

# **Out-of-Distribution Generalization in Visual Perception**

## **Dissertation**

der Mathematisch-Naturwissenschaftlichen Fakultät  
der Eberhard Karls Universität Tübingen  
zur Erlangung des Grades eines  
Doktors der Naturwissenschaften  
(Dr. rer. nat.)

vorgelegt von

Haiwen Huang  
aus Zhejiang/China

Tübingen  
2026

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der  
Eberhard Karls Universität Tübingen.

|                                   |                          |
|-----------------------------------|--------------------------|
| Tag der mündlichen Qualifikation: | 13.07.2026               |
| Dekan:                            | Prof. Dr. Thilo Stehle   |
| 1. Berichterstatter:              | Prof. Dr. Andreas Geiger |
| 2. Berichterstatterin:            | Prof. Dr. Hilde Kuehne   |

© 2026 Haiwen Huang

This work is licensed under the Creative Commons Attribution 4.0 International License (CC BY 4.0). The full license text is available at

<https://creativecommons.org/licenses/by/4.0/legalcode>





# Abstract

Out-of-distribution (OOD) generalization remains a central obstacle to deploying visual perception systems in open-ended environments where models encounter objects, concepts, and conditions absent from their training data. Deep learning models achieve remarkable performance on standardized benchmarks. Yet they frequently fail when deployed in real-world settings that differ from their training distribution. This thesis addresses this challenge through the concept of *bridging invariants*—structural properties that remain sufficiently stable across distributional gaps and enable the transfer of knowledge from familiar to novel conditions. Rather than treating all distributional shifts generically, this work identifies and exploits task-specific invariants by leveraging foundation models as external sources of these properties and transferring them into task-specific models via pseudo-labeling.

The thesis presents three complementary contributions, each addressing a distinct type of generalization gap. First, GOOD (ICLR 2023) bridges the semantic shift gap by exploiting geometric structure—depth and surface normals shared across all physical objects—to discover novel objects via pseudo-labeling. Training with only a single annotated class, GOOD surpasses prior methods by 5.0% AR@100. Second, RENOVATE (NeurIPS 2024) bridges the label-space misalignment gap by exploiting vision-language alignment to generate precise, segment-level category names. RENOVATE yields a 15% relative mIoU improvement and matches  $5\times$  more training data with original names. Third, LoftUp (ICCV 2025 Oral) bridges the resolution gap by exploiting boundary structure from SAM masks to construct pseudo-groundtruth high-resolution features. With less than 20% additional parameters, LoftUp achieves 10–20% improvement across six dense prediction tasks. Across all three works, the shared mechanism is consistent: identify the invariant that holds across the distributional gap, extract it from a foundation model, and transfer it through pseudo-labels to guide the task-specific model.

Yet each of these bridges was built across a gap that a human researcher had already identified. Beyond constructing bridges, this thesis therefore discusses the upstream problem of gap discovery—systematically identifying distributional shifts rather than assuming they are known. A structured analysis of recent OOD research reveals that gap discovery remains the field’s least systematized activity, and a retrospective case study demonstrates that automated tools from adjacent communities can accelerate this process. The thesis concludes by analyzing the biases that bridging invariants themselves introduce and identifying open problems at the intersection of gap discovery and invariant design. Taken together, these contributions establish bridging invariants as a principled and practical framework for OOD generalization in visual perception.



# Kurzfassung

Die Generalisierung auf Daten außerhalb der Trainingsverteilung (Out-of-Distribution, OOD) stellt ein zentrales Hindernis für den Einsatz visueller Wahrnehmungssysteme in offenen Umgebungen dar, in denen Modelle auf Objekte, Konzepte und Bedingungen treffen, die in ihren Trainingsdaten nicht enthalten waren. Während Deep-Learning-Modelle auf standardisierten Benchmarks bemerkenswerte Leistungen erzielen, versagen sie häufig in realen Einsatzszenarien, die sich von ihrer Trainingsverteilung unterscheiden. Diese Dissertation begegnet dieser Herausforderung durch das Konzept der *überbrückenden Invarianten* (bridging invariants) — strukturelle Eigenschaften, die über Verteilungslücken hinweg hinreichend stabil bleiben und den Wissenstransfer von bekannten zu neuartigen Bedingungen ermöglichen. Anstatt alle Verteilungsverschiebungen generisch zu behandeln, identifiziert und nutzt diese Arbeit aufgabenspezifische Invarianten, indem sie Foundation Models als externe Quellen dieser Eigenschaften einsetzt und sie über Pseudo-Labeling in aufgabenspezifische Modelle überträgt.

Die Dissertation präsentiert drei komplementäre Beiträge, die jeweils eine unterschiedliche Art von Generalisierungslücke adressieren. Erstens nutzt GOOD (Geometry-guided Open-world Object Detector, ICLR 2023) geometrische Struktur als überbrückende Invariante für Open-World-Objekterkennung. Das Verfahren trainiert ein Objektvorschlags-Netzwerk auf geometrischen Merkmalen monokularer Tiefen- und Normalenschätzer, um neuartige Objekte im Trainingsdatensatz zu entdecken, und verwendet diese Entdeckungen als Pseudo-Labels zum Training eines RGB-basierten Detektors. Mit nur einer einzigen annotierten Klasse übertrifft GOOD bisherige Methoden um 5,0% AR@100 auf neuartigen Kategorien. Zweitens identifiziert RENOVATE (NeurIPS 2024) die Vision-Language-Ausrichtung als überbrückende Invariante für Open-Vocabulary-Segmentierung. Das Verfahren nutzt Bildbeschreibungsmodele zur Generierung kontextuell relevanter Namenskandidaten und trainiert ein Umbenennungsmodell zur Auswahl der besten Übereinstimmung für jede Segmentierungsmaske. Umbenannte Kategorienamen verbessern sowohl Training als auch Evaluation: Modelle, die auf  $10^4$  Bildern mit erneuerten Namen trainiert werden, erreichen die Leistung von Modellen, die auf  $5 \times 10^4$  Bildern mit Originalnamen trainiert werden, und die beste Konfiguration erzielt eine relative Verbesserung von 15% im mIoU. Drittens nutzt LoftUp (ICCV 2025, Oral) Grenz- und Regionsstruktur als überbrückende Invariante, um grobe Merkmale von Vision Foundation Models auf Pixelauflösung hochzurechnen. Das Verfahren konstruiert Pseudo-Grundwahrheits-Merkmale hoher Auflösung mittels klassenagnostischer SAM-Masken und trainiert einen koordinatenbasierten Cross-Attention-Transformer für das Upsampling unter Beibehaltung von Objektgrenzen. Mit weniger als 20% zusätzlichen Parametern erzielt LoftUp Verbesserungen von 10–20% über sechs dichte Vorhersageaufgaben hinweg. Über alle drei Arbeiten hinweg ist der gemeinsame Mechanismus konsistent: Identifikation der Invariante, die über die Vertei-

lungslücke hinweg gilt, Extraktion aus einem Foundation Model und Übertragung durch Pseudo-Labels zur Steuerung des aufgabenspezifischen Modells.

Über die Konstruktion von Brücken hinaus diskutiert diese Dissertation das vorgelagerte Problem der Lückenentdeckung (Gap Discovery) — die systematische Identifikation von Verteilungsverschiebungen, anstatt diese als bekannt vorauszusetzen. Eine strukturierte Analyse aktueller OOD-Forschung zeigt, dass die Lückenentdeckung (Gap Discovery) die am wenigsten systematisierte Aktivität des Feldes bleibt, und eine retrospektive Fallstudie demonstriert, dass automatisierte Werkzeuge aus benachbarten Fachgebieten diesen Prozess beschleunigen können. Die Dissertation schließt mit einer Analyse der Verzerrungen, die überbrückende Invarianten selbst einführen, und identifiziert offene Probleme an der Schnittstelle von Lückenentdeckung und Invarianten-Design. Zusammengefasst etablieren diese Beiträge überbrückende Invarianten als ein prinzipiengeleitetes und praktisches Rahmenwerk für OOD-Generalisierung in der visuellen Wahrnehmung.

# Acknowledgments

First and foremost, I would like to thank my two supervisors, Andreas Geiger and Dan Zhang. I am deeply grateful that you gave me the opportunity to pursue a PhD in such an inspiring environment. From both of you, I learned how to do research—how to identify the right questions, how to design rigorous experiments, how to write and present clearly—and through that process I gained a confidence in conducting high-quality research that I did not have before.

Andreas, thank you for giving me both the freedom and the framework to grow as an independent researcher. You taught me a way of thinking about research that I carry with me: what are the “obvious” problems that should be solved, what are the right baselines, what should our goals be. Your fundamental questions shaped how I approach problems, and your high standards showed me what rigorous work looks like. I especially appreciate the research atmosphere you created, free from unnecessary distractions, and I am grateful for the retreats, the conference trips, and all the moments that made this journey enjoyable. Beyond research, I have learned from you something harder to put into words: your integrity, your focus and passion for your work, and your care and tolerance for others while holding yourself to the highest standards. These are lessons that extend far beyond academia.

Dan, thank you for your continuous support and guidance throughout my PhD. You have been an invaluable source of insights and ideas that shaped the direction of my work in ways I could not have arrived at alone. From you I learned how to communicate with reviewers—how to respond to criticism constructively, how to strengthen a paper through the revision process. Before my PhD, I had never successfully navigated a borderline or negative review; the skills you taught me made all the difference. Your detailed feedback on manuscripts, your patience in discussing experimental design, and your encouragement during difficult periods kept me going when I needed it most. I am also deeply grateful for the support you secured through the Bosch Industry on Campus Lab at the University of Tübingen.

I would like to thank my collaborators, whose contributions were essential to the work in this thesis. Anpei Chen, thank you for your substantial contributions to LoftUp—from the core experiments to the intense rebuttal period, your dedication and skill were indispensable. Songyou Peng, thank you for your collaboration on RENOVATE. Volodymyr Havrylov—my student as well as my collaborator—thank you for your motivated work and solid experiments; it has been a pleasure working with someone so dedicated.

I am also grateful to the many colleagues, mentors, and friends who shaped my thinking through discussions, feedback, and shared curiosity: Prof. Bob Williamson, Kashyap Chitta, Haoyu He, Katrin Renz, Haofei Xu, Michael Niemeyer, Tim Zhenzhong Xiao, and many more. From my internship at Google, I would like to thank Yongqin Xian, Alessio

Tonioni, and Federico Tombari for a stimulating and formative experience.

Finally, I want to thank my family. To my wife, Tianlin—thank you for your unwavering support, your patience during the long evenings and weekends of paper deadlines, and your help with drawing figures in many of my papers. They would not have looked nearly as good without you. And to Coco, our dog, who provided a steady source of emotional support and life stability throughout these years. The daily walks, the quiet companionship, and the unconditional enthusiasm at the door reminded me that there is a world beyond the screen.

# Contents

|                                                                            |             |
|----------------------------------------------------------------------------|-------------|
| <b>Abstract</b>                                                            | <b>iii</b>  |
| <b>Kurzfassung</b>                                                         | <b>v</b>    |
| <b>Acknowledgments</b>                                                     | <b>vii</b>  |
| <b>List of Figures</b>                                                     | <b>xiii</b> |
| <b>List of Tables</b>                                                      | <b>xix</b>  |
| <b>1 Introduction</b>                                                      | <b>1</b>    |
| 1.1 Motivation: The Generalization Problem in Visual Perception . . . . .  | 1           |
| 1.2 The Generalization Gap . . . . .                                       | 1           |
| 1.3 Bridging Invariants: The Core Idea . . . . .                           | 2           |
| 1.4 Contributions . . . . .                                                | 5           |
| 1.4.1 The Geometric Bridge: Open-World Object Detection (GOOD) . . . . .   | 5           |
| 1.4.2 The Semantic Bridge: Renovating Benchmark Names (RENOVATE) . . . . . | 5           |
| 1.4.3 The Spatial Bridge: Feature Upsampling (LoftUp) . . . . .            | 6           |
| 1.4.4 The Upstream Problem: Gap Discovery (Discussion) . . . . .           | 6           |
| 1.5 Thesis Organization . . . . .                                          | 6           |
| 1.6 Publications . . . . .                                                 | 7           |
| <b>2 Background</b>                                                        | <b>9</b>    |
| 2.1 Out-of-Distribution Generalization . . . . .                           | 9           |
| 2.1.1 Problem Settings: A Taxonomy of Distribution Shift . . . . .         | 9           |
| 2.1.2 Classical Theory: Bounding Generalization Under Shift . . . . .      | 12          |
| 2.1.3 Invariance-Based Approaches to OOD Generalization . . . . .          | 12          |
| 2.1.4 Shortcut Learning and Under-Determination . . . . .                  | 14          |
| 2.2 Invariance and Representation Learning . . . . .                       | 14          |
| 2.2.1 Representations as Bridges . . . . .                                 | 15          |
| 2.2.2 Pseudo-Labeling as Invariant Transfer . . . . .                      | 16          |
| 2.3 Foundation Models as Invariant Sources . . . . .                       | 17          |
| 2.3.1 What Are Foundation Models? . . . . .                                | 17          |
| 2.3.2 Why Foundation Models Carry Bridging Invariants . . . . .            | 19          |
| 2.3.3 Limitations: What Foundation Models Cannot Provide . . . . .         | 19          |
| 2.4 From Background to Bridges . . . . .                                   | 20          |

|          |                                                                                           |           |
|----------|-------------------------------------------------------------------------------------------|-----------|
| <b>3</b> | <b>Geometry-Guided Open-World Object Detection (GOOD)</b>                                 | <b>21</b> |
| 3.1      | Introduction . . . . .                                                                    | 21        |
| 3.2      | Related Work . . . . .                                                                    | 24        |
| 3.3      | Method . . . . .                                                                          | 25        |
| 3.3.1    | Open-World Class-Agnostic Object Detection Problem . . . . .                              | 25        |
| 3.3.2    | Exploiting Geometric Cues . . . . .                                                       | 27        |
| 3.3.3    | Pseudo Labeling Method . . . . .                                                          | 27        |
| 3.4      | Experiments . . . . .                                                                     | 28        |
| 3.4.1    | Detecting Unknown Objects in an Open World . . . . .                                      | 29        |
| 3.4.2    | Advantages of Geometric Cues . . . . .                                                    | 31        |
| 3.4.3    | Ablation Studies . . . . .                                                                | 34        |
| 3.5      | Conclusion . . . . .                                                                      | 39        |
| 3.6      | Reflection: The Geometric Bridge in Retrospect . . . . .                                  | 39        |
| <b>4</b> | <b>Renovating Names in Open-Vocabulary Segmentation Benchmarks (RENOVATE)</b>             | <b>41</b> |
| 4.1      | Introduction . . . . .                                                                    | 41        |
| 4.2      | Related Work . . . . .                                                                    | 43        |
| 4.3      | RENOVATE: Renaming Segmentation Benchmarks . . . . .                                      | 43        |
| 4.3.1    | Generating candidate names . . . . .                                                      | 44        |
| 4.3.2    | Training for candidate name selection . . . . .                                           | 45        |
| 4.3.3    | Obtaining renovated names . . . . .                                                       | 47        |
| 4.4      | Applications of RENOVATE Names . . . . .                                                  | 48        |
| 4.4.1    | Training with RENOVATE names . . . . .                                                    | 48        |
| 4.4.2    | Improving evaluation with RENOVATE names . . . . .                                        | 49        |
| 4.5      | Experiments . . . . .                                                                     | 50        |
| 4.5.1    | Obtaining RENOVATE names . . . . .                                                        | 50        |
| 4.5.2    | Training with RENOVATE names . . . . .                                                    | 51        |
| 4.5.3    | Improving evaluation with RENOVATE names . . . . .                                        | 53        |
| 4.5.4    | Human preference study . . . . .                                                          | 55        |
| 4.5.5    | Ablation analysis on the renaming pipeline . . . . .                                      | 56        |
| 4.5.6    | Full evaluation results . . . . .                                                         | 57        |
| 4.5.7    | More name examples . . . . .                                                              | 57        |
| 4.6      | Conclusion and Limitations . . . . .                                                      | 58        |
| 4.7      | Reflection: The Semantic Bridge in Retrospect . . . . .                                   | 60        |
| <b>5</b> | <b>LoftUp: Learning a Coordinate-Based Feature Upsampler for Vision Foundation Models</b> | <b>63</b> |
| 5.1      | Introduction . . . . .                                                                    | 63        |
| 5.2      | Related Work . . . . .                                                                    | 65        |
| 5.3      | Coordinate-Based Feature Representation . . . . .                                         | 66        |
| 5.4      | Training Objective . . . . .                                                              | 67        |
| 5.4.1    | Stage 1: Training with class-agnostic masks . . . . .                                     | 68        |
| 5.4.2    | Stage 2: Training with self-distillation . . . . .                                        | 70        |



|          |                                                              |            |
|----------|--------------------------------------------------------------|------------|
| 5.5      | Experiments . . . . .                                        | 72         |
| 5.5.1    | Cross-Task Comparison . . . . .                              | 73         |
| 5.5.2    | More Analysis of LoftUp Design . . . . .                     | 74         |
| 5.5.3    | Generalization to Other VFM Backbones . . . . .              | 77         |
| 5.5.4    | Task-Specific Implementation Details . . . . .               | 78         |
| 5.5.5    | Additional Visualizations . . . . .                          | 79         |
| 5.6      | Conclusion . . . . .                                         | 79         |
| 5.7      | Reflection: The Spatial Bridge in Retrospect . . . . .       | 80         |
| <b>6</b> | <b>Beyond Known Unknowns — Discussion and Outlook</b>        | <b>85</b>  |
| 6.1      | The Gap Identification Problem . . . . .                     | 86         |
| 6.1.1    | The structure of OOD research . . . . .                      | 86         |
| 6.1.2    | Known unknowns, unknown unknowns, and the open world . . .   | 87         |
| 6.1.3    | The field’s uneven progress . . . . .                        | 88         |
| 6.2      | Foundation Models and the Limits of Coverage . . . . .       | 90         |
| 6.2.1    | Implicit coverage through scale . . . . .                    | 90         |
| 6.2.2    | Structural limits . . . . .                                  | 91         |
| 6.3      | Failure Discovery Across Communities . . . . .               | 92         |
| 6.3.1    | Failure discovery vs. gap discovery . . . . .                | 92         |
| 6.3.2    | A continuum of discovery approaches . . . . .                | 93         |
| 6.4      | Case Study: Retroactive Gap Discovery for RENOVATE . . . . . | 97         |
| 6.4.1    | Layer 1: CLIP text-space diagnostic . . . . .                | 98         |
| 6.4.2    | Layer 2: LLM-based diagnostic . . . . .                      | 98         |
| 6.4.3    | Layer 3: Validation against RENOVATE . . . . .               | 100        |
| 6.4.4    | What the case study shows . . . . .                          | 101        |
| 6.5      | What Remains Hard . . . . .                                  | 102        |
| 6.5.1    | What can we find: The limits of discovery . . . . .          | 102        |
| 6.5.2    | Which gaps matter: The meaningfulness problem . . . . .      | 103        |
| 6.5.3    | When have we found enough: Completeness and evaluation . . . | 104        |
| 6.5.4    | What do we do about it: From gaps to invariants . . . . .    | 105        |
| 6.6      | Concluding Remarks . . . . .                                 | 106        |
| <b>7</b> | <b>Conclusion</b>                                            | <b>107</b> |
| 7.1      | Summary of Contributions . . . . .                           | 107        |
| 7.2      | The Bias of the Bridge . . . . .                             | 108        |
| 7.3      | From Bridging to Discovery . . . . .                         | 109        |
| 7.4      | Final Thoughts . . . . .                                     | 110        |
|          | <b>Bibliography</b>                                          | <b>111</b> |



# List of Figures

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |    |
|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 1.1 | <b>The bridging invariant pipeline.</b> A distributional gap separates training from deployment (top). A foundation model extracts a <i>bridging invariant</i> —a structural property shared across both distributions (middle). This invariant is converted into pseudo-labels that guide the task-specific model (bottom). The right column shows the three instantiations developed in this thesis. Figure 2.3 formalizes this framework with mathematical notation. .                                                                                                                                                  | 3  |
| 2.1 | <b>A taxonomy of distribution shift.</b> Following the two-dimensional framework of Ye et al. [228], distribution shifts can be characterized along diversity shift (changes in input statistics) and correlation shift (changes in feature-label relationships). The four thesis-specific gaps are placed within this framework: domain/covariate shift involves diversity changes with stable correlations; semantic shift (Ch. 3) and label-space misalignment (Ch. 4) involve correlation shifts; resolution shift (Ch. 5) is a structural mismatch that sits outside the traditional diversity/correlation dichotomy. | 10 |
| 2.2 | <b>Shortcut learning as a cause of OOD failure.</b> (a) During training, the model exploits a spurious correlation between background context (grass) and the label (cow), since this is the simplest predictor that minimizes training loss. (b) At test time, when the cow appears against a novel background (beach), the shortcut fails and the model misclassifies the object. Bridging invariants address this by providing features—such as geometric structure—that are robust to such contextual shifts. . . . .                                                                                                  | 14 |
| 2.3 | <b>Formal view of bridging invariants.</b> This figure formalizes the pipeline introduced schematically in Fig. 1.1. A mapping $\phi$ projects both the source distribution $P_S$ and target distribution $P_T$ independently into an invariant space $\mathcal{Z}$ where distributional divergence is small. This thesis instantiates this framework through three distinct invariants: geometric structure for open-world detection (Ch. 3), vision-language alignment for open-vocabulary segmentation (Ch. 4), and boundary structure for feature upsampling (Ch. 5).                                                  | 15 |

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |    |
|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 3.1 | <b>Comparison of GOOD with different baselines.</b> Images in the first column are from the ADE20K [242] validation set. From the second to fourth columns we show the detection results of OLN [96], GGN [209], and our GOOD. Shown are true-positive proposals from the top 100 proposals of each method; the number of true positives is in parentheses. All models are trained on the PASCAL-VOC classes of COCO [117]. Both OLN and GGN fail to detect many objects not seen during training, while GOOD generalizes better by exploiting geometric cues. . . . .                                                                                                                                   | 22 |
| 3.2 | <b>Geometry cues are complementary to appearance cues for object localization.</b> The depth and normal cues of the RGB image are extracted using off-the-shelf general-purpose monocular predictors. <b>Left:</b> Geometric cues abstract away the appearance details and focus on more holistic information such as object shapes and relative spatial locations (depth) and directional changes (normals). <b>Right:</b> By incorporating geometric cues, GOOD generalizes better than the RGB-based model OLN [96], i.e., much smaller AR gaps between the base and novel classes. . . . .                                                                                                           | 23 |
| 3.3 | <b>Overview of the geometry-guided pseudo labeling method.</b> It consists of two training phases. Phase I: the RGB input is firstly preprocessed by the off-the-shelf model to extract the geometry cues, which are then used to train an object proposal network with the base class bounding box annotations. The proposal networks then pseudo-label the training samples, discovering unannotated novel objects. The top-ranked pseudo boxes are added to the annotation pool for Phase II training, i.e., a class-agnostic object detector is directly trained on the RGB input using both the base class and pseudo annotations. At inference time, we only need the model from Phase II. . . . . | 26 |
| 3.4 | <b>Visualization of top-3 pseudo boxes</b> generated by proposal networks trained on RGB (left), depth (center), and normal (right) inputs. Geometric modalities consistently discover volumetric objects with distinctive 3D structure that the RGB-based model misses. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                         | 32 |
| 3.5 | <b>Per-novel-class AR@5 difference comparison of pseudo boxes on COCO VOC <math>\rightarrow</math> Non-VOC.</b> We train the object proposal network on the geometric cues (Phase-I in Figure 3.3) and also directly on the RGB image. We show their per-novel-class AR@5 differences, which is defined as $(AR_X - AR_{RGB})/AR_{RGB}$ with $X \in \{\text{Depth}, \text{Normal}\}$ . The geometric cues outperform the RGB image on those classes above the zero difference line. . . . .                                                                                                                                                                                                              | 33 |
| 3.6 | <b>Ablation studies.</b> (a) and (b) are conducted on COCO VOC to non-VOC benchmark. The study on the number of base classes (c) uses ADE20K as the evaluation benchmark. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | 35 |
| 3.7 | <b>Additional ablation studies.</b> These experiments investigate different approaches to ensembling geometric cues and the role of RGB in Phase-I training. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 36 |

|      |                                                                                                                                                                                                                                                                                                                                                                                     |    |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 3.8  | <b>Overlap heatmap of top-ranked pseudo boxes</b> across different modalities at IoU threshold 0.7 on COCO VOC→non-VOC. The low overlap confirms that different sources discover complementary sets of novel objects.                                                                                                                                                               | 37 |
| 3.9  | <b>Greedy top-<math>k</math> ensemble of pseudo boxes.</b> $AR_N@100$ improves as pseudo boxes from additional modalities are progressively added in the order: depth → normal → PA → edge → RGB. . . . .                                                                                                                                                                           | 38 |
| 4.1  | <b>Problems of names in current segmentation benchmarks.</b> We demonstrate examples from well-known datasets: MS COCO [117], ADE20K [241], and Cityscapes [33]. Our renovated names are visually more aligned and help models to generalize better. . . . .                                                                                                                        | 42 |
| 4.2  | <b>Overview of candidate name generation and renaming model training.</b> We generate candidate names based on the context names and train the renaming model to match them with the segments. For illustration clarity, we show only one segment. In practice, multiple segments are jointly trained, pairing with the text queries. . . . .                                       | 45 |
| 4.3  | <b>Obtaining renovated names.</b> In (a) we illustrate how we use the renaming model to obtain a renovated name for each segment. In (b) we demonstrate that the renaming results are helpful to dataset analysis with examples from “person” class. . . . .                                                                                                                        | 47 |
| 4.4  | <b>Examples of renovated names on segments</b> from the validation sets of ADE20K and Cityscapes. For each segment, we show the original name below the image and the renovated name in the text box. . . . .                                                                                                                                                                       | 48 |
| 4.5  | <b>RENOVATE uncovers segments with shared semantic concepts across datasets.</b> . . . . .                                                                                                                                                                                                                                                                                          | 51 |
| 4.6  | <b>Data efficiency comparison between RENOVATE and original names.</b>                                                                                                                                                                                                                                                                                                              | 52 |
| 4.7  | <b>Effectiveness of negative sampling.</b> . . . . .                                                                                                                                                                                                                                                                                                                                | 52 |
| 4.8  | <b>Open-vocabulary evaluation on ADE20K with different names, metrics, and models.</b> . . . . .                                                                                                                                                                                                                                                                                    | 53 |
| 4.9  | <b>RENOVATE names enable more fine-grained analysis on models.</b> (a) Per-category IoU with highlighted top/bottom-5 RENOVATE names and selected names from the “person” class in ADE20K. (b) Confusion matrix of frequently misclassified RENOVATE names from “building” to “wall”, showing misclassification proportions (numbers) and pairwise name similarity (color). . . . . | 54 |
| 4.10 | <b>RENOVATE can find wrong annotations and suggest corrections.</b> . .                                                                                                                                                                                                                                                                                                             | 55 |
| 4.11 | <b>Human preference study.</b> A survey of 20 researchers is conducted to compare preferences between the original names versus RENOVATE names on the validation sets. RENOVATE names are favored in 82% cases. . . .                                                                                                                                                               | 55 |
| 4.12 | <b>More examples of fine-grained misclassification analysis.</b> . . . . .                                                                                                                                                                                                                                                                                                          | 58 |
| 4.13 | <b>More examples of renovated names on segments</b> from the validation set of ADE20K. For each segment, we show the original name below the image and the renovated name in the text box. . . . .                                                                                                                                                                                  | 61 |

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |    |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 4.14 | <b>More examples of renovated names on segments</b> from the validation set of COCO. For each segment, we show the original name below the image and the renovated name in the text box. . . . .                                                                                                                                                                                                                                                                                    | 62 |
| 5.1  | <b>LoftUp improves significantly across various tasks over the VFM backbone (DINOv2-S [145]) and current SoTA feature upsampling performance (FeatUp [52] and LiFT [196]).</b> See experiment details in Section 5.5. . . . .                                                                                                                                                                                                                                                       | 64 |
| 5.2  | <b>Comparison of features from upsamplers.</b> Backbone is DINOv2-S/14 [145]. . . . .                                                                                                                                                                                                                                                                                                                                                                                               | 66 |
| 5.3  | <b>Architecture of LoftUp.</b> Our coordinate-based network with cross-attention mechanism effectively integrates the fine-grained details from image RGB values and semantically-rich low-res features to produce high-resolution feature maps. . . . .                                                                                                                                                                                                                            | 67 |
| 5.4  | <b>Our two-stage LoftUp training approach.</b> Stage 1 trains an upsampler with class-agnostic masks to refine bicubic-upsampled features. Stage 2 employs self-distillation, initializing teacher and student upsamplers from Stage 1’s pre-trained model. All VFM image inputs share the resolution ( $H \times W$ ). For visual clarity, the VFM block is omitted from Stage 2’s teacher branch. . . . .                                                                         | 68 |
| 5.5  | <b>Visualization of different pseudo-GT.</b> Both Mask-Bicubic and Self-Distilled are proposed by our work. We set $\alpha = 0.8$ (in Eq. (5.1)) to balance sharp boundaries from masks and fine-grained details from high-res features. . . . .                                                                                                                                                                                                                                    | 69 |
| 5.6  | <b>Visualization of prediction examples</b> on semantic segmentation on COCO-Stuff [17] and depth estimation on NAVI [86]. . . . .                                                                                                                                                                                                                                                                                                                                                  | 71 |
| 5.7  | <b>LoftUp pseudo-GT improves different upsamplers.</b> Here we report semantic segmentation results on COCO. . . . .                                                                                                                                                                                                                                                                                                                                                                | 74 |
| 5.8  | <b>Visualization of attended regions</b> (shown as <b>dots</b> ) in the low-resolution features corresponding to a high-resolution pixel (marked as a <b>cross</b> ). The density of <b>dots</b> represents the values in the attention map. LoftUp leverages relevant information from the global feature map to upsample features at each pixel. . . . .                                                                                                                          | 76 |
| 5.9  | <b>Visualization of prediction examples</b> of video object segmentation on the DAVIS 2017 dataset [153]. Each image displays its corresponding frame number in the top right corner. The groundtruth segmentation for the 0-th frame is provided, and dense feature affinity maps are employed to propagate its segmentation labels to subsequent frames. We can see that LoftUp outperforms all the other baselines in accurately tracking the objects across the frames. . . . . | 79 |
| 5.10 | <b>More visualization of predictions examples</b> on semantic segmentation on COCO-Stuff [17] and depth estimation on NAVI [86]. . . . .                                                                                                                                                                                                                                                                                                                                            | 80 |
| 5.11 | <b>More visualization of different pseudo-GT.</b> . . . .                                                                                                                                                                                                                                                                                                                                                                                                                           | 82 |
| 5.12 | <b>More visualization of features from various upsamplers.</b> Backbone is DINOv2-S/14 [145]. . . . .                                                                                                                                                                                                                                                                                                                                                                               | 83 |

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |     |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 5.13 | <b>More examples of attended regions</b> (in <b>dots</b> ) in the low-res features of a high-res pixel (in <b>cross</b> ). The density of <b>dots</b> reflects the value of the attention map. LoftUp uses relevant information across the global feature map for upsampling features at each pixel. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                       | 84  |
| 6.1  | The gap–discovery pipeline from distributional gap ( $\Delta$ ) through hypothesis ( $H$ ) to method ( $M$ ). Each thesis chapter instantiates this pipeline: Chapters 3–5 contributed at the $H$ and $M$ stages; this chapter examines the $\Delta$ stage. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 86  |
| 6.2  | Structured analysis of 573 OOD-related papers across eight venues (NeurIPS, ICML, ICLR, CVPR, ICCV, ECCV, ACL, EMNLP; 2022–2025), classified by two independent LLM coders (Cohen’s $\kappa = 0.71$ – $0.75$ ). Error bars show Wilson 95% CIs. <b>(a)</b> 85.9% of papers inherit their distributional shift ( $\Delta$ ) from existing benchmarks; only 14.1% introduce a new one. <b>(b)</b> 85.9% focus novelty on the method ( $M$ ), 7.9% on the hypothesis ( $H$ ), and only 6.2% on gap identification ( $\Delta$ ). Only 2.6% describe systematic methodology for gap discovery. <b>(c)</b> Standard domain-generalization benchmarks (PACS, VLCS, OfficeHome) account for the largest share of inherited shifts. . . . . | 90  |
| 6.3  | The coverage frontier. The training distribution (orange) is surrounded by known, benchmarked gaps (green) that bridging invariants (Chapters 3–5) are designed to cross. Foundation models may expand coverage further (blue, dashed to indicate potential rather than guaranteed coverage), but the vast region of unknown unknowns (red) persists beyond the epistemic frontier. Gap discovery methodology is needed to map this unmapped territory. . . . .                                                                                                                                                                                                                                                                    | 92  |
| 6.4  | <b>Top 20 most confusable ADE20K-150 class-name pairs</b> in CLIP text space. Red bars indicate pairs where at least one category has RENOVATE sub-names; grey bars are pairs where neither category was renamed. Nearly all top pairs involve synonym or hypernym–hyponym relationships. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                  | 99  |
| 6.5  | <b>CLIP vs. LLM diagnostic overlap.</b> Of 39 names flagged as problematic across both methods, only 4 appear in both sets, confirming that embedding-based and reasoning-based diagnostics surface complementary aspects of the naming gap. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 99  |
| 6.6  | <b>Text-space confusability before and after RENOVATE.</b> Each point is one of 116 ADE20K classes with RENOVATE sub-names. The horizontal axis shows confusability of the original name (max cosine similarity to any other class); the vertical axis shows the average confusability across RENOVATE sub-names. Points below the diagonal (green) indicate improvement; points above (red) indicate degradation. Renaming reduces confusability for 79 of 116 categories, with a mean reduction of 0.010. . .                                                                                                                                                                                                                    | 100 |





# List of Tables

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |    |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 1.1 | <b>Overview of the three bridging invariants</b> developed in this thesis. Each row summarizes how a specific generalization gap is addressed by identifying an invariant, extracting it from a foundation model, and transferring it via pseudo-labeling. . . . .                                                                                                                                                                                                                                                                                                                                                                                                | 4  |
| 2.1 | <b>Foundation model families used in this thesis.</b> Each family provides access to a different bridging invariant through its pre-training objective and learned representations. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 17 |
| 3.1 | <b>Detecting unknown objects in an open world.</b> FRCNN (oracle) is a standard Faster R-CNN detector trained on all COCO classes and serves as a performance upper bound on the cross-category benchmarks. FRCNN (cls-agn) is a Faster R-CNN trained in a class-agnostic manner, serving as a baseline for comparison. Our geometry-guided methods GOOD-Xs are compared with SOTA open-world class-agnostic object detectors, i.e., OLN and GGN. SelfTrain-RGB is RGB-based self-training, i.e., using the RGB image instead of geometric cues for pseudo labeling. We only report $AR_A$ in b) as the classes in ADE20K do not exactly match those in COCO. . . | 30 |
| 3.2 | <b>More benchmarks.</b> We evaluate on the LVIS COCO to non-COCO cross-category benchmark and the COCO to UVO cross-dataset benchmark. . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | 31 |
| 3.3 | <b>Comparison on different data modalities for pseudo labeling.</b> We report $AR_N@k$ achieved by the object proposal network trained on different modalities in Phase-I, where the benchmark is $VOC \rightarrow Non-VOC$ . Geometric cues (depth and normal) are stronger in discovering novel objects than the edge and pairwise affinity [209]. . . . .                                                                                                                                                                                                                                                                                                      | 33 |
| 3.4 | <b>Comparison of GOOD and two other strategies on VOC to non-VOC.</b> ShapeBias replaced the backbone of SelfTrain-RGB with a stylized ImageNet pre-trained backbone. ScaleAug applied multi-scale augmentation to the RGB input for collecting pseudo boxes at different scales. . . . .                                                                                                                                                                                                                                                                                                                                                                         | 34 |
| 3.5 | <b>Base class choices for studying the influence of the number of base classes.</b> . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 35 |
| 3.6 | <b>GOOD-Both vs. GOOD-All.</b> Adding RGB-based pseudo boxes (GOOD-All) to the geometric ensemble (GOOD-Both) does not consistently improve performance, particularly for medium and large objects. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 38 |

|     |                                                                                                                                                                                                                                                                                                                                                                                        |    |
|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 3.7 | <b>Architecture choice.</b> FCOS is a single-stage proposal-free object detector, and OLN is a two-stage proposal-based object detector (modified from Faster R-CNN). GOOD can significantly improve the open-world performance of both architectures. . . . .                                                                                                                         | 39 |
| 4.1 | <b>Examples of context names and generated candidate names</b> for three selected classes from ADE20K. Context names are key to comprehending general terms such as “field” and “box” and disambiguating polysemous terms like “cradle”, which, in this context, refers to a baby bed rather than a phone cradle or a mining tool. . . . .                                             | 44 |
| 4.2 | <b>Statistics of renovated datasets.</b> . . . .                                                                                                                                                                                                                                                                                                                                       | 50 |
| 4.3 | <b>Training with renovated names.</b> During inference, test names are merged from Original, OpenSeg, and RENOVATE names for fair comparison. Our results demonstrate that RENOVATE names can help train stronger open-vocabulary models. . . . .                                                                                                                                      | 52 |
| 4.4 | <b>Ablation on the context name sources.</b> . . . .                                                                                                                                                                                                                                                                                                                                   | 56 |
| 4.5 | <b>Ablation on renaming model architecture.</b> . . . .                                                                                                                                                                                                                                                                                                                                | 56 |
| 4.6 | <b>Ablation on GPT-4 prompts.</b> . . . .                                                                                                                                                                                                                                                                                                                                              | 56 |
| 4.7 | <b>Open-vocabulary segmentation on using both standard and open metrics across different pre-trained models using different name sets.</b> . .                                                                                                                                                                                                                                         | 57 |
| 4.8 | <b>Examples of context names, candidate names, and RENOVATE names for classes from ADE20K.</b> . . . .                                                                                                                                                                                                                                                                                 | 59 |
| 5.1 | <b>Comparison of different pseudo-GT choices.</b> . . . .                                                                                                                                                                                                                                                                                                                              | 70 |
| 5.2 | <b>Comparison of feature upsamplers across tasks of semantics segmentation, depth estimation, and normal estimation.</b> For each metric, we boldface the best performance and underline the second best. . . . .                                                                                                                                                                      | 72 |
| 5.3 | <b>Comparison of feature upsamplers across tasks of video object segmentation, zero-shot open-vocabulary segmentation, and interactive segmentation.</b> Following previous works [105, 80, 120, 97, 84, 196], we report J Mean, F Mean, and their average for video object segmentation, mIoU for open-vocabulary segmentation, and IoU@1 Click for interactive segmentation. . . . . | 72 |
| 5.4 | <b>Architecture comparison</b> on semantic segmentation and depth and normal estimation. All upsamplers are trained using LoftUp training objective. . . . .                                                                                                                                                                                                                           | 73 |
| 5.5 | <b>Ablation of architecture and training data size choices.</b> Upsamplers are trained using only Stage 1 loss for convenience. . . . .                                                                                                                                                                                                                                                | 75 |
| 5.6 | <b>LoftUp maintains strong performance with diverse input sizes and up-sampling scales.</b> . . . .                                                                                                                                                                                                                                                                                    | 77 |
| 5.7 | <b>Comparison of efficiency for different upsampler architectures.</b> Inference time is computed on a single A100 GPU. . . . .                                                                                                                                                                                                                                                        | 77 |
| 5.8 | <b>Feature upsampler comparison with CLIP-B/16 [157] backbone.</b> . . .                                                                                                                                                                                                                                                                                                               | 78 |
| 5.9 | <b>Feature upsampler comparison with RADIOv2.5-B [160] backbone.</b> .                                                                                                                                                                                                                                                                                                                 | 78 |

|     |                                                                                                                                                                                                                                                              |     |
|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 6.1 | Summary of failure discovery methods organised along the human–autonomous continuum. <i>Throughput</i> estimates relative speed of the discovery loop; <i>justificatory depth</i> indicates how richly warranted the resulting gap descriptions are. . . . . | 96  |
| 7.1 | <b>Summary of bridge biases.</b> Each bridging invariant introduces predictable failure modes inherited from its source. . . . .                                                                                                                             | 109 |



# 1 Introduction

## 1.1 Motivation: The Generalization Problem in Visual Perception

Deep learning has transformed visual perception, achieving remarkable performance on standardized benchmarks in object detection [164, 70, 163], semantic segmentation [123, 24, 170], and depth estimation [44, 62, 158]. Yet a persistent gap separates benchmark success from real-world reliability. When deployed in open-ended environments, models frequently encounter objects, concepts, and conditions that were absent from their training data. This discrepancy between the training distribution and the deployment environment—the challenge of Out-of-Distribution (OOD) generalization—remains one of the central obstacles to building trustworthy visual perception systems.

This gap has immediate practical consequences. An autonomous vehicle must detect objects it has never seen in training [194, 57, 16]. A medical imaging system must segment structures that differ in appearance from the annotated cases it learned from [20, 118, 83]. A robotic manipulation system must recognize and localize objects regardless of their visual novelty [233, 91]. In each of these settings, the inability to generalize beyond the training distribution can lead to silent, potentially catastrophic failures. Solving OOD generalization is therefore a prerequisite for deploying visual perception systems in the real world.

What makes this problem so persistent? At its core, the difficulty stems from a fundamental asymmetry: the training distribution is finite and curated, while the deployment world is open-ended and unpredictable. Models trained on closed datasets are rewarded for exploiting whatever statistical regularities—including superficial ones—best predict the training labels. This often results in *shortcut learning* [58]: models latch onto textures, backgrounds, or discriminative parts that happen to correlate with labels in the training set, rather than learning the deeper structural regularities that would generalize to novel conditions. The result is a model that appears competent within its training distribution but whose behavior is fundamentally under-determined outside it.

## 1.2 The Generalization Gap

The discrepancy between in-distribution (ID) and out-of-distribution (OOD) performance can be understood as a *generalization gap*. But “OOD” encompasses structurally different types of distributional shift, and understanding these differences is essential for addressing them.

This thesis engages with three types of generalization gap, each representative of a broader family of distributional shifts:

- (1) **Semantic shift:** The appearance of novel object categories unseen during training. A detector trained on a fixed set of classes must still localize objects from unknown categories in an open-world setting [90, 10].
- (2) **Label-space misalignment:** A mismatch in the *semantic interface* between model and benchmark, rather than in the visual input itself. When category names in segmentation datasets are imprecise or overly broad, the model’s text-based reasoning is misaligned with the actual visual content—even though the underlying vision-language representations may be adequate [235, 38].
- (3) **Resolution and granularity shift:** A discrepancy between the spatial resolution at which a model represents visual information and the resolution required by downstream tasks. Vision Foundation Models (VFMs) [14] produce highly semantic but spatially coarse features, creating a gap for tasks that require pixel-level precision [52].

What these gaps have in common is that they cannot be resolved by collecting more data from the training distribution or by improving optimization on the existing objective. The training loss constrains model behavior on in-distribution inputs, but it does not uniquely determine what the model does on inputs outside that distribution: many different functions can achieve the same training performance while behaving very differently under shift [35]. This *under-determination* is what makes OOD generalization fundamentally hard.

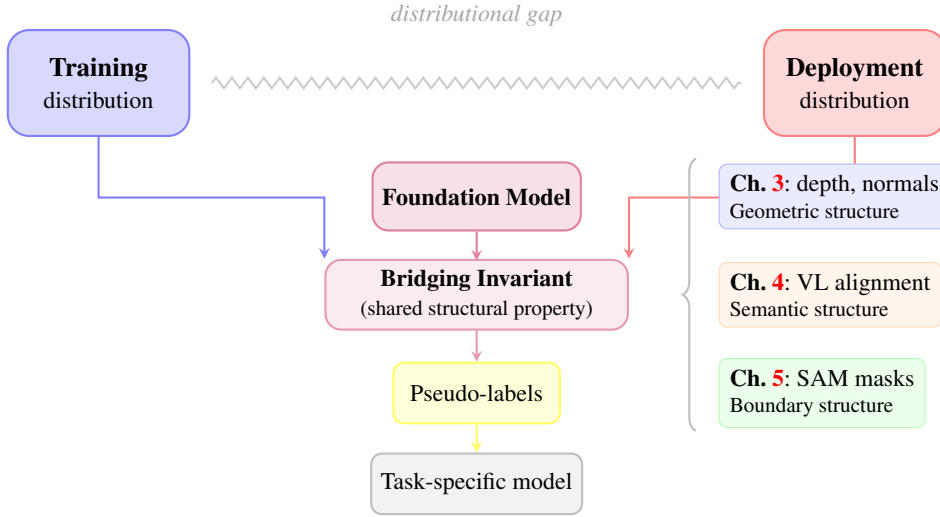
Yet generalization remains possible—provided something *beyond* the training objective guides the model’s behavior in novel conditions. The central question of this thesis is: what is that “something,” and how can we provide it?

### 1.3 Bridging Invariants: The Core Idea

This thesis proposes a unifying answer: OOD generalization becomes possible when we identify a *structural property that remains sufficiently stable across the distributional gap*—a property that, unlike raw input statistics, is shared between the training and deployment distributions—and design a mechanism to exploit it. We call such a property a **bridging invariant**.

Consider the gap between known and novel object categories. RGB appearance varies drastically across semantic classes: a sandwich looks nothing like a person. But three-dimensional structure—depth and surface normals estimated by monocular geometry models [42, 159]—is shared across all physical objects regardless of their semantic identity. A sandwich and a person both occupy space, cast shadows, and have coherent surfaces. Geometric structure is therefore a bridging invariant for the semantic shift gap: it holds on both sides and provides a basis for transferring detection capability from known to unknown categories.

The same logic applies to the other gaps in this thesis. For label-space misalignment, the bridging invariant is *vision-language alignment*: the shared structure between how humans perceive objects and how vision-language models such as CLIP [157] encode them.



**Figure 1.1: The bridging invariant pipeline.** A distributional gap separates training from deployment (top). A foundation model extracts a bridging invariant—a structural property shared across both distributions (middle). This invariant is converted into pseudo-labels that guide the task-specific model (bottom). The right column shows the three instantiations developed in this thesis. Figure 2.3 formalizes this framework with mathematical notation.

This alignment holds as long as objects are named according to common perceptual norms rather than arbitrary dataset conventions. For the resolution gap, the bridging invariant is *boundary and region structure*: object boundaries and region membership, as captured by class-agnostic segmentation models such as SAM [97], are scene-level properties, independent of feature resolution, and therefore hold whether features are computed at a coarse or fine scale. Figure 1.1 illustrates this framework schematically.

**Connection to theory.** The bridging invariant framework is a conceptual lens for understanding and designing methods—it guides the identification of useful invariants and organizes diverse approaches under a common vocabulary, without claiming formal guarantees or tight generalization bounds. More precisely, a bridging invariant can be understood as a mapping  $\phi : \mathcal{X} \rightarrow \mathcal{Z}$  into a signal space  $\mathcal{Z}$  where the training and test distributions are closer than in the original input space. Concretely, while  $P_{\text{train}}(X) \neq P_{\text{test}}(X)$ , the  $\mathcal{H}\Delta\mathcal{H}$ -divergence [12]—a measure of how well a classifier can distinguish between two distributions (see Section 2.1.2)—between the induced distributions  $P_{\text{train}}(\phi(X))$  and  $P_{\text{test}}(\phi(X))$  is small relative to the divergence between the original input distributions. The mapping  $\phi$  projects away the factors that shift across distributions while preserving the structure needed for the downstream task. This connects to the representation learning view of domain adaptation [12], where generalization depends on finding a representation in which source and target domains are close. Classical domain adaptation learns  $\phi$  end-to-end by minimizing a divergence measure between domains, typically requiring access to data from both distributions during training. This thesis takes a different approach: we *identify* what

**Table 1.1: Overview of the three bridging invariants developed in this thesis. Each row summarizes how a specific generalization gap is addressed by identifying an invariant, extracting it from a foundation model, and transferring it via pseudo-labeling.**

|                           | GOOD (Ch. 3)                         | RENOVATE (Ch. 4)                           | LoftUp (Ch. 5)                         |
|---------------------------|--------------------------------------|--------------------------------------------|----------------------------------------|
| <b>Gap type</b>           | Semantic shift (novel categories)    | Label-space misalignment (imprecise names) | Resolution shift (coarse VFM features) |
| <b>Bridging invariant</b> | Geometric structure (depth, normals) | Vision-language alignment                  | Boundary & region structure            |
| <b>Invariant source</b>   | Omnidata / MiDaS                     | CLIP + CaSED                               | SAM / SAM2                             |
| <b>Transfer mechanism</b> | Pseudo bounding boxes                | Pseudo category names                      | Pseudo HR features                     |
| <b>Key result</b>         | +5.0% AR@100 (24% rel.)              | +15% rel. mIoU; 5× data eff.               | 10–20% improvement across 6 tasks      |

the right  $\phi$  is for a given gap—what structural property is actually shared—and then *realize* it using an external source. The intellectual contribution is in recognizing which invariant matters for a given distributional shift, with the optimization procedure that exploits it being secondary.

The framework also makes the failure modes of a bridging invariant explicit:

- (i) The structural property assumed to be shared actually shifts between training and deployment—for example, geometric structure is uninformative for objects defined by texture rather than shape.
- (ii) The foundation model that provides the invariant introduces its own distortions—monocular depth estimates degrade on unusual materials, CLIP embeddings reflect internet-centric biases.
- (iii) The invariant is task-relevant for only a subset of the test distribution—boundary structure aids segmentation but may harm tasks requiring smooth spatial variation.

Each of these failure modes is observed empirically across the technical chapters and discussed in the Conclusion (Chapter 7), where we analyze how the biases of each bridge relate to its invariant source.

**Foundation Models** (FMs) [14] are remarkably effective sources of bridging invariants. Pre-trained on diverse, internet-scale data, foundation models internalize structural regularities—geometric, semantic, spatial—that hold across a far broader range of conditions than any task-specific training set covers. This thesis leverages foundation models as off-the-shelf tools for accessing bridging invariants: monocular depth and normal estimators [42, 159] provide geometric structure, vision-language models [157] provide semantic alignment, and segmentation models like SAM [97] provide boundary structure. However, foundation models are *sources* of bridging invariants; the conceptual contribution is in identifying *which* invariant bridges a given gap and designing a mechanism—typically a pseudo-labeling pipeline—to transfer it into the task-specific model (Table 1.1). Foundation models thus play a dual role in this thesis: they supply the raw invariant, and their broad training coverage explains why that invariant holds across a wide range of deployment conditions. The limits of this coverage—and the biases it introduces—are analyzed in Chapter 7.



## 1.4 Contributions

This thesis presents three works, each addressing a specific generalization gap by identifying and exploiting a bridging invariant.

### 1.4.1 The Geometric Bridge: Open-World Object Detection (GOOD)

**The gap:** RGB-based object detectors overfit to the appearance of training classes and fail to detect novel-looking objects in open-world settings. The more visually dissimilar a novel category is from the training classes, the more severe the failure.

**The bridging invariant:** Geometric structure—depth and surface normals—is shared across all physical objects regardless of semantic class. Objects that look nothing alike in RGB (e.g., food items vs. vehicles) share common geometric properties: spatial extent, surface orientation, and closure.

**The method:** We train an object proposal network on geometric cues extracted by off-the-shelf monocular estimators. This network discovers unannotated novel objects in the training set, which are then used as pseudo-labels to train an RGB-based open-world detector. The resulting Geometry-guided Open-world Object Detector (GOOD) combines the complementary strengths of geometric and appearance cues.

**Key result:** Training with only a single “person” class on the COCO dataset, GOOD surpasses state-of-the-art methods by 5.0% AR@100 on non-person classes (a 24% relative improvement).

### 1.4.2 The Semantic Bridge: Renovating Benchmark Names (RENOVATE)

**The gap:** Open-vocabulary segmentation models generalize to unseen categories by using class names as text prompts. However, existing benchmark names are often imprecise, overly general, or contextually misaligned with the actual visual segments, degrading both training and evaluation.

**The bridging invariant:** Vision-language alignment—the shared structure between human visual perception and the representations learned by vision-language models—which holds when objects are named according to common perceptual norms rather than arbitrary dataset conventions.

**The method:** RENOVATE uses image captioning models to generate contextually relevant candidate names, then trains a specialized renaming model to select the best-matching name for each ground-truth segmentation mask. This automated pipeline produces segment-level names that align more closely with the vision-language model’s semantic space.

**Key result:** Training with renovated names yields up to 15% relative improvement in open-vocabulary segmentation, and models trained on  $10^4$  images with renovated names match the performance of models trained on  $5 \times 10^4$  images with original names.

### 1.4.3 The Spatial Bridge: Feature Upsampling (LoftUp)

**The gap:** Vision Foundation Models produce highly semantic but spatially coarse feature maps (typically  $16\times$  downsampled), limiting their utility for dense prediction tasks that require pixel-level precision. Increasing input resolution is computationally prohibitive; task-specific decoders require extensive retraining.

**The bridging invariant:** Boundary and region structure—object boundaries and region membership are resolution-independent properties of the scene that hold whether features are computed at coarse or fine scale.

**The method:** LoftUp constructs pseudo-groundtruth high-resolution features using class-agnostic masks from SAM, then trains a coordinate-based cross-attention transformer that maps image coordinates and RGB values to high-resolution features via cross-attention with the low-resolution feature map.

**Key result:** With less than 20% additional parameters, LoftUp outperforms existing feature upsamplers by 10–20% on most downstream tasks, and achieves over 35% improvement on video object segmentation.

### 1.4.4 The Upstream Problem: Gap Discovery (Discussion)

**The gap:** The three technical chapters each assumed that a human researcher had already identified the distributional shift to be bridged. Gap discovery—systematically identifying where a model’s training distribution diverges from deployment conditions—remains the least systematized stage of the OOD generalization pipeline.

**The contribution:** Chapter 6 examines gap discovery as the remaining piece needed to complete the bridging invariant framework. It surveys failure-discovery methodology from adjacent communities, analyzes how the OOD field currently identifies gaps, and tests the practical potential through a human–AI collaborative case study that applies automated diagnostic tools to the naming misalignment that motivated RENOVATE.

**Key insight:** Human–AI collaboration emerges as the practical sweet spot: automated tools can rapidly flag candidate gaps, but only within a search space that a human researcher has specified. The chapter identifies four open problems, from the limits of automated discovery to the question of completeness—when has a model been tested against enough distributional shifts?

## 1.5 Thesis Organization

The remainder of this thesis is structured as follows. Chapters 3–5 are largely self-contained but share a common framework established in Chapter 2; each builds on the previous by addressing a different type of gap with a different invariant, progressing from geometric to semantic to spatial structure. Chapter 6 then reflects on the upstream question that all three technical chapters left implicit, and the Conclusion synthesizes the cross-cutting themes.

**Chapter 2** provides background on the theory and practice of OOD generalization, the role of invariance in representation learning, and foundation models as sources of bridging

invariants.

**Chapter 3** presents GOOD, demonstrating how geometric invariants can bridge the gap between known and novel object categories in open-world detection.

**Chapter 4** presents RENOVATE, showing that aligning the semantic interface—category names—with the vision-language model’s representational structure unlocks generalization that was latent but inaccessible.

**Chapter 5** presents LoftUp, introducing a coordinate-based feature upsampler that exploits boundary structure to bridge the resolution gap between foundation model features and dense prediction tasks.

**Chapter 6** steps back from constructing bridges to examine the epistemic challenge of gap discovery: how do we identify distributional shifts in the first place, and can this process be systematized?

**Chapter 7** concludes with a summary of contributions, a discussion of the biases that bridging invariants introduce, and final reflections on the future of OOD generalization.

## 1.6 Publications

The technical contributions of this thesis are based on the following publications:

- (1) **Haiwen Huang**, Andreas Geiger, and Dan Zhang. “GOOD: Exploring Geometric Cues for Detecting Objects in an Open World.” In *International Conference on Learning Representations (ICLR)*, 2023.  
→ Chapter 3.
- (2) **Haiwen Huang**, Songyou Peng, Dan Zhang, and Andreas Geiger. “Renovating Names in Open-Vocabulary Segmentation Benchmarks.” In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.  
→ Chapter 4.
- (3) **Haiwen Huang**, Anpei Chen, Volodymyr Havrylov, Andreas Geiger, and Dan Zhang. “LoftUp: Learning a Coordinate-Based Feature Upsampler for Vision Foundation Models.” In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.  
**Oral presentation.**  
→ Chapter 5.

In all three primary publications, the author of this thesis was the lead researcher, responsible for conceptualization, implementation, experimentation, and writing.

The following publications were produced during the doctoral period and are related to but not directly included in this thesis:

- (4) Volodymyr Havrylov, **Haiwen Huang**, Dan Zhang, and Andreas Geiger. “Benchmarking Feature Upsampling Methods for Vision Foundation Models using Interactive Segmentation.” In *ICCV Workshop on Interactive Learning with Implicit Human Feedback (ILR+G)*, 2025. **Oral presentation.**

- (5) **Haiwen Huang**, Dan Zhang, and Andreas Geiger. “Multimodal Dataset Upgrading: A New Challenge for Data Annotation.” In *ICLR Workshop on Navigating and Addressing Data Problems for Foundation Models*, 2024.

## 2 Background

The bridging invariant framework introduced in Section 1.3 rests on three pillars: a theory of when generalization under shift is possible, a mechanism for providing invariants externally, and a class of models rich enough to carry those invariants. This chapter develops each pillar in turn.

We begin with OOD generalization (Section 2.1): we first establish a taxonomy of distribution shifts (Section 2.1.1), then review classical generalization bounds under shift (Section 2.1.2), survey invariance-based methods such as IRM and their empirical limitations (Section 2.1.3), and discuss shortcut learning as the complementary explanation for why external invariants are needed (Section 2.1.4). We then discuss invariance and representation learning as the theoretical grounding for bridging invariants, and review pseudo-labeling as the shared mechanism through which all three technical chapters realize their bridges (Section 2.2). Next, we introduce foundation models and explain why they serve as effective sources of bridging invariants (Section 2.3). Finally, we preview how the subsequent chapters instantiate these ideas (Section 2.4).

### 2.1 Out-of-Distribution Generalization

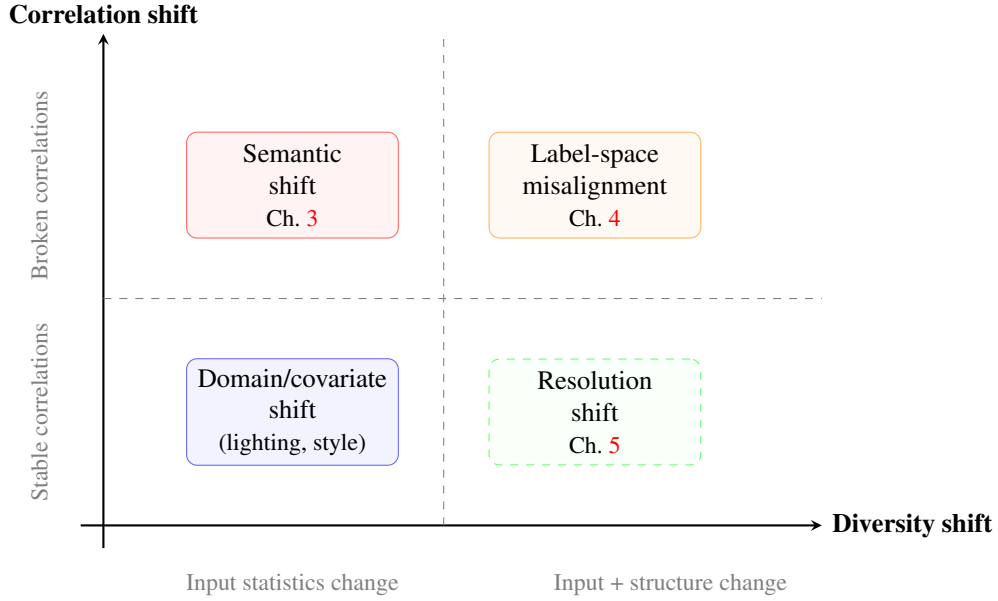
#### 2.1.1 Problem Settings: A Taxonomy of Distribution Shift

The term “out-of-distribution” encompasses heterogeneous phenomena. Prior surveys have organized these along different axes: Shen et al. [184] categorize OOD generalization methods into unsupervised representation learning, supervised model learning, and optimization strategies; Yang et al. [223] unify anomaly detection, novelty detection, open-set recognition, and OOD detection under a single framework of “generalized OOD detection.”

These taxonomies organize *methods*. A complementary question is how to organize the *shifts themselves*—a principled characterization of what changes between training and deployment is necessary before one can reason about which invariants might bridge the gap.

**Principled framework: OoD-Bench’s two-dimensional view.** Ye et al. [228] propose a principled characterization of distribution shift along two fundamental dimensions, which we adopt as the conceptual foundation for understanding OOD phenomena:

- (1) **Diversity shift:** Changes in the diversity or distribution of inputs—variations in style, background, viewpoint, or lighting—that increase the complexity of the input space without fundamentally altering the correlation structure between features and labels. The model encounters more visual variation, but the essential feature-label relationships remain stable.



**Figure 2.1: A taxonomy of distribution shift.** Following the two-dimensional framework of Ye et al. [228], distribution shifts can be characterized along diversity shift (changes in input statistics) and correlation shift (changes in feature-label relationships). The four thesis-specific gaps are placed within this framework: domain/covariate shift involves diversity changes with stable correlations; semantic shift (Ch. 3) and label-space misalignment (Ch. 4) involve correlation shifts; resolution shift (Ch. 5) is a structural mismatch that sits outside the traditional diversity/correlation dichotomy.

- (2) **Correlation shift:** Changes in the spurious or semantic correlations between features and labels. A texture or background that reliably predicts class membership in the training set may no longer do so at test time. These shifts break the assumptions the model made during training about which features matter for prediction.

Mathematically, diversity shift corresponds to a change in the marginal input distribution  $P_{\text{train}}(X) \neq P_{\text{test}}(X)$  while the conditional  $P(Y | X)$  remains approximately stable; correlation shift corresponds to a change in the joint  $P_{\text{train}}(X, Y) \neq P_{\text{test}}(X, Y)$  where the conditional  $P(Y | X)$  itself changes, typically because spurious features that were predictive during training lose their association with labels at test time.

This two-dimensional framework provides a principled way to understand diverse phenomena under a unified structure. Most “standard” domain adaptation scenarios (lighting, weather, style variations) fall under diversity shift, where the feature-label correlations hold across domains even as the input distribution changes. In contrast, open-vocabulary and semantic shift scenarios involve correlation shift: novel categories have different visual-semantic relationships than training categories, or category names may not align with the vision-language model’s learned associations.

**Thesis-specific gaps within the framework.** For the purposes of this thesis, we further refine this framework by organizing phenomena according to *type of shift*—what changes between training and deployment—because this directly maps to the structure of the subsequent chapters:

- (1) **Domain and covariate shift (Diversity Shift):** changes in low-level input statistics—sensor characteristics, illumination, weather, style, or image corruptions—while the label distribution remains fixed [128, 54, 74, 208, 59]. These typically represent diversity shifts within an otherwise stable feature-label correlation structure.
- (2) **Semantic shift (Correlation Shift):** The appearance of novel object categories absent from training, changes in the label set, or mismatches in category granularity between training and deployment [10, 90, 209, 85]. This represents a fundamental change in the feature-label correlations available for learning (Chapter 3).
- (3) **Label-space misalignment (Correlation Shift):** Ambiguity, imprecision, or incompleteness in the semantic interface itself—category names, synonyms, or annotation granularity—that causes misalignment between model predictions and evaluation criteria [235, 65, 38, 80, 221]. This represents a shift in how the semantic space is operationalized, breaking the assumption that category names in the benchmark align with the model’s learned associations (Chapter 4).
- (4) **Resolution and granularity shift:** Discrepancies between the spatial resolution at which a model produces representations and the resolution required by downstream dense prediction tasks [126, 148, 100, 127]. This is a structural mismatch between the representational capacity and task requirements (Chapter 5).

It is instructive to map the thesis-specific gaps to the two-dimensional framework (Fig. 2.1). The semantic shift addressed in Chapter 3 represents a correlation shift: the feature-label relationships for novel categories differ fundamentally from those of training categories, and the model’s learned associations break for objects outside the training taxonomy. The label-space misalignment of Chapter 4 is also a correlation shift, but at the *interface* level rather than the input level—the visual content is well-represented by the model, but the names used to query it do not match its learned associations. We place it at moderate-to-high diversity in Fig. 2.1 because RENOVATE operates across diverse benchmarks (ADE20K, COCO, Cityscapes) with substantially different visual domains. The diversity shift is present at the benchmark level even though the VLM’s internal representation handles it. The resolution shift of Chapter 5 does not fit neatly into the diversity/correlation dichotomy: it is a structural mismatch between the model’s representational capacity and the spatial resolution required by downstream tasks, distinct from shifts in the input distribution. We place it in Fig. 2.1 with a dashed border to indicate this. Although it shares some characteristics with diversity shift (the representation space changes), its root cause is architectural rather than distributional. This illustrates that the OOD landscape is broader than any single taxonomy captures, motivating the type-based organization we adopt.

Many benchmarks operationalize OOD by predefining a train-test split along one of these axes. This is useful experimentally but can obscure an important point: in real-world visual perception, OOD often emerges because the system’s interface with the world—what it is asked to recognize, how it represents inputs, or how finely it must operate—is itself insufficiently aligned with deployment conditions.

### 2.1.2 Classical Theory: Bounding Generalization Under Shift

The theoretical study of learning under distribution shift has a rich history. The foundational framework of Ben-David et al. [12] provides generalization bounds for domain adaptation that are directly relevant to the bridging invariant perspective of this thesis.

Consider a source distribution  $P_S$  (analogous to the training distribution) and a target distribution  $P_T$  (analogous to the deployment distribution) over the same input space  $\mathcal{X}$ . Ben-David et al. showed that the target error of a hypothesis  $h$  can be bounded as:

$$\varepsilon_T(h) \leq \varepsilon_S(h) + d_{\mathcal{H}\Delta\mathcal{H}}(P_S, P_T) + \lambda^*, \quad (2.1)$$

where  $\varepsilon_S(h)$  is the source error,  $d_{\mathcal{H}\Delta\mathcal{H}}(P_S, P_T)$  is the  $\mathcal{H}\Delta\mathcal{H}$ -divergence measuring the discrepancy between domains as seen through the hypothesis class  $\mathcal{H}$ , and  $\lambda^*$  is the error of the ideal joint hypothesis (a term reflecting the intrinsic difficulty of the problem).

The key insight from this bound is that generalization under shift depends on finding a *representation* in which the divergence  $d_{\mathcal{H}\Delta\mathcal{H}}$  is small while the source error remains low. This is precisely the idea behind bridging invariants: the mapping  $\phi$  from Section 1.3 projects inputs into a space where the distributions are closer, enabling transfer. The classical bound provides the theoretical justification for why identifying such a representation matters.

Subsequent work extended this framework in several directions. PAC-Bayes bounds [133, 11] provided tighter, data-dependent guarantees. Distributional robustness [174, 40] framed the problem as optimization under worst-case distributional perturbations. Causal approaches [152, 182] argued that invariances should be grounded in causal structure rather than statistical association. Each of these perspectives reinforces the same core principle: generalization requires identifying what remains stable under shift.

### 2.1.3 Invariance-Based Approaches to OOD Generalization

The theoretical insight—that generalization requires finding representations where distributions are close—motivated a wave of practical methods for learning invariant representations.

**Invariant Risk Minimization.** Arjovsky et al. [7] proposed Invariant Risk Minimization (IRM), arguing that a representation is robust if the optimal classifier on top of it is the same across all training environments. IRM formalizes this as a bi-level optimization problem: learn a representation  $\Phi$  such that the optimal linear classifier  $w^*$  on  $\Phi$  is simultaneously optimal for every environment  $e$ . This principle—that a good representation should support



a stable predictor across environments—is a theoretical precursor to the bridging invariant idea. However, IRM requires access to data from multiple environments during training, which is often unavailable in practice.

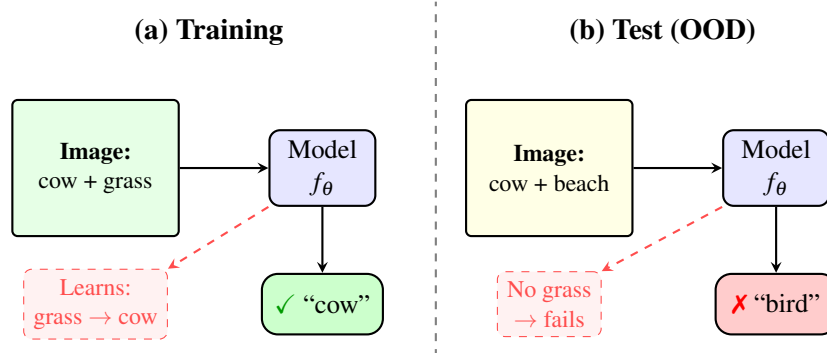
The relationship between IRM’s invariance and the bridging invariants of this thesis deserves explicit attention. IRM defines invariance relative to a *classifier*: a representation is invariant if the optimal linear predictor on top of it is the same across all training environments. In contrast, bridging invariants are defined relative to a *representation*: the mapping  $\phi$  should produce distributions that share task-relevant structure across training and test conditions. These notions are related—both seek stability across distributional shifts—but differ in important ways. IRM learns invariance end-to-end from data with multiple environment labels. Our approach identifies the invariant from domain knowledge and injects it via an external source. IRM’s invariance is optimized jointly with the task. Ours is provided independently by a foundation model. The DomainBed results of Gulrajani and Lopez-Paz [66], showing that IRM-style methods do not consistently outperform ERM, motivate the approach taken in this thesis: if learning invariance end-to-end is unreliable, perhaps *providing* it externally—from a source that has already internalized the relevant structural regularity—is a more practical path.

Subsequent variants addressed different limitations. V-REx [103] penalized variance of risks across environments. IB-IRM [1] combined information bottleneck constraints with invariance objectives. Fish [186] used inter-domain gradient alignment. Despite theoretical elegance, the empirical gains of these methods proved inconsistent. Gulrajani and Lopez-Paz [66] showed in a large-scale study (DomainBed) that no invariance-based method consistently outperformed well-tuned Empirical Risk Minimization (ERM) across diverse benchmarks—a sobering finding that highlighted the gap between theory and practice.

**Domain generalization.** A broader family of methods attempts to train models that generalize to unseen domains without adaptation at test time. These range from data augmentation strategies [207, 245] and meta-learning [109, 8] to distribution alignment [139, 55]. Benchmarks such as DomainBed [66] and WILDS [98] provide standardized evaluation. A common finding across these benchmarks is that the effectiveness of any invariance strategy depends critically on whether the assumed invariance actually holds—when it does not, enforcing the wrong invariance can harm performance.

### Implications for this thesis

The invariance-based literature reveals a tension: the *principle* of finding invariant representations is sound, but *learning* these representations end-to-end from data is difficult, unstable, and dependent on having access to the right environment diversity during training. This thesis takes a different approach. We *identify* specific structural invariants that are known to hold for a given distributional gap—geometric structure for semantic shift, vision-language alignment for label-space misalignment, boundary structure for resolution shift—and inject them into the learning process using external sources, sidestepping the optimization difficulties of end-to-end invariance learning while preserving the theoretical motivation.



**Figure 2.2: Shortcut learning as a cause of OOD failure.** (a) During training, the model exploits a spurious correlation between background context (grass) and the label (cow), since this is the simplest predictor that minimizes training loss. (b) At test time, when the cow appears against a novel background (beach), the shortcut fails and the model misclassifies the object. Bridging invariants address this by providing features—such as geometric structure—that are robust to such contextual shifts.

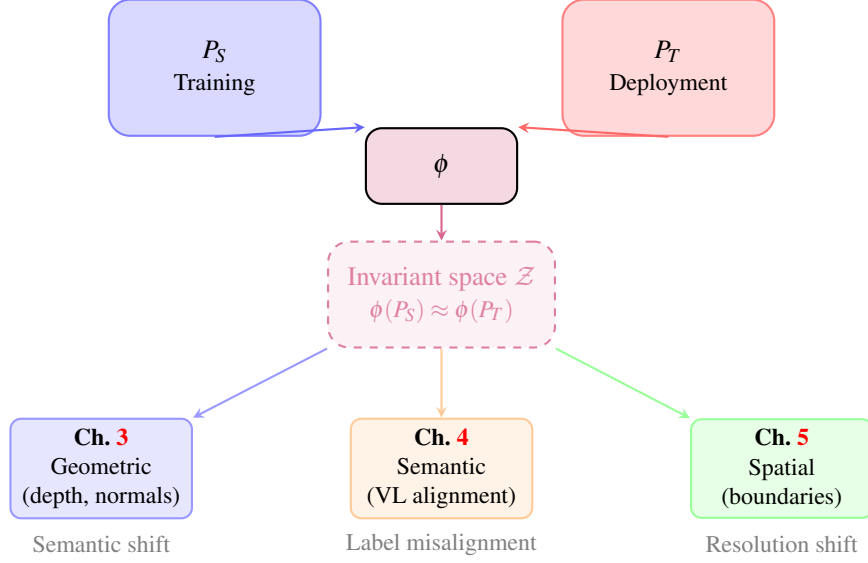
### 2.1.4 Shortcut Learning and Under-Determination

A complementary perspective on why OOD generalization fails comes from the shortcut learning literature. Geirhos et al. [59] demonstrated that convolutional neural networks trained on ImageNet are heavily biased toward texture rather than shape—a bias that explains many generalization failures, since texture statistics shift more readily across domains than object shape. Geirhos et al. [58] generalized this observation, arguing that shortcut learning is the default outcome of standard training: models exploit the simplest available correlations that predict training labels, regardless of whether these correlations hold under distribution shift (Fig. 2.2).

The shortcut learning perspective is important for this thesis because it explains *why* external bridging invariants are needed. If models reliably learned the “right” features during standard training, OOD generalization would be solved by scaling up data and compute. The persistence of shortcut learning even in large-scale models [58] suggests that the training objective alone is insufficient to determine robust behavior—the model’s representation is *under-determined* with respect to OOD generalization, and something beyond the training loss must constrain it. Bridging invariants provide exactly this additional constraint.

## 2.2 Invariance and Representation Learning

The previous section established that generalization under shift requires finding representations where distributional divergence is small. This section develops that idea in two parts: we first examine the role of intermediate representations as bridges between distributions (Section 2.2.1), and then show how pseudo-labeling serves as the shared mechanism through which all three technical chapters transfer their bridging invariants into the training pipeline (Section 2.2.2).



**Figure 2.3: Formal view of bridging invariants.** This figure formalizes the pipeline introduced schematically in Fig. 1.1. A mapping  $\phi$  projects both the source distribution  $P_S$  and target distribution  $P_T$  independently into an invariant space  $\mathcal{Z}$  where distributional divergence is small. This thesis instantiates this framework through three distinct invariants: geometric structure for open-world detection (Ch. 3), vision-language alignment for open-vocabulary segmentation (Ch. 4), and boundary structure for feature upsampling (Ch. 5).

### 2.2.1 Representations as Bridges

The role of intermediate representations in enabling generalization has been a central theme in machine learning since the early days of feature engineering. The transition from hand-crafted features (SIFT [125], HOG [36]) to learned representations (CNNs [102], Transformers [204]) changed how the answer is obtained, but the question of *what* a good representation should capture persists.

From the perspective of this thesis, a good representation for OOD generalization is one that retains task-relevant information while discarding the factors that vary across distributions. This connects to several theoretical frameworks. The information bottleneck principle [201, 187] formalizes this as a trade-off between compression and prediction: a representation should compress the input as much as possible while preserving information about the target. Disentanglement [13, 73] seeks representations where different factors of variation are encoded in separate dimensions, making it possible to change one factor (e.g., lighting) without affecting others (e.g., object identity). These principles are complementary: Huang, Amersfoort, and Gal [79] showed that decomposing learned representations along information-theoretic and disentanglement principles can improve uncertainty estimation under distribution shift, connecting representation structure directly to OOD robustness.

The bridging invariant formalization from Section 1.3 can be understood through this

lens (Fig. 2.3). The mapping  $\phi : \mathcal{X} \rightarrow \mathcal{Z}$  acts as a representation that discards the shifting factors (appearance for GOOD, naming conventions for RENOVATE, spatial resolution for LoftUp) while preserving the task-relevant structure (object presence, semantic identity, dense features). The difference from classical representation learning is that we do not learn  $\phi$  from the downstream task—we obtain it from an external source (a foundation model) that has already internalized the relevant invariance through large-scale pre-training.

### 2.2.2 Pseudo-Labeling as Invariant Transfer

A practical observation unifies the three technical chapters of this thesis: all three exploit their bridging invariant through some form of *pseudo-labeling*. GOOD generates pseudo bounding boxes from geometry-trained proposal networks. RENOVATE generates pseudo category names from vision-language models. LoftUp generates pseudo high-resolution features from SAM-guided self-distillation. In each case, the bridging invariant is realized as pseudo-groundtruth that augments the training signal.

Pseudo-labeling and self-training have a long history in semi-supervised and unsupervised learning. Lee [106] introduced pseudo-labels for semi-supervised classification, assigning the model’s own high-confidence predictions as training targets. Xie et al. [216] scaled this to Noisy Student Training, demonstrating that iterative pseudo-labeling with noise injection improves ImageNet accuracy. FixMatch [192] combined consistency regularization with pseudo-labeling, using strong augmentation to prevent collapse. More recent work has shown that pseudo-labeling benefits from curriculum learning [3], confidence estimation [169], and careful threshold selection [211] to mitigate the accumulation of errors from mislabeled data. These advances establish pseudo-labeling as a mature mechanism for leveraging unlabeled data, but they all assume the pseudo-labels originate from the model’s own predictions—an assumption that this thesis relaxes.

In the context of this thesis, pseudo-labeling serves a different purpose than in classical semi-supervised learning. In classical self-training, the model teaches itself from its own predictions on unlabeled data. Here, the pseudo-labels are generated by an *external source*: a foundation model that provides access to a bridging invariant. GOOD’s pseudo boxes come from geometry, not from RGB appearance. RENOVATE’s pseudo names come from vision-language alignment, not from dataset conventions. LoftUp’s pseudo features come from boundary-aware refinement, not from the VFM’s own upsampling. We call this mechanism *invariant transfer*: the external source injects a structural property into the training pipeline that lies outside the downstream model’s training signal.

This external sourcing partially mitigates the well-known problem of *confirmation bias* in pseudo-labeling [3], where errors in self-generated labels are reinforced during retraining. Because the error modes of the foundation model differ from those of the downstream model, the pseudo-labels provide complementary information rather than reinforcing existing mistakes. However, this does not eliminate noise entirely: when the foundation model itself fails—on out-of-domain inputs, unusual materials, or underrepresented visual concepts—the pseudo-labels degrade, and with them the bridge. This motivates the two-stage self-distillation strategy in LoftUp (Chapter 5), where iterative refinement progressively reduces noise in the pseudo-groundtruth. More broadly, the quality of the pseudo-

**Table 2.1: Foundation model families used in this thesis.** Each family provides access to a different bridging invariant through its pre-training objective and learned representations.

| Family                 | Representative Models      | Training Signal             | Key Property              | Thesis Chapter |
|------------------------|----------------------------|-----------------------------|---------------------------|----------------|
| Self-supervised vision | DINOv2 [145], MAE [68]     | Self-distillation / masking | Dense semantic features   | Ch. 5          |
| Vision-language        | CLIP [157], SigLIP [236]   | Image-text contrastive      | Text-image alignment      | Ch. 4, 5       |
| Segmentation           | SAM [97], SAM 2 [162]      | Prompted segmentation       | Class-agnostic boundaries | Ch. 5          |
| Geometry estimation    | Omnidata [42], MiDaS [159] | Supervised regression       | Depth, surface normals    | Ch. 3          |

labels depends on the quality of the external source—a theme we return to when discussing the biases of bridges in the conclusion (Chapter 7).

## 2.3 Foundation Models as Invariant Sources

Bridging invariants require both a structural property and a mechanism for injecting it. Where do we obtain these invariants in practice? This section introduces foundation models as the answer. We describe the relevant model families (Section 2.3.1), explain why their training regime produces useful invariants (Section 2.3.2), and discuss the limitations of relying on them (Section 2.3.3).

### 2.3.1 What Are Foundation Models?

Foundation models are large-scale models pre-trained on broad, diverse data that can be adapted to a wide range of downstream tasks [14]. Their distinguishing property is generality: rather than being trained for a specific application, they learn representations that transfer across tasks, domains, and even modalities.

The success of foundation models can be attributed to three factors: scale (billions of parameters trained on billions of data points), diversity (training data spanning a wide range of visual and linguistic conditions), and self-supervised or weakly-supervised objectives that do not depend on task-specific annotations.

This thesis engages with four families of foundation models, each providing access to a different bridging invariant (Table 2.1). We briefly introduce the key architectural ideas relevant to understanding the subsequent chapters.

**Vision Transformers.** The Vision Transformer (ViT) [39] divides an input image into fixed-size patches (typically  $16 \times 16$  pixels), linearly embeds each patch, and processes the resulting sequence with a standard Transformer encoder [204], augmented with positional embeddings that encode spatial location. The key consequence for this thesis is the resolution bottleneck: the patch embedding reduces spatial resolution by a factor of 16, producing features that are highly semantic but spatially coarse. This creates the resolution gap that Chapter 5 addresses.

Self-supervised pre-training of ViTs has produced a family of powerful vision foundation models. DINO [19] pioneered self-distillation for vision transformers, learning features that emerge without clustering-based losses. DINOv2 [145] extended this with im-

proved scaling, learning representations that capture both semantic and geometric structure. DINOv3 [188] further advances self-supervised vision learning with improved training recipes and stronger representations. Masked Autoencoder (MAE) [68] learns by reconstructing masked image patches, while BEiT [9] combines masked image modeling with vector-quantized tokens. EVA [49] achieves strong performance through improved training and architectural innovations. SigLIP [236] uses sigmoid-based contrastive learning between image and text, and SigLIP-2 [202] extends this with further optimizations. These models internalize structural regularities from diverse training data, making them natural sources of bridging invariants. Of these, this thesis directly uses DINOv2 as the primary backbone in the LoftUp pipeline (Ch. 5).

**Contrastive vision-language models.** CLIP [157] pioneered large-scale vision-language pre-training, learning joint representations of images and text by contrastive learning on hundreds of millions of image-text pairs from the internet. The resulting embedding space aligns visual appearance with linguistic description, enabling zero-shot transfer: a model can classify images into categories it has never been explicitly trained on, simply by comparing image embeddings to text embeddings of category names.

Subsequent work has refined and extended CLIP’s approach. EVA-CLIP [195] incorporates architectural improvements to achieve stronger vision-language alignments. OpenCLIP [82] provides open-source implementations and stronger baselines across various architectures.

For this thesis, the key property of CLIP-like models is that they provide a *semantic bridge*: the alignment between vision and language is learned from diverse web data and therefore reflects common perceptual norms rather than the conventions of any single benchmark. This is the invariant that Chapter 4 exploits—when category names are aligned with these norms, the vision-language alignment enables generalization.

**Promptable segmentation models.** The Segment Anything Model (SAM) [97] was trained on over one billion masks to segment any object in any image, given a prompt (point, box, or mask). SAM 2 [162] extends this to video by incorporating temporal consistency and recurrent connections, enabling prompt-based instance segmentation across video frames. SAM 3 [18] further advances the family by integrating semantic concepts into the segmentation pipeline, enabling concept-aware prompting. The key property is *class-agnostic boundary detection*: SAM and its successors produce high-quality object boundaries regardless of object category, because their training data spans an enormous diversity of visual concepts.

For this thesis, SAM provides access to a *spatial bridge*: object boundaries are resolution-independent properties of the scene, and SAM’s class-agnostic masks can serve as pseudo-groundtruth for training feature upsamplers that respect boundary structure at any resolution. This is the invariant that Chapter 5 exploits.

**Monocular geometry estimators.** Models such as Omnidata [42] and MiDaS [159] estimate depth and surface normals from a single RGB image. Trained on diverse real and

synthetic 3D scenes, these estimators generalize well across domains—their predictions are accurate even on images very different from their training data. This robustness arises because geometric structure (relative depth, surface orientation) is a physical property that transcends appearance. For this thesis, monocular estimators provide access to a *geometric bridge*: the invariant that Chapter 3 exploits.

### 2.3.2 Why Foundation Models Carry Bridging Invariants

Why should foundation models provide useful bridging invariants? The key is the relationship between training diversity and implicit invariance.

A model trained on internet-scale data encounters an enormous range of visual conditions: different objects, viewpoints, lighting, styles, and scenes. To achieve low training loss across this diversity, the model must learn representations that capture regularities *across* conditions rather than memorizing condition-specific patterns. From an information-theoretic perspective [201], diverse training data forces the model to compress away irrelevant variation—the factors that differ across conditions contribute noise rather than signal—leaving representations that emphasize the invariant structure shared across contexts. Geometric structure is such a regularity—it governs how objects appear regardless of their identity. Semantic alignment between images and text is another—it reflects how humans describe the visual world across contexts. Boundary structure is a third—it organizes visual scenes at every scale. Empirically, foundation models trained with different objectives (contrastive, masked prediction, generative) converge on surprisingly similar internal representations [81]. This convergence suggests that training diversity drives models toward a shared set of structural regularities regardless of the specific learning signal.

However, this implicit coverage has limits. Foundation models learn whatever regularities their training data contains, but they cannot learn regularities absent from that data. Novel physical phenomena, rare cultural contexts, or combinatorial interactions of individually-seen factors may fall outside the implicit boundary [14, 199]. The invariants that foundation models carry are therefore *empirically robust* across a broad range of conditions, but not *guaranteed* to hold universally. This has consequences for the reliability of bridges built on them—a theme we return to in the conclusion (Chapter 7).

### 2.3.3 Limitations: What Foundation Models Cannot Provide

Foundation models are powerful sources of bridging invariants, but they do not solve the OOD generalization problem by themselves. Three limitations are relevant:

First, foundation models do not tell us *which* invariant matters for a given gap. A monocular depth estimator carries geometric structure, but it takes human insight to recognize that geometric structure is the right bridging invariant for open-world object detection. The identification of the invariant—the conceptual step—remains a human contribution.

Second, foundation models are not trained for specific downstream tasks. They learn general-purpose representations, but adapting these to a concrete OOD task requires non-trivial method design. For example, CLIP features alone perform poorly for open-vocabulary segmentation without task-specific adaptation [38, 221]—the gap between a



general-purpose representation and task-specific utility must be bridged by intellectual insight about *how* to transfer the invariant. This is where the methodological contribution of each chapter lies.

Third, foundation models carry biases from their training data, and these biases propagate into any bridge built on them. CLIP’s vision-language alignment reflects the distribution of internet image-text pairs, which is neither culturally neutral nor taxonomically complete. SAM’s boundary quality depends on the distribution of objects seen during training. Monocular depth estimators can fail on unusual materials or adversarial inputs. The invariants that foundation models provide are therefore *empirically robust* across a broad range of conditions, but not *formally guaranteed*—a bridge is only as reliable as the source that provides it.

These limitations motivate two responses in this thesis: a discussion of the biases of bridges in the conclusion (Chapter 7), and a broader examination of the gap discovery problem in Chapter 6.

## 2.4 From Background to Bridges

The preceding sections established the conceptual machinery: OOD generalization requires finding representations where distributional divergence is small (Section 2.1); bridging invariants provide such representations by identifying structural properties shared across the gap (Section 2.2); and foundation models serve as practical sources of these invariants (Section 2.3).

The following three chapters instantiate this framework for three distinct generalization gaps:

- **Chapter 3 (GOOD):** Exploits geometric structure (monocular depth and normals) as a bridging invariant for open-world object detection. The invariant is realized as pseudo bounding boxes generated by a geometry-trained proposal network.
- **Chapter 4 (RENOVATE):** Exploits vision-language alignment as a bridging invariant for open-vocabulary segmentation. The invariant is realized as renovated category names generated by a VLM-guided renaming pipeline.
- **Chapter 5 (LoftUp):** Exploits boundary and region structure as a bridging invariant for feature upsampling. The invariant is realized as pseudo high-resolution features constructed from SAM masks via self-distillation.

In each case, the pattern is the same: identify the invariant, extract it using a foundation model, convert it into pseudo-labels, and use those pseudo-labels to train or guide the task-specific model. Following these concrete instantiations, Chapter 6 examines the upstream question of how distributional gaps are identified in the first place.



## 3 Geometry-Guided Open-World Object Detection (GOOD)

This chapter addresses the first generalization gap: the failure of appearance-based object detectors to discover novel object categories in an open-world setting. The bridging invariant we identify is *geometric structure*—depth and surface normals capture the shape and spatial extent of objects independently of their semantic class. Because all physical objects share geometric properties regardless of their visual appearance, geometry provides a foundation for transferring detection capability from annotated categories to the open world. Figure 3.1 previews the result.

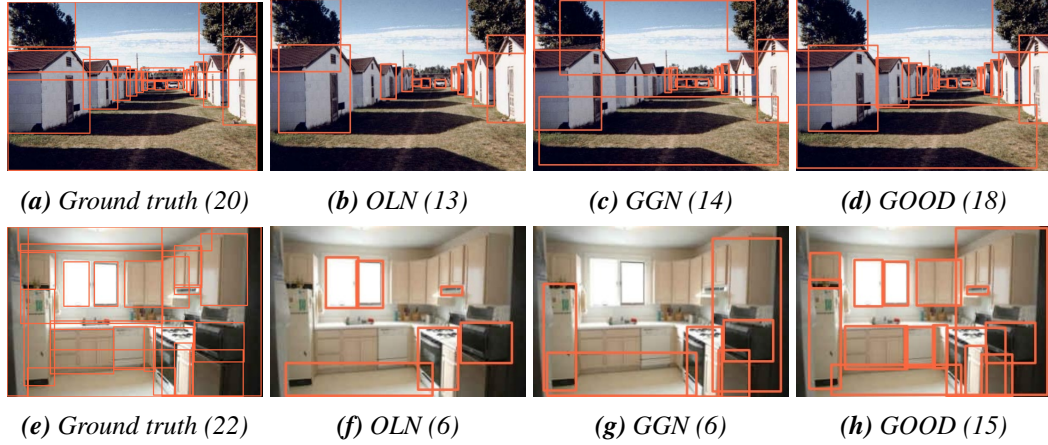
*Note on terminology:* Throughout this chapter, we use “geometric cues” to refer to monocular depth and surface normal estimates. The original publication used the broader term “mid-level representations,” which also encompasses edge maps and pairwise affinities; we retain that term where the broader category is intended.

### 3.1 Introduction

This chapter presents GOOD, a geometry-guided approach to open-world object detection that uses monocular depth and surface normal estimates to discover novel objects beyond the training taxonomy. We show that incorporating these geometric cues through pseudo-labeling substantially improves detection recall for unseen categories across multiple benchmarks.

The standard object detection task is to detect objects from a predefined class list. However, when deploying the model in the real world, it is rarely the case that the model will only encounter objects from its predefined taxonomy. In the open-world setup, object detectors are required to detect all the objects in the scene even though they have only been trained on objects from a limited number of classes. Current state-of-the-art object detectors typically struggle in the open-world setup. As a consequence, open-world object detection has gained increased attention over the last few years [85, 96, 90, 209]. In this work, we specifically address the task of open-world class-agnostic object detection, which is a fundamental task for downstream applications like open-world multi-object tracking [122], robotics [88], and autonomous AI agents [119].

One reason for the failure of current object detectors in the open-world setting is that during training, they are penalized for detecting unlabeled objects in the background and are thus discouraged from detecting them. Motivated by this, previous works have designed different architectures [96, 99] and training pipelines [175, 209] to avoid suppressing the unannotated objects in the background, which have led to significant performance improvements. However, these methods still suffer from overfitting the training classes.

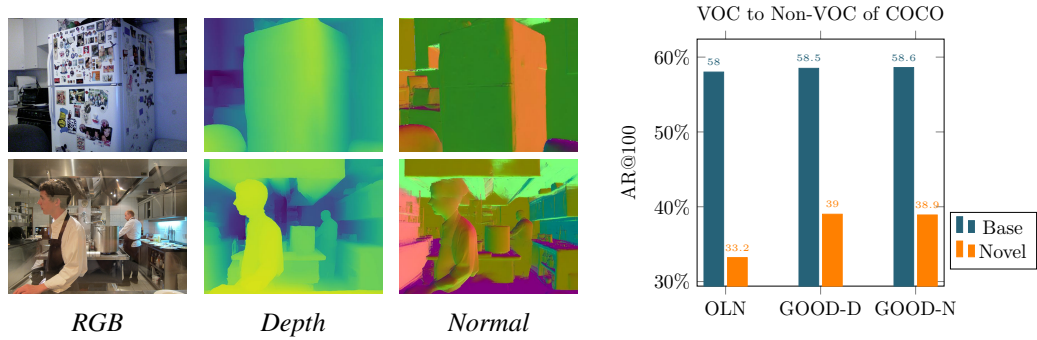


**Figure 3.1: Comparison of GOOD with different baselines.** Images in the first column are from the ADE20K [242] validation set. From the second to fourth columns we show the detection results of OLN [96], GGN [209], and our GOOD. Shown are true-positive proposals from the top 100 proposals of each method; the number of true positives is in parentheses. All models are trained on the PASCAL-VOC classes of COCO [117]. Both OLN and GGN fail to detect many objects not seen during training, while GOOD generalizes better by exploiting geometric cues.

Training only on RGB images, they mainly rely on appearance cues to detect objects of new categories and have great difficulty generalizing to novel-looking objects. Also, there are known shortcut learning problems with regard to training on RGB images [59, 58, 178]—there is no constraint for overfitting the textures or the discriminative parts of the known classes during training. In this work, we propose to tackle this challenge by incorporating geometric cues extracted by general-purpose monocular estimators from the RGB images. We show that such cues significantly improve detection recall for novel object categories on challenging benchmarks.

Estimating geometric cues such as depth and normals from a single RGB image has been an active research area for a long time. Such mid-level representations possess built-in invariance to many changes (e.g., brightness, color) and are more class-agnostic than RGB signals: Fig. 3.2 illustrates how depth and normal maps retain object boundaries while discarding appearance details that distinguish known from unknown categories. In other words, there is less discrepancy between known and unknown objects in terms of geometric cues. In recent years, thanks to stronger architectures and larger datasets [158, 159, 42], monocular estimators for mid-level representations have significantly advanced in terms of prediction quality and generalization to novel scenes. These models are able to compute high-quality geometric cues efficiently when used off-the-shelf as pre-trained models on new datasets. Therefore, it becomes natural to ask if these models can provide additional knowledge for current RGB-based open-world object detectors to overcome the generalization problem.

In this chapter, we propose to use a pseudo-labeling method for incorporating geometric



**Figure 3.2: Geometry cues are complementary to appearance cues for object localization.** The depth and normal cues of the RGB image are extracted using off-the-shelf general-purpose monocular predictors. **Left:** Geometric cues abstract away the appearance details and focus on more holistic information such as object shapes and relative spatial locations (depth) and directional changes (normals). **Right:** By incorporating geometric cues, GOOD generalizes better than the RGB-based model OLN [96], i.e., much smaller AR gaps between the base and novel classes.

cues into open-world object detector training. We first train an object proposal network on the predicted depth or normal maps to discover novel unannotated objects in the training set. The top-ranked novel object predictions are used as pseudo boxes for training the open-world object detector on the original RGB input. We observe that incorporating the geometric cues can significantly improve the detection recall of unseen objects, especially those that differ strongly from the training objects. Figure 3.2 highlights this effect: objects that appear indistinguishable from the background in RGB (e.g., food items on a table) stand out clearly in depth and normal maps. We attribute this improvement to the complementary nature of geometric and RGB-based detection cues. Geometric cues help discover novel-looking objects that RGB-based detectors miss. In turn, the RGB-based detectors leverage the additional annotations from pseudo-labeling, using their strong representation learning ability to generalize to novel, unseen categories.

Our resulting Geometry-guided Open-world Object Detector (GOOD) surpasses the state-of-the-art performance on multiple benchmarks for open-world class-agnostic object detection. Thanks to the rich geometry information, GOOD can generalize to unseen categories with only a few known classes for training. Particularly, with a single training class “person” on the COCO dataset [117], GOOD can surpass state-of-the-art methods by 5.0% AR@100 (a relative improvement of 24%) on detecting objects from non-person classes. With 20 PASCAL-VOC classes for training, GOOD surpasses state-of-the-art methods even by 6.1% AR@100 in detecting non-VOC classes. Furthermore, we also analyze the advantages of geometric cues and show that they are less sensitive to semantic shifts across classes, and are better than other strategies for improving generalization.

## 3.2 Related Work

We review two bodies of work that motivate our approach: open-world class-agnostic object detection methods that aim to generalize beyond the training taxonomy, and methods that incorporate geometric cues to improve generalization in visual perception tasks.

**Open-world class-agnostic object detection** is the task of localizing all the objects in an image by learning with only a limited number of object classes (base classes). The core problem with standard object detection training is that the model learns to classify unannotated objects as background and thus suppresses their detection at inference time. To solve this issue, Kim et al. [96] proposed object localization network (OLN), which replaces the classification heads of Faster RCNN [164] with class-agnostic objectness heads so that the training loss is only calculated on positive samples, i.e., known objects, and thus not suppressing the detection of unannotated novel objects. Saito et al. [175] synthesized a training set by copy-pasting known objects onto synthetic backgrounds. However, the model struggles with the synthetic-to-real domain gap in solving the object detection task. Besides background non-suppression, a more proactive approach is to exploit unannotated objects for training. Wang et al. [209] built upon traditional learning-free methods and developed a pairwise affinity predictor to discover unannotated objects. Their object detector, GGN, is then trained using the newly-discovered object masks and ground truth base class annotations as supervision. Finally, another promising direction is to use open-world knowledge from large pretrained multi-modal models. Recently, Minderer et al. [138] and Maaz et al. [130] made use of pretrained language-vision model [157] to detect open-world objects using text queries. Our work is most related to OLN and GGN. We used the OLN architecture and training loss, but additionally incorporated geometric cues through our pseudo-labeling method. GGN used the pairwise affinity for pseudo labeling. However, since the pairwise affinity is trained on RGB inputs using the base class annotations, GGN still suffers from the overfitting problems of RGB-based methods. Our experiments showed geometric cues as a better source of pseudo boxes.

**Incorporating geometric cues for generalization.** The estimation of geometric cues has been an active research area for decades. With the introduction of deep neural networks, the seminal work by Eigen, Puhrsch, and Fergus [44] and Eigen and Fergus [43] significantly improved over early works [75, 76, 78, 77, 179, 180, 181]. Recent progress in estimating geometric cues can be attributed to the use of modern architectures [159], stronger training strategies [234] and large-scale datasets [42]. In particular, Omnidata [42] has made significant headway in prediction quality and cross-dataset generalization. Since geometric cues abstract away the appearance details and retain more holistic information about the objects, such as shapes, they have been incorporated into many applications for generalization. For example, Xian et al. [214] incorporated them into the 3D shape completion pipeline to generalize to novel classes. Yu et al. [232] used them to guide the optimization of neural implicit surface models for tackling scenes captured from sparse viewpoints. Chen et al. [22] applied mid-level visual representations to reinforcement training and gained robustness under domain shifts. Building on this evidence that geometric cues transfer well across visual domains, we propose to incorporate them through pseudo-labeling for open-world class-agnostic object detection, extending their generalization benefits to a

new task.

**Post-publication developments.** Since the publication of GOOD at ICLR 2023, the open-world detection landscape has evolved significantly. Foundation-model-based approaches have gained prominence: Grounding DINO [121] combines DINO-based detection with language grounding for open-set object detection, while OWLv2 [137] scales open-vocabulary detection through self-training on web-scale data. YOLO-World [29] extends YOLO architectures with vision-language pre-training for real-time open-vocabulary detection. These methods leverage vision-language alignment rather than geometric cues, addressing the same semantic shift gap through a different bridging invariant.

GOOD’s geometric-cue philosophy has also influenced subsequent work. He et al. [71] extend the approach from 2D to 3D, building a 3D Object Discovery Network that uses geometric cues—box location and scale in 3D space—to generate class-agnostic proposals for autonomous driving. Zhang et al. [237] take a complementary route: rather than using geometric cues as direct supervision, they use colorized depth maps as appearance-destroying view transformations and enforce cross-view consistency to learn appearance-invariant representations for open-world instance segmentation. Both works validate the central premise of GOOD: that geometric structure generalizes more reliably than appearance for detecting novel objects.

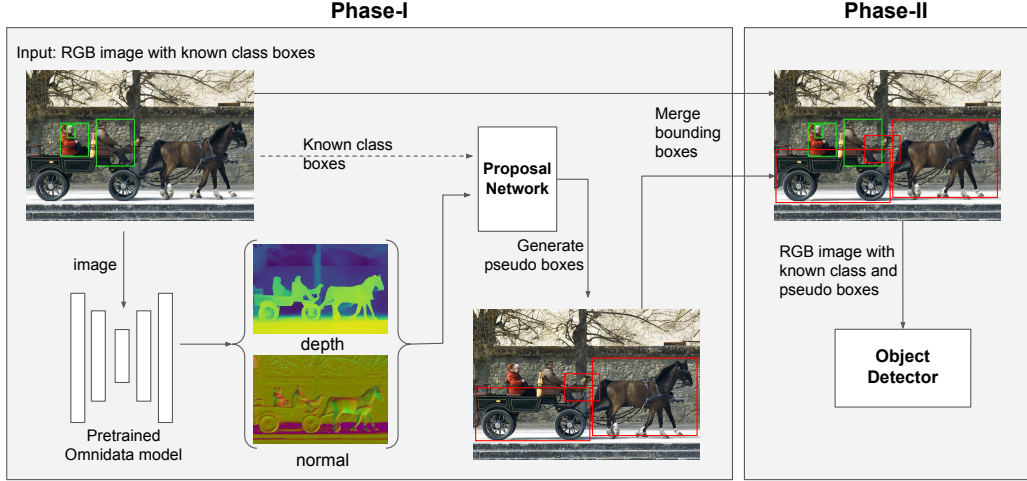
The complementarity between geometric and semantic approaches remains an open direction: combining GOOD’s geometry-guided pseudo-labeling with the broader vocabulary coverage of vision-language detectors could yield detectors that are robust to both appearance-defined and shape-defined novel objects.

### 3.3 Method

Our goal is to incorporate geometric cues for an improved open-world class-agnostic object detection performance. Concretely, we propose a pseudo labeling method, which can effectively utilize the geometric cues to detect unannotated novel objects in the training set and then use them for training the object detector. Figure 3.3 provides an overview, showing the two-phase pipeline: Phase-I trains a proposal network on geometric cues to generate pseudo boxes, and Phase-II trains an RGB-based detector using these pseudo boxes alongside ground-truth annotations.

#### 3.3.1 Open-World Class-Agnostic Object Detection Problem

Current state-of-the-art object detection methods work well under the closed-world assumption. They are trained with a set of object bounding box annotations  $\{t_i\}$  from a pre-specified list  $\mathcal{K}$  of semantic classes, i.e., the base classes. At test time, their generalization is evaluated by detecting the objects from the known base classes. The standard training loss of object detection for an image  $\mathcal{I}$  has two parts: classification loss and bounding box



**Figure 3.3: Overview of the geometry-guided pseudo labeling method.** It consists of two training phases. Phase I: the RGB input is firstly preprocessed by the off-the-shelf model to extract the geometry cues, which are then used to train an object proposal network with the base class bounding box annotations. The proposal networks then pseudo-label the training samples, discovering unannotated novel objects. The top-ranked pseudo boxes are added to the annotation pool for Phase II training, i.e., a class-agnostic object detector is directly trained on the RGB input using both the base class and pseudo annotations. At inference time, we only need the model from Phase II.

regression loss

$$\mathcal{L}(\mathcal{I}) = \frac{1}{N_{cls}} \sum_{i \in \mathcal{B}} \mathcal{L}_{cls}(p_i, p_i^*) + \frac{1}{N_{reg}} \sum_{i \in \mathcal{B}_{\mathcal{K}}} \mathcal{L}_{reg}(t_i, t_i^*), \quad (3.1)$$

where  $i$  is the index of an anchor from the candidate set  $\mathcal{B}$ ,  $\{p_i, t_i\}$  are the predicted label and bounding box coordinates, and  $\{p_i^*, t_i^*\}$  are the corresponding ground truths.  $N_{cls}$  is the total number of anchors in the candidate set  $\mathcal{B}$ .  $N_{reg}$  is the size of the subset  $\mathcal{B}_{\mathcal{K}}$ , which only contains the anchors with the ground truth label  $p_i^* = 1$ . Note  $p_i^*$  equals 1 only when the anchor  $i$  can be associated to an annotated object bounding box from the known classes  $\mathcal{K}$ ; otherwise it equals 0 (background).

From the closed-world to the open-world setup, the generalization goal extends to localizing every object in the image, which can belong to an unknown novel class  $u \in \mathcal{U}$ . Under the training loss in Eq. (3.1), an unannotated object will be classified as “background”. As a result, the model will treat similar types of objects as “background” at inference time. To avoid suppressing the detection of novel objects in the background, Kim et al. [96] proposed to replace the classification loss in Eq. (3.1) by the objectness score prediction loss,



yielding

$$\mathcal{L}_{OLN}(\mathcal{I}) = \frac{1}{N_{reg}} \sum_{i \in \mathcal{B}_K} \mathcal{L}_{reg}(t_i, t_i^*) + \frac{1}{N_{reg}} \sum_{i \in \mathcal{B}_K} \mathcal{L}_{obj}(o_i, o_i^*), \quad (3.2)$$

where  $o_i$  and  $o_i^*$  are the predicted objectness score and its ground truth of anchor  $i$ . In doing so, only the anchors with  $p_i^* = 1$  are involved in training, completely removing any “background” prediction. At inference time, the objectness score is used to rank the detections. However, since these anchors only capture the annotated objects from the base classes, this loss modification cannot effectively mitigate the overfitting to the base classes. We further resort to adding additional “objects” into training, especially novel ones with very different appearances than objects from the base classes.

### 3.3.2 Exploiting Geometric Cues

Models trained on RGB images tend to over-rely on the appearance cues for object detection. Therefore, it is hard for them to detect novel objects that appear very differently from the base classes. For instance, a model trained on cars is likely to detect trucks, but unlikely to also detect sandwiches. Involving such novel objects, e.g., food, into training is then an effective way to mitigate the appearance bias towards the base classes, e.g., vehicles. To this end, we exploit two types of geometric cues, i.e., depth and normals, for detecting unannotated novel objects in the training set, see some examples in Fig. 3.2. Both of them are common geometric cues that capture local information. Depth focuses on the relative spatial difference of objects and abstracts away the details on the object surfaces. Surface normals focus on the directional difference and remain the same on flat surfaces. Compared with the original RGB image, they discard most appearance details and focus on the geometry information such as object shapes and relative spatial locations. Models trained with them can thus discover many novel-looking objects that RGB-based ones cannot detect.

In the terminology of this thesis, the bridging invariant exploited by GOOD is geometric structure: depth and normals are shared across known and novel categories, providing a stable signal that bridges the semantic shift gap.

We use off-the-shelf pretrained models to extract geometric cues. Specifically, we use Omnidata models [42] trained using cross-task consistency [234] and 2D/3D data augmentations [93]. The training dataset for the models is the Omnidata Starter Dataset (OSD) [42] which contains 2200 real and rendered scenes. Despite the difference between the OSD to the object detection benchmark datasets, the Omnidata model can produce high-quality results, implying that the invariances behind these geometric cues are robust.

### 3.3.3 Pseudo Labeling Method

To use the geometric cues to discover unannotated novel objects in the training set, we first train an object proposal network on the depth or normal input using the same training loss as in Eq. (3.2), i.e., Phase-I training in Fig. 3.3. Then, this object proposal network will pseudo-label the training images using its detected bounding boxes. After filtering out

the detected bounding boxes which overlap with the base class annotations, we then add the remaining top- $k$  boxes to the ground truth annotations. Here  $k \in \{1, 2, 3\}$  is determined for each detector on a small holdout validation set. Finally, we train a new class-agnostic object detector using the RGB image as input and the extended bounding box annotation pool as ground truth, i.e., Phase-II in Fig. 3.3. The training loss is

$$\mathcal{L}_{GOOD}(\mathcal{I}) = \frac{1}{N_{reg}} \sum_{i \in \mathcal{B}_K \cup \mathcal{B}_N} \mathcal{L}_{reg}(t_i, t_i^*) + \frac{1}{N_{reg}} \sum_{i \in \mathcal{B}_K \cup \mathcal{B}_N} \mathcal{L}_{obj}(o_i, o_i^*). \quad (3.3)$$

Compared with Eq. (3.2), the anchors that overlap with the pseudo boxes of the detected novel objects, i.e.,  $i \in \mathcal{B}_N$ , are also involved in training. The pseudo boxes can be acquired from a single source, i.e., one of the geometric cues, and from both, i.e., pseudo label ensembling. We name our method GOOD-X when using a specific geometric cue X as the pseudo labeling source, whereas GOOD-Both stands for ensembling the pseudo labels from both the depth and normals.

Inspired by previous works in self-training [216, 192, 219], we use strong data augmentation during Phase-II to counteract the noise in pseudo boxes and further boost the performance of GOOD. Specifically, for Phase-II training, we use AutoAugment [34] which includes random resizing, flip, and cropping.

### 3.4 Experiments

**Benchmarks.** We target two major challenges of open-world class-agnostic object detection: *cross-category and cross-dataset generalization*. For the cross-category evaluation, we follow the common practice in the literature [96, 209] to split the class category list into two parts. One is used as the base class for training, whereas the other is reserved only for testing cross-category generalization. Specifically, we adopt two splits of the 80 classes in the COCO dataset [117]. The first benchmark splits the COCO classes into a single “person” class and 79 non-person classes. This is to stress-test the generalization ability of the methods. We follow Wang et al. [209] to choose the “person” category as the training class because it contains diverse viewpoints and shapes. The second benchmark splits the COCO classes into 20 PASCAL-VOC [46] classes for training and the other 60 for testing. For the cross-dataset evaluation, we use the ADE20K dataset [241] for testing. We compare models trained using only 20 PASCAL-VOC classes or all 80 COCO classes on detecting objects in ADE20K. This is to evaluate open-world class-agnostic object detectors when used in the wild.

**Implementation.** The object proposal network in Phase I and the object detector in Phase II both use the OLN architecture [96], built on top of a standard Faster R-CNN [164] with a ResNet-50 backbone pre-trained on ImageNet [37]. For geometric cue extraction, we use Omnidata’s DPT-Hybrid model [42], which provides monocular depth and surface normal predictions. All experiments are implemented using the MMDetection framework [23]. We use SGD with an initial learning rate of 0.01 and batch size of 16. Phase I models are trained for 8 epochs. Phase II models are trained for 16 epochs when using AutoAugment [34] (which includes random resizing, flipping, and cropping) and 8 epochs



otherwise; we did not find longer training beneficial without data augmentation. During pseudo box generation, we filter out boxes that overlap with base class annotations ( $\text{IoU} > 0.5$ ) and retain the top- $k$  remaining boxes per image, where  $k \in \{1, 2, 3\}$  is selected on a small holdout validation set. For pseudo label ensembling (GOOD-Both), we merge the pseudo box pools from both the depth- and normal-trained proposal networks before selecting the final set.

**Evaluation metrics.** Following [96, 209], we use Average Recall ( $\text{AR}@k$ ) over multiple IoU thresholds (0.5:0.95), and set the detection budget  $k$  as 100 by default. All ARs are shown in percentage.  $\text{AR}_A$  and  $\text{AR}_N$  respectively denote the AR score on detecting all classes (including the base and novel ones) and on detecting the novel classes. To evaluate  $\text{AR}_N$ , we do not count the boxes associated to the base classes into the budget  $k$ . The same protocol is applied when evaluating per-class ARs and ARs for small, medium and large size of objects, i.e.,  $\text{AR}^{s/m/l}$ .

### 3.4.1 Detecting Unknown Objects in an Open World

Table 3.1a and Table 3.1b compare open-world class-agnostic object detectors on two cross-category benchmarks and two cross-dataset benchmarks, respectively. Our method GOOD, which incorporates geometric cues, considerably outperforms state-of-the-art RGB-based open-world class-agnostic object detection methods, i.e., OLN [96] and GGN [209]. OLN did not involve any novel object into training. Although GGN proposed to use an intermediate pairwise affinity (PA) representation for pseudo labeling, the PA predictor is still trained on the RGB images, therefore it still has the bias towards the known classes as other RGB-based methods.

On the cross-category benchmarks shown in Table 3.1a, with a single training class “person” on the COCO dataset, GOOD surpasses state-of-the-art methods by 5.0%  $\text{AR}_N@100$  on detecting objects from non-person classes, a relative improvement of 24%. With 20 PASCAL-VOC classes for training, GOOD surpasses state-of-the-art methods by 6.1%  $\text{AR}_N@100$  in detecting non-VOC classes, a relative improvement over 18%. On the cross-dataset benchmarks, Table 3.1b shows that GOOD achieves 2.4% to 4.7% gain on  $\text{AR}_A@100$  in different setups. We observe that GOOD is particularly strong when there are fewer training classes, i.e., person  $\rightarrow$  non-person and VOC  $\rightarrow$  ADE20k. For RGB-based methods, the overfitting problems become more severe as the object diversity from the base classes reduces. Geometric cues remain effective even in this regime, detecting novel-looking objects that are particularly helpful when training with only a limited number of base class annotations. Finally, ensembling both geometric cues offers additional performance gains, confirming that depth and normals capture complementary aspects of object structure.

We further evaluate on additional benchmarks: the cross-category LVIS COCO to non-COCO and the cross-dataset COCO to UVO. The results in Table 3.2 show that while performance gains are smaller than on benchmarks with fewer base classes, GOOD still surpasses all state-of-the-art RGB-based methods, demonstrating that even with a large number of base classes, geometric cues improve open-world detection.

To provide further insight into the quality of pseudo boxes, Fig. 3.4 shows example pre-

|                 | Person→Non-Person |                 |                              |                              |                              | VOC→Non-VOC     |                 |                              |                              |                              |
|-----------------|-------------------|-----------------|------------------------------|------------------------------|------------------------------|-----------------|-----------------|------------------------------|------------------------------|------------------------------|
| Method          | AR <sub>A</sub>   | AR <sub>N</sub> | AR <sub>N</sub> <sup>s</sup> | AR <sub>N</sub> <sup>m</sup> | AR <sub>N</sub> <sup>l</sup> | AR <sub>A</sub> | AR <sub>N</sub> | AR <sub>N</sub> <sup>s</sup> | AR <sub>N</sub> <sup>m</sup> | AR <sub>N</sub> <sup>l</sup> |
| FRCNN (oracle)  | 55.9              | 53.4            | 37.9                         | 59.5                         | 73.0                         | 55.9            | 52.6            | 37.1                         | 60.0                         | 73.1                         |
| FRCNN (cls-agn) | 25.7              | 12.2            | 8.7                          | 12.4                         | 18.2                         | 38.5            | 27.3            | 10.8                         | 30.2                         | 55.8                         |
| OLN [96]        | 30.9              | 16.5            | 8.7                          | 14.7                         | 33.4                         | 47.5            | 33.2            | 18.7                         | 39.3                         | 58.6                         |
| GGN [209]       | 30.3              | 20.7            | 12.0                         | 25.6                         | 29.6                         | 39.8            | 31.5            | 11.8                         | 37.4                         | <b>63.8</b>                  |
| SelfTrain-RGB   | 32.5              | 18.7            | 11.3                         | 18.6                         | 32.6                         | 48.1            | 37.4            | <b>22.8</b>                  | 43.9                         | 57.7                         |
| GOOD-Depth      | 37.0              | 25.6            | 12.8                         | 30.4                         | <b>42.4</b>                  | <b>49.6</b>     | 39.0            | 21.1                         | 47.5                         | 63.2                         |
| GOOD-Normal     | 35.6              | 23.7            | 13.9                         | 27.6                         | 36.0                         | <b>49.6</b>     | 38.9            | 21.2                         | 47.9                         | 62.0                         |
| GOOD-Both       | <b>37.3</b>       | <b>25.9</b>     | <b>14.2</b>                  | <b>32.6</b>                  | 38.9                         | 49.5            | <b>39.3</b>     | 21.6                         | <b>48.2</b>                  | 62.4                         |

(a) Cross-category benchmarks

|                 | VOC→ADE20K      |                              |                              |                              | COCO→ADE20K     |                              |                              |                              |
|-----------------|-----------------|------------------------------|------------------------------|------------------------------|-----------------|------------------------------|------------------------------|------------------------------|
| Method          | AR <sub>A</sub> | AR <sub>A</sub> <sup>s</sup> | AR <sub>A</sub> <sup>m</sup> | AR <sub>A</sub> <sup>l</sup> | AR <sub>A</sub> | AR <sub>A</sub> <sup>s</sup> | AR <sub>A</sub> <sup>m</sup> | AR <sub>A</sub> <sup>l</sup> |
| FRCNN (cls-agn) | 22.6            | 15.5                         | 23.7                         | 26.5                         | 25.9            | 20.5                         | 28.5                         | 27.4                         |
| OLN [96]        | 29.2            | 19.7                         | 30.7                         | 34.4                         | 32.9            | 25.1                         | 35.9                         | 35.6                         |
| GGN [209]       | 27.0            | 16.9                         | 27.5                         | 33.6                         | 29.8            | 18.9                         | 29.1                         | 38.2                         |
| SelfTrain-RGB   | 27.7            | 18.4                         | 29.7                         | 32.4                         | 33.8            | <b>26.5</b>                  | 37.6                         | 35.4                         |
| GOOD-Depth      | 33.9            | 21.1                         | 35.9                         | 41.2                         | <b>35.3</b>     | 25.7                         | 38.0                         | 39.6                         |
| GOOD-Normal     | 33.4            | 22.0                         | 36.2                         | 38.8                         | 33.5            | 25.7                         | 37.1                         | 35.5                         |
| GOOD-Both       | <b>34.0</b>     | 21.9                         | <b>37.0</b>                  | <b>39.9</b>                  | <b>35.3</b>     | 25.1                         | 38.2                         | <b>39.9</b>                  |

(b) Cross-dataset benchmarks

**Table 3.1: Detecting unknown objects in an open world.** FRCNN (oracle) is a standard Faster R-CNN detector trained on all COCO classes and serves as a performance upper bound on the cross-category benchmarks. FRCNN (cls-agn) is a Faster R-CNN trained in a class-agnostic manner, serving as a baseline for comparison. Our geometry-guided methods GOOD-Xs are compared with SOTA open-world class-agnostic object detectors, i.e., OLN and GGN. SelfTrain-RGB is RGB-based self-training, i.e., using the RGB image instead of geometric cues for pseudo labeling. We only report AR<sub>A</sub> in b) as the classes in ADE20K do not exactly match those in COCO.

|                 | LVIS COCO→Non-COCO |                 |                              |                              |                              | COCO→UVO        |                              |                              |                              |
|-----------------|--------------------|-----------------|------------------------------|------------------------------|------------------------------|-----------------|------------------------------|------------------------------|------------------------------|
| Method          | AR <sub>A</sub>    | AR <sub>N</sub> | AR <sub>N</sub> <sup>s</sup> | AR <sub>N</sub> <sup>m</sup> | AR <sub>N</sub> <sup>l</sup> | AR <sub>A</sub> | AR <sub>A</sub> <sup>s</sup> | AR <sub>A</sub> <sup>m</sup> | AR <sub>A</sub> <sup>l</sup> |
| FRCNN (cls-agn) | 26.6               | 21.0            | 14.9                         | 32.7                         | 36.2                         | 42.3            | 22.2                         | 38.3                         | 52.0                         |
| OLN [96]        | 32.2               | 27.4            | 17.9                         | 44.7                         | 53.1                         | 49.2            | 35.0                         | 48.7                         | 55.1                         |
| GGN [209]       | 27.1               | 22.5            | 15.7                         | 35.5                         | 38.4                         | 45.6            | 25.6                         | 43.2                         | 54.9                         |
| SelfTrain-RGB   | 32.6               | 28.3            | 19.0                         | 45.7                         | 52.2                         | 48.7            | 35.8                         | 48.8                         | 53.5                         |
| GOOD-Depth      | 32.8               | 28.3            | 18.3                         | 46.8                         | <b>54.1</b>                  | <b>50.3</b>     | 35.6                         | 49.9                         | <b>56.4</b>                  |
| GOOD-Normal     | <b>33.4</b>        | <b>29.2</b>     | <b>19.3</b>                  | <b>49.8</b>                  | 53.3                         | 49.8            | 35.8                         | 50.0                         | 55.0                         |
| GOOD-Both       | 33.2               | 29.0            | 19.0                         | 47.7                         | 53.0                         | <b>50.3</b>     | <b>36.1</b>                  | <b>50.2</b>                  | 55.4                         |

**Table 3.2: More benchmarks.** We evaluate on the LVIS COCO to non-COCO cross-category benchmark and the COCO to UVO cross-dataset benchmark.

dictions from proposal networks trained on RGB, depth, and normal inputs. The depth and normal modalities consistently discover volumetric objects with distinctive 3D structure—such as furniture, animals, and appliances—that the RGB-based proposal network misses or localizes imprecisely. Conversely, the RGB model can detect smaller and flatter objects where geometric cues provide limited discriminative signal.

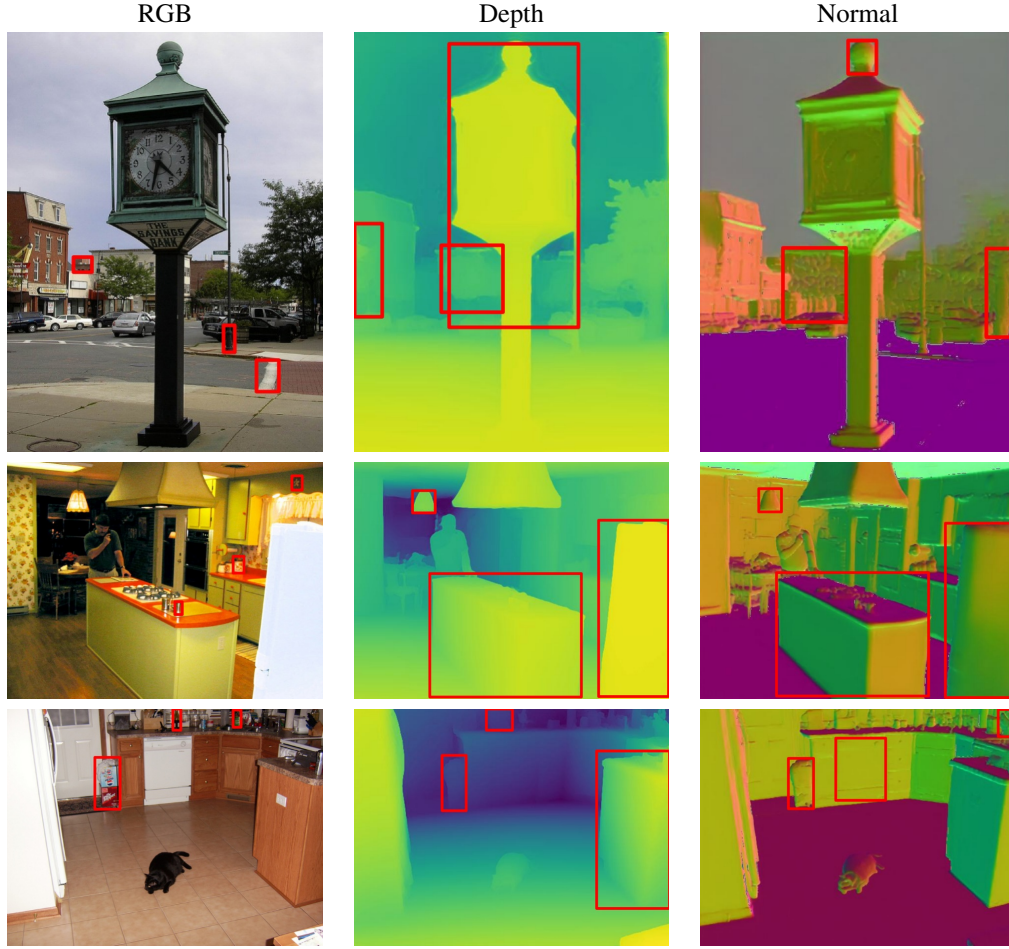
### 3.4.2 Advantages of Geometric Cues

Having established that GOOD improves detection recall across benchmarks, we now analyze why geometric cues are effective and compare them against alternative strategies for improving generalization.

**Geometric cues are less sensitive to appearance shifts across classes.** We first compare per-novel-class AR@5 of the object proposal network trained on geometric cues with that trained on the RGB image. Here, AR@5 is of interest as geometric cues are used to discover novel-looking objects during Phase-I and no more than five pseudo boxes per image will be used in Phase-II (see the pseudo-label filtering step in Fig. 3.3).

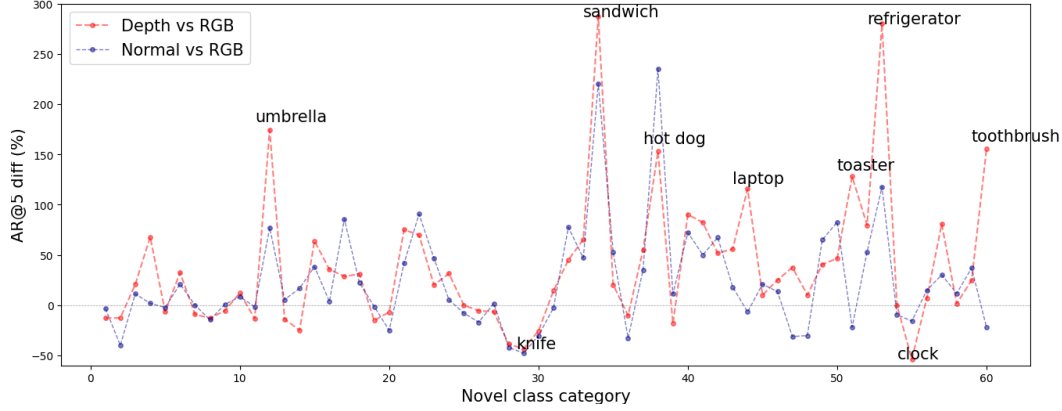
Figure 3.5 shows that geometric cues can achieve much higher per-novel-class AR@5 than RGB in many categories. An example is the novel supercategory “food”, including classes such as “hot dog” and “sandwich”. The base classes, belonging to the supercategory “person”, “animal”, “vehicle”, and “indoor”, have very different appearances to the “food” supercategory. The RGB-based model has difficulty detecting food using appearance cues, whereas geometric cues generalize well across supercategories. For categories that geometric cues are worse than RGB, we find that they typically are of small sizes, such as knives, forks, and clocks. This shows that while abstracting away appearance details, geometric cues may also lose some information about small objects. This again shows complementarity of RGB and geometric cues.

**Geometric cues are better than edges and pairwise affinities.** We further compare the geometric cues with two other mid-level representations: 2D edge and PA. The 2D edge map is extracted using the Holistically-nested Edge Detection (HED) model [217],



**Figure 3.4: Visualization of top-3 pseudo boxes** generated by proposal networks trained on RGB (left), depth (center), and normal (right) inputs. Geometric modalities consistently discover volumetric objects with distinctive 3D structure that the RGB-based model misses.

which shows more robust performance across the datasets than more recent methods. The HED model is trained on the Berkeley Segmentation Dataset and Benchmark (BSDS500) dataset [6] which contains 500 images with segmentation annotations. PA can be thought of as a learned object boundary predictor. Wang et al. [209] trained it directly on RGB images and then grouped the predictions into object masks using a combination of traditional grouping methods [185, 5, 4]. We compare these four data modalities as the source of pseudo labeling. From Table 3.3, we can see that depth and normals outperform 2D edge and PA on detecting novel objects. This likely reflects the fact that 2D edge and PA mainly capture object boundaries, whereas depth and normals encode an additional spatial understanding of the scene and can thus better detect objects in complex scenes. Owing to their better pseudo labels, GOOD-Depth/Normal outperform GOOD-Edge/PA on the final  $AR_N@100$ , i.e., 39.0%/38.9% vs. 38.1%/37.1%.



**Figure 3.5: Per-novel-class AR@5 difference comparison of pseudo boxes on COCO VOC  $\rightarrow$  Non-VOC.** We train the object proposal network on the geometric cues (Phase-I in Figure 3.3) and also directly on the RGB image. We show their per-novel-class AR@5 differences, which is defined as  $(AR_X - AR_{RGB})/AR_{RGB}$  with  $X \in \{\text{Depth}, \text{Normal}\}$ . The geometric cues outperform the RGB image on those classes above the zero difference line.

| Modality               | $AR_N @ 1$ | $AR_N @ 5$  | $AR_N^s @ 5$ | $AR_N^m @ 5$ | $AR_N^l @ 5$ |
|------------------------|------------|-------------|--------------|--------------|--------------|
| RGB                    | 4.0        | 12.6        | <b>5.4</b>   | 16.4         | 21.8         |
| Depth                  | <b>4.7</b> | <b>13.2</b> | 2.4          | 16.0         | <b>31.6</b>  |
| Normal                 | 4.3        | 12.8        | 3.0          | <b>16.6</b>  | 27.7         |
| Edge                   | 3.2        | 10.4        | 3.5          | 14.5         | 18.5         |
| Pairwise affinity (PA) | 3.2        | 8.1         | 0.5          | 9.1          | 30.6         |

**Table 3.3: Comparison on different data modalities for pseudo labeling.** We report  $AR_N @ k$  achieved by the object proposal network trained on different modalities in Phase-I, where the benchmark is VOC  $\rightarrow$  Non-VOC. Geometric cues (depth and normal) are stronger in discovering novel objects than the edge and pairwise affinity [209].

**Geometric cues are better than adding shape bias and multi-scale augmentation.** It is thus natural to compare with an RGB-based model with inductive shape bias to counteract the texture bias. We adopt the stylized ImageNet pretrained ResNet-50 from [59] as the backbone for SelfTrain-RGB (an OLN model that self-trains with its own RGB-based pseudo boxes). This backbone is trained using heavy style-based augmentation to mitigate the texture bias, where texture is one of the most discussed appearance features prone to overfit by RGB-based models. It showed performance improvements on the COCO object detection benchmark under the closed-world assumption [59]. Moreover, we observe from Fig. 3.5 that RGB is relatively stronger in detecting smaller objects. This leads to the question of whether augmenting SelfTrain-RGB with pseudo boxes extracted at different input scales can already help this RGB-based method achieve comparable performance to those incorporating geometric cues.

From Table 3.4, we can see that while the shape bias backbone can improve the  $AR_A$  of SelfTrain-RGB, it degrades the performance on novel classes, i.e.,  $AR_N$ , indicating that

| Method        | AR <sub>A</sub> | AR <sub>N</sub> | AR <sub>N</sub> <sup>s</sup> | AR <sub>N</sub> <sup>m</sup> | AR <sub>N</sub> <sup>l</sup> |
|---------------|-----------------|-----------------|------------------------------|------------------------------|------------------------------|
| SelfTrain-RGB | 48.1            | 37.4            | <b>22.8</b>                  | 43.9                         | 57.7                         |
| ShapeBias     | 48.8            | 36.5            | 21.4                         | 42.5                         | 58.6                         |
| ScaleAug      | 48.5            | 37.9            | 21.2                         | 44.7                         | 62.2                         |
| GOOD-Both     | <b>49.5</b>     | <b>39.3</b>     | 21.6                         | <b>48.2</b>                  | <b>62.4</b>                  |

**Table 3.4: Comparison of GOOD and two other strategies on VOC to non-VOC.** Shape-Bias replaced the backbone of SelfTrain-RGB with a stylized ImageNet pre-trained backbone. ScaleAug applied multi-scale augmentation to the RGB input for collecting pseudo boxes at different scales.

shape bias obtained from training on ImageNet may not be very helpful in generalizing to novel objects. As for the multi-scale augmentation, although it can improve the detection recall of medium and large objects, the overall performance is still worse than our models that incorporate geometric cues. Overall, these comparisons show that geometry provides strong cues for object localization.

### 3.4.3 Ablation Studies

In this section, we conduct ablation studies to understand the effectiveness of pseudo-labeling, data augmentation, and the influence of the number of base classes.

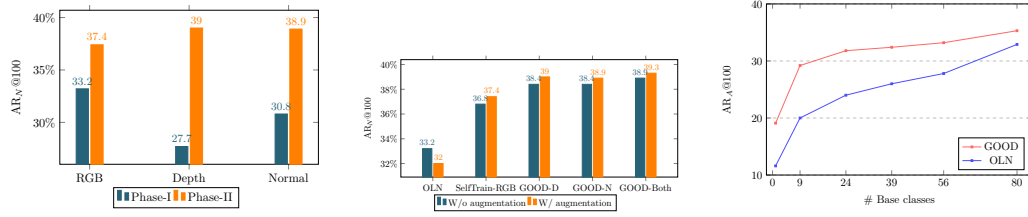
Figure 3.6 shows the AR<sub>N</sub>@100 achieved by the object proposal network (Phase-I) and the object detector (Phase-II). The latter uses the pseudo boxes generated by the former, where the former can be trained with different data modalities. The RGB-based object proposal network outperforms the depth- and normal-based one on AR<sub>N</sub>@100. This indicates that depth and normal maps cannot simply replace the RGB image for object detection.

However, as evidenced in Table 3.3 and Fig. 3.5, geometric cues are competitive on AR<sub>N</sub>@k with  $k \leq 5$ , meaning their top predictions are of high quality. Importantly, AR<sub>N</sub>@5 directly measures pseudo-label quality in our pipeline: it quantifies how many of the top-ranked proposals from Phase-I correspond to actual novel objects, and these top proposals are precisely the pseudo-labels transferred to Phase-II. Depth achieves AR<sub>N</sub>@5 of 13.2 compared to RGB’s 12.6 on non-VOC classes (Table 3.3), confirming that geometric pseudo-labels are of higher quality for novel categories despite their lower overall AR<sub>N</sub>@100.

Furthermore, the overlap heatmap in Fig. 3.8 shows that different modalities discover largely non-overlapping sets of novel objects (below 20% overlap at IoU 0.7), meaning that ensembling adds genuinely new pseudo-labels rather than redundant ones. Pseudo labeling is thus an effective way for geometry-guided open-world object detector training.

We further study the effect of AutoAugment [34]. Using it during Phase-II training, we achieve higher AR<sub>N</sub>@100 as shown in Fig. 3.6. Data augmentation is helpful to counteract the noise in pseudo labels. However, using AutoAugment to train OLN on the ground truth base class annotations, we observe only improvement in recalling the base class objects (from 58.4% to 61.7% AR<sub>Base</sub>@100), but degradation in novel object detection. This





(a) Effectiveness of pseudo labeling. (b) Effectiveness of data augmentation. (c) Varying the number of base classes.

**Figure 3.6: Ablation studies.** (a) and (b) are conducted on COCO VOC to non-VOC benchmark. The study on the number of base classes (c) uses ADE20K as the evaluation benchmark.

| Supercategories      | Person | +Vehicle | +Outdoor, Animal | +Accessory, Sports | +Kitchen, Food | +Furniture, Electronic, Appliance, Indoor |
|----------------------|--------|----------|------------------|--------------------|----------------|-------------------------------------------|
| # Training classes   | 1      | 9        | 24               | 39                 | 56             | 80                                        |
| # Training images    | 64115  | 74152    | 92169            | 93939              | 107036         | 117266                                    |
| # Training instances | 609666 | 654460   | 715582           | 727207             | 824535         | 860001                                    |

**Table 3.5: Base class choices for studying the influence of the number of base classes.**

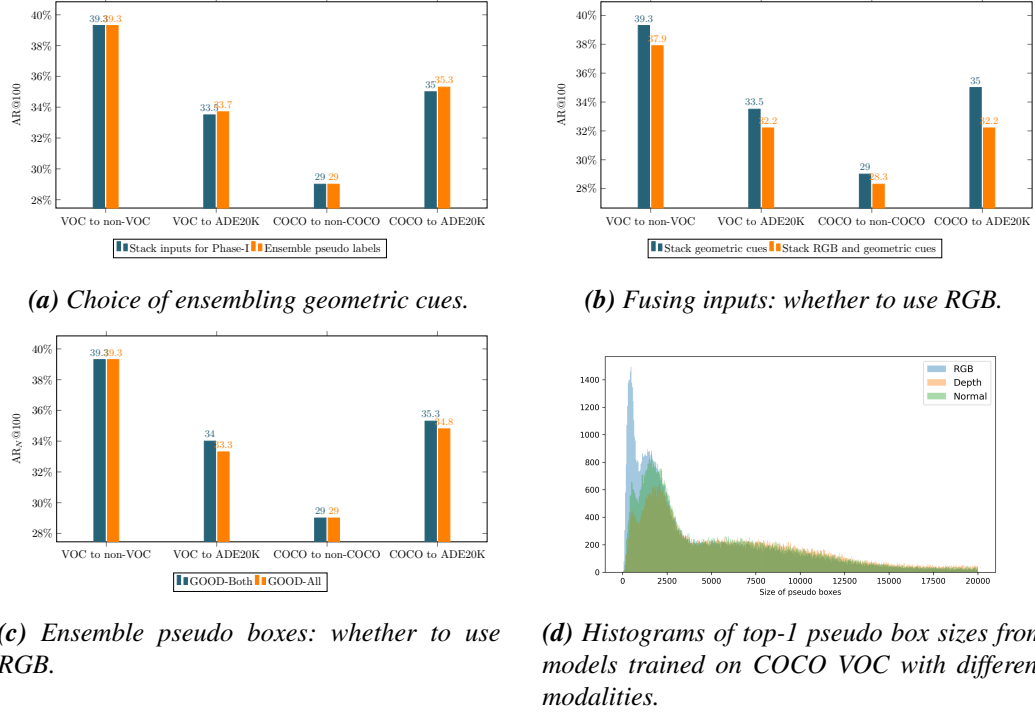
shows that data augmentation via random resizing, cropping, and flipping cannot improve generalization across categories.

Finally, we study the influence of the number of base classes. As shown in Table 3.5, we split the COCO dataset based on the supercategories to create six different splits. We then train GOOD and OLN on these splits and evaluate them on ADE20K. More base classes in training allow object detectors to learn a more generic sense of objectness so that they can better detect novel objects. As shown in Fig. 3.6c, both GOOD and OLN achieve better AR<sub>N</sub>@100 along with the number of base classes, and their performance gap reduces. However, the annotation cost also grows along with the number of classes. GOOD can achieve a similar AR<sub>N</sub>@100 as OLN with only half the number of base classes, e.g., 39 vs. 80. This shows that GOOD is more effective at learning a generic sense of objectness and less prone to overfitting to the base classes.

Beyond the main ablation studies, we conduct further experiments to understand the design choices in GOOD, particularly regarding how geometric cues can be effectively ensembled.

### Ensembling strategies for geometric cues

There are two possible ways to ensemble geometric cues: (1) Stack the two geometric cues together and train a single object proposal network on these stacked inputs in Phase-I; (2) Train two separate object proposal networks and extract pseudo boxes separately, then merge them into a single pseudo box pool for Phase-II training. Figure 3.7a demonstrates that ensembling pseudo labels is slightly better than using stacked inputs for Phase-I



**Figure 3.7: Additional ablation studies.** These experiments investigate different approaches to ensembling geometric cues and the role of RGB in Phase-I training.

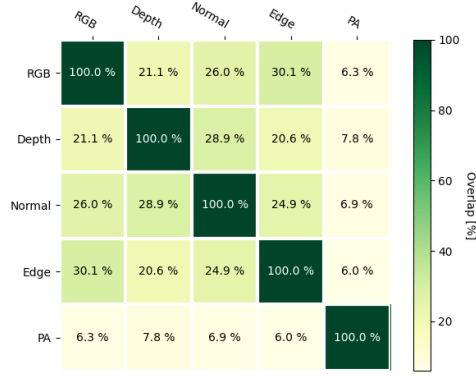
training. Throughout our experiments, we use the pseudo label ensembling approach for GOOD-Both.

### Incorporating RGB in Phase-I

It is natural to consider incorporating RGB in Phase-I training. We explore two approaches for this. If we stack RGB with geometric cues to train the proposal network, the model tends to rely more heavily on RGB, since RGB is a stronger input signal than geometric cues in the closed-world setup. This reliance on RGB prevents the model from effectively leveraging geometric cues to discover novel objects in Phase-I. As shown in Fig. 3.7b, stacking RGB with geometric cues leads to inferior performance across many benchmarks.

Alternatively, we can train a separate object proposal network on RGB inputs, extract pseudo boxes, and merge them with those from geometric cues, which we call GOOD-All. Figure 3.7c shows that adding pseudo boxes from RGB either leads to no performance gains or even worsens performance. This is because RGB-based models favor smaller detection boxes, as shown in Fig. 3.7d. These smaller boxes can be small objects or textures rather than complete objects, potentially hurting the detector’s performance on larger objects.





**Figure 3.8: Overlap heatmap of top-ranked pseudo boxes across different modalities at IoU threshold 0.7 on COCO VOC→non-VOC. The low overlap confirms that different sources discover complementary sets of novel objects.**

### Complementarity of different modalities

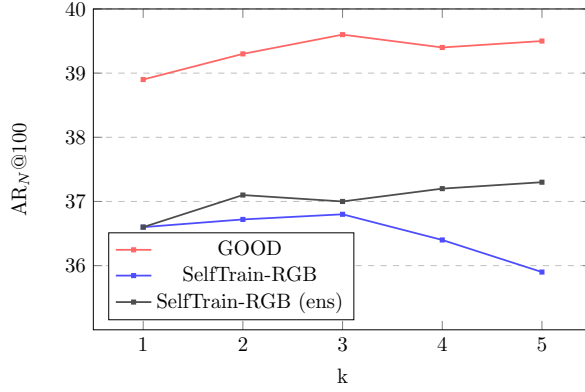
We further analyze how different modalities behave when used for Phase-I training. Pseudo-labeling using geometric cues leads to stronger performance than alternative approaches, with depth and normal maps achieving higher quality pseudo boxes (higher  $AR_N@5$ ) compared to other modality combinations.

Examining the overlap of top-ranked pseudo boxes across different modalities reveals their complementary nature. The low overlap across input types motivates ensembling different sources of pseudo boxes to exploit their diversity for training the detector. To quantify the complementarity of pseudo boxes from different sources, we compute the pairwise overlap between the top-ranked pseudo boxes generated by proposal networks trained on each modality. Figure 3.8 shows an overlap heatmap at IoU threshold 0.7: the overlap between any two sources is remarkably low (below 20%), confirming that different modalities discover largely non-overlapping sets of novel objects. This motivates a greedy ensemble strategy that selects pseudo boxes based on a combination of *utility* (the quality of each box as measured by the proposal network’s objectness score) and *uniqueness* (the degree to which the box is not already covered by previously selected sources).

Following this analysis, we adopt an ordering for the greedy ensemble: depth → normal → PA → edge → RGB, which reflects the trade-off between the quality and diversity of each source. Figure 3.9 shows the  $AR_N@100$  as we progressively add pseudo boxes from each source in this order. The monotonic improvement confirms that each additional source contributes unique information, with the geometric cues (depth and normal) providing the strongest initial contributions.

### GOOD-Both vs. GOOD-All: detailed breakdown

Table 3.6 provides a detailed AR breakdown by object size for the GOOD-Both and GOOD-All configurations. While GOOD-All includes pseudo boxes from an RGB-based



**Figure 3.9: Greedy top-k ensemble of pseudo boxes.**  $AR_N@100$  improves as pseudo boxes from additional modalities are progressively added in the order: depth  $\rightarrow$  normal  $\rightarrow$  PA  $\rightarrow$  edge  $\rightarrow$  RGB.

| Method    | Person $\rightarrow$ Non-Person |        |          |          |          | VOC $\rightarrow$ Non-VOC |        |          |          |          |
|-----------|---------------------------------|--------|----------|----------|----------|---------------------------|--------|----------|----------|----------|
|           | $AR_A$                          | $AR_N$ | $AR_N^s$ | $AR_N^m$ | $AR_N^l$ | $AR_A$                    | $AR_N$ | $AR_N^s$ | $AR_N^m$ | $AR_N^l$ |
| GOOD-Both | 33.1                            | 25.9   | 14.3     | 31.1     | 47.1     | 49.5                      | 39.3   | 21.6     | 48.2     | 62.4     |
| GOOD-All  | 33.0                            | 25.6   | 14.5     | 30.3     | 46.5     | 49.2                      | 38.9   | 22.0     | 47.3     | 61.5     |

**Table 3.6: GOOD-Both vs. GOOD-All.** Adding RGB-based pseudo boxes (GOOD-All) to the geometric ensemble (GOOD-Both) does not consistently improve performance, particularly for medium and large objects.

proposal network in addition to depth and normal, the results show that adding RGB pseudo boxes does not consistently improve performance and can even degrade detection of medium and large objects. This is consistent with our analysis in Fig. 3.7d: RGB-based models favor smaller pseudo boxes, which can introduce noise from textures rather than complete objects.

### Architecture choices

While our main results use OLN, a two-stage proposal-based detector, our approach is model-agnostic and compatible with different architectures. We demonstrate this by experimenting with FCOS [200], a more recent single-stage proposal-free object detector. Specifically, we keep Phase-I unchanged using OLN for generating pseudo boxes, but train a class-agnostic FCOS in Phase-II. Table 3.7 shows that GOOD significantly improves FCOS’s novel object detection performance, though OLN remains a stronger architecture for this task.

| Method           | VOC→Non-VOC     |                 |                              |                              |                              | VOC→ADE20K      |                              |                              |                              |
|------------------|-----------------|-----------------|------------------------------|------------------------------|------------------------------|-----------------|------------------------------|------------------------------|------------------------------|
|                  | AR <sub>A</sub> | AR <sub>N</sub> | AR <sub>N</sub> <sup>s</sup> | AR <sub>N</sub> <sup>m</sup> | AR <sub>N</sub> <sup>l</sup> | AR <sub>A</sub> | AR <sub>A</sub> <sup>s</sup> | AR <sub>A</sub> <sup>m</sup> | AR <sub>A</sub> <sup>l</sup> |
| FCOS             | 43.6            | 29.3            | 14.6                         | 35.7                         | 50.2                         | 25.6            | 17.2                         | 26.5                         | 30.9                         |
| OLN              | 47.5            | 33.2            | 18.7                         | 39.3                         | 58.6                         | 29.2            | 19.7                         | 30.7                         | 34.4                         |
| GOOD-Both (FCOS) | 48.4            | 36.3            | 18.4                         | 46.5                         | 58.0                         | 32.5            | 20.1                         | 34.5                         | 39.5                         |
| GOOD-Both (OLN)  | 49.5            | 39.3            | 21.6                         | 48.2                         | 62.4                         | 34.0            | 21.9                         | 37.0                         | 39.9                         |

**Table 3.7: Architecture choice.** *FCOS* is a single-stage proposal-free object detector, and *OLN* is a two-stage proposal-based object detector (modified from *Faster R-CNN*). *GOOD* can significantly improve the open-world performance of both architectures.

### 3.5 Conclusion

In this chapter, we proposed a method *GOOD* for tackling the challenging problem of open-world class-agnostic object detection. It exploits the easy-to-obtain geometric cues such as depth and normals for detecting unannotated novel objects in the training set. As the geometric cues focus on object shapes and relative spatial locations, they can detect novel objects that RGB-based methods cannot detect. By further involving these novel objects into RGB-based object detector training, *GOOD* demonstrates strong generalization performance across categories as well as datasets.

The geometric bridge demonstrates that invariants external to the RGB modality can dramatically improve open-world detection. The limitations of this geometric bridge—analyzed in the Reflection that follows—motivate looking beyond geometry to other structural invariants. In the next chapter, we turn to a different type of generalization gap: misalignment not in what the model *sees*, but in how the benchmark *names* what it sees.

### 3.6 Reflection: The Geometric Bridge in Retrospect

The geometric bridge was most effective for volumetric objects with distinctive 3D structure and less effective for flat or texture-defined objects, revealing a general principle: every bridging invariant has a domain of effectiveness determined by the structural property it exploits. The reliance on off-the-shelf monocular estimators further introduces a dependency on their quality and generalization. These limitations—and the broader principle that every bridge imports the biases of its source—are analyzed systematically in the Conclusion (Chapter 7).



## 4 Renovating Names in Open-Vocabulary Segmentation Benchmarks (RENOVATE)

The previous chapter demonstrated that geometric invariants can bridge the gap between known and novel object categories. We now turn to a different kind of gap: the misalignment between the *semantic interface*—category names in segmentation benchmarks—and the actual visual content. The bridging invariant here is *vision-language alignment*: the shared structure between human visual perception and the representations learned by vision-language models, which holds when objects are named according to common perceptual norms rather than arbitrary dataset conventions. By repairing this alignment, we unlock generalization that was latent in the vision-language model but inaccessible through misspecified names.

### 4.1 Introduction

Categorizations which humans make of the concrete world are not arbitrary but highly determined. In taxonomies of concrete objects, there is one level of abstraction at which the most basic category cuts are made.

— Eleanor Rosch, *Basic Objects in Natural Categories* [171]

We use names every day. Imagine wandering the trails of a national park — do you pause by a “lake” or simply a body of “water”? While driving through urban sprawl, do you see “trees” or “vegetation” along the road? Our instinctual use of terms like “lake” and “trees” illustrates the human propensity to categorize the world around us into “basic categories” — rich in information and readily identifiable without unnecessary complexity [171, 172, 190]. Such categorization is fundamental to human cognition and communication.

In stark contrast, recent advancements in open-vocabulary models [87, 60, 38, 220, 115, 65, 94, 238, 104] struggle to replicate this nuanced aspect of human categorization. These models have made strides in generalizing to both familiar and novel categories via textual prompts. However, they are often hampered by the imprecise and sometimes wrong names provided in existing benchmarks. In fact, most datasets are labeled with class names that serve merely as identifiers to distinguish classes within a dataset, rather than descriptive labels aligning with the “basic categories” that match with the actual visual contents. As shown in Fig. 4.1, existing names are often inaccurate, too general, or lack enough context, leading to discrepancies between the model’s outputs and the actual visual segments. This misalignment indicates a pressing need for a reassessment and refinement of name labeling

practices. By adopting a naming scheme that aligns more closely with human categorization, we can pave the way for enhanced open-vocabulary generalization and more accurate model evaluation.

In this work, we focus on renovating the names for open-vocabulary segmentation benchmarks, as their dense annotations pose a greater challenge and offer broader applicability than other recognition tasks. The importance of names in open-vocabulary segmentation is often overlooked, with one exception, OpenSeg [60], which manually inspected and modified the class names of several segmentation benchmarks, resulting in considerable performance gains of their proposed model. While their work demonstrated the importance of choosing good names, a manual approach is subjective and difficult to scale. To date, there exists no systematic study on how to rename benchmarks in a scalable and principled way. Our work is the first attempt to tackle this challenge.

In this work, we introduce a scalable, simple, yet general method for renaming segmentation benchmarks that outperforms manual label efforts like OpenSeg [60]. Our approach leverages foundation models to automate the renaming process, significantly reducing the manual labor traditionally required. Towards this goal, our method can be divided into two steps. First, we narrow down the name search space from the whole language space to a curated list of candidate names. This list is generated by leveraging the original class names with contextually relevant nouns extracted from visual contents via an image captioning model, thereby streamlining the search for names. Next, we employ a specially trained renaming model to identify and select the best-matching candidate name for every ground-truth segmentation mask. In this way, we match an individual name for each instance without any extra human annotations.

To demonstrate the value of RENOVATE names, we explore two practical applications. We first use our renovated names for training open-vocabulary models and show that they substantially enhance generalization performance across various benchmarks by up to 15%, indicating a more accurate alignment with visual segments. Our new names also significantly increase training efficiency thanks to their enriched textual information. Second, we apply our renovated names to improve evaluation of open-vocabulary segmentation. Specifically, we reveal that current pre-trained models are making many “benign” misclassifications, i.e., misclassified names that are semantically close to the ground-truth names. RENOVATE names enable more fine-grained model performance analysis, provid-



**Figure 4.1: Problems of names in current segmentation benchmarks.**

We demonstrate examples from well-known datasets: MS COCO [117], ADE20K [241], and Cityscapes [33]. Our renovated names are visually more aligned and help models to generalize better.

ing valuable information for further model enhancement and dataset curation. In summary, our work makes the following contributions:

- We point out the naming problem in existing open-vocabulary segmentation benchmarks.
- We propose a scalable, simple, and general framework for automatic dataset renaming.
- We demonstrate that our RENOVATE names help to train models with stronger open-vocabulary capabilities and refine evaluation benchmarks, providing valuable insights for future research.

Together, these contributions establish that principled renaming is a scalable and effective lever for improving open-vocabulary segmentation, both at training and evaluation time.

## 4.2 Related Work

**Open-Vocabulary Segmentation Methods.** Prior work in open-vocabulary segmentation [60, 38, 220, 115, 240, 15, 215, 221, 30] focused on adapting vision-language models (VLMs) [25, 193, 157] to the segmentation task without compromising pre-trained vision-language alignment. These efforts have concentrated mainly on the architecture design, exploring different backbones [218, 231], mask proposal branch designs [221, 60], and mask feature extraction techniques [220, 149, 198]. Our work differs from them in that we aim to adapt pre-trained vision-language representations for segment-wise name matching through our novel designs like candidate name generation techniques and our specialized renaming model.

**Names in Open-Vocabulary Segmentation Benchmarks.** Despite its apparent importance, the naming problem in open-vocabulary segmentation benchmarks has not received adequate attention. An exception is the work by [147], which proposes to learn class-specific word embeddings for improved model adaptation to new datasets. However, these learned embeddings are not in the language space and cannot be used by other models. In addition, some works [129, 224] propose to decompose the class names into a set of attributes or prototypes for open-vocabulary segmentation. OpenSeg [60] is the only prior work that proposes to directly improve the quality of the names. However, their approach relies on manual inspection of the datasets, which can be subjective and hard to scale. In this paper, we systematically explore the renaming challenge, culminating in a scalable, simple, yet general framework to overcome these limitations.

## 4.3 RENOVATE: Renaming Segmentation Benchmarks

In the terminology of this thesis, the bridging invariant is vision-language alignment: when names match common perceptual norms, the shared structure between human visual categorization and VLM representations remains stable across distribution shifts.

| Original name | Context names                                                                            | Candidate names                                                                                       |
|---------------|------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|
| field         | lush, field, sky, green,<br>grass, tree, road,<br>hillside, grassy, rural                | rural field, roadside field, green field, crop field,<br>sports field, grassland, grassy hillside     |
| box           | room, stool, chair, floor, market,<br>table, food, lamp, paper                           | storage box, packing box, file box, box container<br>decorative box, display box, paper box, food box |
| cradle        | room, infant bed, nursery, dresser,<br>carpet, bedroom, chair, bureau,<br>lamp, armchair | bedroom cradle, cradle, infant bed, nursery cradle,<br>baby cradle, infant cradle, wooden cradle      |

**Table 4.1: Examples of context names and generated candidate names for three selected classes from ADE20K.** Context names are key to comprehending general terms such as “field” and “box” and disambiguating polysemous terms like “cradle”, which, in this context, refers to a baby bed rather than a phone cradle or a mining tool.

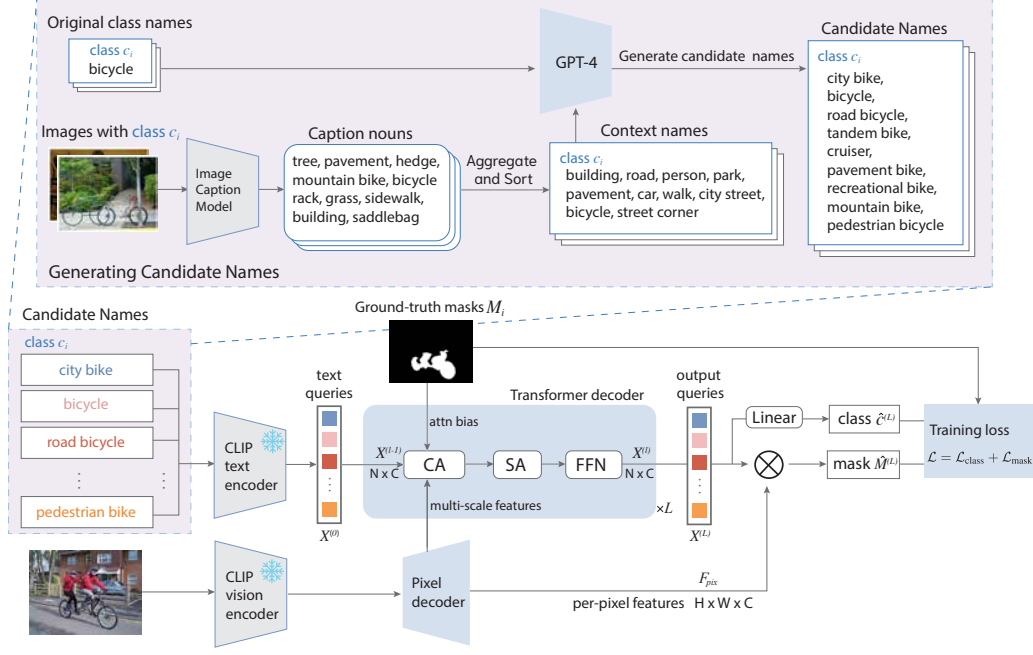
Our method RENOVATE aims to rename the original names in a given dataset with ground-truth mask annotations. As shown in Fig. 4.2, we first generate a pool of candidate names, covering potential variants of the original class names. Then, we train a specially designed renaming model that can measure the alignment between segments in pixel space and names in language space. Equipped with the renaming model, we select the top-matching names from the candidate pool for each segment, thereby producing segment-level, refined names that offer enhanced accuracy and details in characterizing the dataset. Fig. 4.3 illustrates the name selection process, and Fig. 4.4 shows how the resulting names better capture the visual content of individual segments.

#### 4.3.1 Generating candidate names

We use GPT-4 [143] for creating a pool of candidate names. A naive solution is to prompt GPT-4 with the original name and ask it to generate synonyms and hierarchical concepts. However, since the original names are often too general and ambiguous, GPT-4 does not have sufficient knowledge to generate high-quality candidates. Therefore, we propose to exploit the visual contents for generating some “context names” (Table 4.1) that assist GPT-4 in comprehending the category’s meaning prior to generating candidate names.

As shown in Fig. 4.2, for each category, we use an image captioning method to process all training images that contain that specific category based on ground-truth annotations. From the generated captions, we further extract nouns by text parsing and filtering as done in CaSED [32]. We observe that nouns appearing more frequently tend to offer deeper insights into the typical environments or traits associated with the category. Therefore, we construct *context names* by selecting the top 10 most recurrent nouns for each category. We then use them as additional inputs alongside the original class names to prompt GPT-4. Specifically, we instruct GPT-4 to generate two types of candidate names: (1) *Synonyms or subcategories*, which can expand the lexical scope of the overly general class name. For instance, the class name “grass” could yield variations like “lawn” or “turf”. For this type, synonyms and subcategories are considered together, acknowledging their often in-





**Figure 4.2: Overview of candidate name generation and renaming model training.** We generate candidate names based on the context names and train the renaming model to match them with the segments. For illustration clarity, we show only one segment. In practice, multiple segments are jointly trained, pairing with the text queries.

terchangeable nature. (2) *Providing short, distinct contexts* to polysemous names. As an example, the ambiguous name “fan” could be elaborated into “ceiling fan” or “floor fan”, offering clearer specificity. As shown in Table 4.1, the 5-10 candidate names generated by GPT-4 are more precise than the original names, reflecting the context information encoded by context names. The GPT-4 prompts used for candidate generation are detailed in the appendix of the original publication [80].

### 4.3.2 Training for candidate name selection

Among the generated candidates, we will only keep those that are better aligned with the visual contents. Particularly, we train a model that can assess the alignment between each name and segment, using segmentation as the proxy task. Conceptually, a candidate name that well describes the segment in the image should help the model recover its ground-truth mask and make a correct classification. Therefore, the model should allow vision-language interaction, and the supervision signal should encourage to use textual information for mask prediction and classification.

Our renaming model is illustrated in Fig. 4.2. Its vision part starts from a CLIP-based vision backbone followed by a pixel decoder that gradually upscales the backbone features to generate per-pixel embeddings  $F_{\text{pix}} \in \mathbb{R}^{H \times W \times C}$ . The candidate names of the categories

present in the image are processed by the CLIP text encoder. The interaction of textual and visual information takes place at the transformer decoder. The initial queries  $X^{(0)}$  are from the text embeddings, and then updated by the transformer decoder consisting of  $L$  blocks using the multi-scale visual features from the pixel decoder. As output of each block, the queries  $X^{(l)}$  generate intermediate mask and class predictions:

$$\hat{M}^{(l)} = \text{sigmoid}(X^{(l)} * F_{\text{pix}}), \quad \hat{c}^{(l)} = \text{softmax} \left[ \text{Linear}(X^{(l)}) \right] \quad (4.1)$$

where  $*$  refers to pixel-wise multiplication. The final predictions are made by the queries  $X^{(L)}$ . Our pixel decoder, mask prediction and classification follow the design of Mask2Former [27], which handles panoptic, instance, and semantic segmentation in a unified manner. The key differences lie in 1) the transformer decoder design, and 2) the training strategy, which we detail subsequently.

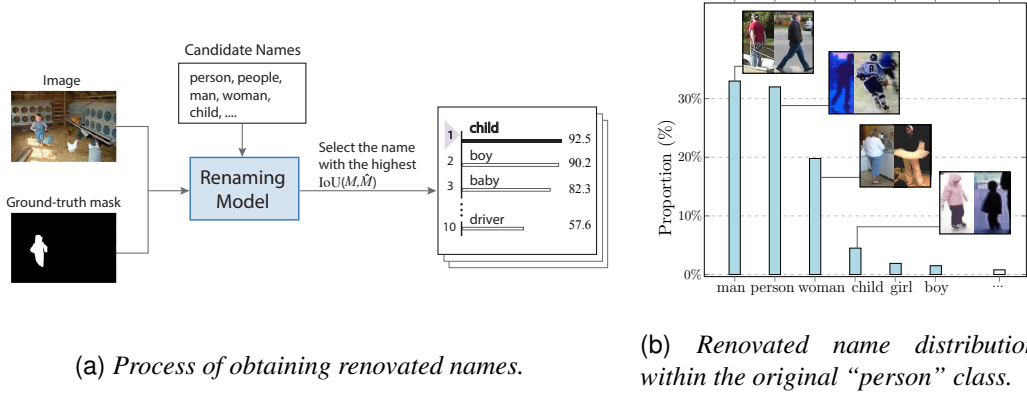
**Transformer Decoder.** Our transformer decoder uses text embeddings as input queries, unlike most open-vocabulary segmentation models that employ text embeddings solely as weights of the Linear layer in Eq. (4.1) for classification. This key change fosters an earlier integration of text and visual features at varying scales, essential for leveraging text information in segmentation tasks. As shown in Fig. 4.2, the text embedding of each candidate name from the segment’s class “bicycle” feeds into the transformer decoder, generating their corresponding mask and class predictions.

Next, within each block of the transformer decoder, the masked cross-attention (CA) layer adopts ground-truth masks as the attention biases, and the self-attention (SA) layer [204] and feed-forward network (FFN) are the same as in Mask2Former. The queries interact with the visual features at the masked CA layers, which attend within the region defined by the ground-truth mask, i.e., the attention bias. In the example of Fig. 4.2, the visual features corresponding to the segment “bicycle” can therefore contribute more effectively to refining the query. To encourage the model to rely more on the textual information in prediction, we further randomly replace the ground-truth masks with predicted masks from the preceding block in the intermediate layers. Ground-truth mask based attention biases augment the V-L interaction by focusing on the precise segment regions. Importantly, they do not reduce the task to a straightforward use of ground-truth mask inputs for mask prediction, as the output queries remain a set of vectors. The final mask prediction  $\hat{M}^{(L)}$  is based on the correlation of these output queries  $X^{(L)}$  with the feature map  $F_{\text{pix}}$  generated by the pixel decoder as in Eq. (4.1).

**Training Strategy.** With the CLIP backbone kept frozen, we train the transformer decoder together with the pixel decoder. To recover the ground-truth mask  $M_i$  and class  $c_i$ , the transformer decoder makes multiple mask and class predictions with candidate names from the class  $c_i$ . The prediction with the highest Intersection over Union (IoU) with  $M_i$  is selected for loss computation:

$$\mathcal{L}_i = \mathcal{L}_{\text{mask}}(M_i, \hat{M}_{\text{best}}) + \mathcal{L}_{\text{class}}(c_i, \hat{c}_{\text{best}}), \quad (4.2)$$

where  $\mathcal{L}_{\text{mask}}$  and  $\mathcal{L}_{\text{class}}$  are the mask localization and classification loss functions in



**Figure 4.3: Obtaining renovated names.** In (a) we illustrate how we use the renaming model to obtain a renovated name for each segment. In (b) we demonstrate that the renaming results are helpful to dataset analysis with examples from “person” class.

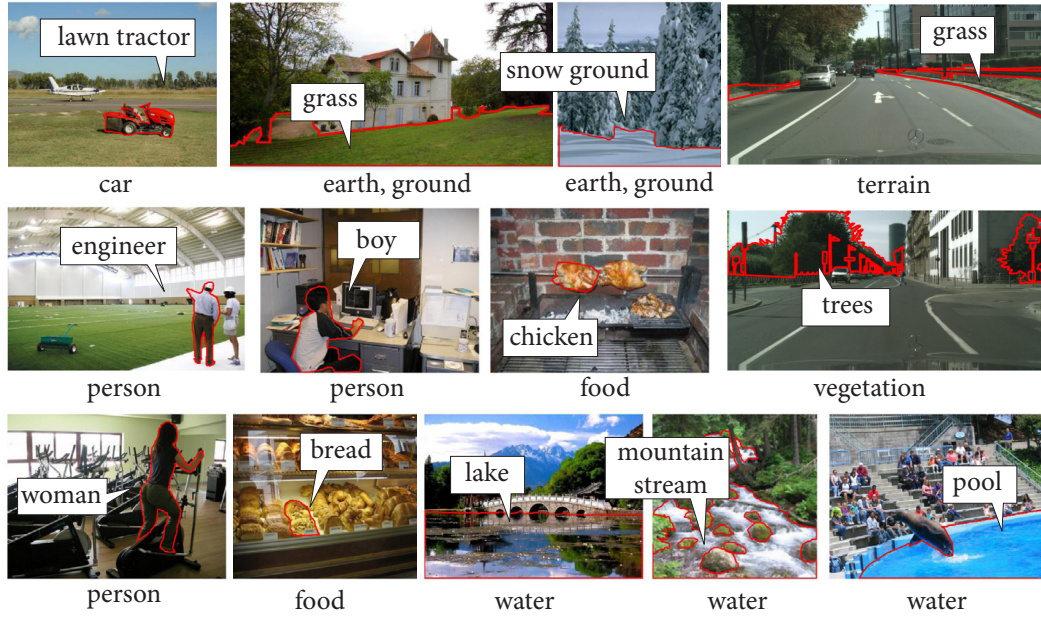
Mask2Former. To provide extra supervision on the name quality, we append the candidate names with a “negative” name randomly selected from a different category than the ground-truth class  $c_i$ . If a “negative” name scores the highest IoU, we supervise the predictions with an empty mask and the “void” class ( $c_{void}$ ) which is one extra class in addition to the training classes:

$$\mathcal{L}_i = \mathcal{L}_{\text{mask}}(\mathbf{0}, \hat{M}_{\text{best}}) + \mathcal{L}_{\text{class}}(c_{\text{void}}, \hat{c}_{\text{best}}). \quad (4.3)$$

The “void” class was also used in Mask2Former classification when the prediction mask is not matched to any ground-truth mask in the image. Having both the positive and negative supervision, we effectively incentivize the model to favor names of high quality and penalize those of low quality, thereby aiding in the accurate identification of the best-matching names for each segment.

### 4.3.3 Obtaining renovated names

As illustrated in Fig. 4.3a, to obtain the renovated names, we run the trained renaming model to associate each ground-truth mask with the candidate name. For each segment, our renaming model first ranks the names by the IoU scores between the predicted masks and the ground-truth mask, then selects the top-scoring name as the renovated name. This design assumes that a single best name exists for each segment—a simplification that may not hold when naming conventions vary across cultures, languages, or professional domains. We adopt this assumption for tractability and revisit its implications in the discussion (Section 4.6). While the IoU ranking scores are not probabilities, they are valuable to understand the relative and absolute model confidence in the names, i.e., how the chosen name stands against the other candidates and whether the selection is indeed confident. In the experiments (Section 4.5.3), we make use of this information to speed up human verification. We show some examples of renovated names in Fig. 4.4.



**Figure 4.4:** Examples of renovated names on segments from the validation sets of ADE20K and Cityscapes. For each segment, we show the original name below the image and the renovated name in the text box.

Our renovated names are readily usable for analyzing the dataset distribution and biases in a more fine-grained manner, as exemplified in Fig. 4.3b. Note that our renovated names in each original class are not necessarily mutually exclusive because different visual segments may be matched to names at different hierarchies. For example, in Fig. 4.4, a person is named as “engineer” (second row), likely due to the clothing, while the other person is named as “woman” (third row), possibly from the physical appearance. But “engineer” and “woman” are not mutually exclusive names. Exploring the hierarchical relationships among these names within a dataset remains a promising direction that could further enrich the semantic structure of the annotations.

## 4.4 Applications of RENOVATE Names

We further demonstrate the value of the renovated names with two application tasks: training stronger open-vocabulary segmentation models and improving existing evaluation benchmarks. The improvements in both tasks indicate the quality of RENOVATE names.

### 4.4.1 Training with RENOVATE names

RENOVATE names offer precise and diverse text information. We exploit this to improve model training and achieve better generalization and data efficiency. Following prior works [218, 231, 222], we replace original ground-truth names with RENOVATE names

to compute text embeddings. In addition, we employ the “negative sampling” technique, commonly used in natural language processing to enhance training with a large number of classes [135, 63, 101, 226]. For each segment, we first randomly select  $C - 1$  negative classes and choose one RENOVATE name from each. These names are then concatenated with the ground-truth RENOVATE name to form the text prompt. This effectively constructs text embeddings of length  $C$  with semantically distinct names, improving training efficiency and generalization. For computing the classification loss, we use the cross entropy function on a per-segment basis, i.e., for an image with segments  $\{m_i\}_{i=1}^N$ ,  $\mathcal{L}_{\text{class}} = \sum_{i=1}^N \text{CE}(c_i, \text{softmax}(v_i T_i^T))$ , where  $c_i$  is the ground-truth label,  $v_i \in \mathbb{R}^d$  is the predicted visual embeddings and  $T_i \in \mathbb{R}^{C \times d}$  is the constructed text embeddings through negative sampling. Throughout our experiments, we use negative sampling in our model training unless specified.

#### 4.4.2 Improving evaluation with RENOVATE names

RENOVATE names also enable more fine-grained analysis of open-vocabulary models. For instance, for a misclassification from “car” to “van”, we may distinguish between “SUV” to “minivan” versus “sedan” to “delivery van”, which isn’t possible with class-level annotations in current datasets. However, even with RENOVATE names, standard segmentation metrics obscure our desired fine-grained performance insights since they penalize all such misclassification cases equally. To this end, we adopt “open” evaluation metrics [243] that consider semantic similarity between names. Specifically, for a groundtruth mask pixel  $g_i$  with label  $c_i$  and a predicted mask pixel  $d_j$  with predicted label  $c_j$ , we compute soft  $TP_{c_i} = S_{c_i, c_j}$ , soft  $FP_{c_j} = 1 - S_{c_i, c_j}$ , and soft  $FN_{c_i} = 1 - S_{c_i, c_j}$ , where  $S_{c_i, c_j}$  is the semantic similarity between  $c_i$  and  $c_j$ . These soft metrics are then used to compute panoptic quality (PQ), Average Precision (AP), and mean IoU (mIoU) in a standard way. In Section 4.5.3, we demonstrate how open metrics effectively reveal that there are indeed many “benign” misclassification scenarios using RENOVATE names.

While our renaming process is automated, it is essential to ensure exceptionally high-quality names for evaluation. Therefore, we introduce a rigorous verification process involving five human annotators who review the top three name suggestions and their confidence scores from our model to select the best match. If none of the top three names are suitable, verifiers examine the entire candidate list. In case of disagreements, verifiers discuss and decide collectively. Note that human verification is used only for evaluation sets, while automatically generated names are used for the much larger training sets. Future work could explore using multiple vision-language foundation models to automate this verification process as evaluation sets grow [212, 146, 112].

## 4.5 Experiments

### 4.5.1 Obtaining RENOVATE names

**Setup.** We renovate three panoptic segmentation datasets respectively: COCO [117], ADE20K [241], and Cityscapes [33]. To generate context names, our default image captioning model is CaSED [32], which retrieves captions by matching vision embeddings with text embeddings of captions from a large-scale PMD dataset [189]. We train a renaming model for 60k iterations with a batch size of 16 on the training set, then generate RENOVATE names for the entire dataset.

#### GPT-4 Prompt Details

For candidate name generation, we prompt GPT-4 with the following templates. The system prompt establishes the task:

System: You are an assistant that helps rename object categories in segmentation datasets. Given an original category name and context from image captions, suggest alternative names that are more precise and descriptive.

The user prompt includes four components: (1) the original class name, (2) context names from image captioning, (3) guidelines for two types of names (synonyms and short descriptions) with examples, and (4) instructions on how to use the original name:

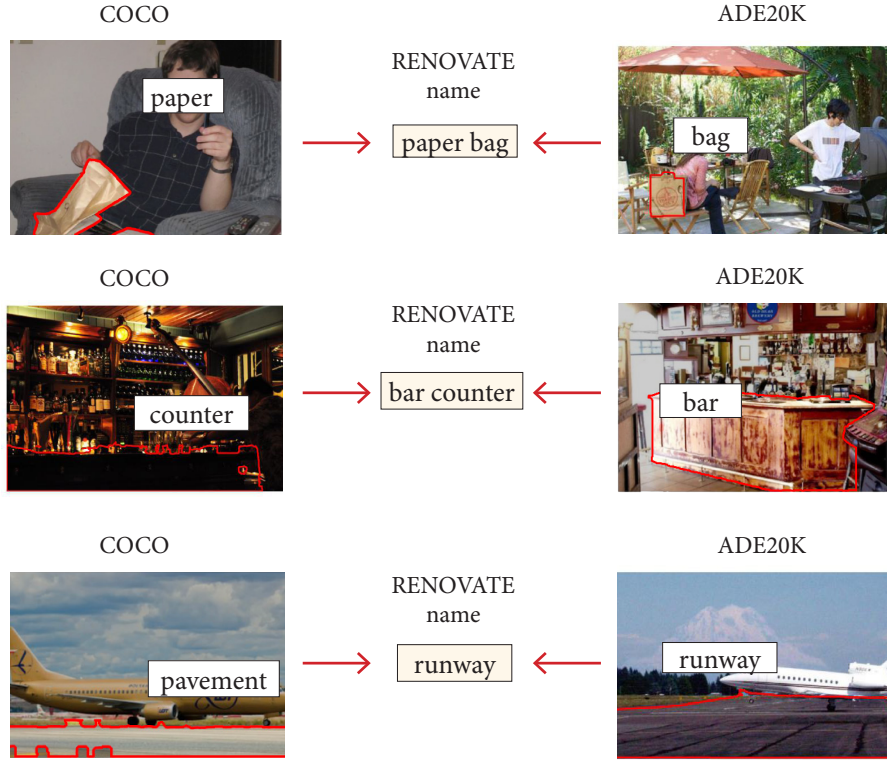
User: The original category name is ‘‘{original.name}’’. Context from image captions: {context.names}. Please suggest alternative names following these guidelines: (a) Provide synonyms that are commonly used in natural language. (b) Provide short descriptive names that add context (e.g., ‘‘dining table’’ instead of ‘‘table’’). Do not generate names that are too long or too specific. Keep the original name’s semantic scope.

|                    | COCO   | ADE   | CS    |
|--------------------|--------|-------|-------|
| # Original classes | 133    | 150   | 19    |
| # Segments/Class   | 11,274 | 2,009 | 4,964 |
| # RENOVATE names   | 741    | 578   | 108   |
| # Segments/Name    | 1,871  | 521   | 873   |

*Table 4.2: Statistics of renovated datasets.*

**Results.** Table 4.2 summarizes the statistics of RENOVATE names. Compared to the original class names, each dataset has approximately 5-6 times more distinct RENOVATE names, indicating that they provide more diverse and fine-grained semantic annotations





**Figure 4.5:** *RENOVAE uncovers segments with shared semantic concepts across datasets.*

to the visual segments. We show some visual examples in Fig. 4.4 and more in Section 4.5.7. To further validate the quality of the names, we conduct a human preference study, see Section 4.5.4. We also conduct an ablation analysis on the components of the renaming pipeline, see Section 4.5.5.

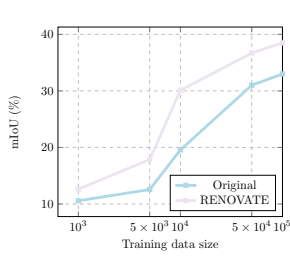
An additional benefit of RENOVAE is cross-dataset semantic alignment. A key challenge in open-vocabulary benchmarks is the inconsistency in labeling similar concepts across different datasets—for example, COCO and ADE20K may use different names for the same visual concept. Fig. 4.5 shows how RENOVAE inadvertently aids in aligning label spaces: by identifying more appropriate names for each category in natural language, shared concepts receive aligned names even when each dataset is renamed independently. This standardization facilitates cross-dataset generalization and adaptation.

#### 4.5.2 Training with RENOVAE names

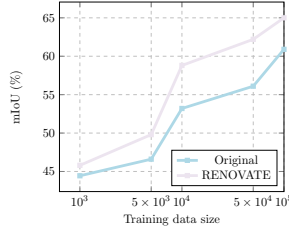
**Setup.** We train FC-CLIP [231] models on COCO with our renovated segment-matched names and compare them with models trained using other name sets on COCO, ADE20K, and Cityscapes. Specifically, “OpenSeg” names [60] were manually curated by previous

| Train names     | Source dataset |              |              | Target datasets |              |              |              |              |              |
|-----------------|----------------|--------------|--------------|-----------------|--------------|--------------|--------------|--------------|--------------|
|                 | MS COCO        |              |              | ADE20K          |              |              | Cityscapes   |              |              |
|                 | PQ             | AP           | mIoU         | PQ              | AP           | mIoU         | PQ           | AP           | mIoU         |
| Original names  | 52.70          | 42.72        | 60.91        | 25.61           | 15.97        | 33.06        | 46.06        | 23.67        | 57.99        |
| OpenSeg names   | 53.60          | 43.87        | 60.56        | 27.14           | 17.41        | 35.63        | 45.34        | 24.05        | 58.42        |
| Synonym names   | 23.38          | 35.87        | 47.20        | 15.56           | 14.78        | 29.74        | 22.94        | 19.58        | 45.91        |
| Candidate names | 41.32          | 40.96        | 56.68        | 19.80           | 15.85        | 33.43        | 39.20        | 20.79        | 51.80        |
| RENOVATE        | <b>56.62</b>   | <b>45.50</b> | <b>65.48</b> | <b>27.98</b>    | <b>17.94</b> | <b>38.50</b> | <b>46.10</b> | <b>27.99</b> | <b>61.25</b> |

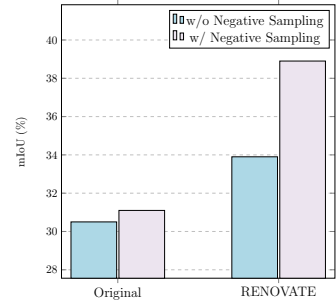
**Table 4.3: Training with renovated names.** During inference, test names are merged from Original, OpenSeg, and RENOVAE names for fair comparison. Our results demonstrate that RENOVAE names can help train stronger open-vocabulary models.



(a) MS COCO  $\rightarrow$  ADE20K.



(b) MS COCO  $\rightarrow$  MS COCO.



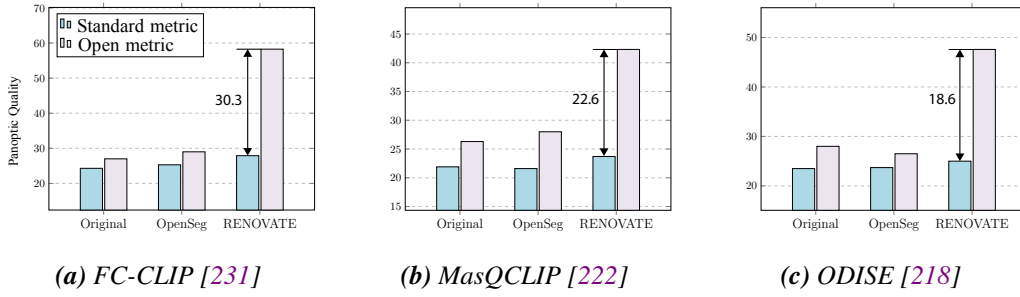
**Figure 4.7: Effectiveness of negative sampling.**

**Figure 4.6: Data efficiency comparison between RENOVAE and original names.**

researchers. “Synonym names” are generated by prompting GPT-4 based on the original class names. “Candidate names” are the outputs of the initial step in our renaming process, using image captioning to provide context for prompting GPT-4. A key difference between these names and RENOVAE names is that the former uses a set of names for a class of segments, whereas RENOVAE names provide a name for each individual segment. For model evaluation, we follow prior practices [60, 218, 231] to group predictions based on the original class divisions. For a fair comparison, our test name set merges names from Original, OpenSeg, and RENOVAE names.

**Results.** Our results in Table 4.3 demonstrate that using RENOVAE names for training improves segmentation on both source dataset and target datasets. Compared to the Original names, the model trained with RENOVAE names improves by near 4 PQ on COCO and over 5% mIoU on ADE20K. These gains significantly outperform those from OpenSeg names. This underscores the value of name refinement on per-segment level and indicates the high quality of RENOVAE names. We also note that models trained with Synonym names and Candidate names show inferior performance even to the Original names, reflecting the importance of high name quality.





**Figure 4.8: Open-vocabulary evaluation on ADE20K with different names, metrics, and models.**

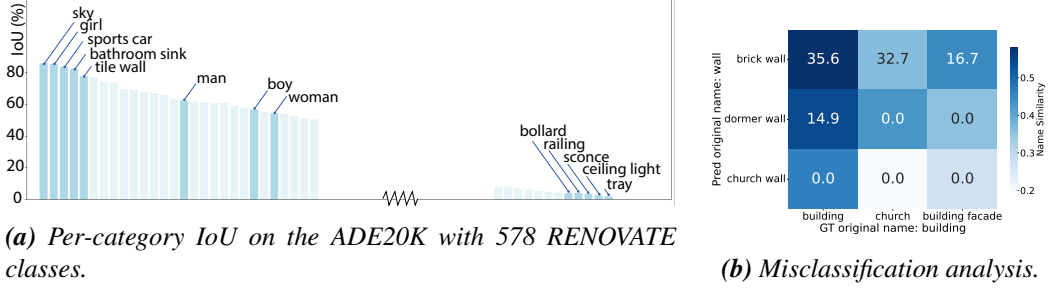
The richer text information offered by RENOVATE names also improves training efficiency. In Fig. 4.6, we show that models trained with RENOVATE names can achieve comparable performance with significantly less data compared to those trained with the Original names. For instance, models trained with  $10^4$  images using RENOVATE names can match or exceed the performance of models trained with  $5 \times 10^4$  images using the Original names. This corroborates previous findings that improving data quality can reduce the required quantity of training data [53, 64].

**Ablation on the negative sampling.** In Fig. 4.7, we study the effectiveness of negative sampling. Interestingly, we found that negative sampling is especially helpful to RENOVATE names while minorly impacting models trained with original names. This is because RENOVATE names have much more class names and the names are also more semantically close. Negative sampling effectively controls the number of classes used in cross entropy calculation and keeps the name embeddings relatively distinct, thus significantly improving training effectiveness for RENOVATE names.

### 4.5.3 Improving evaluation with RENOVATE names

**Setup.** We evaluate pre-trained FC-CLIP [231], MasQCLIP [222], and ODISE [218] models on ADE20K and Cityscapes using three different name sets with both standard and open metrics. For open metric computation, we need the similarity between the ground-truth and predicted class names. For Original and OpenSeg names, which use a set of names for a class of segments, we use the averaged pairwise name similarity between classes. For RENOVATE names, we can directly use the pairwise name similarity between renovated names. Fig. 4.8 shows PQ evaluation of pre-trained models on ADE20K. Full evaluation results, including mAP and mIoU and results on Cityscapes are provided in Section 4.5.6.

**Results.** In Fig. 4.8, we first note that the discrepancy between standard and open metrics is significantly greater when evaluated using RENOVATE names. This indicates that many misclassification cases are indeed “benign”, i.e., when evaluated using more fine-grained names, the misclassified names turned out to be semantically close to the ground-truth names. For example, when an “area rug” is predicted to be a “floor rug” (RENOVATE



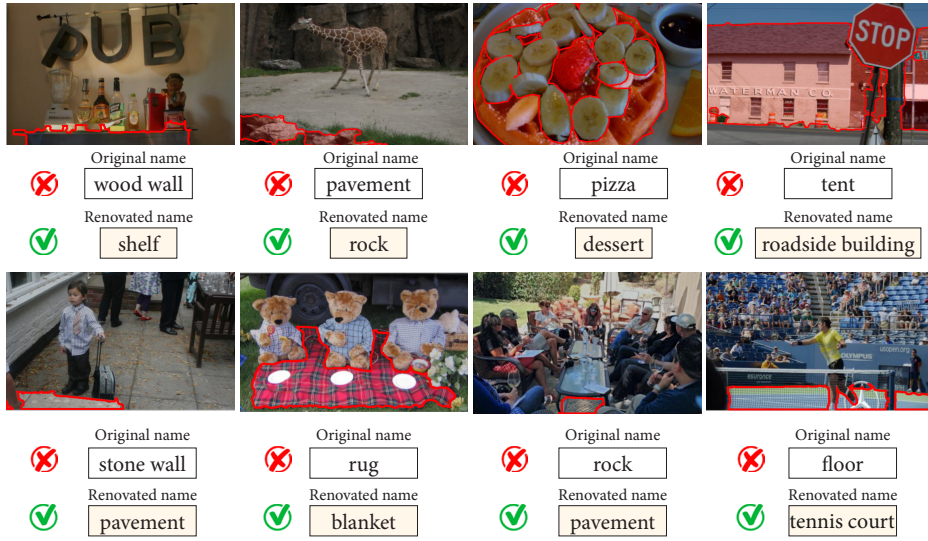
**Figure 4.9: RENOVA names enable more fine-grained analysis on models.** (a) Per-category IoU with highlighted top/bottom-5 RENOVA names and selected names from the “person” class in ADE20K. (b) Confusion matrix of frequently misclassified RENOVA names from “building” to “wall”, showing misclassification proportions (numbers) and pairwise name similarity (color).

names), it will be penalized much less than from “rug” to “floor” (original names); but if an “area rug” is predicted as “concrete floor”, the classification error will be even higher. This shows that current pre-trained models have learned relatively good semantic concepts, while still being sensitive to the text prompts at inference time.

RENOVA names also enable a more fine-grained analysis of the models. In Fig. 4.9a, we present a per-category IoU analysis on RENOVA names within the ADE20K dataset. We can see that large, well-defined objects such as “sky” and “girl” have high IoU scores, while small, deformable, and rare classes like “bollard” and “ceiling light” are more challenging for current models. Note that most of these names do not exist in the original benchmarks. In addition, our analysis reveals model biases, evidenced by the disparity in IoU scores between “man” and “woman” as well as “girl” and “boy”, suggesting an imbalance in the training dataset. In Fig. 4.9b, we further illustrate how RENOVA names facilitate a more detailed investigation of misclassifications. For example, among “building” segments misclassified as “wall”, 32.7% “churches” from the “building” class are predicted to be “brick wall” from the “wall” class. Additionally, we observe a strong correlation between name similarity and misclassifications. This showcases that our RENOVA names can help identify problematic sub-categories within the original class that models still struggle with.

Finally, we note that the open metrics are penalizing more on misclassified concepts with lower semantic similarity, e.g., “car/road” mistakes are penalized more than “car/truck” mistakes. In other use scenarios, different types of mistakes (e.g., outlier patterns) may be preferred to be penalized more and the evaluation metrics should be designed accordingly. However, regardless of the design of the evaluation metric, RENOVA names make it possible to conduct fine-grained analysis on mistakes with different semantic distances. Without RENOVA names, we can only see the coarse misclassifications without a fine-grained understanding of the model.

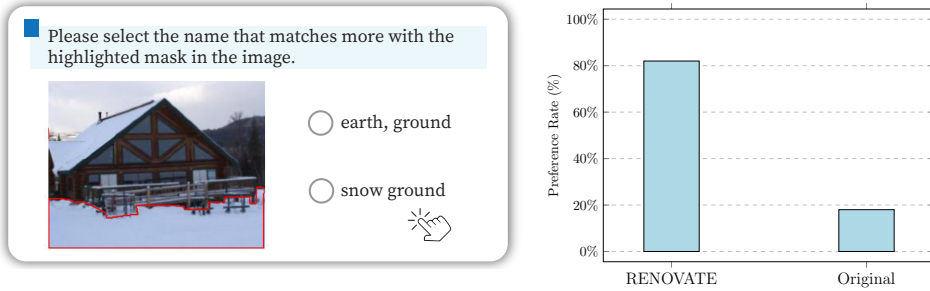
Beyond improving naming granularity, RENOVA can also discover outright annotation errors in existing datasets. In post-training evaluations, we identify segments where



**Figure 4.10:** *RENOVATE* can find wrong annotations and suggest corrections.

the renaming model consistently prefers a class name different from the ground-truth label. Visual inspection often confirms the model’s selections, revealing inaccuracies in the original annotations. Fig. 4.10 shows representative examples where *RENOVATE* finds more accurate labels from the candidate name pool. This capability arises naturally from the model’s ability to match names to visual contents.

#### 4.5.4 Human preference study



**Figure 4.11:** *Human preference study*. A survey of 20 researchers is conducted to compare preferences between the original names versus *RENOVATE* names on the validation sets. *RENOVATE* names are favored in 82% cases.

To validate the quality of *RENOVATE* names beyond automated evaluation, we conduct a human preference study on the ADE20K and Cityscapes validation sets, which contain

30,249 and 14,670 segments, respectively. We randomly select 10% of all segments in each validation set. For each segment, we ask 20 researchers unfamiliar with the project about their preference between the original name and the RENOVATE name. As shown in Fig. 4.11, RENOVATE names are preferred in 82% of cases, demonstrating a clear advantage over the original names.

#### 4.5.5 Ablation analysis on the renaming pipeline

To ablate the renaming pipeline, we leverage a pre-trained open-vocabulary model (FC-CLIP) as an automated evaluator: names are considered better if they help the model achieve better open-vocabulary segmentation when used as text prompts.

| Context name source         | ADE20K       |              |              |
|-----------------------------|--------------|--------------|--------------|
|                             | PQ           | AP           | mIoU         |
| None                        | 25.50        | 16.53        | 33.80        |
| Caption nouns (BLIP2 [110]) | 25.89        | 17.36        | 34.74        |
| Image tags (RAM [239])      | 26.95        | 17.55        | 35.54        |
| Caption nouns (CaSED [32])  | <b>27.90</b> | <b>17.88</b> | <b>37.05</b> |

**Table 4.4: Ablation on the context name sources.**

| Attn bias    | Query       | Classifier | ADE20K       |              |              |
|--------------|-------------|------------|--------------|--------------|--------------|
|              |             |            | PQ           | AP           | mIoU         |
| gt mask      | learnable   | text       | 21.09        | 14.91        | 26.96        |
| gt mask      | text        | learnable  | 25.65        | 16.33        | 32.67        |
| rand gt mask | text        | learnable  | 26.21        | 17.50        | 35.62        |
| rand gt mask | text w/ neg | learnable  | <b>27.90</b> | <b>17.88</b> | <b>37.05</b> |

**Table 4.5: Ablation on renaming model architecture.**

| GPT4 prompt                      | ADE20K      |             |             |
|----------------------------------|-------------|-------------|-------------|
|                                  | PQ          | AP          | mIoU        |
| full prompt                      | <b>27.9</b> | <b>17.9</b> | <b>37.1</b> |
| - context names                  | 25.5        | 16.5        | 33.8        |
| - suggestions on name types      | 25.6        | 16.8        | 34.0        |
| - instructions on original names | 26.5        | 17.3        | 35.2        |

**Table 4.6: Ablation on GPT-4 prompts.**

In Table 4.4, we analyze the impact of context names for candidate generation. The results show significant performance improvements when context information is provided, as opposed to the “None” baseline which relies only on original names. CaSED outperforms both BLIP2 and RAM through its retrieval-based approach and utilization of large-scale external image-caption datasets, underlining the benefits of contextual enrichment.

In Table 4.5, we analyze the design choices of the renaming model. Text queries are superior to a standard text classifier because they interact directly with spatial information from image features and exploit the mask prediction branch for more precise name-segment matching. Randomly substituting ground-truth masks with predicted ones in attention biases (“rand gt mask”) further improves matching quality, and augmenting with negative name appending (“text w/ neg”) enhances discriminative capacity.

In Table 4.6, we ablate the contribution of different components in the GPT-4 prompts. Our prompts consist of four components: (1) original name; (2) context names from image captioning; (3) suggestions on two types of names (synonyms and short contexts) to focus, with examples; (4) instructions on how to use the original name, with examples. Removing

any key component (especially (2) and (3)) significantly degrades results. The exact prompt templates are provided in Section 4.5.1.

#### 4.5.6 Full evaluation results

| Model    | Benchmark Names | Metric   | ADE20K [241] |              |              | Cityscapes [33] |              |              |
|----------|-----------------|----------|--------------|--------------|--------------|-----------------|--------------|--------------|
|          |                 |          | PQ           | AP           | mIoU         | PQ              | AP           | mIoU         |
| ODISE    | Original        | Standard | 21.88        | 13.94        | 29.16        | 39.72           | 27.73        | 49.60        |
|          |                 | Open     | 26.31        | 14.49        | 34.39        | 44.09           | 27.91        | 58.20        |
|          | OpenSeg         | Standard | 21.63        | 13.95        | 28.88        | 43.26           | 28.45        | 54.53        |
|          |                 | Open     | 28.02        | 15.32        | 38.86        | 46.66           | <b>28.88</b> | 58.61        |
|          | RENOVATE        | Standard | <b>23.69</b> | <b>14.38</b> | <b>31.64</b> | <b>43.61</b>    | 28.38        | <b>57.42</b> |
|          |                 | Open     | <b>42.32</b> | <b>50.11</b> | <b>44.65</b> | <b>53.10</b>    | <b>34.52</b> | <b>65.05</b> |
| MasQCLIP | Original        | Standard | 23.46        | 12.80        | 30.32        | 33.78           | 18.11        | 45.35        |
|          |                 | Open     | 26.53        | 13.01        | 36.33        | 37.94           | 18.35        | 55.75        |
|          | OpenSeg         | Standard | 23.70        | 12.84        | 31.17        | 35.05           | 17.79        | 46.46        |
|          |                 | Open     | 28.04        | 13.89        | 39.40        | 38.83           | 18.62        | 52.11        |
|          | RENOVATE        | Standard | <b>25.00</b> | <b>12.93</b> | <b>32.30</b> | <b>35.63</b>    | <b>18.19</b> | <b>50.07</b> |
|          |                 | Open     | <b>47.59</b> | <b>67.01</b> | <b>42.75</b> | <b>49.30</b>    | <b>37.25</b> | <b>61.44</b> |
| FC-CLIP  | Original        | Standard | 24.30        | 16.79        | 32.14        | 39.42           | 22.76        | 53.08        |
|          |                 | Open     | 27.04        | 16.52        | 36.59        | 42.57           | 26.54        | 62.69        |
|          | OpenSeg         | Standard | 25.35        | 17.30        | 33.06        | 43.68           | 26.93        | 56.15        |
|          |                 | Open     | 29.03        | 17.52        | 41.03        | 45.92           | 27.09        | 60.21        |
|          | RENOVATE        | Standard | <b>27.90</b> | <b>17.88</b> | <b>37.05</b> | <b>45.90</b>    | <b>29.79</b> | <b>62.55</b> |
|          |                 | Open     | <b>58.25</b> | <b>66.99</b> | <b>48.31</b> | <b>58.63</b>    | <b>38.91</b> | <b>70.60</b> |

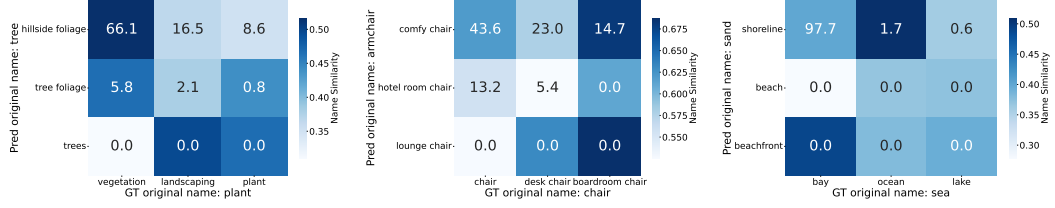
*Table 4.7: Open-vocabulary segmentation on using both standard and open metrics across different pre-trained models using different name sets.*

In Table 4.7, we present the full evaluation results of the experiments described in Section 4.5.3. Across ADE20K and Cityscapes, evaluated with PQ, AP, and mIoU, we observe the same pattern: the performance gap between standard and open metrics is significantly larger when evaluating with RENOVATE names, confirming that many misclassification cases are indeed “benign”—the misclassified names are semantically close to the ground-truth names.

We also provide more examples of fine-grained misclassification analysis in Fig. 4.12.

#### 4.5.7 More name examples

In Table 4.8, we show examples of original names, context names, candidate names, and RENOVATE names. For RENOVATE, the names for each class are the set of unique names for all the instances in that class after renaming. The context names are important for understanding vague class names such as “earth” and “field” and are helpful for providing more



**Figure 4.12: More examples of fine-grained misclassification analysis.**

context to names with multiple potential meanings like “counter” and “bar”. RENOVATE names provide more comprehensive descriptions of the corresponding classes.

In Fig. 4.13 and Fig. 4.14, we present more examples of RENOVATE names along with the original names of segments from ADE20K and COCO. RENOVATE refines the original names by being more precise and closer to natural language.

## 4.6 Conclusion and Limitations

In this work, we addressed the naming issues and showed that renaming improves both model training and evaluation of open-vocabulary segmentation. While RENOVATE uncovers model biases, it could inadvertently propagate biases from the foundational models into the new names. To mitigate potential negative societal impacts, we advocate for verification of names in critical applications, as exemplified by our verification of names in evaluation sets. As the first attempt to propose a generalized renaming framework, we acknowledge that our exploration remains incomplete. In the future, we aim to further refine our method, exploring more design choices such as other model backbones [236], and scale it up to the publicly available large-scale datasets [183, 21].

An important assumption underlying our approach deserves explicit acknowledgment: RENOVATE assumes that a single best name exists for each segment, but naming conventions vary across cultures, languages, and professional domains. The vision-language alignment we exploit reflects predominantly English-language, Western internet data, and may not generalize to all naming contexts. What counts as the “right” name depends on the audience, task, and cultural setting—a limitation inherited from the foundation models that define the semantic bridge.

**Post-publication developments.** Since the publication of RENOVATE at NeurIPS 2024, the open-vocabulary segmentation landscape has continued to evolve. Stronger vision-language backbones such as SigLIP-2 [202] and DINOv3 [188] have emerged, offering improved vision-language alignment that could both strengthen the renaming pipeline (via better candidate name evaluation) and reduce the naming gap itself (as models become more robust to name variation). Whether stronger VLMs diminish the need for renaming or amplify its benefits remains an open empirical question: even if future VLMs are more tolerant of imprecise names, the data efficiency gains from training with better-aligned names may persist. The RENOVATE framework is model-agnostic and could incorporate

| Original name                            | Context names                                                                       | Candidate names                                                                                                                                                                                           | Renovated names                                                                                      |
|------------------------------------------|-------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|
| earth, ground                            | building, tree, stand, water, person, floor, sky, road, snow, car                   | ground cover,earth floor,ground cover, ground,earth surface,snow ground, land,natural ground,outdoor ground,terrain, earth,dirt ground,terrestrial terrain                                                | ground cover, terrestrial terrain, natural ground, snow ground, dirt ground, ground, grass           |
| field                                    | lush, field, sky, green, grass, tree, road, hillside, grassy, rural                 | rural field, roadside field, green field, crop field, sports field, grassland, grassy hillside,                                                                                                           | rural field, roadside field, grassland, crop field, sports field                                     |
| rock, stone                              | stone, water, tree, lush, sea, sky, shoreline, coast, creek, stand                  | rock,stone,standing stone,beach pebble, shoreline rock,national park boulders, river stone,stones,lush stone,rocks                                                                                        | river stone, stones, beach pebble rock, national park boulders, shoreline rock                       |
| fountain                                 | city, fountain, building, garden, square, park, water, hotel, place, design         | garden fountains,fountain,urban fountains, square fountains,park fountains, water feature designs,hotel water features                                                                                    | hotel water features, urban fountains,park fountains                                                 |
| counter                                  | room, area, lobby, reception, shop, store, hotel, design, office, interior          | checkout counter,reception desk, office counter area,countertop,hotel lobby counter, cashier counter,counter,retail store counter, service desk,storefront counter,information kiosk                      | information kiosk, reception desk, cashier counter, counter, office counter area, checkout counter   |
| stairway, staircase                      | building, stair, floor, room, ceiling, lead to, pillar, balustrade, stairwell, rail | balustrade staircase,ceiling stairwell,stairway, building stairs,staircase,pillar-supported stairs, room stairway,railing stairway,floor staircase                                                        | pillar-supported stairs,railing stairway, room stairway, floor staircase                             |
| bar                                      | bar, restaurant, room, area, hotel, interior, pub, table, house, counter            | bar table,bar area,restaurant bar,hotel bar, home bar,bar counter,bar interior,bar,pub bar                                                                                                                | bar area, bar table bar counter                                                                      |
| canopy                                   | room, bed, bedroom, house, idea, design, palace, garden, home, royal                | home canopy,canopy,house canopy, garden canopy,royal palace canopy,patio canopy, bedroom canopy,room canopy                                                                                               | bedroom canopy, patio canopy, garden canopy                                                          |
| grandstand, covered stand                | stadium, game, field, arena, world, park, football, school, sport, high             | high-capacity grandstand, park bleachers, covered game stand, world-class viewing area, spectator stand, football field seating,grandstand, sports arena stand,stadium grandstand, school spectator seats | sports venue grandstand, sports arena stands, park bleachers, spectator stand football field seating |
| swimming pool, swimming bath, natatorium | pool, swimming, house, hotel, villa, resort, beach, indoor, luxury, apartment       | luxury resort pool,swimming bath,villa pool, apartment complex pool,hotel swimming pool, home swimming pool,swimming pool, beachfront swimming pool,indoor swimming bath                                  | hotel swimming pool, beachfront swimming pool, villa pool, indoor swimming pool                      |

**Table 4.8: Examples of context names, candidate names, and RENOVATE names for classes from ADE20K.**

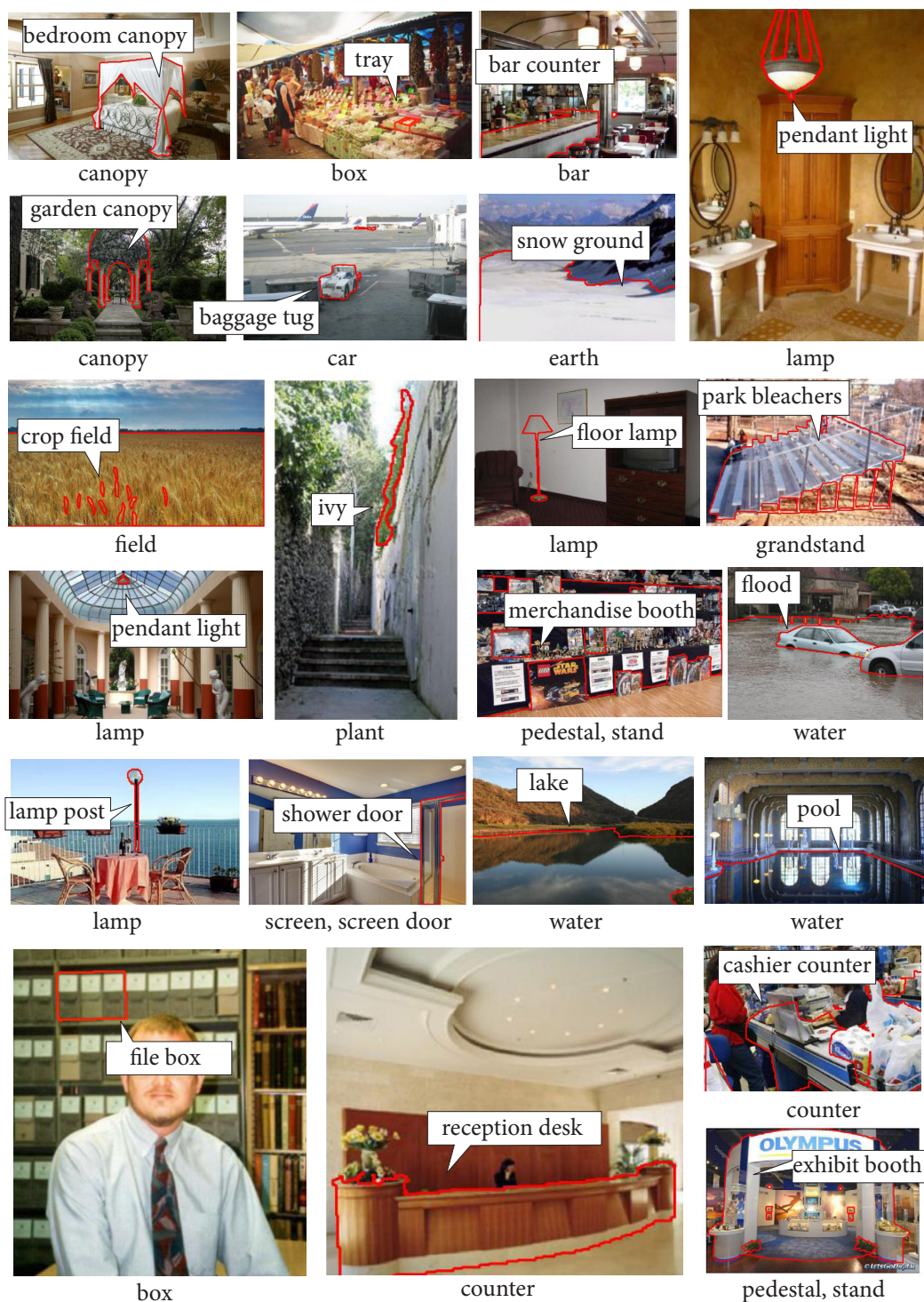
these newer VLMs as drop-in replacements for CLIP and CaSED.

The semantic bridge demonstrates that correcting the interface between model and benchmark can unlock generalization that was always latent in the vision-language representation. However, the renaming process inherits the biases of its foundation model carriers: GPT-4’s world model and CLIP’s visual-linguistic alignment are neither culturally neutral nor taxonomically complete, and these biases propagate into the renovated names. The bridge carries the assumptions of its source—a limitation we examine more broadly in the thesis conclusion. In the next chapter, we turn from what models can *recognize* to how precisely they can *operate*, addressing the spatial gap between low-resolution foundation model features and the requirements of dense prediction tasks.

## 4.7 Reflection: The Semantic Bridge in Retrospect

The semantic bridge proved remarkably effective—name quality can matter as much as data quantity—but it inherits the biases of its carriers: CLIP and CaSED reflect internet-scale, Western-centric image-text data, and the pipeline assumes segmentation masks are correct without questioning their boundaries. These limitations, and the broader principle that every bridge imports the biases of its source, are analyzed systematically in the Conclusion (Chapter 7).





**Figure 4.13: More examples of renovated names on segments from the validation set of ADE20K.** For each segment, we show the original name below the image and the renovated name in the text box.



**Figure 4.14: More examples of renovated names on segments from the validation set of COCO. For each segment, we show the original name below the image and the renovated name in the text box.**



## 5 LoftUp: Learning a Coordinate-Based Feature Upsampler for Vision Foundation Models

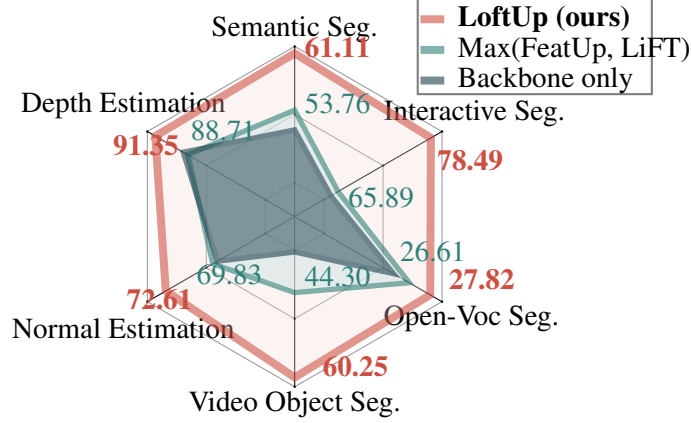
Chapters 3 and 4 addressed gaps in *what* models can recognize—novel object categories and misaligned semantic interfaces. This chapter addresses a gap in *how precisely* they can operate: the spatial gap between the low-resolution features produced by Vision Foundation Models and the high-resolution demands of dense prediction tasks. The bridging invariant is *boundary and region structure*—object boundaries and region membership are resolution-independent properties of the visual scene, holding whether features are computed at  $16\times$  downsampled or at full resolution. We exploit this invariant by constructing pseudo-groundtruth high-resolution features guided by class-agnostic masks from segmentation foundation models.

### 5.1 Introduction

High-quality pretrained representations from Vision Foundation Models (VFMs) have become standard for a wide range of computer vision tasks [45, 157, 231, 80, 145, 203, 160, 97, 162, 236, 202]. However, patchification and aggressive pooling operations in VFMs produce output features that are typically  $16\times$  or more smaller in spatial resolution than the input images. This resolution gap limits their utility for tasks that require fine-grained, pixel-level understanding.

To address the challenge of limited feature resolution in VFMs, one straightforward approach is to use larger image inputs to obtain higher-resolution features. However, processing high-resolution inputs incurs a quadratic increase in computational cost. Additionally, VFMs not trained at such resolutions can introduce severe artifacts. Moreover, training or fine-tuning VFMs on high-resolution images demands substantial computational resources and meticulous tuning [145, 160, 162, 202]. An alternative strategy is to train task-specific decoders that leverage multi-layer intermediate features to upsample VFM outputs [120, 27, 231, 80, 113]. Yet, this approach often comes with significant training costs and necessitates retraining the decoder for each new application. In addition, many downstream tasks lack sufficient data needed to fine-tune a high-quality decoder.

More recently, FeatUp [52] and LiFT [196] have independently introduced task-agnostic feature upsamplers trained with general reconstruction losses, demonstrating that such methods can substantially enhance VFM performance across a variety of tasks. By up-sampling VFM features and pairing them with a lightweight task-specific decoder, these approaches provide a promising alternative that avoids heavy, task-specific training while



**Figure 5.1:** *LoftUp improves significantly across various tasks over the VFM backbone (DINOv2-S [145]) and current SoTA feature upsampling performance (FeatUp [52] and LiFT [196]). See experiment details in Section 5.5.*

achieving a more generalizable solution. Nonetheless, as shown in Fig. 5.1, current feature upsamplers still fall short of reaching optimal performance.

Here, we systematically explore the design space of feature upsamplers and identify two critical components: the upsampler architecture and the training objective. The architecture determines the capacity of the upsampler to learn effectively, while the training objective defines the upper performance limit. By optimizing both elements, our approach achieves substantially stronger results than previous state-of-the-art methods.

Regarding the **upsampler architecture**, to capture high-resolution details while avoiding the cumulative artifacts from multiple layers of interpolation or deconvolution, we propose a simple coordinate-based transformer that directly predicts high-resolution features for each pixel. Specifically, our model takes image coordinates and RGB values as inputs and performs cross-attention with the low-resolution VFM feature map. This facilitates a fine-grained, content-aware mapping from coordinates to high-resolution features, effectively bypassing the constraints imposed by fixed local kernels and standard upsampling layers. Moreover, unlike the implicit approach in FeatUp [52], which requires test-time optimization, our method learns feature upsampling directly from the training dataset and generalizes to diverse scenes without additional test-time adjustments.

For the **training objective**, the primary challenge is the absence of groundtruth high-resolution feature annotations. Although downstream task labels such as depth or masks can be used, this approach risks compromising the task-agnostic nature of the upsampler and hinders its ability to generalize to unseen tasks. Due to this challenge, previous task-agnostic feature upsampling works, FeatUp [52] and LiFT [196], both compute the training loss at a low resolution. Our approach addresses these limitations by constructing pseudo-groundtruth (pseudo-GT) features directly at the input image resolution. Specifically, we leverage class-agnostic masks generated by off-the-shelf segmentation foundation models [97, 162] to ensure that the pseudo-GT accurately reflects the underlying geometry

and delineates object boundaries. We further refine the pseudo-GT using a self-distillation strategy that reduces noise and artifacts. This high-resolution pseudo-GT enables loss computation at a high resolution, empowering the upsampler to learn fine-grained details.

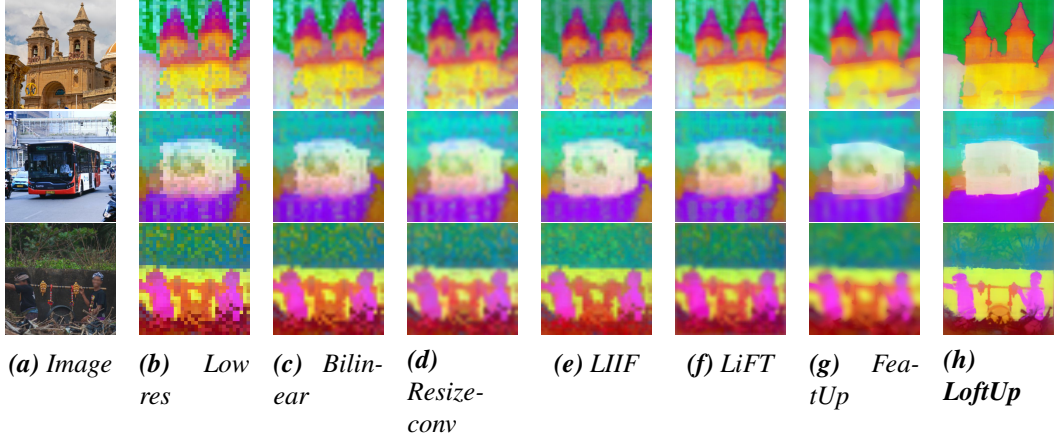
We qualitatively show that LoftUp yields markedly sharper and more detailed features than alternative upsampling methods (see Fig. 5.2). To further demonstrate the versatility and effectiveness of our approach, we evaluate it on a range of downstream tasks such as semantic segmentation, depth estimation, and video object segmentation. Fig. 5.1 summarizes these gains: our approach leads to performance improvements of 10–20% over previous SoTA upsamplers for most tasks. On video object segmentation [153], the improvement exceeds 35%. Furthermore, thanks to its coordinate-based design, our upsampler adapts seamlessly to various input and feature resolutions, catering to the diverse requirements of downstream applications. Overall, with less than a 20% increase in parameters compared to the original foundation models, our feature upsampler offers a task-agnostic, lightweight, and plug-and-play enhancement that significantly boosts VFM backbones across multiple tasks.

## 5.2 Related Work

Feature upsampling refers to increasing the spatial resolution of a feature map. Our goal is to increase the feature resolution to the original image resolution, that is, full resolution. Traditional non-learnable methods include various ways of interpolation [134, 41] and image-adaptive filtering such as joint bilateral filtering (JBU) [100] and guided image filtering [69]. In modern deep learning, previous work has proposed various architecture- and downstream-task-specific feature upsamplers. More recently, with the success of vision foundation models such as DINOv2 [145] and CLIP [157], there is a trend for feature upsamplers to be downstream-task-agnostic so that they can be used with the VFM backbone together in various applications [52, 196]. Our work falls into this category.

**Architecture for Feature Upsamplers.** Inspired by coordinate-based methods in 3D reconstruction [229, 116, 227, 197], we adopt a coordinate-based approach and view feature upsampling as a mapping from high-resolution coordinates to high-resolution features. Our method requires no test-time optimization and enables global interactions between image inputs and low-resolution features through a cross-attention mechanism. This combination of coordinate-based design and global attention yields stronger upsampling performance than prior architectures.

**Task-agnostic Training Objective.** FeatUp [52] and LiFT [196] were the first two works to propose a task-agnostic training pipeline for feature upsamplers. However, their training is at a low resolution, leaving the high-resolution features severely under-constrained. In this chapter, we propose a self-distillation approach to generate full-resolution pseudo-GT, which is then used to supervise feature upsampling training at full resolution.



**Figure 5.2: Comparison of features from upsamplers.** Backbone is DINOv2-S/14 [145].

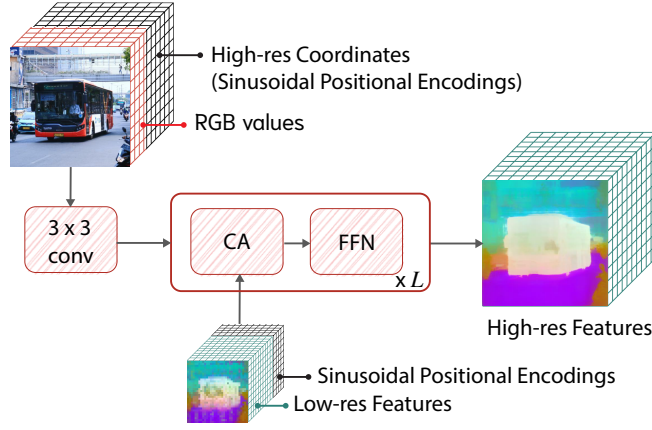
### 5.3 Coordinate-Based Feature Representation

Prior works typically address feature upsampling as a gradual resolution enhancement process, where features are progressively lifted to a higher resolution. We adopt a coordinate-based representation of the high-resolution features and view feature upsampling as a mapping from a pixel coordinate  $(x, y)$  to the high-resolution features at that pixel. Specifically, we propose a cross-attention mechanism to effectively incorporate the low-resolution features and the high-resolution image inputs with coordinates.

As shown in Fig. 5.3, following prior work on 3D coordinate-based reconstruction [229, 136], our model encodes full-resolution coordinates with sinusoidal positional embeddings and concatenates them with RGB values. A convolutional layer then projects this combined input into the feature dimension. Next, an  $L$ -block cross-attention transformer uses these high-resolution features as queries and low-resolution VFM features as keys and values, producing the final high-resolution feature map.

Our cross-attention design enables high-frequency details to interact globally with semantically rich representations, facilitating global-content-aware upsampling. This overcomes the limitations of fixed or locally predicted kernels in prior feature upsamplers. As illustrated in Fig. 5.2, multi-layer upsamplers such as resize-conv [142], LiFT (U-Net) [196], and FeatUp (a modified JBU) [52] tend to produce blurry outputs with artifacts. Additionally, LIIF [26] employs a coordinate-based design but relies on local interactions, which limits its upsampling quality. Our LoftUp leverages global cross-attention to accurately capture object boundaries and produce fine-grained feature maps with minimal artifacts.

Furthermore, our coordinate-based approach allows flexible generation of feature maps at arbitrary resolutions by adjusting the input coordinate resolution—unlike traditional upsampling methods, which are restricted to fixed scaling factors due to their multi-layer structures. Finally, we note that while attention mechanisms can be computationally expensive, our cross-attention remains relatively efficient because it processes a much smaller set of low-resolution tokens as keys and values, rather than the progressively higher-resolution



**Figure 5.3: Architecture of LoftUp.** Our coordinate-based network with cross-attention mechanism effectively integrates the fine-grained details from image RGB values and semantically-rich low-res features to produce high-resolution feature maps.

feature maps as in multi-layer upsampler architectures. In fact, as our experiments (Table 5.7) later show, our upsampler’s inference speed is comparable to bilinear upsampling.

## 5.4 Training Objective

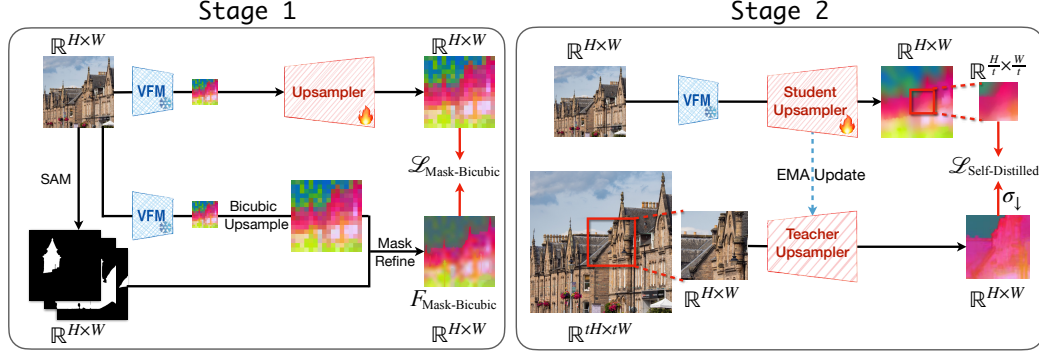
While the architecture of the upsampler establishes its capacity, the training objective ultimately sets the performance ceiling. A central challenge for training a task-agnostic feature upsampler is the absence of full-resolution groundtruth features. As a result, previous work has either relied on proxy tasks or constructed pseudo-GT at lower resolutions. For instance, FeatUp [52] employs a multi-view reconstruction task: the predicted high-resolution features,  $\hat{F}_{\text{HR}}$ , are first transformed via an affine mapping  $t$ , then downsampled, and finally compared against the low-resolution features extracted from images that go through the same transformation. Here,  $\mathcal{D}$  denotes a generic discrepancy measure between feature maps (the specific choice varies across methods):

$$\mathcal{L}_{\text{FeatUp}} = \mathcal{D}\left(f(t(I)), \sigma_{\downarrow}(t(\hat{F}_{\text{HR}}))\right).$$

LiFT [196] takes a different approach, directly using features from  $2\times$  larger images  $I_{2\times}$  as pseudo-GT, constraining the upsampler to predict  $2\times$  upsampled features  $\hat{F}_{2\times}$  via

$$\mathcal{L}_{\text{LiFT}} = \mathcal{D}(\hat{F}_{2\times}, f(I_{2\times})).$$

However, since both  $\mathcal{L}_{\text{FeatUp}}$  and  $\mathcal{L}_{\text{LiFT}}$  are at relatively low resolutions (only 1/16 or 1/8 of the target resolution), they provide only weak supervisory signals for capturing fine-grained details inherent in high-resolution outputs, potentially leaving the upsampled features under-constrained.



**Figure 5.4: Our two-stage LoftUp training approach.** Stage 1 trains an upsampler with class-agnostic masks to refine bicubic-upsampled features. Stage 2 employs self-distillation, initializing teacher and student upsamplers from Stage 1’s pre-trained model. All VFM image inputs share the resolution ( $H \times W$ ). For visual clarity, the VFM block is omitted from Stage 2’s teacher branch.

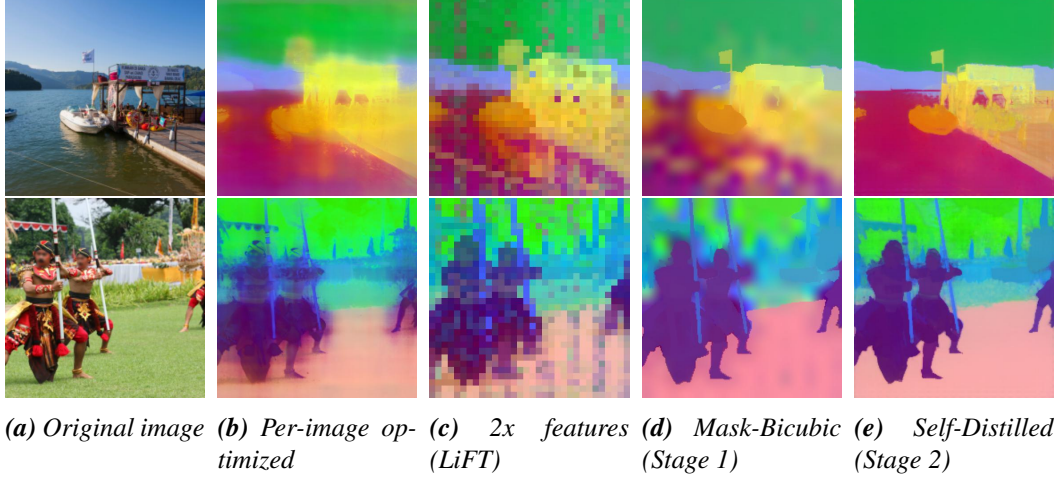
We employ a self-distillation strategy to generate high-quality pseudo-GT for supervising full-resolution features, as shown in Fig. 3.3. First, we train an upsampler using high-resolution class-agnostic masks, which emphasize sharp boundaries and geometric awareness. In the second stage, we enhance training through self-distillation. Specifically, we first initialize a teacher and a student upsampler both from the trained upsampler in Stage 1. Then, the teacher processes high-resolution image crops, and its outputs serve as supervision for the corresponding student upsampler outputs. By distilling more detailed and accurate feature maps from the teacher, Stage 2 further reduces noise and enhances sharpness in the student’s outputs. Overall, by constructing high-quality pseudo-GT feature maps at full-resolution, our approach enables the upsampler to capture more detailed structures and boundaries, ultimately pushing the limits of feature upsampling.

#### 5.4.1 Stage 1: Training with class-agnostic masks

Low-resolution features and their upsampled features using existing upsamplers are often too blurry and noisy to be used as pseudo-GT. Segment Anything Model (SAM) [97] offers a stronger alternative: it produces full-resolution, class-agnostic masks that capture fine-grained details such as small object parts and boundaries. In the terminology of this thesis, the bridging invariant is boundary structure: object boundaries are resolution-independent properties that hold whether features are coarse or fine. Previous work has shown that training with these masks leads to strong task generalization [97, 162, 161, 165]. These qualities make them well suited for refining feature maps to reduce artifacts and noise while promoting smoothness in homogeneous regions.

To leverage these masks, we first upsample a low-resolution feature map to full resolution via bicubic interpolation, yielding  $F_{\text{Bicubic}}$ . Then, for each mask  $m \in M = \{m_1, m_2, \dots, m_N\}$ , we compute the mean feature,  $\overline{F_{\text{Bicubic}}[m]}$ , and blend it with the original





**Figure 5.5: Visualization of different pseudo-GT.** Both Mask-Bicubic and Self-Distilled are proposed by our work. We set  $\alpha = 0.8$  (in Eq. (5.1)) to balance sharp boundaries from masks and fine-grained details from high-res features.

features, yielding a *mask-refined* feature map at pixels within mask  $m$ :

$$F_{\text{Mask-Bicubic}}[m] = \alpha * \overline{F_{\text{Bicubic}}[m]} + (1 - \alpha) * F_{\text{Bicubic}}[m], \quad (5.1)$$

where  $\alpha \in [0, 1]$  controls the degree of mask refinement.

We then supervise the upsampler using the loss:

$$\mathcal{L}_{\text{Mask-Bicubic}} = \|\hat{F}_{\text{HR}} - F_{\text{Mask-Bicubic}}\|_2, \quad (5.2)$$

which encourages high-resolution outputs with homogeneous features within each mask region, thereby leveraging the rich structural information from the class-agnostic masks. Note while these masks can also refine features from more complicated upsampling methods such as FeatUp [52] or JBU [100], we observe that simple bicubic upsampling already yields strong results, making it the preferred choice for Stage 1 training.

| Metrics                 | Depth map | Per-image implicit | 2x features (LiFT) | Mask-Bicubic | Self-Distilled |
|-------------------------|-----------|--------------------|--------------------|--------------|----------------|
| High-Res Loss           | ✓         | ✓                  | ✗                  | ✓            | ✓              |
| Task-Agnostic           | ✗         | ✓                  | ✓                  | ✓            | ✓              |
| Geometry Fidelity       | ✓         | ✓                  | ✗                  | ✓            | ✓              |
| Free of Noise/Artifacts | ✓         | ✗                  | ✗                  | ✗            | ✓              |
| Scalability             | ✓         | ✗                  | ✓                  | ✓            | ✓              |
| COCO Seg. (IoU)         | 56.02     | 59.08              | 52.90              | 59.87        | <b>61.11</b>   |
| Cityscapes Seg. (IoU)   | 44.82     | 46.05              | 35.85              | 50.43        | <b>53.10</b>   |

Table 5.1: Comparison of different pseudo-GT choices.

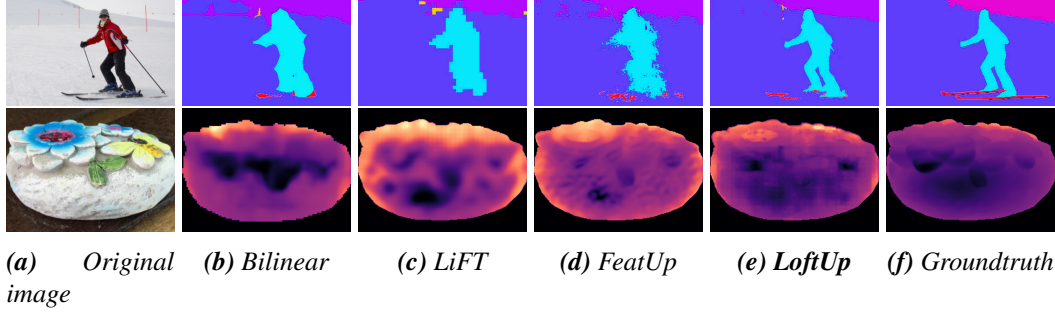
#### 5.4.2 Stage 2: Training with self-distillation

We further enhance our upsampler training using self-distillation. To distill high-quality features, we adopt a teacher-student dual-branch design. Both the teacher and student up-samplers are initialized with the upsampler trained in Stage 1. The teacher is updated via an Exponential Moving Average (EMA) during training, ensuring a stable and progressively improving target.

For the student branch, each image is resized to a fixed resolution  $I \in \mathbb{R}^{H \times W}$  (224px or 336px), matching the VFM’s input requirements. For the teacher branch, the original image is resized to a larger size  $I_{\text{HR}} \in \mathbb{R}^{tH \times tW}$  with  $t \in [2, 4]$ . The models then process on a crop from  $I_{\text{HR}}$  that also matches the VFM’s resolution, i.e.,  $\text{crop}(I_{\text{HR}}) \in \mathbb{R}^{H \times W}$ . Note our original images, sourced from the SA1B dataset [97], have a minimum resolution of 1500px on the shortest side and are thus larger than  $tH \times tW$ . While a higher  $t$  yields more fine-grained teacher outputs, it also lowers the supervision resolution on the student ( $H/t \times W/t$ ), reducing supervision effectiveness. In practice, we find  $t \in [2, 4]$  strikes a good balance. The teacher’s output on this crop,  $f_{\text{teacher}}(\text{crop}(I_{\text{HR}})) \in \mathbb{R}^{H \times W}$ , is subsequently downsampled via  $\sigma_{\downarrow}(\cdot)$  to match the resolution of the corresponding student output,  $\text{crop}(f_{\text{student}}(I)) \in \mathbb{R}^{H/t \times W/t}$ , and used as pseudo GT for the student. Formally, the self-distillation loss is defined as:

$$\mathcal{L}_{\text{Self-Distilled}} = \mathcal{D}\left(\sigma_{\downarrow}(f_{\text{teacher}}(\text{crop}(I_{\text{HR}}))), \text{crop}(f_{\text{student}}(I))\right),$$

where  $\mathcal{D}$  denotes a discrepancy function between features. In our experiments, we adopt the affinity matrix loss [213, 156, 210], which consistently outperforms the standard  $L_2$  loss. This self-distillation strategy leverages the teacher’s ability to handle an easier task—generating high-quality features for cropped regions—thus providing a more reli-



**Figure 5.6: Visualization of prediction examples on semantic segmentation on COCO-Stuff [17] and depth estimation on NAVI [86].**

able pseudo-GT to guide the student. Furthermore, to capitalize on the sharp boundary priors provided by the class-agnostic masks, we apply the same mask refinement described in Eq. (5.1) to the teacher’s features. For simplicity, we denote this refined output as  $F_{\text{Self-Distilled}}$ . As shown in Fig. 5.5,  $F_{\text{Self-Distilled}}$  reduces the blurry artifacts in  $F_{\text{Mask-Bicubic}}$  and captures the underlying geometry much better.

**Further Discussion on Pseudo-GT Choices.** Earlier in this section, we identified low-resolution loss as a bottleneck in prior training objectives. Here, we expand on that discussion by addressing additional key factors uncovered during our pseudo-GT exploration. In particular, Table 5.1 compares our approach with three alternative pseudo-GT choices: (1) depth maps, (2) per-image optimized high-resolution features from FeatUp-Implicit [52], and (3) features from  $2\times$  larger inputs, as in LiFT [196]. First, pseudo-GT should be **task-agnostic** to ensure generalizability across downstream tasks. For example, although depth maps contain fine-grained geometric details, they lack semantic information, potentially limiting the upsampler’s ability to maintain global semantic consistency. Furthermore, pseudo-GT must **capture image geometry accurately**, maintain sharp boundaries and clear color transitions, and **minimize artifacts and noise**. As shown in Fig. 5.5, per-image optimized features exhibit halo artifacts around objects and speckle noise in smooth regions, while  $2\times$  features introduces significant mosaic artifacts.  $F_{\text{Mask-Bicubic}}$  still suffers from blurriness and blobby artifacts inherited from bicubic upsampling. Among these, only  $F_{\text{Self-Distilled}}$  meets these requirements, demonstrating the effectiveness of our self-distillation framework. Finally, **scalability** is another crucial factor. Per-image optimization is computationally expensive, requiring 1 minute per HR feature map on an A100 GPU and 74MB for storage —equating to 35 GPU days and 3.5TB storage for just 50K images. The remaining methods generate pseudo-GT in real time with a single forward pass, taking less than 0.1 seconds per image. Overall, we find that  $F_{\text{Self-Distilled}}$  best satisfies these requirements and delivers the strongest performance.

| Method        | Semantic Seg.        |                    | Depth Estimation  |                   | Normal Estimation |                   |
|---------------|----------------------|--------------------|-------------------|-------------------|-------------------|-------------------|
|               | COCO mIoU $\uparrow$ | CS mIoU $\uparrow$ | RMSE $\downarrow$ | Recall $\uparrow$ | RMSE $\downarrow$ | Recall $\uparrow$ |
| Low-res       | 51.21                | 36.54              | <u>0.1071</u>     | 89.08             | 32.29             | 69.56             |
| Bilinear      | 56.15                | <u>44.79</u>       | 0.1132            | 87.68             | <u>32.27</u>      | <u>70.03</u>      |
| LiFT          | 53.35                | 35.80              | 0.1078            | 88.71             | 32.31             | 69.78             |
| FeatUp        | <u>56.30</u>         | 44.19              | 0.1092            | 88.57             | 32.25             | 69.83             |
| <b>LoftUp</b> | <b>61.11</b>         | <b>53.10</b>       | <b>0.0921</b>     | <b>91.35</b>      | <b>30.79</b>      | <b>72.76</b>      |

**Table 5.2: Comparison of feature upsamplers across tasks of semantics segmentation, depth estimation, and normal estimation.** For each metric, we boldface the best performance and underline the second best.

| Method        | Video Obj. Seg. |              |              | Open-Voc Seg. |              |              | Interactive Seg. |              |              |
|---------------|-----------------|--------------|--------------|---------------|--------------|--------------|------------------|--------------|--------------|
|               | J Mean          | F Mean       | Avg          | COCO*         | CS*          | ADE*         | GrabCut*         | Berkeley*    | DAVIS*       |
| Low-res       | 42.05           | 31.27        | 36.66        | 25.70         | 34.48        | 19.50        | 64.90            | 55.77        | 52.82        |
| Bilinear      | 42.62           | 33.90        | 38.26        | 25.78         | 34.56        | 19.53        | 65.04            | 55.83        | 54.26        |
| LiFT          | <u>47.68</u>    | 36.78        | 42.23        | 25.96         | <u>35.39</u> | 19.50        | 29.66            | 31.99        | 41.14        |
| FeatUp        | 45.70           | <u>42.90</u> | <u>44.30</u> | <u>26.61</u>  | 35.01        | <u>19.99</u> | <u>65.89</u>     | <u>56.67</u> | <u>55.03</u> |
| <b>LoftUp</b> | <b>58.72</b>    | <b>61.79</b> | <b>60.25</b> | <b>27.82</b>  | <b>38.82</b> | <b>21.29</b> | <b>78.49</b>     | <b>65.24</b> | <b>67.31</b> |

\* Dataset references: COCO [117], Cityscapes [33], ADE [241], GrabCut [173], Berkeley [131], DAVIS [150].

**Table 5.3: Comparison of feature upsamplers across tasks of video object segmentation, zero-shot open-vocabulary segmentation, and interactive segmentation.** Following previous works [105, 80, 120, 97, 84, 196], we report J Mean, F Mean, and their average for video object segmentation, mIoU for open-vocabulary segmentation, and IoU@1 Click for interactive segmentation.

## 5.5 Experiments

In this section, we compare LoftUp with other upsampling approaches on various tasks and conduct in-depth analysis on the strengths of LoftUp. By default, we use the DINOv2-S/14 model [145] as VFM. Experiments with CLIP [157] and RADIO [160] show that DINOv2 produces the best downstream performance. Following previous works [52, 196], we resize input images to a resolution of 224 and upsample the VFM features to match the input resolution ( $14\times$  for DINOv2-S/14). All upsamplers are trained for 1 epoch on a 1M-image subset of the SA1B dataset [97]. Our LoftUp upsampler is a 2-block cross-attention transformer trained with AdamW [124] (learning rate  $10^{-3}$  in Stage 1,  $10^{-4}$  in Stage 2; batch size 8). In Stage 2, we take 2 random crops per image and update the teacher upsampler’s weights every 10 steps using EMA with a decay factor of 0.99. We use  $\alpha = 0.8$  for mask refinement when constructing pseudo-GT.

| Architecture  | COCO         | Cityscapes   | Depth Rec.   | Normal Rec.  |
|---------------|--------------|--------------|--------------|--------------|
| Low-res       | 51.21        | 36.54        | 89.08        | <u>69.56</u> |
| Resize-conv   | 56.05        | <u>45.94</u> | 88.38        | 65.59        |
| LIIF          | 52.24        | 40.36        | 79.85        | 46.06        |
| FeatUp-JBU    | <u>57.90</u> | 44.83        | <u>89.38</u> | 68.83        |
| <b>LoftUp</b> | <b>61.11</b> | <b>53.10</b> | <b>91.35</b> | <b>72.61</b> |

**Table 5.4: Architecture comparison on semantic segmentation and depth and normal estimation. All upsamplers are trained using LoftUp training objective.**

### 5.5.1 Cross-Task Comparison

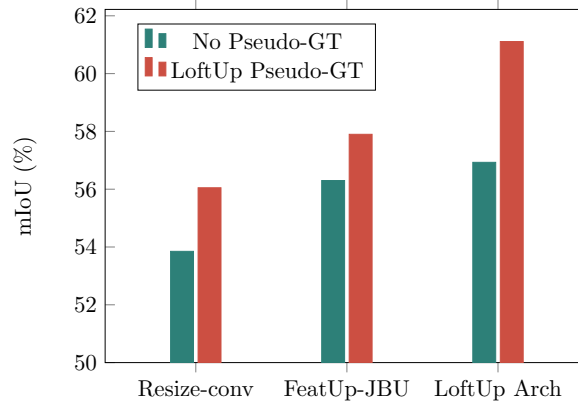
We compare LoftUp with two most recent task-agnostic feature upsamplers: FeatUp (JBU) [52] and LiFT [196], along with bilinear upsampling as a baseline. As demonstrated in Table 5.2 and Table 5.3, LoftUp consistently outperforms all alternatives on six downstream tasks.

Since we expect high-resolution features to *enhance fine-grained scene understanding*, we first evaluate them on **semantic segmentation** following [52, 67]. We train a linear projection on upsampled features to predict coarse classes in COCO-Stuff [17] and Cityscapes [33]. LoftUp achieves 7.3% and 15.6% relative improvements over previous best methods. As expected, Cityscapes benefits more from feature upsampling due to its high number of small objects.

To assess the upsamplers’ *generalizability to geometry prediction*, we follow [45] and evaluate **depth and normal estimation** using a lightweight DPT decoder head [158] on the NAVI dataset [86]. Among the compared methods, LoftUp is the *only* upsampling method that significantly outperforms the low-res baseline, achieving 2.5% and 4.4% relative improvements in recall.

We further evaluate the *temporal consistency* of upsampled features through **video object segmentation** on DAVIS2017 [153], which requires consistent object tracking across video frames. Following prior works [196, 84], we use feature affinity maps to track objects across frames and report J Mean (mIoU, reflecting region similarity), F Mean (mean F-score, reflecting contour accuracy) and their average. LoftUp achieves significant performance improvements, with a 39.6% increase in J Mean and a 97.6% increase in F Mean over the low-resolution baseline, demonstrating a strong ability to identify object boundaries. We visualize qualitative tracking results in Fig. 5.9.

Finally, to assess the *flexibility of usage across modalities* like text and click embeddings, we evaluate on **zero-shot open-vocabulary segmentation** with ProxyCLIP [105] where DINOv2 features adjust CLIP features spatially for better vision-language alignment, and on **interactive segmentation** using a modified SimpleClick [120] where visual features are incorporated with click embeddings for segmentation. Results show that LoftUp is the *only* method to achieve consistent, significant improvements over the low-res and bilinear baselines, demonstrating its adaptability to multimodal alignment.



**Figure 5.7: LoftUp pseudo-GT improves different upsamplers.** Here we report semantic segmentation results on COCO.

### 5.5.2 More Analysis of LoftUp Design

**Our pseudo-GT provides strong training signal.** In Fig. 5.5 and Table 5.1, we already discuss and compare different pseudo-GT choices both qualitatively and quantitatively by training LoftUp architecture using different pseudo-GTs. In Fig. 5.7, we further show that our pseudo-GT can be used to enhance other upsamplers such as resize-conv and FeatUp (JBU), demonstrating broad applicability of the strong, full-resolution supervision signal of our pseudo-GT alone.

**Our architecture significantly outperforms other alternatives.** when training under the same LoftUp training objective as in Section 5.4. As shown in Table 5.4, we compare LoftUp with resize-conv, LIIF, and FeatUp-JBU. We demonstrate that while other methods may bias towards certain tasks, LoftUp achieves the strongest performance across all of the tasks, indicating its stronger capacity to leverage the high-quality pseudo-GT. We further provide ablation studies of our architecture in Table 5.5.

|     | Feat PE   | Image conv | # blocks | # Train data | COCO         | Cityscapes   |
|-----|-----------|------------|----------|--------------|--------------|--------------|
| (a) | no        | 1x1        | 2        | 50k          | 56.46        | 43.13        |
|     | learnable | 1x1        | 2        | 50k          | 57.89        | 46.06        |
|     | learnable | 3x3        | 2        | 50k          | 58.40        | 47.35        |
|     | Sine      | 1x1        | 2        | 50k          | 58.10        | 47.24        |
|     | Sine      | 3x3        | 2        | 50k          | <b>58.65</b> | <b>48.63</b> |
|     | RoPE      | 3x3        | 2        | 50k          | 58.56        | 48.50        |
| (b) | Sine      | 3x3        | 1        | 50k          | 57.94        | 48.13        |
|     | Sine      | 3x3        | 2        | 50k          | <b>58.65</b> | <b>48.63</b> |
|     | Sine      | 3x3        | 3        | 50k          | 58.35        | 48.32        |
| (c) | Sine      | 3x3        | 2        | 50k          | 58.65        | 48.63        |
|     | Sine      | 3x3        | 2        | 200k         | 59.40        | 49.84        |
|     | Sine      | 3x3        | 2        | 1M           | <b>59.87</b> | <b>50.43</b> |

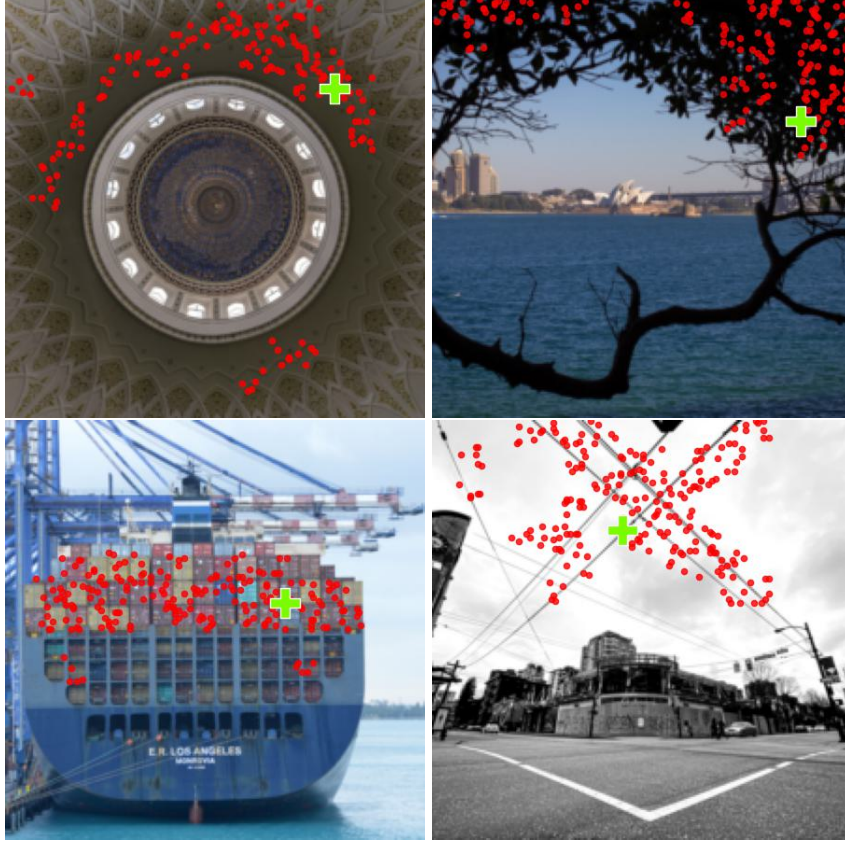
**Table 5.5: Ablation of architecture and training data size choices.** Upsamplers are trained using only Stage 1 loss for convenience.

**LoftUp benefits from the long-range, content-aware image-feature interaction thanks to the cross-attention mechanism.** Instead of using a local kernel as in most previous feature upsampling works, our cross-attention mechanism effectively allows global attention and content-aware upsampling. In Fig. 5.8, we visualize the attended region of a pixel in high-res coordinate (in green) in the corresponding low-resolution features (in red) where the density of red reflects the values of attention maps. As shown, the attended regions often exhibit semantically similar patterns across the image. Consequently, our upsampler leverages relevant information from the entire image, enhancing global semantic consistency and refining details at object boundaries.

**LoftUp supports arbitrary upsampling scales via its coordinate-based design.** As shown in Table 5.6, LoftUp maintains strong performance across varying input and feature resolutions. Here, “input res” refers to the image resolution fed into the VFM, while “feature res” denotes the final feature map size generated by LoftUp. Notably, LoftUp outperforms the original VFM backbone even when the latter processes images at  $2\times$  resolution—an approach that incurs nearly  $4\times$  the computational and memory cost. This highlights LoftUp’s efficiency and adaptability, making it well-suited for diverse scaling needs in downstream tasks.

**LoftUp also enjoys relatively high efficiency.** In Table 5.7, we compare the parameter size and inference time of each model. LoftUp has slightly more parameters than most alternatives, though still less than 20% of the VFM. Despite this, it is the second fastest upsampler after LiFT, which only performs  $2\times$  upsampling. This efficiency arises because LoftUp directly maps coordinates to features, bypassing multiple intermediate upsampling layers used by other methods. Moreover, compared to the coordinate-based FeatUp-Implicit that requires per-image optimization, LoftUp is both significantly faster and more





**Figure 5.8:** Visualization of attended regions (shown as *dots*) in the low-resolution features corresponding to a high-resolution pixel (marked as a *cross*). The density of *dots* represents the values in the attention map. LoftUp leverages relevant information from the global feature map to upsample features at each pixel.

effective.



| Method         | Input res | Feature res | COCO         | Cityscapes   |
|----------------|-----------|-------------|--------------|--------------|
| Low-res        | 224       | 16          | 51.21        | 36.54        |
| 2x-large input | 448       | 32          | 55.42        | 50.29        |
| 4x-large input | 896       | 64          | 56.79        | 58.45        |
| LoftUp         | 224       | 56          | 59.87        | 48.20        |
|                |           | 112         | 60.75        | 51.04        |
|                |           | 224         | <b>61.11</b> | <b>53.10</b> |
|                |           | 448         | 60.61        | 52.29        |
|                | 448       | 56          | 60.16        | 51.57        |
|                |           | 112         | 61.50        | 56.46        |
|                |           | 224         | 61.78        | 58.55        |
|                |           | 448         | <b>62.26</b> | <b>61.24</b> |

**Table 5.6: LoftUp maintains strong performance with diverse input sizes and up-sampling scales.**

| Architecture    | COCO         | Cityscapes   | Params (M)         | Infer time (s/img) |
|-----------------|--------------|--------------|--------------------|--------------------|
| Bilinear        | 56.15        | 44.79        | 22.1               | 0.0604             |
| Resize-conv     | 56.05        | <u>45.94</u> | 27.5 (+5.4)        | 0.0922             |
| LIIF            | 52.24        | 40.36        | 24.0 (+1.9)        | 0.1757             |
| LiFT            | 53.35        | 35.80        | 23.3 (+1.2)        | <b>0.0773</b>      |
| FeatUp-JBU      | <u>57.90</u> | 44.83        | <b>22.3 (+0.2)</b> | 0.1213             |
| FeatUp-Implicit | 51.12        | 44.81        | <u>22.5 (+0.4)</u> | 54.302             |
| LoftUp          | <b>61.11</b> | <b>53.10</b> | 26.4 (+4.3)        | <u>0.0893</u>      |

**Table 5.7: Comparison of efficiency for different upsampler architectures. Inference time is computed on a single A100 GPU.**

### 5.5.3 Generalization to Other VFM Backbones

To demonstrate that LoftUp generalizes beyond DINOv2, we evaluate with CLIP [157] and RADIO [160] as VFM backbones. The upsamplers are trained following the same procedure and evaluated at resolutions 224 and 448.

As shown in Table 5.8 and Table 5.9, LoftUp consistently outperforms all baselines when using CLIP and RADIO as VFM backbone, demonstrating the general applicability of our approach across different VFMs.

| Resolution | Upsampler | COCO         | Cityscapes   |
|------------|-----------|--------------|--------------|
| 224        | NA        | 40.30        | 30.79        |
|            | Bilinear  | 47.12        | 39.84        |
|            | FeatUp    | 52.08        | 33.50        |
|            | LoftUp    | <b>52.58</b> | <b>44.66</b> |
| 448        | NA        | 42.14        | 37.36        |
|            | Bilinear  | 48.32        | 45.58        |
|            | FeatUp    | 52.55        | 40.00        |
|            | LoftUp    | <b>53.87</b> | <b>50.14</b> |

**Table 5.8: Feature upsampler comparison with CLIP-B/16 [157] backbone.**

| Resolution | Upsampler | COCO         | Cityscapes   |
|------------|-----------|--------------|--------------|
| 224        | NA        | 51.00        | 34.42        |
|            | Bilinear  | 56.77        | 43.09        |
|            | FeatUp    | 56.59        | 42.23        |
|            | LoftUp    | <b>58.36</b> | <b>46.98</b> |
| 448        | NA        | 58.94        | 49.30        |
|            | Bilinear  | 62.29        | 57.42        |
|            | FeatUp    | 62.18        | 56.58        |
|            | LoftUp    | <b>63.55</b> | <b>60.83</b> |

**Table 5.9: Feature upsampler comparison with RADIOv2.5-B [160] backbone.**

#### 5.5.4 Task-Specific Implementation Details

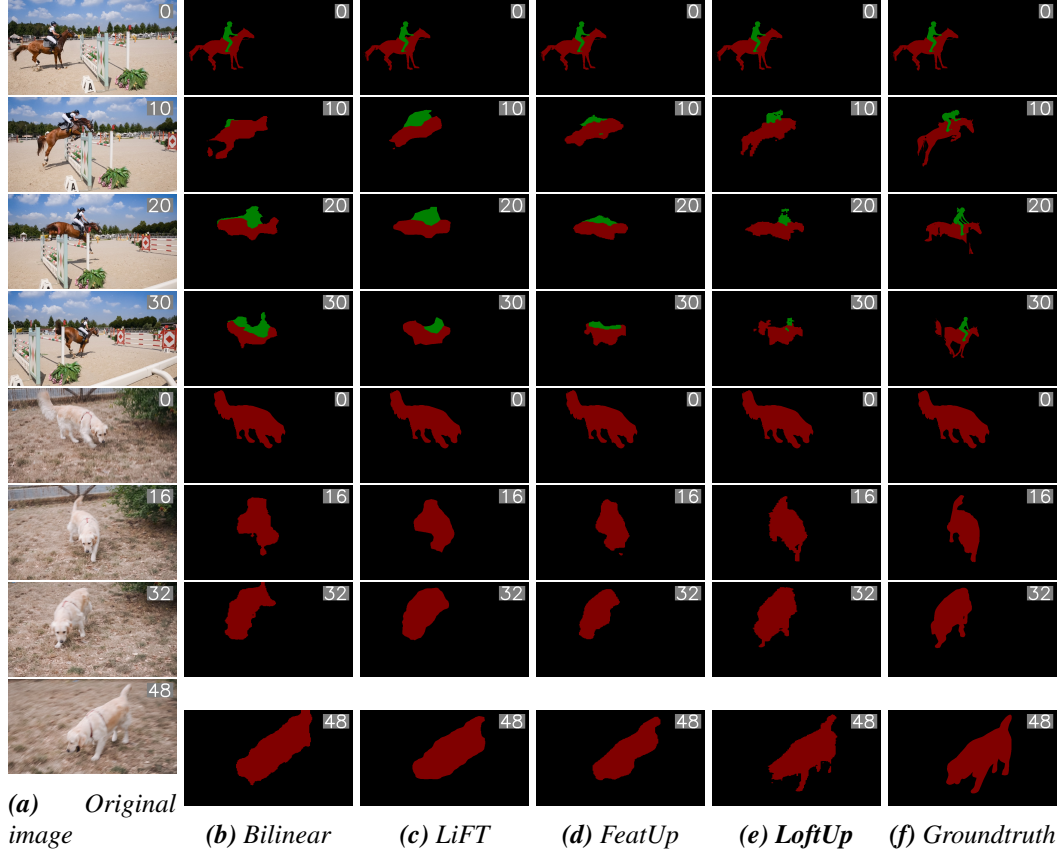
**Semantic segmentation.** Following [52, 67], we perform semantic segmentation on coarse classes in COCO-Stuff [17] (27 classes) and Cityscapes [33] (19 classes) and report mean Intersection-over-Union (mIoU) for each dataset. We train a linear decoder layer on up-sampled features with a batch size of 8 and AdamW optimizer [124] with a learning rate of 1e-4 for 10 epochs.

**Depth and normal estimation.** Following [45], we evaluate depth and normal estimation using NAVI dataset [86] and train a DPT decoder head with 7 convolutional layers on top of the VFM features. We use a batch size of 8 and AdamW optimizer [124] with a learning rate of 5e-4 for 10 epochs.

**Video object segmentation.** This task involves propagating an object segmentation mask across video frames, given the ground truth mask for the first frame. Following prior evaluation protocols [196, 84], we compute dense feature affinity maps between frames to track objects. We evaluate on the DAVIS validation set [153], a popular benchmark for video object segmentation comprising 30 videos of varying lengths, each containing between 1 and 4 objects.

**Zero-shot open-vocabulary segmentation.** We incorporate upsampled VFM features into ProxyCLIP [105]. Due to the high computational cost of proxy attention, we perform up-sampling to  $8\times$  for all upsampling methods. We use CLIP ViT-B/16 [157] as the CLIP backbone, DINOv2-S/14 [145] as the proxy VFM, and set the input resolution to 336px, matching the resolution of the CLIP backbone.

**Interactive segmentation.** We adapt the SimpleClick [120] architecture to evaluate up-sampled features. Specifically, we use a frozen VFM backbone and train a single-layer click encoder that directly adds to the image patch embedding, along with a three-layer convolutional decoder head on top of the upsampled features for interactive segmentation. For evaluation, following common practice [120, 191, 97], we sample the first click point as the farthest point from the object boundary, and report the mean IoU of the predicted segmentation masks with the groundtruth, denoted as IoU@1 Click.



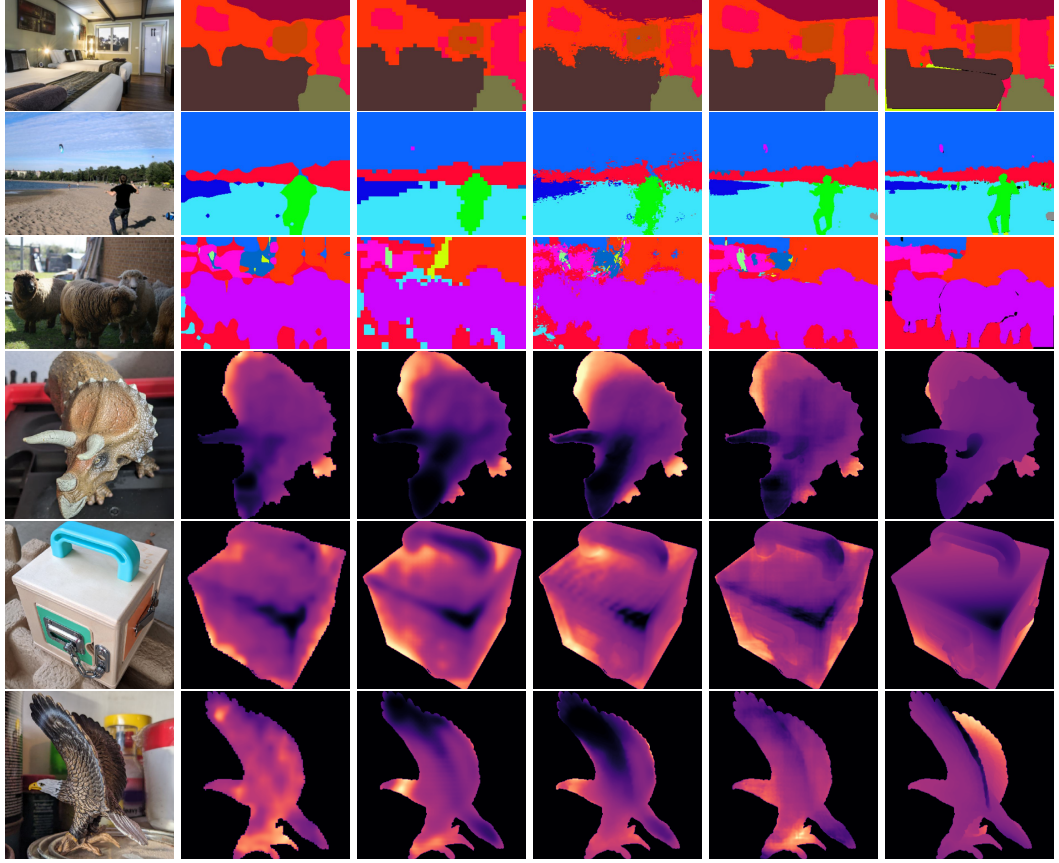
**Figure 5.9: Visualization of prediction examples of video object segmentation on the DAVIS 2017 dataset [153].** Each image displays its corresponding frame number in the top right corner. The groundtruth segmentation for the 0-th frame is provided, and dense feature affinity maps are employed to propagate its segmentation labels to subsequent frames. We can see that LoftUp outperforms all the other baselines in accurately tracking the objects across the frames.

### 5.5.5 Additional Visualizations

This subsection collects extended visualizations that complement the quantitative evaluations above.

## 5.6 Conclusion

In this work, we systematically explored upsampler architectures and training objectives for feature upsampling and introduced LoftUp, a coordinate-based cross-attention transformer upsampler trained with our proposed self-distilled high-resolution pseudo-GT. Our experiments demonstrate that LoftUp generates sharp, fine-grained feature maps, enables global content-aware image-feature interaction, supports arbitrary upsampling scales, and



(a) Original (b) Bilinear (c) LiFT (d) FeatUp (e) LoftUp (f) Groundtruth image

**Figure 5.10: More visualization of predictions examples on semantic segmentation on COCO-Stuff [17] and depth estimation on NAVI [86].**

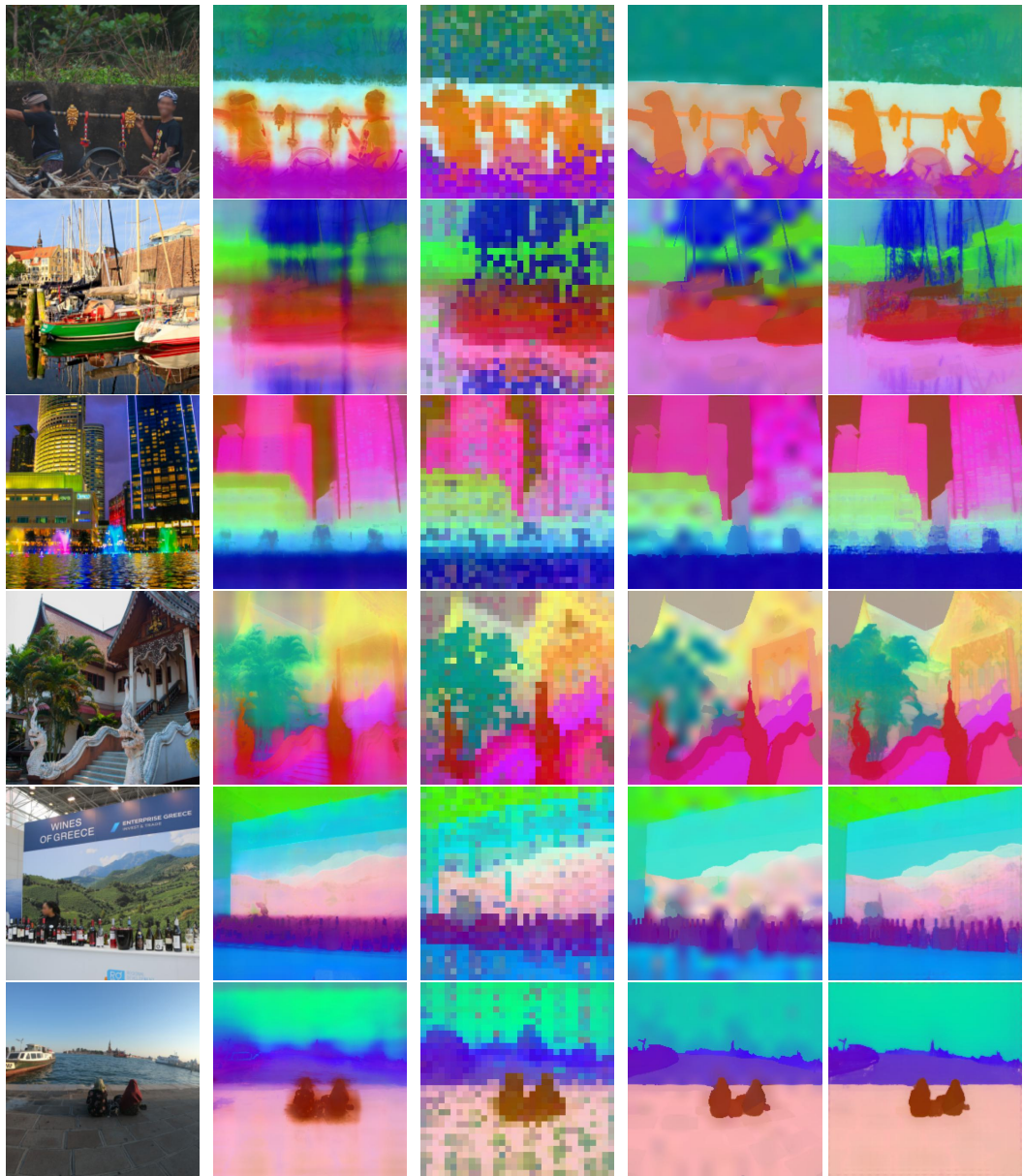
achieves fast inference. Notably, LoftUp outperforms existing methods across six diverse downstream tasks. We hope our work inspires further research in feature upsampling and contributes to the vision community by enhancing VFMs for a wide range of downstream applications.

## 5.7 Reflection: The Spatial Bridge in Retrospect

The spatial bridge demonstrates that boundary-aware pseudo-features provide an effective invariant for transferring dense structure from coarse to fine resolution, achieving large gains with modest parameter overhead. The bridge’s strength derives from SAM’s class-agnostic mask quality; its weakness emerges where SAM itself struggles—ambiguous boundaries, out-of-domain textures, or tasks where sharp boundaries may not be the relevant invariant (e.g., smooth depth gradients). When masks are poor, the pseudo-groundtruth

features are corrupted, and the upsampler inherits these errors.

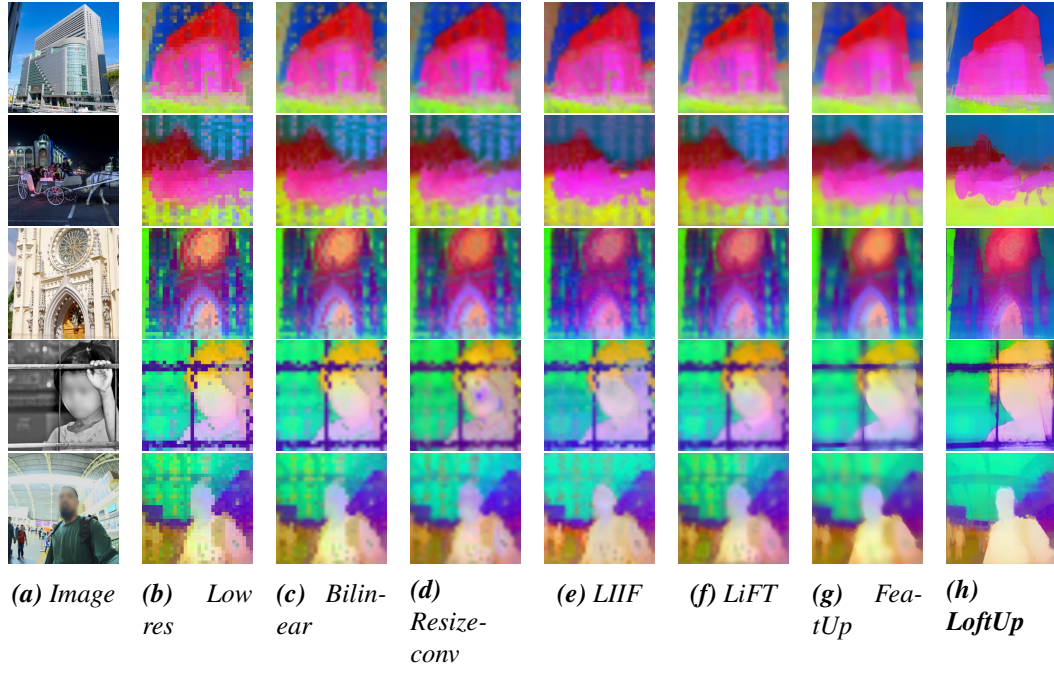
More fundamentally, boundary structure is the right invariant for edge-sensitive tasks but may be suboptimal for tasks requiring smooth spatial variation. This pattern—that every bridge has a domain of validity—recurs across all three technical chapters and is analyzed systematically in the Conclusion (Chapter 7). This chapter completes the trilogy of bridging invariants: geometric, semantic, and spatial. In each case, the invariant enabled generalization, but also introduced the biases of its carrier. The next chapter steps back from constructing bridges to ask a more fundamental question: how do we identify the gaps that need bridging in the first place?



(a) Original image (b) Per-image optimized (c)  $2\times$  features (d) Mask-Bicubic (e) Self-Distilled (LiFT)

**Figure 5.11: More visualization of different pseudo-GT.**





**Figure 5.12:** More visualization of features from various upsamplers. Backbone is DINOv2-S/14 [145].



**Figure 5.13:** More examples of attended regions (in *dots*) in the low-res features of a high-res pixel (in *cross*). The density of *dots* reflects the value of the attention map. LoftUp uses relevant information across the global feature map for upsampling features at each pixel.



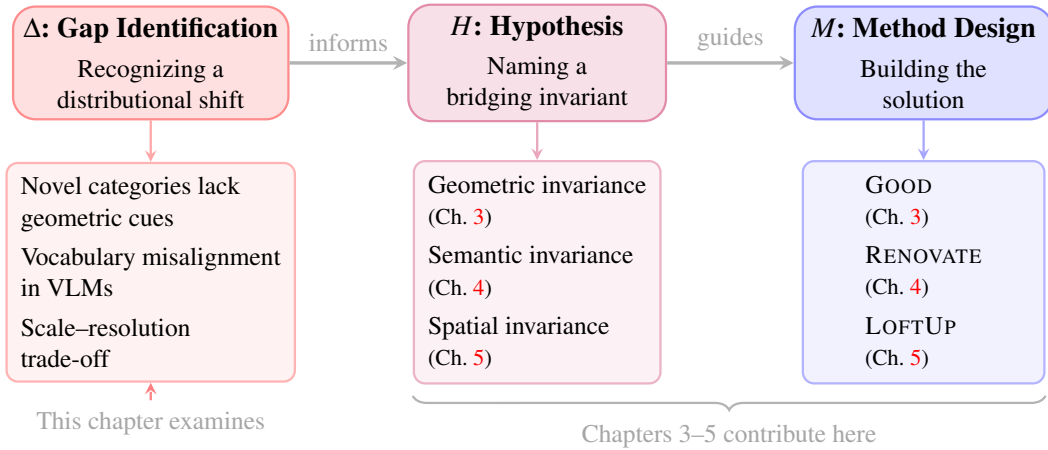
## 6 Beyond Known Unknowns — Discussion and Outlook

The preceding three chapters constructed a trilogy of bridges across known distributional gaps: geometric invariance for novel-category detection (Chapter 3), semantic invariance for vocabulary alignment (Chapter 4), and spatial invariance for dense prediction (Chapter 5). In every case, the gap was identified by a human researcher who understood *where* the model fails and *why* a particular invariant would help. This chapter steps back from bridge construction to examine that upstream step: the identification of the gap itself.

We advance two claims, each supported by a distinct strand of evidence developed below.

**Claim 1: Gap discovery is under-systematized.** Gap discovery—recognizing distributional shifts and defining what successful generalization across them looks like—is a necessary component of the full  $\Delta \rightarrow H \rightarrow M$  pipeline (introduced in Section 6.1), yet it remains the least systematized. This thesis contributed to the hypothesis (H) and method (M) stages through bridging invariants (Chapters 3–5); this chapter examines the gap identification stage ( $\Delta$ ) as the remaining piece needed to complete the stack. It is routinely treated as an exogenous precondition rather than a research contribution. Section 6.1 establishes this claim by analyzing the structure of OOD research and showing how the field’s publication norms relegate  $\Delta$  to a background assumption. Section 6.2 reinforces it by showing that foundation models expand the frontier of implicit coverage but leave the epistemic problem—knowing *where that frontier lies*—entirely unsolved.

**Claim 2: More developed methodology exists elsewhere and can be adapted.** Adjacent communities—including NLP behavioural testing [168], LLM safety [151], autonomous systems [51, 50], and data management [31, 47]—have developed failure-discovery methodology substantially more mature than what the OOD community currently employs. This methodology can be adapted, though not directly transplanted, to serve as infrastructure for systematic gap discovery. Section 6.3 surveys that methodology, organizes it along a continuum from human-led to fully autonomous, and identifies the representational limits that constrain each mode of discovery. Section 6.4 tests this claim empirically through a human–AI collaborative experiment that applies two off-the-shelf tools from the survey to the thesis’s own work, demonstrating both their diagnostic power and their dependence on a human-specified search space. Section 6.5 distils the open problems that remain even after adopting these methods, arguing that the OOD community’s theoretical apparatus is uniquely positioned to address them.



**Figure 6.1:** The gap–discovery pipeline from distributional gap ( $\Delta$ ) through hypothesis ( $H$ ) to method ( $M$ ). Each thesis chapter instantiates this pipeline: Chapters 3–5 contributed at the  $H$  and  $M$  stages; this chapter examines the  $\Delta$  stage.

Together, these two claims motivate the chapter’s central argument: that gap discovery should become a first-class research concern for the OOD community, and should integrate and extend methodology that already exists but remains fragmented across disciplinary boundaries.

## 6.1 The Gap Identification Problem

### 6.1.1 The structure of OOD research

Every OOD generalization effort follows a three-step pipeline:

- (1) **Gap identification ( $\Delta$ ):** A human recognizes a distributional shift and defines what success looks like.
- (2) **Hypothesis formation ( $H$ ):** The researcher identifies why a particular bridging invariant could close the gap.
- (3) **Method design ( $M$ ):** The concrete implementation that realizes the hypothesis.

To illustrate: in GOOD,  $\Delta$  was the observation that appearance-based detectors fail on novel categories;  $H$  was the hypothesis that geometric structure bridges the semantic gap;  $M$  was the pseudo-labeling pipeline. In RENOVATE,  $\Delta$  was the naming misalignment;  $H$  was that vision-language alignment provides the right invariant;  $M$  was the renaming model. The pattern is the same across all three chapters. Figure 6.1 illustrates this pipeline.

This thesis exemplifies the gap-as-given framing it critiques. In each technical chapter, the distributional gap was identified through artisanal domain expertise and manual error analysis—exactly the process this chapter argues needs systematization. The geometric, semantic, and spatial gaps were noticed because a researcher happened to examine the

right failure mode at the right time; a more systematic process might have identified them earlier, or surfaced additional gaps that remain unaddressed.

Despite its primacy,  $\Delta$  receives the least attention. The field’s publication norms treat  $H$  and  $M$  as the claimed contribution;  $\Delta$  is treated as a precondition. Yet the asymmetry is informational:  $\Delta$  determines *what problem is being solved*.  $H$  and  $M$  determine *how well it is solved*. A suboptimal method ( $M$ ) applied to a correctly identified gap still produces useful information—the gap is understood, and better methods can follow. A correct method applied to a misidentified or unconceptualized gap produces misleading results—it optimizes for the wrong problem and may conceal the actual failure mode. In short, a wrong  $\Delta$  corrupts the entire research trajectory; a wrong  $M$  is a local setback.

This asymmetry is visible in the field’s history. Some of the most impactful OOD contributions—ImageNet-C [72], DomainBed [66], WILDS [98]—were primarily  $\Delta$  contributions that reshaped entire research agendas, whereas individual method contributions, however innovative, rarely had comparable downstream influence. This thesis illustrates the same pattern: in GOOD (Chapter 3), we observed that appearance-based detectors systematically fail on novel categories and defined what “novel categories” meant; in RENOVATE (Chapter 4), we noticed that benchmark names are semantically misaligned with visual content; in LoftUp (Chapter 5), we recognized that VFM features lack spatial resolution for dense prediction. In each case, the gap identification came first; the bridging invariant and method were downstream.

The pattern generalizes beyond this thesis: domain adaptation [12, 55] assumes someone identified the source–target gap; domain generalization [66, 244] assumes someone enumerated the relevant domains; robustness benchmarks [72, 199] assume someone specified the perturbation types. This framing may seem to contradict how many researchers think about their work—most begin with a general goal (“detect any object”) rather than a specific named gap. But between any general goal and any specific improvement, someone must understand *where and how* the model fails. That understanding is gap identification, and it remains necessary even when the goal is general. The  $\Delta \rightarrow H \rightarrow M$  pipeline captures this intermediate step—the process by which a general aspiration becomes a specific, addressable problem. We argue that the process of discovering where one falls short of a general goal should become a systematic discipline.

### 6.1.2 Known unknowns, unknown unknowns, and the open world

A useful distinction separates distributional shifts we can name—**known unknowns** such as novel categories, domain gaps, and covariate shifts of specified types—from those we have not conceptualized at all: **unknown unknowns**. Unknown unknowns can arise along many dimensions: new visual domains (satellite imagery for a model trained on natural photos), new semantic categories (object types absent from any training taxonomy), new compositional rules (unusual object-context combinations), or new environmental conditions (sensor degradation, extreme weather, adversarial lighting). The distinction concerns whether the failure lies within the hypothesis space that the system’s designers prepared for, regardless of severity. A self-driving car encountering fog, where fog was a recognized challenge, faces a known unknown. The same car encountering systematic reflec-

tions from wet-road/low-sunlight interactions that mimic lane markings faces an unknown unknown—because no one had conceptualized it as a category of shift, regardless of how often it occurs. Because no one conceptualized it, it appears in no benchmark, no method targets it, no training data is curated for it, and no evaluation measures robustness to it. Every OOD benchmark tests known unknowns; deployment exposes unknown unknowns.

Industry has long recognized this asymmetry. In autonomous driving, companies operate what is known as a **closed-loop data engine** [111]—deploy, log failures, replay error scenarios, retrain, redeploy—that constitutes gap discovery in its most primitive form. The closed-loop data engine works, but it is purely reactive, instance-level, and silent on structural gaps. Kalra and Paddock [92] demonstrated mathematically that billions of miles of real-world driving would be needed to statistically validate safety through naturalistic testing alone, motivating the proactive testing approaches discussed in Section 6.3.

### 6.1.3 The field’s uneven progress

The OOD generalization community has made substantial progress on  $H \rightarrow M$ : given a known gap, the progression from hand-crafted domain adaptation to foundation-model-based bridging invariants (including this thesis) is dramatic. But progress on  $\Delta$ —identifying which gaps exist—tells a more complex story.

For most of its history, the OOD community treated gap identification as artisanal: a researcher examines failures, has an insight, names a problem, constructs a benchmark. The core methodological apparatus—DomainBed [66], WILDS [98], and their successors—still begins with “given domains...” or “given a known spurious correlation...” We call this the **gap-as-given framing**: the distributional shift is posited as an exogenous input to the research pipeline, and the contribution is judged entirely on the quality of the hypothesis ( $H$ ) and method ( $M$ ) that follow. The framing is epistemically limiting because it treats the hardest step—recognizing what the gap *is*—as a precondition rather than a problem to be solved, and thereby discourages the community from building methodology for it. One might argue that treating the distributional shift as given is simply normal science: every field starts from problem definitions. The distinction we draw is that most fields *also* develop systematic methodology for identifying which problems to solve—epidemiology has surveillance systems, software engineering has fuzz testing—whereas the OOD community’s problem-identification step remains informal.

A structured analysis of 573 recent OOD-related papers across eight top venues (NeurIPS, ICML, ICLR, CVPR, ICCV, ECCV, ACL, EMNLP; 2022–2025) quantifies this asymmetry (Fig. 6.2).

**Survey methodology.** Papers were collected via the DBLP [108] and OpenAlex [155] APIs using 12 topical query templates (e.g., [Venue Year] domain generalization, [Venue Year] out-of-distribution detection) across all venue-year combinations, yielding a corpus of 573 papers after deduplication and relevance filtering.<sup>1</sup> Each paper was classified along three dimensions—gap origin ( $\Delta$  inherited vs. new), novelty

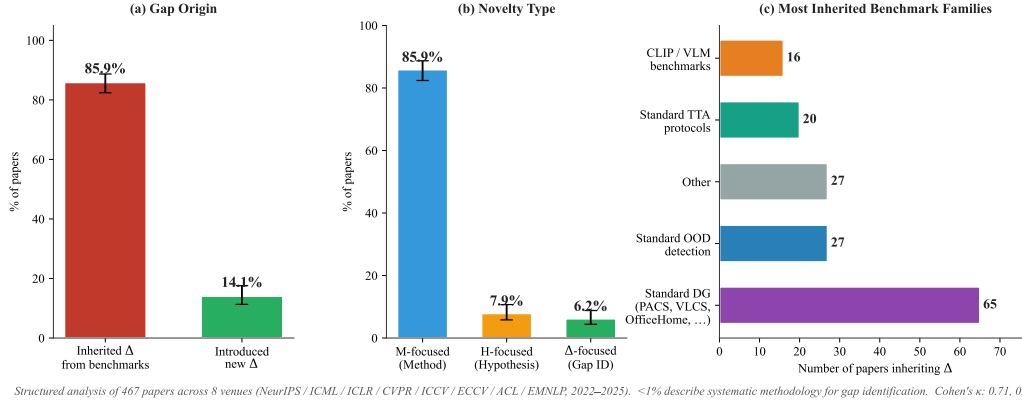
<sup>1</sup>Survey code and data: <https://github.com/andrehuang/ood-gap-survey>

type ( $\Delta$ /H/M), and presence of systematic gap methodology—using a structured coding rubric applied to the paper’s title and abstract. To ensure reliability, two independent LLM coders (Claude Sonnet [2] and GPT-4o [144]) classified each paper using the same rubric; inter-rater agreement was substantial (Cohen’s  $\kappa = 0.71$  for gap origin,  $\kappa = 0.75$  for novelty type,  $\kappa = 0.62$  for systematic methodology). Of the 573 papers, 467 received full consensus; the remaining 106 were adjudicated by the author. A title-based keyword heuristic served as a comparison baseline, showing substantially lower agreement with the LLM consensus ( $\kappa = 0.25$ ), confirming the value of abstract-based classification. Two limitations merit attention: classification was based on titles and abstracts rather than full-text analysis, and LLM coders may share biases from overlapping training data, so both factors could systematically misattribute novelty type. Despite these caveats, the magnitude of the asymmetry—over 85% of papers inheriting their gap—is large enough that even substantial classification noise would not alter the qualitative conclusion.

**Findings.** The analysis reveals a pronounced asymmetry between method innovation and gap innovation: 85.9% of papers (95% CI: 82.4–88.7%) inherit their distributional shift from existing benchmarks rather than identifying a new one, and a similar 85.9% locate their novelty contribution in the method (M), while only 7.9% contribute primarily at the hypothesis level (H) and 6.2% at gap identification ( $\Delta$ ). Systematic methodology for discovering gaps is rarer still, appearing in just 2.6% of the corpus. This asymmetry likely reflects the field’s incentive structure: proposing a new method on an established benchmark yields a clear experimental comparison, whereas proposing a new gap requires justifying both the gap’s existence and its relevance—a higher argumentative burden with less standardized evaluation. Standard benchmarks such as PACS [109], VLCS [48], and OfficeHome [205] account for the largest share of inherited shifts, indicating that the community overwhelmingly optimizes methods for a small, fixed set of known gaps rather than expanding the frontier of what gaps are known.

Critiquing the framing, however, is not the same as dismissing the benchmarks. DomainBed [66] and WILDS [98] are invaluable precisely because they crystallised distributional shifts clearly enough for controlled experimentation; the question is what the benchmark’s design reveals about implicit discovery methodology. Under a diagnostic reading—asking *how* gaps were identified rather than which algorithm wins—the same benchmarks become instruments for studying the gap-identification process itself. The continuum of discovery approaches surveyed in Section 6.3 organizes existing methodology precisely around this shift in perspective.

There are, moreover, early signs that researchers within the OOD community are arriving at questions adjacent to gap discovery from within the evaluative tradition. Yu et al. [230] explicitly reframe OOD evaluation as not only “whether” a model generalizes but “*where*” it generalizes well or poorly. Salaudeen et al. [176] demonstrated with OODSelect that aggregate OOD metrics can mask critical failure modes. SubpopBench [225] similarly examines subpopulation-level failures. These efforts remain evaluation-oriented—they ask whether current benchmarks measure the right thing, not how to systematically discover unconceptualised distributional shifts—but they represent genuine movement toward the



**Figure 6.2:** Structured analysis of 573 OOD-related papers across eight venues (NeurIPS, ICML, ICLR, CVPR, ICCV, ECCV, ACL, EMNLP; 2022–2025), classified by two independent LLM coders (Cohen’s  $\kappa = 0.71$ – $0.75$ ). Error bars show Wilson 95% CIs. **(a)** 85.9% of papers inherit their distributional shift ( $\Delta$ ) from existing benchmarks; only 14.1% introduce a new one. **(b)** 85.9% focus novelty on the method (M), 7.9% on the hypothesis (H), and only 6.2% on gap identification ( $\Delta$ ). Only 2.6% describe systematic methodology for gap discovery. **(c)** Standard domain-generalization benchmarks (PACS, VLCS, OfficeHome, ...) account for the largest share of inherited shifts.

epistemic questions this chapter foregrounds.

Meanwhile, outside the OOD community, more mature methodology exists, including automated slice discovery [31, 47], behavioural testing [168], automated red-teaming [151, 177], hidden stratification analysis [141], and proactive scenario generation [50]. The OOD community has begun to recognize the problem; what remains is connecting this recognition with the more developed methodology already available in adjacent fields. The remainder of this chapter traces that connection: Section 6.2 examines the epistemic limits of foundation-model coverage, Section 6.3 surveys the failure discovery methodology, Section 6.4 tests it on the thesis’s own work, and Section 6.5 distils the open problems.

## 6.2 Foundation Models and the Limits of Coverage

### 6.2.1 Implicit coverage through scale

Foundation models [157, 145]—trained on diverse, internet-scale data—demonstrate surprising robustness to distributional shifts no one specifically trained them for. Their power comes from scale: large datasets, large parameter counts, and pretraining objectives that serve as proxies for many downstream tasks. This scaling is what makes them effective carriers of bridging invariants but also determines the boundaries of what they can carry—their coverage extends only as far as the structural regularities present in their training data. In this thesis, foundation models served as sources of bridging invariants that made GOOD, RENOVATE, and LoftUp possible: they generalized to our target domains despite no one

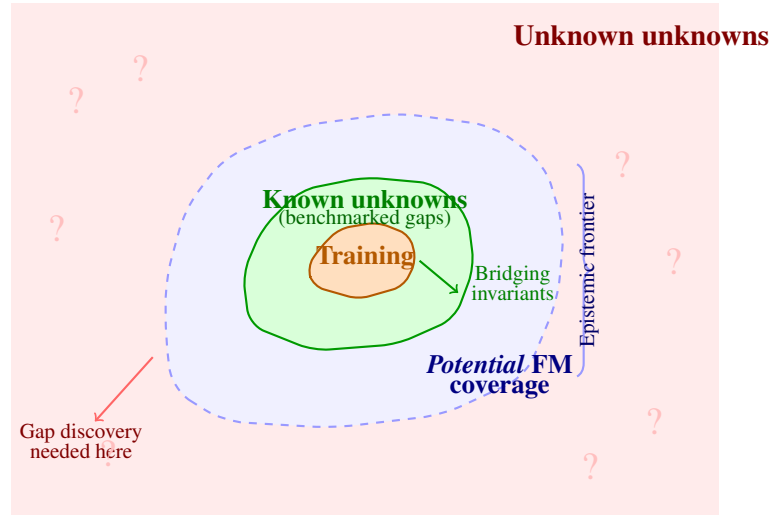
designing them to, because their representations were rich enough to cover our gaps incidentally. The Platonic Representation Hypothesis [81]—that representations across modalities and architectures are converging toward a shared statistical model of reality—suggests this implicit coverage may be far broader than any human could explicitly enumerate.

### 6.2.2 Structural limits

Despite its power, the coverage strategy has structural limitations that no amount of scaling can fully resolve. Several are well characterized: novel *combinations* of individually-seen factors occupy an exponentially growing interaction space that no training set can exhaustively cover [7]; genuinely novel phenomena (emerging sensor modalities, unprecedented environmental conditions) remain outside the implicit boundary because foundation models interpolate rather than extrapolate [72]; and shortcut learning persists at scale [58], because larger models find subtler shortcuts but the dynamic is structural, not a consequence of limited data.

A less recognized but more consequential limitation is **epistemic**: foundation models have made the OOD boundary far broader—yet tools for knowing *where that boundary is* remain absent. Coverage has expanded, but the epistemic problem persists. The question for the OOD community is not only how to make coverage broader but how to map **what lies beyond it**. Figure 6.3 illustrates this relationship: foundation models may expand the frontier of implicit coverage (shown as potential rather than guaranteed), but the vast region of unknown unknowns—by definition unmapped—persists beyond the epistemic frontier and remains the domain where deployment failures occur.





**Figure 6.3:** The coverage frontier. The training distribution (orange) is surrounded by known, benchmarked gaps (green) that bridging invariants (Chapters 3–5) are designed to cross. Foundation models may expand coverage further (blue, dashed to indicate potential rather than guaranteed coverage), but the vast region of unknown unknowns (red) persists beyond the epistemic frontier. Gap discovery methodology is needed to map this unmapped territory.

## 6.3 Failure Discovery Across Communities

If foundation models expand coverage but leave the epistemic problem unsolved, where should the OOD community look for methodology to address it? Across data management, NLP, LLM safety, medical AI, and autonomous systems, researchers have built methodology for systematically finding model failures. This section synthesizes that work through the lens of the bridging invariant framework: just as the field transitioned from hand-crafted to foundation-model-based invariants, the transition from artisanal to systematic gap discovery is underway. The analogy to learned features is instructive: the shift from SIFT [125]/HOG [36] to CNNs [102] changed how features were obtained—from artisanal craft to systematic methodology—while the underlying need persisted.

### 6.3.1 Failure discovery vs. gap discovery

Before surveying specific methods, a key distinction is needed. Most work surveyed here is more precisely described as **failure discovery**: finding where an existing model underperforms on data that is already available or can be generated. This overlaps with, but is narrower than, **gap discovery**: identifying distributional shifts between training and deployment that have not yet been conceptualized.

Failure discovery asks: given a model and evaluation data, where does performance break down? Gap discovery asks: what distributional shifts exist in the deployment world that are not represented in any existing dataset or ontology? A discovered failure often

implies a distributional gap, and the methods for the former are among the best tools for approaching the latter. But they are not the same. Failure discovery is bounded by available data and representations; gap discovery, at its hardest, requires going beyond both.

### 6.3.2 A continuum of discovery approaches

We organize existing work into three groups by degree of human involvement: *human-led* systematic discovery, which provides the baseline methodology; *fully automated* discovery, which offers the highest scalability and represents the frontier with the greatest potential; and *human-AI collaborative* discovery, which currently achieves the strongest empirical results by combining automated coverage with human ontological richness. The progression is not strictly linear—collaborative methods build on automated ones and currently compensate for their limitations, which is why we discuss them last.

#### Human-led systematic discovery

Human-led systematic discovery provides the baseline methodology but is bounded by what experts can articulate. **CheckList** [168] adapted software engineering testing principles to NLP; practitioners with CheckList found three times more bugs than those without it. **ImageNet-C** [72] demonstrated that standard models are systematically brittle to 75 human-enumerated visual corruptions. In both cases, gap identification was the contribution; everything downstream was method work. Yet human-led discovery scales poorly to combinatorial interactions among shift types, and it is fundamentally a methodology for known unknowns—powerful for structuring what experts suspect but cannot yet measure, insufficient for what no one has yet imagined.

The three technical chapters of this thesis exemplify human-led gap discovery: the geometric, semantic, and spatial gaps were identified through manual error analysis and domain knowledge. The question is whether that process could have been accelerated or made more reliable.

#### Fully automated discovery

Fully automated methods represent the scalable frontier of failure discovery. We organize them into four sub-topics reflecting distinct strategies for removing human involvement from the discovery loop.

**Automated slice discovery.** **Slice discovery** methods find coherent, semantically meaningful data subgroups where models underperform. **Domino** [47] established the paradigm: embed inputs using CLIP, cluster errors via a mixture model, and generate natural language descriptions. It also introduced **DCBench**, the first quantitative evaluation framework for slice discovery. Subsequent methods include **PRIME** [166], which reverses the paradigm by first obtaining interpretable concept tags and then analyzing model behavior conditioned on tag combinations, and **LADDER** [61], which uses LLMs to generate testable hypotheses about model biases without clustering.

**Automated adversarial and proactive testing.** The LLM safety and autonomous driving communities have developed complementary methods for discovering model vulnerabilities at scale. In the safety domain, the central challenge—**mode collapse**, where RL-trained attackers converge on few strategies—has driven the most important work. **Rainbow Teaming** [177] reframes adversarial discovery as quality-diversity optimization using MAP-Elites, mapping the topology of the vulnerability landscape. **GFlowNet Red Teaming** [107] trains attackers to sample from the full distribution over strategies. In the autonomous driving domain, Feng et al. [50] demonstrated 1,000–10,000× acceleration in safety testing by training adversarial background agents (published in *Nature*), while **Scenic** [51, 206] provides a probabilistic programming language for scenario specification with formal guarantees. Across both domains, the unifying progression is from finding individual failures to mapping vulnerability *landscapes*—understanding the structure of the failure space rather than cataloguing isolated examples.

**LLM-driven algorithmic discovery.** AlphaEvolve [140] and subsequent work [114] demonstrate that LLMs can serve as generators of structured hypotheses in combinatorial search spaces. While gap discovery lacks the clean fitness functions that algorithmic discovery enjoys, the underlying capability—LLMs as generators of testable hypotheses—is directly relevant to the gap discovery problem.

**Autonomous model diagnosis.** At the far end, systems such as **AutoDetect** [28] aim for fully autonomous discovery. In vision, **LANCE** [154] chains captioning, LLM perturbation, and text-to-image diffusion to surface class-level biases. These approaches face the **ontological barrier**: a model cannot discover failures along dimensions absent from its representations. This ontological constraint applies broadly across all automated methods: slice discovery operates on existing data manifolds, red-teaming generates inputs within the model’s operational domain, and even LLM-driven algorithmic search explores within human-defined fitness landscapes.

### Human–AI collaborative discovery

Collaborative methods currently represent the practical sweet spot, compensating for the limitations of fully automated approaches by injecting human judgment into the discovery loop. **AdaVision** [56] allows users to describe hypotheses in natural language and iteratively hill-climbs toward high-error regions using CLIP embeddings. User-discovered failure groups have 2–3× higher failure rates than automatic clustering alone. **AdaTest** [167] uses GPT-3 to generate test cases with humans steering, making practitioners 5–10× more effective. **Dynabench** [95] provides infrastructure for continuous adversarial benchmarking.

Collaborative methods currently outperform some fully automated approaches—notably automated clustering—because human ontological richness names failure dimensions the model lacks representations for. As automated representations improve and the ontological barrier recedes, this advantage may narrow, but for now the human contribution remains essential for gap discovery in its fullest sense. Both partners bring constraints, however:

human priors shape the search direction so that discovered failures cluster around concepts humans find salient, while the AI component operates within fixed embedding spaces. Collaborative discovery also requires skilled users; Johnson et al. [89] found that practitioners construct inconsistent descriptions of the same discovered groups, suggesting that the human contribution is itself a source of noise, not only of signal.

The progression from human-led to collaborative discovery mirrors the progression within this thesis: from artisanal gap discovery (the researcher notices errors and has an insight) toward a more systematic process where automated tools surface patterns and humans provide interpretive structure.

Across this continuum, a consistent trade-off emerges: as the locus of discovery migrates from human to machine, throughput increases while justificatory depth decreases. Human-led methods yield richly warranted but costly gap descriptions; fully automated methods achieve the highest coverage but reduce the gap to a performance delta stripped of explanatory structure; collaborative methods occupy a middle ground that currently yields the largest empirical gains by retaining human-mediated coherence checks. The appropriate validation regime must therefore be calibrated to the autonomy level: human-led discovery can rely on peer-review norms, automated discovery demands formal verification layers, and collaborative discovery requires audit of the generative partner’s blind spots. Table 6.1 summarises the surveyed methods along these dimensions.

**Table 6.1:** Summary of failure discovery methods organised along the human–autonomous continuum. Throughput estimates relative speed of the discovery loop; justificatory depth indicates how richly warranted the resulting gap descriptions are.

| Method                | Mode   | Key mechanism                      |                              | Throughput | Justification |
|-----------------------|--------|------------------------------------|------------------------------|------------|---------------|
| CheckList [168]       | Human  | Software-testing                   | templates for NLP            | Low        | High          |
| ImageNet-C [72]       | Human  | Expert-enumerated                  | visual corruptions           | Low        | High          |
| Domino [47]           | Auto   | CLIP embedding                     | clustering + NL descriptions | High       | Low           |
| PRIME [166]           | Auto   | Concept-tag                        | conditioning                 | High       | Medium        |
| LADDER [61]           | Auto   | LLM-generated                      | bias hypotheses              | High       | Medium        |
| Rainbow Teaming [177] | Auto   | Quality-diversity                  | adversarial search           | High       | Low           |
| Feng et al. [50]      | Auto   | Adversarial                        | agent training (AV)          | High       | Low           |
| AlphaEvolve [140]     | Auto   | LLM-driven                         | evolutionary code search     | High       | Low           |
| LANCE [154]           | Auto   | Caption→LLM→diffusion pipeline     |                              | Medium     | Low           |
| AdaVision [56]        | Collab | NL hypothesis + CLIP hill-climbing |                              | Medium     | High          |
| AdaTest [167]         | Collab | LLM generation + human filtering   |                              | Medium     | High          |
| Dynabench [95]        | Collab | Continuous                         | adversarial benchmarking     | Medium     | Medium        |

These methods can surface individual failures, but a deeper question remains: when has the discovery process found *enough* failures? Without a definition of completeness—a criterion for when testing against distributional shifts is *done*—gap discovery remains open-ended. A model can always be tested against one more perturbation, one more domain, one more compositional rule. The question of when a model has been tested against *sufficiently many* gaps to warrant a claim of generalization is fundamentally a question about what “sufficiently generalized” means under specified conditions. We return to this completeness problem in Section 6.5, where it connects to the broader challenge of evaluating gap discovery.

Could these methods have surfaced the gaps addressed in this thesis? The naming misalignment identified in RENOVATE (Chapter 4) is particularly amenable to the human–AI collaborative approach, as we demonstrate in the following section.

## 6.4 Case Study: Retroactive Gap Discovery for RENOVATE

The previous section surveyed failure discovery methodology along a continuum from human-led to fully automated. This section instantiates the collaborative approach that Section 6.3 argued is the practical sweet spot: a human researcher provides the hypothesis—that the semantic interface may be misaligned—and automated tools validate and quantify the gap. Using the thesis’s own work as a test bed, we ask whether the naming misalignment that motivated RENOVATE (Chapter 4) could have been detected and measured using only off-the-shelf tools available at the time of its development.

This collaborative framing is central to the experiment’s design, not a caveat. The creative insight “the semantic interface may be misaligned” is itself a  $\Delta$ -level contribution that precedes any automated analysis. Our tools *validate* this hypothesis—they compress what might otherwise require weeks of manual error analysis into minutes, with the creative hypothesis still coming from the researcher. The experiment thus demonstrates the power of human–AI collaboration in gap discovery: the human defines the search space, and automated tools provide rapid, quantitative answers within it.

Three additional clarifications frame the experiment.

1. In RENOVATE (Chapter 4), we identified the naming gap through manual error analysis and domain expertise. The question is whether automated tools could have *accelerated* identification, particularly for researchers without the specific domain intuition.
2. Detection and resolution require different kinds of insight: even if automated tools flag problematic names, RENOVATE’s instance-level renaming pipeline remains necessary for the solution.
3. We use no information from RENOVATE: the experiment operates on the original ADE20K names and off-the-shelf tools, making it a genuinely prospective diagnostic applied retroactively.

### 6.4.1 Layer 1: CLIP text-space diagnostic

Following the automated slice discovery paradigm surveyed in Section 6.3—embed inputs in a foundation model’s representation space and analyze the resulting structure—we embed the 150 class names of the ADE20K panoptic benchmark [241] in the text-embedding space of CLIP [157] (ViT-B/32, laion2b\_s34b\_b79k) using the prompt template “a photo of a {name},” compute the  $150 \times 150$  pairwise cosine similarity matrix, and rank all  $\binom{150}{2} = 11,175$  pairs by similarity. Unlike Domino [47], which clusters image embeddings to find failure slices, we operate entirely in CLIP’s text space—requiring only a single forward pass through the text encoder, no images.

Figure 6.4 shows the 20 most similar pairs. The top results—*stairs/stairway* (cos = 0.973), *chair/armchair* (0.956), *cushion/pillow* (0.954)—are near-synonyms that a vision-language model cannot reliably distinguish in text space. Of all pairs, 62 exceed a cosine similarity of 0.85 and 321 exceed 0.80. These are precisely the cases most likely to produce what we termed “benign misclassifications” in Chapter 4: predictions that are semantically reasonable but penalized by the benchmark’s rigid category boundaries. Names with the highest mean similarity to all other names—*screen*, *chair*, *computer*, *bed*, *monitor*—are the overly broad categories that cover multiple visually distinct sub-types.

### 6.4.2 Layer 2: LLM-based diagnostic

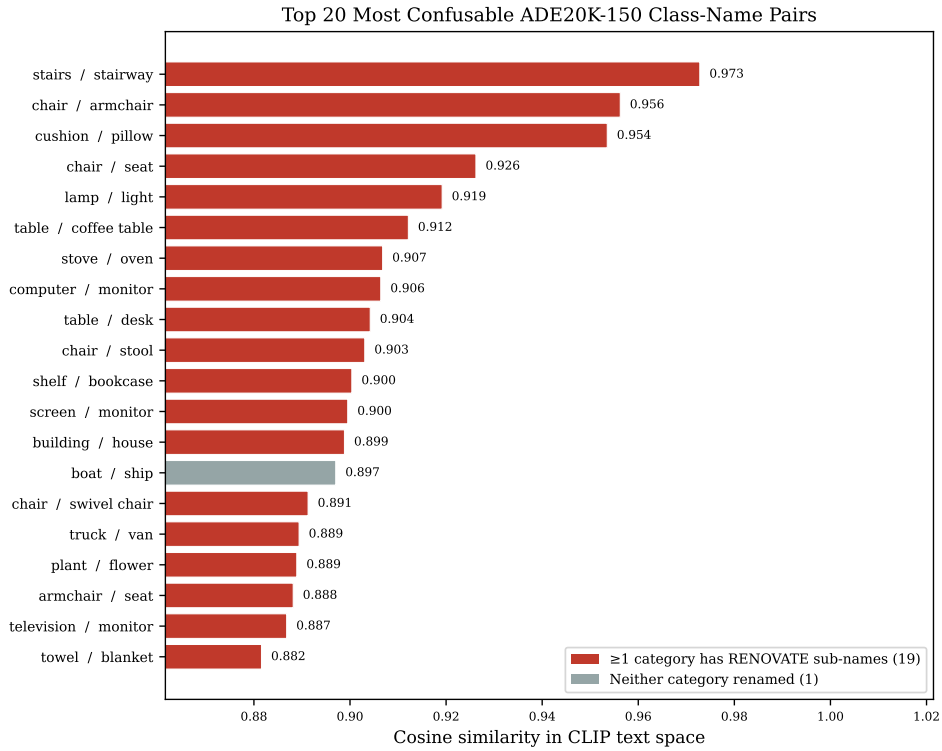
Text-space geometry captures structural proximity but misses *semantic* issues that require world knowledge—precisely the kind of issue that the LLM-driven diagnosis approaches surveyed in Section 6.3 are designed to surface. We therefore query a large language model (Claude Sonnet 4 [2]; May 2025) with the 150 ADE20K class names and ask it to predict which names would be problematic for an open-vocabulary segmentation model using CLIP-based text-image matching. The LLM receives no images, no model outputs, and no information about RENOVATE—only the list of names.

The LLM identifies three categories of issues, each largely non-overlapping with the CLIP analysis:

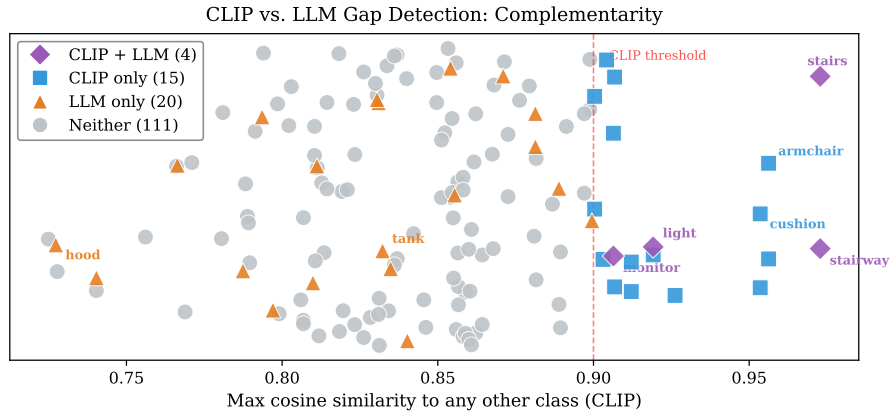
- **Context-dependent ambiguity:** *screen* (computer screen, window screen, room divider), *hood* (kitchen range, car, clothing), *case* (suitcase, display case, phone case), *tank* (water, fish, military). These are names whose referent shifts with scene context—an issue invisible to pairwise text similarity.
- **Functional overlaps:** *countertop/counter* (part vs. whole), *floor/earth* (indoor/outdoor ground), *windowpane/glass* (object vs. material). These require reasoning about the relationship between categories, not just their distance.
- **Over-generic categories:** *apparel*, *plaything*, *animal*, *food*—hypernyms whose visual diversity is far too broad for a single text prompt to capture.

The LLM flags 24 names as problematic, while the CLIP analysis (at a threshold of cos > 0.90) flags 19. Only 4 names appear in both sets (Fig. 6.5), confirming that the two methods surface largely *complementary* aspects of the naming gap. Together they cover a wider diagnostic range than either alone.





**Figure 6.4: Top 20 most confusable ADE20K-150 class-name pairs in CLIP text space.** Red bars indicate pairs where at least one category has RENOVATE sub-names; grey bars are pairs where neither category was renamed. Nearly all top pairs involve synonym or hypernym–hyponym relationships.

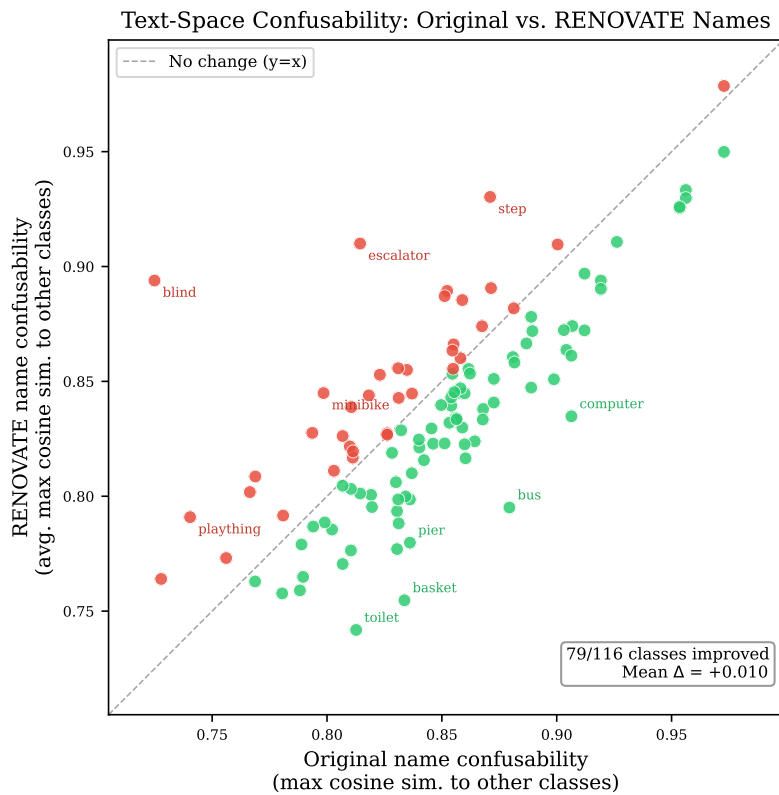


**Figure 6.5: CLIP vs. LLM diagnostic overlap.** Of 39 names flagged as problematic across both methods, only 4 appear in both sets, confirming that embedding-based and reasoning-based diagnostics surface complementary aspects of the naming gap.

### 6.4.3 Layer 3: Validation against RENOVATE

Detecting confusable names is suggestive; validating that RENOVATE’s solution addresses the detected issues is more convincing. We embed the RENOVATE sub-names for 116 ADE20K classes—a total of 369 alternative names—in the same CLIP text space, and measure whether renaming reduces inter-class confusability.

Figure 6.6 shows the result as a scatter plot: each point is one category, with original confusability on the horizontal axis and RENOVATE confusability on the vertical axis; points below the diagonal indicate improvement. For 79 of 116 categories (68%), RENOVATE’s more specific names reduce the maximum cosine similarity to any other original class. *Bus* (−0.084), *basket* (−0.079), *computer* (−0.072), and *toilet* (−0.071) show the clearest improvements—cases where the original name was highly generic and RENOVATE’s context-specific alternatives are better separated in CLIP’s text space.



**Figure 6.6: Text-space confusability before and after RENOVATE.** Each point is one of 116 ADE20K classes with RENOVATE sub-names. The horizontal axis shows confusability of the original name (max cosine similarity to any other class); the vertical axis shows the average confusability across RENOVATE sub-names. Points below the diagonal (green) indicate improvement; points above (red) indicate degradation. Renaming reduces confusability for 79 of 116 categories, with a mean reduction of 0.010.

However, for 37 categories (32%) the confusability *increases*. The most pronounced

cases—*blind* (+0.169), *escalator* (+0.096), and *step* (+0.059)—reveal a systematic pattern: RENOVATE replaces an ambiguous name with a more descriptive alternative that happens to overlap textually with a related category. For example, “window screen” (RENOVATE’s name for “blind”) is closer to “glass” and “windowpane” in CLIP’s text space; “moving stairway” (for “escalator”) is closer to “stairway” and “stairs.” The pattern reveals a limitation of text-space confusability as a diagnostic metric. These renamings trade inter-class separation in text space for improved alignment between names and actual visual content—a subtler objective that requires the image-level analysis that RENOVATE’s full pipeline provides.

#### 6.4.4 What the case study shows

This experiment yields three findings relevant to the chapter’s argument.

**Finding 1: Given a human-specified hypothesis, automated tools can rapidly validate and quantify the gap.** Both tools operated within a human-specified search space: “analyze these category names.” The prior insight that names are worth examining—rather than, say, the backbone architecture or the augmentation pipeline—is itself the most consequential part of the gap discovery process and was not automated here. Once a researcher formulates the right *question* (“is the semantic interface problematic?”), automated tools can provide a rapid, quantitative *answer*, compressing weeks of manual error analysis into minutes. This positions automated validation as a screening tool that can be applied systematically across multiple hypothesized gap dimensions. The broader completeness question—when has the discovery process tested enough dimensions of shift?—is discussed in Section 6.5.

**Finding 2: Different tools catch different aspects.** The near-zero overlap between CLIP-flagged and LLM-flagged names confirms that embedding-based and reasoning-based diagnostics surface complementary failure dimensions. This complementarity mirrors the collaborative discovery paradigm discussed in Section 6.3 and suggests that a principled mapping from gap types to diagnostic tools (Section 6.5) would have direct practical value.

**Finding 3: Detection and resolution require different kinds of insight.** Text-space confusability is a valid diagnostic for *identifying* the naming gap but does not fully characterize *what a good solution looks like*—RENOVATE sometimes increases text-space confusability while improving downstream performance by aligning names with actual visual content. This gap between detection and remediation is a concrete instance of the gap-to-invariant pipeline problem discussed in Section 6.5: discovering that a problem exists and understanding the right fix require different kinds of insight, and the connection between them is non-trivial.

**Scope and limitations.** This case study demonstrates that, given a human-specified hypothesis, off-the-shelf tools can detect and quantify the *semantic* gap addressed by RENOVATE, which is particularly amenable to text-space analysis because RENOVATE operates within CLIP’s text-image matching framework. The geometric gap (GOOD) and the spatial gap (LoftUp) would require fundamentally different diagnostics—image-level analysis for geometric structure, feature-level analysis for spatial resolution—confirming that different

gap types demand different discovery tools. The before/after comparison covers all 116 ADE20K classes for which RENOVATE provides alternative names, making it comprehensive. More fundamentally, the entire experiment presupposes that someone identified “naming” as the dimension worth examining. This is a concrete instance of the ontological barrier (Section 6.5): automated tools can discover gaps along dimensions they are directed to examine, but the choice of which dimensions to examine remains a human contribution—and it is often the most consequential one.

## 6.5 What Remains Hard

The failure discovery methods surveyed in Section 6.3 and tested in Section 6.4 reveal that the hard problems concern understanding what the tools can and cannot reach, and what it means to have reached enough.

Four questions structure the remainder of this discussion, each depending on the previous one. First, **what can we find?** Every discovery system operates within representational and ontological limits; understanding these limits determines which gaps remain structurally invisible to which methods (Section 6.5.1). Given these limits, **which gaps matter?** The space of discoverable gaps is enormous, and the challenge of distinguishing meaningful from trivial gaps constrains both what we should look for and when we can stop looking (Section 6.5.2). With meaningfulness defined, **when have we found enough?** Even with perfect tools, gap discovery is open-ended without a criterion for completeness that specifies when testing against distributional shifts is sufficient (Section 6.5.3). Finally, **what do we do about it?** Discovered gaps must be connected to remediation—the gap-to-invariant pipeline that closes the practical loop (Section 6.5.4).

### 6.5.1 What can we find: The limits of discovery

The most fundamental constraint on any gap discovery system is the **ontological barrier**: a model cannot discover failures along dimensions absent from its representations. A model without temporal context cannot discover temporally-correlated failures. A model without social reasoning cannot discover failures caused by misunderstood social intent. The case study (Section 6.4) illustrated a related but distinct limitation: the automated tools had no difficulty analyzing names once directed to do so, but the prior hypothesis that *names* are the relevant dimension to examine had to come from a human researcher. This is less an ontological barrier—the tools’ representations covered names perfectly well—than a *search-direction* problem: the space of possible diagnostic dimensions is vast, and selecting which dimension to probe remains a human contribution.

The barrier has been observed empirically throughout the failure discovery literature. LADDER [61] discovers biases that embedding-based methods miss, precisely because it uses LLM-generated hypotheses that can transcend the vision model’s ontology. AdaVision [56] and AdaTest [167] show human-AI collaborative methods outperforming fully automated ones, because human ontological richness can name failure dimensions the model has no representation for. AlphaEvolve [140] pushes further by using LLMs as gen-

erators of novel algorithmic structures—but even here, the search remains bounded by the human-specified problem formulation. The Platonic Representation Hypothesis [81] suggests the ontological horizon is expanding as representations converge, but convergence to the “statistical structure of reality as captured by internet-scale data” still has limits—reality contains causal, social, and physical structure not fully captured by co-occurrence statistics. Formalizing the ontological barrier would allow the community to characterize, for any given representational paradigm, which classes of gaps remain structurally invisible.

The ontological barrier sets an outer boundary on discovery, but within that boundary different tools have different blind spots. Most existing methods perform failure discovery rather than gap discovery in the full sense. The progression from one to the other requires *extrapolation*: using observed failure patterns to hypothesize about structurally similar failures in unobserved distributional regions. Given that “the model confuses transparent objects with empty space,” can we systematically predict deployment scenarios where structurally analogous failures occur? The deeper question is whether there is a **science of gap structure**: a taxonomy of *ways distributional shifts go wrong*, analogous to how software engineering has taxonomies of bug types. The failure discovery literature already hints at structure: shortcut learning [58] is a recurring class of failure; combinatorial interactions—where individually-seen factors combine in untested ways [7]—constitute another; missing representational capabilities—where the model lacks an entire modality or reasoning faculty, such as temporal context or 3D spatial understanding [182]—a third. Each has different signatures and different discoverable-by-whom properties.

No Free Lunch theorems tell us that no universal gap discoverer exists. The central question is therefore “**what inductive biases should a gap discovery system have, and what determines which biases are appropriate for a given deployment context?**” The case study (Section 6.4) provides a concrete instance of this principle: CLIP text embeddings and LLM reasoning surface largely non-overlapping naming issues—only 4 of their combined 39 flagged names overlap—suggesting that a principled mapping from gap types to diagnostic tools, rather than ad hoc tool selection, would have direct practical value. Embedding analysis provides breadth while LLM reasoning provides depth, and the complementarity mirrors the collaborative discovery paradigm discussed in Section 6.3. The OOD community’s theoretical apparatus—distributional shift formalization, representation learning, causal modeling—is uniquely positioned to develop such a theory of gap-type-to-tool correspondence.

Even within what discovery systems can reach, however, the space of discoverable gaps is enormous—and most are trivial. The question of which gaps are worth bridging is the subject of the next subsection.

## 6.5.2 Which gaps matter: The meaningfulness problem

Not all discoverable gaps are worth bridging. Adversarial OOD generators such as Rainbow Teaming [177] and GFlowNet Red Teaming [107] can produce a large volume of distributional shifts, but most are trivial—they demonstrate that a model breaks under adversarial perturbation without revealing a structural limitation or suggesting a remedy. The

challenge is designing gap discovery methods that find *meaningful* gaps: those whose identification yields genuine insight rather than mere volume.

We propose three criteria for meaningfulness. First, **ecological validity**: the gap should correspond to conditions that actually arise in deployment. A model that fails on imperceptibly perturbed images reveals an interesting mathematical property but not a deployment risk. Second, **structural insight**: the gap should expose a representational limitation of the model—one that reveals which features it relies on. A texture-bias gap [59] is meaningful because it reveals that the model relies on the wrong features; a random-noise gap lacks structural insight because it says nothing about *which* features the model uses. Third, **actionability**: knowing about the gap should suggest a remediation strategy—ideally, a bridging invariant that would close it. Trivial gaps fail on this criterion because they suggest no remedy beyond adversarial training on the specific perturbation that caused them.

These criteria provide a practical test: a gap is meaningful if one can name a bridging invariant that would close it. The geometric gap addressed by GOOD is meaningful because geometric structure is the invariant; the naming gap addressed by RENOVATE is meaningful because vision-language alignment is the invariant. A gap discovered by random perturbation typically fails this test—there is no structural property that “bridges” random noise. Several directions for operationalizing meaningfulness have appeared across the surveyed literature, though no unified framework exists. Quality-diversity optimization (MAP-Elites [177]) can be steered toward ecologically valid regions of the perturbation space. Frequency-weighted scoring, where gaps are weighted by their estimated deployment probability, draws on the operational testing literature in autonomous driving [92, 50]. Semantic coherence constraints, where discovered gaps are required to be describable in natural language, serve as a proxy for structural insight and underlie methods such as CheckList [168] and LADDER [61]. These are illustrative rather than exhaustive; other operationalizations—such as causal-graph-based criteria [182] or coverage metrics adapted from software testing—may prove equally valuable.

The meaningfulness criteria define *what* we are trying to be complete over, which enables the next subsection’s question of *when we have found enough*.

### 6.5.3 When have we found enough: Completeness and evaluation

Existing evaluation frameworks for failure discovery—DCBench [47] for slice discovery, HarmBench [132] for red-teaming—evaluate methods against known ground-truth failures. This creates a fundamental circularity: the purpose of gap discovery is finding failures that are *not yet known*, but evaluation assumes that the set of failures is known in advance.

This circularity raises several open questions about recall. How should one measure recall when the full set of gaps is unknown? The case study (Section 6.4) illustrates this in miniature: the CLIP and LLM diagnostics each flagged naming issues, but without RENOVATE’s ground truth there would be no way to know what fraction of real problems was caught—and RENOVATE’s ground truth covers 116 of 150 classes. Possible approaches include planted-failure protocols, where known failures are hidden in evaluation data and discovery rates are measured, and coverage estimation techniques analogous to species-richness estimation in ecology. Inter-rater reliability compounds the difficulty:

Johnson et al. [89] found that only about 60% of practitioner-generated hypotheses about discovered failure groups were correct, questioning whether current methods actually help practitioners form correct mental models of model behaviour. More broadly, evaluating gap discovery may be more like evaluating scientific insight than measuring benchmark accuracy—the field may need methodology evaluation (does this process reliably produce valid hypotheses?) rather than purely outcome evaluation (did this method find the ground-truth failures?).

With the meaningfulness criteria of the preceding subsection in hand, the completeness problem—colloquially, the “definition of DONE”—can be stated precisely. A model can be considered “sufficiently generalized under specified conditions” when it has been tested against all *meaningful* gaps within a defined scope—where meaningfulness is judged by ecological validity, structural insight, and actionability. OOD generalization is ill-defined without both gap discovery and a notion of completeness: finding gaps is necessary, but without a criterion for when the discovery process has covered sufficiently many dimensions of shift, one can neither claim that a model generalizes nor that it does not. Adversarial methods can surface failures mechanically; the hard question is knowing when to stop. Completeness is thus not a fixed threshold but a function of the deployment context: what counts as sufficient generalization for a medical imaging system differs from what counts for a photo-tagging application. Formalizing this context-dependent completeness criterion—rather than leaving it implicit in benchmark selection—is among the deepest open questions for the field.

#### 6.5.4 What do we do about it: From gaps to invariants

Finally, discovering a gap is necessary but not sufficient—it must be connected to remediation. For **data-distributional gaps** (the model has not seen this kind of input), the fix often follows from discovery: collect data, retrain. For **structural gaps** (missing capabilities such as temporal reasoning or social modeling), more data of the same kind does not help. Current failure discovery methods cannot reliably distinguish between these two cases, though interpretability-first approaches such as PRIME [166] and LADDER [61] are beginning to provide richer diagnoses.

The strongest empirical finding across the failure discovery literature is that human-AI collaborative methods consistently outperform fully automated ones. Human judgment provides ontological richness (naming dimensions absent from the model’s representation), value alignment (judging which failures matter), and analogical reasoning (predicting where similar failures occur in different domains). The human role may shift from specifying distributions to specifying objectives, but remains essential. Near-term, the realistic loop is: system finds the gap, human defines the bridging invariant, method is designed accordingly. This already improves on the status quo where humans must also find the gap.



## 6.6 Concluding Remarks

OOD generalization requires a full stack: discovering gaps, identifying invariants, and building bridges. This thesis contributed primarily to the bridge-construction half of that stack—identifying and exploiting bridging invariants through Chapters 3–5. This chapter mapped out the gap-discovery half as the complementary piece needed to complete the picture. The retrospective case study (Section 6.4) demonstrated that human–AI collaboration can validate a hypothesized gap rapidly and cheaply, but also that the most consequential step—choosing which dimension to examine—remains a human contribution.

Four open problems define the research frontier. They form a logical chain in which each depends on the resolution of the previous one. First, **the limits of discovery**: understanding what each representational paradigm can and cannot discover determines which gaps remain structurally invisible—and therefore which dimensions of shift no tool can currently reach. Given these limits, **the meaningfulness problem** asks which of the discoverable gaps are worth bridging; without criteria for ecological validity, structural insight, and actionability, gap discovery produces volume rather than understanding. With meaningfulness defined, **completeness** (colloquially, the “definition of DONE”) can be stated precisely: when has a model been tested against sufficiently many meaningful gaps to warrant a claim of generalization under specified conditions? Finally, **the gap-to-invariant pipeline** closes the practical loop: connecting a discovered, meaningful, and sufficiently validated gap to the bridging invariant that would close it—demanding richer diagnoses that distinguish data-distributional from structural gaps and that inform the design of targeted remediation.

The shift from hand-crafted to learned features transformed computer vision by turning feature design from artisanal craft into systematic methodology. An analogous transformation is underway for gap discovery. This thesis contributed to the current paradigm of crossing known gaps with bridging invariants; this chapter charted the remaining territory needed for a complete OOD methodology.

# 7 Conclusion

## 7.1 Summary of Contributions

This thesis addressed the challenge of Out-of-Distribution generalization in visual perception by identifying and exploiting *bridging invariants*—structural properties that hold on both sides of a distributional gap and enable the transfer of knowledge from familiar to novel conditions.

In Chapter 3, we tackled the gap between known and novel object categories in open-world detection. The bridging invariant was *geometric structure*: depth and surface normals capture the shape and spatial extent of objects independently of their semantic class. We introduced GOOD, a pseudo-labeling pipeline that trains an object proposal network on geometric cues from monocular estimators to discover unannotated novel objects in the training set. These pseudo-labels augment RGB-based detector training, enabling detection of objects that appearance-based methods systematically miss. Training with only a single “person” class, GOOD surpassed state-of-the-art methods by 5.0% AR@100 on non-person classes, demonstrating that geometric invariants provide a powerful foundation for open-world generalization.

In Chapter 4, we addressed the misalignment between category names in segmentation benchmarks and the actual visual content they describe. The bridging invariant was *vision-language alignment*: the shared structure between human visual categorization and the representations learned by vision-language models, which holds when objects are named according to common perceptual norms. RENOVATE uses image captioning models to generate contextually relevant candidate names and trains a specialized renaming model to select the best match for each segmentation mask. Training with renovated names yielded up to 15% relative improvement in open-vocabulary segmentation. Models trained on  $10^4$  images with renovated names matched those trained on  $5 \times 10^4$  images with original names, showing that the quality of the semantic interface can matter as much as the quantity of training data.

In Chapter 5, we bridged the spatial gap between the low-resolution features of Vision Foundation Models and the requirements of dense prediction tasks. The bridging invariant was *boundary and region structure*: object boundaries and region membership are resolution-independent properties of the visual scene. LoftUp constructs pseudo-groundtruth high-resolution features using class-agnostic SAM masks, then trains a coordinate-based cross-attention transformer to upsample features while respecting these boundaries. With less than 20% additional parameters, LoftUp outperformed existing feature upsamplers by 10–20% on most downstream tasks and achieved over 35% improvement on video object segmentation.

Across all three chapters, the pattern was the same: identify the invariant, extract it using

a foundation model, convert it into pseudo-labels, and use those pseudo-labels to guide the task-specific model. Foundation models served as *carriers* of bridging invariants—practical tools for accessing structural regularities that hold across distributional shifts—but the conceptual contribution in each case was in recognizing *which* invariant bridges the gap.

## 7.2 The Bias of the Bridge

Every bridging invariant is a hypothesis about what is shared across a distributional gap. When that hypothesis is correct, the bridge enables generalization. When it is wrong—when the “invariant” actually does shift, or when its source introduces its own distortions—the bridge fails. As introduced in Section 1.3, three conditions can cause a bridging invariant to fail: (i) the structural property assumed to be shared actually shifts between training and deployment; (ii) the foundation model that provides the invariant introduces its own distortions; (iii) the invariant is task-relevant for only a subset of the test distribution. Each of the three bridges in this thesis exhibits these failure modes in concrete, predictable ways.

**Geometric bias (GOOD).** The geometric bridge relies on the quality of monocular depth and normal estimators. These estimators are trained on diverse but finite 3D scene data, and their predictions can fail on textureless surfaces, unusual materials, extreme viewpoints, or adversarial lighting conditions. When the geometric signal is weak or inaccurate, the pseudo-labels degrade—and with them, the bridge. Furthermore, geometric structure is not equally informative for all object categories: objects defined primarily by appearance rather than shape (e.g., flat road signs, printed patterns) may not benefit from geometric cues. The invariant “all objects have geometric structure” is approximately true, but the approximation breaks in predictable ways.

**Semantic bias (RENOVATE).** The semantic bridge inherits the biases of the foundation models that generate and evaluate candidate names. GPT-4’s world model reflects the distribution of its training data, which is neither culturally neutral nor taxonomically complete—object naming conventions vary across cultures, languages, and domains. CLIP’s visual-linguistic alignment similarly reflects internet-scale image-text pairs, which overrepresent certain visual concepts and underrepresent others. These biases propagate into the renovated names: the bridge assumes that “correct” naming has a stable, universal definition, but in practice, what counts as the right name depends on context, audience, and purpose. The bridge is only as unbiased as its carrier.

**Spatial bias (LoftUp).** The spatial bridge depends on SAM’s class-agnostic mask quality. When SAM produces poor masks—on out-of-domain images, ambiguous textures, or scenes where “object” is not well-defined—the pseudo-groundtruth features are corrupted, and the upsampler learns incorrect boundary patterns. More fundamentally, the bridge assumes that *boundary structure* is the right invariant for all dense prediction tasks. This

**Table 7.1: Summary of bridge biases.** Each bridging invariant introduces predictable failure modes inherited from its source.

| Bridge            | Bias source         | When it fails                              | Example                     |
|-------------------|---------------------|--------------------------------------------|-----------------------------|
| Geometric (Ch. 3) | Monocular estimator | Textureless surfaces, unusual materials    | Road signs, thin structures |
| Semantic (Ch. 4)  | CLIP training data  | Culturally specific or rare concepts       | Non-Western categories      |
| Spatial (Ch. 5)   | SAM mask quality    | Ambiguous boundaries, out-of-domain images | Smooth gradients, fog       |

assumption is well-supported for tasks like segmentation and object tracking, where boundaries are directly relevant. But for tasks where the relevant structure is smooth rather than edge-aligned—depth estimation in gradually varying scenes, or optical flow in textureless regions—crisp boundary priors may introduce artifacts rather than help.

**A general principle.** These per-chapter limitations share a deeper structure: *a bridging invariant imports the biases of the source from which it is extracted*. A monocular depth estimator trained on indoor scenes may produce poor geometric bridges for aerial imagery. A vision-language model trained on English web data may produce biased semantic bridges for non-Western visual cultures. A segmentation model trained on natural images may produce unreliable spatial bridges for medical or satellite imagery.

This is not a flaw specific to our approach—it is an inherent property of any method that uses external knowledge to guide generalization. The external source provides invariance over a broad but finite range of conditions, and beyond that range, the invariance assumption breaks. The honest response is to *characterize* the biases of external invariants: to understand when and where each bridge is likely to fail. This connects directly to the argument of Chapter 6: systematic gap discovery is needed not only to identify the primary distributional gaps that motivate bridges, but also to identify the secondary gaps—the biases—that bridges themselves introduce.

## 7.3 From Bridging to Discovery

This thesis contributed to the bridge-construction half of OOD generalization; Chapter 6 mapped out the gap-discovery half that completes the stack. The biases that bridging invariants introduce (discussed above) reinforce the need for systematic gap discovery: these biases are themselves distributional gaps—secondary mismatches that require the same kind of methodology to identify. Chapter 6 poses four cascading questions—what can we find, which gaps matter, when have we found enough, and what do we do about it—and demonstrates through a retrospective case study that structurally different diagnostic tools surface complementary failure dimensions, providing an empirical foundation for that analysis.

## 7.4 Final Thoughts

A full stack of OOD generalization requires both building bridges and knowing where to build them. This thesis contributed primarily to bridge construction—identifying and exploiting bridging invariants—while also charting the landscape of gap discovery as the complementary upstream challenge. Geometric structure, semantic alignment, and boundary structure each provided a foundation for a specific gap, enabling substantial improvements in open-world detection, open-vocabulary segmentation, and dense feature upsampling.

Bridging invariants make explicit the structural assumptions that underlie OOD methods. The quality of generalization depends on the quality of these assumptions: a bridge built on a correct invariant enables robust transfer; a bridge built on a shifting property fails in predictable ways. Making these assumptions explicit—rather than leaving them implicit in architecture choices or training procedures—is itself a contribution to the field’s clarity.

As foundation model capabilities expand and gap discovery methodology matures, the field moves toward systematic OOD generalization. The deepest open question is the “definition of DONE”: when has a model been tested against enough distributional shifts, across enough dimensions of variation, to warrant a claim of sufficient generalization? This question connects meaningfulness, evaluation, and the ontological barrier into a single challenge. Until the field can answer it, OOD generalization remains an open-ended aspiration rather than a verifiable property.

# Bibliography

- [1] K. Ahuja, E. Caballero, D. Zhang, J.-C. Gagnon-Audet, Y. Bengio, I. Mitliagkas, and I. Rish. “Invariance Principle Meets Information Bottleneck for Out-of-Distribution Generalization”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2021.
- [2] Anthropic. *Claude Sonnet 4*. <https://www.anthropic.com/claude>. Accessed: 2025-05-01. 2025.
- [3] E. Arazo, D. Ortego, P. Albert, N. E. O’Connor, and K. McGuinness. “Pseudo-Labeling and Confirmation Bias in Deep Semi-Supervised Learning”. In: *IEEE International Joint Conference on Neural Networks (IJCNN)*. 2020.
- [4] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. “Multiscale Combinatorial Grouping”. In: *Computer Vision and Pattern Recognition*. 2014.
- [5] P. Arbeláez. “Boundary extraction in natural images using ultrametric contour maps”. In: *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW’06)*. IEEE. 2006, pp. 182–182.
- [6] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. “Contour Detection and Hierarchical Image Segmentation”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 33.5 (May 2011), pp. 898–916. ISSN: 0162-8828. DOI: [10.1109/TPAMI.2010.161](https://doi.org/10.1109/TPAMI.2010.161). URL: <http://dx.doi.org/10.1109/TPAMI.2010.161>.
- [7] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz. “Invariant Risk Minimization”. In: *arXiv preprint arXiv:1907.02893* (2019).
- [8] Y. Balaji, S. Sankaranarayanan, and R. Chellappa. “MetaReg: Towards Domain Generalization using Meta-Regularization”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2018.
- [9] H. Bao, L. Dong, S. Piao, and F. Wei. “BEiT: BERT Pre-Training of Image Transformers”. In: *International Conference on Learning Representations (ICLR)*. 2022.
- [10] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua. “Towards open-set identity preserving face synthesis”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2018.
- [11] L. Bégin, P. Germain, F. Laviolette, and J.-F. Roy. “PAC-Bayesian Theory for Transductive Learning”. In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*. 2014.
- [12] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. “A theory of learning from different domains”. In: *Machine Learning* 79.1–2 (2010), pp. 151–175.

- [13] Y. Bengio, A. Courville, and P. Vincent. “Representation Learning: A Review and New Perspectives”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2013), pp. 1798–1828.
- [14] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al. “On the Opportunities and Risks of Foundation Models”. In: *arXiv preprint arXiv:2108.07258* (2021).
- [15] M. Bucher, T.-H. Vu, M. Cord, and P. Pérez. “Zero-shot semantic segmentation”. In: *NeurIPS*. 2019.
- [16] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. “nuScenes: A Multimodal Dataset for Autonomous Driving”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.
- [17] H. Caesar, J. Uijlings, and V. Ferrari. “Coco-stuff: Thing and stuff classes in context”. In: *CVPR*. 2018, pp. 1209–1218.
- [18] N. Carion, L. Gustafson, Y.-T. Hu, S. Debnath, R. Hu, D. Suris, C. Ryali, K. V. Alwala, H. Khedr, A. Huang, J. Lei, T. Ma, B. Guo, A. Kalla, M. Marks, J. Greer, M. Wang, P. Sun, R. Rädle, T. Afouras, E. Mavroudi, K. Xu, T.-H. Wu, Y. Zhou, L. Momeni, R. Hazra, S. Ding, S. Vaze, F. Porcher, F. Li, S. Li, A. Kamath, H. K. Cheng, P. Dollár, N. Ravi, K. Saenko, P. Zhang, and C. Feichtenhofer. “SAM 3: Segment Anything with Concepts”. In: *arXiv preprint arXiv:2511.16719* (2025).
- [19] M. Caron, H. Touvron, I. Misra, H. Jegou, J. Mairal, P. Bojanowski, and A. Joulin. “Emerging Properties in Self-Supervised Vision Transformers”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021.
- [20] D. C. Castro, I. Walker, and B. Glocker. “Causality Matters in Medical Imaging”. In: *Nature Communications* 11.1 (2020), p. 3673.
- [21] S. Changpinyo, P. Sharma, N. Ding, and R. Soricut. “Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts”. In: *CVPR*. 2021.
- [22] B. Chen, A. Sax, F. Lewis, I. Armeni, S. Savarese, A. Zamir, J. Malik, and L. Pinto. “Robust Policies via Mid-Level Visual Representations: An Experimental Study in Manipulation and Navigation”. In: *Proceedings of the 2020 Conference on Robot Learning*. Ed. by J. Kober, F. Ramos, and C. Tomlin. Vol. 155. Proceedings of Machine Learning Research. PMLR, 16–18 Nov 2021, pp. 2328–2346. URL: <https://proceedings.mlr.press/v155/chen21f.html>.
- [23] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin. “MMDetection: Open MMLab Detection Toolbox and Benchmark”. In: *arXiv preprint arXiv:1906.07155* (2019).



- [24] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. “DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2018).
- [25] Y.-C. Chen, L. Li, L. Yu, A. El Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu. “Uniter: Universal image-text representation learning”. In: *ECCV*. 2020.
- [26] Y. Chen, S. Liu, and X. Wang. “Learning continuous image representation with local implicit image function”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 8628–8638.
- [27] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar. “Masked-attention Mask Transformer for Universal Image Segmentation”. In: *CVPR*. 2022.
- [28] J. Cheng, Y. Lu, X. Gu, P. Ke, X. Liu, Y. Dong, H. Wang, J. Tang, and M. Huang. “AutoDetect: Towards a Unified Framework for Automated Weakness Detection in Large Language Models”. In: *Findings of the Association for Computational Linguistics: EMNLP 2024*. 2024.
- [29] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan. “YOLO-World: Real-Time Open-Vocabulary Object Detection”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024.
- [30] S. Cho, H. Shin, S. Hong, S. An, S. Lee, A. Arnab, P. H. Seo, and S. Kim. “CAT-Seg: Cost Aggregation for Open-Vocabulary Semantic Segmentation”. In: *arXiv preprint arXiv:2303.11797* (2023).
- [31] Y. Chung, T. Kraska, N. Polyzotis, K. H. Tae, and S. E. Whang. “Slice Finder: Automated Data Slicing for Model Validation”. In: *IEEE International Conference on Data Engineering (ICDE)*. 2019, pp. 1550–1553.
- [32] A. Conti, E. Fini, M. Mancini, P. Rota, Y. Wang, and E. Ricci. “Vocabulary-free Image Classification”. In: *arXiv preprint arXiv:2306.00917* (2023).
- [33] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. “The Cityscapes Dataset for Semantic Urban Scene Understanding”. In: *CVPR*. 2016.
- [34] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le. “Autoaugment: Learning augmentation strategies from data”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 113–123.
- [35] A. D’Amour, K. Heller, D. Moldovan, B. Adlam, B. Alipanahi, A. Beutel, C. Chen, J. Deaton, J. Eisenstein, M. D. Hoffman, et al. “Underspecification Presents Challenges for Credibility in Modern Machine Learning”. In: *Journal of Machine Learning Research* 23.226 (2022), pp. 1–61.
- [36] N. Dalal and B. Triggs. “Histograms of Oriented Gradients for Human Detection”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2005, pp. 886–893.

- [37] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. “ImageNet: A Large-Scale Hierarchical Image Database”. In: *CVPR09*. 2009.
- [38] Z. Ding, J. Wang, and Z. Tu. “Open-Vocabulary Universal Image Segmentation with MaskCLIP”. In: *ICML*. 2023.
- [39] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *International Conference on Learning Representations (ICLR)*. 2021.
- [40] J. C. Duchi, P. W. Glynn, and H. Namkoong. “Statistics of Robust Optimization: A Generalized Empirical Likelihood Approach”. In: *Mathematics of Operations Research* 46.3 (2021), pp. 946–969.
- [41] C. E. Duchon. “Lanczos filtering in one and two dimensions”. In: *Journal of Applied Meteorology* (1979), pp. 1016–1022.
- [42] A. Eftekhari, A. Sax, J. Malik, and A. Zamir. “Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 10786–10796.
- [43] D. Eigen and R. Fergus. “Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2650–2658.
- [44] D. Eigen, C. Puhrsch, and R. Fergus. “Depth map prediction from a single image using a multi-scale deep network”. In: *Advances in neural information processing systems*. Vol. 27. 2014.
- [45] M. El Banani, A. Raj, K.-K. Maninis, A. Kar, Y. Li, M. Rubinstein, D. Sun, L. Guibas, J. Johnson, and V. Jampani. “Probing the 3d awareness of visual foundation models”. In: *CVPR*. 2024, pp. 21795–21806.
- [46] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. “The Pascal Visual Object Classes (VOC) Challenge”. In: *International Journal of Computer Vision* 88.2 (June 2010), pp. 303–338.
- [47] S. Eyuboglu, M. Varma, K. Saab, J.-B. Delbrouck, C. Lee-Messer, J. Dunnmon, J. Zou, and C. Ré. “Domino: Discovering Systematic Errors with Cross-Modal Embeddings”. In: *International Conference on Learning Representations (ICLR)*. 2022.
- [48] C. Fang, Y. Xu, and D. N. Rockmore. “Unbiased Metric Learning: On the Utilization of Multiple Datasets and Web Images for Softening Bias”. In: *IEEE International Conference on Computer Vision (ICCV)*. 2013, pp. 1657–1664.
- [49] Y. Fang, W. Wang, B. Xie, Q. Sun, L. Wu, X. Wang, T. Huang, X. Wang, and Y. Cao. “EVA: Exploring the Limits of Masked Visual Representation Learning at Scale”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023.

- [50] S. Feng, H. Sun, X. Yan, H. Zhu, Z. Zou, S. Shen, and H. X. Liu. “Dense Reinforcement Learning for Safety Validation of Autonomous Vehicles”. In: *Nature* 615 (2023), pp. 620–627.
- [51] D. J. Fremont, T. Dreossi, S. Ghosh, X. Yue, A. L. Sangiovanni-Vincentelli, and S. A. Seshia. “Scenic: A Language for Scenario Specification and Scene Generation”. In: *ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*. 2019, pp. 63–78.
- [52] S. Fu, M. Hamilton, L. E. Brandt, A. Feldmann, Z. Zhang, and W. T. Freeman. “FeatUp: A Model-Agnostic Framework for Features at Any Resolution”. In: *ICLR*. 2024. URL: <https://openreview.net/forum?id=GkJiNn2QDF>.
- [53] S. Y. Gadre, G. Ilharco, A. Fang, J. Hayase, G. Smyrnis, T. Nguyen, R. Marten, M. Wortsman, D. Ghosh, J. Zhang, et al. “Datacomp: In search of the next generation of multimodal datasets”. In: *NeurIPS*. 2023.
- [54] Y. Ganin and V. S. Lempitsky. “Unsupervised Domain Adaptation by Backpropagation”. In: *Proc. of the International Conf. on Machine learning (ICML)*. 2015.
- [55] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. “Domain-Adversarial Training of Neural Networks”. In: *Journal of Machine Learning Research* 17.59 (2016), pp. 1–35.
- [56] I. Gao, G. Ilharco, S. Lundberg, and M. T. Ribeiro. “Adaptive Testing of Computer Vision Models”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2023, pp. 4003–4014.
- [57] A. Geiger, P. Lenz, and R. Urtasun. “Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.
- [58] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann. “Shortcut learning in deep neural networks”. In: *Nature Machine Intelligence* 2.11 (2020), pp. 665–673.
- [59] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel. “ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness.” In: *International Conference on Learning Representations*. 2019. URL: <https://openreview.net/forum?id=Bygh9j09KX>.
- [60] G. Ghiasi, X. Gu, Y. Cui, and T.-Y. Lin. “Scaling open-vocabulary image segmentation with image-level labels”. In: *ECCV*. 2022.
- [61] S. Ghosh, R. Syed, C. Wang, V. Choudhary, B. Li, C. B. Poynton, S. Visweswaran, and K. Batmanghelich. “LADDER: Language-Driven Slice Discovery and Error Rectification in Vision Classifiers”. In: *Findings of the Association for Computational Linguistics: ACL 2025*. 2025.
- [62] C. Godard, O. Mac Aodha, and G. J. Brostow. “Unsupervised Monocular Depth Estimation with Left-Right Consistency”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

- [63] Y. Goldberg and O. Levy. “word2vec Explained: deriving Mikolov et al.’s negative-sampling word-embedding method”. In: *arXiv preprint arXiv:1402.3722* (2014).
- [64] S. Goyal, P. Maini, Z. C. Lipton, A. Raghunathan, and J. Z. Kolter. “The Science of Data Filtering: Data Curation cannot be Compute Agnostic”. In: *ICLR 2024 Workshop on Navigating and Addressing Data Problems for Foundation Models*. 2024.
- [65] X. Gu, L. Tsung-Yi, W. Kuo, and Y. Cui. “Open-vocabulary Object Detection via Vision and Language Knowledge Distillation”. In: *ICLR*. 2022.
- [66] I. Gulrajani and D. Lopez-Paz. “In Search of Lost Domain Generalization”. In: *International Conference on Learning Representations (ICLR)*. 2021.
- [67] M. Hamilton, Z. Zhang, B. Hariharan, N. Snavely, and W. T. Freeman. “Unsupervised semantic segmentation by distilling feature correspondences”. In: *arXiv preprint arXiv:2203.08414* (2022).
- [68] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. “Masked Autoencoders Are Scalable Vision Learners”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022.
- [69] K. He, J. Sun, and X. Tang. “Guided image filtering”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 35.6 (2013), pp. 1397–1409.
- [70] K. He, X. Zhang, S. Ren, and J. Sun. “Deep Residual Learning for Image Recognition”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.
- [71] Z. He, X. Li, J. Tang, S. Qiu, W. Wang, X. Xue, and J. Pu. “Towards Camera Open-set 3D Object Detection for Autonomous Driving Scenarios”. In: *IEEE Transactions on Intelligent Transportation Systems* (2025). arXiv:2406.17297.
- [72] D. Hendrycks and T. Dietterich. “Benchmarking Neural Network Robustness to Common Corruptions and Perturbations”. In: *International Conference on Learning Representations (ICLR)*. 2019.
- [73] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. “beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework”. In: *International Conference on Learning Representations (ICLR)*. 2017.
- [74] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell. “CyCADA: Cycle-Consistent Adversarial Domain Adaptation”. In: *Proc. of the International Conf. on Machine learning (ICML)*. 2018.
- [75] D. Hoiem, A. A. Efros, and M. Hebert. “Automatic photo pop-up”. In: *ACM SIG-GRAPH 2005 Papers*. 2005, pp. 577–584.
- [76] D. Hoiem, A. A. Efros, and M. Hebert. “Geometric context from a single image”. In: *Tenth IEEE International Conference on Computer Vision (ICCV’05) Volume 1*. Vol. 1. IEEE. 2005, pp. 654–661.

- [77] D. Hoiem, A. A. Efros, and M. Hebert. “Putting objects in perspective”. In: *International Journal of Computer Vision* 80.1 (2008), pp. 3–15.
- [78] D. Hoiem, A. A. Efros, and M. Hebert. “Recovering surface layout from an image”. In: *International Journal of Computer Vision* 75.1 (2007), pp. 151–172.
- [79] H. Huang, J. van Amersfoort, and Y. Gal. “Decomposing Representations for Deterministic Uncertainty Estimation”. In: *arXiv preprint arXiv:2112.00856* (2021).
- [80] H. Huang, S. Peng, D. Zhang, and A. Geiger. “Renovating Names in Open-Vocabulary Segmentation Benchmarks”. In: *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. 2024. URL: <https://openreview.net/forum?id=Uw2eJ0I822>.
- [81] M. Huh, B. Cheung, T. Wang, and P. Isola. “The Platonic Representation Hypothesis”. In: *arXiv preprint arXiv:2405.07987* (2024).
- [82] G. Ilharco, M. Wortsman, R. Wightman, C. Gordon, N. Carlini, R. Taori, A. Dave, V. Shankar, H. Namkoong, J. Miller, H. Hajishirzi, A. Farhadi, and L. Schmidt. *OpenCLIP*. Zenodo, <https://doi.org/10.5281/zenodo.5143773>. 2021.
- [83] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen, and K. H. Maier-Hein. “nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation”. In: *Nature Methods* 18 (2021), pp. 203–211.
- [84] A. Jabri, A. Owens, and A. Efros. “Space-time correspondence as a contrastive random walk”. In: *NeurIPS*. Vol. 33. 2020, pp. 19545–19560.
- [85] A. Jaiswal, Y. Wu, P. Natarajan, and P. Natarajan. “Class-agnostic object detection”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2021, pp. 919–928.
- [86] V. Jampani, K.-K. Maninis, A. Engelhardt, A. Karpur, K. Truong, K. Sargent, S. Popov, A. Araujo, R. Martin-Brualla, K. Patel, D. Vlasic, V. Ferrari, A. Makadia, C. Liu, Y. Li, and H. Zhou. “NAVI: Category-Agnostic Image Collections with High-Quality 3D Shape and Pose Annotations”. In: *NeurIPS*. 2023. URL: <https://navidataset.github.io/>.
- [87] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig. “Scaling up visual and vision-language representation learning with noisy text supervision”. In: *ICML*. 2021.
- [88] Y. Jiang, N. Walker, J. Hart, and P. Stone. “Open-world reasoning for service robots”. In: *Proceedings of the International Conference on Automated Planning and Scheduling*. Vol. 29. 2019, pp. 725–733.
- [89] N. Johnson, Á. A. Cabrera, G. Plumb, and A. Talwalkar. “Where Does My Model Underperform? A Human Evaluation of Slice Discovery Algorithms”. In: *AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*. 2023.

- [90] K. J. Joseph, S. Khan, F. S. Khan, and V. N. Balasubramanian. “Towards Open World Object Detection”. In: *arXiv:2103.02603 [cs]* (May 2021). arXiv: 2103.02603. URL: <http://arxiv.org/abs/2103.02603> (visited on 01/20/2022).
- [91] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, and S. Levine. “QT-Opt: Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation”. In: *Conference on Robot Learning (CoRL)*. 2018.
- [92] N. Kalra and S. M. Paddock. “Driving to Safety: How Many Miles of Driving Would It Take to Demonstrate Autonomous Vehicle Reliability?”. In: *Transportation Research Part A: Policy and Practice* 94 (2016), pp. 182–193.
- [93] O. F. Kar, T. Yeo, A. Atanov, and A. Zamir. “3D Common Corruptions and Data Augmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 18963–18974.
- [94] P. Kaul, W. Xie, and A. Zisserman. “Multi-Modal Classifiers for Open-Vocabulary Object Detection”. In: *ICML*. 2023.
- [95] D. Kiela, M. Bartolo, Y. Nie, D. Kaushik, A. Geiger, Z. Wu, B. Vidgen, G. Prasad, A. Singh, P. Ringshia, Z. Ma, T. Thrush, S. Riedel, Z. Waseem, P. Stenetorp, R. Jia, M. Bansal, C. Potts, and A. Williams. “Dynabench: Rethinking Benchmarking in NLP”. In: *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*. 2021, pp. 4110–4124.
- [96] D. Kim, T.-Y. Lin, A. Angelova, I. S. Kweon, and W. Kuo. “Learning Open-World Object Proposals without Learning to Classify”. In: *arXiv:2108.06753 [cs]* (Aug. 2021). arXiv: 2108.06753. URL: <http://arxiv.org/abs/2108.06753> (visited on 03/03/2022).
- [97] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. “Segment anything”. In: *ICCV*. 2023.
- [98] P. W. Koh, S. Sagawa, H. Marklund, S. M. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R. L. Phillips, I. Gao, et al. “WILDS: A Benchmark of in-the-Wild Distribution Shifts”. In: *International Conference on Machine Learning (ICML)*. 2021.
- [99] S. Konan, K. J. Liang, and L. Yin. “Extending One-Stage Detection with Open-World Proposals”. In: *arXiv preprint arXiv:2201.02302* (2022).
- [100] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele. “Joint bilateral upsampling”. In: *ACM TOG* 26.3 (2007), 96–es.
- [101] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown. “Text classification algorithms: A survey”. In: *Information* (2019).
- [102] A. Krizhevsky, I. Sutskever, and G. E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2012.

- [103] D. Krueger, E. Caballero, J.-H. Jacobsen, A. Zhang, J. Binas, D. Zhang, R. Le Priol, and A. Courville. “Out-of-Distribution Generalization via Risk Extrapolation (REx)”. In: *International Conference on Machine Learning (ICML)*. 2021.
- [104] W. Kuo, Y. Cui, X. Gu, A. Piergiovanni, and A. Angelova. “Open-Vocabulary Object Detection upon Frozen Vision and Language Models”. In: *ICLR*. 2023.
- [105] M. Lan, C. Chen, Y. Ke, X. Wang, L. Feng, and W. Zhang. “ProxyCLIP: Proxy Attention Improves CLIP for Open-Vocabulary Segmentation”. In: *ECCV*. 2024.
- [106] D.-H. Lee. “Pseudo-Label: The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks”. In: *ICML Workshop on Challenges in Representation Learning*. 2013.
- [107] S. Lee, M. Kim, L. Cherif, D. Dobre, J. Lee, S. J. Hwang, K. Kawaguchi, G. Gidel, Y. Bengio, N. Malkin, and M. Jain. “Learning Diverse Attacks on Large Language Models for Robust Red-Teaming and Safety Tuning”. In: *International Conference on Learning Representations (ICLR)*. 2025.
- [108] M. Ley. “The DBLP Computer Science Bibliography: Evolution, Research Issues, Perspectives”. In: *String Processing and Information Retrieval (SPIRE)*. Springer, 2002, pp. 1–10.
- [109] D. Li, Y. Yang, Y. Song, and T. M. Hospedales. “Deeper, Broader and Artier Domain Generalization”. In: *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*. 2017.
- [110] J. Li, D. Li, S. Savarese, and S. Hoi. “BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models”. In: *Proc. of the International Conf. on Machine learning (ICML)*. 2023.
- [111] L. Li, W. Shao, W. Dong, Y. Tian, Q. Zhang, K. Yang, and W. Zhang. “Data-Centric Evolution in Autonomous Driving: A Comprehensive Survey of Big Data System, Data Mining, and Closed-Loop Technologies”. In: *arXiv preprint arXiv:2401.12888* (2024).
- [112] M. Li, W. Wang, F. Feng, F. Zhu, Q. Wang, and T.-S. Chua. “Think twice before assure: Confidence estimation for large language models through reflection on multiple answers”. In: *arXiv preprint arXiv:2403.09972* (2024).
- [113] Y. Li, H. Mao, R. Girshick, and K. He. “Exploring plain vision transformer backbones for object detection”. In: *European conference on computer vision*. Springer. 2022, pp. 280–296.
- [114] Z. Li, J. Schultz, D. Hennes, and M. Lanctot. “Discovering Multiagent Learning Algorithms with Large Language Models”. In: *arXiv preprint arXiv:2602.16928* (2026).
- [115] F. Liang, B. Wu, X. Dai, K. Li, Y. Zhao, H. Zhang, P. Zhang, P. Vajda, and D. Marculescu. “Open-vocabulary semantic segmentation with mask-adapted clip”. In: *CVPR*. 2023.



- [116] K.-E. Lin, Y.-C. Lin, W.-S. Lai, T.-Y. Lin, Y.-C. Shih, and R. Ramamoorthi. “Vision Transformer for NeRF-Based View Synthesis From a Single Input Image”. In: *Proc. of the IEEE Winter Conference on Applications of Computer Vision (WACV)*. Jan. 2023, pp. 806–815.
- [117] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. “Microsoft COCO: Common Objects in Context”. In: *ECCV*. Ed. by D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars. 2014, pp. 740–755.
- [118] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. van der Laak, B. van Ginneken, and C. I. Sánchez. “A survey on deep learning in medical image analysis”. In: *Medical image analysis* 42 (2017), pp. 60–88.
- [119] B. Liu, E. Robertson, S. Grigsby, and S. Mazumder. “Self-Initiated Open World Learning for Autonomous AI Agents”. In: *arXiv preprint arXiv:2110.11385* (2021).
- [120] Q. Liu, Z. Xu, G. Bertasius, and M. Niethammer. “Simpleclick: Interactive image segmentation with simple vision transformers”. In: *ICCV*. 2023, pp. 22290–22300.
- [121] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, and L. Zhang. “Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection”. In: *arXiv preprint arXiv:2303.05499* (2023).
- [122] Y. Liu, I. E. Zulfikar, J. Luiten, A. Dave, D. Ramanan, B. Leibe, A. Ošep, and L. Leal-Taixé. “Opening up Open World Tracking”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 19045–19055.
- [123] J. Long, E. Shelhamer, and T. Darrell. “Fully Convolutional Networks for Semantic Segmentation”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.
- [124] I. Loshchilov and F. Hutter. “Decoupled weight decay regularization”. In: *arXiv preprint arXiv:1711.05101* (2017).
- [125] D. G. Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”. In: *International Journal of Computer Vision* 60.2 (2004), pp. 91–110.
- [126] H. Lu, W. Liu, H. Fu, and Z. Cao. “FADE: Fusing the assets of decoder and encoder for task-agnostic upsampling”. In: *ECCV*. Springer. 2022, pp. 231–247.
- [127] H. Lu, W. Liu, Z. Ye, H. Fu, Y. Liu, and Z. Cao. “SAPA: Similarity-aware point affiliation for feature upsampling”. In: *NeurIPS*. Vol. 35. 2022, pp. 20889–20901.
- [128] Y. Luo, L. Zheng, T. Guan, J. Yu, and Y. Yang. “Taking a Closer Look at Domain Shift: Category-Level Adversaries for Semantics Consistent Domain Adaptation”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2019.

- [129] C. Ma, Y. Yuhuan, C. Ju, F. Zhang, Y. Zhang, and Y. Wang. “AttrSeg: Open-Vocabulary Semantic Segmentation via Attribute Decomposition-Aggregation”. In: *Advances in Neural Information Processing Systems*. Ed. by A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine. Vol. 36. Curran Associates, Inc., 2023, pp. 10258–10270. URL: [https://proceedings.neurips.cc/paper\\_files/paper/2023/file/2093ed77c549eda95bd6f7212b735b43-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2023/file/2093ed77c549eda95bd6f7212b735b43-Paper-Conference.pdf).
- [130] M. Maaz, H. Rasheed, S. Khan, F. S. Khan, R. M. Anwer, and M.-H. Yang. “Class-agnostic Object Detection with Multi-modal Transformer”. In: *17th European Conference on Computer Vision*. Springer. 2022.
- [131] D. Martin, C. Fowlkes, D. Tal, and J. Malik. “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics”. In: *ICCV*. Vol. 2. IEEE. 2001, pp. 416–423.
- [132] M. Mazeika, L. Phan, X. Yin, A. Zou, Z. Wang, N. Mu, E. Sakhaee, N. Li, S. Basart, B. Li, D. Forsyth, and D. Hendrycks. “HarmBench: A Standardized Evaluation Framework for Automated Red Teaming and Robust Refusal”. In: *International Conference on Machine Learning (ICML)*. 2024.
- [133] D. A. McAllester. “PAC-Bayesian Model Averaging”. In: *Conference on Computational Learning Theory (COLT)*. 1999.
- [134] S. McKinley and M. Levine. “Cubic spline interpolation”. In: *College of the Redwoods* 45.1 (1998), pp. 1049–1060.
- [135] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. “Distributed representations of words and phrases and their compositionality”. In: *NeurIPS*. 2013.
- [136] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. “Nerf: Representing scenes as neural radiance fields for view synthesis”. In: *Communications of the ACM* 65.1 (2021), pp. 99–106.
- [137] M. Minderer, A. Gritsenko, and N. Houlsby. “Scaling Open-Vocabulary Object Detection”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2023.
- [138] M. Minderer, A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen, et al. “Simple Open-Vocabulary Object Detection with Vision Transformers”. In: *arXiv preprint arXiv:2205.06230* (2022).
- [139] K. Muandet, D. Balduzzi, and B. Schölkopf. “Domain Generalization via Invariant Feature Representation”. In: *Proc. of the International Conf. on Machine learning (ICML)*. 2013.
- [140] A. Novikov, N. Vű, M. Eisenberger, E. Dupont, P.-S. Huang, A. Z. Wagner, S. Shirobokov, B. Kozlovskii, F. J. R. Ruiz, A. Mehrabian, M. P. Kumar, A. See, S. Chaudhuri, G. Holland, A. Davies, S. Nowozin, P. Kohli, and M. Balog. “AlphaEvolve: A Coding Agent for Scientific and Algorithmic Discovery”. In: *arXiv preprint arXiv:2506.13131* (2025).

- [141] L. Oakden-Rayner, J. Dunnmon, G. Carneiro, and C. Ré. “Hidden Stratification Causes Clinically Meaningful Failures in Machine Learning for Medical Imaging”. In: *ACM Conference on Health, Inference, and Learning (CHIL)*. 2020.
- [142] A. Odena, V. Dumoulin, and C. Olah. “Deconvolution and checkerboard artifacts”. In: *Distill* 1.10 (2016), e3.
- [143] OpenAI. “GPT-4 technical report”. In: *arXiv* (2023).
- [144] OpenAI. *GPT-4o System Card*. <https://openai.com/index/gpt-4o-system-card/>. Accessed: 2025-05-01. 2024.
- [145] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. HAZIZA, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski. “DINOv2: Learning Robust Visual Features without Supervision”. In: *TMLR* (2024). Featured Certification. ISSN: 2835-8856. URL: <https://openreview.net/forum?id=a68SUt6zFt>.
- [146] L. Pan, M. Saxon, W. Xu, D. Nathani, X. Wang, and W. Y. Wang. “Automatically correcting large language models: Surveying the landscape of diverse self-correction strategies”. In: *arXiv preprint arXiv:2308.03188* (2023).
- [147] S. Parisot, Y. Yang, and S. McDonagh. “Learning to Name Classes for Vision and Language Models”. In: *CVPR*. 2023.
- [148] J. Park, H. Kim, Y. Tai, M. S. Brown, and I. Kweon. “High quality depth map upsampling for 3D-TOF cameras”. In: *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*. 2011.
- [149] S. Peng, K. Genova, C. ” Jiang, A. Tagliasacchi, M. Pollefeys, and T. Funkhouser. “OpenScene: 3D Scene Understanding with Open Vocabularies”. In: *CVPR*. 2023.
- [150] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. “A benchmark dataset and evaluation methodology for video object segmentation”. In: *CVPR*. 2016, pp. 724–732.
- [151] E. Perez, S. Huang, F. Song, T. Cai, R. Ring, J. Aslanides, A. Glaese, N. McAleese, and G. Irving. “Red Teaming Language Models with Language Models”. In: *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2022.
- [152] J. Peters, P. Bühlmann, and N. Meinshausen. “Causal inference by using invariant prediction: identification and confidence intervals”. In: *Journal of the Royal Statistical Society: Series B* 78.5 (2016), pp. 947–1012.
- [153] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool. “The 2017 DAVIS Challenge on Video Object Segmentation”. In: *arXiv:1704.00675* (2017).
- [154] V. Prabhu, S. Yenamandra, P. Chattopadhyay, and J. Hoffman. “LANCE: Stress-testing Visual Models by Generating Language-guided Counterfactual Images”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2023.

- [155] J. Priem, H. Piwowar, and R. Orr. “OpenAlex: A fully-open index of scholarly works, authors, venues, institutions, and concepts”. In: *arXiv preprint arXiv:2205.01833* (2022).
- [156] G. Puy, S. Gidaris, A. Boulch, O. Siméoni, C. Sautier, P. Pérez, A. Bursuc, and R. Marlet. “Three Pillars Improving Vision Foundation Model Distillation for Lidar”. In: *CVPR*. June 2024, pp. 21519–21529.
- [157] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. “Learning transferable visual models from natural language supervision”. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 8748–8763.
- [158] R. Ranftl, A. Bochkovskiy, and V. Koltun. “Vision Transformers for Dense Prediction”. In: *ICCV*. Oct. 2021, pp. 12179–12188.
- [159] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun. “Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot Cross-Dataset Transfer”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.3 (2022).
- [160] M. Ranzinger, G. Heinrich, J. Kautz, and P. Molchanov. “AM-RADIO: Agglomerative Vision Foundation Model Reduce All Domains Into One”. In: *CVPR*. June 2024, pp. 12490–12500.
- [161] H. Rasheed, M. Maaz, S. Shaji, A. Shaker, S. Khan, H. Cholakkal, R. M. Anwer, E. Xing, M.-H. Yang, and F. S. Khan. “Glamm: Pixel grounding large multimodal model”. In: *CVPR*. 2024, pp. 13009–13018.
- [162] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer. “SAM 2: Segment Anything in Images and Videos”. In: *arXiv preprint arXiv:2408.00714* (2024). URL: <https://arxiv.org/abs/2408.00714>.
- [163] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. “You Only Look Once: Unified, Real-Time Object Detection”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.
- [164] S. Ren, K. He, R. Girshick, and J. Sun. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett. Vol. 28. Curran Associates, Inc., 2015. URL: <https://proceedings.neurips.cc/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf>.
- [165] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, Z. Zeng, H. Zhang, F. Li, J. Yang, H. Li, Q. Jiang, and L. Zhang. *Grounded SAM: Assembling Open-World Models for Diverse Visual Tasks*. 2024. arXiv: [2401.14159](https://arxiv.org/abs/2401.14159) [cs.CV].

- [166] K. Rezaei, M. Saberi, M. Moayeri, and S. Feizi. “PRIME: Prioritizing Interpretability in Failure Mode Extraction”. In: *arXiv preprint arXiv:2310.00164* (2023).
- [167] M. T. Ribeiro and S. Lundberg. “Adaptive Testing and Debugging of NLP Models”. In: *Annual Meeting of the Association for Computational Linguistics (ACL)*. 2022.
- [168] M. T. Ribeiro, T. Wu, C. Guestrin, and S. Singh. “Beyond Accuracy: Behavioral Testing of NLP Models with CheckList”. In: *Annual Meeting of the Association for Computational Linguistics (ACL)*. 2020, pp. 4902–4912.
- [169] M. N. Rizve, K. Duarte, Y. S. Rawat, and M. Shah. “In Defense of Pseudo-Labeling: An Uncertainty-Aware Pseudo-label Selection Framework for Semi-Supervised Learning”. In: *International Conference on Learning Representations (ICLR)*. 2021.
- [170] O. Ronneberger, P. Fischer, and T. Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. 2015.
- [171] E. Rosch, C. B. Mervis, W. D. Gray, D. M. Johnson, and P. Boyes-Braem. “Basic objects in natural categories”. In: *Cognitive psychology* 8.3 (1976), pp. 382–439.
- [172] E. H. Rosch. “Natural categories”. In: *Cognitive psychology* 4.3 (1973), pp. 328–350.
- [173] C. Rother, V. Kolmogorov, and A. Blake. “GrabCut: Interactive Foreground Extraction Using Iterated Graph Cuts”. In: *ACM TOG* 23.3 (2004), pp. 309–314.
- [174] S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang. “Distributionally Robust Neural Networks for Group Shifts: On the Importance of Regularization for Worst-Case Generalization”. In: *International Conference on Learning Representations (ICLR)*. 2020.
- [175] K. Saito, P. Hu, T. Darrell, and K. Saenko. “Learning to Detect Every Thing in an Open World”. In: *arXiv:2112.01698 [cs]* (Dec. 2021). arXiv: 2112.01698 version: 1. URL: <http://arxiv.org/abs/2112.01698> (visited on 01/30/2022).
- [176] O. Salaudeen, H. Zhang, K. Alhamoud, S. Beery, and M. Ghassemi. “Aggregation Hides Out-of-Distribution Generalization Failures from Spurious Correlations”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2025.
- [177] M. Samvelyan, S. C. Raparthy, A. Lupu, E. Hambro, A. H. Markosyan, M. Bhatt, Y. Mao, M. Jiang, J. Parker-Holder, J. Foerster, T. Rocktäschel, and R. Raileanu. “Rainbow Teaming: Open-Ended Generation of Diverse Adversarial Prompts”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2024.
- [178] A. Sauer and A. Geiger. “Counterfactual Generative Networks”. In: *International Conference on Learning Representations*. 2021. URL: <https://openreview.net/forum?id=BXewfAYMmJw>.

- [179] A. Saxena, S. Chung, and A. Ng. “Learning depth from single monocular images”. In: *Advances in neural information processing systems* 18 (2005).
- [180] A. Saxena, S. H. Chung, and A. Y. Ng. “3-d depth reconstruction from a single still image”. In: *International journal of computer vision* 76.1 (2008), pp. 53–69.
- [181] A. Saxena, M. Sun, and A. Y. Ng. “Make3d: Learning 3d scene structure from a single still image”. In: *IEEE transactions on pattern analysis and machine intelligence* 31.5 (2009), pp. 824–840.
- [182] B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio. “Toward Causal Representation Learning”. In: *Proceedings of the IEEE* 109.5 (2021), pp. 612–634.
- [183] C. Schuhmann, R. Vencu, R. Beaumont, R. Kaczmarczyk, C. Mullis, A. Katta, T. Coombes, J. Jitsev, and A. Komatsuzaki. “Laion-400m: Open dataset of clip-filtered 400 million image-text pairs”. In: *arXiv preprint arXiv:2111.02114* (2021).
- [184] Z. Shen, J. Liu, Y. He, X. Zhang, R. Xu, H. Yu, and P. Cui. “Towards Out-of-Distribution Generalization: A Survey”. In: *arXiv preprint arXiv:2108.13624* (2021).
- [185] J. Shi and J. Malik. “Normalized cuts and image segmentation”. In: *IEEE Transactions on pattern analysis and machine intelligence* 22.8 (2000), pp. 888–905.
- [186] Y. Shi, J. Seely, P. H. Torr, N. Siddharth, A. Hannun, N. Usunier, and G. Synnaeve. “Gradient Matching for Domain Generalization”. In: *International Conference on Learning Representations (ICLR)*. 2022.
- [187] R. Shwartz-Ziv and N. Tishby. “Opening the Black Box of Deep Neural Networks via Information”. In: *arXiv preprint arXiv:1703.00810* (2017).
- [188] O. Siméoni, H. V. Vo, M. Seitzer, F. Baldassarre, M. Oquab, C. Jose, V. Khalidov, M. Szafraniec, S. Yi, M. Ramamonjisoa, F. Massa, D. Haziza, L. Wehrstedt, J. Wang, T. Darcet, T. Moutakanni, L. Sentana, C. Roberts, A. Vedaldi, J. Tolan, J. Brandt, C. Couprie, J. Mairal, H. Jégou, P. Labatut, and P. Bojanowski. “DINOv3”. In: *arXiv preprint arXiv:2508.10104* (2025).
- [189] A. Singh, R. Hu, V. Goswami, G. Couairon, W. Galuba, M. Rohrbach, and D. Kiela. “Flava: A foundational language and vision alignment model”. In: *CVPR*. 2022.
- [190] E. E. Smith and D. L. Medin. *Categories and concepts*. Harvard University Press, 1981.
- [191] K. Sofiiuk, I. A. Petrov, and A. Konushin. “Reviving iterative training with mask guidance for interactive segmentation”. In: *ICIP*. IEEE. 2022, pp. 3141–3145.
- [192] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li. “Fixmatch: Simplifying semi-supervised learning with consistency and confidence”. In: *Advances in neural information processing systems*. Vol. 33. 2020, pp. 596–608.
- [193] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai. “VL-BERT: Pre-training of Generic Visual-Linguistic Representations”. In: *ICLR*. 2020.

- [194] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, et al. “Scalability in Perception for Autonomous Driving: Waymo Open Dataset”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.
- [195] Q. Sun, Y. Fang, L. Wu, X. Wang, and Y. Cao. “EVA-CLIP: Improved Training Techniques for CLIP at Scale”. In: *arXiv preprint arXiv:2303.15389* (2023).
- [196] S. Suri, M. Walmer, K. Gupta, and A. Shrivastava. “Lift: A surprisingly simple lightweight feature transform for dense vit descriptors”. In: *ECCV*. Springer. 2024, pp. 110–128.
- [197] S. Szymanowicz, C. Rupprecht, and A. Vedaldi. “Splatter Image: Ultra-Fast Single-View 3D Reconstruction”. In: *CVPR*. June 2024, pp. 10208–10217.
- [198] A. Takmaz, E. Fedele, R. W. Sumner, M. Pollefeys, F. Tombari, and F. Engelmann. “OpenMask3D: Open-Vocabulary 3D Instance Segmentation”. In: *NeurIPS*. 2023.
- [199] R. Taori, A. Dave, V. Shankar, N. Carlini, B. Recht, and L. Schmidt. “Measuring Robustness to Natural Distribution Shifts in Image Classification”. In: *Advances in Neural Information Processing Systems*. Vol. 33. 2020, pp. 18583–18599.
- [200] Z. Tian, C. Shen, H. Chen, and T. He. “FCOS: Fully Convolutional One-Stage Object Detection”. In: *Proc. Int. Conf. Computer Vision (ICCV)*. 2019.
- [201] N. Tishby, F. C. Pereira, and W. Bialek. “The Information Bottleneck Method”. In: *arXiv preprint physics/0004057* (2000).
- [202] M. Tschannen, A. Gritsenko, X. Wang, M. F. Naeem, I. Alabdulmohsin, N. Parthasarathy, T. Evans, L. Beyer, Y. Xia, B. Mustafa, et al. “SigLIP 2: Multilingual Vision-Language Encoders with Improved Semantic Understanding, Localization, and Dense Features”. In: *arXiv preprint arXiv:2502.14786* (2025).
- [203] N. Tumanyan, A. Singer, S. Bagon, and T. Dekel. “Dino-tracker: Taming dino for self-supervised point tracking in a single video”. In: *ECCV*. Springer. 2024, pp. 367–385.
- [204] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. “Attention is all you need”. In: *NeurIPS*. 2017.
- [205] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan. “Deep Hashing Network for Unsupervised Domain Adaptation”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 5018–5027.
- [206] E. Vin, S. Kashiwa, M. Rhea, D. J. Fremont, E. Kim, T. Dreossi, S. Ghosh, X. Yue, A. L. Sangiovanni-Vincentelli, and S. A. Seshia. “3D Environment Modeling for Falsification and Beyond with Scenic 3.0”. In: *Computer Aided Verification (CAV)*. 2023, pp. 253–265.
- [207] R. Volpi, P. Morerio, S. Savarese, and V. Murino. “Adversarial Feature Augmentation for Unsupervised Domain Adaptation”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2018.



- [208] Q. Wang, J. Gao, and X. Li. “Weakly Supervised Adversarial Domain Adaptation for Semantic Segmentation in Urban Scenes”. In: *IEEE Trans. on Image Processing (TIP)* 28 (2019).
- [209] W. Wang, M. Feiszli, H. Wang, J. Malik, and D. Tran. “Open-World Instance Segmentation: Exploiting Pseudo Ground Truth From Learned Pairwise Affinity”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 4422–4432.
- [210] Y. Wang, X. Shen, S. X. Hu, Y. Yuan, J. L. Crowley, and D. Vaufreydaz. “Self-supervised transformers for unsupervised object discovery using normalized cut”. In: *CVPR*. 2022, pp. 14543–14553.
- [211] Y. Wang, H. Chen, Q. Heng, W. Hou, Y. Fan, Z. Wu, J. Wang, M. Savvides, T. Shinnozaki, B. Raj, B. Schiele, and X. Xie. “FreeMatch: Self-adaptive Thresholding for Semi-supervised Learning”. In: *International Conference on Learning Representations (ICLR)*. 2023.
- [212] Y. Weng, M. Zhu, F. Xia, B. Li, S. He, S. Liu, B. Sun, K. Liu, and J. Zhao. “Large language models are better reasoners with self-verification”. In: *arXiv preprint arXiv:2212.09561* (2022).
- [213] M. Wysoczańska, O. Siméoni, M. Ramamonjisoa, A. Bursuc, T. Trzciński, and P. Pérez. “CLIP-DINOiser: Teaching CLIP a few DINO tricks for open-vocabulary semantic segmentation”. In: *ECCV*. Springer. 2024, pp. 320–337.
- [214] Y. Xian, J. Chibane, B. L. Bhatnagar, B. Schiele, Z. Akata, and G. Pons-Moll. “Any-Shot GIN: Generalizing Implicit Networks for Reconstructing Novel Classes”. In: *2022 International Conference on 3D Vision (3DV)*. IEEE. 2022.
- [215] Y. Xian, S. Choudhury, Y. He, B. Schiele, and Z. Akata. “Semantic projection network for zero-and few-label semantic segmentation”. In: *CVPR*. 2019.
- [216] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le. “Self-training with noisy student improves imagenet classification”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 10687–10698.
- [217] S. Xie and Z. Tu. “Holistically-nested edge detection”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1395–1403.
- [218] J. Xu, S. Liu, A. Vahdat, W. Byeon, X. Wang, and S. De Mello. “Open-Vocabulary Panoptic Segmentation with Text-to-Image Diffusion Models”. In: *CVPR*. 2023.
- [219] M. Xu, Z. Zhang, H. Hu, J. Wang, L. Wang, F. Wei, X. Bai, and Z. Liu. “End-to-End Semi-Supervised Object Detection with Soft Teacher”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021.
- [220] M. Xu, Z. Zhang, F. Wei, H. Hu, and X. Bai. “Side adapter network for open-vocabulary semantic segmentation”. In: *CVPR*. 2023.
- [221] M. Xu, Z. Zhang, F. Wei, Y. Lin, Y. Cao, H. Hu, and X. Bai. “A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model”. In: *ECCV*. 2022.

- [222] X. Xu, T. Xiong, Z. Ding, and Z. Tu. “MasQCLIP for Open-Vocabulary Universal Image Segmentation”. In: *ICCV*. 2023.
- [223] J. Yang, K. Zhou, Y. Li, and Z. Liu. “Generalized Out-of-Distribution Detection: A Survey”. In: *International Journal of Computer Vision* 132 (2024), pp. 4667–4698.
- [224] Y. Yang, C. Ma, C. Ju, F. Zhang, J. Yao, Y. Zhang, and Y. Wang. “Multi-modal Prototypes for Open-World Semantic Segmentation”. In: *International Journal of Computer Vision* (2024), pp. 1–17.
- [225] Y. Yang, H. Zhang, D. Katabi, and M. Ghassemi. “Change is Hard: A Closer Look at Subpopulation Shift”. In: *International Conference on Machine Learning (ICML)*. 2023.
- [226] Z. Yang, M. Ding, T. Huang, Y. Cen, J. Song, B. Xu, Y. Dong, and J. Tang. “Does Negative Sampling Matter? A Review with Insights into its Theory and Applications”. In: *IEEE TPAMI* (2024).
- [227] J. Ye, N. Wang, and X. Wang. “FeatureNeRF: Learning Generalizable NeRFs by Distilling Foundation Models”. In: *ICCV*. Oct. 2023, pp. 8962–8973.
- [228] N. Ye, K. Li, H. Bai, R. Yu, L. Hong, F. Zhou, Z. Li, and J. Zhu. “OoD-Bench: Quantifying and Understanding Two Dimensions of Out-of-Distribution Generalization”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022.
- [229] A. Yu, V. Ye, M. Tancik, and A. Kanazawa. “pixelNeRF: Neural Radiance Fields from One or Few Images”. In: *CVPR*. 2021.
- [230] H. Yu, J. Liu, X. Zhang, J. Wu, and P. Cui. “A Survey on Evaluation of Out-of-Distribution Generalization”. In: *arXiv preprint arXiv:2403.01874* (2024).
- [231] Q. Yu, J. He, X. Deng, X. Shen, and L.-C. Chen. “Convolutions die hard: Open-vocabulary segmentation with single frozen convolutional clip”. In: *Advances in Neural Information Processing Systems*. Vol. 36. 2023, pp. 32215–32234.
- [232] Z. Yu, S. Peng, M. Niemeyer, T. Sattler, and A. Geiger. “MonoSDF: Exploring Monocular Geometric Cues for Neural Implicit Surface Reconstruction”. In: *Advances in Neural Information Processing Systems (NeurIPS)* (2022).
- [233] Z. Yuan, T. Wei, S. Cheng, G. Zhang, Y. Chen, and H. Xu. “Learning to Manipulate Anywhere: A Visual Generalizable Framework for Reinforcement Learning”. In: *arXiv preprint arXiv:2407.15815* (2024).
- [234] A. R. Zamir, A. Sax, N. Cheerla, R. Suri, Z. Cao, J. Malik, and L. J. Guibas. “Robust learning through cross-task consistency”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 11197–11206.
- [235] A. Zareian, K. D. Rosa, D. H. Hu, and S.-F. Chang. “Open-Vocabulary Object Detection Using Captions”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2021.
- [236] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer. “Sigmoid loss for language image pre-training”. In: *arXiv preprint arXiv:2303.15343* (2023).

- [237] C.-B. Zhang, J. Ni, Y. Zhong, and K. Han. “v-CLR: View-Consistent Learning for Open-World Instance Segmentation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2025.
- [238] H. Zhang, F. Li, X. Zou, S. Liu, C. Li, J. Yang, and L. Zhang. “A simple framework for open-vocabulary segmentation and detection”. In: *ICCV*. 2023.
- [239] Y. Zhang, X. Huang, J. Ma, Z. Li, Z. Luo, Y. Xie, Y. Qin, T. Luo, Y. Li, S. Liu, et al. “Recognize Anything: A Strong Image Tagging Model”. In: *arXiv preprint arXiv:2306.03514* (2023).
- [240] H. Zhao, X. Puig, B. Zhou, S. Fidler, and A. Torralba. “Open vocabulary scene parsing”. In: *ICCV*. 2017.
- [241] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. “Scene parsing through ade20k dataset”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2017.
- [242] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba. “Semantic understanding of scenes through the ade20k dataset”. In: *International Journal of Computer Vision* 127.3 (2019), pp. 302–321.
- [243] H. Zhou, T. Shen, X. Yang, H. Huang, X. Li, L. Qi, and M.-H. Yang. “Re-thinking evaluation metrics of open-vocabulary segmentaion”. In: *arXiv preprint arXiv:2311.03352* (2023).
- [244] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy. “Domain Generalization: A Survey”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.4 (2022), pp. 4396–4415.
- [245] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang. “Domain Generalization with MixStyle”. In: *International Conference on Learning Representations (ICLR)*. 2021.