-VOS and KITTI-MOTS in Appendix G and we provide evidence that breaking these dependencies has no clear consequence on disentanglement in Appendix F.2. A more formal treatment of dependence is done by Khemakhem et al. (2020b) who relax the independence assumption of ICA to Independently Modulated Components Analysis (IMCA) and introduce a family of conditional energy-based models that are identifiable up to simple transformations. Furthermore, the hypothesis class $\mathcal{G}$ of learnable functions in the VAE architecture may not contain the invertible ground truth generator $g^* \notin \mathcal{G}$, if it exists at all (e.g. occlusions may already lead to non-invertibility). Despite these violations, we consider it a strength of our method that the practical implementation still achieves improved disentanglement over previous approaches. However, we note understanding the impact of these violations as an important focus area for continued progress towards developing a practical yet theoretically supported method for disentanglement on natural scenes.

4 Datasets with Natural Transitions

While the standard datasets compiled by DisLib are an important step towards real-world applications, they still assume the data is *i.i.d.*. As described in section 2, Locatello et al. (2020) proposed uniformly sampling the number of factors to be changed, $k = \text{Rnd}$, and changing said factors by uniformly sampling over the possible set of values. What we refer to as "UNI" is a dataset variant modeled after the described scheme (Locatello et al., 2020) (further details in Appendix D). Considering our natural data analysis presented in Figure 1, such transitions are certainly unnatural. Given the current state of evaluation, we provide a set of incrementally more natural datasets which are otherwise comparable to existing work. We propose that said datasets should be included in the standard benchmark suite to provide a step towards disentanglement in natural data.

(1) Laplace Transitions (LAP) is a procedure for constructing image pairs from DisLib datasets by sampling from a sparse conditional distribution. For each ground-truth factor, the first value in the pair is chosen *i.i.d.* from the dataset and the second is chosen by weighting nearby factor values using Laplace distributed probabilities. LAP is a step towards natural data that closely resembles previous extensions of DisLib datasets to the time domain, but in a way that matches the marginal distribution of natural transitions (see Appendix D.2 for more details).
(2) Natural Sprites consists of pairs of rendered sprite images with generative factors sampled from real YouTube-VOS transitions. For a given image pair, the position and scale of the sprites are set using measured values from adjacent time points in YouTube-VOS. The sprite shapes and orientations are simple, like dSprites, and are fixed for a given pair. While fixing shape follows the natural transitions of objects, it is unclear how to accurately estimate object orientation from the masks, and thus we fixed the factor to avoid introducing artificial transitions. We additionally consider a version that is discretized to the same number of object states as dSprites, which i) allows us to use the standard DisLib evaluation metrics and ii) helps isolate the effect of including natural transitions from the effect of increasing data complexity (see Appendix D.4 for more details).
(3) KITTI Masks is composed of pedestrian segmentation masks from the autonomous driving vision benchmark KITTI-MOTS, thus with natural shapes and continuous natural transitions in all underlying factors. We consider adjacent frames which correspond to mean$(\Delta t) = 0.05s$ in physical time (we report the mean because of variable sampling rates in the original data); as well as frames with a larger temporal gap of mean$(\Delta t) = 0.15s$, which corresponds to samples of pairs that are at most 5 frames apart. We show in Appendix G.3 that SlowVAE disentanglement performance increases and then plateaus as we continue to increase mean(Δt).

In summary, we construct datasets with (1) imposed sparse transitions, (2) augmented with natural continuous generative factors using measurements from unstructured natural videos, as well as (3) data from unstructured natural videos themselves, but provided as segmentation masks to ensure visual complexity is manageable for current methods. For the provided datasets, the object categories

Model	Data	BetaVAE	FactorVAE	MIG	MCC	DCI	Modularity	SAP
PCL	dSprites (Uniform)	80.1 (0.4)	62.1 (0.9)	16.0 (7.4)	41.6 (1.5)	42.4 (1.2)	**99.7** (0.6)	6.0 (2.7)
Ada-GVAE	dSprites (Uniform)	88.0 (2.7)	73.1 (3.9)	17.3 (4.7)	46.0 (4.8)	32.3 (4.6)	93.3 (1.8)	6.6 (2.0)
SlowVAE	dSprites (Uniform)	87.0 (5.1)	75.2 (11.1)	**28.3** (11.5)	**58.8** (8.9)	47.7 (8.5)	86.9 (2.8)	4.4 (2.0)
PCL	dSprites (Laplace)	99.9 (0.1)	94.7 (3.1)	19.2 (3.1)	67.9 (3.3)	52.0 (3.5)	93.2 (0.9)	8.1 (1.6)
Ada-GVAE	dSprites (Laplace)	91.4 (1.6)	83.0 (5.9)	21.8 (4.9)	56.9 (4.2)	39.0 (4.2)	87.6 (1.8)	7.2 (0.3)
SlowVAE	dSprites (Laplace)	100.0 (0.0)	97.5 (3.0)	**29.5** (9.3)	69.8 (2.3)	**65.4** (3.6)	**96.5** (1.6)	8.1 (3.0)
PCL	Natural (Discrete)	82.4 (6.7)	68.3 (8.0)	7.8 (2.8)	50.2 (4.2)	14.3 (3.0)	88.9 (3.1)	2.5 (1.1)
Ada-GVAE	Natural (Discrete)	83.4 (1.1)	74.8 (4.4)	14.5 (3.2)	51.6 (2.5)	21.8 (2.9)	87.8 (2.5)	5.3 (1.4)
SlowVAE	Natural (Discrete)	82.6 (2.2)	76.2 (4.8)	11.7 (5.0)	52.6 (4.1)	18.9 (5.5)	88.1 (3.6)	4.4 (2.3)

Table 1: Mean and standard deviation (s.d.) metric scores across 10 random seeds. PCL is a scaled-up implementation of the method described by Hyvärinen and Morioka (2017), leveraging the encoding architecture and training hyperparameters specified in appendix E. Ada-GVAE is the leading method proposed by Locatello et al. (2020). Bold indicates statistical significance above the next highest score (independent T-test, $p < 0.05$). Red indicates statistical significance below the next lowest score. Results for additional datasets and models are in Table 2 and Appendix G.

never change across transitions – reflecting natural object permanence. Finally, as (2) and (3) use factor transitions measured from natural videos, they exhibit any natural statistical structure present for those factors, such as natural dependencies (further discussion is in Appendix F.2).

5 Experiments

5.1 Empirical Studies

We evaluate models using the DisLib implementation for the following supervised metrics: BetaVAE (Higgins et al., 2017); FactorVAE (Kim and Mnih, 2018); Mutual Information Gap (MIG; Chen et al., 2018); Disentanglement, Compactness, and Informativeness (DCI / Disentanglement; Eastwood and Williams, 2018); Modularity (Ridgeway and Mozer, 2018); and Separated Attribute Predictability (SAP; Kumar et al., 2018) (see Appendix C for metric details). None of the DisLib metrics support ground-truth labels with continuous variation, which is required for evaluation on the continuous Natural Sprites and KITTI Masks datasets. To reconcile this, we measure the Mean Correlation Coefficient (MCC), a standard metric in the ICA literature that is applicable to continuous variables. We report mean and standard deviation across 10 random seeds.

In order to select the conditional prior regularization and the prior rate in an unsupervised manner, we perform a random search over $\gamma \in [1, 16]$ and $\lambda \in [1, 10]$ and compute the recently proposed unsupervised disentanglement ranking (UDR) scores (Duan et al., 2020). We notice that the optimal values are close to $\gamma = 10$ and $\lambda = 6$ on most datasets, and thus use these values for all experiments. We leave finding optimal values for specific datasets to future work, but note that it is a strong advantage of our approach that it works well with the same model specification across 13 datasets (counting LAP and UNI for DisLib and optional discretization for Natural Sprites), addressing a concern posed in (Locatello et al., 2018). Additional details on model selection and training can be found in Appendix E. Although we train on image pairs, our model does not need paired data points at test time. For all visualizations, we pick the models with the highest average score across the DisLib metrics.

To compare our model fairly against other methods that also take image pairs as inputs, we also present performance for Permutation-Contrastive Learning from nonlinear ICA (PCL, Hyvärinen and Morioka, 2017) and Ada-GVAE, the leading method in the study by (Locatello et al., 2020). We scaled up the implementation of PCL for evaluation on our high-dimensional pixel inputs, and note this method does not have any hyperparameters. For Ada-GVAE, following the paper's recommendations, we select β (per dataset) using the considered parameter set $[1, 2, 4, 6, 8, 16]$, and use the reconstruction loss as the unsupervised model selection criterion (Locatello et al., 2020).

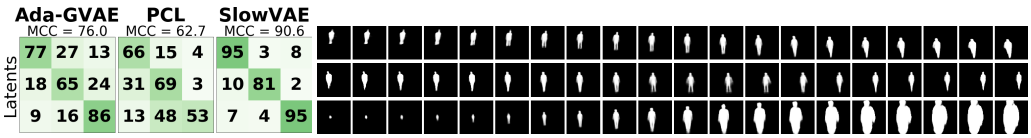


Figure 4: **KITTI Masks** (mean$(\Delta t) = 0.15s$)**.** (Left) MCC correlation matrix of the top 3 latents corresponding to y-position, x-position and scale. (Right) Images produced by varying the SlowVAE latent unit that corresponds to the corresponding row in the MCC matrix.

5.2 Results on DisLib and New Benchmarks

In Table 1 we demonstrate favorable performance compared to PCL and Ada-GVAE across all applicable metrics for discrete ground-truth variable datasets. The relative improvement on UNI is particularly surprising given the drastic mismatch between UNI and SlowVAE's assumptions. In Appendix G, we report results for the remaining DisLib datasets, where the observed dSprites results largely transfer. We also outperform PCL with a (flow-based) exact likelihood implementation of our slow transition prior in Appendix F.1.1. In Appendix F.3, we show that a model with an L_2 transition ($\alpha = 2$) prior performs much worse, supporting our theoretical prediction.

On the KITTI Masks dataset, one source of variation in the data is the average temporal separation within pairs of images mean(Δt). We present two settings (mean$(\Delta t) = 0.05s$, mean$(\Delta t) = 0.15s$) and observe a comparative increase in MCC for the latter (Table 2). Namely, the increase in performance for larger time gap is more pronounced with SlowVAE than the baselines, resulting in a statistically significant MCC gain. We provide details on the settings and ablate over the mean(Δt) parameter in Appendix G.3, where we observe a positive trend between mean(Δt) and MCC (reflecting Table 2, in Oord et al., 2018). Finally, we also verify that the transition distributions remain sparse despite the increase in this parameter (Appendix G.3). In Fig. 4, we can see that SlowVAE has learned latent dimensions which have correspondence with the estimated ground truth factors of x/y-position and scale.

Model	Data	MCC
PCL	Natural (Continuous)	51.7 (3.0)
Ada-GVAE	Natural (Continuous)	48.4 (4.8)
SlowVAE	Natural (Continuous)	49.1 (4.0)
PCL	Kitti (mean$(\Delta t) = 0.05s$)	52.6 (5.1)
Ada-GVAE	Kitti (mean$(\Delta t) = 0.05s$)	62.6 (7.5)
SlowVAE	Kitti (mean$(\Delta t) = 0.05s$)	66.1 (4.5)
PCL	Kitti (mean$(\Delta t) = 0.15s$)	58.5 (3.3)
Ada-GVAE	Kitti (mean$(\Delta t) = 0.15s$)	67.6 (6.7)
SlowVAE	Kitti (mean$(\Delta t) = 0.15s$)	**79.6** (5.8)

Table 2: Continuous ground-truth variable datasets. See Table 1 for details.

Locatello et al. (2018) showed that all *i.i.d.* models performed similarly across the DisLib datasets and metrics when testing was carefully controlled. However, in Fig. 5 we observe that the different modeling assumptions result in differences in representation quality. To construct the visuals, we first compute the sorted correlation matrix between the latents (rows) and generative factors (columns), which we visualize as a correlation matrices. The matrices are sorted via linear sum assignment such that each ground-truth factor is non-greedily associated with the latent variable with highest correlation (Hyvärinen and Morioka, 2016). Below the matrices are scatter plots that reveal the decodability of the assigned latent factors. In each scatter plot, the horizontal axis indicates the ground truth value, the vertical axis indicates the corresponding latent value, and the colors indicate object shape. The models displayed are those with the maximum average score across evaluated metrics.

The latent space visualizations use the known ground-truth factors to aid in understanding how each factor is encoded in a way that is more informative than exclusively visualizing latent traversals or embeddings of pairs of latent units (Cheung et al., 2014; Chen et al., 2016; Szabó et al., 2017; Ma et al., 2018). For example, in the third row, we observe that several models have a sinusoidal variation with frequencies $\sim \omega, 2\omega$, and 4ω, which correspond to the three distinct rotational symmetries of the shapes: heart, ellipse and square. This directly impacts MCC performance (third row in the MCC matrix), which measures rank correlation between the matching latent factor (an angular variable) and the ground truth, which encodes the angles with monotonically increasing indices. Furthermore, the square has a four-fold rotational symmetry and repeats after 90°, but it is represented in a full 360° rotation in the DisLib ground truth encoding format, resulting in different ground truth labels for identical input images.

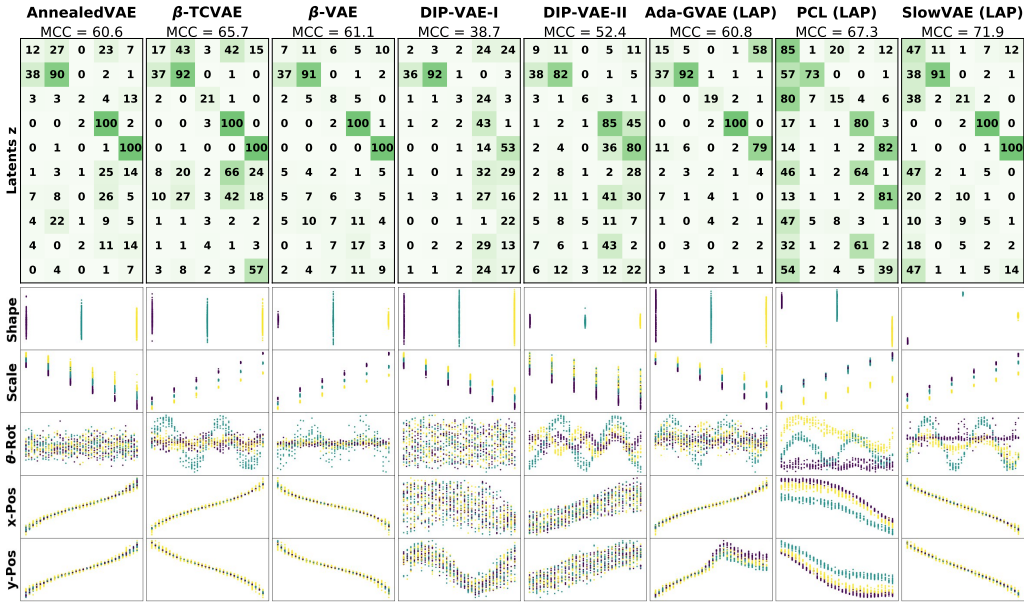


Figure 5: **DSprites Latent Representations:** (Top) shows absolute MCC between generative and model factors (rows are rearranged for maximal correlation on the main diagonal). The columns correspond to generative factors (shape, scale, rotation, x/y-position) and the values correspond to percent correlation. A more diagonal structure in the upper half corresponds to a better one-to-one mapping between generative and latent factors. (Bottom) shows individual latent dimensions (y-axis) over the matched generative factors (x-axis). Colors encode shapes: heart/yellow, ellipse/turquoise, and square/purple.

A similar observation can be made with respect to the categorical factors, which are also represented as ordinal ground truth variables. For example, the PCL correlation score (top left element in the PCL MCC matrix) is quite high, while the corresponding shape correlation score for SlowVAE is quite low. However, if we consider the shape scatter plots, we clearly see that SlowVAE separates the three shapes more distinctively than PCL, only in an order that differs from the ground truth. One solution is to modify MCC to report the maximum correlation over all permutations of the ground truth assignments, although brute force methods for this would scale poorly with the number of categories. We also note that datasets where we see small performance differences among models (e.g., Cars3D) have significantly more discrete categories (e.g., 183) than the other datasets ($3 - 6$). This could also explain why all models considered in Table 1 and 2 perform comparably on the Natural Sprites datasets, where unlike KITTI Masks the ground truth evaluation includes categorical and angular variables. We note that properly evaluating disentanglement is an ongoing area of research (Duan et al., 2020), with notable preliminary results in recent work (Higgins et al., 2018; Bouchacourt et al., 2021; Tonnaer et al., 2020).

6 Conclusion

We provide evidence to support the hypothesis that natural scenes exhibit highly sparse marginal transition probabilities. Leveraging this finding, we contribute a novel nonlinear ICA framework that is provably identifiable up to permutations and sign-flips — a stronger result than has been achieved previously. With the SlowVAE model we provide a parsimonious implementation that is inspired by a long history of learning visual representations from temporal data (Sutton, 1988; Hinton, 1990; Földiák, 1991). We apply this model to current metric-based disentanglement benchmarks to demonstrate that it outperforms existing approaches (Locatello et al., 2020; Hyvärinen and Morioka, 2017) on aggregate without any tuning of hyperparameters to individual datasets. We also provide novel video dataset benchmarks to guide disentanglement research towards more natural domains.

We observe that these datasets have complex dependencies that our theory will have to be extended to account for, although we demonstrate with empirical comparisons the efficacy of our approach. In addition to Natural Sprites and KITTI Masks, we suggest that YouTube-VOS will be valuable as a large-scale dataset that is unconstrained by object type and scenario for more advanced models. Variance in such categorical factors is problematic for evaluation due to the cited drawbacks of existing quantitative metrics, which should be addressed in tandem with scaling to natural data. Taken together, our dataset and model proposals set the stage for utilizing knowledge of natural scene statistics to advance unsupervised disentangled representation learning.

In our experiments we see that approximate identification as measured by the different disentanglement metrics increases despite violations of theoretical assumptions, which is in line with prior studies (Shu et al., 2019; Khemakhem et al., 2020a; Locatello et al., 2020). Nevertheless, future work should address gaining a better understanding of the theoretical and empirical consequences of such model misspecifications, in order to make the theory of disentanglement more predictive about empirically found solutions.

ACKNOWLEDGEMENTS

The authors would like to thank Francesco Locatello for valuable discussions and providing numerical results to facilitate our experimental comparisons. Additionally, we thank Luigi Gresele, Matthias Tangemann, Roland Zimmermann, Robert Geirhos, Matthias Kümmerer, Cornelius Schröder, Charles Frye, and Sarah Master for helpful feedback in preparing the manuscript. Finally, the authors would like to thank Johannes Ballé, Jon Shlens and Eero Simoncelli for early discussions related to the ideas developed in this paper.

This work was supported by the Deutsche Forschungsgemeinschaft (DFG) in the priority program 1835 under grant BR2321/5-2 and by SFB 1233, Robust Vision: Inference Principles and Neural Mechanisms (TP3), project number: 276693517. We thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting LS and YS. DP was supported by the German Federal Ministry of Education and Research (BMBF) through the Tübingen AI Center (FKZ: 01IS18039A). IU, WB, and MB are supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DoI/IBC) contract number D16PC00003. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/IBC, or the U.S. Government.

The authors declare no conflicts of interests.

BROADER IMPACT

Representation learning is at the heart of model building for cognition. Our specific contribution is focused on core methods for modeling natural videos and the datasets used are more simplistic than real-world examples. However, foundational research on unsupervised representation learning has potentially large impact on AI for advancing the power of self-learning systems.

The broader field of representation learning has a large number of focused research directions that span machine learning and computational neuroscience. As such, the application space for this work is vast. For example, applications in unsupervised analysis of complicated and unintuitive data, such as medical imaging and gene expression information, have great potential to solve fundamental problems in health sciences. A future iteration of our disentangling approach could be used to encode such complicated data into a lower-dimensional and more understandable space that might reveal important factors of variation to medical researchers. Another important and complex modeling space that could potentially be improved by this line of research is in environmental sciences and combating global climate change.

Nonetheless, we acknowledge that any machine learning method can be used for nefarious purposes, which can be mitigated via effective, scientifically informed communication, outreach, and policy direction. We unconditionally denounce the use of derivatives of our work for weaponized or wartime applications. Additionally, due to the lack of interpretability generally found in modern deep learning approaches, it is possible for practitioners to inadvertently introduce harmful biases or errors in machine learning applications. Although we certainly do not solve this problem, our focus on providing identifiable solutions to representation learning is likely beneficial for both interpretability and fairness in machine learning.

REFERENCES

Tameem Adel, Zoubin Ghahramani, and Adrian Weller. Discovering interpretable representations for both deep generative and discriminative models. In *International Conference on Machine Learning*, pages 50–59, 2018.

Jonathan T Barron and Jitendra Malik. Shape, albedo, and illumination from a single image of an unknown object. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 334–341. IEEE, 2012.

Anthony J Bell and Terrence J Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, Apr 2017. ISSN 1537-274X. doi: 10.1080/01621459.2017.1285773. URL http://dx.doi.org/10.1080/01621459.2017.1285773.

Diane Bouchacourt, Mark Ibrahim, and Stéphane Deny. Addressing the topological defects of disentanglement via distributed operators, 2021.

Samuel R. Bowman, L. Vilnis, Oriol Vinyals, Andrew M. Dai, R. Józefowicz, and S. Bengio. Generating sentences from a continuous space. In *CoNLL*, 2016.

Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in beta-vae. *arXiv preprint arXiv:1804.03599*, 2018.

Christpher P. Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019.

Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on scientific computing*, 16(5):1190–1208, 1995.

Charles F Cadieu and Bruno A Olshausen. Learning intermediate-level representations of form and motion from natural movies. *Neural Computation*, 24(4):827–866, 2012.

J-F Cardoso. Source separation using higher order moments. In *International Conference on Acoustics, Speech, and Signal Processing,*, pages 2109–2112. IEEE, 1989.

Tian Qi Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, pages 2610–2620, 2018.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180, 2016.

Brian Cheung, Jesse A Livezey, Arjun K Bansal, and Bruno A Olshausen. Discovering hidden factors of variation in deep networks. *arXiv preprint arXiv:1412.6583*, 2014.

Pierre Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994.

Elliot Creager, David Madras, Jörn-Henrik Jacobsen, Marissa A Weis, Kevin Swersky, Toniann Pitassi, and Richard Zemel. Flexibly fair representation learning by disentanglement. In *International Conference on Machine Learning*, page 1436–1445, 2019.

Emily L Denton and Vighnesh Birodkar. Unsupervised learning of disentangled representations from video. In *Advances in Neural Information Processing Systems*, pages 4414–4423, 2017.

Adji B. Dieng, Yoon Kim, Alexander M. Rush, and D. Blei. Avoiding latent variable collapse with generative skip models. *ArXiv*, abs/1807.04863, 2019.

Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *ArXiv*, abs/1410.8516, 2017a.

Laurent Dinh, Jascha Sohl-Dickstein, and S. Bengio. Density estimation using real nvp. *ArXiv*, abs/1605.08803, 2017b.

Sunny Duan, Loic Matthey, Andre Saraiva, Nick Watters, Christopher Burgess, Alexander Lerchner, and Irina Higgins. Unsupervised model selection for variational disentangled representation learning. In *International Conference on Learning Representations (ICLR)*, 2020.

Cian Eastwood and Christopher KI Williams. A framework for the quantitative evaluation of disentangled representations. In *In International Conference on Learning Representations*, 2018.

Peter Földiák. Learning invariance from transformation sequences. *Neural Computation*, 3(2):194–200, 1991.

Lijian Gao, Qirong Mao, Ming Dong, Yu Jing, and Ratna Chinnam. On learning disentangled representation for acoustic event detection. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2006–2014, 2019.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

Muhammad Waleed Gondal, Manuel Wuthrich, Djordje Miladinovic, Francesco Locatello, Martin Breidt, Valentin Volchkov, Joel Akpo, Olivier Bachem, Bernhard Schölkopf, and Stefan Bauer. On the transfer of inductive bias from simulation to the real world: a new disentanglement dataset. In *Advances in Neural Information Processing Systems*, pages 15714–15725, 2019.

Will Grathwohl and Aaron Wilson. Disentangling space and time in video with hierarchical variational auto-encoders. *arXiv preprint arXiv:1612.04440*, 2016.

Klaus Greff, Raphael Lopez Kaufman, Rishabh Kabra, Nick Watters, Chris Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. *arXiv preprint arXiv:1903.00450*, 2019.

Luigi Gresele, Giancarlo Fissore, Adrian Javaloy, Bernhard Scholkopf, and Aapo Hyvarinen. Relative gradient optimization of the jacobian term in unsupervised deep learning. *ArXiv*, abs/2006.15090, 2020.

Wakako Hashimoto. Quadratic forms in natural images. *Network: Computation in Neural Systems*, 14(4): 765–788, 2003.

Junxian He, Daniel Spokoyny, Graham Neubig, and Taylor Berg-Kirkpatrick. Lagging inference networks and posterior collapse in variational autoencoders. *arXiv preprint arXiv:1901.05534*, 2019.

Olivier J Hénaff, Robbe LT Goris, and Eero P Simoncelli. Perceptual straightening of natural videos. *Nature neuroscience*, 22(6):984–991, 2019.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *International Conference on Learning Representations (ICLR)*, 2(5):6, 2017.

Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*, 2018.

Geoffrey E Hinton. Connectionist learning procedures. In *Machine learning*, page 208. Elsevier, 1990.

Aapo Hyvärinen and Patrik Hoyer. Emergence of phase-and shift-invariant features by decomposition of natural images into independent feature subspaces. *Neural computation*, 12(7):1705–1720, 2000.

Aapo Hyvärinen and Hiroshi Morioka. Unsupervised feature extraction by time-contrastive learning and nonlinear ica. In *Advances in Neural Information Processing Systems*, pages 3765–3773, 2016.

Aapo Hyvärinen and Hiroshi Morioka. Nonlinear ica of temporally dependent stationary sources. In *Proceedings of Machine Learning Research*, 2017.

Aapo Hyvärinen and Petteri Pajunen. Nonlinear independent component analysis: Existence and uniqueness results. *Neural Networks*, 12(3):429–439, 1999.

Aapo Hyvärinen, Jarmo Hurri, and Jaakko Väyrynen. Bubbles: a unifying framework for low-level statistical properties of natural image sequences. *JOSA A*, 20(7):1237–1252, 2003.

Aapo Hyvärinen, Hiroaki Sasaki, and Richard E Turner. Nonlinear ica using auxiliary variables and generalized contrastive learning. *arXiv preprint arXiv:1805.08651*, 2018.

Christian Jutten and Jeanny Herault. Blind separation of sources, part i: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24(1):1–10, 1991.

Ilyes Khemakhem, Diederik P Kingma, and Aapo Hyvärinen. Variational autoencoders and nonlinear ica: A unifying framework. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020a.

Ilyes Khemakhem, Ricardo Monti, Diederik Kingma, and Aapo Hyvarinen. Ice-beem: Identifiable conditional energy-based deep models based on nonlinear ica. *Advances in Neural Information Processing Systems*, 33, 2020b.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*, 2018.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in neural information processing systems*, pages 10215–10224, 2018.

Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in neural information processing systems*, pages 4743–4751, 2016.

Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum. Deep convolutional inverse graphics network. In *Advances in neural information processing systems*, pages 2539–2547, 2015.

Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. In *International Conference on Learning Representations*, 2018.

Yann LeCun, Fu Jie Huang, and Leon Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pages II–104. IEEE, 2004.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. *arXiv preprint arXiv:1811.12359*, 2018.

Francesco Locatello, Gabriele Abbati, Thomas Rainforth, Stefan Bauer, Bernhard Schölkopf, and Olivier Bachem. On the fairness of disentangled representations. In *Advances in Neural Information Processing Systems*, pages 14611–14624, 2019.

Francesco Locatello, Ben Poole, Gunnar Rätsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen. Weakly-supervised disentanglement without compromises. *arXiv preprint arXiv:2002.02886*, 2020.

James Lucas, George Tucker, Roger B Grosse, and Mohammad Norouzi. Don't blame the elbo! a linear vae perspective on posterior collapse. In *Advances in Neural Information Processing Systems*, pages 9408–9418, 2019.

Liqian Ma, Qianru Sun, Stamatios Georgoulis, Luc Van Gool, Bernt Schiele, and Mario Fritz. Disentangled person image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 99–108, 2018.

Lars Maaløe, Marco Fraccaro, Valentin Liévin, and Ole Winther. Biva: A very deep hierarchy of latent variables for generative modeling. In *Advances in neural information processing systems*, pages 6551–6562, 2019.

Emile Mathieu, Tom Rainforth, N Siddharth, and Yee Whye Teh. Disentangling disentanglement in variational autoencoders. In *Proceedings of the 36th International Conference on Machine Learning*, pages 4402–4412, 2019.

Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

Stanisław Mazur and Stanisław Ulam. Sur les transformations isométriques d'espaces vectoriels normés. *CR Acad. Sci. Paris*, 194(946-948):116, 1932.

Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. MOT16: A benchmark for multi-object tracking. *arXiv:1603.00831 [cs]*, March 2016.

Graeme Mitchison. Removing time variation with the anti-hebbian differential synapse. *Neural Computation*, 3 (3):312–320, 1991.

Hossein Mobahi, Ronan Collobert, and Jason Weston. Deep learning from temporal coherence in video. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 737–744, 2009.

Hiroshi Morioka. Time-contrastive learning (tcl), 2018. URL https://github.com/hirosm/TCL.

Bruno A Olshausen. Learning sparse, overcomplete representations of time-varying natural images. In *Proceedings 2003 International Conference on Image Processing (Cat. No. 03CH37429)*, volume 1, pages I–41. IEEE, 2003.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Edouard Pineau, S. Razakarivony, and T. Bonald. Time series source separation with slow flows. *ArXiv*, abs/2007.10182, 2020.

Scott E Reed, Yi Zhang, Yuting Zhang, and Honglak Lee. Deep visual analogy-making. In *Advances in neural information processing systems*, pages 1252–1260, 2015.

Karl Ridgeway. A survey of inductive biases for factorial representation-learning. *arXiv preprint arXiv:1612.05299*, 2016.

Karl Ridgeway and Michael C Mozer. Learning deep disentangled embeddings with the f-statistic loss. In *Advances in Neural Information Processing Systems*, pages 185–194, 2018.

Michal Rolinek, Dominik Zietlow, and Georg Martius. Variational autoencoders pursue pca directions (by accident). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12406–12415, 2019.

Rui Shu, Yining Chen, Abhishek Kumar, Stefano Ermon, and Ben Poole. Weakly supervised disentanglement with guarantees. *arXiv preprint arXiv:1910.09772*, 2019.

Eero P Simoncelli and Bruno A Olshausen. Natural image statistics and neural representation. *Annual review of neuroscience*, 24(1):1193–1216, 2001.

Fabian Sinz, Sebastian Gerwinn, and Matthias Bethge. Characterization of the p-generalized normal distribution. *Journal of Multivariate Analysis*, 100(5):817–820, 2009.

Jascha Sohl-Dickstein, Ching Ming Wang, and Bruno A Olshausen. An unsupervised algorithm for learning lie group transformations. *arXiv preprint arXiv:1001.1027*, 2010.

Peter Sorrenson, Carsten Rother, and Ulrich Kothe. Disentanglement by nonlinear ica with general incompressible-flow networks (gin). *ArXiv*, abs/2001.04872, 2017.

Mikhail Fedorovich Subbotin. On the law of frequency of error. *Mat. Sb.*, 31(2):296–301, 1923.

Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3(1):9–44, 1988.

Attila Szabó, Qiyang Hu, Tiziano Portenier, Matthias Zwicker, and Paolo Favaro. Challenges in disentangling independent factors of variation. *arXiv preprint arXiv:1711.02245*, 2017.

Esteban G Tabak and Cristina V Turner. A family of nonparametric density estimation algorithms. *Communications on Pure and Applied Mathematics*, 66(2):145–164, 2013.

Esteban G Tabak, Eric Vanden-Eijnden, et al. Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences*, 8(1):217–233, 2010.

Loek Tonnaer, Luis A. Pérez Rey, Vlado Menkovski, Mike Holenderski, and Jacobus W. Portegies. Quantifying and learning disentangled representations with limited supervision, 2020.

Frederik Träuble, Elliot Creager, Niki Kilbertus, Anirudh Goyal, Francesco Locatello, Bernhard Schölkopf, and Stefan Bauer. Is independence all you need? on the generalization of representations learned from correlated data. *arXiv preprint arXiv:2006.07886*, 2020.

Michael Tschannen, Josip Djolonga, Marvin Ritter, Aravindh Mahendran, Neil Houlsby, Sylvain Gelly, and Mario Lucic. Self-supervised learning of video-induced visual invariances. *arXiv preprint arXiv:1912.02783*, 2019.

Richard Turner and Maneesh Sahani. A maximum-likelihood interpretation for slow feature analysis. *Neural computation*, 19(4):1022–1038, 2007.

Rishi Veerapaneni, John D. Co-Reyes, Michael Chang, Michael Janner, Chelsea Finn, Jiajun Wu, Joshua B. Tenenbaum, and Sergey Levine. Entity abstraction in visual model-based reinforcement learning. *arXiv preprint arXiv:1910.12827*, 2019.

Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. Mots: Multi-object tracking and segmentation. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Nicholas Watters, Loic Matthey, Sebastian Borgeaud, Rishabh Kabra, and Alexander Lerchner. Spriteworld: A flexible, configurable reinforcement learning environment, 2019. URL `https://github.com/deepmind/spriteworld/`.

Marissa A. Weis, Kashyap Chitta, Yash Sharma, Wieland Brendel, Matthias Bethge, Andreas Geiger, and Alexander S. Ecker. Unmasking the inductive biases of unsupervised object representations for video sequences. *arXiv preprint arXiv:2006.07034*, 2020.

Laurenz Wiskott and Terrence J Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural Computation*, 14(4):715–770, 2002.

Markus Wulfmeier, Arunkumar Byravan, Tim Hertweck, Irina Higgins, Ankush Gupta, Tejas Kulkarni, Malcolm Reynolds, Denis Teplyashin, Roland Hafner, Thomas Lampe, and Martin Riedmiller. Representation matters: Improving perception and exploration for robotics. *arXiv preprint arXiv:2011.01758*, 2020.

Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018.

Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. *arXiv preprint arXiv:1905.04804*, 2019.

Mengyue Yang, Furui Liu, Zhitang Chen, Xinwei Shen, Jianye Hao, and Jun Wang. Causalvae: Structured causal disentanglement in variational autoencoder. *arXiv preprint arXiv:2004.08697*, 2020.

Ilker Yildirim, Mario Belledonne, Winrich Freiwald, and Josh Tenenbaum. Efficient inverse graphics in biological face processing. *Science Advances*, 6(10):eaax5979, 2020.

Will Zou, Shenghuo Zhu, Kai Yu, and Andrew Y Ng. Deep learning of invariant features via simulated fixations in video. In *Advances in neural information processing systems*, pages 3203–3211, 2012.

APPENDIX

A FORMAL METHODS

Function / variable	Description
g	Generator
α	Prior shape
λ	Prior rate
$p(\mathbf{z})$	Prior
$\mathbf{z} \sim p(\mathbf{z})$	Latent variables
$\mathbf{x} = g(\mathbf{z})$	Generated images
$q(\mathbf{z}\|\mathbf{x})$	Variational posterior

Table 3: Glossary of terms. We use a * (i.e. g^*) when necessary to highlight that we are referring to the ground truth model.

A.1 PROOF OF IDENTIFIABILITY

To study disentanglement, we assume that the generative factors $\mathbf{z} \in \mathbb{R}^D$ are mapped to images $\mathbf{x} \in \mathbb{R}^N$ (usually $D \ll N$, but see section B) by a nonlinear ground-truth generator $g^* : \mathbf{z} \mapsto \mathbf{x}$.

Theorem 1 *Let $(g^*, \lambda^*, \alpha^*)$ and (g, λ, α) respectively be ground-truth and learned generative models as defined in Eq.* (2)*. If the following conditions are satisfied:*

(i) The generators g^* and g are defined everywhere in the latent space. Moreover, they are injective and differentiable almost everywhere,

(ii) There is no model misspecification i.e. $\alpha = \alpha^*$ and $\lambda = \lambda^*$, so $\mathbf{z} \sim p(\mathbf{z}) = p^*(\mathbf{z})$,

(iii) Pairs of images are generated as $(\mathbf{x}^*_{t-1}, \mathbf{x}^*_t) = (g^*(\mathbf{z}_{t-1}), g^*(\mathbf{z}_t))$ and $(\mathbf{x}_{t-1}, \mathbf{x}_t) = (g(\mathbf{z}_{t-1}), g(\mathbf{z}_t))$,

(iv) The distributions of $(\mathbf{x}^*_{t-1}, \mathbf{x}^*_t)$ and $(\mathbf{x}_{t-1}, \mathbf{x}_t)$ are the same (i.e. the corresponding densities are equal almost everywhere: $p^*(\mathbf{x}_{t-1}, \mathbf{x}_t) = p(\mathbf{x}_{t-1}, \mathbf{x}_t)$,

then $g = g^* \circ \sigma$, where σ is a composition of a permutation and sign flips.

Proof. Since $\mathbf{x} = g(\mathbf{z})$ can be written as $\mathbf{x} = (g^* \circ (g^*)^{-1} \circ g)(\mathbf{z})$, we can assume that $g = g^* \circ h$ for some function h on the latent space.

We first show that the function h is a bijection on the latent space. It is injective, since both g and g^* are injective. Because of continuity of h, if it were not surjective, there would be some neighborhood $\mathbf{U}_{\tilde{\mathbf{z}}}$ of $\tilde{\mathbf{z}}$ that would not have a pre-image under h. This would mean that images generated by g^* from $\mathbf{U}_{\tilde{\mathbf{z}}}$ would have zero density under the distribution of images generated by g (i.e. $p(g^*(\mathbf{U}_{\tilde{\mathbf{z}}})) = 0$). This density would be non-zero under the distribution of images directly generated by the ground-truth generator g^* (i.e. $p^*(g^*(\mathbf{U}_{\tilde{\mathbf{z}}})) \neq 0$), which contradicts the assumption that these distributions are equal. It follows that h is bijective.

In the next step, we show that the distribution of latent space pairs $(h(\mathbf{z}_{t-1}), h(\mathbf{z}_t))$ matches the latent space prior distribution (i.e. h preserves the prior distribution in the latent space). Indeed, using the assumption that the distributions of $(g^*(\mathbf{z}_{t-1}), g^*(\mathbf{z}_t))$ and $((g^* \circ h)(\mathbf{z}_{t-1}), (g^* \circ h)(\mathbf{z}_t))$ are the same, we can write the following equality using the change of variables formula:

$$\begin{aligned} p^*(\mathbf{x}_{t-1}, \mathbf{x}_t) &= p((g^*)^{-1}(\mathbf{x}_{t-1}), (g^*)^{-1}(\mathbf{x}_t)) \left| \det \left(\frac{\mathrm{d}(g^*)^{-1}}{\mathrm{d}(\mathbf{x}_{t-1}, \mathbf{x}_t)} \right) \right| \\ &= p_h((g^*)^{-1}(\mathbf{x}_{t-1}), (g^*)^{-1}(\mathbf{x}_t)) \left| \det \left(\frac{\mathrm{d}(g^*)^{-1}}{\mathrm{d}(\mathbf{x}_{t-1}, \mathbf{x}_t)} \right) \right| \\ &= p(\mathbf{x}_{t-1}, \mathbf{x}_t), \end{aligned} \tag{5}$$

where p and p_h are densities of $(\mathbf{z}_{t-1}, \mathbf{z}_t)$ and $(h(\mathbf{z}_{t-1}), h(\mathbf{z}_t))$. Since the determinants above cancel, these densities are equal at the pre-image of any pair of images $(\mathbf{x}_{t-1}, \mathbf{x}_t)$. Because g^* is defined

everywhere in the latent space, p and p_h are equal for any pair of latent space points. Applying the change of variables formula again, we obtain the following equation:

$$\begin{aligned} p(\mathbf{z}_{t-1}, \mathbf{z}_t) &= p(h^{-1}(\mathbf{z}_{t-1}), h^{-1}(\mathbf{z}_t)) \left|\det\left(\frac{\mathrm{d}h^{-1}}{\mathrm{d}(\mathbf{z}_{t-1}, \mathbf{z}_t)}\right)\right| \\ &= p(h^{-1}(\mathbf{z}_{t-1}))\, p(h^{-1}(\mathbf{z}_t) \mid h^{-1}(\mathbf{z}_{t-1})) \left|\det\left(\frac{\mathrm{d}h^{-1}(\mathbf{z}_{t-1})}{\mathrm{d}\mathbf{z}_{t-1}}\right)\right| \left|\det\left(\frac{\mathrm{d}h^{-1}(\mathbf{z}_t)}{\mathrm{d}\mathbf{z}_t}\right)\right| \\ &= p(\mathbf{z}_{t-1})\, p(\mathbf{z}_t \mid \mathbf{z}_{t-1}). \end{aligned} \tag{6}$$

Note that the probability measure p is the same before and after the change of variables, since we showed that the prior distribution in the latent space must be invariant under the function h. The same condition for the marginal $p(\mathbf{z}_{t-1})$ is as follows:

$$p(\mathbf{z}_{t-1}) = p(h^{-1}(\mathbf{z}_{t-1})) \left|\det\left(\frac{\mathrm{d}h^{-1}(\mathbf{z}_{t-1})}{\mathrm{d}\mathbf{z}_{t-1}}\right)\right|. \tag{7}$$

Solving for the determinant of the Jacobian in (7) and plugging it into (6), we obtain

$$p(\mathbf{z}_t \mid \mathbf{z}_{t-1}) = p(h^{-1}(\mathbf{z}_t) \mid h^{-1}(\mathbf{z}_{t-1})) \frac{p(\mathbf{z}_t)}{p(h^{-1}(\mathbf{z}_t))}. \tag{8}$$

Taking logs of both sides, we arrive at the following equation:

$$A(||\mathbf{z}_t - \mathbf{z}_{t-1}||_\alpha^\alpha - ||h^{-1}(\mathbf{z}_t) - h^{-1}(\mathbf{z}_{t-1})||_\alpha^\alpha) = B(||\mathbf{z}_t||_2^2 - ||h^{-1}(\mathbf{z}_t)||_2^2), \tag{9}$$

where A and B are the constants appearing in the exponentials in $p(\mathbf{z}_{t-1})$ and $p(\mathbf{z}_t \mid \mathbf{z}_{t-1})$. The logs of normalization constants cancel out.

For any $\mathbf{z}_t$ we can choose $\mathbf{z}_{t-1} = \mathbf{z}_t$ making the left hand side in (9) equal to zero. This implies that $||\mathbf{z}_t||_2^2 = ||h^{-1}(\mathbf{z}_t)||_2^2$ for any $\mathbf{z}_t$, i.e. function h^{-1} preserves the 2-norm. Moreover, the preservation of the 2-norm implies that $p(\mathbf{z}_{t-1}) = p(h^{-1}(\mathbf{z}_{t-1}))$ and therefore it follows from (7) that for any $\mathbf{z}$

$$\left|\det\left(\frac{\mathrm{d}h^{-1}(\mathbf{z})}{\mathrm{d}\mathbf{z}}\right)\right| = 1. \tag{10}$$

Thus, the left hand side of (9) can be re-written as

$$||\mathbf{z}_t - \mathbf{z}_{t-1}||_\alpha^\alpha - ||h^{-1}(\mathbf{z}_t) - h^{-1}(\mathbf{z}_{t-1})||_\alpha^\alpha = 0. \tag{11}$$

This means that h^{-1} preserves the α-distances between points. Moreover, because h is bijective, the Mazur-Ulam theorem (Mazur and Ulam, 1932) tells us that h must be an affine transform.

In the next step, to prove that h must be a permutation and sign flip, let us choose an arbitrary point $\mathbf{z}_{t-1}$ and $\mathbf{z}_t = \mathbf{z}_{t-1} + \varepsilon\, \mathbf{e}_k = (z_{1,1}, \ldots, z_{1,k} + \varepsilon, \ldots, z_{1,D})$. Using (11) and performing a Taylor expansion around $\mathbf{z}_{t-1}$, we obtain the following:

$$\begin{aligned} \varepsilon^\alpha &= ||\mathbf{z}_t - \mathbf{z}_{t-1}||_\alpha^\alpha \\ &= ||h^{-1}(\mathbf{z}_{t-1} + \varepsilon\, \mathbf{e}_k) - h^{-1}(\mathbf{z}_{t-1})||_\alpha^\alpha \\ &= \left|\left|\varepsilon \cdot \left(\frac{\partial h_1^{-1}(\mathbf{z}_{t-1})}{\partial z_{t-1,k}}, \ldots, \frac{\partial h_D^{-1}(\mathbf{z}_{t-1})}{\partial z_{t-1,k}}\right) + O(\varepsilon^2)\right|\right|_\alpha^\alpha. \end{aligned} \tag{12}$$

The higher-order terms $O(\varepsilon^2)$ are zero since h is affine, therefore dividing both sides of the above equation by ε^α we find that

$$\left|\left|\left(\frac{\partial h_1^{-1}(\mathbf{z}_{t-1})}{\partial z_{t-1,k}}, \ldots, \frac{\partial h_D^{-1}(\mathbf{z}_{t-1})}{\partial z_{t-1,k}}\right)\right|\right|_\alpha^\alpha = 1. \tag{13}$$

The vectors of k-th partial derivatives of components of h^{-1} are columns of the Jacobian matrix $\left(\frac{\mathrm{d}h^{-1}(\mathbf{z})}{\mathrm{d}\mathbf{z}}\right)$. Using the fact that the determinant of that matrix is equal to one and applying Hadamard's inequality, we obtain that

$$\left|\det\left(\frac{\mathrm{d}h^{-1}(\mathbf{z})}{\mathrm{d}\mathbf{z}}\right)\right| = 1 \leq \prod_{k=1}^{D} \left|\left|\left(\frac{\partial h_1^{-1}(\mathbf{z}_{t-1})}{\partial z_{t-1,k}}, \ldots, \frac{\partial h_D^{-1}(\mathbf{z}_{t-1})}{\partial z_{t-1,k}}\right)\right|\right|_2. \tag{14}$$

Since $\alpha < 2$, for any vector $\mathbf{v}$ it holds that $||\mathbf{v}||_2 \leq ||\mathbf{v}||_\alpha$, with equality only if at most one component of $\mathbf{v}$ is non-zero. This inequality implies that both (13) and (14) hold at the same time if and only if

$$\left|\left|\left(\frac{\partial h_1^{-1}(\mathbf{z}_{t-1})}{\partial z_{t-1,k}}, \ldots, \frac{\partial h_D^{-1}(\mathbf{z}_{t-1})}{\partial z_{t-1,k}}\right)\right|\right|_2 = \left|\left|\left(\frac{\partial h_1^{-1}(\mathbf{z}_{t-1})}{\partial z_{t-1,k}}, \ldots, \frac{\partial h_D^{-1}(\mathbf{z}_{t-1})}{\partial z_{t-1,k}}\right)\right|\right|_\alpha = 1, \quad (15)$$

meaning that only one element of these vectors of k-th partial derivatives is non-zero, and it is equal to 1 or -1. Thus, the function h is a composition of a permutation and sign flips at every point. Potentially, this permutation might be input-dependent, but we argued above that h is affine, therefore the permutation must be the same for all points. □

A.2 Kullback Leibler Divergence of Slow Variational Autoencoder

The VAE learns a variational approximation to the true posterior by maximizing a lower bound on the log-likelihood of the empirical data distribution $\mathcal{D}$

$$\begin{aligned} &E_{\mathbf{x}_{t-1},\mathbf{x}_t\sim\mathcal{D}}[\log p(\mathbf{x}_{t-1},\mathbf{x}_t)] \geq \\ &\quad E_{\mathbf{x}_{t-1},\mathbf{x}_t\sim\mathcal{D}}[E_{q(\mathbf{z}_t,\mathbf{z}_{t-1}|\mathbf{x}_t,\mathbf{x}_{t-1})}[\log p(\mathbf{x}_{t-1},\mathbf{x}_t,\mathbf{z}_{t-1},\mathbf{z}_t) - \log q(\mathbf{z}_t,\mathbf{z}_{t-1}|\mathbf{x}_t,\mathbf{x}_{t-1})]]. \end{aligned} \quad (16)$$

For this, we need to compute the Kullback-Leibler divergence (KL) between the posterior $q(\mathbf{z}_t,\mathbf{z}_{t-1}|\mathbf{x}_t,\mathbf{x}_{t-1})$ and the prior $p(\mathbf{z}_t,\mathbf{z}_{t-1})$. Since all of these distributions are per design factorial, we will, for simplicity, derive the KL below for scalar variables (log-probabilities will simply have to be summed to obtain the full expression). Recall that the model prior and posterior factorize like

$$\begin{aligned} p(z_t,z_{t-1}) &= p(z_t|z_{t-1})\,p(z_{t-1}) \\ q(z_t,z_{t-1}|\mathbf{x}_t,\mathbf{x}_{t-1}) &= q(z_t|\mathbf{x}_t)\,q(z_{t-1}|\mathbf{x}_{t-1}). \end{aligned} \quad (17)$$

Then, given a pair of inputs $(\mathbf{x}_{t-1},\mathbf{x}_t)$, the KL can be written

$$\begin{aligned} &D_{KL}(q(z_t,z_{t-1}|\mathbf{x}_t,\mathbf{x}_{t-1})|p(z_t,z_{t-1})) = E_{z_t,z_{t-1}\sim q(z_t,z_{t-1}|\mathbf{x}_t,\mathbf{x}_{t-1})}\left[\log\frac{q(z_t|\mathbf{x}_t)\,q(z_{t-1}|\mathbf{x}_{t-1})}{p(z_t|z_{t-1})\,p(z_{t-1})}\right] \\ &= E_{z_{t-1}\sim q(z_{t-1}|\mathbf{x}_{t-1})}\left[\log\frac{q(z_{t-1}|\mathbf{x}_{t-1})}{p(z_{t-1})}\right] + E_{z_t,z_{t-1}\sim q(z_t,z_{t-1}|\mathbf{x}_t,\mathbf{x}_{t-1})}\left[\log\frac{q(z_t|\mathbf{x}_t)}{p(z_t|z_{t-1})}\right] \\ &= D_{KL}(q(z_{t-1}|\mathbf{x}_{t-1})|p(z_{t-1})) - H(q(z_t|\mathbf{x}_t)) + E_{z_{t-1}\sim q(z_{t-1}|\mathbf{x}_{t-1})}\left[H(q(z_t|\mathbf{x}_t),p(z_t|z_{t-1}))\right] \end{aligned} \quad (18)$$

Where we use the fact that KL divergences decompose like $D_{KL}(X,Y) = H(X,Y) - H(X)$ into (differential) cross-entropy $H(X,Y)$ and entropy $H(X)$. The first term of the last line in (18) is the same KL divergence as in the standard VAE, namely between a Gaussian distribution $q(z_{t-1}|\mathbf{x}_{t-1})$ with some $\mu(\mathbf{x}_{t-1})$ and $\sigma(\mathbf{x}_{t-1})$ and a standard Normal distribution $p(z_{t-1})$. The solution of the KL is given by $D_{KL}(q(z_{t-1}|\mathbf{x}_{t-1})|q(z_{t-1})) = -\log\sigma(\mathbf{x}_{t-1}) + \frac{1}{2}(\mu(\mathbf{x}_{t-1})^2 + \sigma(\mathbf{x}_{t-1})^2 - 1)$ (Bishop, 2006). The second term on the RHS, i.e. the entropy of a Gaussian is simply given by $H(q(z_t|\mathbf{x}_t)) = \log(\sigma(\mathbf{x}_t)\sqrt{2\pi e})$.

To compute the last term on the RHS, let us recall the Laplace form of the conditional prior

$$p(z_t|z_{t-1}) = \frac{\lambda}{2}\exp -\lambda|z_t - z_{t-1}|. \quad (19)$$

Thus the cross-entropy becomes

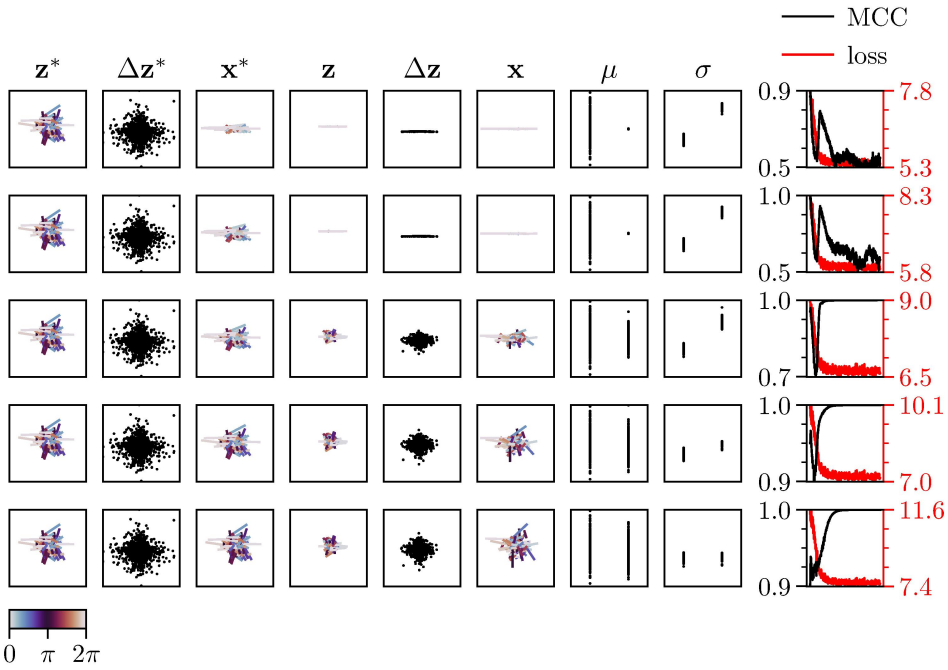
$$\begin{aligned} H(q(z_t|\mathbf{x}_t),p(z_t|z_{t-1})) &= -E_{z_t\sim q(z_t|\mathbf{x}_t)}[\log p(z_t|z_{t-1})] \\ &= -\log\left(\frac{\lambda}{2}\right) + \lambda E_{z_t\sim q(z_t|\mathbf{x}_t)}[|z_t - z_{t-1}|]. \end{aligned} \quad (20)$$

Now, if some random variable $X \sim \mathcal{N}(\mu,\, \sigma^2)$, then $Y = |X|$ follows a *folded normal distribution*, for which the mean is defined as

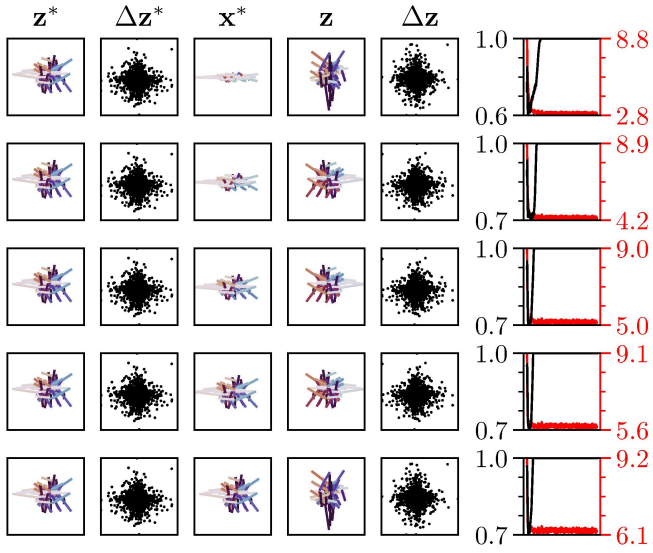
$$E[|x|] = \sigma\sqrt{\frac{2}{\pi}}\exp\left(-\frac{\mu^2}{2\sigma^2}\right) - \mu\left(1 - 2\,\Phi\left(\frac{\mu}{\sigma}\right)\right), \tag{21}$$

where Φ is the cumulative distribution function of a standard normal distribution (mean zero and variance one). Thus, denoting $\mu(\mathbf{x}_t)$ and $\sigma(\mathbf{x}_t)$ the mean and variance of $q(z_t|\mathbf{x}_t)$, and defining $\mu(\mathbf{x}_t, z_{t-1}) = \mu(\mathbf{x}_t) - z_{t-1}$, we can rewrite further

$$H(q(z_t|\mathbf{x}_t), p(z_t|z_{t-1})) = \\ -\log\left(\frac{\lambda}{2}\right) + \lambda\left(\sigma(\mathbf{x}_t)\sqrt{\frac{2}{\pi}}\exp\left(-\frac{\mu(\mathbf{x}_t, z_{t-1})^2}{2\sigma(\mathbf{x}_t)^2}\right) - \mu(\mathbf{x}_t, z_{t-1})\left(1 - 2\,\Phi\left(\frac{\mu(\mathbf{x}_t, z_{t-1})}{\sigma(\mathbf{x}_t)}\right)\right)\right). \tag{22}$$



(a) SlowVAE performance.



(b) SlowFlow performance.

Figure 6: **VAE failure modes.** Rows respectively indicate $\kappa = 0.2, 0.4, 0.6, 0.8, 1.0$ from Eq. (24). The left five columns show values for 100 randomly chosen examples, while the μ and σ columns show values for the full training set. Columns in the sets ($\mathbf{z}^*$, $\mathbf{z}$), ($\Delta\mathbf{z}^*$, $\Delta\mathbf{z}$), ($\mathbf{x}^*$, $\mathbf{x}$) all have the same (arbitrary) scale factors the axes. Lines indicate trajectories from time-point t to $t+1$, and color indicates the angle of the trajectory vector with respect to the canonical variable axes. The μ axes is scaled from -4 to 4, and σ axes are scaled from 0 to 1, where individual dots represent latent encoding values from test images. The rightmost plots show a shift in the relationship between the mean correlation coefficient (MCC) (black, higher is better) and training loss (red, lower is better) as one increases κ.

B CHOOSING A LATENT VARIABLE MODEL

Our proposed method for disentanglement can be implemented in conjunction with different probabilistic latent variable models. In this section, we compare VAEs and normalizing flows as possible candidates.

Variational Autoencoders (VAEs) (Kingma and Welling, 2013) are a widely used probabilistic latent variable model. Despite their simple structure and empirical success, VAEs can converge to a pathological solution called *posterior collapse* (Lucas et al., 2019; Bowman et al., 2016; He et al., 2019). This solution results in the encoder's variational posterior approximation matching the prior, which is typically chosen to be a multivariate standard normal $q(\mathbf{z}|\mathbf{x}) \approx p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$. This disconnects the encoder from the decoder, making them approximately independent, i.e. $p(\mathbf{x}|\mathbf{z}) \approx p(\mathbf{x})$. The failure mode is often observed when the decoder architecture is overly expressive, i.e. with autoregressive models, or when the likelihood $p(\mathbf{x})$ is easy to estimate. Approaches that alleviate this problem rely on modifying the ELBO training objective (Bowman et al., 2016; Kingma et al., 2016) or restricting the decoder structure (Dieng et al., 2019; Maaløe et al., 2019). However, these approaches come with various drawbacks, including optimization issues (Lucas et al., 2019).

Another approach to estimate latent variables are normalizing flows which describe a sequence of invertible mappings by iteratively applying the change of variables rule (Dinh et al., 2017b). Unlike VAEs, flow based latent variable models allow for a direct optimization of the likelihood (Dinh et al., 2017b). Most normalizing flow models rely on a fast and reliable calculation of the determinant of the Jacobian of the outputs with respect to the inputs, which constrains the architectural design and limits the capacity of the network (Tabak et al., 2010; Tabak and Turner, 2013; Dinh et al., 2017b). Thus, competitive flows require very deep architectures in practice (Kingma and Dhariwal, 2018). Furthermore, flows are not directly suited for a scenario where the observation space is higher dimensional than the generating latent factors, $\dim(\mathbf{z}) < \dim(\mathbf{x})$, as the computation of the determinant requires a square Jacobian matrix. We tried setting $\dim(\mathbf{z}) = \dim(\mathbf{x}) > \dim(\mathbf{z}^*)$, but observed instability while optimizing the objective defined below.

It is straightforward to derive a flow-based objective based on the assumptions in Eq. (2). We consider a normalizing flow with with K blocks $f(\mathbf{x}) = f_K \circ ... \circ f_1 : \mathbf{x} \mapsto \mathbf{z}$. The coupling blocks can refer to nonlinear mixing similar to Kingma and Dhariwal (2018), or in the linear case ($K = 1$) to an invertible de-mixing matrix. This leads to the following estimation of the likelihood

$$p(\mathbf{x}_{t-1}, \mathbf{x}_t) = \; p(f(\mathbf{x}_{t-1}))\; p(f(\mathbf{x}_t)|f(\mathbf{x}_{t-1}))\; \prod_{k=1}^{K} \left|\det \frac{\partial f_k}{\partial \mathbf{z}_{k-1,t-1}}\right|^{-1} \prod_{k=1}^{K} \left|\det \frac{\partial f_k}{\partial \mathbf{z}_{k-1,t}}\right|^{-1}. \tag{23}$$

Note that $p(f(\mathbf{x}_{t-1}))$ is Gaussian and $p(f(\mathbf{x}_t)|f(\mathbf{x}_{t-1}))$ is a Laplacian, similar to Eq. (2). During optimization we take the $-\log$ of both sides and minimize w.r.t. the parameters of f. We refer to this estimator as *SlowFlow*. Our SlowFlow model is very similar to the flow described in (Pineau et al., 2020), who use a Gaussian transition prior and therefore would have weaker identifiability guarantees. Next, we compare SlowFlow and SlowVAE in the context of disentanglement.

To demonstrate the posterior collapse in VAEs, we generate data points $(\mathbf{x}_t, \mathbf{x}_{t-1})$ according to Eq. (2) with a two dimensional latent space $\dim(\mathbf{z}^*) = 2$. We consider a trivial linear mixing of $\mathbf{x}^* = \mathbf{W}^*\mathbf{z}^* = g^*(\mathbf{z}^*)$ with

$$\mathbf{W}^* = \text{diag}(1, \kappa) \tag{24}$$

and $\kappa \in [0.1, 1]$. As can be seen by looking at the σ and μ outputs of the encoder in Fig 6a, for $\kappa < 0.4$, the encoder for the minor axis collapses to the prior. The decoder then tries to minimize the reconstruction loss by solely covering the first principal component of the data, which is also described in Rolinek et al. (2019). Despite the collapse and decrease in MCC, the SlowVAE loss from Eq. (4) still improves during training. On the other hand, a simple linear SlowFlow model $f(\mathbf{x}) = \mathbf{W}\mathbf{x}$, which directly optimizes the likelihood, recovers the latents consistently as seen by the MCC measure (Fig 6b).

To show the strength of the VAE model we increase the complexity of the data-distribution by using a non-linear expanding decoder such that $\dim(\mathbf{x}) \gg \dim(\mathbf{z}^*)$. In Fig. 7 we observe that increasing the input dimensionality is sufficient for SlowVAE to find the corresponding latents and achieve high MCC with low loss.

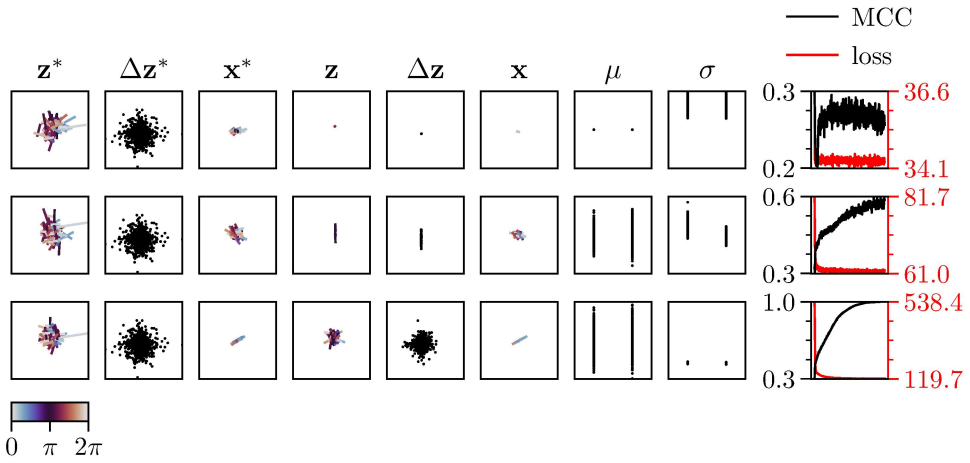


Figure 7: **VAEs perform better when data dimensionality exceeds the latent dimensionality.** VAEs prefer data dimensions to be greater than latent dimensions. Individual subplots are as described in Fig. 6. For all data in this experiment we used a 20-dimensional latent space, $\dim(\mathbf{z}^*) = 20$. Each row corresponds to the dimensionality of the $\mathbf{x}^*$, with values of 20, 200, and 2000. The first two dimensions of $\mathbf{z}^*$ are plotted as well as the two dimensions of $\mathbf{z}$ with the highest corresponding mean correlation coefficient (MCC). The $\mathbf{x}^*$ and $\mathbf{x}$ data are projected onto their first two principal component axes before plotting. A two-layer mixing matrix was used to transform data from Z_{gt} to X_{gt}. As one increases the data dimensionality, the SlowVAE network performs increasingly better in terms of MCC, although worse in terms of total training loss.

Each estimation method is practically useful in different experimental settings. In the case when the mixing operation is trivially defined (Eq. (24), or when the number of dimensions in $\mathbf{z}^*$ match those in $\mathbf{x}^*$), the VAE estimator tends to learn a pathological solution. On the other hand, the normalizing flow estimator does not scale well to high dimensional data due to the requirement of computing the network Jacobian. Additionally, the framework for constructing normalizing flow estimators assumes the latent dimensionality is equal to the data dimensionality to allow for an invertible transform. Together these results lead us to choose an estimator based on the nature of the problem. For our contributed datasets and the DisLib experiments we adopt the VAE framework. However, if one aims to perform simplified experiments such as those typically conducted in the nonlinear ICA literature, it will often make practical sense to switch to a flow-based estimator.

C Disentanglement Metrics

Several recent studies have brought to light shortcomings in a number of proposed disentanglement metrics (Kim and Mnih, 2018; Eastwood and Williams, 2018; Chen et al., 2018; Higgins et al., 2018; Mathieu et al., 2019), many of which have been compiled in the DisLib benchmark. In addition to the concerns they raise, it is important to note that none of the supervised metrics implemented in DisLib allow for continuous ground-truth factors, which is necessary for evaluating with the Natural Sprites and KITTI Masks datasets, as factors such as position and scale are effectively continuous in reality. To rectify this issue without introducing novel metrics, we include the Mean Correlation Coefficient (MCC) in our evaluations, using the implementation of Hyvärinen and Morioka (2016), which is described below.

We measure all metrics presented below between $10,000$ samples of latent factors $\mathbf{z}$ and the corresponding encoded means of our model $\mu(g^*(\mathbf{z}))$. We increase this sample size to $100,000$ for Modularity and MIG to stabilize the entropy estimates.

C.1 MEAN CORRELATION COEFFICIENT

In addition to the DisLib metrics, we also compute the Mean Correlation Coefficient (MCC) in order to perform quantitative evaluation with continuous variables. Because of Theorem 1, perfect disentanglement in the noiseless case should always lead to a correlation coefficient of 1 or -1, although note that we report 100 times the absolute value of the correlation coefficient. In our experiments, MCC is used without modification from the authors' open-sourced code (Morioka, 2018). The method first measures correlation between the ground-truth factors and the encoded latent variables. The initial correlation matrix is then used to match each latent unit with a preferred ground-truth factor. This is an assignment problem that can be solved in polynomial time via the Munkres algorithm, as described in the code release from Morioka (2018). After solving the assignment problem, the correlation coefficients are computed again for the vector of ground-truth factors and the resulting permuted vector of latent encodings, where the output is a matrix of correlation coefficients with D columns for each ground-truth factor and D' rows for each latent variable. We use the (absolute value of the) Spearman coefficient as our correlation measure which assumes a monotonic relationship between the ground-truth factors and latent encodings but tolerates deviations from a strictly linear correspondence.

In the existing implementation for MCC, the ground truth factors, latent encodings, and mixed signal inputs are assumed to have the same dimensionality, i.e. $D = D' = N$. However, in our case, the ground-truth generating factors are much lower dimensional than the signal, $N \ll D$, and the latent encoding is higher dimensional than the ground-truth factors $D' > D$ (see Appendix E for details). To resolve this discrepancy, we add $D' - D$ standard Gaussian noise channels to the ground-truth factors. To compute the MCC score, we take the mean of the absolute value of the upper diagonal of the correlation matrix. The upper diagonal is the diagonal of the square matrix of D ground-truth factors by the top D most correlated latent dimensions after sorting. In this way, we obtain an MCC estimate which averages only over the D correlation coefficients of the D ground truth factors with their corresponding best matching latent factors.

C.2 DISLIB METRICS

BetaVAE (Higgins et al., 2017)

The BetaVAE metric uses a biased estimator with tunable hyperparameters, although we follow the convention established in (Locatello et al., 2018) of using the *scikit-learn* defaults. For a sample in a batch, a pair of images, $(\mathbf{x}_1, \mathbf{x}_2)$, is generated by fixing the value of one of the data generative factors while uniformly sampling the rest. The absolute value of the difference between the latent codes produced from the image pairs is then taken, $\mathbf{z}_{\text{diff}} = |\mathbf{z}_1 - \mathbf{z}_2|$. A logistic classifier is fit with batches of $\mathbf{z}_{\text{diff}}$ variables and the corresponding index of the fixed ground-truth factor serves as the label. Once the classifier is trained, the metric itself is the mean classifier accuracy on a batch of held-out test data. The training minimizes the following loss:

$$L = \frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{i=1}^{n} \log(\exp(-\mathbf{y}_i(\mathbf{z}_{\text{diff},i}^T\mathbf{w} + c)) + 1), \tag{25}$$

where $\mathbf{w}$ and c are the learnable weight matrix and bias, respectively, and $\mathbf{y}$ is the index of the fixed ground-truth factor for the batch. The network is trained using the `lbfgs` optimizer (Byrd et al., 1995), which is implemented via the *scikit-learn* Python package (Pedregosa et al., 2011) in the Disentanglement Library (DisLib, Locatello et al., 2018). In the original work, the authors argue that their metric improves over a correlation metric such as the mean correlation coefficient by additionally measuring interpretability. However, the linear operation of $\mathbf{z}_{\text{diff},i}^T\mathbf{w} + c$ can perform demixing, which means the measure gives no direct indication of identifiability and thus does not guarantee that the latent encodings are interpretable, especially in the case of dependent factors. Additionally, as noted by Kim and Mnih (2018), BetaVAE can report perfect accuracy when all but one of the ground-truth factors are disentangled, since the classifier can trivially attribute the remaining factor to the remaining latents.

FactorVAE (Kim and Mnih, 2018)

For the FactorVAE metric, the variance of the latent encodings is computed for a large (10,000 in DisLib) batch of data where all factors could possibly be changing. Latent dimensions with variance

below some threshold (0.05 in DisLib) are rejected and not considered further. Next, the encoding variance is computed again on a smaller batch (64 in DisLib) of data where one factor is fixed during sampling. The quotient of these two quantities (with the larger batch variance as the denominator) is then taken to obtain a normalized variance estimate per latent factor. Finally, a majority-vote classifier is trained to predict the index of the ground-truth factor with the latent unit that has the lowest normalized variance. The FactorVAE score is the classification accuracy for a batch of held-out data.

Mutual Information Gap (Chen et al., 2018)

The Mutual Information Gap (MIG) metric was introduced as an alternative to the classifier-based metrics. It provides a normalized measure of the mean difference in mutual information between each ground truth factor and the two latent codes that have the highest mutual information with the given ground truth factor. As it is implemented in DisLib, MIG measures entropy by discretizing the model's latent code using a histogram with 20 bins equally spaced between the representation minimum and maximum. It then computes the discrete mutual information between the ground-truth values and the discretized latents using the *scikit-learn* `metrics.mutual_info_score` function (Pedregosa et al., 2011). For the normalization it divides this difference by the entropy of the discretized ground truth factors.

Modularity (Ridgeway and Mozer, 2018)

Ridgeway and Mozer (2018) measure disentanglement in terms of three factors: modularity, compactness, and explicitness. For modularity, they first measure the mutual information between the discretized latents and ground-truth factors using the same histogram procedure that was used for the MIG, resulting in a matrix, $M \in \mathbb{R}^{D' \times D}$ with entries for each mutual information pair. Their measure of modularity is then

$$\text{modularity} = \frac{1}{D'} \sum_{i=1}^{D'} \Theta \left(1 - \frac{\sum_{j=1}^{D} M_{i,j}^2 - \max(M_i^2)}{\max(M_i^2)(D-1)} \right), \tag{26}$$

where $\max(M_i^2)$ returns the maximum of the vector of squared mutual information measurements between ground truth i and each latent factor. Additionally, Θ is a selection function that returns zero for any i where $\max(M_i^2) = 0$ and otherwise acts as the identity function.

DCI Disentanglement (Eastwood and Williams, 2018)

The DCI scores measure disentanglement, completeness, and informativeness, which have intuitive correspondence to the modularity, compactness, and explicitness of (Ridgeway and Mozer, 2018), respectively. To measure DCI Disentanglement, D regressors are trained to predict each ground truth factor state given the latent encoding. The DisLib implementation uses the `ensemble.GradientBoostingClassifier` function from *scikit-learn* with default parameters, which trains D gradient boosted logistic regression tree classifiers. Importance is assigned to each latent factor using the built-in `feature_importance_` property of the classifier, which computes the normalized total reduction of the classifier criterion loss contributed by each latent. Disentanglement is then measured as

$$\sum_{i=1} D(1 - H(I_i))\tilde{I}_i, \tag{27}$$

where H is the entropy computed with the `stats.entropy` function from *scikit-learn*, $I \in \mathbb{R}^{D \times D'}$ is a matrix of the absolute value of the feature importance between each factor and each ground truth, and $\tilde{I}$ is a normalized version of the matrix

$$\tilde{I}_i = \frac{\sum_{j=1}^{D'} I_{i,j}}{\sum_{k=1}^{D} \sum_{j=1}^{D'} I_{k,j}} \tag{28}$$

SAP Score (Kumar et al., 2018)

To compute the SAP score, Kumar et al. (2018) first train a linear support vector classifier with squared hinge loss and L_2 penalty to predict each ground truth factor from each latent variable. In DisLib this is implemented with the `svm.LinearSVC` function with default parameters from *scikit-learn*. They construct a score matrix $S \in \mathbb{R}^{D' \times D}$, where each entry in the matrix is the

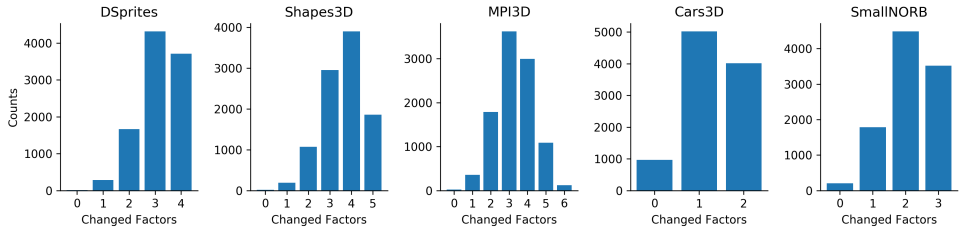


Figure 8: **Number of changing factors in LAP dataset**. For each dataset we sample 10,000 transitions and record the number of changing factors. These are indicated in the histograms. $\lambda = 1$, see Appendix D.

batch-mean classifier accuracy for predicting each ground truth given each individual latent encoding. For each generative factor, they compute the difference between the top two most predictive latent dimensions, which are the two highest scores in a given column of S. The mean (across ground-truth factors) of these differences is the SAP score.

D NATURAL DATASETS

We introduce several datasets to investigate disentanglement in more natural scenarios. Here, we provide an overview on the motivation and design of each dataset.

We have chosen to work with pairs of inputs as minimal sequences because we are interested in the first temporal derivative, more specifically in the sparsity of the transitions between pairs of images. Other methods that look at the second temporal derivative, such as work from Hénaff et al. (2019) on straightening, would require triplets as minimal sequences. Extending our approach beyond this minimal requirement would be simple in terms of the resulting ELBO (which would still factorise like in Eq. 4 because of the Markov property). The only additional complexity would be in the data and loss handling.

An issue with evaluating disentanglement on natural datasets is the fact that the existing disentanglement metrics require knowledge of the underlying generative process of the given data. Although we can observe that the world is composed of distinct entities that vary according to rules imposed by physics, we are unable to determine the appropriate "factors" that generate such scenes. To mitigate this problem, we compile object measurements by calculating the x and y coordinates of the center of mass as well as the area of object masks in natural video frames. We use these measurements to a) inform new disentanglement benchmarks with natural transitions that have similar complexity to existing benchmarks (Natural Sprites) and b) evaluate the ability of algorithms to decode intrinsic object properties (KITTI Masks). We additionally propose a simple extension to the existing DisLib datasets in the form of collecting images into pairs that exhibit sparse (i.e. Laplace) transition probabilities.

D.1 UNIFORM TRANSITIONS (UNI)

The UNI extension is based on the description given by Locatello et al. (2020), where the number of changing factors is determined using draws from a uniform distribution. The key differences between our implementation and theirs is: (i) their code[3] randomly (with 50% probability) sets $k = 1$ even in the $k = \text{Rnd}$ setting, and (ii) we ensure that exactly k factors change. Though we consider these discrepancies minor, we nonetheless label all results reported directly from Locatello et al. (2020) with "LOC", as opposed to "UNI", for clarity.

D.2 LAPLACE TRANSITIONS (LAP)

For each of the datasets in DisLib, we collect pairs of images. For each ground-truth factor, the first value in the pair is chosen from a uniform distribution across all possible values in latent space, while the second is chosen by weighting nearby values in latent space using Laplace distributed probabilities (see Eq. 2). We reject samples that would push a factor outside of the preset range provided by the dataset. We call this the *LAP* DisLib extension. Although the sparse prior indicates that any individual factor is more likely to remain constant, the number of factors that change in a given transition is still typically greater than one. To show this in Fig. 8, we sampled 10,000 transitions from each DisLib dataset with LAP transitions and computed the number of factors that had changed within a pair. This extension of the DisLib datasets provides a bridge from i.i.d. data to natural data by explicitly modeling the observed sparse marginal transition distributions. When training models on the LAP dataset it is possible to reject samples without transitions (i.e. all factors remain constant) since the pair would not result in any temporal learning signal. However, it would arguably be more natural to leave these samples as they would more accurately reflect occurrences of stationary objects in real data. We report the rejection setting in the main text, but found no significant difference between the two settings (see Appendix G).

This dataset also introduces a hyper-parameter λ that controls the rate of the Laplace sampling distribution, while the location is set by the initial factor value. Effectively, when this rate is $\lambda = 1$ most of the factors change most of the time, whereas for a rate of $\lambda = 10$ most of the factors will not change most of the time. Note that this means λ (inversely) changes the scale, which results in larger or smaller movements, but does not affect the distribution itself. In other words, the sparsity is unchanged, as the sparsity is controlled by the shape α. We fix $\lambda = 1$, which yields multiple changes, thus making this dataset fundamentally different both in spirit and in practice, from the UNI dataset.

D.3 YOUTUBE-VOS

For the YouTube dataset, we download annotations from the 2019 version of the video instance segmentation (Youtube-VIS) dataset (Yang et al., 2019)[4], which is built on top of the video object segmentation (Youtube-VOS) dataset (Xu et al., 2018). The dataset has multi-object annotations for every five frames in a 30fps video, which results in a 6fps sampling rate. The authors state that the temporal correlation between five consecutive frames is sufficiently strong that annotations can be omitted for intermediate frames to reduce the annotation efforts. Such a skip-frame annotation strategy enables scaling up the number of videos and objects annotated under the same budget, yielding 131,000 annotations for 2,883 videos, with 4,883 unique video object instances. Although we do not evaluate against YouTube-VOS in this study, we see it as the logical next step in transitioning to natural data. The large scale, lack of environmental constraints, and abundance of object types makes it the most challenging of the datasets considered herein.

The original image size of the YouTube-VOS dataset is 720×1280. In order to preserve the statistics of the transitions, we choose not to directly downsample to 64×64, but instead preserve the aspect ratio by downsampling to 64×128. In order to minimize the bias yielded by the extraction method, noting the center bias typically present in human videos, we extract three overlapping, equally spaced 64×64 pixel windows with a stride of 32. For each resulting $64 \times 64 \times T$ sequence, where T denotes the number of time steps in the sequence, we filter out all pairs where the given object instance is not present in adjacent frames, resulting in 234,652 pairs.

D.4 NATURAL SPRITES

The benchmark is available at `https://zenodo.org/record/3948069`.

Without a metric for disentanglement that can be applied to unknown data generating processes, we are limited to synthetic datasets with known ground-truth factors. Let us take dSprites (Matthey et al., 2017) as an example. The dataset consists of all combinations of a set of latent factor values, namely,

- Color: white

[3] `https://github.com/google-research/disentanglement_lib/blob/master/disentanglement_lib/methods/weak/train_weak_lib.py#L48`

[4] `https://competitions.codalab.org/competitions/20127`

Config	Scale	X	Y	(R, G, B)	Shape	Orientation
Continuous	YT [2375]	YT [197342]	YT [187112]	(1.0, 1.0, 1.0)	(square, triangle, star_4, spoke_4)	(0,9,...,342,351)
Discrete	YT [6]	YT [32]	YT [32]	(1.0, 1.0, 1.0)	(square, triangle, star_4, spoke_4)	(0,9,...,342,351)

Table 4: Natural Sprite Configs. Values in brackets refer to the number of unique values. Shapes presented are predefined in Spriteworld (Watters et al., 2019).

- Shape: square, ellipse, heart
- Scale: 6 values linearly spaced in $[0.5, 1]$
- Orientation: 40 values in $[0, 2\pi]$
- Position X: 32 values in $[0, 1]$
- Position Y: 32 values in $[0, 1]$

Given the limited set of discrete values each factor can take on, all possible samples can be described by a tractable dataset, compiled and released to the public. But, in reality, all of these factors should be continuous: a spectrum of possible colors, shapes, scales, orientations, and positions exist. We address this by constructing a dataset that is augmented with natural and continuous ground truth factors, using the mask properties measured from the YouTube dataset described in Appendix D.3.

We can choose the complexity of the dataset by discretizing the 234,652 transition pairs of position and scale into an arbitrary number of bins. In this study, we discretize to match the number of possible object states as dSprites, which we present in Table 4. This helps isolate the effect of including natural transitions from the effect of increasing data complexity. We produce a pair by fixing the color, shape, and orientation, but updating the position and scale with transitions sampled from the YouTube measurements. We motivate fixing shape and color by noting that this is consistent with object permanence in the real world. We decided to fix the orientation because we do not currently have a way to approximate it from object masks and we did not want to introduce artificial transition probabilities. To minimize the effect of extreme outliers, we filter out 10% of the data by removing frames if the mask area falls below the 5% or above the 95% quantiles, which reduces the number of pairs to 207,794. Finally, we use the Spriteworld (Watters et al., 2019) renderer to generate the images. Spriteworld allows us to render entirely new sprite objects at the precise position and scale as was measured from YouTube. For example, if one would want to apply YouTube-VOS transitions to MPI3D (Gondal et al., 2019), this option is unavailable without the associated renderer.

In relation to the Laplace transitions described in section D.2, this update i) produces pairs that correspond to transitions observed in real data, ii) allows for smooth transitions by defining the data generation process as opposed to being limited by the given collected dataset (e.g. dSprites), and iii) includes complex dependencies among factors that are present in natural data. We generate the data online, thus training the model to fit the underlying distribution as opposed to a sampled finite dataset.

However, as noted previously, all supervised metrics aggregated in DisLib are inapplicable to continuous factors, which is problematic as the generating distribution is effectively continuous with respect to a subset of the factors. Therefore, we limit our quantitative evaluation to MCC for continuous datasets. However, we are able to evaluate disentanglement with the standard metrics on the discretized version.

D.5 KITTI MOTS PEDESTRIAN MASKS (KITTI MASKS)

The benchmark is available at `https://zenodo.org/record/3931823`.

While Natural Sprites enables evaluation of disentanglement with natural transitions, we note that any disentanglement framework that requires knowledge of the underlying generative factors is unrealistic for real-world data. Measurements such as scale and position correspond to object properties that are ecologically relevant to the observer and can serve as suitable alternatives to the typical generative factors. We directly test this using our KITTI Masks dataset.

To create the dataset, we download annotations from the Multi-Object Tracking and Segmentation (MOTS) Evaluation Benchmark (Voigtlaender et al., 2019; Geiger et al., 2012; Milan et al., 2016),

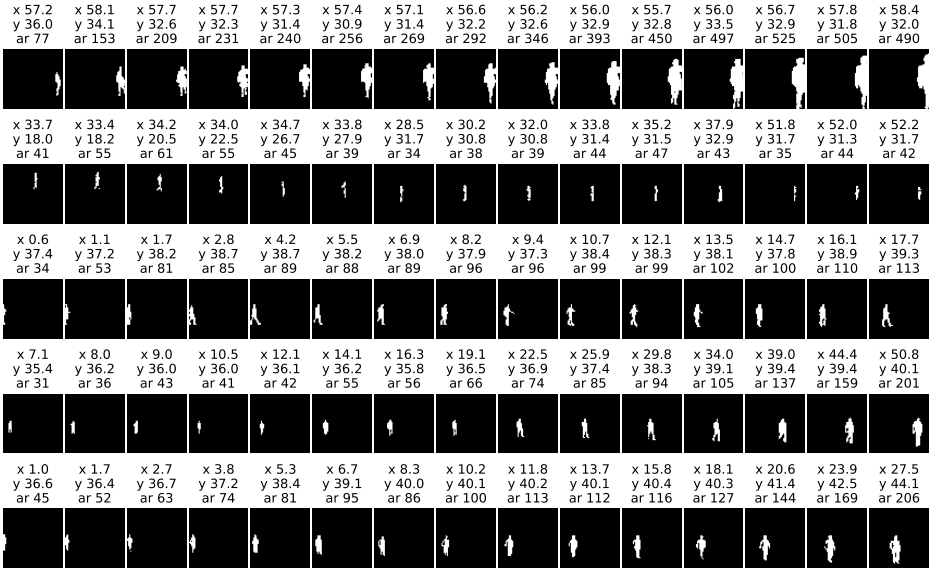


Figure 9: **KITTI Masks**. Each row corresponds to sequential frames from random sequences in the KITTI Mssks dataset. Above each image we denote measured object properties where x, y correspond the center of mass position and ar corresponds to the area.

which is split into KITTI MOTS and MOTSChallenge[5]. Both datasets contain sequences of pedestrians with their positions densely annotated in the time and pixel domains. For simplicity, we only consider the instance segmentation masks for pedestrians and do not use the raw data.

The resulting KITTI Masks dataset consists of 2,120 sequences of individual pedestrians with lengths between 2 and 710 frames each, resulting in a total of 84,626 individual frames. As we did with YouTube-VOS, we estimate ground truth factors by calculating the x and y coordinates of the center of mass of each pedestrian mask in each frame. We define the object size as the area of the mask, i.e. the total number of pixels. We consider the disentanglement performance for different mean time gaps between image pairs in table 2 and Appendix G.3. For samples and the corresponding ground truth factors see Fig. 9.

The original KITTI image sizes are 1080×1920 or 480×640 resolution for MOTSChallenge and between 370 and 374 pixels tall by 1224 and 1242 pixels wide for KITTI MOTS. The frame rates of the videos vary from 14 to 30 fps, which can be seen in Table 2 of Milan et al. (2016). We use nearest neighbor down-sampling for each frame such that the height was 64 pixels and the width is set to conserve the aspect ratio. After down-sampling, we use a horizontal sliding window approach to extract six equally spaced windows of size 64×64 (with overlap) for each sequence in both datasets. This results in a $64 \times 64 \times T$ sequence, where T denotes the number of time steps in the sequence. Note that here we make reasonable assumptions on horizontal translation and scale invariance of the dataset. We justify the assumed scale invariance by observing that the data is collected from a camera mounted onto a car which has varying distance to pedestrians. To confirm the translation invariance, we performed an ablation study on the number of horizontal images. Instead of six horizontal, equally spaced sliding windows, we only use two which leads to differently placed windows. We do not observe significant changes in the reported data statistics (e.g. the kurtosis of the fit stays within $\pm 10\%$ of the previous value for Δx transitions). The values of Δy and $\Delta area$ do not change significantly compared to Table 7.

For each resulting $64 \times 64 \times T$ sequence, where T denotes the number of time steps in the sequence, we extract all individual pedestrian masks based on their object instance identity and create a new sequence for each pedestrian such that each resulting sequence only contains a single pedestrian. We ignore images with masks that have less than 30 pixels as they are too far away or occluded and were

[5] `https://www.vision.rwth-aachen.de/page/mots`

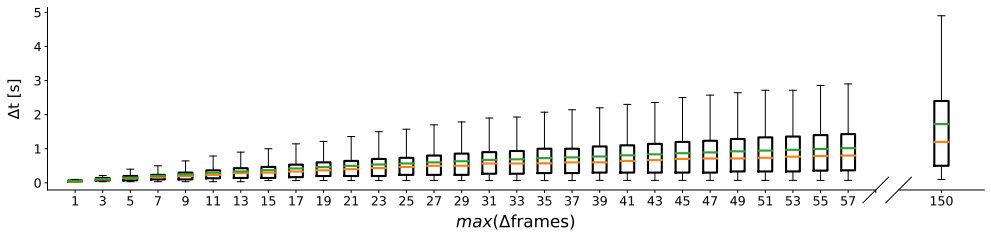


Figure 10: **KITTI Masks Δt**. Boxes indicate correspondence to physical time for different max(Δframes) in the KITTI Masks datasets. The orange line denotes the median and the green line the mean. The whiskers cover the 5th and 95th percentile of data.

not recognizable by the authors. We keep all sequences of two or more frames, as the algorithm only requires pairs of frames for training.

We leave the maximum distance between time frames within a pair, max(Δframes), as a hyperparameter. For a given max(Δframes), we report the mean change in physical time in seconds (denoted by mean(Δt)). We test adjacent frames (max(Δframes) $= 1$), which corresponds to a mean($\Delta t = 0.05$) and max(Δframes) $= 5$, which corresponds to a mean($\Delta t = 0.15$). This procedure is motivated by the fact that different sequences were recorded with different frame rates and reporting the mean(Δt) in seconds allows for a physical interpretation. The relationship between max(Δframes) and mean(Δt) is in Fig. 10. We show results for testing additional values of mean(Δt) in Appendix G.3.

During training, we augment the data by applying horizontal and vertical translations of ± 5 pixels and rotations of $\pm 2^\circ$ degree. We apply the exact same data augmentation to both images within a pair to not change any transition statistics.

We note that both YouTube-VOS (Xu et al., 2018; Yang et al., 2019) and KITTI-MOTS (Voigtlaender et al., 2019; Geiger et al., 2012; Milan et al., 2016) are multi-object datasets, although we consider each unique object (mask) separately. Multi-object representation learning and disentanglement are highly connected, in fact they have recently begun to be used interchangeably (Wulfmeier et al., 2020).

To briefly comment on possible extensions in this direction, we see no reason why our prior would not be beneficial to multi-object methods such as MONet (Burgess et al., 2019) and IODINE (Greff et al., 2019), or video extensions such as ViMON (Weis et al., 2020) and OP3 (Veerapaneni et al., 2019).

E Model Training and Selection

We train all models on all datasets provided in DisLib with the UNI and LAP variants.

All models are implemented in PyTorch (Paszke et al., 2019). To facilitate comparison, the training parameters, e.g. optimizer, batch size, number of training steps, as well as the VAE encoder and decoder architecture are identical to those reported in (Locatello et al., 2018; 2020). We use this architecture for all datasets, only adjusting the number of input channels (greyscale for dSprites, smallNORB, and KITTI Masks; three color channels for all other datasets).

The model formulation is agnostic to the direction of time. Therefore, to increase the temporal training signal at a fixed computational cost for each batch of input pairs $(\mathbf{x}_0, \mathbf{x}_1)$, we optimize the model in both directions i.e. optimizing the model objective for both $t_0 = 0$, $t_1 = 1$ as well as $t_0 = 1$, $t_1 = 0$.

F EXTENDED COMPARISONS AND CONTROLS

F.1 COMPARISON TO NONLINEAR ICA

F.1.1 THEORETICAL COMPARISON

Nonlinear ICA has recently been advanced significantly by several papers from Hyvärinen and colleagues. Of these studies, the two that are most comparable to our work is Hyvärinen and Morioka (2017), which uses an unsupervised contrastive loss for nonlinear demixing and Khemakhem et al. (2020a), which extends the nonlinear ICA framework to include variational autoencoders (VAEs). However, our theory covers an important class of transitions relevant for natural data that is not covered by the identifiability proofs of either of the aforementioned studies.

As a specific comparison to the first paper, the non-Gaussian autoregressive model that their identifiability proof rests upon (Eq. 8 in Hyvärinen and Morioka, 2017) assumes that the second derivative of the innovation probability density function is less than zero to satisfy *uniform dependence*, which is only met for $\alpha > 1$ for generalized Laplace transition distributions. While they denote (footnote 3) that Laplace distributions ($\alpha = 1$) are not covered by their theory, they offer a suggestion for a smooth approximation. However, they do not demonstrate that this approximation is useful in practice, or offer a solution to a general class of sparse distributions for $\alpha \leq 1$. We chose a generalized Laplacian to fit our data and for our model assumption as it allows for simple parameterization of fits to data (e.g. $\alpha = 0.5$ for natural movie transitions), but is simultaneously quite expressive (Sinz et al., 2009). Though we use $\alpha = 1$ in practice for our estimation method, we prove identifiability up to permutations and sign flips for any $\alpha < 2$, covering all sparse distributions under the expressive generalized Laplacian model. In addition, we assume a Gaussian marginal distribution that allows us to derive a fundamentally stronger proof of identifiability – where we identify up to permutation and sign-flips. Hyvärinen and Morioka (2017) only identify the sources up to arbitrary non-linear element-wise transformations. Thus they require a subsequent step of ICA (under the typical assumption that at most one marginal source distribution is Gaussian) to recover the signal up to permutations and sign flips for a class of distributions where it is unclear whether they account for temporal sparsity.

The work of Khemakhem et al. (2020a) has a couple of differences from our own, most notable of which is the form of the conditional prior, $p(\mathbf{z}_t|\mathbf{z}_{t-1})$. They assume that the conditional posterior is part of the exponential family, which does not include Laplacian conditionals. Though the exponential family contains the Laplace distribution with fixed mean as its member, it does not allow their approach to model sparse transitions. They assume that the natural parameters of the exponential family distribution are conditioned on $\mathbf{z}_{t-1}$, meaning that only the scale but not the mean of the Laplace prior for $\mathbf{z}_t$ can be modulated by the previous time step, thus not allowing for sparse transition probabilities. Additionally, their implementation requires the number of classes (i.e. states of the conditioning variable) to equal the number of stationary segments, which is impractical for the datasets we consider.

Thus, we provide a closer match to natural data transitions, with a stronger identifiability result. We provide validation by performing an extensive evaluation leveraging our contributed datasets as well as the models, metrics, and datasets provided by the Disentanglement Library (DisLib, discussed in section 4). We consider methods from the disentanglement literature (Locatello et al., 2020) as well as nonlinear ICA (Hyvärinen and Morioka, 2017), that are functionally capable of processing transitions.

F.1.2 EMPIRICAL COMPARISON

Hyvärinen and Morioka (2017) conducted a simulation where the sources in the nonlinear ICA model come from a linear autoregressive (AR) model with non-Gaussian innovations. Specifically, temporally dependent 20-dimensional source signals were randomly generated according to $\log p(s(t)|s(t-1)) = -|s(t) - 0.7s(t-1)|$. Though this generative process was noted to not be covered by the theory presented in (Hyvärinen and Morioka, 2017), the authors demonstrated that PCL could reconstruct the source signals reasonably well even for the nonlinear mixture case. Given our practical use of a Laplacian conditional, we found it a valuable comparison to evaluate our theory in this artificial setting.

Method	L=1	L=2	L=3	L=4	L=5
PCL	**0.998**	0.960	0.950	0.917	0.902
PCL (NF)	0.946	0.918	0.918	0.917	0.876
SlowFlow	0.997	**0.987**	**0.982**	**0.975**	**0.975**

Table 5: MCC using linear correlation where L denotes the number of mixing layers.

Given the discussion in Appendix B, we use SlowFlow for these experiments. For computational tractability in demixing highly nonlinear transformations, we consider normalizing flows (Dinh et al., 2017a;b; Kingma and Dhariwal, 2018), namely volume-preserving flows (Sorrenson et al., 2017), as we find constraining the Jacobian determinant stabilizes learning. To ensure sufficient expressivity, we consider 6 coupling blocks, each containing a 2-layer MLP with 500 hidden units and ReLU nonlinearities. We compare to the PCL implementation presented in (Hyvärinen and Morioka, 2017), where an MLP with the same number of hidden layers as the mixing MLP was adopted. We use 100 hidden units as we did not find increasing the value improved performance. To account for the architectural difference serving as a possible confounder, we use the same normalizing flow encoder for optimizing the PCL objective, which we term "PCL (NF)".

While (Hyvärinen and Morioka, 2017) used leaky ReLU nonlinearities to make the mixing invertible, said mixing is non-differentiable. This is problematic for SlowFlow, as it involves gradient optimization of the Jacobian term, and more importantly, unlike PCL, aims to explicitly recover the mixing process. We thus use a a smooth version of the leaky-ReLU activation function with a hyperparameter α (Gresele et al., 2020),

$$\mathrm{s_L}(x) = \alpha x + (1-\alpha)\log(1+e^x). \tag{29}$$

By ensuring the mixing process is smooth, we find that SlowFlow performs favorably relative to PCL (Table 5) when evaluated in the same setting, converging to a better optimum at higher levels of mixing.

F.2 Joint Factor Dependence Evaluation

In order to consider joint dependencies among natural generative factors, we leverage Natural Sprites to construct modified datasets where time-pairs of factors are shuffled per-factor (e.g. combining the x transition from one clip with the y transition from a different clip). This destroys dependencies between the factors, while maintaining the sparse marginal distributions. In Fig. 11 (right), we show 2D marginals before (blue) and after (orange) this shuffling. The additional density on the diagonals in the unshuffled data reveals dependencies between pairs of factors on both datasets. As mentioned in section 3.4, the observed dependency is mismatched from the theoretical assumptions of our model.

We test how robust SlowVAE is to such a mismatch by training it on the *permuted* data and re-evaluating disentanglement. In Table 22, we highlight that the improvement of SlowVAE on the permuted (i.e. independent) continuous Natural Sprites is not significant. In Table 21, we surprisingly find an overall improved score with non-permuted transitions (i.e. with dependencies), with three out of seven metrics showing a significant improvement. This is in line with Fig. 1f in Khemakhem et al. (2020b), where, at least for simple mixing, a model (Khemakhem et al., 2020a) that does not account for dependencies performs as well as one that does (Khemakhem et al., 2020b). We conclude that these preliminary results do not support the hypothesis that SlowVAE's disentanglement is reliant upon the model assumption that the factors are independent, but do acknowledge that the empirical effect of statistical dependence in natural video warrants further exploration (Träuble et al., 2020; Yang et al., 2020).

F.3 Transition Prior Ablation

We consider an ablated model which minimizes a KL-divergence term between the posteriors at time-step t and time-step $t-1$. This encourages the model to match the posteriors of both time points as closely as possible, and resembles a probabilistic variant of Slow Feature Analysis (Turner and Sahani, 2007). Specifically, we set $p(\mathbf{z}_t|\mathbf{z}_{t-1}) = q(\mathbf{z}_{t-1}|\mathbf{x}_{t-1})$, replacing the Laplace prior with the

posterior of the previous time step. This is equivalent to a Gaussian ($\alpha = 2$) transition prior, where the mean and variance are specified by the previous time step. We ablate over the regularization parameter γ and provide results in Tables 14 and 15, although we note that we still use the same hyperparameter values for SlowVAE as in all other experiments. As predicted by our theoretical result, $\alpha = 2$ leads to *entangled* representations in aggregate across evaluated datasets and metrics, even when considering a spectrum of γ values, resulting in a drastic reduction in scores, particularly on dSprites and Natural Sprites.

G Additional Results

G.1 Extended Data Analysis

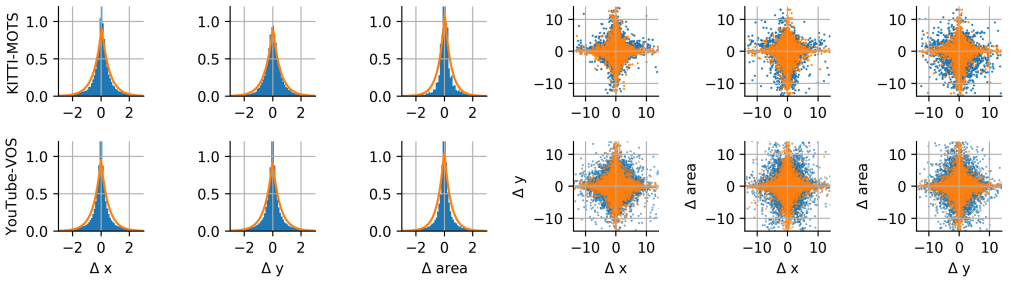


Figure 11: **Statistics of Natural Transitions**. Left) Distribution over transitions for horizontal (Δx) and vertical (Δy) position as well as mask/object size ($\Delta area$) for both datasets. Orange lines indicate fits of generalized Laplace distributions (Eq. 2). Right) 2D marginal distribution over pairs of factor transitions (blue) and permuted pairs (orange) that indicate the marginal distributions when made independent.

dataset	N	Δ area	Δ x	Δ y
KITTI-MOTS	82506	0.45	0.59	0.69
YouTube-VOS	234652	0.44	0.52	0.55

Table 6: Shape parameters (α) of the fitted generalized Laplace distributions in Fig. 11.

We report the empirical estimates of Kurtosis in Table 7. We report the log-likelihood scores for the Δ area, Δ x, Δ y statistics in Tables 8, 9, and 10, respectively for a Normal, a Laplace and a generalized Laplace/Normal distribution. For these distributions, we also report the fit parameters for the Δ area, Δ x, Δ y statistics in Tables 11, 12, and 13, respectively, where the shape parameter α of the generalized Laplacian is in bold face. As a higher likelihood indicates a better fit, we can see further evidence that natural transitions are highly leptokurtic; a Laplace distribution ($\alpha = 1$) is a better fit than a Gaussian ($\alpha = 2$), while the generalized Laplacian yields the highest likelihood consistently with $\alpha \approx 0.5$ for all measurements, as indicated in the main paper. For the plots in Figs. 1 and 11, we set the standard deviation of each component to 1 and clipped the minimum (-5) and maximum (5) values.

We note that while the marginal transitions appear sparse in metrics computed from the given object masks, our analysis considers 2D projections of objects instead of the transition statistics in their 3D environment. Understanding the relationship between 3D and 2D transition statistics is a compelling question from a broader perspective of visual processing, but unfortunately, the KITTI-MOTS masks (Voigtlaender et al., 2019; Geiger et al., 2012; Milan et al., 2016) lack the associated depth data required to answer it. Nonetheless, the natural scene statistics we compute are relevant, given that most computer vision models and vision-based animals see the 3D world as projected onto their 2D receptor arrays.

dataset	N	Δ area	Δ x	Δ y
KITTI	82506	68.92	38.50	65.39
YouTube	234652	76.49	39.98	35.59

Table 7: Empirical estimates of Kurtosis for mask transitions per metric for each dataset.

dataset	N	genlaplace	normal	laplace
KITTI	82506	-3.21e+05	-3.79e+05	-3.35e+05
YouTube	234652	-1.29e+06	-1.45e+06	-1.33e+06

Table 8: Maximum likelihood scores for the considered distributions on Δ **area** for each dataset.

dataset	N	genlaplace	normal	laplace
KITTI	82506	-8.72e+04	-1.20e+05	-9.25e+04
YouTube	234652	-4.50e+05	-5.64e+05	-4.74e+05

Table 9: Maximum likelihood scores for the considered distributions on Δx for each dataset.

dataset	N	genlaplace	normal	laplace
KITTI	82506	-7.59e+04	-1.07e+05	-7.86e+04
YouTube	234652	-4.40e+05	-5.45e+05	-4.60e+05

Table 10: Maximum likelihood scores for the considered distributions on Δy for each dataset.

dataset	N	genlaplace	normal	laplace
KITTI	82506	[**4.55e-01**, 1.00e+00, 1.01e+00]	[4.53e-01, 2.39e+01]	[1.00e+00, 1.07e+01]
YouTube	234652	[**4.44e-01**, 1.47e-16, 5.04e+00]	[2.25e-01, 1.16e+02]	[7.73e-09, 5.28e+01]

Table 11: Parameter fits for the considered distributions on Δ **area** for each dataset. The parameters are (alpha, location, scale) for generalized Laplace/Normal, (location, scale) for the other two distributions.

dataset	N	genlaplace	normal	laplace
KITTI	82506	[**5.87e-01**, 4.76e-02, 1.69e-01]	[5.34e-02, 1.04e+00]	[5.49e-02, 5.64e-01]
YouTube	234652	[**5.15e-01**, 1.15e-14, 2.57e-01]	[2.32e-03, 2.68e+00]	[7.54e-09, 1.38e+00]

Table 12: Parameter fits for the considered distributions on Δ **x** for each dataset. The parameters are (alpha, location, scale) for generalized Laplace/Normal, (location, scale) for the other two distributions.

dataset	N	genlaplace	normal	laplace
KITTI	82506	[**6.94e-01**, 1.02e-02, 2.32e-01]	[3.84e-02, 8.86e-01]	[1.71e-02, 4.77e-01]
YouTube	234652	[**5.48e-01**, 2.93e-13, 3.08e-01]	[8.81e-03, 2.47e+00]	[9.15e-04, 1.30e+00]

Table 13: Parameter fits for the considered distributions on Δ **y** for each dataset. The parameters are (alpha, location, scale) for generalized Laplace/Normal, (location, scale) for the other two distributions.

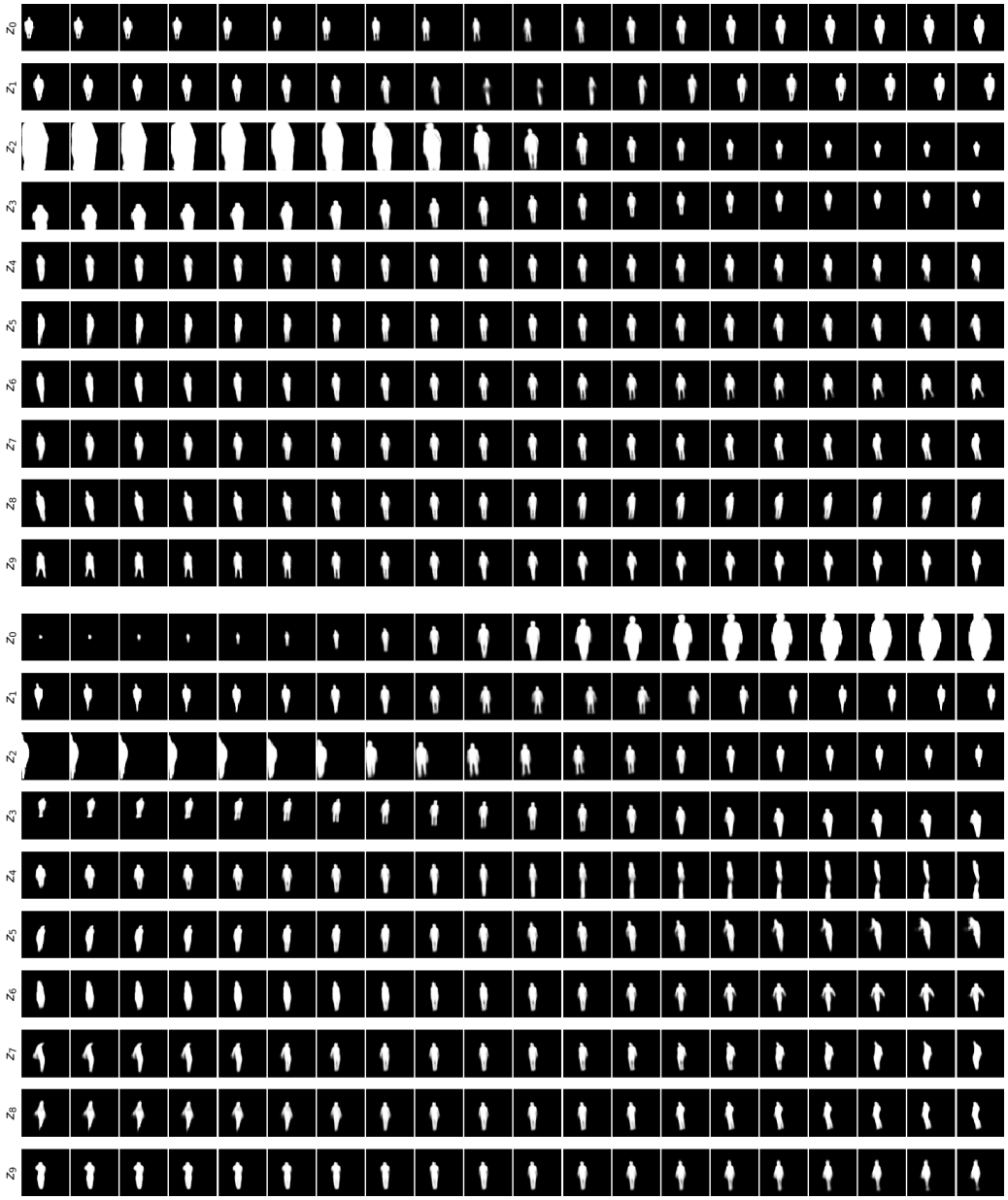


Figure 12: **KITTI Masks Latent Representations.** We show axis latent traversals along each dimension for the β-VAE (top) and SlowVAE (bottom). Here, the latents z_i are sorted from top to bottom in ascending order according to the mean variance output of the encoder. With MCC correlation (see e.g. Fig. 20) the known ground truth factors are matched as following: β-VAE: scale$\sim$ z_2, x-position$\sim$ z_1 and y-position$\sim$ z_3; SlowVAE: scale$\sim$ z_0, x-position$\sim$ z_1 and y-position$\sim$ z_3. With these latent visualizations alone, there is no significant difference visible between β-VAE and SlowVAE. However, we see a quantitative difference with the MCC score (see Table 2) and a qualitative difference when directly observing latent embeddings (see Fig. 20).

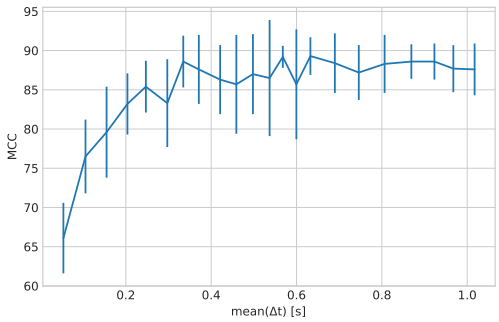


Figure 13: Ablation over mean(Δt) for SlowVAE. Mean and standard deviation (s.d.) MCC scores

Model	Data	BetaVAE	FactorVAE	MIG	MCC	DCI	Modularity	SAP
SlowVAE	dSprites (Laplace)	100.0 (0.0)	97.5 (3.0)	29.5 (9.3)	69.8 (2.3)	65.4 (3.6)	96.5 (1.6)	8.1 (3.0)
PM-VAE (16)	dSprites (Laplace)	64.1 (7.0)	44.8 (13.0)	5.2 (2.3)	45.0 (5.5)	5.9 (3.9)	93.5 (1.9)	1.7 (0.8)
PM-VAE (10)	dSprites (Laplace)	78.8 (7.5)	59.4 (11.2)	5.9 (1.8)	49.2 (4.3)	13.6 (5.6)	92.7 (3.0)	3.9 (1.7)
PM-VAE (8)	dSprites (Laplace)	82.9 (2.8)	61.2 (5.7)	7.1 (2.6)	49.6 (3.3)	14.5 (3.5)	91.6 (3.0)	4.3 (1.6)
PM-VAE (4)	dSprites (Laplace)	86.6 (2.7)	64.1 (7.2)	11.6 (5.0)	52.0 (3.8)	22.9 (3.7)	90.9 (2.7)	5.7 (2.8)
PM-VAE (2)	dSprites (Laplace)	86.3 (2.4)	62.9 (7.7)	10.9 (3.2)	50.0 (3.5)	21.2 (5.3)	92.3 (1.9)	5.5 (2.0)
PM-VAE (1)	dSprites (Laplace)	82.5 (5.4)	58.4 (6.0)	7.6 (3.6)	45.9 (4.9)	14.4 (5.1)	92.1 (4.0)	4.0 (2.0)
SlowVAE	Natural (Discrete)	82.6 (2.2)	76.2 (4.8)	11.7 (5.0)	52.6 (4.1)	18.9 (5.5)	88.1 (3.6)	4.4 (2.3)
PM-VAE (16)	Natural (Discrete)	72.7 (2.8)	49.2 (3.7)	2.8 (1.2)	38.3 (3.2)	6.9 (1.8)	85.3 (1.8)	1.2 (0.7)
PM-VAE (10)	Natural (Discrete)	76.6 (3.6)	52.0 (4.9)	3.8 (2.2)	39.0 (3.9)	7.3 (1.8)	87.0 (2.2)	2.0 (1.0)
PM-VAE (8)	Natural (Discrete)	74.6 (3.4)	49.3 (4.4)	3.1 (1.8)	38.9 (3.2)	7.1 (1.8)	87.8 (1.7)	1.6 (1.0)
PM-VAE (4)	Natural (Discrete)	73.8 (3.8)	48.8 (5.3)	2.7 (1.5)	35.7 (3.5)	6.7 (2.0)	87.4 (2.2)	1.6 (0.9)
PM-VAE (2)	Natural (Discrete)	73.4 (3.1)	47.0 (5.3)	2.2 (1.1)	36.8 (2.4)	6.2 (1.5)	87.4 (1.9)	1.1 (0.6)
PM-VAE (1)	Natural (Discrete)	73.5 (3.3)	49.7 (5.4)	3.1 (1.6)	36.9 (3.2)	6.9 (1.8)	86.9 (2.2)	1.8 (0.7)

Table 14: Mean and standard deviation (s.d.) metric scores across 10 random seeds. PM-VAE (γ) refers to replacing the Laplace prior with a KL-divergence term between the (Gaussian) posteriors at time-step t and time-step $t-1$, with conditional prior regularization, γ.

G.2 All DisLib Results

We include results on all DisLib datasets, dSprites (Matthey et al., 2017), Cars3D (Reed et al., 2015), SmallNORB (LeCun et al., 2004), Shapes3D (Kim and Mnih, 2018), MPI3D (Gondal et al., 2019), in Tables 16, 17, 18, 19, and 20, respectively. We report both median (a.d.) to compare to the previous median scores reported in (Locatello et al., 2020), as well as the the more common mean (s.d.) scores for future comparisons and straightforward statistical estimates of significant differences between models. We also consider allowing for static transitions, which we denote with "NC", e.g. "LAP-NC", in the tabular results. As mentioned in Section 5, we use the same parameter settings for SlowVAE in all experiments, while model selection was performed not only per dataset, but per seed, for results from (Locatello et al., 2020).

G.3 KITTI Masks Δt Ablation

As seen in the main text, considering image pairs separated further apart in time appears beneficial. Here we evaluate a wider range by taking frames which are further apart in a sequence. max(Δframes) $= N$ indicates that all pairs differ by *at most* N frames. We chose an upper bound of N, rather than sampling pairs with a fixed separation, to account for the variable frame rates and sequence lengths in the original dataset (Milan et al., 2016) without introducing a confounding factor of varying dataset size. We report in Fig. 10 how the max(Δframes) criterion corresponds to the mean time gap between image pairs (mean(Δt)) in seconds. For further details, we refer to Appendix D.5.

In Fig. 13 we visualize an ablation over mean(Δt). We find that model performance increased initially with larger temporal separation between data points, then plateaued. We also observe in Fig. 14 that the measured factor marginals remain sparse, with $\alpha < 1$, for all tested settings of mean(Δt).

Model	Data	MCC
SlowVAE	Natural (Continuous)	49.1 (4.0)
PM-VAE (16)	Natural (Continuous)	35.2 (3.7)
PM-VAE (10)	Natural (Continuous)	33.2 (2.1)
PM-VAE (8)	Natural (Continuous)	32.7 (3.1)
PM-VAE (4)	Natural (Continuous)	33.7 (2.3)
PM-VAE (2)	Natural (Continuous)	32.4 (3.2)
PM-VAE (1)	Natural (Continuous)	34.2 (3.4)
SlowVAE	Kitti (mean(Δt) $= 0.05s$)	66.1 (4.5)
PM-VAE (16)	Kitti (mean(Δt) $= 0.05s$)	63.1 (9.3)
PM-VAE (10)	Kitti (mean(Δt) $= 0.05s$)	57.4 (8.5)
PM-VAE (8)	Kitti (mean(Δt) $= 0.05s$)	59.0 (5.6)
PM-VAE (4)	Kitti (mean(Δt) $= 0.05s$)	51.8 (9.2)
PM-VAE (2)	Kitti (mean(Δt) $= 0.05s$)	50.3 (7.4)
PM-VAE (1)	Kitti (mean(Δt) $= 0.05s$)	38.4 (6.8)
SlowVAE	Kitti (mean(Δt) $= 0.15s$)	79.6 (5.8)
PM-VAE (16)	Kitti (mean(Δt) $= 0.15s$)	69.6 (5.9)
PM-VAE (10)	Kitti (mean(Δt) $= 0.15s$)	78.2 (6.0)
PM-VAE (8)	Kitti (mean(Δt) $= 0.15s$)	73.8 (10.0)
PM-VAE (4)	Kitti (mean(Δt) $= 0.15s$)	67.9 (10.4)
PM-VAE (2)	Kitti (mean(Δt) $= 0.15s$)	60.7 (8.8)
PM-VAE (1)	Kitti (mean(Δt) $= 0.15s$)	60.9 (9.1)

Table 15: Continuous ground-truth variable datasets. See Table 14 for details.

Model (Data)	BetaVAE	FactorVAE	MIG	DCI	Modularity	SAP
β-VAE (*i.i.d.*)	82.3	66.0	10.2	18.6	82.2	4.9
Ada-ML-VAE (LOC)	89.6	70.1	11.5	29.4	89.7	3.6
Ada-GVAE (LOC)	92.3	84.7	26.6	47.9	91.3	7.4
SlowVAE (UNI)	89.7 (3.8)	81.4 (8.4)	34.5 (9.6)	50.0 (6.9)	87.1 (2.0)	5.1 (1.5)
SlowVAE (LAP)	100.0 (0.0)	99.2 (2.3)	28.2 (8.2)	65.5 (3.1)	96.8 (1.4)	6.0 (2.4)
SlowVAE (LAP-NC)	100.0 (0.2)	97.4 (4.4)	29.1 (7.1)	62.0 (4.2)	97.4 (1.6)	8.2 (2.9)
SlowVAE (UNI)	87.0 (5.1)	75.2 (11.1)	28.3 (11.5)	47.7 (8.5)	86.9 (2.8)	4.4 (2.0)
SlowVAE (LAP)	100.0 (0.0)	97.5 (3.0)	29.5 (9.3)	65.4 (3.6)	96.5 (1.6)	8.1 (3.0)
SlowVAE (LAP-NC)	99.8 (0.6)	95.2 (6.0)	27.6 (8.6)	61.5 (5.3)	96.8 (1.8)	8.4 (3.4)

Table 16: **dSprites.** Median and absolute deviation (a.d.) metric scores across 10 random seeds (first three rows are from (Locatello et al., 2020)). The bottom three rows give mean and standard deviation (s.d.) for the models presented in this paper.

Increasing mean(Δt) leads to increased diversity, and thus more information in the learning signal. However, it is worth noting that since SlowVAE assumes $\alpha = 1$ in the transitions, an increase in α from increasing the temporal gap leads to a reduction in mismatch.

Our results on increasing the temporal difference within pairs of inputs is in agreement with recent work by Oord et al. (2018, Table 2), who show increased performance in representation learning for larger separation between positive samples in a contrastive objective function. Additional related work from Tschannen et al. (2019) shows that temporal separation between frame embeddings influences the representation that is learned from videos.

G.4 Latent Space Visualizations

We visualize differences in learned latent representations using image embedding in Figures 15- 28. We show four different plots for each dataset considered and include all available models. Each figure corresponds to a different dataset.

Model (Data)	BetaVAE	FactorVAE	MIG	DCI	Modularity	SAP
β-VAE (*i.i.d.*)	100.0	87.9	8.8	22.5	90.2	1.0
Ada-ML-VAE (LOC)	100.0	87.4	14.7	45.6	94.6	2.8
Ada-GVAE (LOC)	100.0	90.2	15.0	54.0	93.9	9.4
SlowVAE (UNI)	100.0 (0.0)	90.4 (0.4)	15.7 (1.5)	48.9 (1.7)	95.7 (1.0)	1.6 (0.4)
SlowVAE (LAP)	100.0 (0.0)	91.0 (2.5)	9.7 (1.1)	51.0 (2.2)	94.4 (1.1)	1.7 (0.9)
SlowVAE (LAP-NC)	100.0 (0.0)	90.8 (1.1)	9.3 (1.1)	50.0 (2.0)	94.6 (0.9)	0.9 (0.9)
SlowVAE (UNI)	100.0 (0.0)	90.4 (0.5)	15.4 (2.2)	48.0 (2.4)	95.4 (1.5)	1.6 (0.5)
SlowVAE (LAP)	100.0 (0.0)	90.2 (3.5)	10.4 (1.8)	50.9 (2.7)	94.1 (1.2)	2.0 (1.1)
SlowVAE (LAP-NC)	100.0 (0.0)	90.9 (1.2)	9.5 (1.4)	50.2 (2.7)	95.0 (1.2)	1.7 (1.4)

Table 17: **Cars3D.** Median and absolute deviation (a.d.) metric scores across 10 random seeds (first three rows are from (Locatello et al., 2020)). The bottom three rows give mean and standard deviation (s.d.) for the models presented in this paper.

Model (Data)	BetaVAE	FactorVAE	MIG	DCI	Modularity	SAP
β-VAE (*i.i.d.*)	74.0	49.5	21.4	28.0	89.5	9.8
Ada-ML-VAE (LOC)	91.0	72.1	31.1	34.1	86.1	15.3
Ada-GVAE (LOC)	87.9	55.5	25.6	33.8	78.8	10.6
SlowVAE (UNI)	78.8 (2.1)	46.2 (1.9)	23.7 (1.3)	28.8 (0.6)	92.1 (1.6)	7.8 (1.0)
SlowVAE (LAP)	86.0 (0.2)	72.9 (0.7)	25.8 (0.5)	42.7 (0.9)	97.7 (0.3)	6.5 (0.4)
SlowVAE (LAP-NC)	86.1 (0.7)	73.7 (0.6)	26.3 (0.5)	42.5 (0.6)	97.6 (0.3)	6.5 (0.9)
SlowVAE (UNI)	78.2 (3.8)	47.0 (2.9)	23.8 (1.8)	28.7 (0.7)	90.9 (2.1)	7.8 (1.1)
SlowVAE (LAP)	85.9 (0.3)	73.1 (0.9)	25.7 (0.6)	42.6 (0.9)	97.5 (0.3)	6.8 (0.5)
SlowVAE (LAP-NC)	85.7 (1.0)	73.3 (0.8)	26.2 (0.7)	42.6 (0.8)	97.6 (0.5)	6.6 (1.3)

Table 18: **SmallNORB.** Median and absolute deviation (a.d.) metric scores across 10 random seeds (first three rows are from (Locatello et al., 2020)). The bottom three rows give mean and standard deviation (s.d.) for the models presented in this paper.

In Figures 15- 21 we display the mean correlation coefficient matrix and the latent representations for each ground-truth, as described in the main text for Fig. 5.

The top row is the sorted absolute correlation coefficient matrix between the latents (rows) and the ground truth generating factors (columns). The latent dimensions are permuted such that the sum on the diagonal is maximal. This is achieved by an optimal, non-greedy matching process for each ground truth factor with its corresponding latent, as described in appendix C. As such, a more

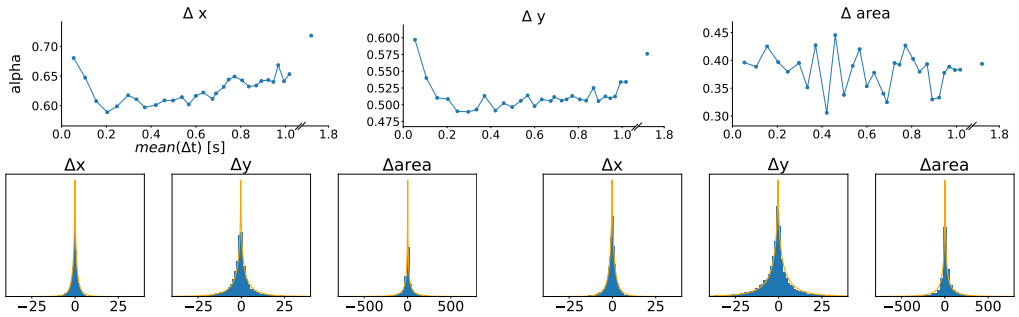


Figure 14: **KITTI Masks Sparseness**. We show the sparseness over time of the transitions for horizontal (Δx), vertical (Δy) as well as mask/object size (Δarea) in KITTI Masks by plotting the α of a generalized Laplace fit for different mean(Δt) (top). To display the quality of the fits, we show two exemplary fits at mean(Δt) $= 0.63$ (bottom-left) and mean(Δt) $= 1.02$ (bottom-right).

Model (Data)	BetaVAE	FactorVAE	MIG	DCI	Modularity	SAP
β-VAE (*i.i.d.*)	98.6	83.9	22.0	58.8	93.8	6.2
Ada-ML-VAE (LOC)	100.0	100.0	50.9	94.0	98.8	12.7
Ada-GVAE (LOC)	100.0	100.0	56.2	94.6	97.5	15.3
SlowVAE (UNI)	100.0 (0.1)	97.3 (4.0)	64.4 (8.4)	82.6 (4.4)	95.5 (1.6)	5.8 (0.9)
SlowVAE (LAP)	100.0 (0.0)	95.9 (2.6)	62.5 (3.1)	85.6 (4.0)	98.1 (0.6)	8.2 (1.7)
SlowVAE (LAP-NC)	100.0 (1.6)	97.0 (2.0)	63.6 (5.4)	86.7 (4.1)	98.4 (1.4)	7.0 (2.1)
SlowVAE (UNI)	99.9 (0.3)	95.4 (5.2)	58.8 (13.0)	82.3 (5.4)	95.2 (2.0)	5.7 (1.4)
SlowVAE (LAP)	100.0 (0.0)	95.0 (3.2)	61.5 (4.5)	85.0 (4.7)	98.3 (0.8)	8.9 (2.6)
SlowVAE (LAP-NC)	98.4 (4.9)	97.4 (2.4)	61.6 (10.6)	86.1 (5.2)	98.2 (1.6)	8.2 (2.6)

Table 19: **Shapes3D.** Median and absolute deviation (a.d.) metric scores across 10 random seeds (first three rows are from (Locatello et al., 2020)). The bottom three rows give mean and standard deviation (s.d.) for the models presented in this paper.

Model (Data)	BetaVAE	FactorVAE	MIG	DCI	Modularity	SAP
β-VAE (*i.i.d.*)	54.6	32.2	7.2	19.5	87.4	3.7
Ada-ML-VAE (LOC)	72.6	47.6	24.1	28.5	87.5	7.4
Ada-GVAE (LOC)	78.9	62.1	28.4	40.1	91.6	21.5
SlowVAE (UNI)	58.5 (0.9)	38.6 (2.3)	32.2 (1.0)	29.9 (1.3)	89.2 (2.0)	8.8 (0.8)
SlowVAE (LAP)	67.6 (6.1)	42.4 (6.1)	32.0 (1.8)	35.9 (2.2)	89.5 (1.5)	9.7 (0.8)
SlowVAE (LAP-NC)	60.1 (2.7)	39.2 (1.7)	30.6 (0.7)	34.3 (0.7)	85.9 (1.1)	9.3 (0.9)
SlowVAE (UNI)	58.6 (1.1)	38.5 (3.2)	32.2 (1.2)	30.1 (1.6)	89.4 (2.6)	8.7 (1.0)
SlowVAE (LAP)	66.6 (6.9)	45.5 (8.3)	32.9 (2.6)	35.5 (2.7)	89.2 (1.9)	9.7 (1.2)
SlowVAE (LAP-NC)	61.0 (3.6)	40.3 (2.5)	30.4 (0.8)	34.2 (1.0)	86.6 (1.7)	9.3 (1.0)

Table 20: **MPI3D.** Median and absolute deviation (a.d.) metric scores across 10 random seeds (first three rows are from (Locatello et al., 2020)). The bottom three rows give mean and standard deviation (s.d.) for comparison with other tables.

prevalent diagonal structure corresponds to a better mapping between the ground-truth factors and latent encoding.

The middle set of plots are latent embeddings of random training data samples. The x-axis denotes the ground truth generating factor and the y-axis denotes the corresponding latent factor as matched according to the main diagonal of the correlation matrix. For each dataset, we further color-code the latents by a categorical variable as denoted in each figure.

The bottom set of plots show the ground truth encoding compared to the second best latent as opposed to the diagonally matched latent. This plot can be used to judge how much the correspondence between latents is one-to-one or rather one-to-many.

To further investigate the latent representations, we show a scatter plot over the best and second best latents in figures 22-28. Here, the color-coding is matched by the ground truth factor denoted in each row.

When comparing the correlation matrix with the corresponding scatter plots, one can see that embeddings with sinusoidal curves have low correlation, which illustrates a shortcoming of the metric. Another limitation is that categorical variables which have no natural ordering have an order-dependent MCC score, indicating the permutation variance of MCC. With SlowVAE, we can infer three different types of embeddings. First, we have simple ordered ground truth factors with non-circular boundary conditions. Here, SlowVAE models often show a clear one-to-one correspondence

Model	γ	λ	Data	Permuted?	BetaVAE	FactorVAE	MIG	MCC	DCI	Modularity	SAP
SlowVAE	10	6	Natural (Discrete)	Yes	77.6 (4.1)	69.7 (6.5)	8.5 (4.4)	49.9 (3.5)	17.6 (2.8)	89.8 (3.2)	1.8 (0.9)
SlowVAE	10	6	Natural (Discrete)	No	**82.6** (2.2)	**76.2** (4.8)	11.7 (5.0)	52.6 (4.1)	18.9 (5.5)	88.1 (3.6)	**4.4** (2.3)

Table 21: Impact of removing natural dependence on Discrete Natural Sprites.

Model	γ	λ	Data	Permuted?	MCC
SlowVAE	10	6	Natural (Continuous)	Yes	52.9 (4.2)
SlowVAE	10	6	Natural (Continuous)	No	49.1 (4.0)

Table 22: Impact of removing natural dependence on Continuous Natural Sprites.

(e.g. Fig 22 scale, x-position and y-position; Fig 25 θ-rotation; Fig 26 Φ-rotation). Second, we observe circular embeddings due to boundary conditions for certain factors (e.g. Fig 15, 22 3rd row; Fig 16, 23 2nd row). Note that not all datasets with orientations exhibit full rotations and thus do not have circular boundary conditions, e.g. smallNORB. Finally, we have categorical variables, where no order exists (e.g. Fig. 16, 23 top row, Fig 17, 24 top row, Fig 18, 25 top row) resulting in separated but not necessarily ordered clusters.

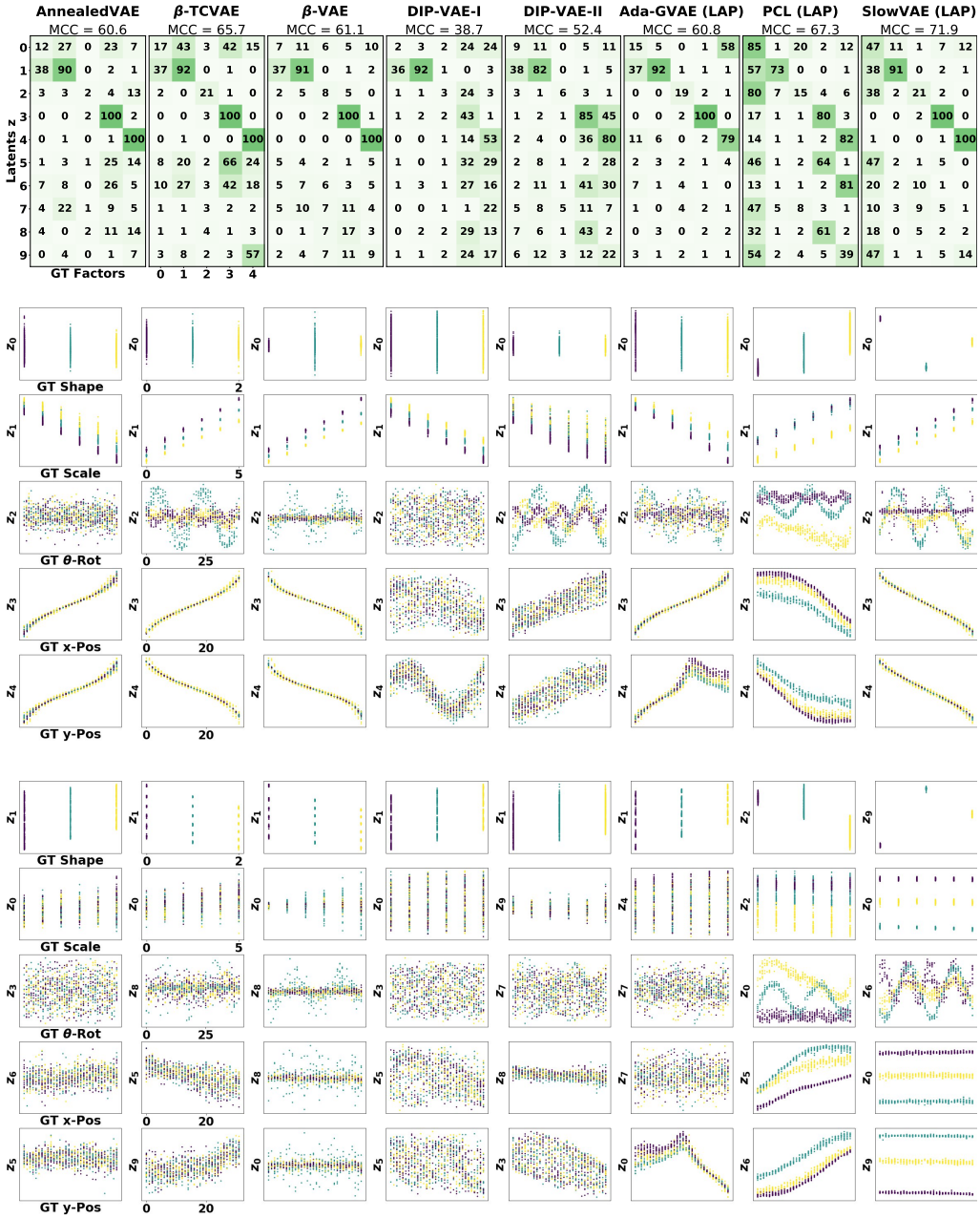


Figure 15: **DSprites Latent Representations.** Top, MCC correlation matrices. Middle five rows, model latent over highest correlating ground truth factor. Bottom five rows, model latent over second highest correlating ground truth factor. The color-coding corresponds to the shapes: heart/yellow, ellipse/turquoise and square/purple.

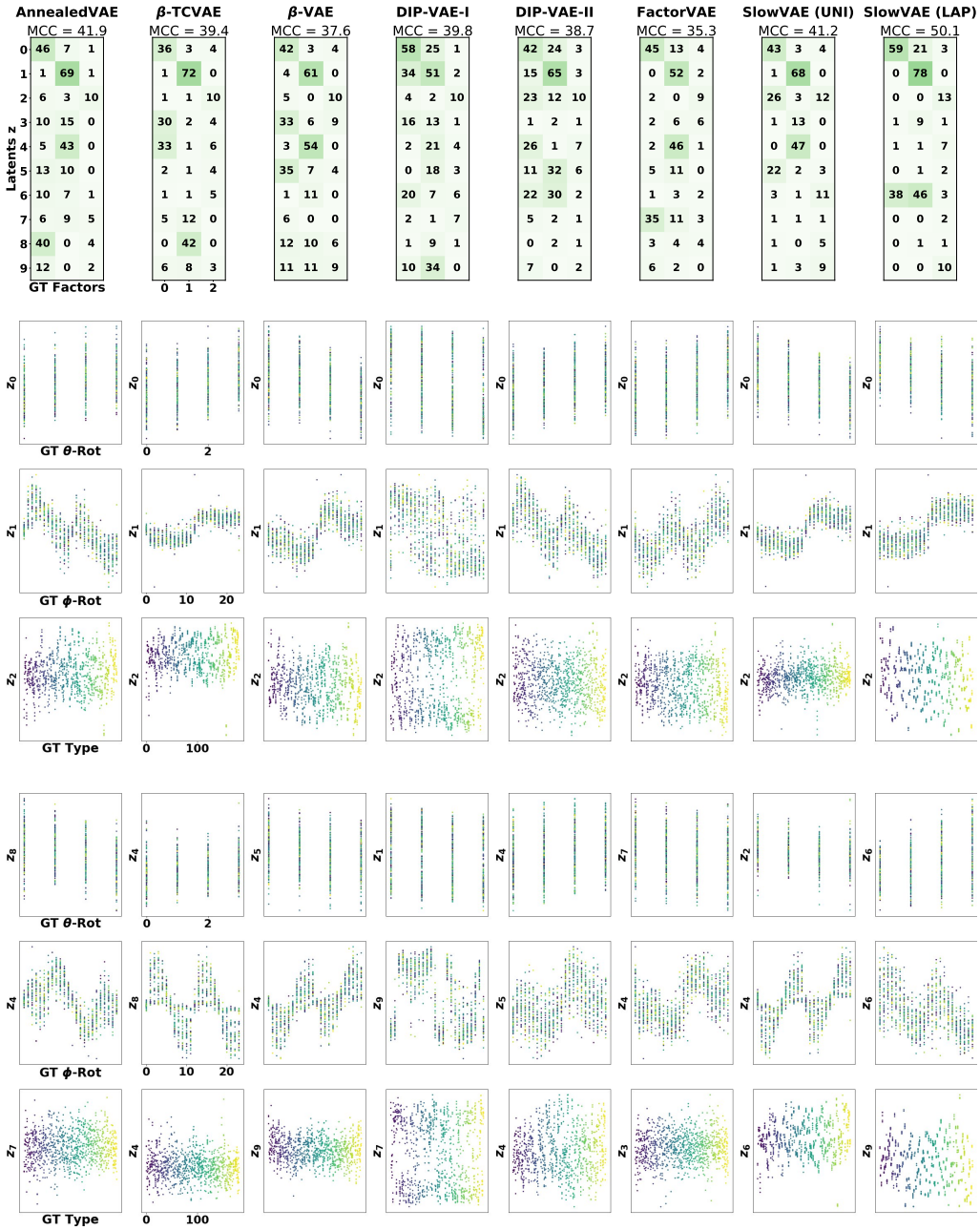


Figure 16: **Cars3D Latent Representations.** Top, MCC correlation matrices. Middle three rows, model latent over highest correlating ground truth factor. Bottom three rows, model latent over second highest correlating ground truth factor. The color-coding corresponds to the 183 different car types (GT Types) in the dataset.

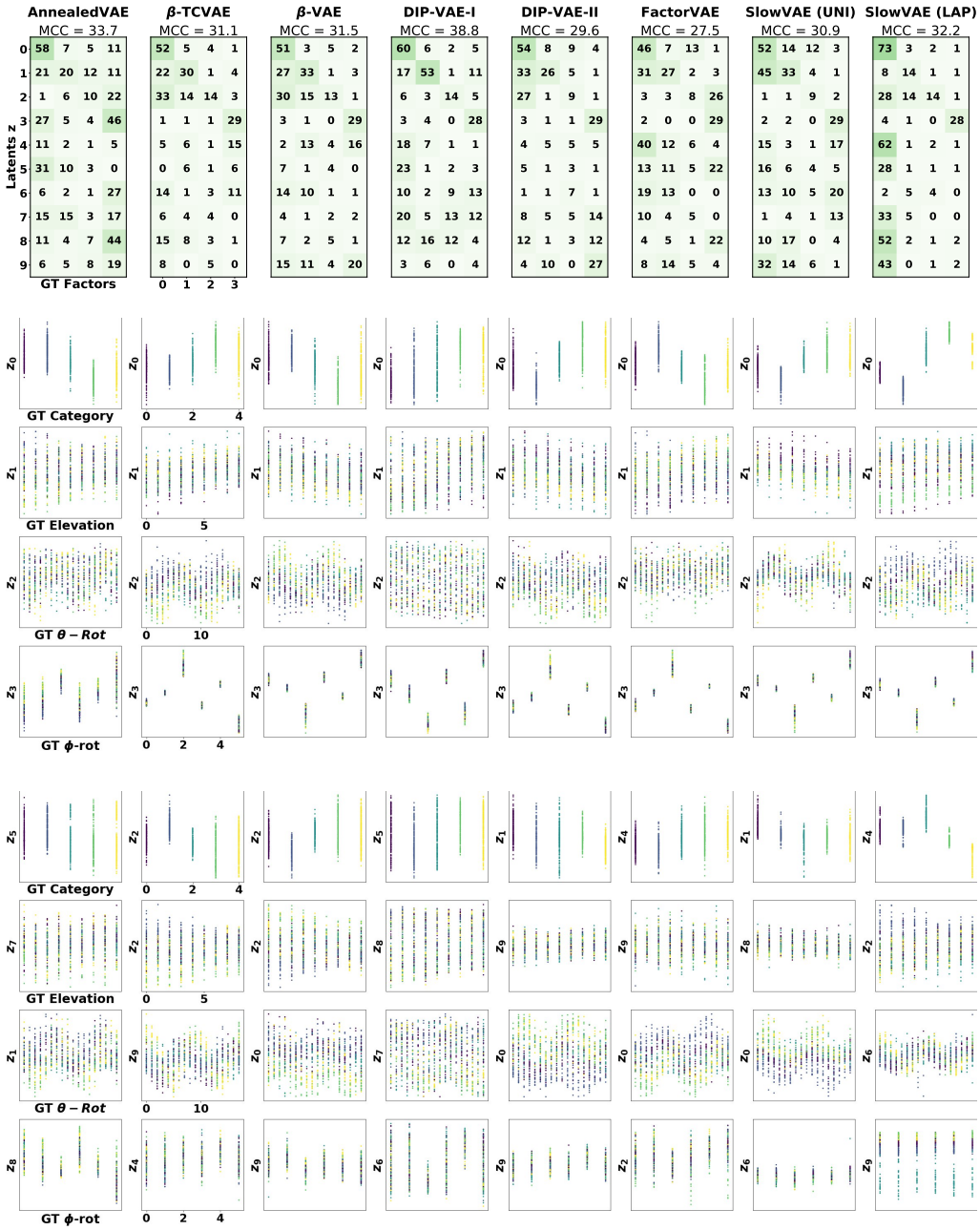


Figure 17: **SmallNorb Latent Representations.** Top, MCC correlation matrices. Middle four rows, model latent over highest correlating ground truth factor. Bottom four rows, model latent over second highest correlating ground truth factor. The color-coding corresponds to the five different GT categories in the dataset.

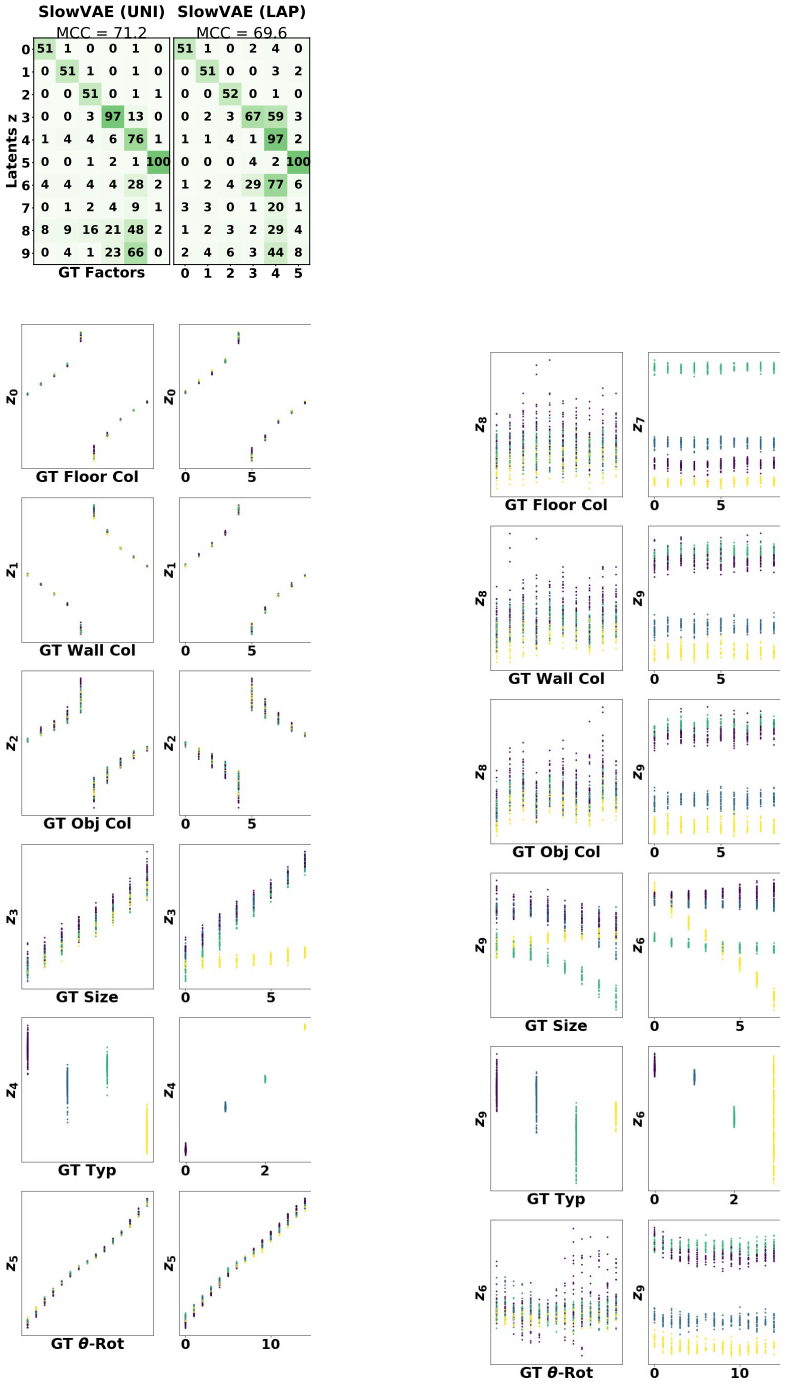


Figure 18: **Shapes3D Latent Representations.** Top, MCC correlation matrices. Left two columns, model latent over highest correlating ground truth factor. Right two columns, model latent over second highest correlating ground truth factor. The color-coding corresponds to the four different object types (GT-Type) in the dataset.

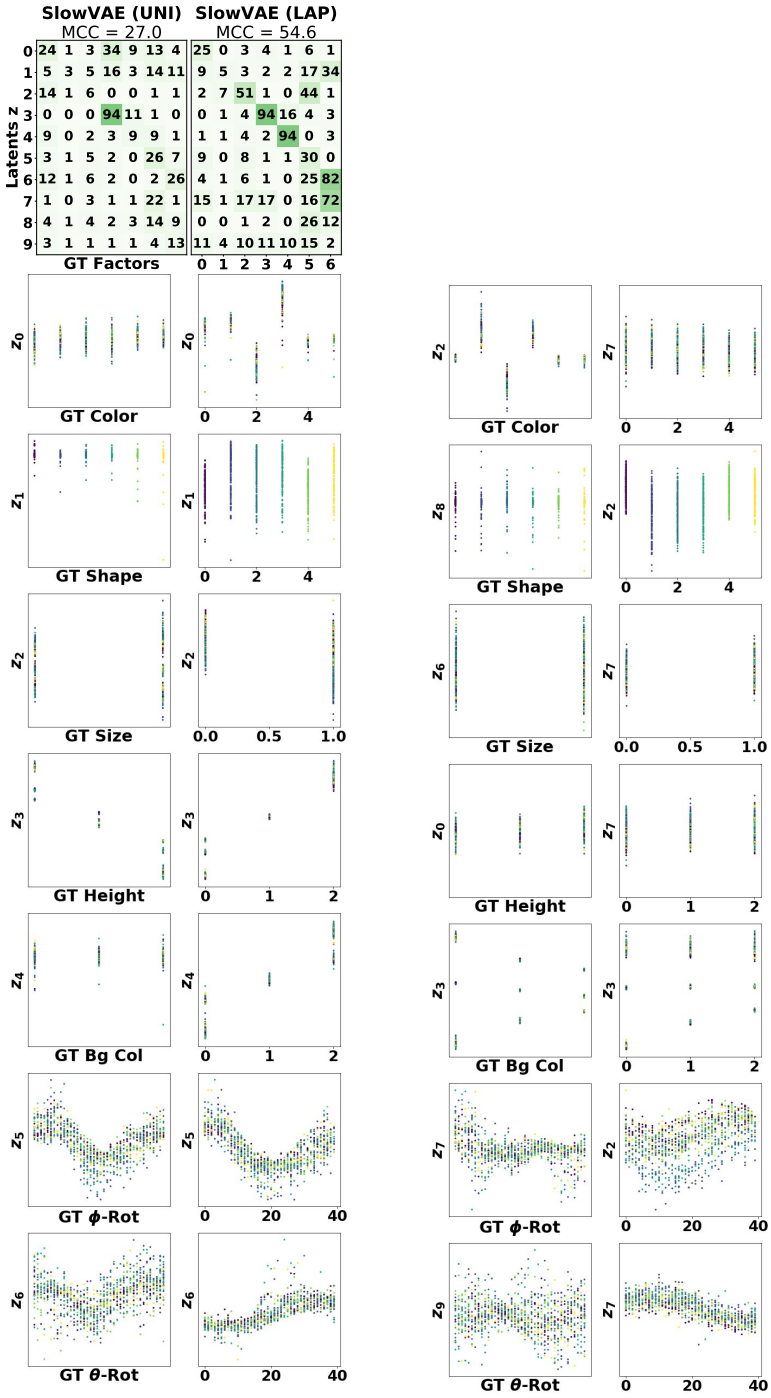


Figure 19: **MPI3DReal Latent Representations.** Top, MCC correlation matrices. Left two columns, model latent over highest correlating ground truth factor. Right two columns, model latent over second highest correlating ground truth factor. The color-coding corresponds to the six different object shapes (GT Shape) in the dataset.

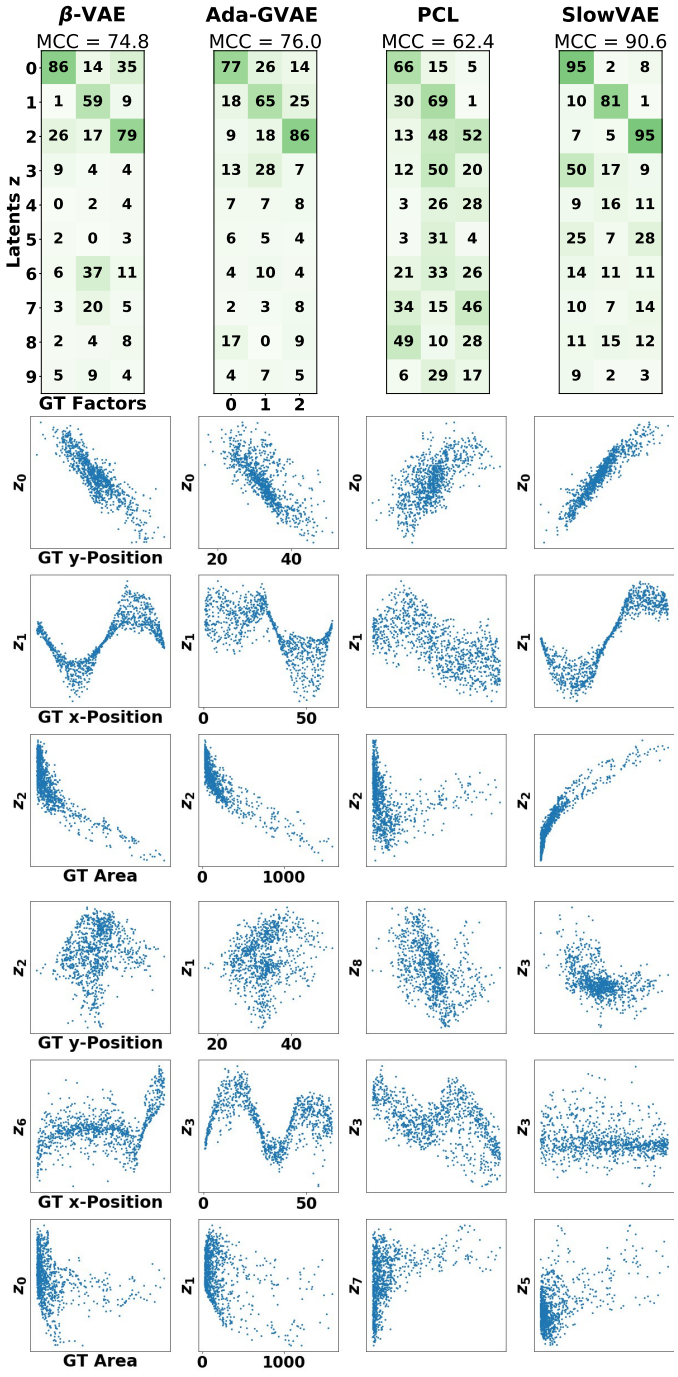


Figure 20: **KITTI Masks Latent Representations.** Top, MCC correlation matrices. Middle three rows, model latent over highest correlating ground truth factor. Bottom three rows, model latent over second highest correlating ground truth factor.

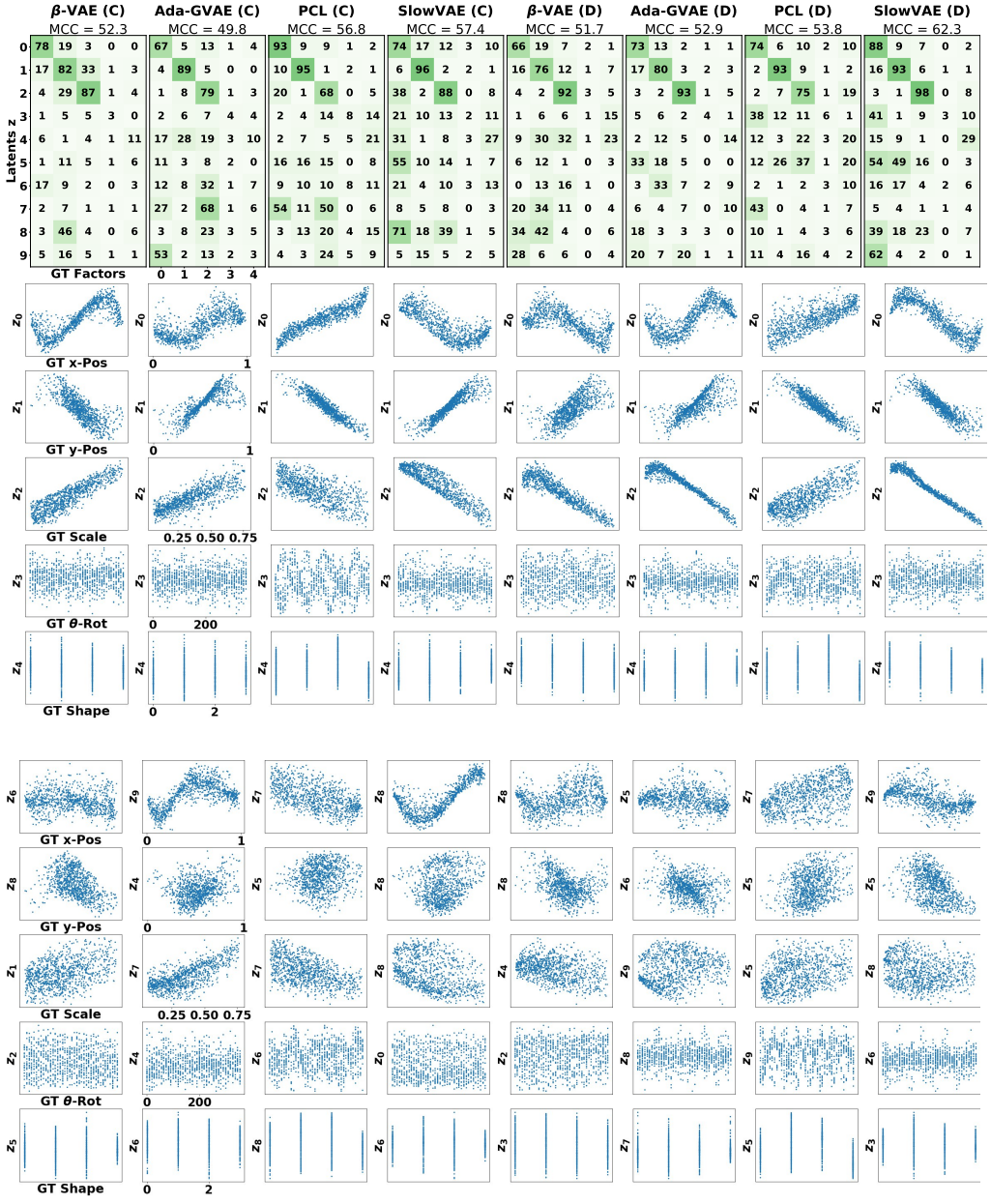


Figure 21: **Natural Sprites Latent Representations.** Top, MCC correlation matrices. Middle five rows, model latent over highest correlating ground truth factor (colored by category). Bottom five rows, model latent over second highest correlating ground truth factor. The left two columns denote the continuous (C) version of Natural Sprites, whereas the right two columns correspond to the discretized (D) version.

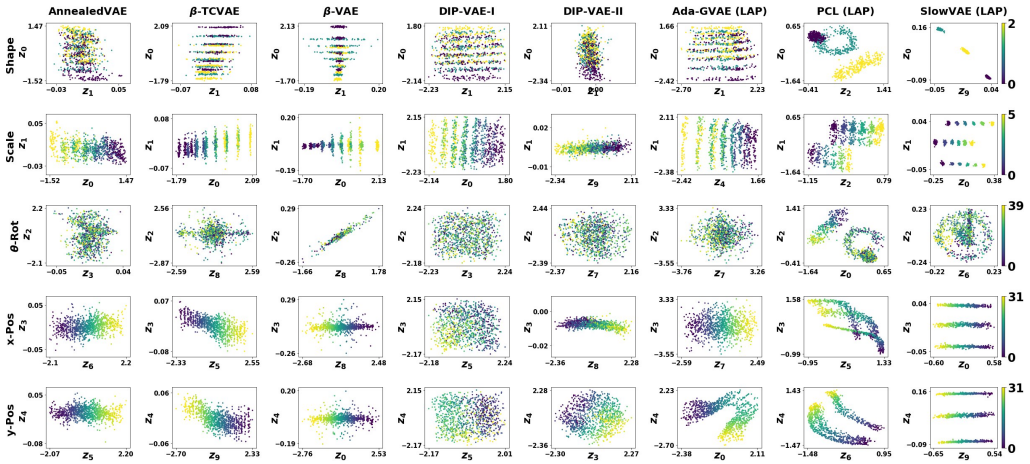


Figure 22: **DSprites Latent Representations**. Best two latents selected from Fig 15. Color-coded by the corresponding ground truth factor.

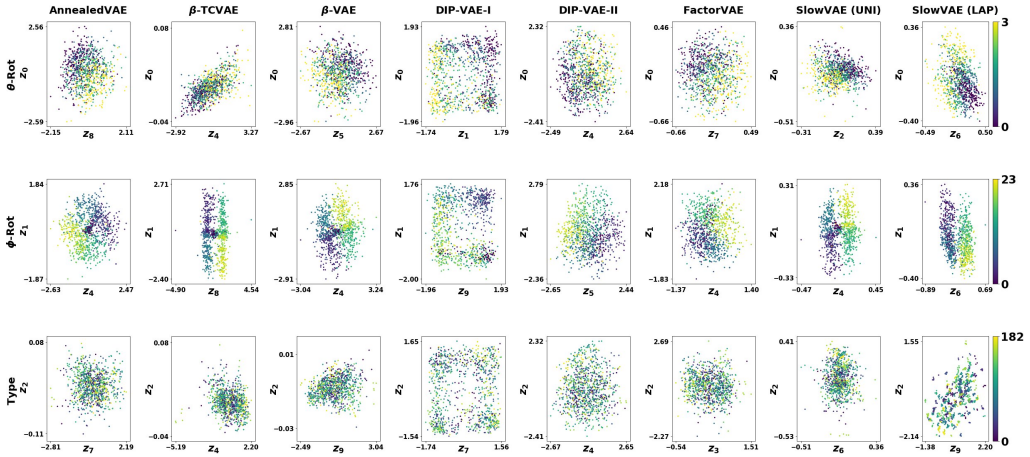


Figure 23: **Cars3D Latent Representations**. Best two latents selected from Fig 16. Color-coded by ground truth.

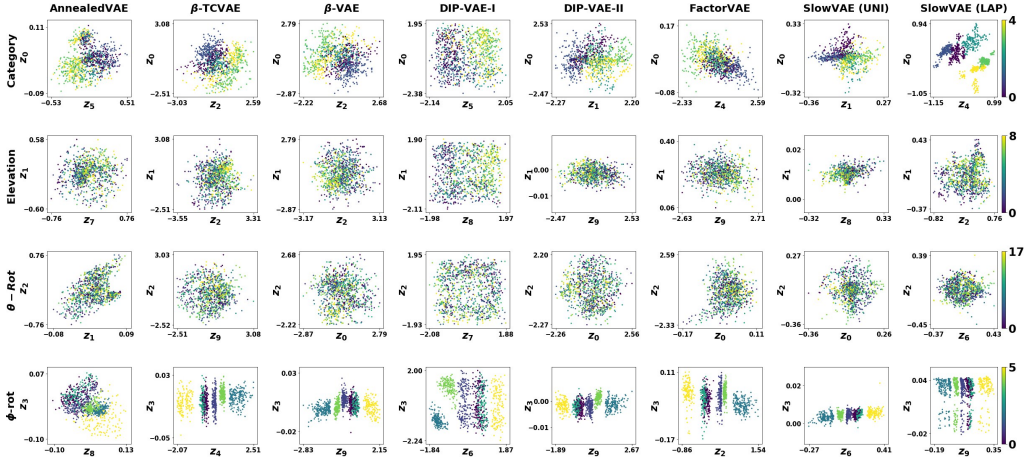


Figure 24: **SmallNorb Latent Representations**. Best two latents selected from Fig 17. Color-coded by ground truth.

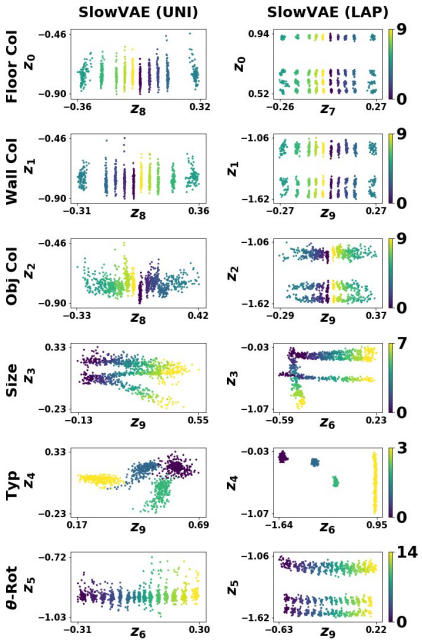


Figure 25: **Shapes3D Latent Representations**. Best two latents selected from Fig 18. Color-coded by ground truth.

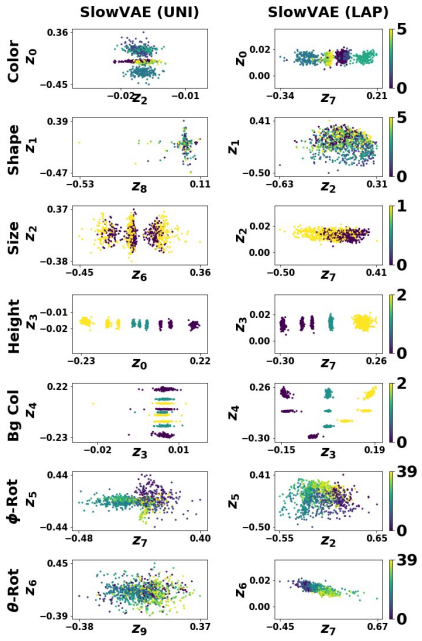


Figure 26: **MPI3DReal Latent Representations**. Best two latents selected from Fig 19. Color-coded by ground truth.

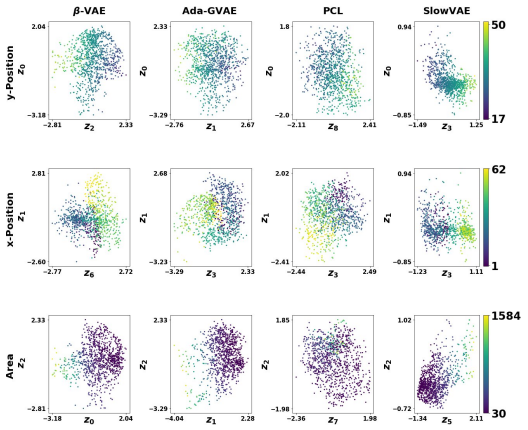


Figure 27: **KITTI Masks Latent Representations**. Best two latents selected from Fig 20. Color-coded by ground truth.

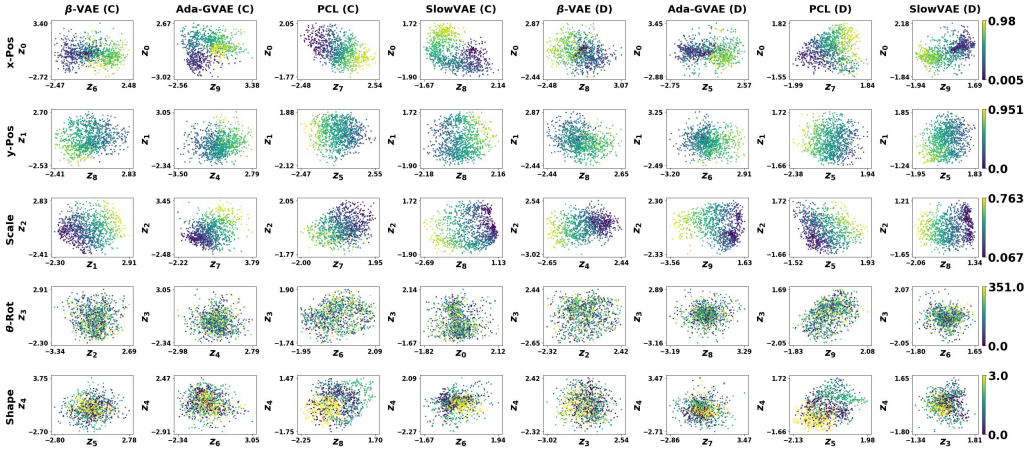


Figure 28: **Natural Sprites Latent Representations**. Best two latents selected from Fig 21. The left four columns denote the continuous (C) version of Natural Sprites, whereas the right four columns correspond to the discretized (D) version. Color-coded by ground truth.

Appendix B

Publication 2:
Visual representation learning does not generalize strongly within the same domain

Published as a conference paper and as a poster at the ICLR 2022.

VISUAL REPRESENTATION LEARNING DOES NOT GENERALIZE STRONGLY WITHIN THE SAME DOMAIN

Lukas Schott[1,‡], **Julius von Kügelgen**[2,3,4], **Frederik Träuble**[2,4],
Peter Gehler[4], **Chris Russell**[4], **Matthias Bethge**[1,4], **Bernhard Schölkopf**[2,4],
Francesco Locatello[4,†], **Wieland Brendel**[1,†]
[1]University of Tübingen, [2]Max Planck Institute for Intelligent Systems, Tübingen
[3]University of Cambridge, [4]Amazon Web Services
[†]Joint senior authors, [‡]Work done during an internship at Amazon
lukas.schott@bethgelab.org

ABSTRACT

An important component for generalization in machine learning is to uncover underlying latent factors of variation as well as the mechanism through which each factor acts in the world. In this paper, we test whether 17 unsupervised, weakly supervised, and fully supervised representation learning approaches correctly infer the generative factors of variation in simple datasets (dSprites, Shapes3D, MPI3D) from controlled environments, and on our contributed CelebGlow dataset. In contrast to prior robustness work that introduces novel factors of variation during test time, such as blur or other (un)structured noise, we here recompose, interpolate, or extrapolate only existing factors of variation from the training data set (e.g., small and medium-sized objects during training and large objects during testing). Models that learn the correct mechanism should be able to generalize to this benchmark. In total, we train and test 2000+ models and observe that all of them struggle to learn the underlying mechanism regardless of supervision signal and architectural bias. Moreover, the generalization capabilities of all tested models drop significantly as we move from artificial datasets towards more realistic real-world datasets. Despite their inability to identify the correct mechanism, the models are quite modular as their ability to infer other in-distribution factors remains fairly stable, providing only a single factor is out-of-distribution. These results point to an important yet understudied problem of learning mechanistic models of observations that can facilitate generalization.

1 INTRODUCTION

Humans excel at learning underlying physical mechanisms or inner workings of a system from observations (Funke et al., 2021; Barrett et al., 2018; Santoro et al., 2017; Villalobos et al., 2020; Spelke, 1990), which helps them generalize quickly to new situations and to learn efficiently from little data (Battaglia et al., 2013; Dehaene, 2020; Lake et al., 2017; Téglás et al., 2011). In contrast, machine learning systems typically require large amounts of curated data and still mostly fail to generalize to out-of-distribution (OOD) scenarios (Schölkopf et al., 2021; Hendrycks & Dietterich, 2019; Karahan et al., 2016; Michaelis et al., 2019; Roy et al., 2018; Azulay & Weiss, 2019; Barbu et al., 2019). It has been hypothesized that this failure of machine learning systems is due to shortcut learning (Kilbertus* et al., 2018; Ilyas et al., 2019; Geirhos et al., 2020; Schölkopf et al., 2021). In essence, machines seemingly learn to solve the tasks they have been trained on using auxiliary and spurious statistical relationships in the data, rather than true mechanistic relationships. Pragmatically, models relying on statistical relationships tend to fail if tested outside their training distribution, while models relying on (approximately) the true underlying mechanisms tend to generalize well to novel scenarios (Barrett et al., 2018; Funke et al., 2021; Wu et al., 2019; Zhang et al., 2018; Parascandolo et al., 2018; Schölkopf et al., 2021; Locatello et al., 2020a;b). To learn effective statistical relationships, the training data needs to cover most combinations of factors of variation (like shape, size, color, viewpoint, etc.). Unfortunately, the number of combinations scales exponentially with the number of factors. In contrast, learning the underlying mechanisms behind the factors of variation should greatly reduce the need for training data and scale more gently with the number of factors (Schölkopf et al., 2021; Peters et al., 2017; Besserve et al., 2021).

Benchmark: Our goal is to quantify how well machine learning models already learn the mechanisms underlying a data generative process. To this end, we consider four image data sets where each image is described by a small number of independently controllable factors of variation such

as scale, color, or size. We split the training and test data such that models that learned the underlying mechanisms should generalize to the test data. More precisely, we propose several systematic out-of-distribution (OOD) test splits like composition (e.g., *train* = small hearts, large squares $\rightarrow$ *test* = small squares, large hearts), interpolation (e.g., small hearts, large hearts $\rightarrow$ medium hearts) and extrapolation (e.g., small hearts, medium hearts $\rightarrow$ large hearts). While the factors of variation are independently controllable (e.g., there may exist large and small hearts), the observations may exhibit spurious statistical dependencies (e.g., observed hearts are typically small, but size may not be predictive at test time). Based on this setup, we benchmark 17 representation learning approaches and study their inductive biases. The considered approaches stem from un-/weakly supervised disentanglement, supervised learning, and the transfer learning literature.

Results: Our benchmark results indicate that the tested models mostly struggle to learn the underlying mechanisms regardless of supervision signal and architecture. As soon as a factor of variation is outside the training distribution, models consistently tend to predict a value in the previously observed range. On the other hand, these models can be fairly modular in the sense that predictions of in-distribution factors remain accurate, which is in part against common criticisms of deep neural networks (Greff et al., 2020; Csordás et al., 2021; Marcus, 2018; Lake & Baroni, 2018).

New Dataset: Previous datasets with independent controllable factors such as dSprites, Shapes3D, and MPI3D (Matthey et al., 2017; Kim & Mnih, 2018; Gondal et al., 2019) stem from highly structured environments. For these datasets, common factors of variations are scaling, rotation and simple geometrical shapes. We introduce a dataset derived from celebrity faces, named CelebGlow, with factors of variations such as smiling, age and hair-color. It also contains all possible factor combinations. It is based on latent traversals of a pretrained Glow network provided by Kingma et al. (Kingma & Dhariwal, 2018) and the Celeb-HQ dataset (Liu et al., 2015).

We hope that this benchmark can guide future efforts to find machine learning models capable of understanding the true underlying mechanisms in the data. To this end, all data sets and evaluation scripts are released alongside a leaderboard on GitHub. [1]

2 Problem Setting

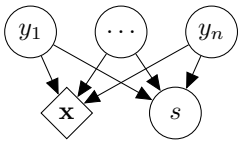


Figure 1: Assumed graphical model connecting the *factors of variations* $\mathbf{y} = (y_1, ..., y_n)$ to *observations* $\mathbf{x} = g(\mathbf{y})$. The *selection* variable $s \in \{\text{tr}, \text{te}\}$ leads to different train and test splits $p_s(\mathbf{y})$, thereby inducing correlation between the FoVs.

Assume that we render each observation or image $\mathbf{x} \in \mathbb{R}^d$ using a "computer graphic model" which takes as input a set of independently controllable factors of variation (FoVs) $\mathbf{y} \in \mathbb{R}^n$ like size or color. More formally, we assume a generative process of the form $\mathbf{x} = g(\mathbf{y})$, where $g : \mathbb{R}^n \mapsto \mathbb{R}^d$ is an injective and smooth function. In the standard independently and identically distributed (IID) setting, we would generate the training and test data in the same way, i.e., we would draw $\mathbf{y}$ from the same prior distribution $p(\mathbf{y})$ and then generate the corresponding images $\mathbf{x}$ according to $g(\cdot)$. Instead, we here consider an OOD setting where the prior distribution $p_{\text{tr}}(\mathbf{y})$ during training is different from the prior distribution $p_{\text{te}}(\mathbf{y})$ during testing. In fact, in all settings of our benchmark, the training and test distributions are completely disjoint, meaning that each point can only have non-zero probability mass in either $p_{\text{tr}}(\mathbf{y})$ or $p_{\text{te}}(\mathbf{y})$. Crucially, however, the function g which maps between FoVs and observations is shared between training and testing, which is why we refer to it as an *invariant mechanism*. As shown in the causal graphical model in Fig. 1, the factors of variations $\mathbf{y}$ are independently controllable to begin with, but the binary split variable s introduces spurious correlations between the FoVs that are different at training and test time as a result of selection bias (Storkey, 2009; Bareinboim & Pearl, 2012). In particular, we consider *Random*, *Composition*, *Interpolation*, and *Extrapolation* splits as illustrated in Fig. 2. We refer to §4.2 for details on the implementation of these splits.

The task for our machine learning models f is to estimate the factors of variations $\mathbf{y}$ that generated the sample $\mathbf{x}$ on both the training and test data. In other words, we want that (ideally) $f = g^{-1}$. The main challenge is that, during training, we only observe data from p_{tr} but wish to generalize to p_{te}. Hence, the learned function f should not only invert g locally on the training domain $\text{supp}(p_{\text{tr}}(\mathbf{y})) \subseteq \mathbb{R}^n$ but ideally globally. In practice, let $\mathcal{D}_{\text{te}} = \{(\mathbf{y}^k, \mathbf{x}^k)\}$ be the test data with $\mathbf{y}_k$ drawn from $p_{\text{te}}(\mathbf{y})$ and let $f : \mathbb{R}^d \mapsto \mathbb{R}^n$ be the model. Now, the goal is to design and optimize the

[1] https://github.com/bethgelab/InDomainGeneralizationBenchmark

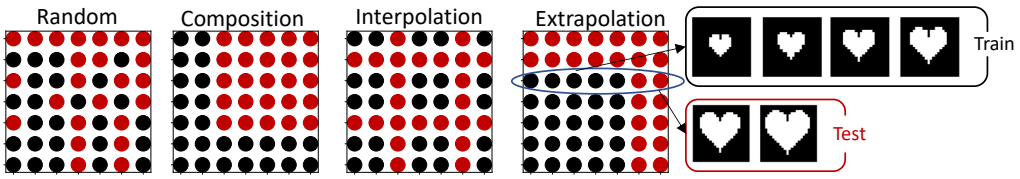


Figure 2: **Systematic test and train splits for two factors of variation.** Black dots correspond to the training and red dots to the test distribution. Examples of the corresponding observations are shown on the right.

model f on the training set $\mathcal{D}_{\text{tr}}$ such that it achieves a minimal R-squared distance between $\mathbf{y}^k$ and $f(\mathbf{x}^k)$ on the test set $\mathcal{D}_{\text{te}}$.

During training, models are allowed to sample the data from all non-zero probability regions $\text{supp}(p_{\text{tr}}(\mathbf{y}))$ in whatever way is optimal for its learning algorithm. This general formulation covers different scenarios and learning methods that could prove valuable for learning independent mechanisms. For example, supervised methods will sample an IID data set $\mathcal{D}_{\text{tr}} = \{(\mathbf{y}^k, \mathbf{x}^k)\}$ with $\mathbf{y}^k \sim p_{\text{tr}}(\mathbf{y})$, while self-supervised methods might sample a data set of unlabeled image pairs $\mathcal{D}_{\text{tr}} = \{(\mathbf{x}^k, \tilde{\mathbf{x}}^k)\}$. We aim to understand what inductive biases help on these OOD settings and how to best leverage the training data to learn representations that generalize.

3 Inductive biases for generalization in visual representation learning

We now explore different types of assumptions, or *inductive biases*, on the representational format (§3.1), architecture (§3.2), and dataset (§3.3) which have been proposed and used in the past to facilitate generalization. Inductive inference and the generalization of empirical findings is a fundamental problem of science that has a long-standing history in many disciplines. Notable examples include Occam's razor, Solomonoff's inductive inference (Solomonoff, 1964), Kolmogorov complexity (Kolmogorov, 1998), the bias-variance-tradeoff (Kohavi et al., 1996; Von Luxburg & Schölkopf, 2011), and the no free lunch theorem (Wolpert, 1996; Wolpert & Macready, 1997). In the context of statistical learning, Vapnik and Chervonenkis (Vapnik & Chervonenkis, 1982; Vapnik, 1995) showed that generalizing from a sample to its population (i.e., IID generalization) requires restricting the capacity of the class of candidate functions—a type of inductive bias. Since shifts between train and test distributions violate the IID assumption, however, statistical learning theory does not directly apply to our types of OOD generalization.

OOD generalization across different (e.g., observational and experimental) conditions also bears connections to causal inference (Pearl, 2009; Peters et al., 2017; Hernán & Robins, 2020). Typically, a causal graph encodes assumptions about the relation between different distributions and is used to decide how to "transport" a learned model (Pearl & Bareinboim, 2011; Pearl et al., 2014; Bareinboim & Pearl, 2016; von Kügelgen et al., 2019). Other approaches aim to learn a model which leads to invariant prediction across multiple environments (Schölkopf et al., 2012; Peters et al., 2016; Heinze-Deml et al., 2018; Rojas-Carulla et al., 2018; Arjovsky et al., 2019; Lu et al., 2021). However, these methods either consider a small number of causally meaningful variables in combination with domain knowledge, or assume access to data from multiple environments. In our setting, on the other hand, we aim to learn from higher-dimensional observations and to generalize from a single training set to a different test environment.

Our work focuses on OOD generalization in the context of visual representation learning, where deep learning has excelled over traditional learning approaches (Krizhevsky et al., 2012; LeCun et al., 2015; Schmidhuber, 2015; Goodfellow et al., 2016). In the following, we therefore concentrate on inductive biases specific to deep neural networks (Goyal & Bengio, 2020) on visual data. For details regarding specific objective functions, architectures, and training, we refer to the supplement.

3.1 Inductive bias 1: representational format

Learning useful representations of high-dimensional data is clearly important for the downstream performance of machine learning models (Bengio et al., 2013). The first type of inductive bias we consider is therefore the *representational format*. A common approach to representation learning is to postulate *independent latent variables* which give rise to the data, and try to infer these in an *unsupervised* fashion. This is the idea behind independent component analysis (ICA) (Comon,

1994; Hyvärinen & Oja, 2000) and has also been studied under the term *disentanglement* (Bengio et al., 2013). Most recent approaches learn a deep generative model based on the variational auto-encoder (VAE) framework (Kingma & Welling, 2013; Rezende et al., 2014), typically by adding regularization terms to the objective which further encourage independence between latents (Higgins et al., 2017; Kim & Mnih, 2018; Chen et al., 2018; Kumar et al., 2018; Burgess et al., 2018).

It is well known that ICA/disentanglement is theoretically non-identifiable without additional assumptions or supervision (Hyvärinen & Pajunen, 1999; Locatello et al., 2018). Recent work has thus focused on *weakly supervised* approaches which can provably identify the true independent latent factors (Hyvärinen & Morioka, 2016; Hyvarinen & Morioka, 2017; Shu et al., 2019; Locatello et al., 2020a; Klindt et al., 2020; Khemakhem et al., 2020; Roeder et al., 2020). The general idea is to leverage additional information in the form of paired observations $(\mathbf{x}^i, \tilde{\mathbf{x}}^i)$ where $\tilde{\mathbf{x}}^i$ is typically an auxiliary variable (e.g., an environment indicator or time-stamp) or a second view, i.e., $\tilde{\mathbf{x}}^i = g(\tilde{\mathbf{y}}^i)$ with $\tilde{\mathbf{y}}^i \sim p(\tilde{\mathbf{y}}|\mathbf{y}^i)$, where $\mathbf{y}^i$ are the FoVs of $\mathbf{x}^i$ and $p(\tilde{\mathbf{y}}|\mathbf{y})$ depends on the method. We remark that such identifiability guarantees only hold for the training distribution (and given infinite data), and thus may break down once we move to a different distribution for testing. In practice, however, we hope that the identifiability of the representation translates to learning mechanisms that generalize.

In our study, we consider the popular β-VAE (Higgins et al., 2017) as an unsupervised approach, as well as Ada-GVAE (Locatello et al., 2020a), Slow-VAE (Klindt et al., 2020) and PCL (Hyvarinen & Morioka, 2017) as weakly supervised disentanglement methods. First, we learn a representation $\mathbf{z} \in \mathbb{R}^n$ given only (pairs of) observations (i.e., without access to the FoVs) using an encoder $f_{\text{enc}} : \mathbb{R}^d \to \mathbb{R}^n$. We then freeze the encoder (and thus the learned representation $\mathbf{z}$) and train a multi-layer perceptron (MLP) $f_{\text{MLP}} : \mathbb{R}^n \to \mathbb{R}^n$ to predict the FoVs $\mathbf{y}$ from $\mathbf{z}$ in a supervised way. The learned inverse mechanism f in this case is thus given by $f = f_{\text{MLP}} \circ f_{\text{enc}}$.

3.2 Inductive bias 2: architectural (supervised learning)

The physical world is governed by symmetries (Nother, 1915), and enforcing appropriate task-dependent symmetries in our function class may facilitate more efficient learning and generalization. The second type of inductive bias we consider thus regards properties of the learned regression function, which we refer to as *architectural bias*. Of central importance are the concepts of *invariance* (changes in input should not lead to changes in output) and *equivariance* (changes in input should lead to proportional changes in output). In vision tasks, for example, object *localization* exhibits *equivariance* to translation, whereas object *classification* exhibits *invariance* to translation. E.g., translating an object in an input image should lead to an equal shift in the predicted bounding box (equivariance), but should not affect the predicted object class (invariance).

A famous example is the convolution operation which yields translation equivariance and forms the basis of convolutional neural networks (CNNs) (Le Cun et al., 1989; LeCun et al., 1989). Combined with a set operation such as pooling, CNNs then achieve translation invariance. More recently, the idea of building equivariance properties into neural architectures has also been successfully applied to more general transformations such as rotation and scale (Cohen & Welling, 2016; Cohen et al., 2019; Weiler & Cesa, 2019) or (coordinate) permutations (Zhang et al., 2019; Achlioptas et al., 2018). Other approaches consider affine transformations (Jaderberg et al., 2015), allow to trade off invariance vs dependence on coordinates (Liu et al., 2018), or use residual blocks and skip connections to promote feature re-use and facilitate more efficient gradient computation (He et al., 2016; Huang et al., 2017). While powerful in principle, a key challenge is that relevant equivariances for a given problem may be unknown a priori or hard to enforce architecturally. E.g., 3D rotational equivariance is not easily captured for 2D-projected images, as for the MPI3D data set.

In our study, we consider the following architectures: standard MLPs and CNNs, CoordConv (Liu et al., 2018) and coordinate-based (Sitzmann et al., 2020) nets, Rotationally-Equivariant (Rotation-EQ) CNNs (Cohen & Welling, 2016), Spatial Transformers (STN) (Jaderberg et al., 2015), ResNet (RN) 50 and 101 (He et al., 2016), and DenseNet (Huang et al., 2017). All networks f are trained to directly predict the FoVs $\mathbf{y} \approx f(\mathbf{x})$ in a purely supervised fashion.

3.3 Inductive bias 3: leveraging additional data (transfer learning)

The physical world is modular: many patterns and structures reoccur across a variety of settings. Thus, the third and final type of inductive bias we consider is leveraging additional data through transfer learning. Especially in vision, it has been found that low-level features such as edges or

simple textures are consistently learned in the first layers of neural networks, which suggests their usefulness across a wide range of tasks (Sun et al., 2017). State-of-the-art approaches therefore often rely on pre-training on enormous image corpora prior to fine-tuning on data from the target task (Kolesnikov et al., 2020; Mahajan et al., 2018; Xie et al., 2020). The guiding intuition is that additional data helps to learn common features and symmetries and thus enables a more efficient use of the (typically small amount of) labeled training data. Leveraging additional data as an inductive bias is connected to the representational format §3.1 as they are often combined during pre-training.

In our study, we consider three pre-trained models: RN-50 and RN-101 pretrained on ImageNet-21k (Deng et al., 2009; Kolesnikov et al., 2020) and a DenseNet pretrained on ImageNet-1k (ILSVRC) (Russakovsky et al., 2015). We replace the last layer with a randomly initialized readout layer chosen to match the dimension of the FoVs of a given dataset and fine-tune the whole network for 50,000 iterations on the respective train splits.

4 Experimental Setup

4.1 Datasets

We consider datasets with images generated from a set of discrete Factors of Variation (FoVs) following a deterministic generative model. All selected datasets are designed such that all possible combinations of factors of variation are realized in a corresponding image. *dSprites* (Matthey et al., 2017), is composed of low resolution binary images of basic shapes with 5 FoVs: shape, scale, orientation, x-position, and y-position. Next, *Shapes3D* (Kim & Mnih, 2018), a popular dataset with 3D shapes in a room with 6 FoVs: floor, color, wall color, object color, object size, object type, and camera azimuth. Furthermore, with *CelebGlow* we introduce a novel dataset that has more natural factors of variations such as smiling, hair-color and age. For more details and samples, we refer to Appendix B. Lastly, we consider the challenging and realistic *MPI3D* (Gondal et al., 2019), which contains real images of physical 3D objects attached to a robotic finger generated with 7 FoVs: color, shape, size, height, background color, x-axis, and y-axis. For more details, we refer to Appendix H.1.

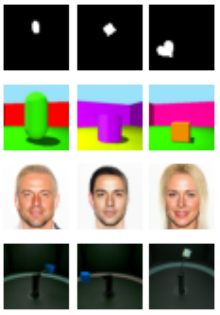


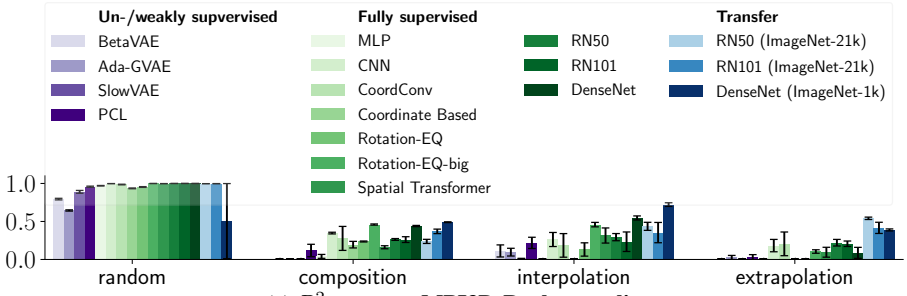
Figure 3: Random dataset samples from dSprites (1st), Shapes3D (2nd), CelebGlow (3rd), and MPI3D-real (4th).

4.2 Splits

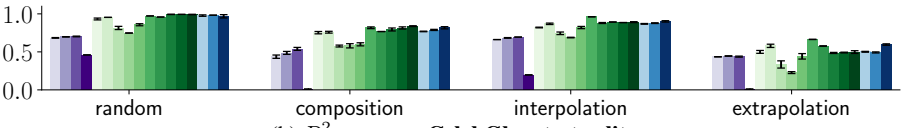
For each of the above datasets, denoted by $\mathcal{D}$, we create disjoint splits of train sets $\mathcal{D}_{\text{tr}}$ and test sets $\mathcal{D}_{\text{te}}$. We systematically construct the splits according to the underlying factors to evaluate different modalities of generalization, which we refer to as *composition*, *interpolation*, *extrapolation*, and *random*. See Fig. 2 for a visual presentation of such splits regarding two factors.

Composition: We exclude all images from the train split if factors are located in a particular corner of the FoV hyper cube given by all FoVs. This means certain systematic combinations of FoVs are never seen during training even though the value of each factor is individually present in the train set. The related test split then represents images of which at least two factors resemble such an unseen composition of factor values, thus testing generalization w.r.t. composition.
Interpolation: Within the range of values of each FoV, we periodically exclude values from the train split. The corresponding test split then represents images of which at least one factor takes one of the unseen factor values in between, thus testing generalization w.r.t. interpolation.
Extrapolation: We exclude all combinations having factors with values above a certain label threshold from the train split. The corresponding test split then represents images with one or more extrapolated factor values, thus testing generalization w.r.t. extrapolation.
Random: Lastly, as a baseline to test our models performances across the full dataset in distribution, we cover the case of an IID sampled train and test set split from D. Compared to inter- and extrapolation where factors are systematically excluded, here it is very likely that all individual factor values have been observed in a some combination.

We further control all considered splits and datasets such that $\sim 30\%$ of the available data is in the training set $\mathcal{D}_{\text{tr}}$ and the remaining $\sim 70\%$ belong to the test set $\mathcal{D}_{\text{te}}$. Lastly, we do not split along factors of variation if no intuitive order exists. Therefore, we do not split along the categorical variable *shape* and along the axis of factors where only two values are available.



(a) R^2**-score on MPI3D-Real test splits.**



(b) R^2**-score on CelebGlow test splits.**

Figure 4: R^2**-score on various test-train splits.** Compared to the in-distribution random splits, on the out-of-distribution (OOD) splits composition, interpolation, and extrapolation, we observe large drops in performance.

4.3 Evaluation

To benchmark the generalization capabilities, we compute the R^2-score, the coefficient of determination, on the respective test set. We define the R^2-score based on the MSE score per FoV y_j

$$R_i^2 = 1 - \frac{\text{MSE}_i}{\sigma_i^2} \quad \text{with} \quad \text{MSE}_j = \mathbb{E}_{(\mathbf{x},\mathbf{y})\in D_{\text{te}}}\left[(\mathbf{y}_j - f_j(\mathbf{x}))^2\right], \tag{1}$$

where σ_i^2 is the variance per factor defined on the full dataset D. Under this score, $R_i^2 = 1$ can be interpreted as perfect regression and prediction under the respective test set whereas $R_i^2 = 0$ indicates random guessing with the MSE being identical to the variance per factor. For visualization purposes, we clip the R^2 to 0 if it is negative. We provide all unclipped values in the Appendix.

5 Experiments and Results

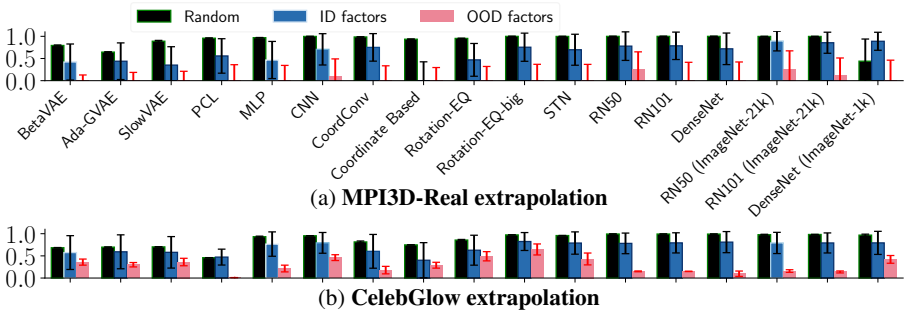
Our goal is to investigate how different visual representation models perform on our proposed systematic out-of-distribution (OOD) test sets. We consider un-/weakly supervised, fully supervised, and transfer learning models. We focus our conclusions on MPI3D-Real as it is the most realistic dataset. Further results on dSprites and Shapes3D are, however, mostly consistent.

In the first subsection, §5.1, we investigate the overall model OOD performance. In Sections 5.2 and 5.3, we focus on a more in-depth error analysis by controlling the splits s.t. only a single factor is OOD during testing. Lastly, in §5.4, we investigate the connection between the degree of disentanglement and downstream performance.

5.1 Model performance decreases on OOD test splits

In Fig. 4 and Appendix Fig. 11, we plot the performance of each model across different generalization settings. Compared to the in-distribution (ID) setting (random), we observe large drops in performance when evaluating our OOD test sets on all considered datasets. This effect is most prominent on MPI3D-Real. Here, we further see that, on average, the performances seem to increase as we increase the supervision signal (comparing RN50, RN101, DenseNet with and without additional data on MPI3D). On CelebGlow, models also struggle to extrapolate. However, the results on composition and interpolation only drop slightly compared to the random split.

For Shapes3D (shown in the Appendix E), the OOD generalization is partially successful, especially in the composition and interpolation settings. We hypothesize that this is due to the dataset specific, fixed spatial composition of the images. For instance, with the object-centric positioning, the floor, wall and other factors are mostly at the same position within the images. Thus, they can reliably be inferred by only looking at a certain fixed spot in the image. In contrast, for MPI3D this is more difficult as, e.g., the robot finger has to be found to infer its tip color. Furthermore, the factors of variation in Shapes3D mostly consist of colors which are encoded within the same input dimensions,



(a) **MPI3D-Real extrapolation**

(b) **CelebGlow extrapolation**

Figure 5: **Extrapolation and modularity, R^2-score on subsets.** In the extrapolation setting, we further differentiate between factors that have been observed during training (ID factors) and extrapolated values (OOD factors) and measure the performances separately. As a reference, we compare to a random split. A model is considered modular, if it still infers ID factors correctly despite other factors being OOD.

and not across pixels as, for instance, x-translation in MPI3D. For this color interpolation, the ReLU activation function might be a good inductive bias for generalization. However, it is not sufficient to achieve extrapolations, as we still observe a large drop in performance here.

Conclusion: The performance generally decreases when factors are OOD regardless of the supervision signal and architecture. However, we also observed exceptions in Shapes3D where OOD generalization was largely successful except for extrapolation.

5.2 Errors stem from inferring OOD factors

While in the previous section we observed a general decrease in R^2 score for the interpolation and extrapolation splits, our evaluation does not yet show how errors are distributed among individual factors that are in- and out-of-distribution.

In contrast to the previous section where *multiple* factors could be OOD distribution simultaneously, here, we control data splits (Fig. 2) interpolation, extrapolation s.t. only a *single* factor is OOD. Now, we also estimate the R^2-score separately per factor, depending on whether they have individually been observed during training (ID factor) or are exclusively in the test set (OOD factor). For instance, if we only have images of a heart with varying scale and position, we query the model with hearts at larger scales than observed during training (OOD factor), but at a previously observed position (ID factor). For a formal description, see Appendix Appendix H.2. This controlled setup enables us to investigate the modularity of the tested models, as we can separately measure the performance on OOD and ID factors. As a reference for an approximate upper bound, we additionally report the performance of the model on a random train/test split.

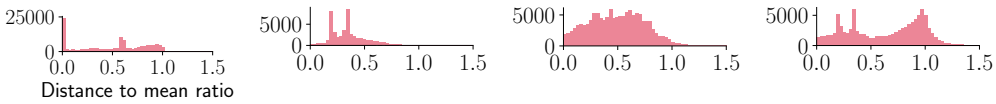
In Figs. 5 and 14, we observe significant drops in performance for the OOD factors compared to a random test-train split. In contrast, for the ID factors, we see that the models still perform close to the random split, although with much larger variance. For the interpolation setting (Appendix Fig. 14), this drop is also observed for MPI3D and dSprites but not for Shapes3D. Here, OOD and ID are almost on par with the random split. Note that our notion of modularity is based on systematic splits of individual factors and the resulting outputs. Other works focus on the inner behavior of a model by, e.g., investigating the clustering of neurons within the network (Filan et al., 2021). Preliminary experiments showed no correlations between the different notions of modularity.

Conclusion: The tested models can be fairly modular, in the sense that the predictions of ID factors remain accurate. The low OOD performances mainly stem from incorrectly extrapolated or interpolated factors. Given the low inter-/extrapolation (i.e., OOD) performances on MPI3D and dSprites, evidently no model learned to invert the ground-truth generative mechanism.

5.3 Models extrapolate similarly and towards the mean

In the previous sections, we observed that our tested models specifically extrapolate poorly on OOD factors. Here, we focus on quantifying the behavior of how different models extrapolate.

To check whether different models make similar errors, we compare the extrapolation behavior across architectures and seeds by measuring the similarity of model predictions for the OOD factors described in the previous section. No model is compared to itself if it has the same random



(a) **MPI3D-Real** (b) **CelebGlow** (c) **Shapes3D** (d) **dSprites**

Figure 6: **Extrapolation towards the mean.** We calculate (2) on the extrapolated OOD factors to measure the closeness towards the mean compared to the ground-truth. Here, the values are mostly in $[0, 1]$. Thus, models tend to predict values in previously observed ranges.

seed. On MPI3D, Shapes3D and dSprites, all models strongly correlate with each other (Pearson $\rho \geq 0.57$) but anti-correlate compared to the ground-truth prediction (Pearson $\rho \leq -0.48$), the overall similarity matrix is shown in Appendix Fig. 17. One notable exception is on CelebGlow. Here, some models show low but positive correlations with the ground truth generative model (Pearson $\rho \geq 0.57$). However, visually the models are still quite off as shown for the model with the highest correlation in Fig. 18. In most cases, the highest similarity is along the diagonal, which demonstrates the influence of the architectural bias. This result hints at all models making similar mistakes extrapolating a factor of variation.

We find that models collectively tend towards predicting the mean for each factor in the training distribution when extrapolating. To show this, we estimate the following ratio of distances

$$r = |f(\mathbf{x}^i)_j - \bar{\mathbf{y}}_j| \;/\; |\mathbf{y}^i_j - \bar{\mathbf{y}}_j|, \tag{2}$$

where $\bar{\mathbf{y}}_j = \frac{1}{n}\sum_{i=1}^{n} \mathbf{y}^i_j$ is the mean of FoV y_j. If values of (2) are $\in [0, 1]$, models predict values which are closer to the mean than the corresponding ground-truth. We show a histogram over all supervised and transfer-based models for each dataset in Fig. 6. Models tend towards predicting the mean as only few values are $>= 1$. This is shown qualitatively in Appendix Figs. 15 and 16.

Conclusion: Overall, we observe only small differences in *how* the tested models extrapolate, but a strong difference compared to the ground-truth. Instead of extrapolating, all models regress the OOD factor towards the mean in the training set. We hope that this observation can be considered to develop more diverse future models.

5.4 On the relation between disentanglement and downstream performance

Previous works have focused on the connection between disentanglement and OOD downstream performance (Träuble et al., 2020; Dittadi et al., 2020; Montero et al., 2021). Similarly, for our systematic splits, we measure the degree of disentanglement using the DCI-Disentanglement (Eastwood & Williams, 2018) score on the latent representation of the embedded test and train data. Subsequently, we correlate it with the R^2-performance of a supervised readout model which we report in §5.1. Note that the simplicity of the readout function depends on the degree of disentanglement, e.g., for a perfect disentanglement up to permutation and sign flips this would just be an assignment problem. For the disentanglement models, we consider the un-/ weakly supervised models β-VAE(Higgins et al., 2017), SlowVAE (Klindt et al., 2020), Ada-GVAE(Locatello et al., 2020a) and PCL (Hyvarinen & Morioka, 2017).

We find that the degree of downstream performance correlates positively with the degree of disentanglement (Pearson $\rho = 0.63$, Spearman $\rho = 0.67$). However, the correlations vary per dataset and split (see Appendix Fig. 7). Moreover, the overall performance of the disentanglement models followed by a supervised readout on the OOD split is lower compared to the supervised models (see e.g. Fig. 4). In an ablation study with an oracle embedding that disentangles the test data up to permutations and sign flips, we found perfect generalization capabilities ($R^2_{test} \geq 0.99$).

Conclusion: Disentanglement models show no improved performance in OOD generalization. Nevertheless, we observe a mostly positive correlation between the degree of disentanglement and the downstream performance.

6 Other related benchmark studies

In this section, we focus on related benchmarks and their conclusions. For related work in the context of inductive biases, we refer to §3.

Corruption benchmarks: Other current benchmarks focus on the performance of models when adding common corruptions (denoted by -C) such as noise or snow to current dataset test sets,

resulting in ImageNet-C, CIFAR-10-C, Pascal-C, Coco-C, Cityscapes-C and MNIST-C (Hendrycks & Dietterich, 2019; Michaelis et al., 2019; Mu & Gilmer, 2019). In contrast, in our benchmark, we assure that the factors of variations are present in the training set and merely have to be generalized correctly. In addition, our focus lies on identifying the ground truth generative process and its underlying factors. Depending on the task, the requirements for a model are very different. E.g., the ImageNet-C classification benchmark requires spatial invariance, whereas regressing factors such as, e.g., shift and shape of an object, requires in- and equivariance.

Abstract reasoning: Model performances on OOD generalizations are also intensively studied from the perspective of abstract reasoning, visual and relational reasoning tasks (Barrett et al., 2018; Wu et al., 2019; Santoro et al., 2017; Villalobos et al., 2020; Zhang et al., 2016; Yan & Zhou, 2017; Funke et al., 2021; Zhang et al., 2018). Most related, (Barrett et al., 2018; Wu et al., 2019) also study similar interpolation and extrapolation regimes. Despite using notably different tasks such as abstract or spatial reasoning, they arrive at similar conclusions: They also observe drops in performance in the generalization regime and that interpolation is, in general, easier than extrapolation, and also hint at the modularity of models using distractor symbols (Barrett et al., 2018). Lastly, posing the concept of using correct generalization as a necessary condition to check whether an underlying mechanism has been learned has also been proposed in (Wu et al., 2019; Zhang et al., 2018; Funke et al., 2021).

Disentangled representation learning: Close to our work, Montero et al. (Montero et al., 2021) also study generalization in the context of extrapolation, interpolation and a weak form of composition on dSprites and Shapes3D, but not the more difficult MPI3D-Dataset. They focus on reconstructions of unsupervised disentanglement algorithms and thus the *decoder*, a task known to be theoretically impossible(Locatello et al., 2018). In their setup, they show that OOD generalization is limited. From their work, it remains unclear whether the generalization along known factors is a general problem in visual representation learning, and how neural networks fail to generalize. We try to fill these gaps. Moreover, we focus on representation learning approaches and thus on the *encoder* and consider a broader variety of models, including theoretically identifiable approaches (Ada-GAVE, SlowVAE, PCL), and provide a thorough in-depth analysis of how networks generalize.

Previously, Träuble et al. (2020) studied the behavior of unsupervised disentanglement models on correlated training data. They find that despite disentanglement objectives, the learned latent spaces mirror this correlation structure. In line with our work, the results of their supervised post-hoc regression models on Shapes3D suggest similar generalization performances as we see in our respective disentanglement models in Figs. 4 and 11. OOD generalization w.r.t. extrapolation of one single FoV is also analyzed in (Dittadi et al., 2020). Our experimental setup in §5.4 is similar to their 'OOD2' scenario. Here, our results are in accordance, as we both find that the degree of disentanglement is lightly correlated with the downstream performance.

Others: To demonstrate shortcuts in neural networks, Eulig et al. (2021) introduce a benchmark with factors of variations such as color on MNIST that correlate with a specified task but control for those correlations during test-time. In the context of reinforcement learning, Packer et al. (2018) assess models on systematic test-train splits similar to our inter-/extrapolation and show that current models cannot solve this problem. For generative adversarial networks (GANs), it has also been shown that their learned representations do not extrapolate beyond the training data (Jahanian et al., 2019).

7 Discussion and Conclusion

In this paper, we highlight the importance of learning the independent underlying mechanisms behind the factors of variation present in the data to achieve generalization. However, we empirically show that among a large variety of models, no tested model succeeds in generalizing to all our proposed OOD settings (extrapolation, interpolation, composition). We conclude that the models are limited in learning the underlying mechanism behind the data and rather rely on strategies that do not generalize well. We further observe that while one factor is out-of-distribution, most other in-distribution factors are inferred correctly. In this sense, the tested models are surprisingly modular.

To further foster research on this intuitively simple, yet unsolved problem, we release our code as a benchmark. This benchmark, which allows various supervision types and systematic controls, should promote more principled approaches and can be seen as a more tractable intermediate milestone towards solving more general OOD benchmarks. In the future, a theoretical treatment identifying further inductive biases of the model and the necessary requirements of the data to solve our proposed benchmark should be further investigated.

ACKNOWLEDGEMENTS

The authors thank Steffen Schneider, Matthias Tangemann and Thomas Brox for their valuable feedback and fruitful discussions. The authors would also like to thank David Klindt, Judy Borowski, Dylan Paiton, Milton Montero and Sudhanshu Mittal for their constructive criticism of the manuscript. The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting FT and LS. We acknowledge support from the German Federal Ministry of Education and Research (BMBF) through the Competence Center for Machine Learning (TUE.AI, FKZ 01IS18039A) and the Bernstein Computational Neuroscience Program Tübingen (FKZ: 01GQ1002). WB acknowledges support via his Emmy Noether Research Group funded by the German Science Foundation (DFG) under grant no. BR 6382/1-1 as well as support by Open Philantropy and the Good Ventures Foundation. MB and WB acknowledge funding from the MICrONS program of the Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DoI/IBC) contract number D16PC00003.

REFERENCES

Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pp. 40–49. PMLR, 2018.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *Journal of Machine Learning Research*, 20(184):1–25, 2019.

Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In *Advances in Neural Information Processing Systems*, pp. 9448–9458, 2019.

Elias Bareinboim and Judea Pearl. Controlling selection bias in causal inference. In *Artificial Intelligence and Statistics*, pp. 100–108. PMLR, 2012.

Elias Bareinboim and Judea Pearl. Causal inference and the data-fusion problem. *Proceedings of the National Academy of Sciences*, 113(27):7345–7352, 2016.

David Barrett, Felix Hill, Adam Santoro, Ari Morcos, and Timothy Lillicrap. Measuring abstract reasoning in neural networks. In *International conference on machine learning*, pp. 511–520. PMLR, 2018.

Peter W Battaglia, Jessica B Hamrick, and Joshua B Tenenbaum. Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences*, 110(45):18327–18332, 2013.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

M. Besserve, R. Sun, D. Janzing, and B. Schölkopf. A theory of independent mechanisms for extrapolation in generative models. In *35th AAAI Conference on Artificial Intelligence: A Virtual Conference*, 2021.

Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-VAE. *arXiv preprint arXiv:1804.03599*, 2018.

Ricky TQ Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in vaes. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 2615–2625, 2018.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pp. 2990–2999. PMLR, 2016.

Taco Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. Gauge equivariant convolutional networks and the icosahedral cnn. In *International Conference on Machine Learning*, pp. 1321–1330. PMLR, 2019.

Pierre Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994.

Róbert Csordás, Sjoerd van Steenkiste, and Jürgen Schmidhuber. Are neural nets modular? inspecting functional modularity through differentiable weight masks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=7uVcpu-gMD.

Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D Hoffman, et al. Underspecification presents challenges for credibility in modern machine learning. *arXiv preprint arXiv:2011.03395*, 2020.

Stanislas Dehaene. *How We Learn: Why Brains Learn Better Than Any Machine... for Now*. Penguin, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Andrea Dittadi, Frederik Träuble, Francesco Locatello, Manuel Wüthrich, Vaibhav Agrawal, Ole Winther, Stefan Bauer, and Bernhard Schölkopf. On the transfer of disentangled representations in realistic settings. *arXiv preprint arXiv:2010.14407*, 2020.

Cian Eastwood and Christopher KI Williams. A framework for the quantitative evaluation of disentangled representations. In *In International Conference on Learning Representations*, 2018.

Elias Eulig, Piyapat Saranrittichai, Chaithanya Kumar Mummadi, Kilian Rambach, William Beluch, Xiahan Shi, and Volker Fischer. Diagvib-6: A diagnostic benchmark suite for vision models in the presence of shortcut and generalization opportunities. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10655–10664, 2021.

Muhammad Fahad, Arsalan Shahid, Ravi Reddy Manumachu, and Alexey Lastovetsky. A comparative study of methods for measurement of energy of computing. *Energies*, 12(11):2204, 2019.

Daniel Filan, Stephen Casper, Shlomi Hod, Cody Wild, Andrew Critch, and Stuart Russell. Clusterability in neural networks. *arXiv preprint arXiv:2103.03386*, 2021.

C. M. Funke, J. Borowski, K. Stosio, W. Brendel, T. S. A. Wallis, and M. Bethge. Five points to check when comparing visual perception in humans and machines. *Journal of Vision*, Mar 2021. URL https://jov.arvojournals.org/article.aspx?articleid=2772393.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11): 665–673, 2020.

Muhammad Waleed Gondal, Manuel Wuthrich, Djordje Miladinovic, Francesco Locatello, Martin Breidt, Valentin Volchkov, Joel Akpo, Olivier Bachem, Bernhard Schölkopf, and Stefan Bauer. On the transfer of inductive bias from simulation to the real world: a new disentanglement dataset. In *Advances in Neural Information Processing Systems*, pp. 15714–15725, 2019.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

Anirudh Goyal and Yoshua Bengio. Inductive biases for deep learning of higher-level cognition. *arXiv preprint arXiv:2011.15091*, 2020.

Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. On the binding problem in artificial neural networks. *arXiv preprint arXiv:2012.05208*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Christina Heinze-Deml, Jonas Peters, and Nicolai Meinshausen. Invariant causal prediction for nonlinear models. *Journal of Causal Inference*, 6(2), 2018.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

Miguel A Hernán and James M Robins. *Causal inference: what if*. Boca Raton: Chapman & Hall/CRC, 2020.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Aapo Hyvärinen and Hiroshi Morioka. Unsupervised feature extraction by time-contrastive learning and nonlinear ica. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 3772–3780, 2016.

Aapo Hyvarinen and Hiroshi Morioka. Nonlinear ica of temporally dependent stationary sources. In *Artificial Intelligence and Statistics*, pp. 460–469. PMLR, 2017.

Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430, 2000.

Aapo Hyvärinen and Petteri Pajunen. Nonlinear independent component analysis: Existence and uniqueness results. *Neural networks*, 12(3):429–439, 1999.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/e2c420d928d4bf8ce0ff2ec19b371514-Paper.pdf.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. *arXiv preprint arXiv:1506.02025*, 2015.

Ali Jahanian, Lucy Chai, and Phillip Isola. On the" steerability" of generative adversarial networks. *arXiv preprint arXiv:1907.07171*, 2019.

Samil Karahan, Merve Kilinc Yildirum, Kadir Kirtac, Ferhat Sukru Rende, Gultekin Butun, and Hazim Kemal Ekenel. How image degradations affect deep cnn-based face recognition? In *2016 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pp. 1–5. IEEE, 2016.

Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, pp. 2207–2217. PMLR, 2020.

N. Kilbertus*, G. Parascandolo*, and B. Schölkopf*. Generalization in anti-causal learning. In *NeurIPS 2018 Workshop on Critiquing and Correcting Trends in Machine Learning*, 2018. URL https://ml-critique-correct.github.io/. *authors are listed in alphabetical order.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pp. 2649–2658. PMLR, 2018.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/d139db6a236200b21cc7f752979132d0-Paper.pdf.

David Klindt, Lukas Schott, Yash Sharma, Ivan Ustyuzhaninov, Wieland Brendel, Matthias Bethge, and Dylan Paiton. Towards nonlinear disentanglement in natural data with temporal sparse coding. *arXiv preprint arXiv:2007.10930*, 2020.

Ron Kohavi, David H Wolpert, et al. Bias plus variance decomposition for zero-one loss functions. In *ICML*, volume 96, pp. 275–83, 1996.

Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pp. 491–507. Springer, 2020.

Andrei N. Kolmogorov. On tables of random numbers (reprinted from "sankhya: The indian journal of statistics", series a, vol. 25 part 4, 1963). *Theor. Comput. Sci.*, 207(2):387–395, 1998. URL http://dblp.uni-trier.de/db/journals/tcs/tcs207.html#Kolmogorov98.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. In *International Conference on Learning Representations*, 2018.

Brenden Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International Conference on Machine Learning*, pp. 2873–2882. PMLR, 2018.

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017.

Yann Le Cun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Proceedings of the 2nd International Conference on Neural Information Processing Systems*, pp. 396–404, 1989.

Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1 (4):541–551, 1989.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/60106888f8977b71e1f15db7bc9a88d1-Paper.pdf.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. *arXiv preprint arXiv:1811.12359*, 2018.

Francesco Locatello, Ben Poole, Gunnar Rätsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen. Weakly-supervised disentanglement without compromises. *arXiv preprint arXiv:2002.02886*, 2020a.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems*, 2020b.

Chaochao Lu, Yuhuai Wu, Jośe Miguel Hernández-Lobato, and Bernhard Schölkopf. Nonlinear invariant risk minimization: A causal approach. *arXiv preprint arXiv:2102.12353*, 2021.

Dhruv Kumar Mahajan, Ross B. Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *ECCV*, 2018.

Gary Marcus. Deep learning: A critical appraisal. *arXiv preprint arXiv:1801.00631*, 2018.

Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S Ecker, Matthias Bethge, and Wieland Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. *arXiv preprint 1907.07484*, 2019.

Milton Llera Montero, Casimir JH Ludwig, Rui Ponte Costa, Gaurav Malhotra, and Jeffrey Bowers. The role of disentanglement in generalisation. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=qbH974jKUVy.

Norman Mu and Justin Gilmer. Mnist-c: A robustness benchmark for computer vision. *arXiv preprint arXiv:1906.02337*, 2019.

David Mytton. Assessing the suitability of the greenhouse gas protocol for calculation of emissions from public cloud computing workloads. *Journal of Cloud Computing*, 9(1):1–11, 2020.

Emmy Nother. The finiteness theorem for invariants of finite groups. In *Mathematische Annalen, 77*, pp. 89–92, 1915.

Charles Packer, Katelyn Gao, Jernej Kos, Philipp Krähenbühl, Vladlen Koltun, and Dawn Song. Assessing generalization in deep reinforcement learning. *arXiv preprint arXiv:1810.12282*, 2018.

G. Parascandolo, N. Kilbertus, M. Rojas-Carulla, and B. Schölkopf. Learning independent causal mechanisms. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4033–4041. PMLR, July 2018. URL http://proceedings.mlr.press/v80/parascandolo18a.html.

Judea Pearl. *Causality*. Cambridge university press, 2009.

Judea Pearl and Elias Bareinboim. Transportability of causal and statistical relations: A formal approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 25, 2011.

Judea Pearl, Elias Bareinboim, et al. External validity: From do-calculus to transportability across populations. *Statistical Science*, 29(4):579–595, 2014.

Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, pp. 947–1012, 2016.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pp. 1278–1286. PMLR, 2014.

Geoffrey Roeder, Luke Metz, and Diederik P Kingma. On linear identifiability of learned representations. *arXiv preprint arXiv:2007.00810*, 2020.

Mateo Rojas-Carulla, Bernhard Schölkopf, Richard Turner, and Jonas Peters. Invariant models for causal transfer learning. *The Journal of Machine Learning Research*, 19(1):1309–1342, 2018.

Prasun Roy, Subhankar Ghosh, Saumik Bhattacharya, and Umapada Pal. Effects of degradations on deep neural network architectures. *arXiv preprint 1807.10108*, 2018.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/e6acf4b0f69f6f6e60e9a815938aa1ff-Paper.pdf.

Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

B. Schölkopf, D. Janzing, J. Peters, E. Sgouritsa, K. Zhang, and J. M. Mooij. On causal and anticausal learning. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, pp. 1255–1262, 2012.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 2021.

Rui Shu, Yining Chen, Abhishek Kumar, Stefano Ermon, and Ben Poole. Weakly supervised disentanglement with guarantees. In *International Conference on Learning Representations*, 2019.

Vincent Sitzmann, Julien NP Martel, Alexander W Bergman, David B Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *arXiv preprint arXiv:2006.09661*, 2020.

R. Solomonoff. A formal theory of inductive inference. *Information and Control, Part II*, 7(2):224–254, 1964.

Elizabeth S Spelke. Principles of object perception. *Cognitive science*, 14(1):29–56, 1990.

Amos Storkey. When training and test sets are different: characterizing learning transfer. *Dataset shift in machine learning*, 30:3–28, 2009.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp, 2019.

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pp. 843–852, 2017.

Ernő Téglás, Edward Vul, Vittorio Girotto, Michel Gonzalez, Joshua B Tenenbaum, and Luca L Bonatti. Pure reasoning in 12-month-old infants as probabilistic inference. *Science*, 332(6033):1054–1059, 2011.

Frederik Träuble, Elliot Creager, Niki Kilbertus, Francesco Locatello, Andrea Dittadi, Anirudh Goyal, Bernhard Schölkopf, and Stefan Bauer. On disentangled representations learned from correlated data. *arXiv preprint arXiv:2006.07886*, 2020.

Vladimir N Vapnik. *The nature of statistical learning theory*. Springer International Publishing, 1995.

Vladimir N Vapnik and A Ya Chervonenkis. Necessary and sufficient conditions for the uniform convergence of means to their expectations. *Theory of Probability & Its Applications*, 26(3):532–553, 1982.

Kimberly Villalobos, Vilim Štih, Amineh Ahmadinejad, Shobhita Sundaram, Jamell Dozier, Andrew Francl, Frederico Azevedo, Tomotake Sasaki, and Xavier Boix. Do neural networks for segmentation understand insideness? Technical report, Center for Brains, Minds and Machines (CBMM), 2020.

Julius von Kügelgen, Alexander Mey, and Marco Loog. Semi-generative modelling: Covariate-shift adaptation with cause and effect features. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1361–1369. PMLR, 2019.

Ulrike Von Luxburg and Bernhard Schölkopf. Statistical learning theory: Models, concepts, and results. In *Handbook of the History of Logic*, volume 10, pp. 651–706. Elsevier, 2011.

Maurice Weiler and Gabriele Cesa. General E(2)-Equivariant Steerable CNNs. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

David H. Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Comput.*, 8(7): 1341–1390, October 1996. ISSN 0899-7667. doi: 10.1162/neco.1996.8.7.1341. URL https://doi.org/10.1162/neco.1996.8.7.1341.

D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997. doi: 10.1109/4235.585893.

Xiaolin Wu, Xi Zhang, and Jun Du. Challenge of spatial cognition for deep learning. *CoRR*, abs/1908.04396, 2019. URL http://arxiv.org/abs/1908.04396.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10687–10698, 2020.

Z. Yan and X. Zhou. How intelligent are convolutional neural networks? *ArXiv*, abs/1709.06126, 2017.

Renqiao Zhang, Jiajun Wu, Chengkai Zhang, William T. Freeman, and Joshua B. Tenenbaum. A comparative evaluation of approximate probabilistic simulation and deep neural networks as accounts of human physical scene understanding. *CoRR*, abs/1605.01138, 2016. URL http://arxiv.org/abs/1605.01138.

Xinhua Zhang, Yijing Watkins, and Garrett T Kenyon. Can deep learning learn the principle of closed contour detection? In *International Symposium on Visual Computing*, pp. 455–460. Springer, 2018.

Yan Zhang, Jonathon Hare, and Adam Prugel-Bennett. Deep set prediction networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/6e79ed05baec2754e25b4eac73a332d2-Paper.pdf.

ETHICS STATEMENT

Our current study focuses on basic research and has no direct application or societal impact. Nevertheless, we think that the broader topic of generalization should be treated with great care. Especially oversimplified generalization and automation without a human in the loop could have drastic consequences in safety critical environments or court rulings.

Large-scale studies require a lot of compute due to multiple random seeds and exponentially growing sets of possible hyperparameter combinations. Following claims by Strubel et al. (Strubell et al., 2019), we tried to avoid redundant computations by orienting ourselves on current common values in the literature and by relying on systematic test runs. In a naive attempt, we tried in to estimate the power consumption and greenhouse gas impact based on the used cloud compute instance. However, too many factors such as external thermal conditions, actual workload, type of power used and others are involved (Mytton, 2020; Fahad et al., 2019). In the future, especially with the trend towards larger network architectures, compute clusters should be required to enable options which report the estimated environmental impact. However, it should be noted that cloud vendors are already among the largest purchasers of renewable electricity (Mytton, 2020).

For an impact statement for the broader field of representation learning, we refer to Klindt et al. (2020).

REPRODUCIBILITY STATEMENT

Our code is attached, and all important details to reproduce our results are repeated in Appendix H.

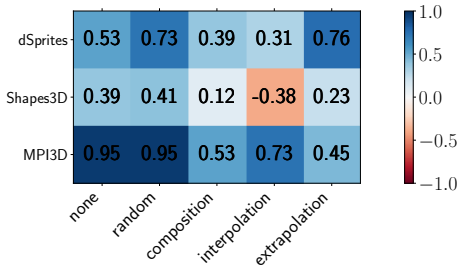


Figure 7: **Spearman Correlation of degree of disentanglement with downstream performances**. We measure the DCI-Disentanglement metric on the 10-dimensional representation for β-VAE, PCL, SlowVAE and Ada-GVAE and the corresponding R^2-score on the downstream performance. All p-values are below 0.01 except for composition on Shapes3D which has p-value=0.14. Note that, we here provide Spearman's rank correlation instead Pearson as the p-values are slightly lower.

Data set	Modification	R-squared Test	Modification	R-squared Test
dSprites	random	1.000	random + sign-flip	1.000
dSprites	composition	1.000	composition + sign-flip	1.000
dSprites	interpolation	1.000	interpolation + sign-flip	1.000
dSprites	extrapolation	1.000	extrapolation + sign-flip	0.999
Shapes3D	random	1.000	random + sign-flip	1.000
Shapes3D	composition	1.000	composition + sign-flip	1.000
Shapes3D	interpolation	1.000	interpolation + sign-flip	1.000
Shapes3D	extrapolation	1.000	extrapolation + sign-flip	1.000
MPI3D-Real	random	1.000	random + sign-flip	1.000
MPI3D-Real	composition	1.000	composition + sign-flip	0.996
MPI3D-Real	interpolation	1.000	interpolation + sign-flip	1.000
MPI3D-Real	extrapolation	0.999	extrapolation + sign-flip	0.997

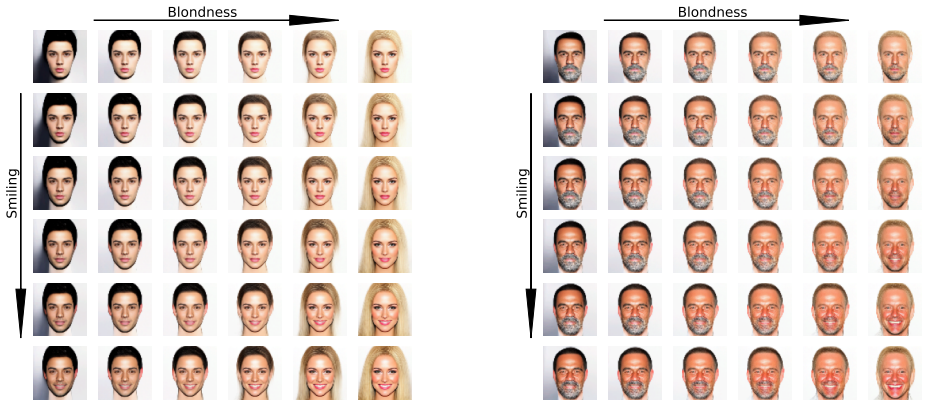
Table 1: **Performances of the readout-MLP on the ground-truth.**

A Connection between readout performance and disentanglement of the representation

Here, we narrow down the root cause of the limited extrapolation performance of disentanglement models in the OOD settings as observed in Figs. 4 and 11. More precisely, we investigate how the readout-MLP would perform on a perfectly disentangled representation. Therefore, we train our readout MLP directly on the ground-truth factors of variation for all possible test-train splits described in Fig. 2 and measured the R^2-score test error for each split. Here, the MLP only has to learn the identity function. In a slightly more evolved setting, termed *sign-flip*, we switched the sign input to train the readout-MLP on a mapping from -ground-truth to ground-truth. This mimics the identifiability guarantees of models like SlowVAE which are up to permutation and sign flips under certain assumptions. The R-squared for all settings in Table 1 are $>$ .99, therefore the readout model should not be the limitation for OOD generalization in our setting if the representation is identified up to permutation and sign flips. Note that this experiment does not cover disentanglement up to point-wise nonlinearities or linear/ affine transformations as required by other models.

B CelebGlow Dataset

The current disentanglement datasets such as dSprites, Shapes3D, MPI3D, and others are constructed based on highly controlled environments (Matthey et al., 2017; Kim & Mnih, 2018; Gondal et al., 2019). Here, common factors of variations are rotations or scaling of simple geometric objects, such as a square. For a more intuitive investigation of other factors, we created the CelebGlow dataset. Here, the factors of variations are smiling, blondness and age. Samples are shown in Fig. 8. Note that we rely on the Glow model instead of taking a real-world dataset, as this allows for a gradual control of individual factors of variation.



(a) Age 1/6 (young) (b) Age 6/6 (old)

Figure 8: **CelebGlow Dataset**

The CelebGlow dataset is created based on the invertible generative neural network of Kingma et al. (Kingma & Dhariwal, 2018). We used their provided network[2] that is pretrained on the Celeb-HQ dataset, and has labelled directions in the model-latent space that correspond to specific attributes of the dataset. Based on this latent space, we created the dataset as follows:

1. In the latent space of the model, we sample from a high dimensional Gaussian with zero mean and a low standard deviation of 0.3 to avoid too much variability.
2. Next, we perform a latent walk into the directions that correspond to "Smiling", "Age" and "Blondness" in image space. To estimate the spacing, we rely on the function `manipulate_range`[3]. We perform 6 steps along each axis and all combinations (6x6x6 cube). As a scale parameter to the function, we use 0.8. Those factors were chosen s.t. the images differ significantly, but also to stay in the valid range of the model based on visual inspection.
3. We pass all latent coordinates through the glow network in the generative direction.
4. We further down-sample the images from 256x256x3 to 64x64x3 to match the resolution of common disentanglement datasets.
5. Finally, we store each image and the corresponding factor combination.

This procedure is repeated for 1000 samples to get $6 * 6 * 6 * 1000 = 216000$ samples in total, which is around the same size as other common datasets.

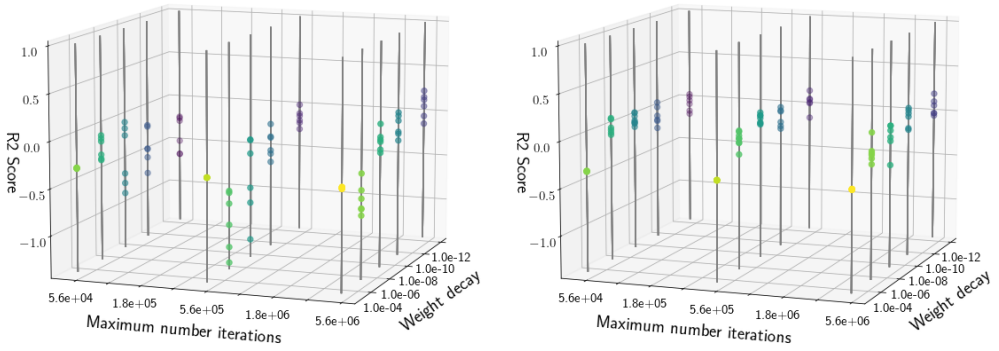
C Hyperparameter Tuning Ablation

As described in the implementation details, we use common values from the literature to train the proposed models. Here, we investigate effects of such hyperparameters on the CNN architecture. Due to the combinatorial complexity, we do not perform a search for other architectures. As hyperparameters, we varied the number or training iterations (3 different numbers of iterations), we introduced 5 different strengths of regularization, 2 different depths for the CNN architecture [6 layers, 9 layers] and ran multiple random seeds for each combination.

The results on the extrapolation test on MPI3D set are shown in Fig. 9. Given this hyperparameter search, we find no improvement over our reported numbers for the CNN.

[2]The network can be found at: https://github.com/openai/glow/blob/master/demo/script.sh#L24

[3]https://github.com/openai/glow/blob/master/demo/model.py#L219



(a) **6 layer CNN** (b) **9 layer CNN**

Figure 9: **Hyperparameter search on MPI3D for a CNN**.

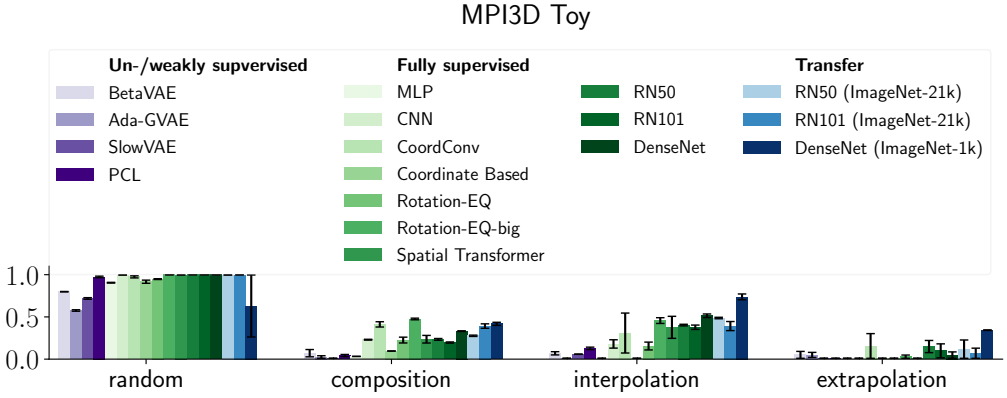


Figure 10: **R^2-score on MPI3D-Synthetic**.

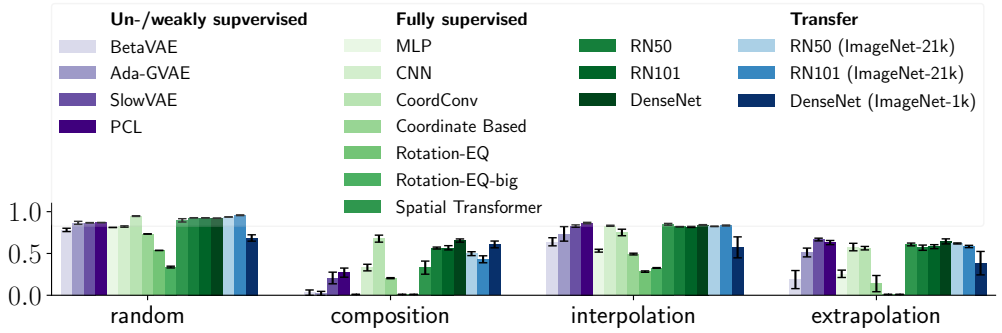
D Real versus synthetic dataset

To narrow down the question "why the generalization capabilities drop on real-world dataset MPI3D?", we run a comparison on MPI3D dataset with real and synthetic images.

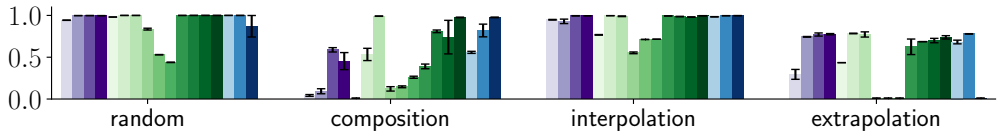
The results on the MPI3D dataset with synthetic images is shown in Fig. 10 and Table 10. Comparing this with R-squared performances to MPI3D with real-world images (Fig. 4 and Table 9), we observe that the results do not change significantly (most results are in a 1-2sigma range). We conclude that the larger drops in performance on MPI3D compared to Shapes3D or dSprites, are not due to the real images as opposed to synthetic images. Instead, we hypothesize that it is due to the more realistic setup of the MPI3D dataset itself. For instance, it contains complex factors like rotation in 3D projected on 2D. Here, occluded parts of objects have to be guessed based on certain symmetry assumptions.

E Ablation on non-ambiguous dSprites

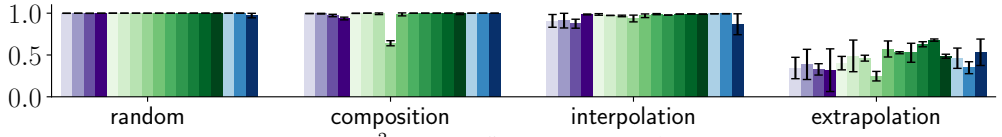
The setup of dSprites is non-injective, as different rotations map to the same image. E.g., the square at a rotation $90°$ is identical to the one rotated by $180°$ and therefore ambiguous. Thus, the training process is noisy. In an ablation study, we controlled for this by constraining the rotations to lie in $[0, 90)$. We again ran all our proposed models and report the R^2-Score in Fig. 11b.



(a) R^2**-score on dSprites**.



(b) R^2**-score on dSprites rotation** $[0, 90)$.



(c) R^2**-score on Shapes3D test splits.**

Figure 11: R^2**-score on various splits.**

Comparing the new results with the original dSprites results shows: First, for the random test-train split, resolving the rotational ambiguity leads to almost perfect performance (close to 100% R-squared scores for most models). In the previous dSprites setup with rotational ambiguity, top accuracies are around 70-95% R-squared scores for most models. Second, large drops in performance can still be observed when we move towards the systematic out-of-distribution splits (composition, interpolation, and extrapolation). Also, our insights on how models extrapolate remain the same. Lastly, for the random split, the Rotation-EQ model shows non-perfect performance. Tracing this error to individual factors, it turns out this is due to limited capabilities in predicting the x, y positions. We hypothesize that this is due to limitations of convolutions in propagating spatial positions, as discussed in (Liu et al., 2018). The DenseNet performs perfectly on the train set and might be overfitting.

We conclude that the rotational ambiguity explains the drops on the random split. However, the clear drops in performance on the systematic splits remain nonetheless. Thus, the analysis we perform in the paper and the conclusions we draw remain the same.

F Data Augmentations

We investigate the effects of data augmentation during training time on the generalization performance in the extrapolation setting of our proposed benchmark.

As data augmentations, we applied random erasing, Gaussian Noise, small shearings, and blurring. Note that we could not use arbitrary augmentations. For instance, shift augmentations would lead to ambiguities with the "shift" factor in dSprites. Next, we trained CNNs with and without data augmentations on all four datasets (dSprites, Shapes3D, MPI3D, CelebGlow) on the extrapolation splits with multiple random seeds.

The results are visualized in Fig. 12. For the mean performance, we observe no significant improvement by adding augmentations. However, the overall spread of the scores seems to decrease given augmentations on some datasets. We explain this by the fact that the augmentations enforce cer-

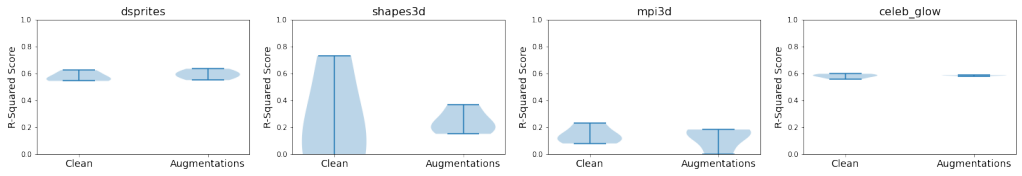


Figure 12: R^2**-score on data augmenations**. We depict the performance on the extrapolation setting with and without data augmentations for a CNN network with various random seeds on our considered datasets.

tain invariances, narrowing the solution space of optimal training solutions by providing a further specification (specification in the sense of D'Amour et al. (2020)).

G Performance with respect to individual factors

We here try to attribute the performance losses to individual OOD factors (see §5.2). Thus, on the extrapolation setting, we modify the test-splits such that only a single factor is out-of-distribution. Next, we measure the overall performance across models (all fully supervised and transfer models) to demonstrate the effect of this factor. The results are depicted for all models in Fig. 13. Overall, factors like "height" on MPI3D that control the viewing of the camera and, subsequently, change attributes like the absolute position in the image of other factors (e.g., the tip of the robot arm) have a high effect.

H Implementation details

H.1 Data sets

Each dataset consists of multiple factors of variation and every possible combination of factors generates a corresponding image. Here, we list all datasets and their corresponding factor ranges. Note, to estimate the reported R^2-score, we normalize the factors by dividing each factor $\mathbf{y}_i$ by $|\mathbf{y}_i^{\max} - \mathbf{y}_i^{\min}|$, i.e., all factors are in the range $[0, 1]$. *dSprites* (Matthey et al., 2017), represents some low resolution binary images of basic shapes with the 5 FoVs shape {0, 1, 2}, scale {0, ..., 4}, orientation[4] {0, ..., 39}, x-position {0, ..., 31}, and y-position {0, ..., 31}. Next, *Shapes3D* (Kim & Mnih, 2018) which is a similarly popular dataset with 3D shapes in a room scenes defined by the 6 FoVs floor color {0, ..., 9}, wall color {0, ..., 9}, object color {0, ..., 9}, object size {0, ..., 7}, object type {0, ..., 3} and azimuth {0, ..., 14}. Lastly, we consider the challenging and more realistic dataset *MPI3D* (Gondal et al., 2019) containing real images of physical 3D objects attached to a robotic finger generated by 7 FoVs color {0, ..., 5}, shape {0, ..., 5}, size {0, 1}, height {0, 1, 2}, background color {0, 1, 2}, x-axis {0, ..., 39} and y-axis {0, ..., 39}.

H.2 Data set Splits

Each dataset is complete in the sense that it contains all possible combinations of factors of variation. Thus, the interpolation and extrapolation test-train splits are fully defined by specifying which factors are exclusively in the test set. Starting from all possible combinations, if a given factor value

[4]Note that this dataset contains a non-injective generative model as square and ellipses have multiple rotational symmetries.

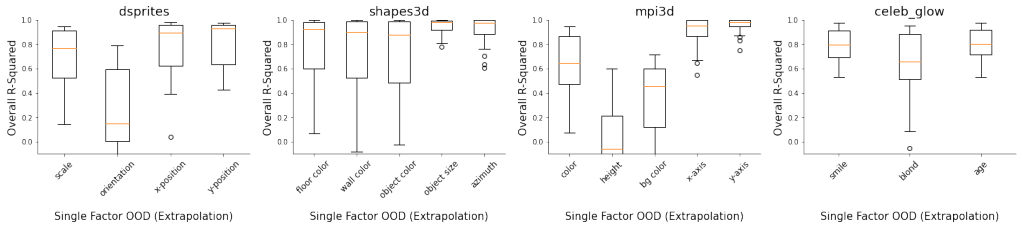


Figure 13: R^2**-score on in individual factors**. Extrapolation performance across models when only a single factor (x-axis) is OOD.

is defined to be exclusively in the test set, the corresponding image is part of the test set. E.g. for the extrapolation case in dSprites, all images containing x-positions > 24 are part of the test set and the train set its respective complement $D \backslash D_{test}$. Composition can be defined equivalently to extrapolation but with interchanged test and train sets. The details of the splits are provided in table Tables 2 and 3. The resulting train vs. test sample number ratios are roughly $30:70$. See Table 4. We will release the test and train splits to allow for a fair comparison and benchmarking for future work.

For the setting where only a single factor is OOD, we formally define this as

$$\mathcal{D}_{\textit{one-ood}} = \{(\mathbf{y}^k, \mathbf{x}^k) \in \mathcal{D}_{\text{te}} \mid \exists! i \in \mathbb{N}\ s.t.\ \ y_i^k \neq y_i^l\ \forall (\mathbf{y}^l, \mathbf{x}^l) \in \mathcal{D}_{\text{tr}}\}. \tag{3}$$

Here, we used the superscript indices to refer to a sample and the subscript to denote the factor. Note that the defined set is only nonempty in the interpolation and extrapolation settings.

H.3 Training

All models are implemented using PyTorch 1.7. If not specified otherwise, the hyperparameters correspond to the default library values.

Un-/ weakly supervised For the un-/weakly supervised models, we consider 10 random seeds per hyperparameter setup. As hyperparameters, we optimize one parameter of the learning objective per model similar to Table 2 from Locatello et al. (Locatello et al., 2020a). For the SlowVAE, we took the optimal values from Klindt et al. (Klindt et al., 2020) and tuned for $\gamma \in \{1, 5, 10, 15, 20, 25\}$. The PCL model itself does not have any hyperparameters (Hyvarinen & Morioka, 2017). For simplicity, we determine the optimal setup in a supervised manner by measuring the DCI-Disentanglement score (Eastwood & Williams, 2018) on the training split. The PCL and SlowVAE models are trained on pairs of images that only differ sparsely in their underlying factors of variation following a Laplace transition distribution, the details correspond to the implementation [5] of Klindt et al. (Klindt et al., 2020). The Ada-GVAE models are trained on pairs of images that differ uniformly in a single, randomly selected factor. Other factors are kept fixed. This matches the strongest model from Locatello et al. (Locatello et al., 2020a) implemented on GitHub[6]. All β-VAE models are trained in an unsupervised manner. All un- and weakly supervised models are trained with the Adam optimizer with a learning rate of 0.0001. We train each model for $500,000$ iterations with a batch size of 64, which for the weakly supervised models, corresponds to 64 pairs. Lastly, we train a supervised readout model on top of the latents for 8 epochs with the Adam optimizer on the full corresponding training dataset and observe convergence on the training and test datasets - no overfitting was observed.

Fully supervised: All fully supervised models are trained with the same training scheme. We use the Adam optimizer with a learning rate of 0.0005. The only exception is DenseNet, which is trained with a learning rate of 0.0001, as we observe divergences on the training loss with the higher learning rate. We train each model with three random seeds for $500,000$ iterations with a batch size of $b = 64$. As a loss function, we consider the mean squared error $\text{MSE} = \sum_{j=0}^{b} ||\mathbf{y}_j - f_j(\mathbf{x})||_2^2 / b$ per mini-batch.

Transfer learning: The pre-trained models are fine-tuned with the same loss as the fully supervised models. We train for $50,000$ iterations and with a lower learning rate of 0.0001. We fine-tune all model weights. As an ablation, we also tried only training the last layer while freezing the other weights. In this setting, we consistently observed worse results and, therefore, do not include them in this paper.

H.4 Model implementations

Here, we shortly describe the implementation details required to reproduce our model implementation. We denote code from Python libraries in grey. If not specified otherwise, the default parameters and nomenclature correspond to the PyTorch 1.7 library.

[5] https://github.com/bethgelab/slow_disentanglement/blob/master/scripts/dataset.py#L94

[6] https://github.com/google-research/disentanglement_lib/blob/master/disentanglement_lib/methods/weak/weak_vae.py#L62 and https://github.com/google-research/disentanglement_lib/blob/master/disentanglement_lib/methods/weak/weak_vae.py#L317

	dataset	split	name	exclusive test factors
0	dSprites	interpolation	shape	{}
1	dSprites	interpolation	scale	{1, 4}
2	dSprites	interpolation	orientation	{32, 2, 37, 7, 12, 17, 22, 27}
3	dSprites	interpolation	x-position	{2, 7, 11, 15, 20, 24, 29}
4	dSprites	interpolation	y-position	{2, 7, 11, 15, 20, 24, 29}
5	dSprites	extrapolation	shape	{}
6	dSprites	extrapolation	scale	{4, 5}
7	dSprites	extrapolation	orientation	{32, 33, 34, 35, 36, 37, 38, 39}
8	dSprites	extrapolation	x-position	{25, 26, 27, 28, 29, 30, 31}
9	dSprites	extrapolation	y-position	{25, 26, 27, 28, 29, 30, 31}
10	Shapes3D	interpolation	floor color	{2, 7}
11	Shapes3D	interpolation	wall color	{2, 7}
12	Shapes3D	interpolation	object color	{2, 7}
13	Shapes3D	interpolation	object size	{2, 5}
14	Shapes3D	interpolation	object type	{}
15	Shapes3D	interpolation	azimuth	{2, 12, 7}
16	Shapes3D	extrapolation	floor color	{8, 9}
17	Shapes3D	extrapolation	wall color	{8, 9}
18	Shapes3D	extrapolation	object color	{8, 9}
19	Shapes3D	extrapolation	object size	{6, 7}
20	Shapes3D	extrapolation	object type	{}
21	Shapes3D	extrapolation	azimuth	{12, 13, 14}
22	MPI3D	interpolation	color	{3}
23	MPI3D	interpolation	shape	{}
24	MPI3D	interpolation	size	{}
25	MPI3D	interpolation	height	{1}
26	MPI3D	interpolation	background color	{1}
27	MPI3D	interpolation	x-axis	{24, 34, 5, 15}
28	MPI3D	interpolation	y-axis	{24, 34, 5, 15}
29	MPI3D	extrapolation	color	{5}
30	MPI3D	extrapolation	shape	{}
31	MPI3D	extrapolation	size	{}
32	MPI3D	extrapolation	height	{2}
33	MPI3D	extrapolation	background color	{2}
34	MPI3D	extrapolation	x-axis	{36, 37, 38, 39}
35	MPI3D	extrapolation	y-axis	{36, 37, 38, 39}
36	CelebGlow	interpolation	person	{}
37	CelebGlow	interpolation	smile	{1, 4}
38	CelebGlow	interpolation	blond	{1, 4}
39	CelebGlow	interpolation	age	{1, 4}
40	CelebGlow	extrapolation	person	{}
41	CelebGlow	extrapolation	smile	{4, 5}
42	CelebGlow	extrapolation	blond	{4, 5}
43	CelebGlow	extrapolation	age	{4, 5}

Table 2: **Interpolation and extrapolation splits.**

	dataset	split	name	exclusive train factors
0	dSprites	composition	shape	{}
1	dSprites	composition	scale	{}
2	dSprites	composition	orientation	{0, 1, 2, 3}
3	dSprites	composition	x-position	{0, 1, 2}
4	dSprites	composition	y-position	{0, 1, 2}
5	Shapes3D	composition	floor color	{0}
6	Shapes3D	composition	wall color	{0}
7	Shapes3D	composition	object color	{0}
8	Shapes3D	composition	object size	{}
9	Shapes3D	composition	object type	{}
10	Shapes3D	composition	azimuth	{0}
11	MPI3D	composition	color	{}
12	MPI3D	composition	shape	{}
13	MPI3D	composition	size	{}
14	MPI3D	composition	height	{}
15	MPI3D	composition	background color	{}
16	MPI3D	composition	x-axis	{0, 1, 2, 3, 4, 5}
17	MPI3D	composition	y-axis	{0, 1, 2, 3, 4, 5}

Table 3: **Composition splits.**

	dataset	split	% test	% train	Total samples
0	dSprites	random	32.6	67.4	737280
1	dSprites	composition	26.1	73.9	737280
2	dSprites	interpolation	32.6	67.4	737280
3	dSprites	extrapolation	32.6	67.4	737280
4	Shapes3D	random	30.7	69.3	480000
5	Shapes3D	composition	32.0	68.0	480000
6	Shapes3D	interpolation	30.7	69.3	480000
7	Shapes3D	extrapolation	30.7	69.3	480000
8	MPI3D	random	30.0	70.0	1036800
9	MPI3D	composition	27.8	72.2	1036800
10	MPI3D	interpolation	30.0	70.0	1036800
11	MPI3D	extrapolation	30.0	70.0	1036800

Table 4: **Test train ratio.**

The un- and weakly supervised models β-**VAE**, **Ada-GVAE** and **SlowVAE** all use the same encoder-decoder architecture as Locatello et al. (Locatello et al., 2020a). The **PCL** model uses the same architecture as the encoder as well and with the same readout structure for the contrastive loss as used by Hyvärinen et al. (Hyvarinen & Morioka, 2017). For the supervised readout MLP, we use the sequential model `[Linear(10, 40), ReLU(), Linear(40, 40), ReLU(40, 40), Linear(40, 40), ReLU(), Linear(40, number-factors)]`.

The **MLP** model consists of `[Linear(64*64*number-channels, 90), ReLU(), Linear(90, 90), ReLU(), Linear(90, 90), ReLU(), Linear(90, 90), ReLU(), Linear(90, 45), ReLU(), Linear(22, number-factors)]`. The architecture is chosen such that it has roughly the same number of parameters and layers as the CNN.

The **CNN** architecture corresponds the one used by Locatello et al. (Locatello et al., 2020a). We only adjust the number of outputs to match the corresponding datasets.

The **CoordConv** consists of a *CoordConv2D* layer following the PyTorch implementation[7] with 16 output channels. It is followed by 5 ReLU-Conv layers with 16 in- and output channels each and a MaxPool2D layer. The final readout consists of `[Linear(32, 32), ReLU(), Linear(32, number-factors)]`.

The **SetEncoder** concatenates each input pixel with its i, j pixel coordinates normalized to $[0, 1]$. All concatenated pixels (i, j, pixel-value) are subsequently processed with the same network which consists of `[Linear(2+number-channels), ReLU(), Linear(40, 40), ReLU(), Linear(40, 20), ReLU()]`. This is followed by a mean pooling operation per image which guarantees an invariance over the order of the inputs, i.e. one could shuffle all inputs and the output would remain the same. As a readout, it follows a sequential fully connected network consisting of `[Linear(20, 20), ReLU(), Linear(20, 20), ReLU(), Linear(20, number-factors)]`.

The rotationally equivariant network **RotEQ** is similar to the architecture from Locatello et al. (Locatello et al., 2020a). One difference is that it uses the R2Conv module[8] from Weiler et al. (Weiler & Cesa, 2019) instead of the PyTorch *Conv2d* with an 8-fold rotational symmetry. We thus decrease the number of feature maps by a factor of 8, which roughly corresponds to the same computational complexity as the CNN. We provide a second version which does not decrease the number of feature maps and, thus, has the same number of trainable parameters as the CNN but a higher computational complexity. We refer to this version as **RotEQ-big.**

To implement the spatial transformer (**STN**) (Wu et al., 2019), we follow the PyTorch tutorial implementation[9] which consists of two steps. In the first step, we estimate the parameters of a (2, 3)-shaped affine matrix using a sequential neural network with the following architecture `[Conv2d(number_channels, 8, kernel_size=7), MaxPool2d(2, stride=2), ReLU(), Conv2d(8, 10, kernel_size=5), MaxPool2d(2, stride=2), ReLU(), Conv2d(10, 10, kernel_size=6), MaxPool2d(2, stride=2), ReLU(), Linear(10*3*3, 31), ReLU(), Linear(32, 3*2)]`. In the second step, the input image is transformed by the estimated affine matrix and subsequently processed by a CNN which has the same architecture as the CNN described above.

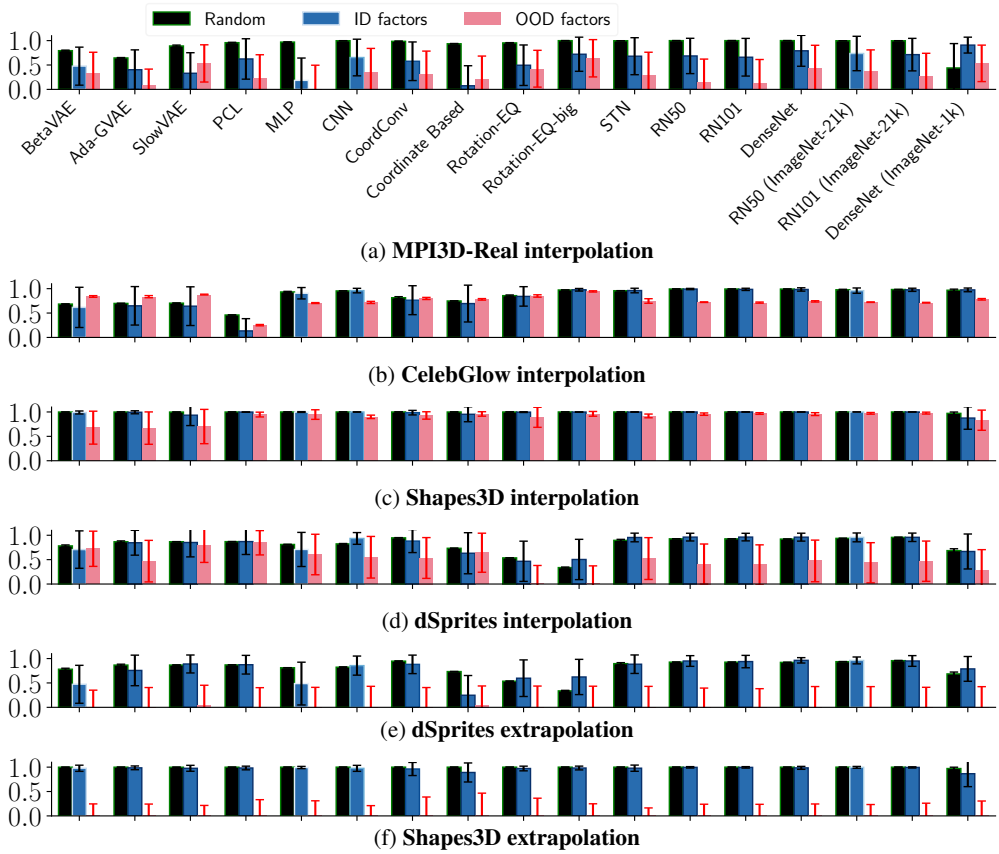
For the transfer learning models ResNet50 (**RN50**) and ResNet101 (**RN101**) pretrained on ImageNet-21k (IN-21k), we use the big-transfer (Kolesnikov et al., 2020) implementation[10]. For the RN50, we download the weights with the tag `"BiT-M-R50x1"`, and for the RN101, we use the tag `"BiT-M-R101x3"`. For the **DenseNet** trained on ImageNet-1k (IN-1k), we used the weights from `densenet121`. For all transfer learning methods, we replace the last layer of the pre-trained models with a randomly initialized linear layer which matches the number of outputs to the number

[7] https://github.com/walsvid/CoordConv

[8] https://github.com/QUVA-Lab/e2cnn

[9] https://pytorch.org/tutorials/intermediate/spatial_transformer_tutorial.html

[10] https://colab.research.google.com/github/google-research/big_transfer/blob/master/colabs/big_transfer_pytorch.ipynb and for the weights https://storage.googleapis.com/bit_models/{bit_variant}.npz



(a) **MPI3D-Real interpolation**

(b) **CelebGlow interpolation**

(c) **Shapes3D interpolation**

(d) **dSprites interpolation**

(e) **dSprites extrapolation**

(f) **Shapes3D extrapolation**

Figure 14: **Interpolation / Extrapolation and modularity.**

of factors in each dataset. As an ablation, we also provide a randomly initialized version for each transfer learning model.

H.5 Compute

All models are run on the NVIDIA T4 Tensor Core GPUs on the AWS g4dn.4xlarge instances with an approximate total compute of 20 000 GPUh. To save computational cost, we gradually increased the number of seeds until we achieved acceptable p-values of ≤ 0.05. In the end, we have 3 random seeds per supervised model and 10 random seeds per hyperparameter setting for the un and weakly supervised models.

I Additional results

modification models	random	composition	interpolation	extrapolation
BetaVAE	78.2± 2.1	0.1± 6.5	64.0± 5.1	18.3± 12.3
Ada-GVAE	86.6± 1.9	-1.4± 5.5	73.4± 9.2	51.2± 5.5
SlowVAE	86.7± 0.2	20.7± 7.3	82.8± 1.4	66.7± 1.7
PCL	87.0± 0.1	27.1± 5.5	86.4± 0.9	63.0± 2.5
MLP	81.1± 0.2	-10.6± 1.3	53.3± 2.1	25.7± 5.4
CNN	82.3± 0.9	33.1± 4.8	83.1± 0.8	57.7± 5.5
CoordConv	94.7± 0.6	67.7± 5.1	75.1± 4.8	56.3± 2.6
Coordinate Based	73.3± 0.3	20.4± 0.8	49.3± 1.3	8.8± 20.5
Rotation-EQ	53.6± 0.1	-12.9± 7.1	28.3± 1.0	-23.8± 4.1
Rotation-EQ-big	33.7± 1.3	-8.6± 1.6	32.7± 0.5	-25.9± 1.2
Spatial Transformer	89.6± 2.4	33.0± 9.6	84.9± 1.3	60.8± 2.0
RN50	92.7± 0.1	56.5± 1.4	82.1± 0.1	56.9± 3.9
RN101	92.6± 0.1	56.8± 3.1	81.7± 0.8	58.1± 2.9
DenseNet	92.2± 0.2	65.4± 2.4	84.3± 0.2	64.4± 3.7
RN50 (ImageNet-21k)	93.6± 0.2	49.7± 2.9	82.5± 0.3	62.0± 0.8
RN101 (ImageNet-21k)	95.7± 0.5	43.0± 4.8	83.5± 0.6	58.3± 1.6
DenseNet (ImageNet-1k)	68.5± 5.4	60.8± 5.7	57.3± 17.7	38.4± 19.8

Table 5: R^2**-score on dSprites**

modification models	random	composition	interpolation	extrapolation
BetaVAE	94.3+- 0.3	4.3+- 1.5	94.8+- 0.5	29.5+- 8.4
Ada-GVAE	99.8+- 0.0	9.3+- 4.3	92.9+- 3.8	74.4+- 0.6
SlowVAE	99.8+- 0.0	58.9+- 3.7	99.5+- 0.3	77.1+- 2.7
PCL	99.7+- 0.0	45.3+- 14.4	99.6+- 0.0	77.5+- 1.1
MLP	98.1+- 0.1	-8.8+- 0.2	76.7+- 0.5	43.5+- 0.1
CNN	100.0+- 0.0	53.3+- 10.4	99.7+- 0.0	78.4+- 0.5
CoordConv	100.0+- 0.0	99.1+- 0.3	99.0+- 0.6	77.2+- 4.4
Coordinate Based	83.6+- 1.5	12.1+- 3.7	55.2+- 1.5	-11.5+- 15.6
Rotation-EQ	53.0+- 0.3	14.6+- 1.6	71.3+- nan	-6.9+- 0.8
Rotation-EQ-big	43.9+- 0.4	26.0+- 1.7	71.5+- 0.0	-1.9+- 1.3
Spatial Transformer	100.0+- 0.0	39.0+- 4.0	99.4+- 0.2	62.5+- 13.1
RN50	100.0+- 0.0	81.0+- 2.2	98.7+- 0.4	68.6+- 0.1
RN101	100.0+- 0.0	74.0+- 28.4	98.0+- 0.4	69.9+- 3.7
DenseNet	100.0+- 0.0	97.6+- 0.5	99.6+- 0.2	73.8+- 2.8
RN50 (ImageNet-21k)	100.0+- 0.0	55.8+- 1.8	98.3+- 0.3	68.2+- 3.1
RN101 (ImageNet-21k)	100.0+- 0.0	82.0+- 10.7	99.6+- 0.1	77.9+- 0.4
DenseNet (ImageNet-1k)	87.0+- 18.2	97.6+- 0.5	99.8+- 0.1	-420.5+- 504.1

Table 6: R^2**-score on dSprites rotation** $[0, 90)$

modification models	random	composition	interpolation	extrapolation
BetaVAE	99.9± 0.1	99.6± 0.2	90.8± 8.0	34.4± 13.5
Ada-GVAE	99.9± 0.0	99.4± 0.4	91.1± 9.2	37.6± 21.6
SlowVAE	99.8± 0.1	97.2± 1.5	87.4± 5.7	32.8± 7.2
PCL	99.9± 0.0	93.6± 1.6	98.5± 0.5	29.8± 29.1
MLP	100.0± 0.0	99.8± 0.2	98.5± 1.2	40.3± 9.9
CNN	100.0± 0.0	100.0± 0.0	97.3± 0.1	48.9± 23.2
CoordConv	100.0± 0.0	99.3± 1.2	96.7± 1.1	46.2± 4.3
Coordinate Based	100.0± 0.0	64.0± 3.7	93.6± 5.0	24.7± 7.0
Rotation-EQ	100.0± 0.0	98.5± 2.2	96.9± 2.7	57.2± 11.7
Rotation-EQ-big	100.0± 0.0	100.0± 0.0	98.8± 0.8	52.7± 1.5
Spatial Transformer	100.0± 0.0	100.0± 0.0	97.8± 0.1	52.7± 13.9
RN50	100.0± 0.0	100.0± 0.0	98.8± 0.3	62.8± 3.7
RN101	100.0± 0.0	100.0± 0.0	99.1± 0.1	67.8± 1.7
DenseNet	100.0± 0.0	99.3± 1.2	98.9± 0.3	48.5± 3.0
RN50 (ImageNet-21k)	100.0± 0.0	100.0± 0.0	99.3± 0.1	46.1± 14.8
RN101 (ImageNet-21k)	100.0± 0.0	100.0± 0.0	99.4± 0.2	34.8± 8.3
DenseNet (ImageNet-1k)	97.1± 3.8	100.0± 0.0	86.8± 17.7	53.2± 22.3

Table 7: R^2**-score on Shapes3D**

modification models	random	composition	interpolation	extrapolation
BetaVAE	68.4+- 0.6	43.7+- 3.1	66.2+- 0.1	43.5+- 0.3
Ada-GVAE	69.8+- 0.4	48.6+- 2.7	68.4+- 0.6	44.5+- 0.5
SlowVAE	70.3+- 0.6	53.9+- 2.7	69.5+- 0.0	43.7+- 1.0
PCL	46.0+- 0.1	-7.0+- 1.4	19.7+- 0.1	-48.4+- 2.6
MLP	93.3+- 1.7	75.3+- 2.2	82.0+- 0.6	50.0+- 2.7
CNN	95.4+- 0.0	75.9+- 1.7	86.9+- 1.3	57.9+- 3.0
CoordConv	81.5+- 3.0	57.7+- 1.8	74.4+- 2.4	33.5+- 6.7
Coordinate Based	74.8+- 0.3	58.0+- 3.9	68.8+- 0.5	22.7+- 1.8
Rotation-EQ	85.7+- 1.9	59.9+- 2.9	82.0+- 2.1	44.1+- 5.1
Rotation-EQ-big	97.3+- 0.0	81.7+- 1.6	96.1+- 0.1	66.4+- 0.3
Spatial Transformer	95.8+- 0.1	76.8+- 0.1	88.0+- 0.9	57.7+- 0.5
RN50	99.3+- 0.3	79.5+- 2.8	89.3+- 0.2	48.6+- 1.0
RN101	99.3+- 0.1	81.5+- 2.0	88.6+- 0.1	49.1+- 0.7
DenseNet	99.2+- 0.3	83.9+- 0.6	89.1+- 0.7	49.8+- 2.1
RN50 (ImageNet-21k)	97.7+- 1.3	76.9+- 0.5	86.8+- 0.7	50.1+- 0.8
RN101 (ImageNet-21k)	98.3+- 0.0	78.9+- 1.1	87.8+- 0.7	49.3+- 1.0
DenseNet (ImageNet-1k)	96.8+- 3.1	81.7+- 2.0	90.1+- 1.3	59.6+- 1.3

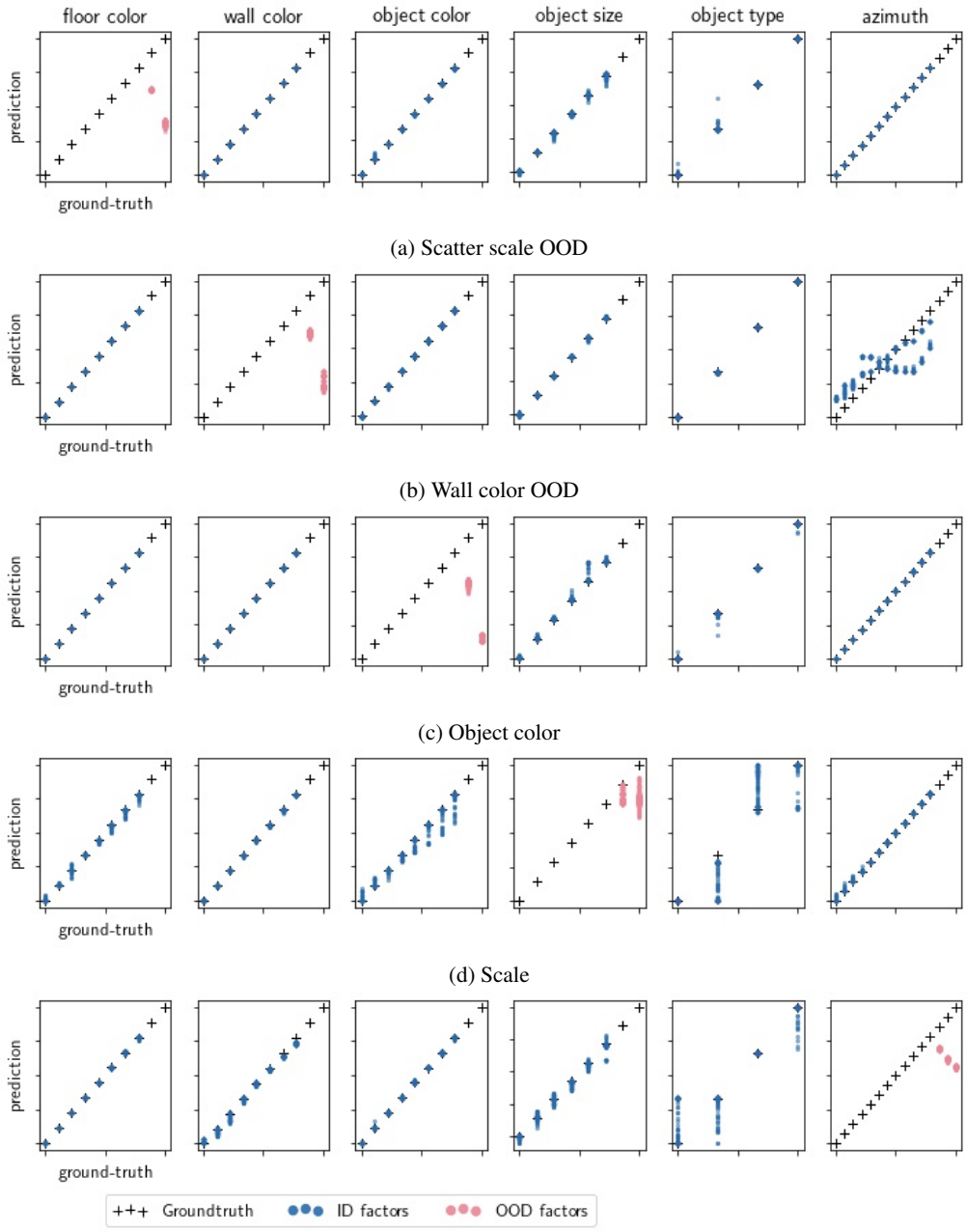
Table 8: R^2**-score on CelebGlow**

modification models	random	composition	interpolation	extrapolation
BetaVAE	79.4± 1.1	-6.2± 2.5	10.9± 8.9	-9.9± 6.3
Ada-GVAE	64.5± 0.8	-3.3± 3.0	9.5± 5.7	-3.6± 9.2
SlowVAE	89.0± 1.9	-16.6± 11.3	-10.9± 8.5	-31.5± 15.2
PCL	95.8± 0.7	10.7± 10.2	21.8± 7.5	-4.1± 10.3
MLP	97.0± 0.5	3.5± 4.0	-37.5± 7.5	-37.5± 12.1
CNN	99.8± 0.0	34.7± 1.4	26.3± 11.3	18.3± 10.0
CoordConv	98.6± 0.5	27.7± 19.3	18.5± 19.0	15.3± 27.5
Coordinate Based	93.5± 0.6	19.5± 5.3	-50.5± 48.0	-421.1± 286.5
Rotation-EQ	95.3± 0.6	23.6± 0.8	12.3± 12.1	-44.3± 15.3
Rotation-EQ-big	99.9± 0.0	45.9± 1.0	45.8± 3.6	10.5± 2.8
Spatial Transformer	99.8± 0.0	16.1± 2.4	31.5± 12.2	9.0± 8.8
RN50	100.0± 0.0	26.3± 1.5	29.4± 5.8	22.0± 5.3
RN101	100.0± 0.0	26.1± 4.6	23.3± 15.7	20.7± 4.4
DenseNet	100.0± 0.0	44.0± 1.0	54.6± 3.3	7.0± 11.2
RN50 (ImageNet-21k)	99.8± 0.0	23.8± 3.3	43.6± 6.5	54.1± 1.9
RN101 (ImageNet-21k)	99.8± 0.1	37.0± 3.4	35.4± 15.4	41.6± 8.5
DenseNet (ImageNet-1k)	44.0± 79.0	49.0± 0.7	72.2± 3.4	38.9± 1.9

Table 9: R^2**-score on MPI3D**

modification models	random	composition	interpolation	extrapolation
BetaVAE	79.9+- 0.4	7.1+- 6.1	6.9+- 2.7	1.1+- 11.4
Ada-GVAE	57.6+- 1.2	1.2+- 4.0	-22.4+- 2.5	5.0+- 4.2
SlowVAE	71.8+- 1.3	-5.6+- 1.7	5.9+- 0.2	-5.0+- 1.8
PCL	97.4+- 1.3	4.3+- 1.8	12.7+- 2.1	-11.3+- 5.9
MLP	90.5+- 0.6	3.2+- 0.0	-33.2+- 0.8	-31.1+- 4.4
CNN	99.5+- 0.0	23.0+- 0.7	18.0+- 7.0	-26.0+- 6.4
CoordConv	97.4+- 1.8	41.2+- 4.4	30.9+- 33.4	2.8+- 38.8
Coordinate Based	91.7+- 2.6	9.6+- 0.1	-113.1+- 45.4	-76.7+- 25.2
Rotation-EQ	94.8+- 0.7	22.6+- 4.9	15.5+- 6.6	-38.1+- 9.5
Rotation-EQ-big	99.9+- 0.0	47.6+- 1.4	45.7+- 4.8	-4.5+- 13.2
Spatial Transformer	99.6+- 0.1	23.6+- 6.4	37.7+- 18.4	-2.2+- 0.6
RN50	99.9+- 0.1	23.4+- 1.5	40.3+- 1.3	14.9+- 10.2
RN101	99.9+- 0.1	19.7+- 1.0	37.8+- 3.7	9.8+- 11.7
DenseNet	99.9+- 0.0	33.2+- 0.5	51.4+- 3.0	4.0+- 6.5
RN50 (ImageNet-21k)	99.6+- 0.1	27.7+- 1.1	48.7+- 1.2	11.2+- 16.2
RN101 (ImageNet-21k)	99.8+- 0.1	39.2+- 3.7	39.2+- 7.6	5.6+- 10.5
DenseNet (ImageNet-1k)	62.9+- 51.9	41.8+- 2.6	73.7+- 4.9	34.3+- 0.5

Table 10: R^2**-score on MPI3D-Toy**



(a) Scatter scale OOD

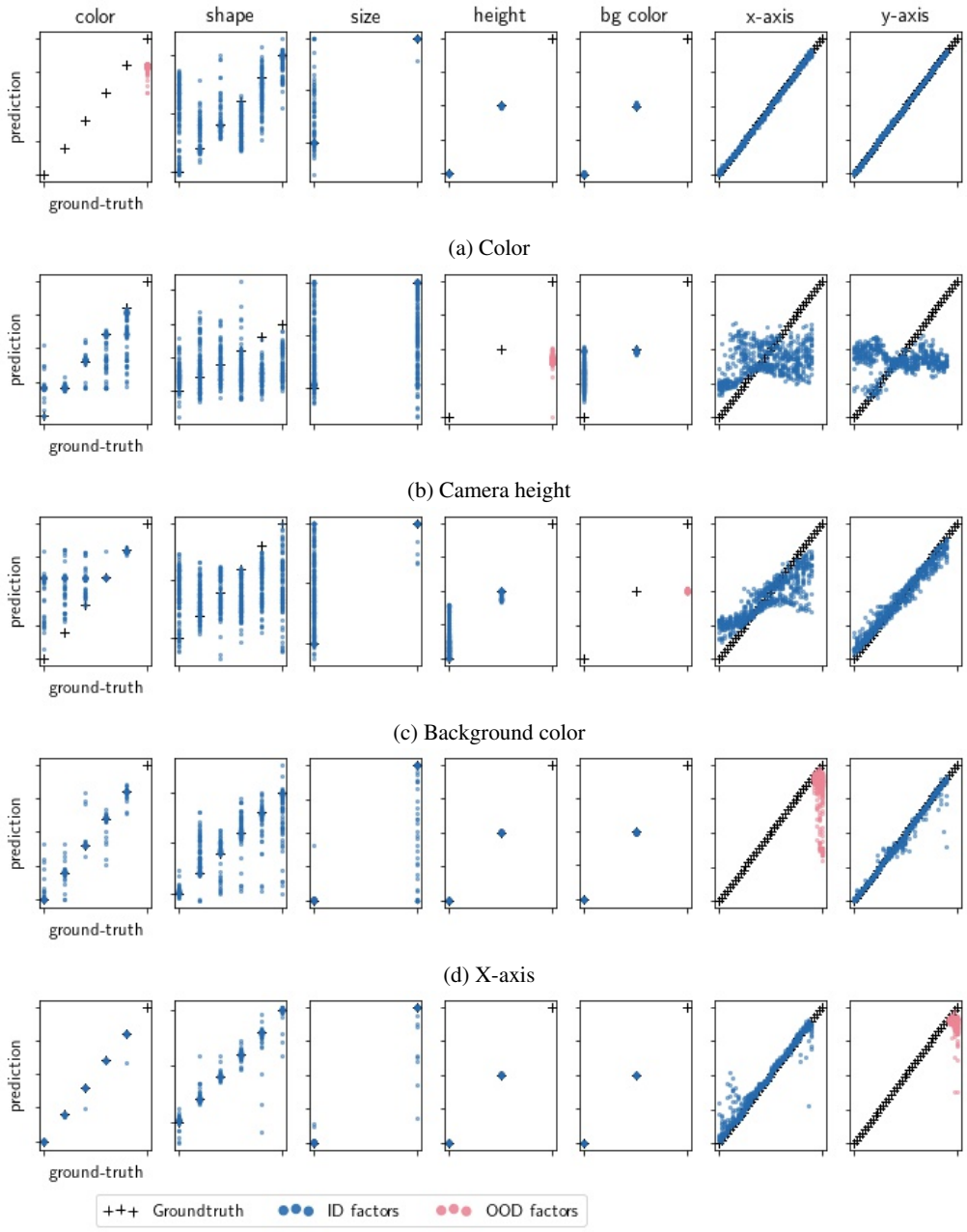
(b) Wall color OOD

(c) Object color

(d) Scale

(e) Orientation

Figure 15: **Shapes3D extrapolation.** We show the qualitative extrapolation of a CNN model. The shape category is excluded because no order is clear.



(a) Color

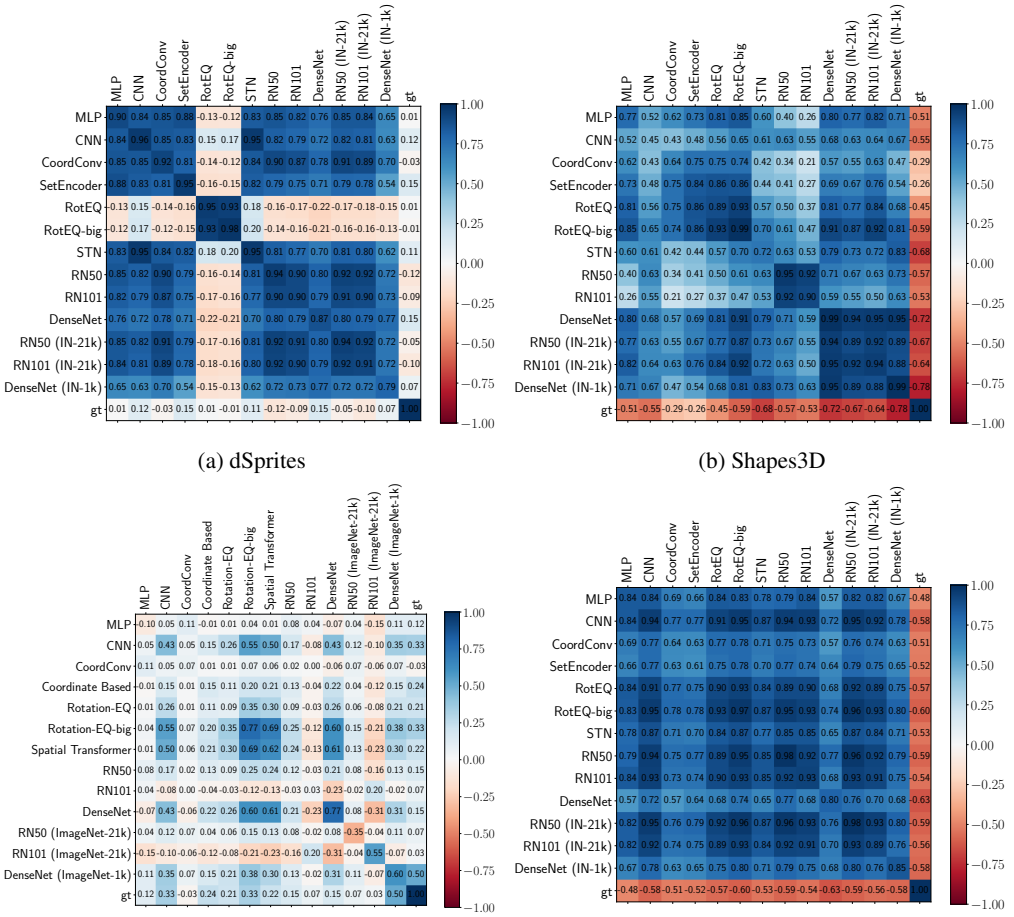
(b) Camera height

(c) Background color

(d) X-axis

(e) Y-axis

Figure 16: **MPI3D-Real extrapolation.** We show the qualitative extrapolation of a CNN model. The shape category is excluded because no order is clear. Size is excluded because only two values are available.



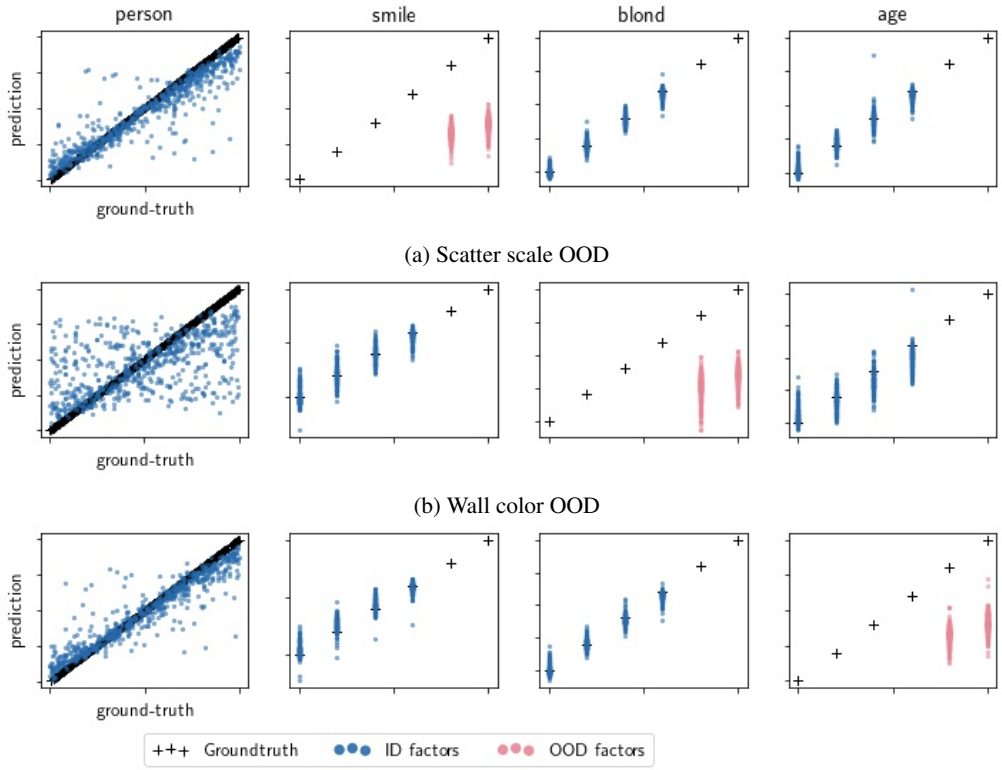
(a) dSprites

(b) Shapes3D

(c) CelebGlow

(d) MPI3D-Real

Figure 17: **Model similarity on extrapolation errors.**



(a) Scatter scale OOD

(b) Wall color OOD

(c) Object color

Figure 18: **CelebGlow extrapolation.** We show the qualitative extrapolation of a DenseNet (ImageNet 1-K) model. This corresponds, to the model with the highest correlation with the ground truth in Fig. 17 on the extrapolated factors (OOd factors). The person category is not extrapolated and used to measure correlations because no order is apparent.

Appendix C

Publication 3:
A simple way to make neural networks robust against diverse image corruptions

Published as a conference paper and as an oral at the ECCV 2020.

A simple way to make neural networks robust against diverse image corruptions

Evgenia Rusak[1,2*], Lukas Schott[1,2*], Roland S. Zimmermann[1,2*], Julian Bitterwolf[2], Oliver Bringmann[1†], Matthias Bethge[1,2†], and Wieland Brendel[1,2†]

[1] University of Tübingen
[2] International Max Planck Research School for Intelligent Systems
[*] joint first / [†] joint senior authors
{first.last}@uni-tuebingen.de

Abstract. The human visual system is remarkably robust against a wide range of naturally occurring variations and corruptions like rain or snow. In contrast, the performance of modern image recognition models strongly degrades when evaluated on previously unseen corruptions. Here, we demonstrate that a simple but properly tuned training with additive Gaussian and Speckle noise generalizes surprisingly well to unseen corruptions, easily reaching the state of the art on the corruption benchmark ImageNet-C (with ResNet50) and on MNIST-C. We build on top of these strong baseline results and show that an adversarial training of the recognition model against locally correlated worst-case noise distributions leads to an additional increase in performance. This regularization can be combined with previously proposed defense methods for further improvement.

Keywords: Image corruptions, robustness, generalization, adversarial training

1 Introduction

While Deep Neural Networks (DNNs) have surpassed the functional performance of humans in a range of complex cognitive tasks [12], [44], [38], [2], [30], they still lag behind humans in numerous other aspects. One fundamental shortcoming of machines is their lack of robustness against input perturbations. Even minimal perturbations that are hardly noticeable for humans can derail the predictions of high-performance neural networks.

For the purpose of this paper, we distinguish between two types of input perturbations. One type are minimal image-dependent perturbations specifically designed to fool a neural network with the smallest possible change to the input. These so-called *adversarial perturbations* have been the subject of hundreds of papers in the past five years, see e.g. [39], [21], [35], [11]. Another, much less studied type are *common corruptions*. These perturbations occur naturally

in many applications and include simple Gaussian or Salt and Pepper noise; natural variations like rain, snow or fog; and compression artifacts such as those caused by JPEG encoding. All of these corruptions do not change the semantic content of the input, and thus, machine learning models should not change their decision-making behavior in their presence. Nonetheless, high-performance neural networks like ResNet50 [12] are easily confused by small deformations [1]. The juxtaposition of adversarial examples and common corruptions was explored in [8] where the authors discuss the relationship between both and encourage researchers working in the field of adversarial robustness to cross-evaluate the robustness of their models towards common corruptions.

We argue that in many practical applications, robustness to common corruptions is often more relevant than robustness to artificially designed adversarial perturbations. Autonomous cars should not change their behavior in the face of unusual weather conditions such as hail or sand storms or small pixel defects in their sensors. Not-Safe-For-Work filters should not fail on images with unusual compression artifacts. Likewise, speech recognition algorithms should perform well regardless of the background music or sounds.

Besides its practical relevance, robustness to common corruptions is also an excellent target in its own right for researchers in the field of adversarial robustness and domain adaptation. Common corruptions can be seen as distributional shifts or as a weak form of adversarial examples that live in a smaller, constrained subspace.

Despite their importance, common corruptions have received relatively little attention so far. Only recently, a modification of the ImageNet dataset [34] to benchmark model robustness against common corruptions and perturbations has been published [13] and is referred to as ImageNet-C. Now, this scheme has also been applied to other common datasets resulting in Pascal-C, Coco-C and Cityscapes-C [25] and MNIST-C [29].

Our contributions are as follows:

- We demonstrate that data augmentation with Gaussian or Speckle noise serves as a simple yet very strong baseline that is sufficient to surpass almost all previously proposed defenses against common corruptions on ImageNet-C for ResNet50. We further show that the magnitude of the additive noise is a crucial hyper-parameter to reach optimal robustness.
- Motivated by our strong results with baseline noise augmentations, we introduce a neural network-based *adversarial noise generator* that can learn arbitrary uncorrelated noise distributions that maximally fool a given recognition network when added to their inputs. We denote the resulting noise patterns as *adversarial noise*.
- We design and validate a constrained Adversarial Noise Training (ANT) scheme through which the recognition network learns to become robust against adversarial i.i.d. noise. We demonstrate that our ANT reaches state-of-the-art robustness on the corruption benchmark ImageNet-C for the commonly used ResNet50 architecture and on MNIST-C, even surpassing the already strong baseline noise augmentations. This result is not due to overfitting on the

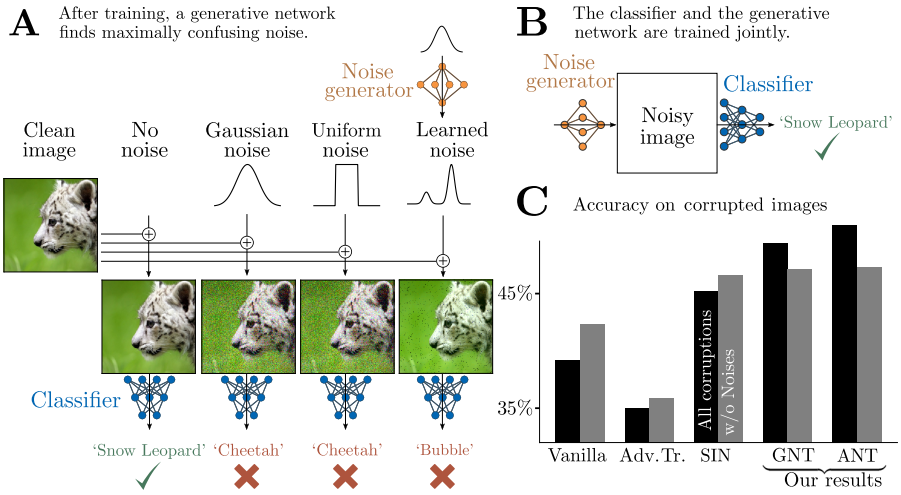


Fig. 1. Outline of our approach. A: First, we train a generative network against a vanilla trained classifier to find the adversarial noise. B: To achieve robustness against adversarial noise, we train the classifier and the noise generator jointly. C: We measure the robustness against common corruptions for a vanilla, adversarially trained (Adv. Tr.), trained on Stylized ImageNet (SIN), trained via Gaussian data augmentation (GNT) and trained with the means of Adversarial Noise Training (ANT). With our methods, we achieve the highest accuracy on common corruptions, both on all and non-noise categories.

noise categories of the respective benchmarks since we find equivalent results on the non-noise corruptions as well.

- We extend the adversarial noise generator towards locally correlated noise thereby enabling it to learn more diverse noise distributions. Performing ANT with the modified noise generator, we observe an increase in robustness for the 'snow' corruption which is visually similar to our learned noise.
- We demonstrate a further increase in robustness when combining ANT with previous defense methods.
- We substantiate the claim that increased robustness against regular or universal adversarial perturbations does not imply increased robustness against common corruptions. This is not necessarily true vice-versa: Our noise trained recognition network has high accuracy on ImageNet-C and also slightly improved accuracy on adversarial attacks on clean ImageNet compared to a vanilla trained ResNet50.

We released our model weights along with the full training code on GitHub. [1]

[1] github.com/bethgelab/game-of-noise

2 Related work

Robustness against common corruptions Several recent publications study the vulnerability of DNNs to common corruptions.

Two recent studies compare humans and DNNs on recognizing corrupted images, showing that DNN performance drops much faster than human performance for increased perturbation sizes [5], [10]. Hendrycks et al. introduce corrupted versions of standard datasets denoted as ImageNet-C, Tiny ImageNet-C and CIFAR10-C as standardized benchmarks for machine learning models [13]. Similarly, common corruptions have been applied to and evaluated on COCO-C, Pascal-C, Cityscapes-C [25] and MNIST-C [29].

There have been attempts to increase robustness against common corruptions. Zhang et al. integrate an anti-aliasing module from the signal processing domain in the ResNet50 architecture to restore the shift-equivariance which can get lost in deep CNNs and report an increased accuracy on clean data and better generalization to corrupted image samples [45]. Concurrent work to ours demonstrates that having more training data [43], [22] or using stronger backbones [43], [25], [18] can significantly improve model performance on common corruptions.

A popular method to decrease overfitting and help the network generalize better to unseen data is to augment the training dataset by applying a set of (randomized) manipulations to the images [26]. Furthermore, augmentation methods have also been applied to make the models more robust against image corruptions [9]. Augmentation with Gaussian [8], [19] or uniform noise [10] has been tried to increase model robustness. Conceptually, Ford et al. is the closest study to our work, since they also apply Gaussian noise to images to increase corruption robustness [8]. They use a different architecture (InceptionV3 versus our ResNet50). Also, they train a new model from scratch solely on images perturbed by Gaussian noise whereas we fine-tune a pretrained model on a mixture of clean and noisy images. They observe a low relative improvement in accuracy on corrupted images whereas we were able to outperform all previous baselines on the commonly used ResNet50 architecture.[2] Lopes et al. restrict the Gaussian noise to small image patches, which improves accuracy but does not yield state-of-the-art performance on the ResNet50 architecture [19]. Geirhos et al. train ImageNet classifiers against a fixed set of corruptions but find no generalized robustness against unseen corruptions [10]. However, they considered vastly higher noise levels than us. Considering the efficacy of Gaussian or uniform data augmentation to increase model robustness, the main difference to our work is that other works have used either munch larger [10] or smaller [8], [19] values for the standard deviation σ. A too large σ leads to an overfitting to the used noise distribution whereas a too small σ leads to noise levels that are not different enough from the clean images. We show that taking σ from the intermediate regime works best for generalization both to other noise types and non-noise corruptions.

[2] To compare with Ford et al., we evaluate our approach for an InceptionV3 architecture, see our results in Appendix H.

Link between adversarial robustness and common corruptions There is currently no agreement on whether adversarial training increases robustness against common corruptions in the literature. Hendrycks et al. report a robustness increase on common corruptions due to adversarial logit pairing on Tiny ImageNet-C [13]. Ford et al. suggest a link between adversarial robustness and robustness against common corruptions, claim that increasing one robustness type should simultaneously increase the other, but report mixed results on MNIST and CIFAR10-C [8]. Additionally, they also observe large drops in accuracy for adversarially trained networks and networks trained with Gaussian data augmentation compared to a vanilla classifier on certain corruptions. On the other hand, Engstrom et al. report that increasing robustness against adversarial ℓ_∞ attacks does not increase robustness against translations and rotations, but they do not present results on noise [7]. Kang et al. study robustness transfer between models trained against ℓ_1, ℓ_2, ℓ_∞ adversaries / elastic deformations and JPEG artifacts [17]. They observe that adversarial training increases robustness against elastic and JPEG corruptions on a 100-class subset of ImageNet. This result contradicts our findings on full ImageNet as we see a slight decline in accuracy on those two classes for the adversarially trained model from [42] and severe drops in accuracy on other corruptions. Jordan et al. show that adversarial robustness does not transfer easily between attack classes [16]. Tramèr et al. [40] also argue in favor of a trade-off between different robustness types. For a simple and natural classification task, they prove that adversarial robustness towards l_∞ perturbations does neither transfer to l_1 nor to input rotations and translations, and vice versa and support their formal analysis with experiments on MNIST and CIFAR10.

3 Methods

3.1 Training with Gaussian noise

As discussed in section 2, several researchers have tried using Gaussian noise as a method to increase robustness towards common corruptions with mixed results. In this work, we revisit the approach of Gaussian data augmentation and increase its efficacy. We treat the standard deviation σ of the distribution as a hyper-parameter of the training and measure its influence on robustness.

To formally introduce the objective, let $\mathcal{D}$ be the data distribution over input pairs $(\boldsymbol{x}, y)$ with $\boldsymbol{x} \in \mathbb{R}^N$ and $y \in \{1, ..., k\}$. We train a differentiable classifier $f_\theta(\boldsymbol{x})$ by minimizing the risk on a dataset with additive Gaussian noise

$$\mathop{\mathbb{E}}_{\boldsymbol{x},y\sim\mathcal{D}} \mathop{\mathbb{E}}_{\boldsymbol{\delta}\sim\mathcal{N}(\mathbf{0},\sigma^2\mathbb{1})} \left[\mathcal{L}_{\mathrm{CE}}\left(f_\theta(\mathrm{clip}(\boldsymbol{x}+\boldsymbol{\delta})), y\right)\right], \tag{1}$$

where σ is the standard deviation of the Gaussian noise and $\boldsymbol{x} + \boldsymbol{\delta}$ is clipped to the input range $[0, 1]^N$. The standard deviation is either kept fixed or is chosen uniformly from a fixed set of standard deviations. In both cases, the possible standard deviations are chosen from a small set of nine values inspired by the

noise variance in the ImageNet-C dataset (cf. section 3.3). To maintain high accuracy on clean data, we only perturb 50% of the training data with Gaussian noise within each batch.

3.2 Adversarial noise

Learning Adversarial Noise Our goal is to find a noise distribution $p_\phi(\boldsymbol{\delta})$, $\boldsymbol{\delta} \in \mathbb{R}^N$ such that noise samples added to $\boldsymbol{x}$ maximally confuse the classifier f_θ. More concisely, we optimize

$$\max_{\phi} \mathop{\mathbb{E}}_{\boldsymbol{x},y\sim\mathcal{D}} \mathop{\mathbb{E}}_{\boldsymbol{\delta}\sim p_\phi(\boldsymbol{\delta})} \left[\mathcal{L}_{\text{CE}}\left(f_\theta(\text{clip}(\boldsymbol{x}+\boldsymbol{\delta})), y\right)\right], \tag{2}$$

where clip is an operator that clips all values to the valid interval (i.e. $\text{clip}(\boldsymbol{x}+\boldsymbol{\delta}) \in [0,1]^N$) and restricts their norm $||\boldsymbol{\delta}||_2 = \epsilon$. [3]

We follow the literature of implicit generative models [28], [4] as we do not have to explicitly model the probability density function $p_\phi(\boldsymbol{\delta})$ since optimizing Eq. (2) only involves samples drawn from $p_\phi(\boldsymbol{\delta})$. We model the samples from $p_\phi(\boldsymbol{\delta})$ as the output of a neural network $g_\phi : \mathbb{R}^N \to \mathbb{R}^N$ which gets its input from a normal distribution $\boldsymbol{\delta} = g_\phi(\boldsymbol{z})$ where $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \mathbb{1})$. We enforce the independence property of $p_\phi(\boldsymbol{\delta}) = \prod_n p_\phi(\delta_n)$ by constraining the network architecture of the noise generator g_ϕ to only consist of convolutions with 1x1 kernels. Lastly, the projection onto a sphere $||\boldsymbol{\delta}||_2 = \epsilon$ is achieved by scaling the generator output with a scalar while clipping $\boldsymbol{x} + \boldsymbol{\delta}$ to the valid range $[0,1]^N$. This fixed size projection (hyper-parameter) is motivated by the fact that Gaussian noise training with a single, fixed σ achieved the highest accuracy. [4]

The noise generator g_ϕ has four 1x1 convolutional layers with ReLU activations and one residual connection from input to output. The weights of the layers are initialized to small numbers; for this initialization, the input is passed through the residual connection to the output. Since we use Gaussian noise as input, the noise generator outputs Gaussian noise at initialization. During training, the weights change and the generator learns to produce more diverse distributions.

Adversarial Noise Training To increase robustness, we now train the classifier f_θ to minimize the risk under adversarial noise distributions jointly with the noise generator

$$\min_{\theta}\max_{\phi} \mathop{\mathbb{E}}_{\boldsymbol{x},y\sim\mathcal{D}} \mathop{\mathbb{E}}_{\boldsymbol{\delta}\sim p_\phi(\boldsymbol{\delta})} \left[\mathcal{L}_{\text{CE}}\left(f_\theta(\text{clip}(\boldsymbol{x}+\boldsymbol{\delta})), y\right)\right], \tag{3}$$

where again $\boldsymbol{x} + \boldsymbol{\delta} \in [0,1]^N$ and $||\boldsymbol{\delta}||_2 = \epsilon$. For a joint adversarial training, we alternate between an outer loop of classifier update steps and an inner loop of

[3] We apply the method derived in [32] and rescale the perturbation by a factor γ to obtain the desired ℓ_2 norm; despite the clipping, the squared ℓ_2 norm is a piece-wise linear function of γ^2 that can be inverted to find the correct scaling factor γ.

[4] We also experimented with an adaptive sphere radius ϵ which grows with the classifier's accuracy. However, we did not see any improvements and followed Occam's razor.

generator update steps. Note that in regular adversarial training, e.g. [21], $\boldsymbol{\delta}$ is optimized directly whereas we optimize a constrained distribution over $\boldsymbol{\delta}$.

To maintain high classification accuracy on clean samples, we sample every mini-batch so that they contain 50% clean data and perturb the rest. The current state of the noise generator is used to perturb 30% of this data and the remaining 20% are augmented with samples chosen randomly from previous distributions. For this, the noise generator states are saved at regular intervals. The latter method is inspired by experience replay from reinforcement learning [27] and is used to keep the classifier from forgetting previous adversarial noise patterns. To prevent the noise generator from being stuck in a local minimum, we halt the Adversarial Noise Training (ANT) at regular intervals and train a new noise generator from scratch. This noise generator is trained against the current state of the classifier to find a current optimum. The new noise generator replaces the former noise generator in the ANT. This technique has been crucial to train a robust classifier.

Learning locally correlated adversarial noise We modify the architecture of the noise generator defined in Eq. 2 to allow for local spatial correlations and thereby enable the generator to learn more diverse distributions. Since we seek to increase model robustness towards image corruptions such as rain or snow that produce locally correlated patterns, it is natural to include local patterns in the manifold of learnable distributions. We replace the 1x1 kernels in one network layer with 3x3 kernels limiting the maximum correlation length of the output noise sample to 3x3 pixels. We indicate the correlation length of noise generator used for the constrained adversarial noise training as $\text{ANT}^{1\text{x}1}$ or $\text{ANT}^{3\text{x}3}$.

Combining Adversarial Noise Training with stylization As demonstrated by [9], using random stylization as data augmentation increases the accuracy on ImageNet-C due to a higher shape bias of the model. We combine our ANT and the stylization approach to achieve robustness gains from both in the following way: we split the samples in each batch into clean data (25%), stylized data (30%) and clean data perturbed by the noise generator (45%).

3.3 Evaluation on corrupted images

Evaluation of noise robustness We evaluate the robustness of a model by sampling a Gaussian noise vector $\boldsymbol{\delta}$ (covariance $\mathbb{1}$). We then do a line search along the direction $\boldsymbol{\delta}$ starting from the original image $\boldsymbol{x}$ until it is misclassified. We denote the resulting minimal perturbation as $\boldsymbol{\delta}_{\text{min}}$. The robustness of a model is then denoted by the median[5] over the test set

$$\epsilon^* = \underset{\boldsymbol{x},y\sim\mathcal{D}}{\text{median}}\ ||\boldsymbol{\delta}_{\text{min}}||_2, \tag{4}$$

[5] Samples for which no ℓ_2-distance allows us to manipulate the classifier's decision contribute a value of ∞ to the median.

with $f_\theta(\boldsymbol{x} + \boldsymbol{\delta}_{\min}) \neq y$ and $\boldsymbol{x} + \boldsymbol{\delta}_{\min} \in [0,1]^N$. Note that a higher ϵ^* denotes a more robust classifier. To test the robustness against adversarial noise, we train a new noise generator at the end of the Adversarial Noise Training until convergence and evaluate it according to Eq. (4).

ImageNet-C The ImageNet-C benchmark[6] [13] is a conglomerate of 15 diverse corruption types that were applied to the validation set of ImageNet. The corruptions are organized into four main categories: noise, blur, weather, and digital. The MNIST-C benchmark is created similarly to ImageNet-C with a slightly different set of corruptions [29]. We report the Top-1 and Top-5 accuracies as well as the 'mean Corruption Error' (mCE) on both benchmarks. We evaluate all proposed methods for ImageNet-C on the ResNet50 architecture for better comparability to previous methods, e.g. [9], [19], [45]. The clean ImageNet accuracy of the used architecture highly influences the results and could be seen as an upper bound for the accuracy on ImageNet-C. Note that our approach is independent of the used architecture and could be applied to any differentiable network.

4 Results

For our experiments on ImageNet, we use a classifier that was pretrained on ImageNet. For the experiments on MNIST, we use the architecture from [21] for comparability. All technical details, hyper-parameters and the architectures of the noise generators can be found in Appendix A-B. We use various open source software packages for our experiments, most notably Docker [24], scipy and numpy [41], PyTorch [31] and torchvision [23].

(In-)Effectiveness of regular adversarial training to increase robustness towards common corruptions In our first experiment, we evaluate whether robustness against regular adversarial examples generalizes to robustness against common corruptions. We display the Top-1 accuracy of vanilla and adversarially trained models in Table 1; detailed results on individual corruptions can be found in Appendix C. For all tested models, we find that regular ℓ_∞ adversarial training can strongly decrease the robustness towards common corruptions, especially for the corruption types Fog and Contrast. Universal adversarial training [37], on the other hand, leads to severe drops on some corruptions but the overall accuracy on ImageNet-C is slightly increased relative to the vanilla baseline model (AlexNet). Nonetheless, the absolute ImageNet-C accuracy of 22.2% is still very low. These results disagree with two previous studies which

[6] For the evaluation, we use the JPEG compressed images from github.com/hendrycks/robustness as is advised by the authors to ensure reproducibility. We note that Ford et al. report a decrease in performance when the compressed JPEG files are used as opposed to applying the corruptions directly in memory without compression artifacts [8].

Table 1: Top-1 accuracy on ImageNet-C and ImageNet-C without the noise category (higher is better). Regular adversarial training decreases robustness towards common corruptions; universal adversarial training seems to slightly increase it.

Model	IN-C	IN-C w/o noises
Vanilla RN50	39.2%	42.3%
Adv. Training [36]	29.1%	32.0%
Vanilla RN152	45.0%	47.9%
Adv. Training [42]	35.0%	35.9%
Vanilla AlexNet	21.1%	23.9%
Universal Adv. Training [37]	22.2%	23.1%

reported that (1) adversarial logit pairing[7] (ALP) increases robustness against common corruptions on Tiny ImageNet-C [13], and that (2) adversarial training can increase robustness on CIFAR10-C [8].

We evaluate adversarially trained models on MNIST-C and present the results and their discussion in Appendix E. The results on MNIST-C show the same tendency as on ImageNet-C: adversarially trained models have lower accuracy on MNIST-C and thus indicate that adversarial robustness does not transfer to robustness against common corruptions. This corroborates the results of Ford et al. [8] on MNIST who also found that an adversarially robust model had decreased robustness towards a set of common corruptions.

Effectiveness of Gaussian data augmentation to increase robustness towards common corruptions We fine-tune ResNet50 classifier pretrained on ImageNet with Gaussian data augmentation from the distribution $\mathcal{N}(\mathbf{0}, \sigma^2\mathbb{1})$ and vary σ. We try two different settings: in one, we choose a single noise level σ while in the second, we sample σ uniformly from a set of multiple possible values. The Top-1 accuracy of the fine-tuned models on ImageNet-C in comparison to a vanilla trained model is shown in Fig. 2. Each black point shows the performance of one model fine-tuned with one specific σ; the vanilla trained model is marked by the point at $\sigma = 0$. The horizontal lines indicate that the model is fine-tuned with Gaussian noise where σ is sampled from a set for each image. For example, for the dark green line, as indicated by the stars, we sample σ from the set $\{0.08, 0.12, 0.18, 0.26, 0.38\}$ which corresponds to the Gaussian corruption of ImageNet-C. Since Gaussian noise is part of the test set, we show both the results on the full ImageNet-C evaluation set and the results on ImageNet-C without noises (namely blur, weather and digital). To show how the different σ-levels manifest themselves in an image, we include example images in Appendix G.

There are three important results evident from Fig. 2:

[7] Note that ALP was later found to not increase adversarial robustness [6].

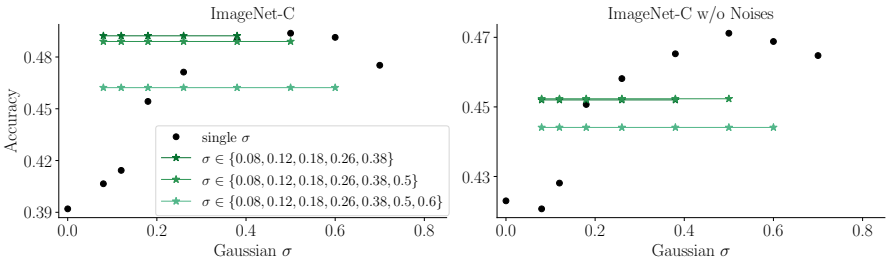


Fig. 2. Top-1 accuracy on ImageNet-C (left) and ImageNet-C without the noise corruptions (right) of a ResNet50 architecture fine-tuned with Gaussian data augmentation of varying σ. Each dot or green line represents one model. We train on Gaussian noise sampled from a distribution with a single σ (black dots) and on distributions where σ is sampled from different sets (green lines with stars). We also compare to a vanilla trained model at $\sigma = 0$.

1. Gaussian noise generalizes well to the non-noise corruptions of the ImageNet-C dataset and is a powerful baseline. This is surprising as it was shown in several recent works that training on Gaussian or uniform noise does not generalize to other corruption types [10], [19] or that the effect is weak [8].
2. The standard deviation σ is a crucial hyper-parameter and has an optimal value of about $\sigma = 0.5$ for ResNet50.
3. If σ is chosen well, using a single σ is enough and sampling from a set of σ values is detrimental for robustness against non-noise corruptions.

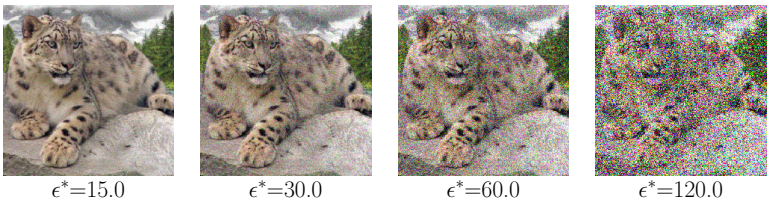
In the following Results sections, we will compare Gaussian data augmentation to our Adversarial Noise Training approach and baselines from the literature. For this, we will use the models with the overall best-performance: The model $\text{GN}_{0.5}$ that was trained with Gaussian data augmentation with a single $\sigma = 0.5$ and the model GN_{mult} where σ was sampled from the set $\{0.08, 0.12, 0.18, 0.26, 0.38\}$.

Evaluation of the severity of adversarial noise as an attack In this section, we focus on the question: Can we learn the most severe uncorrelated additive noise distribution for a classifier? Following the success of simple uncorrelated Gaussian noise data augmentation (section 4) and the ineffectiveness of regular adversarial training (section 4) which allows for highly correlated patterns, we restrict our learned noise distribution to be sampled independently for each pixel.

To measure the effectiveness of our adversarial noise, we report the median perturbation size ϵ^* that is necessary for a misclassification for each image in the test set as defined in section 3.3. We find $\epsilon^*_{\text{GN}} = 39.0$ for Gaussian noise, $\epsilon^*_{\text{UN}} = 39.1$ for uniform noise and $\epsilon^*_{\text{AN}} = 15.7$ for adversarial noise (see Fig. 1 for samples of each noise type). Thus, we see that our AN is much more effective at fooling the classifier compared to Gaussian and uniform noise.

Table 2: Accuracy on clean data and robustness of differently trained models as measured by the median perturbation size ϵ^*. A higher ϵ^* indicates a more robust model. We compute standard deviations for ϵ^*_{AN} for differently initialized generator networks. To provide an intuition for the perturbation sizes indicated by ϵ^*, we show example images for Gaussian noise below and a larger Figure for different noise types in Appendix I.

model	clean acc.	ϵ^*_{GN}	ϵ^*_{UN}	$\epsilon^*_{\text{AN1x1}}$
Vanilla RN50	**76.1%**	39.0	39.1	15.7 ± 0.6
GNT$\sigma_{0.5}$	75.9%	74.8	74.9	31.8 ± 3.9
GNT$_{\text{mult}}$	**76.1%**	130.1	130.7	24.0 ± 2.2
ANT$^{\text{1x1}}$	76.0%	**136.7**	**137.0**	**95.4 ± 5.7**



ϵ^*=15.0 ϵ^*=30.0 ϵ^*=60.0 ϵ^*=120.0

Evaluation of Adversarial Noise Training as a defense In the previous section, we established a method for learning the most adversarial noise distribution for a classifier. Now, we utilize it for a joint Adversarial Noise Training (ANT$^{\text{1x1}}$) where we simultaneously train the noise generator and classifier (see section 3.2). This leads to substantially increased robustness against Gaussian, uniform and adversarial noise, see Table 2. The robustness of models that were trained via Gaussian data augmentation also increases, but on average much less compared to the model trained with ANT$^{\text{1x1}}$. To evaluate the robustness against adversarial noise, we train four noise generators with different random seeds and measure $\epsilon^*_{\text{AN1x1}}$. We report the mean value and the standard deviation over the four runs. To visualize this effect, we visualize the temporal evolution of the probability density function $p_\phi(\delta_n)$ of uncorrelated noise during ANT$^{\text{1x1}}$ in Fig. 3A. This shows that the generator converges to different distributions and therefore, the classifier has been trained against a rich variety of distributions.

Comparison of different methods to increase robustness towards common corruptions We now revisit common corruptions on ImageNet-C and compare the robustness of differently trained models. Since Gaussian noise is part of ImageNet-C, we train another baseline model with data augmentation using the Speckle noise corruption from the ImageNet-C holdout set. We later denote the cases where the corruptions present during training are part of the test set by putting corresponding accuracy values in brackets. Additionally, we compare our results with several baseline models from the literature:

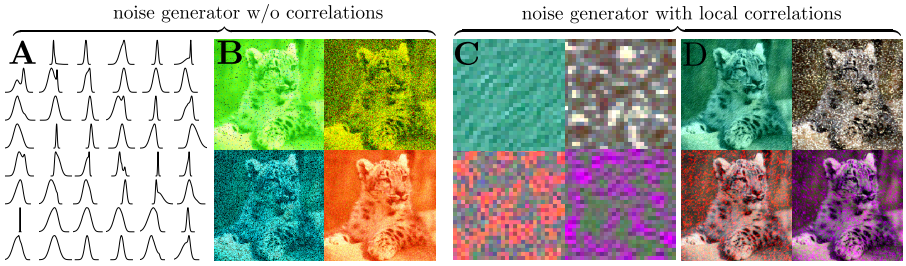


Fig. 3. A: Examples of learned probability densities over the grayscale version of the noise $\boldsymbol{\delta}_n$ during ANT$^{1\text{x}1}$ where each density corresponds to one local minimum; B: Example images with sampled uncorrelated adversarial noise; C: Example patches of locally correlated noise with a size of 28x28 pixels learned during ANT$^{3\text{x}3}$; D: Example images with sampled correlated adversarial noise.

1. Shift Inv: The model is modified to enhance shift-equivariance using anti-aliasing [45].[8]
2. Patch GN: The model was trained on Gaussian patches [19].[9]
3. SIN+IN: The model was trained on a stylized version of ImageNet [9].[10]
4. AugMix: [14] trained their model using diverse augmentations.[11] They use image augmentations from AutoAugment [3] and exclude contrast, color, brightness, sharpness, and Cutout operations to make sure that the test set of ImageNet-C is disjoint from the training set. We would like to highlight the difficulty in clearly distinguishing between the augmentations used during training and testing as there might be a certain overlap. This can be seen by the visual similarity between the Posterize operation and the JPEG corruption (see Appendix J).

The Top-1 accuracies on the full ImageNet-C dataset and ImageNet-C without the noise corruptions are displayed in Table 3; detailed results on individual corruptions in terms of accuracy and mCE are shown in Tables 3 and 4, Appendix D. We also calculate the accuracy on corruptions without the noise category since we observe that the generated noise can sometimes be close to the i.i.d. corruptions of ImageNet-C raising concerns about overfitting. Additionally, the expressiveness of the generated i.i.d. noise is quite limited compared to natural corruptions like 'snow'. We hence extend the ANT$^{1\text{x}1}$ procedure to include spatially correlated noise over 3x3 pixels. Samples are shown in Fig. 3C and Fig. 3D.

The results on full ImageNet-C are striking (see Table 3): a very simple baseline, namely a model trained with Speckle noise data augmentation, beats

[8] Weights were taken from github.com/adobe/antialiased-cnns.
[9] Since no model weights are released, we include the values reported in their paper.
[10] Weights were taken from github.com/rgeirhos/texture-vs-shape.
[11] Weights were taken from github.com/google-research/augmix.

Table 3: Average accuracy on clean data, average Top-1 and Top-5 accuracies on ImageNet-C and ImageNet-C without the noise category (higher is better); all values in percent. We compare the results obtained by the means of Gaussian (GNT) and Speckle noise data augmentation and with Adversarial Noise Training (ANT) to several baselines. Gray numbers in brackets indicate scenarios where a corruption from the test set was used during training.

	IN	IN-C		IN-C w/o noises	
model	clean acc.	Top-1	Top-5	Top-1	Top-5
Vanilla RN50	76.1	39.2	59.3	42.3	63.2
Shift Inv [45]	77.0	41.4	61.8	44.2	65.1
Patch GN [19]	76.0	(43.6)	(n.a.)	43.7	n.a.
SIN+IN [9]	74.6	45.2	66.6	46.6	68.2
AugMix [14]	**77.5**	48.3	69.2	50.4	71.8
Speckle	75.8	46.4	67.6	44.5	65.5
GNT_{mult}	76.1	(49.2)	(70.2)	45.2	66.2
$\text{GNT}\sigma_{0.5}$	75.9	(49.4)	(70.6)	47.1	68.3
$\text{ANT}^{1\text{x}1}$	76.0	(51.1)	(72.2)	47.7	68.8
$\text{ANT}^{1\text{x}1}$+SIN	74.9	(52.2)	(73.6)	49.2	70.6
$\text{ANT}^{1\text{x}1}$ w/o EP	75.7	(48.9)	(70.2)	46.5	67.7
$\text{ANT}^{3\text{x}3}$	76.1	50.4	71.5	47.0	68.1
$\text{ANT}^{3\text{x}3}$+SIN	74.1	**52.6**	**74.4**	**50.6**	**72.5**

almost all previous baselines reaching an accuracy of 46.4% which is larger than the accuracy of SIN+IN (45.2%) and close to AugMix (48.3%). The $\text{GN}\sigma_{0.5}$ surpasses SIN+IN not only on the noise category but also on almost all other corruptions, see a more detailed breakdown in Table 3, Appendix D.

The $\text{ANT}^{3\text{x}3}$+SIN model produces the best results on ImageNet-C both with and without noises. Thus, it is slightly superior to Gaussian data augmentation and pure $\text{ANT}^{3\text{x}3}$. Comparing $\text{ANT}^{1\text{x}1}$ and $\text{ANT}^{3\text{x}3}$, we observe that $\text{ANT}^{3\text{x}3}$ performs better than $\text{ANT}^{1\text{x}1}$ on the 'snow' corruption. We attribute this to the successful modeling capabilities of locally correlated patterns resembling snow of the 3x3 noise generator. We perform an ablation study to investigate the necessity of experience replay and note that we lose roughly 2% without it ($\text{ANT}^{1\text{x}1}$ w/o EP vs $\text{ANT}^{1\text{x}1}$). We also test how the classifier's performance changes if it is trained against adversarial noise sampled randomly from $p_\phi(\delta_n)$. The accuracy on ImageNet-C decreases slightly compared to regular $\text{ANT}^{1\text{x}1}$: 51.1%/ 71.9% (Top-1/ Top-5) on full ImageNet-C and 47.3%/ 68.3% (Top-1/ Top-5) on ImageNet-C without the noise category. We include additional results for $\text{ANT}^{1\text{x}1}$ with a DenseNet121 architecture [15] and for varying parameter counts of the noise generator in Appendix K.

For MNIST, we train a model with Gaussian data augmentation and via $\text{ANT}^{1\text{x}1}$. We achieve similar results with both approaches and report a new state-of-the-art accuracy on MNIST-C: 92.4%, see Appendix E for details.

Table 4: Adversarial robustness on ℓ_2 ($\epsilon = 0.12$) and ℓ_∞ ($\epsilon = 0.001$) compared to a Vanilla ResNet50 on ImageNet.

model	clean acc. [%]	ℓ_2 acc. [%]	ℓ_∞ acc. [%]
Vanilla RN50	**76.1**	41.1	18.1
GNT$\sigma_{0.5}$	75.9	49.0	28.1
ANT$^{1\text{x}1}$	76.0	50.1	28.6
Adv. Training [36]	60.5	**58.1**	**58.5**

Robustness towards adversarial perturbations As regular adversarial training can decrease the accuracy on common corruptions, it is also interesting to check what happens vice-versa: How does a model which is robust on common corruptions behave under adversarial attacks?

Both our ANT$^{1\text{x}1}$ and GNT models have slightly increased ℓ_2 and ℓ_∞ robustness scores compared to a vanilla trained model, see Table 4. We tested this using the white-box attacks PGD [20] and DDN [33]. Expectedly, an adversarially trained model has higher adversarial robustness compared to ANT$^{1\text{x}1}$ or GNT. In this experiment, we only verify that we do not unintentionally reduce adversarial robustness compared to a vanilla ResNet50. For details, see Appendix E for MNIST and Appendix F for ImageNet.

5 Conclusions

So far, attempts to use simple noise augmentations for general robustness against common corruptions have produced mixed results, ranging from no generalization from one noise to other noise types [10] to only marginal robustness increases [8], [19]. In this work, we demonstrate that carefully tuned additive noise patterns in conjunction with training on clean samples can surpass almost all current state-of-the-art defense methods against common corruptions. By drawing inspiration from adversarial training and experience replay, we additionally show that training against simple uncorrelated or locally correlated worst-case noise patterns outperforms our already strong baseline defense, with additional gains to be made in combination with previous defense methods like stylization [9].

There are still a few corruption types (e.g. Motion or Zoom blurs) on which our method is not state of the art, suggesting that additional gains are possible. Future extensions of this work may combine noise generators with varying correlation lengths, add additional interactions between noise and image (e.g. multiplicative interactions or local deformations) or take into account local image information in the noise generation process to further boost robustness across many types of image corruptions.

References

1. Azulay, A., Weiss, Y.: Why do deep convolutional networks generalize so poorly to small image transformations? (2018)
2. Campbell, M., Hoane, Jr., A.J., Hsu, F.h.: Deep blue. Artif. Intell. **134**(1-2), 57–83 (Jan 2002). https://doi.org/10.1016/S0004-3702(01)00129-1, http://dx.doi.org/10.1016/S0004-3702(01)00129-1
3. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation policies from data. arXiv preprint arXiv:1805.09501 (2018)
4. Diggle, P.J., Gratton, R.J.: Monte carlo methods of inference for implicit statistical models. Journal of the Royal Statistical Society: Series B (Methodological) **46**(2), 193–212 (1984)
5. Dodge, S.F., Karam, L.J.: A study and comparison of human and deep learning recognition performance under visual distortions. CoRR **abs/1705.02498** (2017), http://arxiv.org/abs/1705.02498
6. Engstrom, L., Ilyas, A., Athalye, A.: Evaluating and understanding the robustness of adversarial logit pairing. CoRR **abs/1807.10272** (2018), https://arxiv.org/abs/1807.10272
7. Engstrom, L., Tsipras, D., Schmidt, L., Madry, A.: A rotation and a translation suffice: Fooling cnns with simple transformations. ICML (2019)
8. Ford, N., Gilmer, J., Carlini, N., Cubuk, D.: Adversarial examples are a natural consequence of test error in noise. ICML (2019)
9. Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W.: Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=Bygh9j09KX
10. Geirhos, R., Temme, C.R.M., Rauber, J., Schütt, H.H., Bethge, M., Wichmann, F.A.: Generalisation in humans and deep neural networks. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 31, pp. 7538–7550. Curran Associates, Inc. (2018), http://papers.nips.cc/paper/7982-generalisation-in-humans-and-deep-neural-networks.pdf
11. Gilmer, J., Metz, L., Faghri, F., Schoenholz, S.S., Raghu, M., Wattenberg, M., Goodfellow, I.J.: Adversarial spheres. CoRR **abs/1801.02774** (2018), http://arxiv.org/abs/1801.02774
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
13. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=HJz6tiCqYm
14. Hendrycks, D., Mu, N., Cubuk, E.D., Zoph, B., Gilmer, J., Lakshminarayanan, B.: Augmix: A simple data processing method to improve robustness and uncertainty. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=S1gmrxHFvB
15. Huang, G., Liu, Z., Weinberger, K.Q.: Densely connected convolutional networks. CVPR (2017)
16. Jordan, M., Manoj, N., Goel, S., Dimakis, A.G.: Quantifying perceptual distortion of adversarial examples. arXiv preprint arXiv:1902.08265 (2019)

17. Kang, D., Sun, Y., Brown, T., Hendrycks, D., Steinhardt, J.: Transfer of adversarial robustness between perturbation types. CoRR **abs/1905.01034** (2019), http://arxiv.org/abs/1905.01034
18. Lee, J., Won, T., Hong, K.: Compounding the performance improvements of assembled techniques in a convolutional neural network. arXiv preprint arXiv:2001.06268 (2020)
19. Lopes, R.G., Yin, D., Poole, B., Gilmer, J., Cubuk, E.D.: Improving robustness without sacrificing accuracy with patch gaussian augmentation. CoRR **abs/1906.02611** (2019), http://arxiv.org/abs/1906.02611
20. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
21. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: International Conference on Learning Representations (2018), https://openreview.net/forum?id=rJzIBfZAb
22. Mahajan, D., Girshick, R., Ramanathan, V., He, K., Paluri, M., Li, Y., Bharambe, A., van der Maaten, L.: Exploring the limits of weakly supervised pretraining. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 181–196 (2018)
23. Marcel, S., Rodriguez, Y.: Torchvision the machine-vision package of torch. In: ACM International Conference on Multimedia (2010)
24. Merkel, D.: Docker: Lightweight linux containers for consistent development and deployment. Linux J. **2014**(239) (Mar 2014)
25. Michaelis, C., Mitzkus, B., Geirhos, R., Rusak, E., Bringmann, O., Ecker, A.S., Bethge, M., Brendel, W.: Benchmarking robustness in object detection: Autonomous driving when winter is coming. arXiv preprint arXiv:1907.07484 (2019)
26. Mikołajczyk, A., Grochowski, M.: Data augmentation for improving deep learning in image classification problem. 2018 International Interdisciplinary PhD Workshop (IIPhDW) pp. 117–122 (2018)
27. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529 (2015)
28. Mohamed, S., Lakshminarayanan, B.: Learning in implicit generative models. arXiv preprint arXiv:1610.03483 (2016)
29. Mu, N., Gilmer, J.: MNIST-C: A robustness benchmark for computer vision. arXiv preprint arXiv:1906.02337 (2019)
30. OpenAI: Openai five. https://blog.openai.com/openai-five/ (2018)
31. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in PyTorch. In: NIPS Autodiff Workshop (2017)
32. Rauber, J., Bethge, M.: Fast differentiable clipping-aware normalization and rescaling. arXiv preprint arXiv:2007.07677 (2020), https://github.com/jonasrauber/clipping-aware-rescaling
33. Rony, J., Hafemann, L.G., Oliveira, L.S., Ayed, I.B., Sabourin, R., Granger, E.: Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4322–4330 (2019)
34. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M.S., Berg, A.C., Li, F.: Imagenet large scale visual recognition challenge. CoRR **abs/1409.0575** (2014), http://arxiv.org/abs/1409.0575

35. Schott, L., Rauber, J., Bethge, M., Brendel, W.: Towards the first adversarially robust neural network model on MNIST. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=S1EHOsC9tX
36. Shafahi, A., Najibi, M., Ghiasi, A., Xu, Z., Dickerson, J., Studer, C., Davis, L.S., Taylor, G., Goldstein, T.: Adversarial training for free! arXiv preprint arXiv:1904.12843 (2019)
37. Shafahi, A., Najibi, M., Xu, Z., Dickerson, J.P., Davis, L.S., Goldstein, T.: Universal adversarial training. CoRR **abs/1811.11304** (2018), http://arxiv.org/abs/1811.11304
38. Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L.R., Lai, M., Bolton, A., Chen, Y., Lillicrap, T.P., Hui, F.F.C., Sifre, L., van den Driessche, G., Graepel, T., Hassabis, D.: Mastering the game of go without human knowledge. Nature **550**, 354–359 (2017)
39. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
40. Tramèr, F., Boneh, D.: Adversarial training and robustness for multiple perturbations. NeurIPS (2019), http://arxiv.org/abs/1904.13000
41. Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Jarrod Millman, K., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C., Polat, İ., Feng, Y., Moore, E.W., Vand erPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P., Contributors, S...: SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods **17**, 261–272 (2020). https://doi.org/https://doi.org/10.1038/s41592-019-0686-2
42. Xie, C., Wu, Y., van der Maaten, L., Yuille, A.L., He, K.: Feature denoising for improving adversarial robustness. CVPR (2019)
43. Xie, Q., Hovy, E., Luong, M.T., Le, Q.V.: Self-training with noisy student improves imagenet classification. arXiv preprint arXiv:1911.04252 (2019)
44. Xiong, W., Droppo, J., Huang, X., Seide, F., Seltzer, M., Stolcke, A., Yu, D., Zweig, G.: Achieving human parity in conversational speech recognition. IEEE/ACM Transactions on Audio, Speech, and Language Processing (2016)
45. Zhang, R.: Making convolutional networks shift-invariant again. ICML (2019)

Appendix to Increasing the robustness of DNNs against image corruptions by playing the Game of Noise

A Architectures of the noise generators

The architectures of the noise generators are displayed in Tables 1 and 2. The number of color channels is indicated by C. The noise generator displayed in Table 1 only uses kernels with a size of 1 and thus produces spatially uncorrelated noise. With the stride being 1 and no padding, the spatial dimensions are preserved in each layer. The noise generator displayed in Table 2 has one layer with 3x3 convolutions and thus produces noise samples with a correlation length of 3x3 pixels.

Layer	Shape
Conv + ReLU	$20 \times 1 \times 1$
Conv + ReLU	$20 \times 1 \times 1$
Conv + ReLU	$20 \times 1 \times 1$
Conv	C $\times 1 \times 1$

Table 1: Architecture of the noise generator producing uncorrelated noise.

Layer	Shape
Conv + ReLU	$20 \times 1 \times 1$
Conv + ReLU	$20 \times 3 \times 3$
Conv + ReLU	$20 \times 1 \times 1$
Conv	C $\times 1 \times 1$

Table 2: Architecture of the noise generator producing locally correlated noise.

B Implementation details and hyper-parameters

We use PyTorch [7] for all of our experiments.

Preprocessing MNIST images are preprocessed such that their pixel values lie in the range $[0, 1]$. Preprocessing for ImageNet is performed in the standard way for PyTorch ImageNet models from the model zoo by subtracting the mean $[0.485, 0.456, 0.406]$ and dividing by the standard deviation $[0.229, 0.224, 0.225]$. We add Gaussian, adversarial and Speckle noise before the preprocessing step, so the noisy images are first clipped to the range $[0, 1]$ of the raw images and then preprocessed before being fed into the model.

ImageNet experiments For all ImageNet experiments, we used a pretrained ResNet50 architecture from `https://pytorch.org/docs/stable/torchvision/models.html`. We fine-tuned the model with SGD-M using an initial learning rate of 0.001, which corresponds to the last learning rate of the PyTorch model training, and a momentum of 0.9. After convergence, we decayed the learning rate once by a factor of 10 and continued the training. Decaying the learning rate was highly beneficial for the model performance. We tried decaying the learning rate a second time, but this did not bring any benefits in any of our experiments. For GNT, we also tried training from scratch, i.e. starting with a large learning rate of 0.1 and random weights, and trained for 120 epochs, but we got worse results compared to merely fine-tuning the model provided by torchvision. We used a batch size of 70 for all our experiments. We have also tried to use the batch sizes 50 and 100, but did not observe any difference.

Gaussian noise We trained the models until convergence. The total number of training epochs varied between 30 and 90 epochs.

Speckle noise We used the Speckle noise implementation from `https://github.com/hendrycks/robustness/blob/master/ImageNet-C/create_c/make_imagenet_c.py`, line 270. The model trained with Speckle noise converged faster than with Gaussian data augmentation and therefore, we only trained the model for 10 epochs.

Adversarial Noise Training The adversarial noise generator was trained with the Adam optimizer with a learning rate of 0.0001. We have replaced the noise generator every 0.33 epochs. For $\text{ANT}^{1\text{x}1}$, we set the ϵ-sphere to control the size of the perturbation to 135.0 which on average corresponds to the ℓ_2-size of a perturbation caused by additive Gaussian noise sampled from $\mathcal{N}(0, 0.5^2 \cdot \mathbb{1})$. We have trained the classifier until convergence for 80 epochs. For $\text{ANT}^{3\text{x}3}$, we set the ϵ-sphere to 70.0 and trained the classifier for 80 epochs. We decreased the ϵ-sphere for $\text{ANT}^{3\text{x}3}$ to counteract giving the noise generator more degrees of freedom to fool the classifier to maintain a similar training losses and accuracies for $\text{ANT}^{1\text{x}1}$ and $\text{ANT}^{3\text{x}3}$.

MNIST experiments For the MNIST experiments, we used the same model architecture as [6] for our $\text{ANT}^{1\text{x}1}$ and GNT. For $\text{ANT}^{1\text{x}1}$, our learning rate for the generator was between 10^{-4} and 10^{-5}, and equal to 10^{-3} for the classifier. We used a batch size of 300. As an optimizer, we used SGD-M with a momentum of 0.9 for the classifier and Adam [5] for the generator. The splitting of batches in clean, noisy and history was equivalent to the ImageNet experiments. The optimal ϵ hyper-parameter was determined with a line search similar to the optimal σ of the Gaussian noise; we found $\epsilon = 10$ to be optimal. The parameters for the Gaussian noise experiments were equivalent. Both models were trained until convergence (around 500-600 epochs). GNT and $\text{ANT}^{1\text{x}1}$ were performed on a pretrained network.

C Detailed results on the evaluation of corruption robustness due to regular adversarial training

We find that standard adversarial training against minimal adversarial perturbations in general does not increase robustness against common corruptions. While some early results on CIFAR-10 by [1] and Tiny ImageNet-C by [3] suggest that standard adversarial training might increase robustness to common corruptions, we here observe the opposite: Adversarially trained models have lower robustness against common corruptions. An adversarially trained ResNet152 with an additional denoising layer[1] from [12] has lower accuracy across almost all corruptions except Snow and Pixelations. On some corruptions, the accuracy of the adversarially trained model decreases drastically, e.g. from 49.1% to 4.6% on Fog or 42.8% to 9.3% on Contrast. Similarly, the adversarially trained ResNet50[2] from [Shafahi et al., 2019] shows a substantial decrease in performance on common corruptions compared with a vanilla trained model.

An evaluation of a robustified version of AlexNet[2] [10] that was trained with the Universal Adversarial Training scheme on ImageNet-C shows that achieving robustness against universal adversarial perturbations does not noticeably increase robustness towards common corruptions (22.2%) compared with a vanilla trained model (21.1%).

[1] Model weights from `https://github.com/facebookresearch/ImageNet-Adversarial-Training`
[2] Model weights were kindly provided by the authors.

		Noise (Compressed)			Blur (Compressed)			
Model	All	Gaussian	Shot	Impulse	Defocus	Glass	Motion	Zoom
Vanilla RN50	39.2	29.3	27.0	23.8	38.7	26.8	38.7	36.2
AT [9]	29.1	20.5	19.1	12.4	21.4	30.8	30.4	31.4
Vanilla RN152	45.0	35.7	34.3	29.6	45.1	32.8	48.4	40.5
AT [12]	35.0	35.2	34.4	24.8	22.1	31.7	30.9	32.0
Vanilla AlexNet	21.1	11.4	10.6	7.7	18.0	17.4	21.4	20.2
UAT [10]	22.2	20.1	19.1	16.2	13.1	21.6	19.7	19.2

	Weather (Compressed)				Digital (Compressed)			
Model	Snow	Frost	Fog	Brightness	Contrast	Elastic	Pixelate	JPEG
Vanilla RN50	32.5	38.1	45.8	68.0	39.1	45.2	44.8	53.4
AT [9]	24.4	25.6	5.8	51.1	7.8	45.4	53.4	56.3
Vanilla RN152	38.7	43.9	49.1	71.2	42.8	51.1	50.5	60.5
AT [12]	42.0	40.4	4.6	58.8	9.3	47.2	54.1	58.0
Vanilla AlexNet	13.3	17.3	18.1	43.5	14.7	35.4	28.2	39.4
UAT [10]	13.8	18.3	4.3	36.5	4.8	36.8	42.3	47.1

Table 3: Average Top-1 accuracy over 5 severities of common corruptions on ImageNet-C in percent. A high accuracy on a certain corruption type indicates high robustness of a classifier on this corruption type, so higher accuracy is better. Adversarial training (AT) decreases the accuracy on common corruptions, especially on the corruptions Fog and Contrast. Universal Adversarial Training (UAT) slightly increases the overall performance.

D Detailed ImageNet-C results

We show detailed results on individual corruptions in Table 4 in accuracy and in Table 5 in mCE for differently trained models. In Fig. 1, we show the degradation of accuracy for different severity levels. To avoid clutter, we only show results for a vanilla trained model, for the previous state of the art SIN+IN [2], for several Gaussian trained models and for the overall best model ANT$^{3\text{x}3}$+SIN.

The Corruption Error [3] is defined as

$$\text{CE}_c^f = \left(\sum_{s=1}^{5} E_{s,c}^f\right) \Big/ \left(\sum_{s=1}^{5} E_{s,c}^{\text{AlexNet}}\right), \tag{1}$$

where $E_{s,c}^f$ is the Top-1 error of a classifier f for a corruption c with severity s. The mean Corruption error (mCE) is taken by averaging over all corruptions.

		Noise			Blur				Weather				Digital			
model	mean	Gauss	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixel	Jpeg
Vanilla RN50	39	29	27	24	39	27	39	36	33	38	46	68	39	45	45	53
Shift Inv	42	36	34	30	40	29	38	39	33	40	48	68	42	45	49	57
Patch GN	44	45	43	42	38	26	39	38	30	39	54	67	39	52	47	56
SIN+IN	45	41	40	37	43	32	45	36	41	42	47	67	43	50	56	58
AugMix	48	41	41	38	**48**	35	**54**	**49**	40	44	47	69	**51**	52	57	60
Speckle	46	55	58	49	43	32	40	36	34	41	46	68	41	47	49	58
GNT$_{\text{mult}}$	49	**67**	65	**64**	43	33	41	37	34	42	45	68	41	48	50	60
GNT$\sigma_{0.5}$	49	58	59	57	47	38	43	42	35	44	44	68	39	50	55	**62**
ANT$^{1\text{x}1}$	51	65	**66**	**64**	47	37	43	40	36	46	44	**70**	43	49	55	**62**
ANT$^{1\text{x}1}$+SIN	52	64	65	63	46	38	46	39	42	47	49	69	47	50	57	60
ANT$^{1\text{x}1}$ w/o EP	49	59	59	57	46	37	43	40	34	43	43	68	39	49	55	61
ANT$^{3\text{x}3}$	50	65	64	**64**	44	36	42	38	39	46	44	69	41	49	55	61
ANT$^{3\text{x}3}$+SIN	**53**	62	61	60	41	**39**	46	37	**48**	**52**	**55**	68	49	**53**	**59**	59

Table 4: Average Top-1 accuracy over 5 severities of common corruptions on ImageNet-C in percent obtained by different models; higher is better.

E MNIST-C results

Similar to the ImageNet-C experiments, we are interested how vanilla, adversarially and noise trained models perform on MNIST-C.

The adversarially robust MNIST model by [11] was trained with a robust loss function and is among the state of the art in certified adversarial robustness. The other baseline models were trained with Adversarial Training in ℓ_2 (DDN) by [8] and ℓ_∞ (PGD) by [6]. Our GNT and ANT$^{1\text{x}1}$ trained versions are trained as described in the main paper and Appendix B.2. The results are shown in Table 6. Similar to ImageNet-C, the models trained with GNT and ANT$^{1\text{x}1}$ are significantly better than our vanilla trained baseline. Also, regular adversarial training has severe drops and does not lead to significant robustness improvements.

As for ImageNet and GNT, we have treated σ as a hyper-parameter. The accuracy on MNIST-C for different values of σ is displayed in Fig. 2 and has a maximum around $\sigma = 0.5$ like for ImageNet.

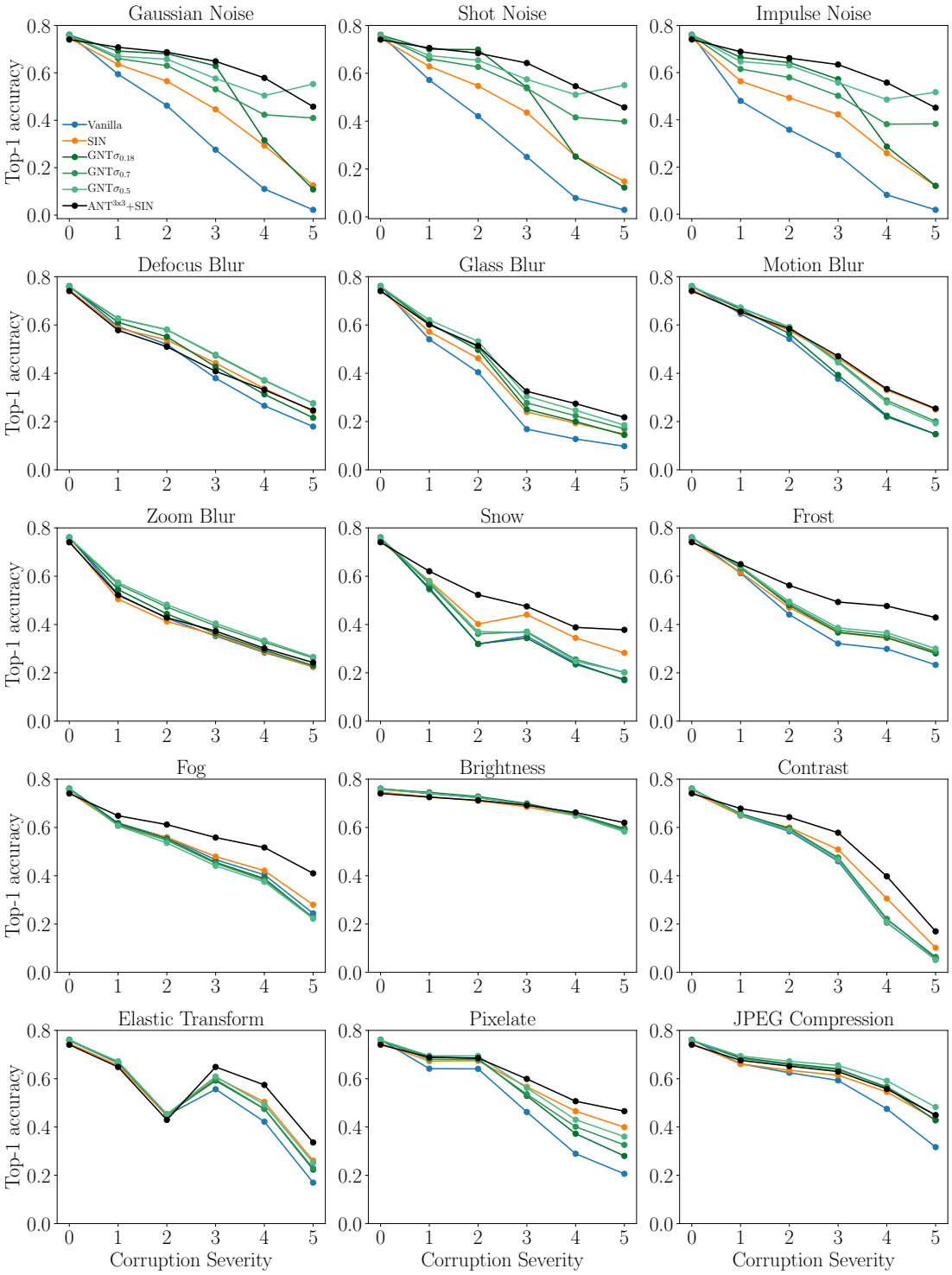


Fig. 1: Top-1 accuracy for each corruption type and severity on ImageNet-C.

		Noise			Blur				Weather				Digital			
model	mCE	Gauss	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixel	Jpeg
Vanilla	77	80	82	83	75	89	78	80	78	75	66	57	71	85	77	77
SIN	69	66	67	68	70	82	69	80	68	71	65	58	66	78	62	70
Patch GN	71	62	63	62	75	90	78	78	81	74	57	59	71	74	74	72
Shift Inv.	73	73	74	76	74	86	78	77	77	72	63	56	68	86	71	71
AugMix	65	67	66	68	**64**	79	**59**	**64**	69	68	65	54	**57**	74	60	65
Speckle	68	51	47	55	70	83	77	80	76	71	66	57	70	82	71	69
GNT_{mult}	65	**37**	39	**39**	69	81	76	79	76	70	67	56	69	81	69	66
$GNT\sigma_{0.5}$	64	46	46	47	65	75	72	74	75	68	69	57	71	78	63	63
ANT^{1x1}	62	39	**38**	**39**	65	77	72	75	74	66	68	**53**	67	78	62	**62**
ANT^{1x1}+SIN	**61**	40	39	40	65	76	69	76	67	64	62	55	63	77	59	66
ANT^{1x1} w/o EP	65	46	46	47	66	76	73	75	76	69	70	57	72	79	63	64
ANT^{3x3}	63	39	40	**39**	68	78	73	77	71	66	68	55	69	79	63	64
ANT^{3x3}+SIN	**61**	43	44	43	71	**74**	69	79	**60**	**58**	**55**	56	59	**73**	**57**	67

Table 5: Average mean Corruption Error (mCE) obtained by different models on common corruptions from ImageNet-C; lower is better.

model	clean acc	mean	Shot	Impulse	Glass Blur	Motion Blur	Shear	Scale	Rotate	Brightness	Translate	Stripe	Fog	Splatter	Dotted Line	Zig Zag	Canny Edges
Vanilla	99.1	86.9	98	96	96	94	98	95	92	88	57	88	50	97	96	86	72
[6]	98.5	75.6	98	55	94	94	97	88	92	27	53	40	63	96	78	74	84
Vanilla	98.8	74.3	98	91	96	88	95	80	89	34	45	41	23	96	96	80	63
[11]	98.2	68.6	97	65	93	93	94	87	89	11	40	20	25	96	89	61	68
Vanilla	99.5	89.8	98	96	95	97	98	96	94	95	61	89	79	98	98	90	63
DDN Tr [8]	99.0	87.0	99	97	96	94	98	91	93	72	55	92	64	99	98	91	66
Vanilla	99.1	86.9	98	96	96	94	98	95	92	88	57	88	50	97	96	86	72
$GNT\sigma_{0.5}$	99.3	92.4	99	99	98	97	98	95	93	98	56	91	91	99	99	96	78
ANT^{1x1}	99.4	92.4	99	99	98	97	98	95	93	98	55	89	91	99	99	96	80

Table 6: Accuracy in percent for the MNIST-C dataset for adversarially robust ([11], [6], DDN [8]) and our noise trained models (GNT and ANT^{1x1}). Vanilla always denotes the same network architecture as its adversarially or noise trained counterpart but with standard training. Note that we used the same network architecture as [6].

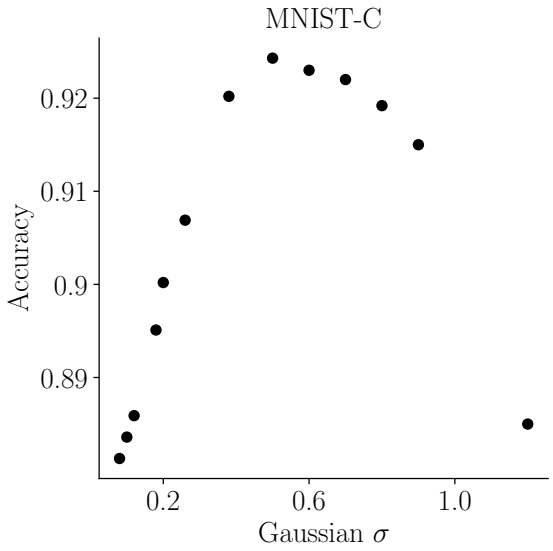


Fig. 2: Average accuracy on MNIST-C over all severties and corruptions for different values of sigma σ of the Gaussian noise training (GNT) during training. Each point corresponds to one converged training.

F Evaluation of adversarial robustness of models trained via GNT and ANT^{1x1}

ImageNet To evaluate adversarial robustness on ImageNet, we used PGD [6] and DDN [8]. For the ℓ_∞ PGD attack, we allowed for 200 iterations with a step size of 0.0001 and a maximum sphere size of 0.001. For the DDN ℓ_2 attack, we also allowed for 200 iterations, set the sphere adjustment parameter γ to 0.02 and the maximum epsilon to 0.125. We note that for both attacks increasing the number of iterations from 100 to 200 did not make a significant difference in robustness of our tested models. The results on adversarial robustness on ImageNet can be found in the main paper in Table 4.

MNIST To evaluate adversarial robustness on MNIST, we also used PGD [6] and DDN [8]. For the ℓ_∞ PGD attack, we allowed for 100 iterations with a step size of 0.01 and a maximum sphere size of 0.1. For the DDN ℓ_2 attack, we also allowed for 100 iterations, set the sphere adjustment parameter γ to 0.05 and the maximum epsilon to 1.5. All models have the same architecture as [6]. The results on adversarial robustness on MNIST can be found in Table 7.

model	clean acc. [%]	ℓ_2 acc. [%]	ℓ_∞ acc. [%]
Vanilla	99.1	73.2	55.8
GNT$_{\sigma_{0.5}}$	99.3	89.2	73.6
ANT^{1x1}	99.4	90.4	76.3

Table 7: Adversarial robustness on MNIST on ℓ_2 ($\epsilon = 1.5$) and ℓ_∞ ($\epsilon = 0.1$) compared to a Vanilla CNN.

G Example images for additive Gaussian noise

Example images with additive Gaussian noise of varying standard deviation σ are displayed in Fig. 3. The considered σ-levels correspond to those studied in section 4.2. in the main paper.

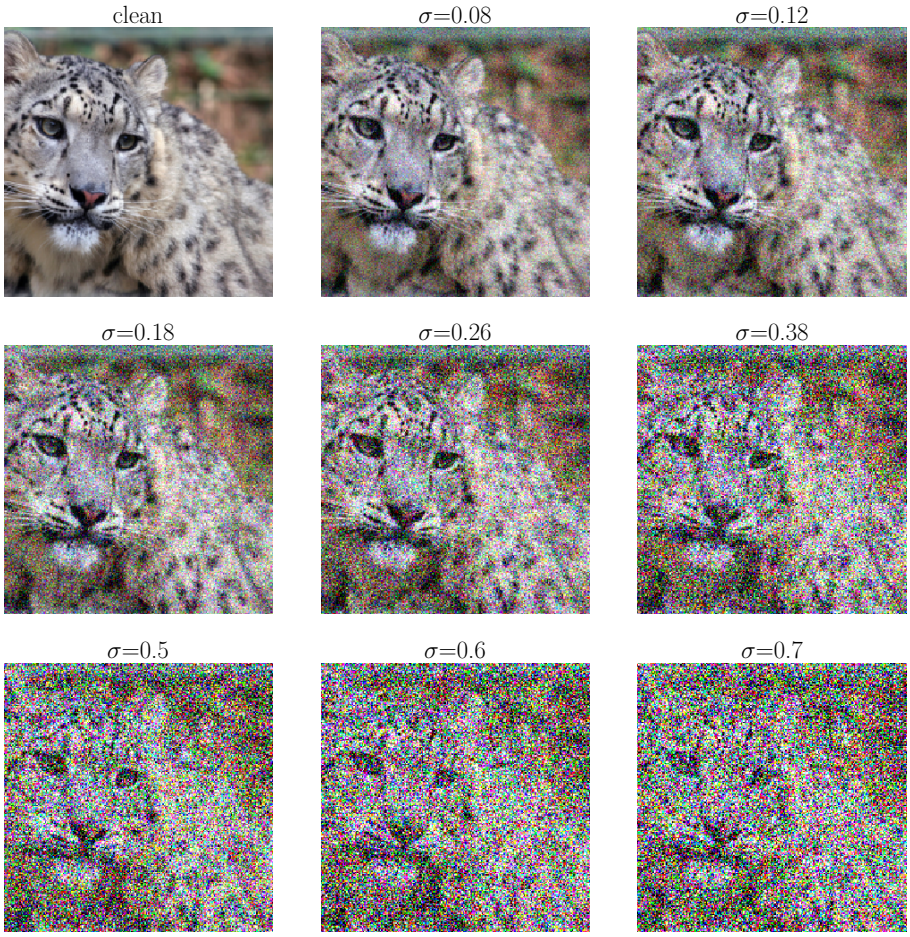


Fig. 3: Example images with different σ-levels of additive Gaussian noise on ImageNet.

H Comparison to Ford et al.

Ford et al. trained an InceptionV3 model from scratch both on clean data from the ImageNet dataset and on data augmented with Gaussian noise [1]. Since we use a very similar approach, we compare our approach to theirs directly. The results for comparison on ImageNet both for the vanilla and the Gaussian noise trained model can be found in Table 8. Since we use a pretrained model provided by PyTorch and fine-tune it instead of training a new one, the performance of our vanilla trained model differs from the performance of their vanilla trained model, both on clean data and on ImageNet-C. The accuracy on clean data is displayed in Table 9. Another difference between our training and theirs is that we split every batch evenly in clean and data augmented by Gaussian noise with one standard deviation whereas they sample σ uniformly between 0 and one specific value. With our training scheme, we were able to outperform their model significantly on all corruptions except for Elastic, Fog and Brightness.

		Noise (Compressed)			Blur (Compressed)			
model	All	Gaussian	Shot	Impulse	Defocus	Glass	Motion	Zoom
Vanilla InceptionV3 [1]	38.8	36.6	34.3	34.7	31.1	19.3	35.3	30.1
Gaussian ($\sigma = 0.4$) [1]	42.7	40.3	38.8	37.7	32.9	29.8	35.3	33.1
Vanilla InceptionV3 [ours]	41.6	42.0	40.3	38.5	33.5	27.1	36.1	28.8
GNT$\sigma_{0.4}$ [ours]	49.5	60.8	59.6	59.4	43.8	37.0	42.8	38.4
GNT$\sigma_{0.5}$ [ours]	**50.2**	**61.6**	**60.9**	**60.8**	**44.6**	**37.3**	**44.0**	**39.3**

	Weather (Compressed)				Digital (Compressed)			
model	Snow	Frost	Fog	Brightness	Contrast	Elastic	Pixelate	JPEG
Vanilla InceptionV3 [1]	33.1	34.0	52.4	66.0	35.9	47.8	38.2	50.0
Gaussian ($\sigma = 0.4$) [1]	36.6	43.5	**52.3**	**67.1**	35.8	**52.2**	47.0	55.5
Vanilla InceptionV3 [ours]	33.5	39.6	42.2	64.2	41.0	43.5	57.4	56.9
GNT$\sigma_{0.4}$ [ours]	35.6	43.7	43.3	64.8	43.0	49.0	59.3	61.7
GNT$\sigma_{0.5}$ [ours]	**37.1**	**44.2**	43.6	64.6	**43.3**	49.4	**59.6**	**61.9**

Table 8: ImageNet-C accuracy for InceptionV3.

model	clean accuracy [%]
Vanilla InceptionV3 [1]	75.9
Gaussian ($\sigma = 0.4$) [1]	74.2
Vanilla InceptionV3 [ours]	77.2
GNT$\sigma_{0.4}$ [ours]	78.1
GNT$\sigma_{0.5}$ [ours]	77.9

Table 9: Accuracy on clean data for differently trained models.

I Visualization of images with different perturbation sizes

In the main paper, we measure model robustness by calculating the median perturbation size ϵ^* and report the results in Table 2. To provide a better intuition for the noise level in an image for a particular ϵ^*, we display example images in Fig. 4.

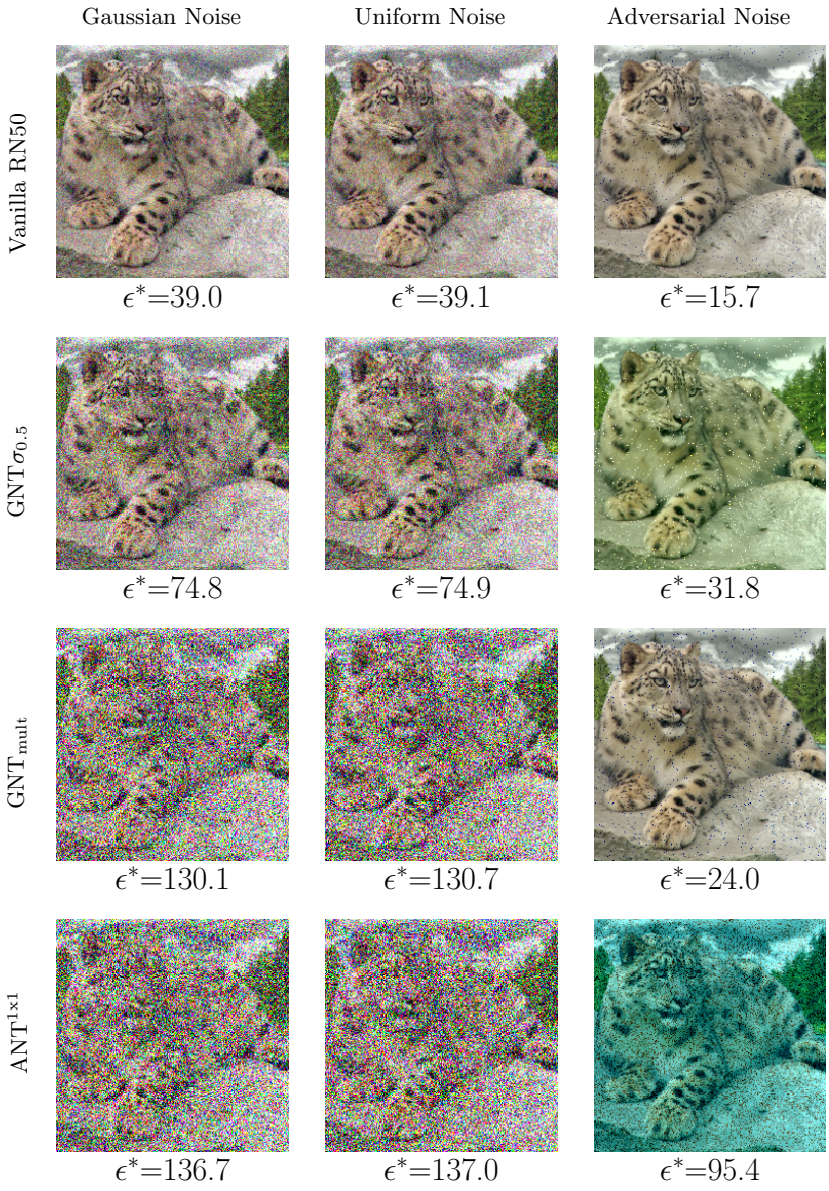


Fig. 4: Example images for the different perturbation sizes ϵ^* and different noise types on ImageNet corresponding to the ϵ^* values in Table 2 in the main paper.

J Visualization of Posterize vs JPEG

AugMix [4] uses Posterize as one of their operations for data augmentation during training. In Fig. 5, we show the visual similarity between the Posterize operation and the JPEG corruption from ImageNet-C.

Fig. 5: Example images for the JPEG compression from ImageNet-C and the `PIL.ImageOps.Posterize` operation.

K Additional results

Adversarial Noise Training with a DenseNet121 architecture To test in how far our results generalize to other backbones, we have trained a DenseNet121 model with $\text{ANT}^{1\text{x}1}$. The DenseNet121 model was finetuned from the checkpoint provided by torchvision. A DenseNet121 has $7.97886 \cdot 10^6$ trainable parameters whereas a ResNet50 has $2.5557 \cdot 10^7$. Our results and a comparison to $\text{ANT}^{1\text{x}1}$ with a ResNet50 model is shown in Table 10: $\text{ANT}^{1\text{x}1}$ increases robustness on full ImageNet-C and ImageNet-C without noises for the DenseNet121 model, showing that adversarial noise training generalizes to other backbones.

	IN	IN-C		IN-C w/o noises	
model	clean acc.	Top-1	Top-5	Top-1	Top-5
Vanilla RN50	76.1	39.2	59.3	42.3	63.2
$\text{ANT}^{1\text{x}1}$ RN50	76.0	(51.1)	(72.2)	47.7	68.8
Vanilla DN121	74.4	42.1	63.4	44.0	65.5
$\text{ANT}^{1\text{x}1}$ DN121	74.3	50.3	71.6	46.8	68.3

Table 10: Average accuracy on clean data, average Top-1 and Top-5 accuracies on full ImageNet-C and ImageNet-C without the noise category (higher is better); all values in percent. We compare the results obtained by $\text{ANT}^{1\text{x}1}$ for a ResNet50 (RN50) architecture to a DenseNet121 (DN121) architecture.

Results for different parameter counts of the noise generator Here, we study the effect of different parameter counts of the adversarial noise generator on $\text{ANT}^{1\text{x}1}$. We provide the results in Table 11. We indicate the depth of the noise generator with a subscript. All experiments in this paper apart from this ablation study were performed with a default depth of 4 layers. We observe that while depth is a tunable hyper-parameter, the performances of $\text{ANT}^{1\text{x}1}$ with the studied noise generators do not differ by a lot. Only the most shallow noise generator with a depth of one layer and only 12 trainable parameters results in a roughly 1% lower accuracy than its deeper counterparts. We note that a $\text{GNT}\sigma_{0.5}$ model has an accuracy of 49.4% on full ImageNet-C and an accuracy of 47.1% on ImageNet-C without noises which roughly corresponds to the respective accuracies of $\text{ANT}^{1\text{x}1}$ with the most shallow noise generator.

model	Number of parameters	IN clean acc.	IN-C Top-1	IN-C w/o noises Top-1
Vanilla RN50	-	76.1	39.2	42.3
$\text{ANT}^{1\text{x}1}$ RN50 NG_1	12	75.1	(49.5)	46.6
$\text{ANT}^{1\text{x}1}$ RN50 NG_2	143	75.5	(50.8)	47.2
$\text{ANT}^{1\text{x}1}$ RN50 NG_3	563	75.3	(50.7)	47.2
$\text{ANT}^{1\text{x}1}$ RN50 NG_4	983	76.0	(51.1)	47.7
$\text{ANT}^{1\text{x}1}$ RN50 NG_5	1403	74.0	(50.7)	47.0

Table 11: Number of trainable parameters of different noise generators, average accuracy on clean data, ImageNet-C and ImageNet-C without the noise category (higher is better); all values in percent. We compare the results obtained by $\text{ANT}^{1\text{x}1}$ with noise generators of different depth. Note that a depth of 4 layers was used in all experiments in this paper apart from this ablation study.

References

1. Ford, N., Gilmer, J., Carlini, N., Cubuk, D.: Adversarial examples are a natural consequence of test error in noise. ICML (2019)
2. Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W.: Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=Bygh9j09KX
3. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=HJz6tiCqYm
4. Hendrycks, D., Mu, N., Cubuk, E.D., Zoph, B., Gilmer, J., Lakshminarayanan, B.: Augmix: A simple data processing method to improve robustness and uncertainty. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=S1gmrxHFvB
5. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
6. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
7. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in PyTorch. In: NIPS Autodiff Workshop (2017)
8. Rony, J., Hafemann, L.G., Oliveira, L.S., Ayed, I.B., Sabourin, R., Granger, E.: Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4322–4330 (2019)
9. Shafahi, A., Najibi, M., Ghiasi, A., Xu, Z., Dickerson, J., Studer, C., Davis, L.S., Taylor, G., Goldstein, T.: Adversarial training for free! arXiv preprint arXiv:1904.12843 (2019)
10. Shafahi, A., Najibi, M., Xu, Z., Dickerson, J.P., Davis, L.S., Goldstein, T.: Universal adversarial training. CoRR **abs/1811.11304** (2018), http://arxiv.org/abs/1811.11304
11. Wong, E., Schmidt, F.R., Metzen, J.H., Kolter, J.Z.: Scaling provable adversarial defenses. CoRR **abs/1805.12514** (2018), http://arxiv.org/abs/1805.12514
12. Xie, C., Wu, Y., van der Maaten, L., Yuille, A.L., He, K.: Feature denoising for improving adversarial robustness. CVPR (2019)

Appendix D

Publication 4: Towards the first adversarially robust neural network model on MNIST

Published as a conference paper and as a poster at the ICLR 2019.

TOWARDS THE FIRST ADVERSARIALLY ROBUST NEURAL NETWORK MODEL ON MNIST

Lukas Schott[1-3*], Jonas Rauber[1-3*], Matthias Bethge[1,3,4†] & Wieland Brendel[1,3†]

[1]Centre for Integrative Neuroscience, University of Tübingen
[2]International Max Planck Research School for Intelligent Systems
[3]Bernstein Center for Computational Neuroscience Tübingen
[4]Max Planck Institute for Biological Cybernetics
[*]Joint first authors
[†]Joint senior authors
firstname.lastname@bethgelab.org

ABSTRACT

Despite much effort, deep neural networks remain highly susceptible to tiny input perturbations and even for MNIST, one of the most common toy datasets in computer vision, no neural network model exists for which adversarial perturbations are large and make semantic sense to humans. We show that even the widely recognized and by far most successful L_∞ defense by Madry et al. (1) has lower L_0 robustness than undefended networks and is still highly susceptible to L_2 perturbations, (2) classifies unrecognizable images with high certainty, (3) performs not much better than simple input binarization and (4) features adversarial perturbations that make little sense to humans. These results suggest that MNIST is far from being solved in terms of adversarial robustness. We present a novel robust classification model that performs *analysis by synthesis* using learned class-conditional data distributions. We derive bounds on the robustness and go to great length to empirically evaluate our model using maximally effective adversarial attacks by (a) applying decision-based, score-based, gradient-based and transfer-based attacks for several different L_p norms, (b) by designing a new attack that exploits the structure of our defended model and (c) by devising a novel decision-based attack that seeks to minimize the number of perturbed pixels (L_0). The results suggest that our approach yields state-of-the-art robustness on MNIST against L_0, L_2 and L_∞ perturbations and we demonstrate that most adversarial examples are strongly perturbed towards the perceptual boundary between the original and the adversarial class.

1 INTRODUCTION

Deep neural networks (DNNs) are strikingly susceptible to *minimal adversarial perturbations* (Szegedy et al., 2013), perturbations that are (almost) imperceptible to humans but which can switch the class prediction of DNNs to basically any desired target class.

One key problem in finding successful defenses is the difficulty of reliably evaluating model robustness. It has been shown time and again (Athalye et al., 2018; Athalye & Carlini, 2018; Brendel & Bethge, 2017) that basically all defenses previously proposed did not increase model robustness but prevented existing attacks from finding minimal adversarial examples, the most common reason being masking of the gradients on which most attacks rely. The few verifiable defenses can only guarantee robustness within a small linear regime around the data points (Hein & Andriushchenko, 2017; Raghunathan et al., 2018).

The only defense currently considered effective (Athalye et al., 2018) is a particular type of adversarial training (Madry et al., 2018). On MNIST, as of today this method is able to reach an accuracy of 88.79% for adversarial perturbations with an L_∞ norm bounded by $\epsilon = 0.3$ (Zheng et al., 2018). In other words, if we allow an attacker to perturb the brightness of each pixel by up to 0.3 (range $[0, 1]$),

then he can only trick the model on $\approx 10\%$ of the samples. This is a great success, but does the model really learn more causal features to classify MNIST? We here demonstrate that this is not the case: For one, the defense by Madry et al. (SOTA on L_∞) has lower L_0 robustness than undefended networks and is still highly susceptible in the L_2 metric. Second, the robustness results by Madry et al. can also be achieved with a simple input quantization because of the binary nature of single pixels in MNIST (which are typically either completely black or white) (Schmidt et al., 2018). Third, it is straight-forward to find unrecognizable images that are classified as a digit with high certainty. Finally, the minimum adversarial examples we find for the defense by Madry et al. make little to no sense to humans.

Taken together, even MNIST cannot be considered solved with respect to adversarial robustness. By "solved" we mean a model that reaches at least 99% accuracy (see accuracy-vs-robustness trade-off (Tsipras et al., 2018; Bubeck et al., 2018)) and whose adversarial examples carry semantic meaning to humans (by which we mean that they start looking like samples that could belong to either class). Hence, despite the fact that MNIST is considered "too easy" by many and a mere toy example, finding adversarially robust models on MNIST is still an open problem.

A potential solution we explore in this paper is inspired by unrecognizable images (Nguyen et al., 2015) or *distal adversarials*. Distal adversarials are images that do not resemble images from the training set but which typically look like noise while still being classified by the model with high confidence. It seems difficult to prevent such images in feedforward networks as we have little control over how inputs are classified that are far outside of the training domain. In contrast, generative models can learn the distribution of their inputs and are thus able to gauge their confidence accordingly. By additionally learning the image distribution within each class we can check that the classification makes sense in terms of the image features being present in the input (e.g. an image of a bus should contain actual bus features). Following this line of thought from an information-theoretic perspective, one arrives at the well-known concept of Bayesian classifiers. We here introduce a fine-tuned variant based on variational autoencoders (Kingma & Welling, 2013) that combines robustness with high accuracy.

In summary, the contributions of this paper are as follows:

- We show that MNIST is unsolved from the point of adversarial robustness: the SOTA defense of Madry et al. (2018) is still highly vulnerable to tiny perturbations that are meaningless to humans.
- We introduce a new robust classification model and derive instance-specific robustness guarantees.
- We develop a strong attack that leverages the generative structure of our classification model.
- We introduce a novel decision-based attack that minimizes L_0.
- We perform an extensive evaluation of our defense across many attacks to show that it surpasses SOTA on L_0, L_2 and L_∞ and features many adversarials that carry semantic meaning to humans.

We have evaluated the proposed defense to the best of our knowledge, but we are aware of the (currently unavoidable) limitations of evaluating robustness. We will release the model architecture and trained weights as a friendly invitation to fellow researchers to evaluate our model independently.

2 Related Work

The many defenses against adversarial attacks can roughly be subdivided into four categories:

- **Adversarial training:** The training data is augmented with adversarial examples to make models more robust (Madry et al., 2018; Szegedy et al., 2013; Tramèr et al., 2017; Ilyas et al., 2017).
- **Manifold projections:** An input sample is projected onto a learned data manifold (Samangouei et al., 2018; Ilyas et al., 2017; Shen et al., 2017; Song et al., 2018).
- **Stochasticity:** Certain inputs or hidden activations are shuffled or randomized (Prakash et al., 2018; Dhillon et al., 2018; Xie et al., 2018).
- **Preprocessing:** Inputs or hidden activations are quantized, projected into a different representation or are otherwise preprocessed (Buckman et al., 2018; Guo et al., 2018; Kabilan et al., 2018).

There has been much work showing that basically all defenses suggested so far in the literature do not substantially increase robustness over undefended neural networks (Athalye et al., 2018; Brendel &

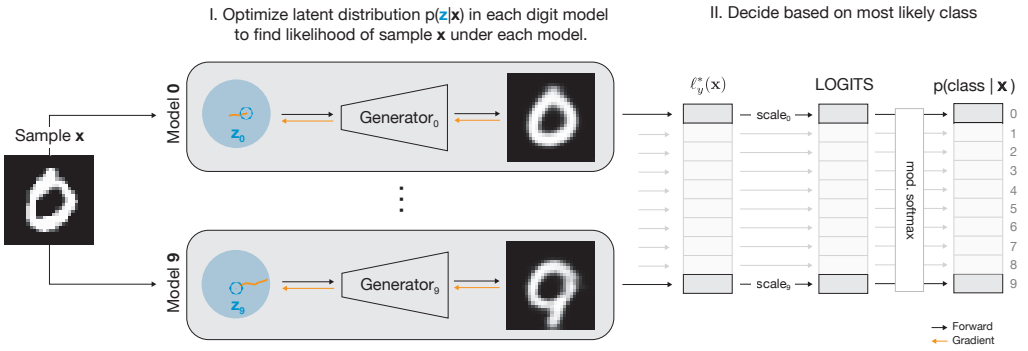


Figure 1: Overview over model architecture. In a nutshell: I) for each sample $\mathbf{x}$ we compute a lower bound on the log-likelihood (ELBO) under each class using gradient descent in the latent space. II) A class-dependent scalar weighting of the class-conditional ELBOs forms the final class prediction.

Bethge, 2017). The only widely accepted exception according to Athalye et al. (2018) is the defense by Madry et al. (2018) which is based on data augmentation with adversarials found by iterative projected gradient descent with random starting points. However, as we see in the results section, this defense is limited to the metric it is trained on (L_∞) and it is straight-forward to generate small adversarial perturbations that carry little semantic meaning for humans.

Some other defenses have been based on generative models. Typically these defenses use the generative model to project onto the (learned) manifold of "natural" inputs. This includes in particular DefenseGAN (Samangouei et al., 2018), Adversarial Perturbation Elimination GAN (Shen et al., 2017) and Robust Manifold Defense (Ilyas et al., 2017), all of which project an image onto the manifold defined by a generator network G. The generated image is then classified by a discriminator in the usual way. A similar idea is used by PixelDefend (Song et al., 2018) which uses an autoregressive probabilistic method to learn the data manifold. Other ideas in similar directions include the use of denoising autoencoders (Liao et al., 2017) as well as MagNets (Meng & Chen, 2017), which projects or rejects inputs depending on their distance to the data manifold. All of these proposed defenses except for the defense by Ilyas et al. (2017) have been tested by Athalye et al. (2018); Athalye & Carlini (2018); Carlini & Wagner (2017) and others, and shown to be ineffective. It is straight-forward to understand why: For one, many adversarials still look like normal data points to humans. Second, the classifier on top of the projected image is as vulnerable to adversarial examples as before. Hence, for any data set with a natural amount of variation there will almost always be a certain perturbation against which the classifier is vulnerable and which can be induced by the right inputs.

We here follow a different approach by modeling the input distribution within each class (instead of modeling a single distribution for the complete data), and by classifying a new sample according to the class under which it has the highest likelihood. This approach, commonly referred to as a Bayesian classifier, gets away without any additional and vulnerable classifier. A very different but related approach is the work by George et al. (2017) which suggested a generative compositional model of digits to solve cluttered digit scenes like Captchas (adversarial robustness was not evaluated).

3 Model Description

Intuitively, we want to learn a causal model of the inputs (Schölkopf, 2017). Consider a cat: we want a model to learn that cats have four legs and two pointed ears, and then use this model to check whether a given input can be generated with these features. This intuition can be formalized as follows. Let $(\mathbf{x}, y)$ with $\mathbf{x} \in \mathbb{R}^N$ be an input-label datum. Instead of directly learning a posterior $p(y|\mathbf{x})$ from inputs to labels we now learn generative distributions $p(\mathbf{x}|y)$ and classify new inputs using Bayes formula,

$$p(y|\mathbf{x}) = \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})} \propto p(\mathbf{x}|y)p(y). \tag{1}$$

The label distribution $p(y)$ can be estimated from the training data. To learn the class-conditional sample distributions $p(\mathbf{x}|y)$ we use variational autoencoders (VAEs) (Kingma & Welling, 2013). VAEs estimate the log-likelihood $\log p(\mathbf{x})$ by learning a probabilistic generative model $p_\theta(\mathbf{x}|\mathbf{z})$

with latent variables $\mathbf{z} \sim p(\mathbf{z})$ and parameters θ (see Appendix A.3 for the full derivation). For class-conditional VAEs we can derive a lower bound on the log-likelihood $\log p(\mathbf{x}|y)$ as

$$\log p(\mathbf{x}|y) \geq \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x},y)} \left[\log p_\theta(\mathbf{x}|\mathbf{z},y)\right] - \mathcal{D}_{KL}\left[q_\phi(\mathbf{z}|\mathbf{x},y)||p(\mathbf{z})\right] =: \ell_y(\mathbf{x}), \tag{2}$$

where $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbb{1})$ is a simple normal prior and $q_\phi(\mathbf{z}|\mathbf{x}, y)$ is the variational posterior with parameters ϕ. The first term on the RHS is basically a reconstruction error while the second term on the RHS is the mismatch between the variational and the true posterior. The term on the RHS is the so-called evidence lower bound (ELBO) on the log-likelihood (Kingma & Welling, 2013). We implement the conditional distributions $p_\theta(\mathbf{x}|\mathbf{z}, y)$ and $q_\phi(\mathbf{z}|\mathbf{x}, y)$ as normal distributions for which the means are parametrized as DNNs (all details and hyperparameters are reported in Appendix A.7).

Our *Analysis by Synthesis* model (ABS) is illustrated in Figure 1. It combines several elements to simultaneously achieve high accuracy and robustness against adversarial perturbations:

- **Class-conditional distributions:** For each class y we train a variational autoencoder VAE_y on the samples of class y to learn the class-conditional distribution $p(\mathbf{x}|y)$. This allows us to estimate a lower bound $\ell_y(\mathbf{x})$ on the log-likelihood of sample $\mathbf{x}$ under each class y.
- **Optimization-based inference:** The variational inference $q_\phi(\mathbf{z}|\mathbf{x}, y)$ is itself a neural network susceptible to adversarial perturbations. We therefore only use variational inference during training and perform "exact" inference over $p_\theta(\mathbf{x}|\mathbf{z}, y)$ during evaluation. This "exact" inference is implemented using gradient descent in the latent space (with fixed posterior width) to find the optimal $\mathbf{z}_y$ which maximizes the lower bound on the log-likelihood for each class:

  $$\ell_y^*(\mathbf{x}) = \max_{\mathbf{z}} \ \log p_\theta(\mathbf{x}|\mathbf{z}, y) - \mathcal{D}_{KL}\left[\mathcal{N}(\mathbf{z}, \sigma_q \mathbb{1})||\mathcal{N}(\mathbf{0}, \mathbb{1})\right]. \tag{3}$$

  Note that we replaced the expectation in equation 2 with a maximum likelihood sample to avoid stochastic sampling and to simplify optimization. To avoid local minima we evaluate 8000 random points in the latent space of each VAE, from which we pick the best as a starting point for a gradient descent with 50 iterations using the Adam optimizer (Kingma & Ba, 2014).
- **Classification and confidence:** Finally, to perform the actual classification, we scale all $\ell_y^*(\mathbf{x})$ with a factor α, exponentiate, add an offset η and divide by the total evidence (like in a softmax),

  $$p(y|\mathbf{x}) = \left(e^{\alpha \ell_y^*(\mathbf{x})} + \eta\right) / \sum\nolimits_c \left(e^{\alpha \ell_c^*(\mathbf{x})} + \eta\right). \tag{4}$$

  We introduced η for the following reason: even on points far outside the data domain, where all likelihoods $q(\mathbf{x}, y) = e^{\alpha \ell_y^*(\mathbf{x})} + \eta$ are small, the standard softmax ($\eta = 0$) can lead to sharp posteriors $p(y|\mathbf{x})$ with high confidence scores for one class. This behavior is in stark contrast to humans, who would report a uniform distribution over classes for unrecognizable images. To model this behavior we set $\eta > 0$: in this case the posterior $p(y|\mathbf{x})$ converges to a uniform distribution whenever the maximum $q(\mathbf{x}, y)$ gets small relative to η . We chose η such that the median confidence $p(y|\mathbf{x})$ is 0.9 for the predicted class on clean test samples. Furthermore, for a better comparison with cross-entropy trained networks, the scale α is trained to minimize the cross-entropy loss. We also tested this graded softmax in standard feedforward CNNs but did not find any improvement with respect to unrecognizable images.
- **Binarization (*Binary ABS* only):** The pixel intensities of MNIST images are almost binary. We exploit this by projecting the intensity b of each pixel to 0 if $b < 0.5$ or 1 if $b \geq 0.5$ during testing.
- **Discriminative finetuning (*Binary ABS* only):** To improve the accuracy of the *Binary ABS* model we multiply $\ell_y^*(\mathbf{x})$ with an additional class-dependent scalar γ_y. The scalars are learned discriminatively (see A.7) and reach values in the range $\gamma_y \in [0.96, 1.06]$ for all classes y.

On important ingredient for the robustness of the ABS model is the Gaussian posterior in the reconstruction term which ensures that small changes in the input (in terms of L2) can only entail small changes to the posterior likelihood and thus to the model decision.

4 Tight Estimates of the Lower Bound for Adversarial Examples

The decision of the model depends on the likelihood in each class, which for clean samples is mostly dominated by the posterior likelihood $p(\mathbf{x}|\mathbf{z})$. Because we chose this posterior to be Gaussian, the

class-conditional likelihoods can only change gracefully with changes in $\mathbf{x}$, a property which allows us to derive lower bounds on the model robustness. To see this, note that equation 3 can be written as,

$$\ell_c^*(\mathbf{x}) = \max_{\mathbf{z}} \; -\mathcal{D}_{KL}\left[\mathcal{N}(\mathbf{z}, \sigma_q \mathbb{1}) || \mathcal{N}(\mathbf{0}, \mathbb{1})\right] - \frac{1}{2\sigma^2} \left\|\mathbf{G}_c(\mathbf{z}) - \mathbf{x}\right\|_2^2 + C, \tag{5}$$

where we absorbed the normalization constants of $p(\mathbf{x}|\mathbf{z})$ into C and $\mathbf{G}_c(\mathbf{z})$ is the mean of $p(\mathbf{x}|\mathbf{z}, c)$. Let y be the ground-truth class and let $\mathbf{z}_{\mathbf{x}}^*$ be the optimal latent for the clean sample $\mathbf{x}$ for class y. We can then estimate a lower bound on $\ell_y^*(\mathbf{x} + \boldsymbol{\delta})$ for a perturbation $\boldsymbol{\delta}$ with size $\epsilon = \|\boldsymbol{\delta}\|_2$ (see derivation in Appendix A.4),

$$\ell_y^*(\mathbf{x} + \boldsymbol{\delta}) \geq \ell_y^*(\mathbf{x}) - \frac{1}{\sigma^2}\epsilon \left\|\mathbf{G}_y(\mathbf{z}_{\mathbf{x}}^*) - \mathbf{x}\right\|_2 - \frac{1}{2\sigma^2}\epsilon^2 + C. \tag{6}$$

Likewise, we can derive an upper bound of $\ell_y^*(\mathbf{x} + \boldsymbol{\delta})$ for all other classes $c \neq y$ (see Appendix A.5),

$$\ell_c^*(\mathbf{x} + \boldsymbol{\delta}) \leq -\mathcal{D}_{KL}\left[\mathcal{N}(\mathbf{0}, \sigma_q \mathbb{1}) || \mathcal{N}(\mathbf{0}, \mathbb{1})\right] + C - \begin{cases} \frac{1}{2\sigma^2}(d_c - \epsilon)^2 & \text{if } d_c \geq \epsilon \\ 0 & \text{else} \end{cases}. \tag{7}$$

for $d_c = \min_z \left\|\mathbf{G}_c(\mathbf{z}) - \mathbf{x}\right\|_2$. Now we can find ϵ for a given image $\mathbf{x}$ by equating (7) = (6),

$$\epsilon_x = \min_{c \neq y} \max \left\{0, \frac{d_c + \ell_y^*(\mathbf{x}) - \mathcal{D}_{KL}\left[\mathcal{N}(\mathbf{0}, \sigma_q \mathbb{1}) || \mathcal{N}(\mathbf{0}, \mathbb{1})\right]}{2(d_c + \left\|\mathbf{G}_y(\mathbf{z}_{\mathbf{x}}^*) - \mathbf{x}\right\|_2)}\right\}. \tag{8}$$

Note that one assumption we make is that we can find the global minimum of $\|\mathbf{G}_c(\mathbf{z}) - \mathbf{x}\|_2^2$. In practice we generally find a very tight estimate of the global minimum (and thus the lower bound) because we optimize in a smooth and low-dimensional space and because we perform an additional brute-force sampling step. We provide quantitative values for ϵ in section 7.

5 Adversarial Attacks

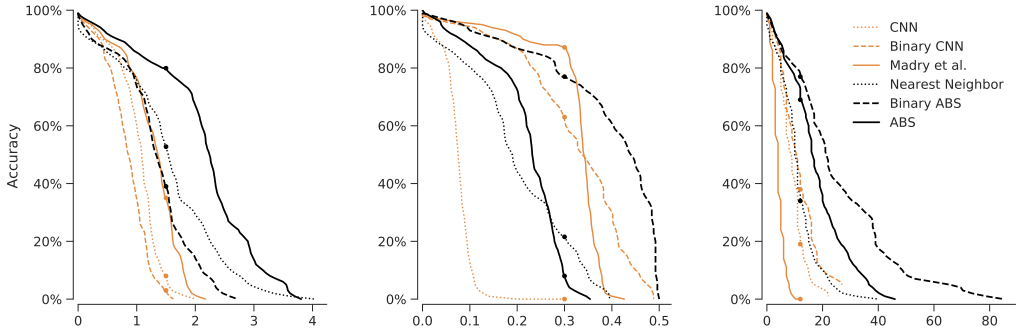
Reliably evaluating model robustness is difficult because each attack only provides an upper bound on the size of the adversarial perturbations (Uesato et al., 2018). To make this bound as tight as possible we apply many different attacks and choose the best one for each sample and model combination (using the implementations in Foolbox v1.3 (Rauber et al., 2017) which often perform internal hyperparameter optimization). We also created a novel decision-based L_0 attack as well as a customized attack that specifically exploits the structure of our model. Nevertheless, we cannot rule out that more effective attacks exist and we will release the trained model for future testing.

Latent Descent attack This novel attack exploits the structure of the ABS model. Let $\mathbf{x}_t$ be the perturbed sample $\mathbf{x}$ in iteration t. We perform variational inference $p(\mathbf{z}|\mathbf{x}_t, y) = \mathcal{N}(\boldsymbol{\mu}_y(\mathbf{x}_t), \sigma_q \boldsymbol{I})$ to find the most likely class $\tilde{y}$ that is different from the ground-truth class. We then make a step towards the maximum likelihood posterior $p(\mathbf{x}|\mathbf{z}, \tilde{y})$ of that class which we denote as $\tilde{\mathbf{x}}_{\tilde{y}}$,

$$\mathbf{x}_t \mapsto (1 - \epsilon)\mathbf{x}_t + \epsilon \tilde{\mathbf{x}}_{\tilde{y}}. \tag{9}$$

We choose $\epsilon = 10^{-2}$ and iterate until we find an adversarial. For a more precise estimate we perform a subsequent binary search of 10 steps within the last ϵ interval. Finally, we perform another binary search between the adversarial and the original image to reduce the perturbation as much as possible.

Decision-based attacks We use several decision-based attacks because they do not rely on gradient information and are thus insensitive to gradient masking or missing gradients. In particular, we apply the *Boundary Attack* (Brendel et al., 2018), which is competitive with gradient-based attacks in minimizing the L_2 norm, and introduce the *Pointwise Attack*, a novel decision-based attack that greedily minimizes the L_0 norm. It first adds salt-and-pepper noise until the image is misclassified and then repeatedly iterates over all perturbed pixels, resetting them to the clean image if the perturbed image stays adversarial. The attack ends when no pixel can be reset anymore. We provide an implementation of the attack in Foolbox (Rauber et al., 2017). Finally, we apply two simple noise attacks, the *Gaussian Noise* attack and the *Salt&Pepper Noise* attack as baselines.



(a) L_2 distance (b) L_∞ distance (c) L_0 distance

Figure 2: Accuracy-distortion plots for each distance metric and all models. In (b) we see that a threshold at 0.3 favors Madry et al. while a threshold of 0.35 would have favored the Binary ABS.

Transfer-based attacks Transfer attacks also don't rely on gradients of the target model but instead compute them on a substitute: given an input $\mathbf{x}$ we first compute adversarial perturbations $\boldsymbol{\delta}$ on the substitute using different gradient-based attacks (L_2 and L_∞ Basic Iterative Method (BIM), Fast Gradient Sign Method (FGSM) and L_2 Fast Gradient Method) and then perform a line search to find the smallest ϵ for which $\mathbf{x} + \epsilon\boldsymbol{\delta}$ (clipped to the range [0, 1]) is still an adversarial for the target model.

Gradient-based attacks We apply the Momentum Iterative Method (MIM) (Dong et al., 2017) that won the NIPS 2017 adversarial attack challenge, the Basic Iterative Method (BIM) (Kurakin et al., 2016) (also known as Projected Gradient Descent (PGD))—for both the L_2 and the L_∞ norm—as well as the Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2014) and its L_2 variant, the Fast Gradient Method (FGM). For models with input binarization (Binary CNN, Binary ABS), we obtain gradients using the straight-through estimator (Bengio et al., 2013).

Score-based attacks We additionally run all attacks listed under *Gradient-based attacks* using numerically estimated gradients (possible for all models). We use a simple coordinate-wise finite difference method (NES estimates (Ilyas et al., 2018) performed comparable or worse) and repeat the attacks with different values for the step size of the gradient estimator.

Postprocessing (binary models only) For models with input binarization (sec. 6) we postprocess all adversarials by setting pixel intensities either to the corresponding value of the clean image or the binarization threshold (0.5). This reduces the perturbation size without changing model decisions.

6 Experiments

We compare our ABS model as well as two ablations—ABS with input binarization during test time (Binary ABS) and a CNN with input binarization during train and test time (Binary CNN)—against three other models: the SOTA L_∞ defense (Madry et al., 2018)[1], a Nearest Neighbour (NN) model (as a somewhat robust but not accurate baseline) and a vanilla CNN (as an accurate but not robust baseline), see Appendix A.7. We run all attacks (see sec. 5) against all applicable models.

For each model and L_p norm, we show how the accuracy of the models decreases with increasing adversarial perturbation size (Figure 2) and report two metrics: the median adversarial distance (Table 1, left values) and the model's accuracy against bounded adversarial perturbations (Table 1, right values). The median of the perturbation sizes (Table 1, left values) is robust to outliers and summarizes most of the distributions quite well. It represents the perturbation size for which the particular model achieves 50% accuracy and does not require the choice of a threshold. Clean samples that are already misclassified are counted as adversarials with a perturbation size equal to 0, failed attacks as ∞. The commonly reported model accuracy on bounded adversarial perturbations, on the other hand, requires a metric-specific threshold that can bias the results. We still report it (Table 1, right values) for completeness and set $\epsilon_{L_2} = 1.5$, $\epsilon_{L_\infty} = 0.3$ and $\epsilon_{L_0} = 12$ as thresholds.

[1]We used the trained model provided by the authors: https://github.com/MadryLab/mnist_challenge

	CNN	Binary CNN	Nearest Neighbor	Madry et al.	Binary ABS	ABS
Clean	99.1%	98.5%	96.9%	98.8%	99.0%	99.0%
L_2-metric ($\epsilon = 1.5$)						
Transfer Attacks	1.1 / 14%	1.4 / 38%	5.4 / 90%	3.7 / 94%	2.5 / 86%	4.6 / 94%
Gaussian Noise	5.2 / 96%	3.4 / 92%	∞ / 91%	5.4 / 96%	5.6 / 89%	10.9 / 98%
Boundary Attack	1.2 / 21%	3.3 / 84%	2.9 / 73%	1.4 / 37%	6.0 / 91%	2.6 / 83%
Pointwise Attack	3.4 / 91%	1.9 / 71%	3.5 / 89%	1.9 / 71%	3.1 / 86%	4.6 / 94%
FGM	1.4 / 48%	1.4 / 50%		∞ / 96%		
FGM w/ GE	1.4 / 42%	2.8 / 51%	3.7 / 79%	∞ / 88%	1.9 / 68%	3.5 / 89%
DeepFool	1.2 / 18%	1.0 / 11%		9.0 / 91%		
DeepFool w/ GE	1.3 / 30%	0.9 / 5%	1.6 / 55%	5.1 / 90%	1.4 / 41%	2.4 / 83%
L2 BIM	1.1 / 13%	1.0 / 11%		4.8 / 88%		
L2 BIM w/ GE	1.1 / 37%	∞ / 50%	1.7 / 62%	3.4 / 88%	1.6 / 63%	3.1 / 87%
Latent Descent Attack					2.6 / 97%	2.7 / 85%
All L_2 Attacks	1.1 / 8%	0.9 / 3%	1.5 / 53%	1.4 / 35%	1.3 / 39%	**2.3** / 80%
L_∞-metric ($\epsilon = 0.3$)						
Transfer Attacks	0.08 / 0%	0.44 / 85%	0.42 / 78%	0.39 / 92%	0.49 / 88%	0.34 / 73%
FGSM	0.10 / 4%	0.43 / 77%		0.45 / 93%		
FGSM w/ GE	0.10 / 21%	0.42 / 71%	0.38 / 68%	0.47 / 89%	0.49 / 85%	0.27 / 34%
L_∞ DeepFool	0.08 / 0%	0.38 / 74%		0.42 / 90%		
L_∞ DeepFool w/ GE	0.09 / 0%	0.37 / 67%	0.21 / 26%	0.53 / 90%	0.46 / 78%	0.27 / 39%
BIM	0.08 / 0%	0.36 / 70%		0.36 / 90%		
BIM w/ GE	0.08 / 37%	∞ / 70%	0.25 / 43%	0.46 / 89%	0.49 / 86%	0.25 / 13%
MIM	0.08 / 0%	0.37 / 71%		0.34 / 90%		
MIM w/ GE	0.09 / 36%	∞ / 69%	0.19 / 26%	0.36 / 89%	0.46 / 85%	0.26 / 17%
All L_∞ Attacks	0.08 / 0%	0.34 / 64%	0.19 / 22%	0.34 / 88%	**0.44** / 77%	0.23 / 8%
L_0-metric ($\epsilon = 12$)						
Salt&Pepper Noise	44.0 / 91%	44.0 / 88%	161.0 / 88%	13.5 / 56%	146.0 / 94%	165.0 / 94%
Pointwise Attack 10x	9.0 / 19%	11.0 / 39%	10.0 / 34%	4.0 / 0%	22.0 / 77%	16.5 / 69%
All L_0 Attacks	9.0 / 19%	11.0 / 38%	10.0 / 34%	4.0 / 0%	**21.5** / 77%	16.5 / 69%

Table 1: Results for different models, adversarial attacks and distance metrics. Each entry shows the median adversarial distance across all samples (left value, black) as well as the model's accuracy against adversarial perturbations bounded by the thresholds $\epsilon_{L_2} = 1.5$, $\epsilon_{L_\infty} = 0.3$ and $\epsilon_{L_0} = 12$ (right value, gray). *"w/ GE"* indicates attacks that use numerical gradient estimation.

7 RESULTS

Minimal Adversarials Our robustness evaluation results of all models are reported in Table 1 and Figure 2. All models except the Nearest Neighbour classifier perform close to 99% accuracy on clean test samples. We report results for three different norms: L_2, L_∞ and L_0.

- For L_2 our ABS model outperforms all other models by a large margin.
- For L_∞, our Binary ABS model is state-of-the-art in terms of median perturbation size. In terms of accuracy (perturbations < 0.3), Madry et al. seems more robust. However, as revealed by the accuracy-distortion curves in Figure 2, this is an artifact of the specific threshold (Madry et al. is optimized for 0.3). A slightly larger one (e.g. 0.35) would strongly favor the Binary ABS model.
- For L_0, both ABS and Binary ABS are much more robust than all other models. Interestingly, the model by Madry et al. is the least robust, even less than the baseline CNN.

In Figure 3 we show adversarial examples. For each sample we show the minimally perturbed L_2 adversarial found by any attack. Adversarials for the baseline CNN and the Binary CNN are almost

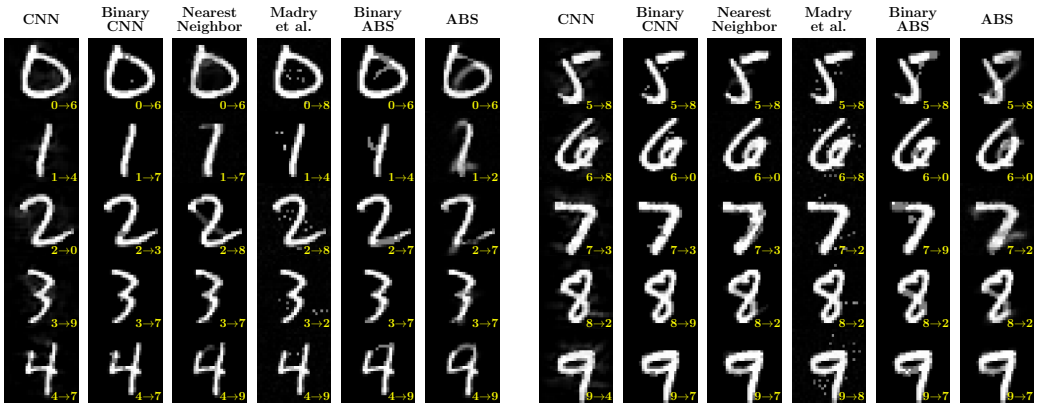


Figure 3: Adversarial examples for the ABS models are perceptually meaningful: For each sample (randomly chosen from each class) we show the minimally perturbed L_2 adversarial found by any attack. Our ABS models have clearly visible and often semantically meaningful adversarials. Madry et al. requires perturbations that are clearly visible, but their semantics are less clear.

imperceptible. The Nearest Neighbour model, almost by design, exposes (some) adversarials that interpolate between two numbers. The model by Madry et al. requires perturbations that are clearly visible but make little semantic sense to humans. Finally, adversarials generated for the ABS models are semantically meaningful for humans and are sitting close to the perceptual boundary between the original and the adversarial class. For a more thorough comparison see appendix Figures 5, 6 and 7.

Lower bounds on Robustness For the ABS models and the L_2 metric we estimate a lower bound of the robustness. The lower bound for the mean perturbation[2] for the MNIST test set is $\epsilon = 0.690 \pm 0.005$ for the ABS and $\epsilon = 0.601 \pm 0.005$ for the binary ABS. We estimated the error by using different random seeds for our optimization procedure and standard error propagation over 10 runs. With adversarial training Hein & Andriushchenko (2017) achieve a mean L_2 robustness guarantee of $\epsilon = 0.48$ while reaching 99% accuracy. In the L_{inf} metric we find a median robustness of 0.06.

Distal Adversarials We probe the behavior of CNN, Madry et al. and our ABS model outside the data distribution. We start from random noise images and perform gradient ascent to maximize the output probability of a fixed label until $p(y|\mathbf{x}) \geq 0.9$ (as computed by the modified softmax from equation (8)). The results are visualized in Figure 4. Standard CNNs and Madry et al. provide high confidence class probabilities for unrecognizable images. Our ABS model does not provide high confidence predictions in out-of-distribution regions.

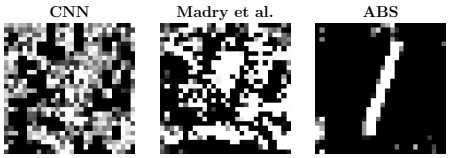


Figure 4: Images of ones classified with a probability above 90%.

8 Discussion & Conclusion

In this paper we demonstrated that, despite years of work, we as a community failed to create neural networks that can be considered robust on MNIST from the point of human perception. In particular, we showed that even today's best defense is susceptible to small adversarial perturbations that make little to no semantic sense to humans. We presented a new approach based on *analysis by synthesis* that seeks to explain its inference by means of the actual image features. We performed an extensive analysis to show that minimal adversarial perturbations in this model are large across all tested L_p norms and semantically meaningful to humans. Note that our architecture derives its robustness from its design and does not require any additionally training with adversarial examples.

We acknowledge that it is not easy to reliably evaluate a model's adversarial robustness and most defenses proposed in the literature have later been shown to be ineffective. In particular, the structure

[2]The mean instead of the median is reported to allow for a comparison with (Hein & Andriushchenko, 2017).

of the ABS model prevents the computation of gradients which might give the model an unfair advantage. We put a lot of effort into an extensive evaluation of adversarial robustness using a large collection of powerful attacks, including one specifically designed to be particularly effective against the ABS model (the *Latent Descent* attack), and we will release the model architecture and trained weights as a friendly invitation to fellow researchers to evaluate our model.

Looking at the results of individual attacks (Table 1) we find that there is no single attack that works best on all models, thus highlighting the importance for a broad range of attacks. Without the Boundary Attack, for example, Madry et al. would have looked more robust to L_2 adversarials than it is. For similar reasons Figure 6b of Madry et al. (2018) reports a median L_2 perturbation size larger than 5, compared to the 1.4 achieved by the Boundary Attack. Moreover,the combination of all attacks of one metric (*All L_2 / L_∞ / L_0 Attacks*) is often better than any individual attack, indicating that different attacks are optimal on different samples.

Our conceptual implementation of the ABS model with one VAE per class neither scales efficiently to more classes nor to more complex datasets (a preliminary experiment on CIFAR10 provided only 54% test accuracy). However, first experiments on two class CIFAR indicate that the proposed model is also robust on CIFAR (we reach a median L2 robustness of 2.6 compared to 0.8 for a vanilla CNN, see Appendix A.1) for details). To increase the accuracy, there are many ways in which the ABS model can be improved, ranging from better and faster generative models (e.g. flow-based) to better training procedures.

In a nutshell, we demonstrated that MNIST is still not solved from the point of adversarial robustness and showed that our novel approach based on analysis by synthesis has great potential to reduce the vulnerability against adversarial attacks and to align machine perception with human perception.

ACKNOWLEDGMENTS

This work has been funded, in part, by the German Federal Ministry of Education and Research (BMBF) through the Bernstein Computational Neuroscience Program Tübingen (FKZ: 01GQ1002) as well as the German Research Foundation (DFG CRC 1233 on "Robust Vision"). The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting L.S. and J.R.; J.R. acknowledges support by the Bosch Forschungsstiftung (Stifterverband, T113/30057/17); W.B. was supported by the Carl Zeiss Foundation (0563-2.8/558/3); M.B. acknowledges support by the Centre for Integrative Neuroscience Tübingen (EXC 307); W.B. and M.B. were supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior / Interior Business Center (DoI/IBC) contract number D16PC00003.

REFERENCES

Anish Athalye and Nicholas Carlini. On the robustness of the cvpr 2018 white-box adversarial example defenses. *arXiv preprint arXiv:1804.03286*, 2018.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

W. Brendel and M. Bethge. Comment on "biologically inspired protection of deep networks from adversarial attacks". *arXiv preprint arXiv:1704.01547*, 2017.

Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=SyZI0GWCZ`.

Sébastien Bubeck, Eric Price, and Ilya Razenshteyn. Adversarial examples from computational constraints. *arXiv preprint arXiv:1805.10204*, 2018.

Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=S18Su--CW`.

Nicholas Carlini and David Wagner. Magnet and" efficient defenses against adversarial attacks" are not robust to adversarial examples. *arXiv preprint arXiv:1711.08478*, 2017.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

Guneet S. Dhillon, Kamyar Azizzadenesheli, Jeremy D. Bernstein, Jean Kossaifi, Aran Khanna, Zachary C. Lipton, and Animashree Anandkumar. Stochastic activation pruning for robust adversarial defense. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1uR4GZRZ.

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Xiaolin Hu, Jianguo Li, and Jun Zhu. Boosting adversarial attacks with momentum. arxiv preprint. *arXiv preprint arXiv:1710.06081*, 2017.

Dileep George, Wolfgang Lehrach, Ken Kansky, Miguel Lázaro-Gredilla, Christopher Laan, Bhaskara Marthi, Xinghua Lou, Zhaoshi Meng, Yi Liu, Huayan Wang, Alex Lavin, and D. Scott Phoenix. A generative vision model that trains with high data efficiency and breaks text-based captchas. *Science*, 358(6368), 2017. ISSN 0036-8075. doi: 10.1126/science.aag2612. URL http://science.sciencemag.org/content/358/6368/eaag2612.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SyJ7ClWCb.

Matthias Hein and Maksym Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. In *Advances in Neural Information Processing Systems 30*, pp. 2266–2276. Curran Associates, Inc., 2017.

Andrew Ilyas, Ajil Jalal, Eirini Asteri, Constantinos Daskalakis, and Alexandros G Dimakis. The robust manifold defense: Adversarial training using generative models. *arXiv preprint arXiv:1712.09196*, 2017.

Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. *arXiv preprint arXiv:1804.08598*, 2018.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Vishaal Munusamy Kabilan, Brandon Morris, and Anh Nguyen. Vectordefense: Vectorization as a defense to adversarial examples. *arXiv preprint arXiv:1804.08529*, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Jun Zhu, and Xiaolin Hu. Defense against adversarial attacks using high-level representation guided denoiser. *arXiv preprint arXiv:1712.02976*, 2017.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 135–147. ACM, 2017.

Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

Aaditya Prakash, Nick Moran, Solomon Garber, Antonella DiLillo, and James Storer. Deflecting adversarial attacks with pixel deflection. *arXiv preprint arXiv:1801.08926*, 2018.

Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Bys4ob-Rb.

Jonas Rauber, Wieland Brendel, and Matthias Bethge. Foolbox: A python toolbox to benchmark the robustness of machine learning models. *arXiv preprint arXiv:1707.04131*, 2017. URL http://arxiv.org/abs/1707.04131.

Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BkJ3ibb0-.

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. *CoRR*, abs/1804.11285, 2018. URL http://arxiv.org/abs/1804.11285.

Bernhard Schölkopf. Causal learning, 2017. URL https://icml.cc/Conferences/2017/Schedule?showEvent=931. Thirty-fourth International Conference on Machine Learning.

Shiwei Shen, Guoqing Jin, Ke Gao, and Yongdong Zhang. Ape-gan: Adversarial perturbation elimination with gan. *arXiv preprint arXiv:1707.05474*, 2017.

Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJUYGxbCW.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. There is no free lunch in adversarial robustness (but there are unexpected benefits). *arXiv preprint arXiv:1805.12152*, 2018.

Jonathan Uesato, Brendan O'Donoghue, Pushmeet Kohli, and Aaron van den Oord. Adversarial risk and the dangers of evaluating against weak attacks. In *Proceedings of the 35th International Conference on Machine Learning*, 2018. URL http://proceedings.mlr.press/v80/uesato18a.html.

Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Sk9yuql0Z.

Tianhang Zheng, Changyou Chen, and Kui Ren. Distributionally adversarial attack. *arXiv preprint arXiv:1808.05537*, 2018.

A APPENDIX

A.1 TWO CLASS CIFAR

We estimate the robustness of our ABS model on two class CIFAR (airplane vs. automobile). Preliminary results suggest that our robustness is not limited to MNIST.

In order to adapt to CIFAR, we modified the ABS slightly by modifying encoder and decoder to fit (32x32x3) CIFAR images. We also increased the number of dimensions in the latent space form 8 to 20.

Model	CNN	ABS
Accuracy	97.1%	89.7%
Median L_2 distance	0.8 (with BIM)	2.5 (with Latent Descent attack)

Table 2: Accuracy and estimated robustness on two class CIFAR.

A.2 FIGURES

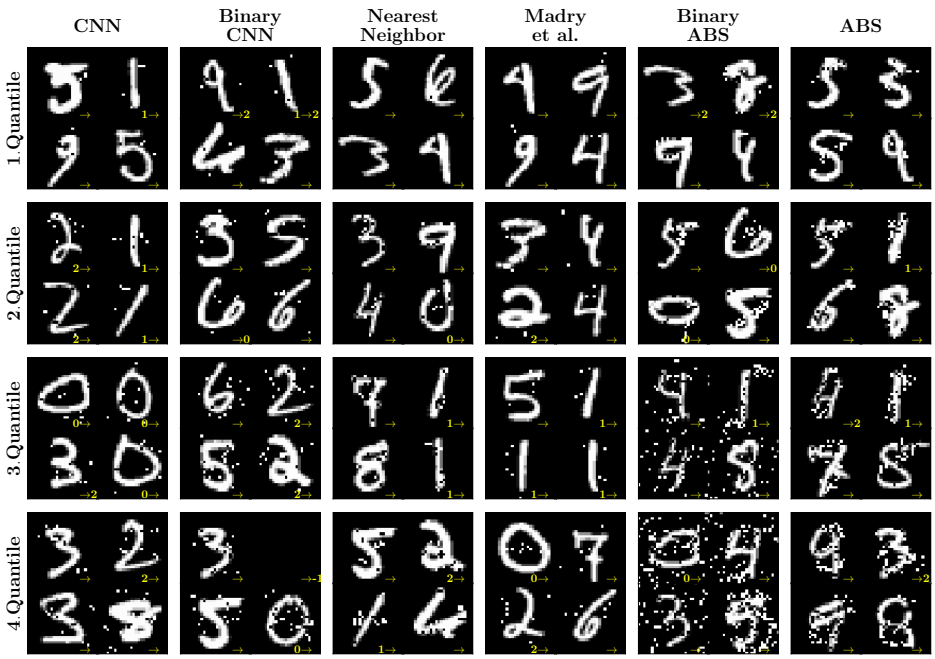


Figure 5: L_0 error quantiles: We always choose the minimally perturbed L_0 adversarial found by any attack for each model. For an unbiased selection, we then randomly sample images within four error quantiles ($0-25\%$, $25-50\%$, $50-75\%$, and $75-100\%$). Where 100% corresponds to the maximal (over samples) minimum (over attacks) perturbation found for each model.

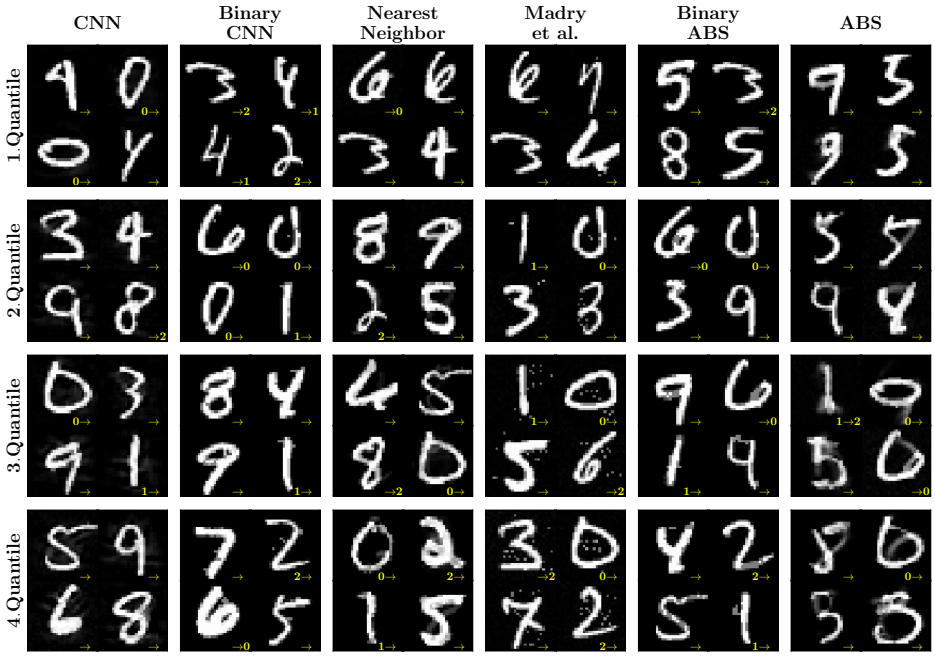


Figure 6: L_2 error quantiles: We always choose the minimally perturbed L_2 adversarial found by any attack for each model. For an unbiased selection, we then randomly sample 4 images within four error quantiles $(0-25\%,\ 25-50\%,\ 50-75\%,$ and $75-100\%)$.

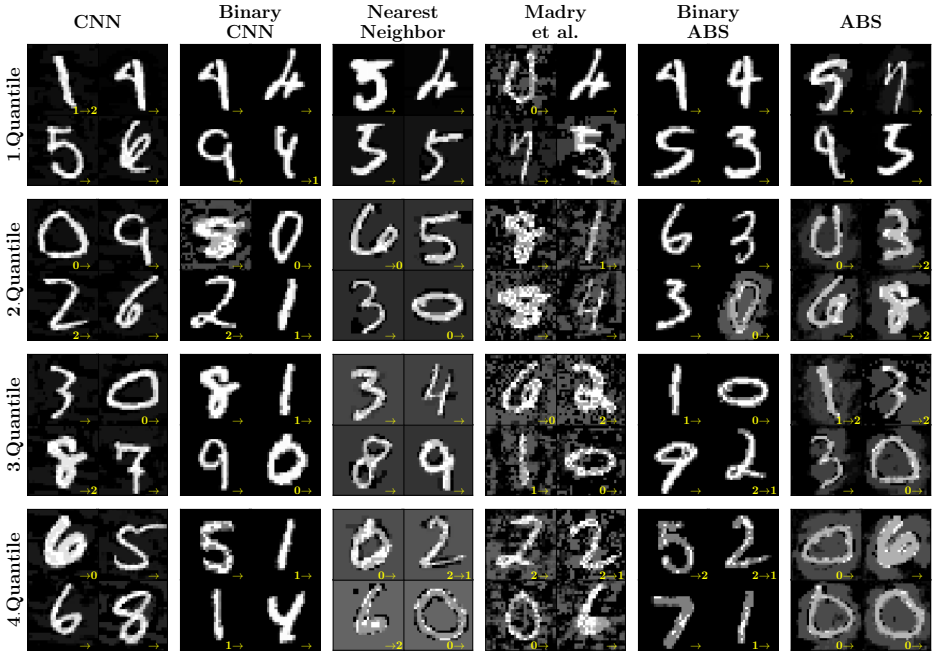
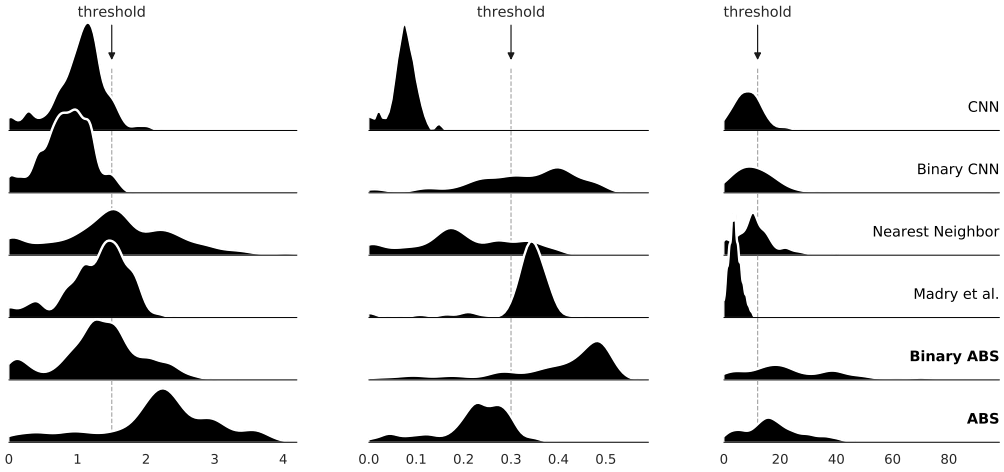


Figure 7: L_∞ error quantiles: We always choose the minimally perturbed L_∞ adversarial found by any attack for each model. For an unbiased selection, we then randomly sample images within four error quantiles $(0-25\%,\ 25-50\%,\ 50-75\%,$ and $75-100\%)$.



(a) L_2 distance (b) L_∞ distance (c) L_0 distance

Figure 8: Distribution of minimal adversarials for each model and distance metric. In (b) we see that a threshold at 0.3 favors Madry et al. while a threshold of 0.35 would have favored the Binary ABS.

A.3 DERIVATION I

Derivation of the ELBO in equation 2.

$$\log p_\theta(\mathbf{x}) = \log \int \mathbf{dz}\, p_\theta(\mathbf{x}|\mathbf{z}) p(\mathbf{z}),$$

where $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbb{1})$ is a simple normal prior. Based on the idea of importance sampling using a variational posterior $q_\phi(\mathbf{z}|\mathbf{x})$ with parameters ϕ and using Jensen's inequality we arrive at

$$\begin{aligned}
&= \log \int \mathbf{dz}\, \frac{q_\phi(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})} p_\theta(\mathbf{x}|\mathbf{z}) p(\mathbf{z}), \\
&= \log \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \left[\frac{p_\theta(\mathbf{x}|\mathbf{z}) p(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \right], \\
&\geq \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \left[\log \frac{p_\theta(\mathbf{x}|\mathbf{z}) p(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \right], \\
&= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \left[\log p_\theta(\mathbf{x}|\mathbf{z}) + \log \frac{p(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \right], \\
&= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \left[\log p_\theta(\mathbf{x}|\mathbf{z}) \right] - \mathcal{D}_{KL} \left[q_\phi(\mathbf{z}|\mathbf{x}) || p(\mathbf{z}) \right].
\end{aligned}$$

This lower bound is commonly referred to as ELBO.

A.4 DERIVATION II: LOWER BOUND FOR L_2 ROBUSTNESS ESTIMATION

Derivation of equation 6. Starting from equation 3 we find that for a perturbation $\boldsymbol{\delta}$ with size $\epsilon = \|\boldsymbol{\delta}\|_2$ of sample $\mathbf{x}$ the lower bound $\ell_y^*(\mathbf{x} + \boldsymbol{\delta})$ can itself be bounded by,

$$\begin{aligned}
\ell_y^*(\mathbf{x} + \boldsymbol{\delta}) &= \max_{\mathbf{z}} -\mathcal{D}_{KL} \left[\mathcal{N}(\mathbf{z}, \sigma_q \mathbb{1}) || \mathcal{N}(\mathbf{0}, \mathbb{1}) \right] - \frac{1}{2\sigma^2} \left\| \mathbf{G}_y(\mathbf{z}) - \mathbf{x} - \boldsymbol{\delta} \right\|_2^2 + C, \\
&\geq -\mathcal{D}_{KL} \left[\mathcal{N}(\mathbf{z}_\mathbf{x}^*, \sigma_q \mathbb{1}) || \mathcal{N}(\mathbf{0}, \mathbb{1}) \right] - \frac{1}{2\sigma^2} \left\| \mathbf{G}_y(\mathbf{z}_\mathbf{x}^*) - \mathbf{x} - \boldsymbol{\delta} \right\|_2^2 + C,
\end{aligned}$$

where $\mathbf{z}_\mathbf{x}^*$ is the optimal latent vector for the clean sample $\mathbf{x}$ for class y,

$$\begin{aligned}
&= \ell_y^*(\mathbf{x}) + \frac{1}{\sigma^2} \boldsymbol{\delta}^\top (\mathbf{G}_y(\mathbf{z}_\mathbf{x}^*) - \mathbf{x}) - \frac{1}{2\sigma^2} \epsilon^2 + C, \\
&\geq \ell_y^*(\mathbf{x}) - \frac{1}{\sigma^2} \epsilon \left\| \mathbf{G}_y(\mathbf{z}_\mathbf{x}^*) - \mathbf{x} \right\|_2 - \frac{1}{2\sigma^2} \epsilon^2 + C. \qquad (10)
\end{aligned}$$

A.5 DERIVATION III: UPPER BOUND FOR L_2 ROBUSTNESS ESTIMATION

Derivation of equation 7.

$$\begin{aligned}
\ell_c^*(\mathbf{x} + \boldsymbol{\delta}) &= \max_{\mathbf{z}} -\mathcal{D}_{KL} \left[\mathcal{N}(\mathbf{z}, \sigma_q \mathbb{1}) || \mathcal{N}(\mathbf{0}, \mathbb{1}) \right] - \frac{1}{2\sigma^2} \left\| \mathbf{G}_y(\mathbf{z}) - \mathbf{x} - \boldsymbol{\delta} \right\|_2^2 + C, \\
&\leq -\mathcal{D}_{KL} \left[\mathcal{N}(\mathbf{0}, \sigma_q \mathbb{1}) || \mathcal{N}(\mathbf{0}, \mathbb{1}) \right] + C - \min_{\mathbf{z}} \frac{1}{2\sigma^2} \left\| \mathbf{G}_c(\mathbf{z}) - \mathbf{x} - \boldsymbol{\delta} \right\|_2^2, \\
&\leq -\mathcal{D}_{KL} \left[\mathcal{N}(\mathbf{0}, \sigma_q \mathbb{1}) || \mathcal{N}(\mathbf{0}, \mathbb{1}) \right] + C - \min_{\mathbf{z}, \boldsymbol{\delta}} \frac{1}{2\sigma^2} \left\| \mathbf{G}_c(\mathbf{z}) - \mathbf{x} - \boldsymbol{\delta} \right\|_2^2, \\
&= -\mathcal{D}_{KL} \left[\mathcal{N}(\mathbf{0}, \sigma_q \mathbb{1}) || \mathcal{N}(\mathbf{0}, \mathbb{1}) \right] + C - \begin{cases} \frac{1}{2\sigma^2} (d_c - \epsilon)^2 & \text{if } d_c \geq \epsilon \\ 0 & \text{else} \end{cases}. \qquad (11)
\end{aligned}$$

for $d_c = \min_z \left\| \mathbf{G}_c(\mathbf{z}) - \mathbf{x} \right\|_2$. The last equation comes from the solution of the constrained optimization problem $\min_d (d - \epsilon)^2 d$ s.t. $d > d_c$. Note that a tighter bound might be achieved by assuming single $\boldsymbol{\delta}$ for upper and lower bound.

A.6 L_∞ ROBUSTNESS ESTIMATION

We proceed in the same way as for L_2. Starting again from

$$\ell_c^*(\mathbf{x}) = \max_{\mathbf{z}} \; -\mathcal{D}_{KL} \left[\mathcal{N}(\mathbf{z}, \sigma_q \mathbb{1}) || \mathcal{N}(\mathbf{0}, \mathbb{1}) \right] - \frac{1}{2\sigma^2} \left\| \mathbf{G}_c(\mathbf{z}) - \mathbf{x} \right\|_2^2 + C, \qquad (12)$$

let y be the predicted class and let $\mathbf{z}^*_{\mathbf{x}}$ be the optimal latent for the clean sample $\mathbf{x}$ for class y. We can then estimate a lower bound on $\ell^*_y(\mathbf{x}+\boldsymbol{\delta})$ for a perturbation $\boldsymbol{\delta}$ with size $\epsilon = \|\boldsymbol{\delta}\|_\infty$,

$$\begin{aligned}\ell^*_y(\mathbf{x}+\boldsymbol{\delta}) &= \max_{\mathbf{z}} -\mathcal{D}_{KL}\left[\mathcal{N}(\mathbf{z},\sigma_q\mathbb{1})||\mathcal{N}(\mathbf{0},\mathbb{1})\right] - \frac{1}{2\sigma^2}\left\|\mathbf{G}_y(\mathbf{z}) - \mathbf{x} - \boldsymbol{\delta}\right\|_2^2 + C,\\ &\geq -\mathcal{D}_{KL}\left[\mathcal{N}(\mathbf{z}^*_{\mathbf{x}},\sigma_q\mathbb{1})||\mathcal{N}(\mathbf{0},\mathbb{1})\right] - \frac{1}{2\sigma^2}\left\|\mathbf{G}_y(\mathbf{z}^*_{\mathbf{x}}) - \mathbf{x} - \boldsymbol{\delta}\right\|_2^2 + C,\end{aligned}$$

where $\mathbf{z}^*_{\mathbf{x}}$ is the optimal latent for the clean sample $\mathbf{x}$ for class y.

$$\begin{aligned}&= \ell^*_y(\mathbf{x}) + \frac{1}{\sigma^2}\boldsymbol{\delta}^\top(\mathbf{G}_y(\mathbf{z}^*_{\mathbf{x}}) - \mathbf{x}) - \frac{1}{2\sigma^2}\left\|\boldsymbol{\delta}\right\|_2^2 + C,\\ &\geq \ell^*_y(\mathbf{x}) + C + \frac{1}{2\sigma^2}\min_{\boldsymbol{\delta}}\left(2\boldsymbol{\delta}^\top(\mathbf{G}_y(\mathbf{z}^*_{\mathbf{x}}) - \mathbf{x}) - \left\|\boldsymbol{\delta}\right\|_2^2\right),\\ &= \ell^*_y(\mathbf{x}) + C + \frac{1}{2\sigma^2}\sum_i \min_{\delta_i}\left(2\delta_i[\mathbf{G}_y(\mathbf{z}^*_{\mathbf{x}}) - \mathbf{x}]_i - \delta_i^2\right),\\ &= \ell^*_y(\mathbf{x}) + C + \frac{1}{2\sigma^2}\sum_i \begin{cases}[\mathbf{G}_y(\mathbf{z}^*_{\mathbf{x}}) - \mathbf{x}]_i^2 & \text{if } |[\mathbf{G}_y(\mathbf{z}^*_{\mathbf{x}}) - \mathbf{x}]_i| \leq \epsilon\\ \epsilon\,|[\mathbf{G}_y(\mathbf{z}^*_{\mathbf{x}}) - \mathbf{x}]_i| & \text{else}\end{cases}.\end{aligned} \tag{13}$$

Similarly, we can estimate an upper bound on $\ell^*_c(\mathbf{x}+\boldsymbol{\delta})$ on all other classes $c \neq y$,

$$\begin{aligned}\ell^*_c(\mathbf{x}+\boldsymbol{\delta}) &\leq -\mathcal{D}_{KL}\left[\mathcal{N}(\mathbf{0},\sigma_q\mathbb{1})||\mathcal{N}(\mathbf{0},\mathbb{1})\right] + C - \min_{\mathbf{z}}\frac{1}{2\sigma^2}\left\|\mathbf{G}_c(\mathbf{z}) - \mathbf{x} - \boldsymbol{\delta}\right\|_2^2,\\ &\leq -\mathcal{D}_{KL}\left[\mathcal{N}(\mathbf{0},\sigma_q\mathbb{1})||\mathcal{N}(\mathbf{0},\mathbb{1})\right] + C - \min_{\mathbf{z},\boldsymbol{\delta}}\frac{1}{2\sigma^2}\left\|\mathbf{G}_c(\mathbf{z}) - \mathbf{x} - \boldsymbol{\delta}\right\|_2^2,\\ &= -\mathcal{D}_{KL}\left[\mathcal{N}(\mathbf{0},\sigma_q\mathbb{1})||\mathcal{N}(\mathbf{0},\mathbb{1})\right] + C - \min_{\mathbf{z}}\frac{1}{2\sigma^2}\sum_i \min_{\delta_i}\left([\mathbf{G}_c(\mathbf{z}) - \mathbf{x}]_i - \delta_i\right)^2,\\ &= -\mathcal{D}_{KL}\left[\mathcal{N}(\mathbf{0},\sigma_q\mathbb{1})||\mathcal{N}(\mathbf{0},\mathbb{1})\right] + C\\ &\quad - \min_{\mathbf{z}}\frac{1}{2\sigma^2}\sum_i \begin{cases}0 & \text{if } |[\mathbf{G}_y(\mathbf{z}^*_{\mathbf{x}}) - \mathbf{x}]_i| \leq \epsilon\\ ([\mathbf{G}_y(\mathbf{z}^*_{\mathbf{x}}) - \mathbf{x}]_i - \epsilon)^2 & \text{if } [\mathbf{G}_y(\mathbf{z}^*_{\mathbf{x}}) - \mathbf{x}]_i > \epsilon\\ ([\mathbf{G}_y(\mathbf{z}^*_{\mathbf{x}}) - \mathbf{x}]_i + \epsilon)^2 & \text{if } [\mathbf{G}_y(\mathbf{z}^*_{\mathbf{x}}) - \mathbf{x}]_i < \epsilon\end{cases}.\end{aligned} \tag{14}$$

In this case there is no closed-form solution for the minimization problem on the RHS (in terms of the minimum of $\|\mathbf{G}_c(\mathbf{z}) - \mathbf{x}\|_2$) but we can still compute the solution for each given ϵ which allows us perform a line search along ϵ to find the point where equation 13 = equation 14.

A.7 Model & training details

Hyperparameters and training details for the ABS model The binary ABS and ABS have the same weights and architecture: The encoder has 4 layers with kernel sizes$= [5,4,3,5]$, strides$= [1,2,2,1]$ and feature map sizes$= [32,32,64,2{*}8]$. The first 3 layers have ELU activation functions (Clevert et al., 2015), the last layer is linear. All except the last layer use Batch Normalization (Ioffe & Szegedy, 2015). The Decoder architecture has also 4 layers with kernel sizes$= [4,5,5,3]$, strides$= [1,2,2,1]$ and feature map sizes$= [32,16,16,1]$. The first 3 layers have ELU activation functions, the last layer has a sigmoid activation function, and all layers except the last one use Batch Normalization.

We trained the VAEs with the Adam optimizer (Kingma & Ba, 2014). We tuned the dimension L of the latent space of the class-conditional VAEs (ending up with $L = 8$) to achieve 99% test error; started with a high weight for the KL-divergence term at the beginning of training (which was gradually decreased from a factor of 10 to 1 over 50 epochs); estimated the weighting $\boldsymbol{\gamma} = [1, 0.96, 1.001, 1.06, 0.98, 0.96, 1.03, 1, 1, 1]$ of the lower bound via a line search on the training accuracy. The parameters maximizing the test cross entropy[3] and providing a median confidence of $p(y|x) = 0.9$ for our modified softmax (equation 8) are $\eta = 0.000039$ and $\alpha = 440$. For our latent prior, we chose $\sigma_q = 1$ and for the posterior width we choose $\sigma = 1/\sqrt{2}$

Hyperparameters for the CNNs The CNN and Binary CNN share the same architecture but have different weights. The architecture has kernel sizes $= [5,4,3,5]$, strides $= [1,2,2,1]$, and feature map sizes $= [20,70,256,10]$. All layers use ELU activation functions and all layers except the last one apply Batch Normalization. The CNNs are both trained on the cross entropy loss with the Adam optimizer (Kingma & Ba, 2014). The parameters maximizing the test cross entropy and providing a median confidence of $p(y|x) = 0.9$ of the CNN for our modified softmax (equation 8) are $\eta = 143900$ and $\alpha = 1$.

[3]Note that this solely scales the probabilities and does not change the classification accuracy.

Hyperparameters for Madry et al. We adapted the pre-trained model provided by Madry et al[4]. Basically the architecture contains two convolutional, two pooling and two fully connected layers. The network is trained on clean and adversarial examples minimizing the cross cross-entropy loss. The parameters maximizing the test cross entropy and providing a median confidence of $p(y|x) = 0.9$ for our modified softmax (equation 8) are $\eta = 60$ and $\alpha = 1$.

Hyperparameters for the Nearest Neighbour classifier For a comparison with neural networks, we imitate logits by replacing them with the negative minimal distance between the input and all samples within each class. The parameters maximizing the test cross entropy and providing a median confidence of $p(y|x) = 0.9$ for our modified softmax (equation 8) are $\eta = 0.000000000004$ and $\alpha = 5$.

[4] https://github.com/MadryLab/mnist_challenge