Privacy by Design: From Distributed Learning to Post-Deployment Risks

Dissertation

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

vorgelegt von
Arjhun Swaminathan
aus Chennai/Indien

Tübingen
2026

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

Tag der mündlichen Qualifikation:	17.07.2026
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatter:	Dr. Mete Akgün
2. Berichterstatter:	Prof. Dr.-Ing. Oliver Kohlbacher

To my wife Saradha

Acknowledgments

It would be remiss of me to not begin by acknowledging how much of these past three plus years rests on quiet unnoticed gifts, moments, vast amounts of privilege and luck that fell by my side, whether it was where and when I was born, or the generosity of my health that allowed me to engage in scientific research. For that, I am deeply grateful, and a lot of this, I naturally owe to my parents.

Professionally, my gratitude extends to my supervisors, Dr. Mete Akgün, and Prof. Dr. Nico Pfeifer. Mete has provided me with immense flexibility with time to find new problems, do research and grow confidently as an independent researcher. My immense gratitude extends to Nico, who listened, was always supportive, and without whom, none of this would truly be possible. I am thankful to Agi for always treating me with kindness, and to Anika for being a good friend and colleague.

Somewhere along the way, in this blissful weird existence on a pale blue dot, as we try to understand the meaning and purpose of it all, we learn a simple thing: we are shaped — by people, by nature, by animals, by the places that hold us, and by everything we consume. We become, in part, what the world makes us feel. During this chaotic, stressful, lonely, and at times hopeless stretch of a doctoral thesis, there were people who kept me steady, sometimes unaware of doing it. I want to thank them for their kindness, the ways they protected my mind, and for the company that brought calm, clarity and laughter.

My in-laws have supported my professional life with nothing but warmth, especially when things were hard. Aaron's optimism has always managed to pull me back from despair, while Paul has always grounded me with practicality when I needed it most. Arjun has cared for me immensely, while Subbu has always been there for conversations that may be difficult to have with anyone else. Manish and Swathi have opened their homes, kitchens, and hearts for me more times than I can count, through thick and thin. Mara has brought colour, peace, and joy into my life in the most wondrous ways. My gratitude extends to Archana, Vineeth, Markus, Timo, Aiman, Anirudh, Kavita, Nadin, and many others, for the countless conversations that kept me afloat.

For all the unwavering support, love and guidance, I am eternally grateful to my brother, Aditya, whose presence has been a constant refuge. Anni, Dhru, and my new bestie, Vidhul, have reminded me what uncomplicated joy feels like.

And finally, or perhaps wholly, I cannot find the words, or the language to express the amount of love, care, support and cheerleading I have been fortunate to receive from the most extraordinary person I have ever met, my wife Saradha. It would be nothing but unfair to try to capture it here, but with her kindness in my life, a PhD was nothing but a breeze.

Arjhun Swaminathan

Abstract

Modern data processing activities increasingly rely on sensitive data, from medical images and genomic data to electronic health records, to enable research, discovery, diagnosis, and decision-making. Yet the very capabilities that make these systems powerful also create privacy risks: data must often be shared across institutions for generalization and accuracy, and once models and datasets are released for downstream use, they become long-lived artifacts that others can query, link, or exploit. *Privacy by design*, the principle that privacy should be a foundational property of any data processing activity rather than a post-incident patch, offers a response to these risks. However, translating this into concrete technical practice remains a challenge, because the right answer depends on where in the processing lifecycle one stands, what is being protected, and what assumptions about adversaries are relevant. This thesis approaches privacy by design as a technical agenda organized around two regimes: pre-deployment and post-deployment.

On the pre-deployment side, we develop privacy-preserving methods for computation on distributed data under semi-honest threat models. We introduce randomized-encoding based approaches for scalable kernel learning on medical images and for multi-site genome-wide association studies on quantitative phenotypes. We further show that widely used classical kernels can be realized through quantum feature maps, and introduce a distributed secure quantum architecture for kernel computation, validated on simulated quantum hardware.

On the post-deployment side, we study what deployed artifacts reveal and how that exposure can be exploited or mitigated. We introduce a targeted adversarial attack for hard-label black-box image classifiers that leverages edge information from images to accelerate attack progress under strict query budgets, consistently outperforming existing methods in the low-query regime across diverse architectures. We also develop a topological framework for adaptive k-anonymisation for dynamic datasets, enabling incremental updates to anonymised data releases without full recomputation when the underlying data changes.

Taken together, the results presented in this thesis demonstrate that privacy by design for data processing activities is not a single technique but a discipline whose realization spans a composition of architectures, threat models, and lifecycle stages.

Zusammenfassung

Moderne Datenverarbeitung arbeitet immer häufiger mit sensiblen Daten, zum Beispiel medizinischen Bildern, genomischen Daten oder elektronischen Gesundheitsakten. Diese Daten werden genutzt, um Forschung, neue Erkenntnisse, Diagnosen und Entscheidungen zu unterstützen. Gleichzeitig entstehen dadurch Datenschutzrisiken. Daten müssen oft zwischen verschiedenen Institutionen geteilt werden, damit Modelle besser generalisiert werden können und genauer werden. Wenn Modelle und Datensätze später veröffentlicht werden, bleiben sie lange verfügbar. Andere Personen können sie dann abfragen, verknüpfen oder missbrauchen. Privacy by Design ist das Prinzip, dass Datenschutz von Anfang an ein grundlegender Teil jeder Datenverarbeitung sein soll und nicht erst später hinzugefügt wird. Die praktische Umsetzung ist jedoch schwierig. Die passende Lösung hängt davon ab, an welcher Stelle im Lebenszyklus der Datenverarbeitung man sich befindet, was genau geschützt werden soll und welche Annahmen über mögliche Angreifer gelten. Diese Dissertation betrachtet Privacy by Design daher als eine technische Forschungsaufgabe mit zwei Bereichen: Pre-Deployment und Post-Deployment.

Im Bereich Pre-Deployment entwickeln wir datenschutzfreundliche Methoden für Berechnungen auf verteilten Daten unter einem Bedrohungsmodell mit semi-honesten Parteien. Wir stellen Ansätze auf Basis von Randomized Encodings für skalierbares Kernel-Lernen auf medizinischen Bildern sowie für standortübergreifende Genome-Wide Association Studies mit quantitativen Phänotypen vor. Außerdem zeigen wir, dass häufig verwendete klassische Kernel durch Quanten-Feature-Maps umgesetzt werden können. Wir formulieren auch eine verteilte sichere Quantenarchitektur für die Berechnung von Kerneln und testen sie auf simulierten Quantencomputern.

Im Bereich Post-Deployment untersuchen wir, welche Informationen veröffentlichte Modelle und Datensätze preisgeben und wie diese Informationen ausgenutzt oder geschützt werden können. Wir entwickeln einen gezielten adversarialen Angriff auf Hard-Label Black-Box Bildklassifikatoren. Der Angriff nutzt Kanteninformationen aus Bildern, um schneller Fortschritte zu machen, auch wenn nur wenige Anfragen an das Modell erlaubt sind. In diesem niedrigen Query-Bereich ist die Methode bei verschiedenen Architekturen besser als bestehende Ansätze. Außerdem entwickeln wir ein topologisches Framework für adaptive k-Anonymisierung von dynamischen Datensätzen. Dieses Framework ermöglicht inkrementelle Updates von anonymisierten Datenveröffentlichungen, ohne dass bei Änderungen der Daten alles neu berechnet werden muss.

Die Ergebnisse dieser Dissertation zeigen, dass Privacy by Design für Datenverarbeitung keine einzelne Technik ist. Es ist eine Disziplin, die verschiedene Architekturen, Bedrohungsmodelle und Phasen im Lebenszyklus der Datenverarbeitung kombiniert.

Contents

List of Figures

List of Tables

Acronyms

ADMM alternating direction method of multipliers. xv, 23

AI artificial intelligence. xv, 4

API application programming interface. xv, 9, 15, 26–29

DP differential privacy. xv, 8, 12, 13, 16, 22

EHR electronic health record. xv, 27

FHE fully homomorphic encryption. xv, 8, 12, 13, 22, 23

GDPR General Data Protection Regulation. xv, 7, 8

GWAS genome-wide association study. xv, 5, 14, 19, 23, 25

ML machine learning. xv, 10–12, 15, 24

PbD privacy by design. xv, 4, 5, 7, 8, 15, 17, 19, 25, 27–29, 31

PCA principal component analysis. xv, 22, 24, 25

PET privacy-enhancing technology. xv, 8, 9, 12, 19

QML quantum machine learning. xv, 10, 14, 24

RBF radial basis function. xv, 22, 24

RE randomized encoding. xv, 13, 22, 23, 25

SMPC secure multiparty computation. xv, 12, 13, 22, 23

SNP single-nucleotide polymorphism. xv, 13, 14, 23

SVM support vector machine. xv, 22, 24, 25

TDA topological data analysis. xv, 16, 27

ViT vision transformer. xv, 26

1 List of Publications

Accepted Publications

1. Hannemann, Anika*, **Swaminathan, Arjhun***, Ünal, Ali Burak, and Akgün, Mete. "Private, Efficient and Scalable Kernel Learning for Medical Image Analysis." *Computational Intelligence Methods for Bioinformatics and Biostatistics (CIBB 2024)*, Lecture Notes in Computer Science, vol. 15276, pp. 81–95. Springer, 2025. [1]

2. **Swaminathan, Arjhun**, Hannemann, Anika, Ünal, Ali Burak, Pfeifer, Nico, and Akgün, Mete. "PP-GWAS: Privacy Preserving Multi-Site Genome-wide Association Studies." *Nature Communications* 16 (2025): 11030. [2]

3. **Swaminathan, Arjhun** and Akgün, Mete. "Distributed and Secure Kernel-Based Quantum Machine Learning." *Transactions on Machine Learning Research* (April 2025). [3]

4. **Swaminathan, Arjhun** and Akgün, Mete. "Accelerating Targeted Hard-Label Adversarial Attacks in Low-Query Black-Box Settings." Accepted at the *IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)* 2026. arXiv:2505.16313. [4]

5. **Swaminathan, Arjhun** and Akgün, Mete. "Dynamic k-Anonymity for Electronic Health Records: A Topological Framework." *Computer Security. ESORICS 2024 International Workshops (ESORICS 2024)*, Lecture Notes in Computer Science, vol. 15263, pp. 137–152. Springer, 2025. [5]

———-

2 Introduction

With big data come big responsibilities.
— attributed to danah boyd and Kate Crawford [6].

We have always been a species of pattern-hunters. Long before we had microscopes or telescopes, we learned to read the world by noticing, remembering, and comparing. We did it to navigate, to heal, to predict weather and famine, to chart the stars and, eventually, to chart ourselves. What changed with the advent of the internet is not our appetite for evidence, but the scale and velocity of what we can capture. The "global datasphere", as it has been termed, is the sum of all data created, captured, or replicated, and it has been measured in zettabytes (1 ZB $\sim 10^{12}$ GB) for years now, with industry forecasts placing it at around 175 ZB in 2025, and climbing steeply beyond that [7]. In concrete terms, these estimates translate into hundreds of millions of terabytes created or consumed each day. In this datasphere, the data is not only about the outer world we live in. It is about bodies and biographies: medical images, clinical narratives, genomic variants, search queries, clicks, pauses, and attention. Human bodies and activity have become data we can collect, record, and in a way, harvest.

Somewhere between maintaining physical ledgers and maintaining databanks, we realized a second thing: relationships inside data can be inferred, and machines can do that inference at a scale, speed, and dimensionality that no single person or group of people can do. In that sense, learning from data is less a sudden invention than a continuation of a mathematical lineage that has been turning observation into prediction for over two centuries. One of its ancestral threads begins with Adrien-Marie Legendre's 1805 publication of the method of least squares, which serves as a foundational step toward regression as we now understand it [8]. This seminal work, motivated by concrete scientific problems, already contained a very modern idea: uncertainty can be handled. It offered a mathematically precise way to reduce a messy world to parameters. Regression and classification, kernel methods and neural networks, each offer a different answer to the same old problem: extract structure from noise, and then generalize to unseen data.

However, the moment the world becomes measurable, it also becomes optimisable. What becomes optimisable becomes exploitable, not as a moral failure of mathematics, but as

a predictable consequence of incentives. Data does not remain only a mirror held up to nature. It becomes a commodity and an infrastructure: the emergence of attention markets, personalization engines, and behavioural prediction products signals the same [9, 10]. Shoshana Zuboff's coining of "surveillance capitalism" captures this pivot: private experiences are rendered into behavioural data and sold into markets that trade in what we might do next [11]. The consequences are not abstract. For example, recommender systems shape what billions of people see [12] and third-party audits of YouTube's recommendation dynamics have examined how personalization can narrow exposure [13]. Further, when data-driven targeting meets political incentives, it can step directly on democratic processes, as the Cambridge Analytica revelations and the regulatory responses they triggered in both the UK and the EU made clear [14]. The recurring lesson is that data practices can become political infrastructure, and weak governance in that infrastructure is in itself a systemic risk. In response to growing concerns, privacy and governance frameworks have hardened into law and compliance practices [15, 16].

At the same time, the computational acceleration of the modern era ensures that regulation often lags behind innovation. It took millennia to move from the wheel to the steam engine. In contrast, the transformer architecture introduced in *Attention Is All You Need* appeared in 2017 [17], and in less than a decade, it has become the backbone of large language models and, increasingly, artificial intelligence (AI) agents. At this pace of progress, it becomes hard to ignore a basic asymmetry: technology iterates quickly, while law and democratic institutions move by deliberation, precedent, and consent. They are meant to. Modern constitutional democracies are designed to be slow where it matters, because slowness is often the price for legitimacy.

This regulatory lag does not absolve us of responsibility. If regulation and institutions move slowly because legitimacy demands it, then the burden shifts to the people who build and deploy modern inference systems today. Privacy cannot be treated as something we "add back" after a system is built, trained, deployed, and integrated into workflows. It has to be built in from the start, as a default rather than an exception. This idea, that privacy should be a foundational design property rather than a post-hoc patch, is the core of what has come to be called privacy by design (PbD) [18], a principle formalised in privacy literature and integral to data protection law. We then face a practical question: what does it mean to embed privacy into a system whose purpose is to infer? Inference systems typically convert datasets into parameters, and parameters into predictions that are then exposed through interfaces. PbD, in this sense, is not a single knob to turn in this complex, often multi-faceted picture. From a technical perspective relevant to this thesis, it is better understood as two complementary design obligations, each tied to a different phase in the lifecycle of a system. Before deployment, the question is what must be revealed for a computation to proceed, and how to ensure that nothing beyond that is made inferable through intermediate artifacts, logs, gradients, or other by-products of learning. After deployment, the question shifts: the artifact is already in the world, and privacy depends on what new attack surfaces emerge.

Both obligations are grounded in literature. In collaborative settings such as federated learning, gradients and model updates can behave like side channels, and in some cases can be used to reconstruct information about underlying inputs [19, 20, 21]. Likewise, modern

systems do not only fail by accident. In computer vision, adversarial examples showed that deep networks can be pushed into misclassification by small, carefully chosen perturbations [22]. In black-box hard-label settings, where an attacker may observe only the final predicted label, even a minimal query interface becomes an attack surface [23, 24, 25, 26]. Models are not the only artifacts at risk after deployment. Released datasets, even after direct identifiers are removed, can be linked against external information to re-identify individuals [27]. The common thread is that trust cannot be inferred from performance alone. It must be argued from the system's behaviour under adversarial pressure, from the privacy properties of the pipeline that produced it, and from the durability of those properties as data and models continue to be used over time.

This thesis approaches PbD as a technical agenda spanning both these regimes. On the pre-deployment side, we investigate techniques that enable inference on sensitive data distributed across institutions, including privacy-preserving frameworks for classical and quantum kernel-based methods and for multi-site genome-wide association studies (GWAS). On the post-deployment side, we study what deployed artifacts reveal and how that exposure can be exploited or mitigated: we examine vulnerability in query-based image classification models through adversarial attacks under strict access constraints, and we address data-focused risks through adaptive anonymisation for dynamic healthcare datasets, where the challenge is to maintain privacy guarantees as the underlying data evolves across repeated data releases. In the context of this thesis, pre-deployment privacy is about making collaboration possible without pooling raw records. Post-deployment privacy is about limiting what can be inferred, manipulated, or linked together from released artifacts. Both are necessary, because a system that is built in a privacy-preserving manner but is exposed after deployment, or is safe to query but built on improperly shared data, has not achieved PbD in any meaningful sense. Taken together, the manuscripts in this thesis show that PbD is not achieved by one universal mechanism, but by matching the privacy mechanism to the stage of a data processing lifecycle, the artifact exposed at that stage, and the trust and resource assumptions that remain relevant there.

3 Background and Preliminaries

3.1 Privacy by design

Privacy by design (PbD) is the idea that privacy should not be treated as a purely technical set of constraints during very specific points of data processing, or as an afterthought, but that it should carry a socio-technical stance and should be a vital design property that is imbued into the entire lifecycle of any data processing activity. Originally developed by Ann Cavoukian [18], the principle is adopted by data protection and privacy commissioners internationally [28]. In the context of European data protection law, the idea of PbD is reflected and legally formalised in Article 25 of the General Data Protection Regulation (GDPR) [15], titled *Data protection by design and default*.

Article 25 requires that data controllers (GDPR Art. 4(7)) implement both technical and organisational measures to safeguard the privacy of personal data (GDPR Art. 4(1)) throughout the lifecycle of any data processing activity. Notably, the legal text explicitly engages with technical measures, citing pseudonymisation as an example. This is also where Article 25's "by default" becomes concrete. Default configuration of any processing activity must ensure that only necessary personal data is processed, and that it is not made accessible to an indefinite number of persons or third parties. This raises a question: what exactly is personal data?

Under the GDPR, personal data includes any information relating to an identified or identifiable natural person, where identifiability depends on the means reasonably likely to be used. This is a moving target, as Recital 26 itself acknowledges, because what is a reasonable measure changes with technological advancements. For example, Sweeney [29] showed that ZIP code, date of birth, and sex together can uniquely identify a large fraction of individuals in the United States. Information from Strava that appeared non-sensitive in isolation was used to ascertain military base locations, illustrating how benign data such as exercise traces become sensitive when aggregated and published [30]. Similarly, the Netflix Prize incident showed that even when direct identifiers were removed, sparse high-dimensional behavioural data (ratings history) could enable re-identification of individuals when linked to publicly available auxiliary information (in this case, IMDB ratings) [27]. The advent of large language models trained on web-crawled corpora [31] complicates

the picture further, since data that was publicly accessible but contextually bounded can be memorized, recombined, and surfaced in unexpected contexts. Taken together, these examples show that the scope of personal data is not static, but shifts with the available means of linkage, inference, and reuse. This is precisely why privacy cannot be reduced to the removal of explicit identifiers or treated as a late-stage compliance exercise.

The GDPR is not the only instrument that advances the idea of PbD. The Organisation for Economic Co-operation and Development (OECD) Privacy Guidelines [32], the modernised Convention 108+ [33], and standards such as ISO 31700-1:2023 [34] all articulate similar principles, framing privacy through architectures, processes, and lifecycle practices rather than after-the-fact mitigation. The European Data Protection Board, which advises on interpreting the GDPR, further emphasises that measures to guarantee PbD can range from advanced technical controls to basic organisational practices such as governance initiatives and personnel training.

In this thesis, PbD functions as a basic principle which we approach from the technical perspective while acknowledging the multi-faceted nature of the problem. To make this technical stance concrete, the next section introduces privacy engineering as the discipline that translates PbD principles into architectural choices, threat models, and implementable controls.

3.1.1 Privacy Engineering

The European Union Agency for Cybersecurity (ENISA) describes data protection engineering (which we refer to as privacy engineering in this thesis) as a practical discipline that helps engineers select and apply measures in a risk-based way, in order to uphold data protection principles throughout the lifecycle of processing [35]. An important reality is that no single privacy solution fits all problems. System architectures fall into different categories depending on how many data holders participate, how often they interact and whether a third party server helps with computations. The choice of architecture shapes which privacy-enhancing technologies (PETs) are appropriate. Some measures are foundational and broadly applicable, such as encryption, key management, access control, and auditing [35]. Others are more specialised to particular settings, such as secure aggregation [36], multi-party computation [37], differential privacy (DP) [38], and fully homomorphic encryption (FHE) [39], among others, and depend heavily on the system setup, inputs, and computational capacities. In practice, privacy engineering is the art of choosing combinations of these building blocks so that a system remains functional while offering a meaningful reduction in privacy risks.

A second facet of privacy engineering is that it is not only a property of the system and its architecture, but also of who is involved, under what roles, and with what incentives. The actors in a data processing activity can include data providers, controllers, processors (GDPR Art. 4(8)), users, data subjects, engineers, system administrators, auditors, and many more. Privacy engineering therefore becomes closely tied to governance and access policies. Meanwhile, an attacker may be external, or may be one (or many) of these actors acting

outside their intended role, whether colluding or acting alone.

A third related facet is that the goals of an attacker are diverse, and whether a given adversary acts passively or actively cannot be assumed in advance. Sometimes, they want to link records across different datasets *(linkage)* [27], infer hidden attributes *(attribute inference)* [40], or determine the membership of an individual in a model's training dataset *(membership inference)* [41]. Sometimes, they might aim to reconstruct training data from shared gradients *(gradient inversion)* [19], or even replicate a proprietary model using only its outputs *(model extraction)* [42]. Sometimes the adversary is not an external attacker at all, but a legitimate participant in a protocol who might seek to learn the data of other data providers in a shared system *(reconstruction)* [19], or they may attempt to corrupt a globally trained model toward targeted misbehaviour *(model poisoning)* [43]. Even when the intended malicious goal is not privacy related and is instead aligned with broader security threats, the downstream consequences can still lead to privacy risks.

A fourth facet is that privacy-relevant assets may extend well beyond the obvious. What requires protection can include raw records, intermediate results, features, labels, metadata, logs, access traces, and model parameters. It can further include the interfaces through which these are exposed, such as application programming interfaces (APIs), auditing pipelines, and monitoring dashboards. Privacy failures may occur at the boundaries between these components, where information that seems harmless in isolation becomes revealing when combined with other signals. This is why privacy engineering guidance often reads like a catalogue of lifecycle controls, spanning data collection, storage, access, transmission, and retention, rather than focusing on a single algorithmic step. The examples discussed earlier (Census data, Strava, Netflix Prize) all illustrate the same engineering lesson. Privacy failures rarely arise from a single problematic cog in the machine, but from a composition of errors, both human and technical, both internal and external.

Privacy engineering is a multi-faceted problem.

It spans architectures, cryptographic techniques, operational controls, human processes, adversarial perspectives, and lifecycle stages.

3.2 Pre-deployment Privacy *(Manuscripts 1, 2 & 3)*

Much of the privacy discourse focuses on design-time choices, on the simple premise that what is not engineered into a system early is difficult to fix post-deployment. There are two reasons as to why. First, privacy protects data before deployment and user data after deployment across the lifecycle of a data processing activity. Second, privacy can enable better insights to be derived. If data providers refuse to share raw data for reasons such as contractual obligations, institutional policies, and jurisdictional or sectoral regulations, incorporating PETs can make it possible to learn from multiple sources without centralizing the data itself. Since combining datasets often improves robustness and accuracy in standard settings [44], privacy becomes not only a safety feature but also a way to build better models.

Our Manuscripts 1, 2 and 3 operate in this setting. Across all three works, the objective is to enable inference tasks in distributed environments where multiple data providers wish to perform computations jointly while keeping their data private, with the assistance of a third-party, often referred to as a *server*. We discuss some of their common preliminaries below.

3.2.1 Kernels *(Manuscripts 1 & 3)*

Kernel methods are among the relics of early machine learning (ML) research that remain widely used today. Their motivation is a geometric limitation: real-world data is often not linearly separable, in the sense that no single hyperplane in the original input space cleanly separates different classes of data. A mathematically motivated remedy is to map inputs into a different space in which an approximate linear divide becomes feasible for most data points, and to study data there. Kernel methods do this exactly, by allowing for inner products in a (typically high-dimensional) feature space to be computed implicitly, without ever constructing that space explicitly. This is often dubbed the "*kernel-trick*". Following the discussion in Manuscript 1 and in greater detail in Manuscript 3, we recall the formal definition.

Definition 1. *Kernel function [45]*
A *kernel function*, or *kernel*, K, is a map $K : X \times X \to \mathbb{C}$ that satisfies $K(x, y) = \langle \phi(x), \phi(y) \rangle$, where $\phi : X \to \mathscr{H}$ is a map from the input space X to a Hilbert space $(\mathscr{H}, \langle \cdot, \cdot \rangle)$.

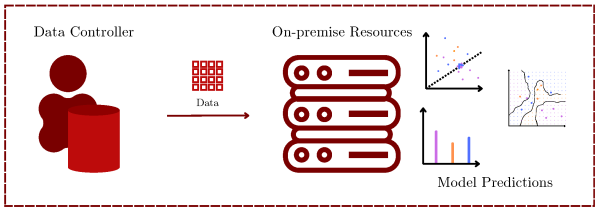
One refers to ϕ as a *feature map*. Different feature maps may induce the same kernel, so in practice the kernel itself is often the central object of interest. Kernel functions play a central role in many ML problems because a broad class of learning and inference tasks can be expressed in terms of pairwise kernel evaluations [46, 47]. This is one of the main reasons kernel methods remain attractive in modern ML, especially in settings involving high-dimensional or heterogeneous data.

In the context of this thesis, kernels are relevant for two reasons. First, in the classical setting, kernel computations can often be decomposed across parties, which makes them well suited to privacy-preserving and distributed learning scenarios where raw data cannot be centralised. Second, in quantum machine learning (QML), kernels admit a natural interpretation in terms of *quantum* feature maps. This makes it possible to study quantum analogues of commonly used kernels and to analyze how such kernel evaluations can be carried out in both centralised and distributed quantum settings. Manuscripts 1 and 3 build on these two perspectives.

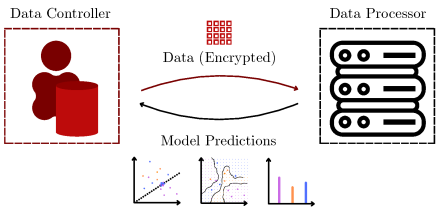
3.2.2 Distributed Architectures *(Manuscripts 1, 2 & 3)*

We distinguish here a few common architectures employed in a data processing pipeline. In a *centralised* architecture, data is collected and processed in one location under a single data controller. A closely related variant is *outsourced* processing, where a data controller relies on an external processor, such as a cloud provider, to execute processing tasks. In

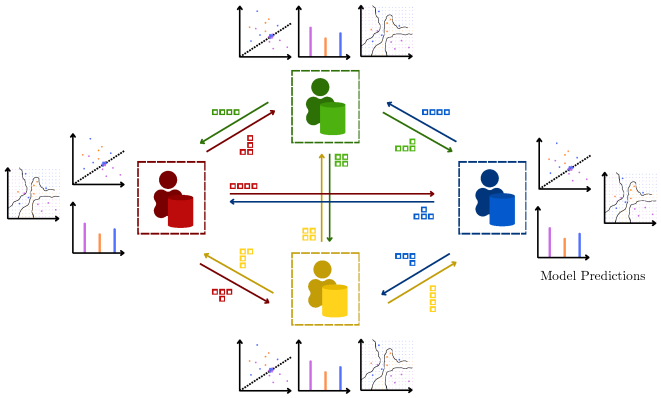
distributed architectures, multiple institutions jointly process data while keeping the data distributed across participants with the help of a third-party server. In *multi-party computation* architectures, multiple parties jointly process data like in distributed architectures but without the assistance of a server. *Federated learning* is an important special case of distributed computation where clients train ML models locally and, with the help of an external coordinating server, construct a global model that behaves as if it were trained on pooled data. These architectures are illustrated in Figure 3.1. This list is not exhaustive. Other variants such as Swarm Learning [48] also exist; here, we focus on the architectures that recur most frequently across the manuscripts included in this thesis.



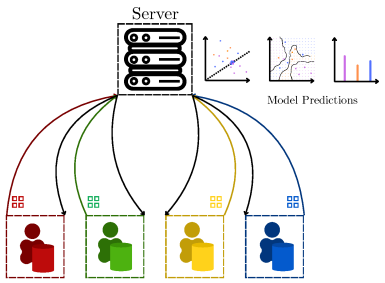
(a) Centralised processing.



(b) Outsourced processing.



(c) Multi-party computation.



(d) Distributed architecture.

Figure 3.1: Common architectures in ML pipelines.

In most ML tasks, larger and more diverse datasets improve model generalisation and reduce variance [44], making models less brittle to outlier data. This is even more important in distributed settings, where each data provider may observe only a narrow slice of the overall distribution. A model trained at a single site can overfit in practice by internalising site-specific artifacts, local controls, and preprocessing choices.

The choice of architecture determines not only performance and stability but also the threat model. In adversarial analyses, participants are often referred to as *semi-honest* (sometimes called honest-but-curious or passive) or *malicious* (sometimes called active). Semi-honest entities follow the prescribed protocol but attempt to infer additional information from what they observe, whereas malicious entities may deviate from the protocol and actively manipulate behaviour to achieve their goals.

In Manuscripts 1, 2 and 3, we adopt a distributed architecture where a central server coordinates computation. Across all three works, clients are allowed to collude with each other, and all entities, including the server, are modelled as semi-honest. This reflects a common real-world collaboration pattern in sensitive data analyses: institutions that wish to run joint analyses may be unable to centralise data and may also lack the computational resources required for large-scale tasks, leading them to outsource parts of the workflow to third-party cloud service providers.

3.2.3 Randomized Encoding *(Manuscripts 1 & 2)*

As mentioned earlier, privacy engineering is often an act of composition: selecting and combining PETs as building blocks for a specific task and a specific architecture. There are many techniques that offer different kinds and levels of privacy across different settings. We therefore do not attempt to cover the full landscape here, and instead focus on the PETs that are repeatedly mentioned throughout the manuscripts that make up this thesis.

Secure multiparty computation (SMPC) [37] is a PET in which multiple entities seek to compute an ML objective on distributed data. They do so by exchanging shares of secrets that enable computation across rounds of local operations and communication, with different tasks requiring different protocol constructions. SMPC is employed in many settings and offers a strongly decentralised model, where trust remains within a cohort of data controllers and processors. Two practical issues that are often discussed in the SMPC context arise from communication. On the one hand, SMPC can require interaction among all participating entities over several rounds, even for basic operations. On the other hand, this dense communication pattern induces a substantial setup cost: if an additional client joins the computation, they typically need to establish communication with each of the existing clients, which can be operationally cumbersome [49].

A common alternative that avoids frequent interaction and reduces setup complexity in distributed settings is FHE [39]. FHE is a cryptographic technique, enabled by tools from ring theory, that allows for computation on encrypted data such that decryption yields (approximately) the same result as if the computation had been performed on plaintext data (hence *homomorphic*). We say approximately because supporting complex operations efficiently has long been a challenge, and many practical schemes introduce approximation to make real-valued computation tractable (as noted, ring-based schemes naturally favour only two primary operations, such as addition and multiplication). Repeated application of homomorphic operations across long computational pipelines can lead to accumulated approximation error [50]. Moreover, homomorphic operations are computationally expensive and can introduce significant overhead. Nevertheless, the ability to outsource computation on encrypted data is powerful: in distributed settings, clients can encrypt their data and send it to a third-party server, which performs operations on data it cannot interpret, and returns encrypted outputs that clients can decrypt.

A parallel technique that is often treated as a gold standard in privacy is DP [38]. DP introduces carefully chosen operation-specific noise such that the outcome of a computa-

tion is nearly unchanged when any single data point is added or removed. This provides a mathematical guarantee that attacks such as membership inference, become difficult. However, the added noise can reduce utility, and this privacy–utility trade-off can be especially consequential in sensitive domains such as medicine and genomics, where small changes in accuracy may be clinically or scientifically meaningful [51].

Each technique carries its own costs and failure modes. SMPC can be constrained by communication requirements inherent in distributed topologies. FHE can be constrained by computational overhead, which becomes particularly challenging for high-dimensional data and complex pipelines. DP introduces noise that cannot be removed and persists as a form of inaccuracy. One approach proposed in sparse literature for privacy-preserving computation is randomized encoding (RE) [52, 53, 54, 55, 56, 57]. Like SMPC and DP, RE is operation-specific. It adds structured noise in a manner reminiscent of DP, but unlike DP the noise can be removed. RE is often adapted to architectures with a third-party server [54, 55, 56], although there are also constructions for multi-party settings [52, 53, 57].

In Manuscripts 1 and 2, we consider high-dimensional medical imaging data and genomic data, and focus on classification and association tasks, respectively. We adopt RE as a way to address practical hurdles of large-scale computation in distributed settings: clients add structured sparse noise to their data before sending it to a third-party server, the server performs computations on the obfuscated data, and the outputs can be decoded to recover the result as if the operations had been performed on noiseless data. Unlike FHE, which aims to hide essentially everything about the underlying values, RE is operation-specific and may reveal certain quantities or properties. Its privacy therefore depends on the original data distribution and on whether what is revealed by the encoded computation can be used to infer sensitive information about the underlying inputs or their properties. Following Manuscript 2, we describe RE formally below.

Definition 2. *Randomized encoding [2]*
Let $f : X \to Y$ be a function, where Y is equipped with a norm $\|\cdot\|$. A *randomized encoding* of f consists of a randomized function $\hat{f} : X \times \mathscr{R} \to \hat{Y}$, where $\mathscr{R}$ denotes the randomness space, and a deterministic decoder $\mathrm{Dec} : \hat{Y} \to Y$ such that

$$\left\| \mathrm{Dec}\big(\hat{f}(x;r)\big) - f(x) \right\| \leq \varepsilon$$

for all $x \in X$ and all $r \in \mathscr{R}$, for some very small $\varepsilon \geq 0$.

In other words, $\hat{f}$ injects structured randomness that conceals the input x while still allowing the value $f(x)$ to be recovered by the decoder.

3.2.4 Genome-Wide Association Studies *(Manuscript 2)*

A genome-wide association study (GWAS) [58] is a statistical approach for identifying associations between genetic variants and observable traits or diseases across a population. The genetic variants of primary interest are single-nucleotide polymorphisms (SNPs) that occur at specific positions in the genome and vary across individuals. A phenotype is any

measurable characteristic of an individual, whether quantitative (such as blood pressure or body mass index) or binary (such as disease presence or absence). The goal of a GWAS is to test, for each SNP across the genome, whether there is a statistically significant association between the variant and the phenotype of interest, after accounting for confounders such as population structure and relatedness.

Because the number of SNPs under consideration is typically large, often in the hundreds of thousands to millions, the computational and statistical demands of a GWAS are substantial even in centralised or non-private settings. This has motivated efficient algorithms such as REGENIE [59], which reduces the problem through a two-stage ridge regression procedure: a first stage compresses genome-wide information into a smaller set of blockwise predictions, and a second stage stacks these predictions to produce adjusted phenotype residuals against which individual SNPs are then tested. In Manuscript 2, we adapt this pipeline to a distributed and privacy-preserving setting using randomized encoding.

3.2.5 Quantum Kernel Computation *(Manuscript 3)*

Quantum computing [60] encodes information in quantum bits, or qubits, which can exist in superpositions of basis states $|0\rangle$ and $|1\rangle$. A system of n qubits spans a 2^n-dimensional state space, so that N-dimensional classical data that would require N classical bits can in principle be represented using $\lceil \log_2 N \rceil$ qubits. In QML, quantum encoding is a procedure that maps a classical input vector x to a quantum state $|\psi(x)\rangle$. This makes classical data accessible to quantum computation, and it allows similarities between inputs to be expressed through overlaps such as $\langle \psi(x), \psi(x') \rangle$, which can be estimated from measurements. Schuld and Killoran [61] formalised an observation that any quantum encoding $x \mapsto |\psi(x)\rangle$ acts as a feature map into a Hilbert space, and therefore induces a kernel. This insight opens the possibility of computing widely used classical kernels, such as polynomial, Gaussian, and Laplacian kernels, through appropriately designed quantum encodings that we introduce in Manuscript 3.

Further, in distributed settings where data is held by separate parties, quantum states can be transmitted between parties using quantum teleportation [62], a protocol that transfers the full quantum state of a qubit from one party to another using a shared entangled pair and classical communication, without physically transmitting the qubit itself. In Manuscript 3, we propose a mechanism to enable kernel computation when data is partitioned across sites: each party encodes its local data into quantum states, which are then teleported to a central server that performs the inner-product measurements needed for kernel evaluation.

3.3 Post-deployment Privacy *(Manuscripts 4 & 5)*

We now shift our attention to a different part of the data processing pipeline: *post-deployment privacy*. By this we mean the privacy questions that emerge once an artifact has been deployed, whether for use by end users, downstream services, or internal actors. Manuscripts 4 and 5 study two different post-deployment exposure surfaces: queryable models and

evolving released datasets.

3.3.1 Exposed models *(Manuscript 4)*

The dominant privacy conversation post model deployment concerns the user, namely what the system learns about them through their interactions. In Manuscript 4, however, we take a reversed and equally practical perspective: we assume the user may be malicious, and we ask what privacy and security properties the *model owner* is expected to preserve. A user may attempt to steal the model, infer sensitive information about its training data, or probe its decision behaviour to uncover vulnerabilities. A particularly relevant case is the one in which the model is accessible through a query interface.

From a PbD perspective, the interface is part of the system boundary. Even when it reveals only minimal outputs, repeated interaction can allow an adversary to infer actionable structure about the model, such as decision regions or local sensitivities. Consider an image classification model exposed through an API that returns only the top-1 label. This hard-label interface appears restrictive because it provides neither probabilities nor internal information, yet an attacker can still issue adaptive queries and use label changes to navigate the decision boundary [63]. Such interaction leads to the construction of adversarial examples, including targeted examples that aim to force a specific misclassification under a small perturbation budget. Formally, consider a classifier $f : [0,1]^{C\times H\times W} \to \mathbb{R}^K$ over images with C colour channels and spatial dimensions $H \times W$, where the attacker observes only the predicted label $\hat{y}(x) = \arg\max_k [f(x)]_k$. Given a source image x_s classified as y_s and a target image x_t classified as a different label y_t, a targeted attack seeks

$$x_{\text{adv}} = \arg\min_x \|x - x_s\|_2, \quad \text{subject to} \quad \hat{y}(x) = y_t.$$

That is, the adversarial image should be as close to the source image as possible while being classified like the target image. Manuscript 4 operates in precisely this setting. The attacker iteratively refines inputs under a strict query budget, reflecting deployed APIs that intentionally minimise output information and often rate-limit access. The broader adversarial literature distinguishes additional axes, such as white-box [64, 65, 22, 66] versus black-box access [63, 24, 25, 67, 26] and untargeted [22, 64, 66, 63, 67, 24, 26] versus targeted objectives [65, 25, 67, 24, 26].

3.3.2 Exposed datasets *(Manuscript 5)*

A parallel to post-deployment model privacy arises when it is not models but *datasets* that are released, often to be used as training data for ML tasks. Yet, as discussed in the Introduction, case studies across decades have shown that many data releases have enabled re-identification of individuals with surprising ease. The risk is amplified by two features of the digital era. First, digital data tends to persist for long periods of time. Second, web-crawling tools have repeatedly ingested public data into downstream analyses and, more recently, into the training of large language models. A dataset that appears benign in its original context may therefore become linkable under future conditions that its publisher

did not anticipate.

This is especially relevant because identifiability rarely hinges on a single field. A single characteristic such as postal code, gender, or age may not uniquely identify an individual. However, combinations of seemingly ordinary attributes can reduce a population from millions to a small set of plausible matches [29]. Such attributes are commonly referred to as *quasi-identifiers*: features that may not be uniquely identifying on their own, but that become identifying when combined with other quasi-identifiers.

Several classical approaches have been proposed to address this risk in tabular data releases, including *k-anonymity* [68], *ℓ-diversity* [69], and *t-closeness* [70]. At a high level, these methods aim to prevent singling out by ensuring that individuals are hidden inside groups, and to prevent sensitive inference by ensuring that these groups contain sufficient diversity in sensitive attributes. This is akin to DP, except DP is operation-specific, while anonymisation techniques are primarily used for data releases, for any possible future data processing.

k-anonymity aims to ensure that each released record is indistinguishable, with respect to its quasi-identifiers, from at least $k-1$ other records. In other words, each equivalence class induced by the quasi-identifiers has size at least k, so that an adversary cannot isolate a single individual based on those fields alone. Such thresholds have been particularly attractive in settings where releasing aggregate or partially generalised data appears necessary for public benefit, for example during periods of heightened public health monitoring such as a pandemic [71]. However, k-anonymity comes with an important caveat. If all, or most, members of a k-anonymous group share the same sensitive attribute, then an adversary who knows that a person appears in the dataset may still infer their sensitive attribute with high confidence. This is often described as a *homogeneity* problem: the individual is hidden among k people, yet the sensitive information remains effectively revealed.

Two well-known extensions address this. ℓ-diversity requires that each equivalence class contain at least ℓ well-represented sensitive values, thereby reducing the risk that membership in a class reveals a single sensitive attribute [69]. Yet ℓ-diversity can still be vulnerable when sensitive values are distinct in name but similar in meaning. t-closeness strengthens this idea further by requiring the distribution of sensitive values within each class to remain close to the global distribution in the released table, as measured by a distance bounded by t [70]. Concretely, an equivalence class satisfies t-closeness if the distance between the two distributions is no more than t.

3.3.2.1 Topological Foundations for anonymisation *(Manuscript 5)*

The anonymisation approach developed in Manuscript 5 draws on tools from topological data analysis (TDA) [72, 73], which we briefly introduce here. The starting point is to treat a table of quasi-identifiers as a point cloud in $\mathbb{R}^M$, where each row becomes a point and M is the number of quasi-identifier attributes. Given such a point cloud, the idea is to connect nearby points and ask what shapes emerge. Formally, one constructs a *simplicial complex*

[73]: a combinatorial structure built from vertices (individual points), edges (pairs of nearby points), triangles (triples of mutually nearby points), and their higher-dimensional analogs. Which simplices are included depends on a scale parameter ϵ: a simplex on a set of points is included if every pair among them lies within distance 2ϵ. As ϵ increases from zero, isolated points first become connected by edges, then fill in to form triangles and larger structures, capturing the geometry of the data at progressively coarser scales. The resulting nested sequence of complexes, one for each value of ϵ, is called a *filtration*.

Persistent homology tracks how topological features such as connected components (at the H_0 level) and holes (at the H_1 level) appear and disappear as ϵ grows. Each feature is summarized by a *birth* value (the ϵ at which it appears) and a *death* value (the ϵ at which it merges into another component or is filled in). The collection of these birth-death pairs is called a *persistence barcode*. In the context of k-anonymity, Speranzon and Bopardikar [74] made the observation that the ϵ value at which all points belong to simplices of size at least k corresponds to a valid k-anonymous generalization, and that the persistence barcode encodes enough information to read off generalizations for multiple values of k from a single computation. Manuscript 5 extends this framework by introducing *hole-weighted persistence barcodes* that additionally track H_1 features, enabling efficient updates when the underlying dataset changes.

Enabling Privacy by Design

What we have discussed so far is not a catalogue of privacy methods and related techniques for their own sake, but a way to pin down the recurring ingredients that surface throughout this thesis. Across the manuscripts, the setting and challenges change, but the underlying question stays the same: what must be revealed for the intended purpose to be achieved, and what must remain hidden? With these preliminaries in place, the remainder of the thesis can be read as a sequence of concrete instantiations of PbD across the data processing lifecycle.

4 Objectives

The overarching goal of this thesis was to operationalise PbD for data processing activities. PbD, from a technical standpoint, looks very different depending on where in the data processing lifecycle one stands, what is being protected, and what assumptions about adversaries and infrastructure are reasonable. The five manuscripts that constitute this thesis reflect that diversity.

The first three works concern pre-deployment privacy, where data is distributed across institutions. The shared objective is to enable inference in this setting without sharing the data. Manuscript 1 does this for kernel-learning on high-dimensional medical imaging data, where kernel methods remain attractive since they are interpretable and impose lower computational overhead when combined with PETs. Manuscript 2 does this for multi-site GWAS on quantitative phenotypes, motivated both by the sensitivity of genomic data and the statistical advantages of association testing at scale. Manuscript 3 returns to kernel computation but in a quantum setting, asking whether widely used classical kernels can be realised through quantum encodings and extended to a distributed quantum architecture.

The final two manuscripts concern post-deployment privacy, where an artifact is released into the world for downstream use, and the question is no longer how to perform computations privately, but what a released artifact reveals or enables. The objective of Manuscript 4 is to determine what attack capability remains available through hard-label query interfaces for image classification models under strict query budgets. Because such models can retain information about sensitive training data in their decision boundaries, this is a PbD concern as much as a security one. The objective of Manuscript 5 is to maintain k-anonymity under repeated data releases in evolving datasets without extensive recomputations.

Together, these works do not pursue a single privacy mechanism. Rather, they show that operationalising PbD requires different technical strategies at different stages of the data processing lifecycle.

5 Results and Discussion

The technical core of this thesis is contained in five manuscripts, reproduced in full in Appendices (I–V). This chapter aims to summarize relevant results and present a comprehensive and integrated overview. It is organized according to the same two regimes discussed before: *pre-deployment* and *post-deployment*. A practical way to read the results across all five manuscripts is through three recurring questions that arise naturally from a privacy engineering perspective, stated here:

1. **What capability is enabled?** What becomes possible in practice that would otherwise require centralizing data, weakening the privacy stance, or accepting a brittle deployment mechanism?
2. **What is the operational cost?** What resources become the limiting factor in the relevant regime: compute, memory, communication, query budget, or the need for repeated re-computation as data evolves?
3. **Which assumption does the result rely on?** What threat model, access model, privacy guarantee or data condition is assumed?

5.1 Pre-deployment Privacy

Across the pre-deployment manuscripts, the recurring task is the same: sensitive data is distributed across sites that benefit from collaborating, but cannot pool raw records. In practice, this often includes settings where narrow models remain attractive – not because more general models are impossible in principle, but because validation, accountability, and resource constraints reward fairly interpretable methods whose failure modes and computational constraints are well understood [44]. Kernel methods fit this regime particularly well: they offer expressive non-linear decision boundaries while reducing learning to operations on inner products and Gram matrices, making them strong candidates for privacy-preserving collaboration.

Across all three manuscripts, we work in a semi-honest, distributed setting with a non-colluding server: participating institutions follow the protocol but may try to infer additional

information, and collusion between the server and the participating institutions is not permitted.

As noted earlier, default cryptographic approaches such as SMPC and FHE can be bottlenecked by communication and compute respectively, while DP can trade away utility in regimes where statistical power is already scarce. Manuscripts 1 and 2 therefore use RE as an alternative that preserves utility while reducing computational costs. Manuscript 3 keeps the kernel-computation target but moves to a quantum setting, where the available security primitives and the practical constraints differ.

5.1.1 OKRA: Scalable Kernel Learning on Distributed Data *(Manuscript 1)*

As discussed in the background chapter, RE is operation-specific: it introduces structured noise tailored to a particular computation, with the goal that the intended function can still be evaluated while sensitive inputs remain obscured. It has been explored in several works [53, 75, 76, 77, 55, 56], each targeting different settings and threat models.

Most relevant to OKRA are ESCAPED [55] and FLAKE [56], two RE approaches for privacy-preserving kernel learning that achieve exact model predictions. ESCAPED operates in a multi-party framework and requires communication among all participating institutions; this becomes operationally cumbersome as the number of participants grows and new entities are introduced. FLAKE, in contrast, adopts a distributed architecture in which each participant communicates only once with a semi-honest central server, reducing communication overhead, but encounters challenges when it comes to computational efficiency and scalability, particularly for high-dimensional data such as medical images. OKRA combines the practical deployment advantages of FLAKE with improved scalability.

We use orthonormal k-frames (mutually orthogonal unit vectors) to form sparse matrices and alleviate the burdens caused by dense matrices introduced in FLAKE. These sparse matrices help map the input data to a higher-dimensional space, enabling recombination with other data to compute the relevant kernels, including Linear, Gaussian (radial basis function (RBF)), Polynomial, and Rational Quadratic kernels. In this setting, the semi-honest server is limited to what can be inferred from the encoded data, learning only the singular values and vectors of the encoded matrices, which is insufficient to deduce the original data without additional information.

Empirically, our method achieves the same predictive performance as centralised learning. In particular, OKRA-based distributed support vector machine (SVM) matched (Table I.1) a naive centralised SVM on both evaluated clinical image datasets (MRI scans covering multiple stages of Alzheimer's disease [78] and augmented white blood cell images spanning four distinct cell types [79]). Likewise, OKRA-based principal component analysis (PCA) and naive Kernel-PCA produced identical results (Table I.2). Beyond correctness, our results emphasize efficiency and scalability relative to ESCAPED and FLAKE. In end-to-end runtime experiments including data encoding, communication, and kernel computation, OKRA scaled more favourably than both baselines: with 10 participating institutions, each

contributing 400 samples with 1000 features, OKRA completed in about 16 seconds on average, compared with roughly 24 seconds for ESCAPED and about 69 seconds for FLAKE (Figure I.2). A similar pattern appeared when scaling image size, where OKRA remained practical while the baselines scaled much less favourably (Figure I.3).

5.1.2 PP-GWAS: Association Testing on Distributed Genomic Data *(Manuscript 2)*

Multi-site GWAS are known to improve statistical power, but genetic data is sensitive, resulting in increased regulatory constraints worldwide. In Manuscript 2, our goal was to apply methods from RE to perform GWAS in a distributed manner. We adopted methods from Manuscript 1, where we introduced sparse matrices that helped with computation on high-dimensional data. This was particularly relevant because the core difficulty in GWAS is that genomic datasets are vast, often with millions of SNPs to analyze, which motivates even methods optimized for the centralised setting. Further, applying privacy-enhancing technologies to genomic data can be expensive [80, 81, 82].

Our approach builds on REGENIE [59], a centralised algorithm for computing associations between SNPs and phenotypes. REGENIE uses ridge regression in two stages: first, it produces intermediate blockwise predictions, which are then stacked to enable a second ridge regression on top. This reduces the large-scale problem to a medium-scale problem. The resulting predictions are further strengthened using k-fold cross-validation. Amongst privacy-preserving methods, SF-GWAS stands out as the state of the art and also adopts REGENIE, applying SMPC and FHE on top of it. We took a similar route, adding REs into the computations of the two regression problems by applying a modified version of the alternating direction method of multipliers (ADMM).

We performed experiments on both real-world and synthetic datasets, comparing performance against both meta-analysis and SF-GWAS across various platforms and computing environments. On two real-world datasets, namely a Bladder Cancer Risk dataset and an Age-Related Macular Degeneration (AMD) dataset, PP-GWAS achieved agreement with the centralised baseline REGENIE with R^2 values essentially equal to 1 when comparing association statistics (Figure II.2). In contrast to meta-analysis, whose accuracy deteriorated as data became more fragmented across participating sites, PP-GWAS remained robust to partitioning (Figures II.5 and II.6). Further, when scaling participating institutions, covariates, samples, and SNPs, PP-GWAS consistently required less than half the runtime of SF-GWAS while maintaining near-centralised accuracy; for example, on a dataset with 9178 samples and 612,794 SNPs, PP-GWAS required about 1.04 h with two nodes compared with 2.4 h for SF-GWAS (Figure II.3(A–E)). Our compute usage was also significantly lower, while communication overhead was comparatively higher; notably, this cost shifts in a predictable manner, with communication growing linearly while SF-GWAS exhibits exponential growth (Figure II.4(A,B)). Despite this, an argument can be made that internet resources are more widely accessible than compute resources. Finally, our results should be interpreted in light of the underlying assumptions and practical constraints (which hold true in related works as well [80, 81, 82]): the protocol relies on a semi-honest, non-colluding server setting and presumes sufficient harmonization of data processing and study design across sites, which

remain limiting factors in real multi-institution deployments.

5.1.3 Computing Kernels for Quantum Machine Learning *(Manuscript 3)*

QML is an emerging discipline, supported by rapid progress in quantum computing hardware. As discussed earlier, a landmark study [61] established a close link between kernel functions and quantum encodings. Just as feature maps map input data to a higher-dimensional Hilbert space, encoding classical data into quantum data maps data to an infinite-dimensional Hilbert space. Building on this observation, subsequent work [83] focused on computation of a linear kernel, but remained limited in scope. In Manuscript 3, we construct quantum encodings for widely used kernels, specifically, the polynomial, Gaussian (RBF), and Laplacian kernels, that act as *quantum feature maps* and enable kernel computation. Moreover, we showed how quantum teleportation can be used to support this computation in a distributed setting. For the Gaussian and Laplacian kernels, the construction builds on Random Fourier Features [84] to obtain kernel estimates via inner products of quantum states, while for the polynomial kernel we apply the multinomial theorem directly [45, 85]. We theoretically showed that the approximation quality of the kernels depends on both the number of random features (for the Gaussian and Laplacian kernels) and the number of repeated circuit executions (called *shots*).

Empirically, we validated the distributed architecture on publicly available datasets using Qiskit's Aer Simulator. Due to simulator constraints (only 31 qubits), the evaluation focuses on linear-kernel computation in a two-party configuration. Using stratified 5-fold cross-validation and 1024 shots, the distributed quantum pipeline achieves accuracies comparable to centralised quantum computation, while remaining below a centralised classical kernel baseline (Table III.1). For example, in kernel-SVM classification on the Parkinson's dataset, the accuracies were 0.7983 ± 0.0798 (distributed quantum) vs. 0.7875 (centralised quantum) and 0.8196 ± 0.0644 (centralised classical); and on the Framingham Heart Study, the accuracies were 0.6340 ± 0.0143 vs. 0.6308 and 0.6788 ± 0.0108 (Table III.1). For kernel-PCA on the evaluated binary datasets, the results are similar to the ones above (Table III.1).

Beyond baseline accuracy, we evaluated two factors that shape practical performance in quantum settings. First, increasing simulator noise degrades accuracy as expected: under depolarizing error models (0.1% and 1%), performance drops relative to the noiseless distributed-quantum baseline (Figure III.3). Second, increasing the number of shots improves accuracy as expected on the evaluated subset (Figure III.4). These results should be interpreted in light of the study's assumptions and current hardware constraints: the security discussion is anchored in a semi-honest model (with additional discussion of security against third-party eavesdropping and collusion scenarios), and the empirical validation is restricted to simulator-based experiments and linear kernels due to qubit limitations. Nevertheless, the theoretical contribution remains that widely used kernels in classical ML can be approximated in the quantum setting using the proposed encodings, in centralised and distributed environments.

Discussion

What capability is enabled? Across the pre-deployment manuscripts, the shared capability is to enable multi-site inference without pooling raw records, by restructuring the computation so that only restricted artifacts move across organizational boundaries. Manuscript 1 does this for kernel learning, enabling distributed kernel-SVM and kernel-PCA style workflows on high-dimensional imaging data while preserving correctness of the target kernel computations and matching centralised performance. Manuscript 2 shares the same collaboration goal, but for GWAS, allowing sites to jointly compute association statistics that remain close to a centralised GWAS and improve substantially over meta-analysis when data is partitioned across institutions. Manuscript 3 returns to the kernel target, but in a quantum setting: it introduces quantum feature maps for commonly used kernels and a distributed architecture that supports kernel estimation using quantum primitives, where accuracy is governed by an explicit approximation trade-off, noise and circuit repetitions.

What is the operational cost? The three manuscripts surface different limiting resources because the target computations differ. In Manuscript 1, the dominant operational cost lies in the encoding and kernel-computation overhead induced by high-dimensional inputs, and the privacy engineering objective is to reduce the computational and memory burden of dense re through sparse constructions. In Manuscript 2, the dominant trade-off is between local resource constraints and cross-site communication: the protocol aims to reduce local compute and memory demands relative to heavy cryptographic baselines while accepting increased communication overhead. In Manuscript 3, costs are dominated by state preparation, circuit depth, and shot complexity. These costs are currently decisive in practice, and they confine empirical validation to simulator-based experiments and small-scale regimes, although the constructions do provide a theoretical solution to approximate kernel computation.

Which assumptions does the result rely on? All three works assume semi-honest adversaries and non-collusion assumptions about the server that match common distributed inference threat models, and they require sufficient cross-site harmonization of preprocessing and study design, typical of cryptographic approaches. In the RE based protocols, security is tied to what the encoded artifacts reveal about underlying inputs, and the arguments are therefore specific to the computation being enabled. In the quantum setting, the security discussion additionally considers risks on quantum channels such as eavesdroppers, while the empirical scope is constrained by current hardware and simulation limits.

The manuscripts collectively illustrate a central PbD lesson: privacy is not a single slider. It is a negotiated boundary between what must be revealed for the computation to proceed, what must remain hidden.

5.2 Post-deployment Privacy

In the post-deployment part of the thesis, we discuss the released artifact that others can interact with over time. We distinguish two such objects: (i) a deployed model, typically

exposed as an API or service that users can query, and (ii) a released dataset, typically published for scientific reuse, archival purposes, or downstream analysis. These two deployment modes create different privacy risk surfaces: models concentrate risk around what can be inferred or manipulated through access, whereas datasets concentrate risk around what can be inferred from the released records, especially when combined with external information.

5.2.1 TEA: Crafting Adversarial Images Using Edge Information *(Manuscript 4)*

In the realm of targeted hard-label black-box attacks, several methods already exist in the literature. In this setting, the attacker's goal is to craft an adversarial image that is close to a target image while using as few model queries as possible. When evaluating existing attacks, we repeatedly observed that outlines of visually discernible objects from the target image often remained visible in the adversarial outputs, even after thousands of queries, across many source–target pairs and model architectures. This motivated the study design in Manuscript 4.

Our proposed method, *Targeted Edge-Informed Attack* (TEA), builds on this observation by preserving edge information from the source image while applying perturbations. We observed that TEA can drive substantial distance reduction until it reaches a narrow decision region; beyond this, the edges in the adversarial images become noisy, uninformative, and increasingly sensitive to small perturbations. As a result, TEA is most effective in the low-query regime, which is often the practically relevant setting. Further, by applying a state-of-the-art method after TEA, we postulated that TEA can serve as an initialization strategy, yielding a practical hybrid for when additional queries are available.

We evaluated TEA in a low-query regime on randomly sampled source–target image pairs from different datasets across diverse architectures, and compared it against other targeted hard-label attacks (HSJA [67], AHA [25], CGBA [24], and CGBA-H [24]). Across different settings and query budgets, TEA consistently achieves lower median distances between the constructed adversarial image and the target image. For example, in one setting, at 400 queries, the median distance on ResNet-50 is 51.9530 for TEA compared to 67.091 (AHA) and 71.479 (CGBA-H), and on vision transformer (ViT) it is 47.8666 for TEA compared to 59.857 (AHA) and 61.300 (CGBA-H) (Table IV.1). Additional analyses (edge ablations (Table IV.3), robustness across datasets and models (Figures IV.7, IV.12, and IV.13), and applicability to modern zero-shot vision–language classification (Figure IV.11)) further support and validate our strategy of prioritising non-edge perturbations.

Our results should be interpreted in light of the setting's constraints. TEA is designed specifically for targeted hard-label low-query black-box access, and its gains are concentrated in rapidly improving distance under tight query budgets (usually under 500 for high-res images). In regimes where many queries are available and TEA is used purely as an initialization, at very high query counts (~ 20000), final performance becomes comparable to that of state-of-the-art attacks run without TEA (Table IV.8 and Figure IV.8). Additionally, the perturbations in TEA can introduce localized artifacts that may be visible to human

observers in human-in-the-loop settings that might flag an attacker (Figure IV.2).

From a PbD perspective, these results illustrate a concrete interface-level lesson: a hard-label API, often considered restrictive precisely because it withholds probabilities and internal model information, still permits an attacker to make rapid targeted progress under adaptive querying with the help of semantic structures in the target image. The minimal output does not translate into minimal risk. This challenges the assumption that restricting an interface to top-1 labels is, by itself, a sufficient post-deployment safeguard, and it suggests that interface design must be complemented by other mechanisms such as query monitoring.

5.2.2 k-anonymising Data Dynamically *(Manuscript 5)*

There are many standardized ways to achieve k-anonymity [68], and to further strengthen it with methods such as ℓ-diversity [69] and t-closeness [70] when releasing data in a privacy-focused manner. A particularly distinctive approach [74] which draws on tools from TDA to derive k-anonymous generalizations, but is limited to *static* datasets. We extend this line of work by proposing a systematic method for achieving k-anonymity when the underlying data is *dynamic*, which is often the practical case in domains such as electronic health records (EHRs): new patients are added over time, mistakes and misdiagnoses are corrected, and records may be periodically scrubbed or removed.

In [74], the central idea is to build a simplicial complex on the available data and compute persistence information over a filtration. The method uses a modified form of persistence barcodes called *weighted persistence barcodes* which track homological structure across the filtration, with the H_0 bars weighted by the number of data points contained in each connected component. A key advantage of this construction is that a *single* simplicial complex and barcode computation can support multiple anonymisations for a fixed k, and can also support multiple values of k without re-running the full pipeline, thereby enabling flexible downstream choices (including stronger criteria such as l-diversity or t-closeness from one computation).

We build on these advantages by introducing *hole-weighted persistence barcodes*, which not only track the formation and merging of components at the H_0 level, but also track the formation of holes at the H_1 level (Figure V.3). This allows us to identify filtration regions in which re-computation may be necessary under data changes, by distinguishing holes that persist from those that disappear. This allows us to propose update procedures to handle data additions, removals, and alterations, together with theoretical evidence showing that, over successive changes to the dataset, applying our method is computationally less expensive than repeatedly applying the original static approach from scratch (Table V.3 and Figure V.5).

Discussion

What capability is enabled? The post-deployment manuscripts emphasize that releasing an artifact creates a long-lived interaction surface, and that this surface enables both legitimate use and adversarial leverage. In Manuscript 4, the capability is adversarial: a hard-label prediction API enables targeted, query-driven attacks, and we show that low-query regimes can be exploited more effectively by incorporating image structure such as edge information rather than relying only on decision-boundary geometry from the outset. In Manuscript 5, the capability is database maintainer-facing: it becomes possible to update a k-anonymised release when the underlying dataset changes, without recomputing persistent homology from scratch after each addition or deletion. This supports repeated releases under evolving data, while preserving the core advantage of topology-informed anonymisation, namely that a single persistence computation can support multiple generalizations and multiple values of k.

What is the operational cost? In Manuscript 4, the primary scarce resource is query budget. TEA trades lightweight local computation (edge extraction and local patch-based updates) for a steeper early reduction in distance under strict query limits. In Manuscript 5, the scarce resource is recomputation under change. The proposed hole-weighted persistence representation and filtration trimming aim to localize updates so that additions, deletions, and minor modifications do not force a full recomputation across the entire filtration, thereby shifting the cost from repeated global recomputation to targeted updates and essentially bookkeeping over time.

Which assumptions does the result rely on? Manuscript 4 assumes targeted, hard-label, black-box access in which the attacker can issue repeated adaptive queries, and it does not model strong active and adaptive defences. Manuscript 5 relies on assumptions about the data, including the treatment of attributes (e.g., numerical versus categorical), and the structure of sensitive information (e.g., a single sensitive attribute versus multiple).

All in all, these manuscripts reinforce a PbD lesson: once an artifact is released, privacy becomes a moving boundary shaped by interface design, monitoring policy, and adaptiveness to changes rather than a static property fixed at publication.

Integrated Discussion

Read together, the five manuscripts support a single thesis-level claim: from a technical point of view, PbD for data processing activities is not a single technique, but an end-to-end engineering discipline whose practical answer depends on when and what privacy constraints are enforced, what is released, and which resources and trust assumptions are relevant.

Across the pre-deployment manuscripts, privacy is used to make collaboration possible without data pooling. Manuscript 1 and Manuscript 2 illustrate this in classical distributed pipelines by shifting cost away from heavy cryptographic primitives. Manuscript 3 highlights

the same design pattern under different primitives.

Across the post-deployment manuscripts, the central observation is that release creates an interaction surface over time. Manuscript 4 makes this concrete by showing how an API that exposes only top-1 labels can still support effective targeted attacks under tight query budgets. Manuscript 5 proposes a method to anonymise data continuously as they are updated and released.

Revisited through the three guiding questions, we see that PbD is not a single layer added to an otherwise fixed pipeline, but a repeated negotiation of what must be revealed and what must remain hidden, shaped by capability, cost, and assumptions, as the protected artifact moves from training and collaboration to deployment and long-term reuse.

6 Conclusion

Data is at the crux of modern science, medicine, agriculture, entertainment, and daily life. The systems we build to collect, process, and learn from it enable inference at scales no individual or collective could achieve alone. However, these systems do not exist in a vacuum. They are trained on records that belong to people, deployed through interfaces that we interact with, and maintained over time periods that outlast any single design decision. When those records are sensitive, whether medical images, genomic variants, or patient histories, the stakes of getting privacy wrong are not abstract. They are clinical, legal, personal, and political.

In my view, the responsible position for those who build data processing pipelines is neither to wait for regulation to catch up with technological progress, nor to build systems that merely comply with the minimum required. It is to treat privacy as a design commitment that is present throughout the lifecycle of any data processing activity.

My research has spanned various scopes and perspectives, but it is unified by a common principle. It substantiates the view that privacy is not a single sticker one can place on a product and claim guarantees, but a composition of techniques and decisions taken at different stages, under different assumptions, for different artifacts. On the pre-deployment side, we showed that randomized encoding can serve as a practical middle ground between the computational demands of conventional cryptographic approaches and the irreversible accuracy cost of noise-based methods, enabling multi-site learning on distributed medical and genomic data while preserving accuracy and privacy. We further showed that widely used classical kernels can be realised through quantum feature maps and computed in a secure distributed quantum setting. On the post-deployment side, we showed that semantic structures in images, specifically edge information, can be exploited to accelerate adversarial attacks under strict query budgets, revealing that the content of a deployed model's inputs can be turned against it in ways that purely geometry-based attacks overlook. We also showed that anonymisation for evolving datasets can be maintained incrementally rather than rebuilt from scratch after every change. What I find motivating is that despite PbD being multi-faceted and layered, it need not be overwhelming at any single point. Each aspect can be tackled in a concrete technical manner when examined closely enough. We should not be discouraged by the absence of a single technique that solves every privacy problem, but instead recognize that different stages of the data processing lifecycle require

different safeguards and guarantees.

We build systems that infer and process data at scale. The least we owe is to be conscious about what they reveal.

7 Appendix

Publications contributing to this doctoral thesis, as listed in Chapter 1, are included in the following appendix. The corresponding citations can be found in Chapter 1. The included manuscripts are presented in custom formatting (i.e., not in the publisher's typeset PDF format). Only minor template and formatting changes were made to ensure consistent integration into this thesis; the scientific content remains unchanged.

License Information

1. **Private, Efficient and Scalable Kernel Learning for Medical Image Analysis** (Springer LNCS chapter, DOI: 10.1007/978-3-031-89704-7_7). Included in this thesis under Springer Nature author thesis reuse rights. *Reproduced with permission from Springer Nature.* Publication record: https://link.springer.com/chapter/10.1007/978-3-031-89704-7_7.

2. **PP-GWAS: Privacy Preserving Multi-Site Genome-wide Association Studies** (Nature Communications, DOI: 10.1038/s41467-025-66771-z). Published Open Access under the Creative Commons Attribution 4.0 International (CC BY 4.0) license. License: https://creativecommons.org/licenses/by/4.0/. Publication record: https://www.nature.com/articles/s41467-025-66771-z.

3. **Distributed and Secure Kernel-Based Quantum Machine Learning** (TMLR via OpenReview). Published under the Creative Commons Attribution 4.0 International (CC BY 4.0) license. License: https://creativecommons.org/licenses/by/4.0/. Publication record: https://openreview.net/forum?id=3jdI0aEW3k.

4. **Accelerating Targeted Hard-Label Adversarial Attacks in Low-Query Black-Box Settings** (arXiv:2505.16313; accepted at IEEE SaTML 2026). Made available under the Creative Commons Attribution 4.0 International (CC BY 4.0) license. License: https://creativecommons.org/licenses/by/4.0/. arXiv record: https://arxiv.org/abs/2505.16313.

5. **Dynamic k-Anonymity for Electronic Health Records: A Topological Framework** (Springer LNCS chapter, DOI: 10.1007/978-3-031-82349-7_10). Included in this thesis under Springer Nature author thesis reuse rights. *Reproduced with permission from*

Springer Nature. Publication record: https://link.springer.com/chapter/10.1007/978-3-031-82349-7_10.

I Private, Efficient and Scalable Kernel Learning for Medical Image Analysis

Anika Hannemann* **Arjhun Swaminathan*** **Ali Burak Ünal** **Mete Akgün**

Abstract

Medical imaging is key in modern medicine. From magnetic resonance imaging (MRI) to microscopic imaging for blood cell detection, diagnostic medical imaging reveals vital insights into patient health. To predict diseases or provide individualized therapies, machine learning techniques like kernel methods have been widely used. Nevertheless, there are multiple challenges for implementing kernel methods. Medical image data often originates from various hospitals and cannot be combined due to privacy concerns, and the high dimensionality of image data presents another significant obstacle. While randomised encoding offers a promising direction, existing methods often struggle with a trade-off between accuracy and efficiency.

Addressing the need for efficient privacy-preserving methods on distributed image data, we introduce OKRA (Orthonormal K-fRAmes), a novel randomized encoding-based approach for kernel-based machine learning. This technique, tailored for widely used kernel functions, significantly enhances scalability and speed compared to current state-of-the-art solutions. Through experiments conducted on various clinical image datasets, we evaluated model quality, computational performance, and resource overhead. Additionally, our method outperforms comparable approaches.

I.1 Introduction

Medical imaging plays a pivotal role in modern medicine, providing crucial insights into the anatomical and physiological aspects of the human body. Diagnostic medical imaging has revolutionized healthcare by aiding in early disease detection, treatment planning, and monitoring patient progress. In the field of medical image analysis, techniques based on kernel-based machine learning, sometimes referred to as kernel learning, have garnered widespread acceptance and application [86, 87, 88, 89]. These techniques map the original data to a higher dimensional space, allowing identification of complex patterns and relationships that are not apparent in the original space. These techniques include kernel-based Support Vector Machines (SVM), Principal Component Analysis (PCA), Canonical Correlation Analysis (CCA), Gaussian processes, k-means, and so on. Kernel learning offers robust solutions due to their resistance to outliers, and offer greater interpretability as compared to deep learning techniques, which is especially important in the healthcare setting. Moreover, they offer superior performance in handling of small datasets [90], as can be the case with the analysis of certain diseases.

However, in many cases, the medical data available within a data source is insufficient to arrive at reliable predictions. Medical data is often distributed across hospitals, clinics and research institutions. This however complicates the implementation of kernel learning techniques, since privacy risks arise from manipulation and mishandling of sensitive patient

*These authors contributed equally to this work.

data. For instance, health data is shown to make it possible to re-identify individuals [91, 92]. Further institutional policies and regulations, such as the General Data Protection Regulation (GDPR) prohibit sharing of sensitive healthcare data [93].

To tackle this, privacy preserving techniques could be employed. However, traditional security techniques such as Homomorphic Encryption (HE) [39], and Secure Multiparty Computation (SMPC) [94] introduce operational challenges. They necessitate significant computational resources or continuous communication, which is challenging in real world settings. Differential Privacy (DP) on the other hand introduces noise, which results in model inaccuracies [95, 96, 97]. Moreover, many healthcare institutions rely on cloud services for computational tasks [98, 99]. With this in mind, our work seeks to perform kernel-based tasks on distributed medical data within a federated architecture using a central server. This design choice is in tune with real-world scenarios where healthcare institutions seek to perform joint-analysis for higher accuracy while keeping their data private, and might additionally lack on-premise computational resources. In particular, one-shot federated learning is a promising approach where in just a single round of communication, a global model is computed by the central server [100, 101]. While the central server is trusted to perform the computations correctly, there's an underlying need to ensure the platform cannot derive sensitive information from the data.

In our approach, we implement kernel learning using a one-shot federated approach. We utilise randomised encoding instead of traditional security techniques due to its lower computational and communication costs, while still maintaining model accuracy. Thereby we introduce OKRA (Orthonormal K-fRAmes), which projects the input data into higher dimensional spaces, and uses a central server to compute kernel functions, facilitating the training of kernel learning models as though the data was centralized. Unlike the comparable state-of-the-art solutions, OKRA requires less computation time, especially for high-dimensional data.

We make three contributions:

1. We present the OKRA methodology to privately compute kernel functions on distributed data.
2. We report a privacy analysis confirming the ability of OKRA to maintain the privacy of both input data and the size of medical images.
3. We evaluate OKRA through extensive experimentation involving various combinations of datasets, kernels and Machine Learning techniques.

Paper structure: Section I.2 introduces related work. Section I.3 describes our methodology. Section I.4 presents our experimental results on both performance and efficiency, followed by conclusion in Section I.5.

I.2 Related Work

I.2.1 Kernel Learning

Kernel learning comprises a collection of techniques that utilise kernel functions to perform pattern analysis in various contexts, including the identification of clusters, principal components and correlations [102]. These functions are maps that measures the similarity between data points by projecting them into a higher dimensional space. This allows for a more nuanced analysis of data patterns that might not be discernible in the original space. A kernel function is hence defined as follows:

Definition 3. *Kernel function [103]*
A *kernel function* K is a map $K : \mathscr{D} \times \mathscr{D} \to \mathbb{R}$ that satisfies $K(x,y) = \langle \phi(x), \phi(y) \rangle$, where $\phi : \mathscr{D} \to \mathscr{H}$ is a map from the input space $\mathscr{D}$ to a Hilbert space $(\mathscr{H}, \langle \cdot, \cdot \rangle)$. ϕ is called the feature map and satisfies the following commutative diagram.

$$\begin{array}{ccc} \mathcal{D} \times \mathcal{D} & \xrightarrow{K} & \mathbb{R} \\ {\scriptstyle \phi \times \phi} \downarrow & \nearrow_{\langle \cdot, \cdot \rangle} & \\ \mathcal{H} \times \mathcal{H} & & \end{array}$$

The wide range of kernel functions, including Linear, Gaussian, Polynomial and Rational Quadratic Kernels allow for a wide range of applications tailored to specific data characteristics. For instance, linear and Gaussian-kernel based regression methods have been effectively employed genomic prediction [104] while Rational Quadratic (RQ) kernels are particularly instrumental in point set registration tasks in medical image analysis, optimizing correspondence and alignment in complex imaging data [105]. Similarly, Kernel Principal Component Analysis (KPCA) has emerged as an essential technique in the realm of microarray data analysis for cancer diagnosis [106]. Meanwhile, kernel-based SVM demonstrates continued relevance and adaptability with classification tasks [107].

I.2.2 One-shot Federated Learning

Federated Learning (FL), as proposed by [108], presents an effective solution for extracting insights from sensitive data present on distributed nodes while preserving privacy. In a conventional machine learning approach, a model M is trained on centralized data D. However, privacy constraints often prohibit data from being extracted from the nodes. In the FL framework, every participant P_i trains a local model M_i using its local dataset D_i and subsubsequently transmits the model parameters to a central server. This central server aggregates the model parameters to establish a global model M and shares back global parameters with the participants, which the participants utilize to build future local model parameters. This is done iteratively until a suitable global model is established. This iterative nature of FL however often leads to overhead and latency. Further, continuous communication could introduce an avenue of security threats such as data poisoning, model poisoning, backdoor attacks, and membership inference attacks [109].

In contrast, one-shot federated learning approaches such as [110], minimize this communication overhead by restricting communication to a single round. Each participant sends its relevant parameters to the central server once, which subsequently undertakes the role of model training. For our methodology, the central server computes relevant kernels with a single round of communication.

I.2.3 Privacy Preserving Techniques

FL incorporates various techniques to safeguard the privacy of input data:

Cryptography-based Approaches:

These primarily constitute the usage of HE and SMPC. HE aims to protect privacy of aggregated models by encrypting the parameters of the local models. However, this is computationally intensive, often leading to slower processing speeds, which can be a bottleneck in real-time applications [111, 112, 113]. Meanwhile, SMPC splits the data into secret shares distributed among participants for shared computation. Despite being less computationally intensive compared to HE, it requires continuous communication, often resulting in high execution times [114, 115, 116].

Differential Privacy:

FL methods using DP introduce noise to individual models to enhance privacy, making it challenging to deduce the original model or identify a specific data point's membership. However, the noise added is irremovable and often impacts model accuracy [117, 118, 119].

Randomized Encoding-based Approaches:

Contrary to more computationally demanding approaches discussed, these methods present an efficient way of preserving data privacy, such as maintaining relationships between dot products [120, 54]. Randomized encoding approaches extensively investigated in [53, 75, 76, 77] use a variety of techniques, including but not limited to geometric perturbations. Geometric perturbations introduce structured noise to the data. This noise can be factored out, thus differing from differential privacy methods in the context of preserving model accuracy. For instance, [121] expanded the concept of perturbation to clustering tasks by using a randomized kernel matrix to obscure details about dot product and distance data. It should be noted that the security of randomized encoding-based approaches is dependent on the specific problem at hand since they are tailored to meet the unique demands and constraints of a privacy-preserving task.

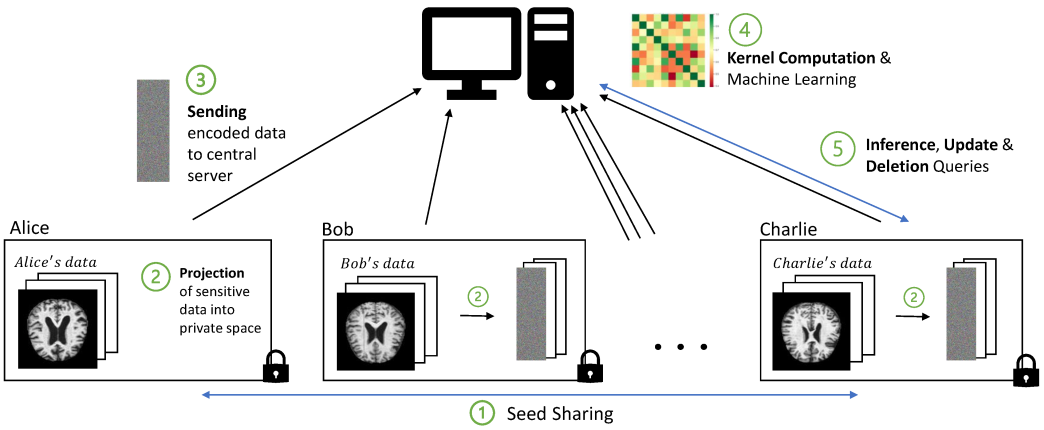


Figure I.1: Overview of OKRA.

I.2.4 Privacy-preserving Kernel Learning using Randomized Encoding

The integration of randomized encoding methods into kernel learning presents a promising avenue, particularly in their cost-effectiveness compared to traditional approaches. This subsection delves into two noteworthy contributions in this domain that achieve exact model predictions: ESCAPED [55] and FLAKE [56], each offering distinct methodologies and implications.

ESCAPED makes use of randomized encoding within a multi-party computational framework for kernel computations. Its core innovation lies in enabling secure, collaborative computing without compromising individual data privacy. However, this approach necessitates intricate communication channels among all participating data providers. This requirement becomes particularly cumbersome with the introduction of new data entities, posing significant logistical challenges in scalable deployments.

On the other hand, FLAKE works towards optimizing communication overhead in kernel-based learning. It employs a one-shot federated learning architecture, requiring data providers to only communicate once with a central server. However, this methodology encounters its own set of challenges, particularly with computational efficiency and scalability. It especially demonstrates limitations in processing high-dimensional data, a common characteristic in complex datasets such as medical images.

In this work, we present an alternative to current randomized encoding methods for kernel computation in machine learning. We propose a one-shot algorithm, hence bypassing ESCAPED's communication overheads, while scaling better than FLAKE to accommodate higher dimensional datasets.

I.3 Methods

In this section, we delve into the methodology of our approach, focusing on private kernel computation in a distributed environment. First, we detail the adversarial architecture we operate in. In our architecture, for simplicity, we consider participants Alice, Bob, and

Charlie, along with a central server orchestrating the training of a machine learning model, as described in Figure I.1. However, this can be extended to any number of participants. We operate under the assumption of semi-honest participants and a non-colluding central server. A semi-honest party follows the protocol correctly but may attempt to infer additional information. This is in line with existing state-of-the-art [55, 56].

Our method must satisfy four key requirements, with **privacy** being the first: Neither the central server, nor the corrupted participants can learn a non-corrupt participant's data, and the image sizes remain concealed from the server. Second, **correctness** is crucial, with the model's accuracy needing to align that of a centrally trained model. Third, **efficiency** is important, as communication costs and execution times should be practical even when input data scales up. Finally, **data adaptability** is essential, allowing the model to accommodate new data and participants with minimal overhead.

We will focus on four of the most widely used kernel functions: Linear, Gaussian, Polynomial, and Rational Quadratic kernels. We begin by briefly defining them.

I.3.1 Preliminaries

Kernel functions project the data into a Hilbert space using a feature map ϕ, and compute the similarity between the data points in this space as described in Definition 3. However, the trick here lies in the ability to compute this similarity without explicitly mapping the data. This is a consequence of Mercer's theorem, which provides necessary and sufficient conditions for a function from an input space to be a 'kernel', ensuring the mapped data resides in a Hilbert space. We define four such widely used kernels below, which are of interest to us.

Definition 4. *Linear Kernel [122]*
The *linear kernel* is a map $K_{lin} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ defined by

$$K_{lin}(x, y) = xy^T.$$

Here the feature map is the identity map, and the feature space and input space are identical.

Definition 5. *Gaussian Kernel [85]*
The *Gaussian kernel*, often termed the Radial Basis Function (RBF) kernel, is defined as $K_{RBf} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ where

$$K_{RBF}(x, y) = \gamma^2 \exp\left(-\frac{\|x - y\|^2}{2l^2}\right).$$

Here γ is called the amplitude, and l, the length-scale. These measure the spread of the kernel.

Definition 6. *Polynomial Kernel [47]*
The *Polynomial kernel* is the map $K_{poly} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ where

$$K_{poly}(x, y) = (1 + xy^T)^d.$$

Here d is called the degree of the defined kernel.

Definition 7. *Rational Quadratic Kernel [123]*
The *Rational Quadratic (RQ)* kernel is the map $K_{RQ} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ where

$$K_{RQ}(x, y) = \gamma^2 \left(1 + \frac{\|x - y\|^2}{2\alpha l^2}\right)^{-\alpha}.$$

Here γ is called the amplitude, and l, the length-scale. The RQ kernel is an infinite sum of Gaussian kernels, with α, the scale mixture, determining the weight of each. The RQ kernel converges to the Gaussian kernel when $\alpha \to \infty$.

Note that we have defined the kernels above without referencing the feature maps explicitly. Having described the kernel functions of interest, we delve into the specifics of our proposed methodology. Central to this is the utility of orthonormal k-frames, defined as follows.

Definition 8. *Orthonormal k-frame [124]*
A *k-frame* is an ordered set of k linearly independent vectors in a vector space of dimension n, with $k \leq n$. An ordered set of k linearly independent orthonormal vectors is called an *Orthonormal k-frame*.

I.3.2 The OKRA Methodology

Data Preprocessing

Prior to delving into our methodology, we detail the preprocessing steps applied to medical imaging datasets. Assuming each dataset comprises n medical images with varying resolution and color depth, we format each image into a four-dimensional array of dimensions $n \times h \times w \times c$, where h and w are height and width in pixels, and c represents the number of color channels used to represent the pixel. The preprocessing involves flattening each image into a 1D array, reshaping the dataset to $n \times (hwc)$ dimensions. This step is crucial for ensuring compatibility with kernel-based methods, particularly important for the heterogeneous nature of medical imaging data.

Participants Alice, Bob, and Charlie possess preprocessed data matrices A, B, and C, respectively. These have sizes $n_{\mathscr{A}} \times f$, $n_{\mathscr{B}} \times f$, and $n_{\mathscr{C}} \times f$ respectively. While $n_{\mathscr{A}}$, $n_{\mathscr{B}}$, and $n_{\mathscr{C}}$ may differ due to variable data availability among participants, we assume all participants contribute more than one image. This condition is essential for meaningful federated learning and analysis.

Seed Generation

A robust Public Key Infrastructure (PKI) ensures every participant possesses the public signing keys of all others. At the onset of the protocol, a participant is arbitrarily designated as the leader, responsible for the generation of a random seed. A secure seed is shared with participants using public-key encryption and verified with digital signatures. We assume a

trusted third party for fair public key distribution, in line with standard privacy-preserving federated learning practices [36, 125].

Randomized Encoding

The participants Alice ($\mathscr{A}$), Bob ($\mathscr{B}$) and Charlie ($\mathscr{C}$) with their shared seeds generate their own sets of j orthonormal k-frames $({}_{\mathscr{A}}O_1,\ldots,{}_{\mathscr{A}}O_j)$, $({}_{\mathscr{B}}O_1,\ldots,{}_{\mathscr{B}}O_j)$, and $({}_{\mathscr{C}}O_1,\ldots,{}_{\mathscr{C}}O_j)$ respectively, using the seed such that

$$ {}_{\mathscr{P}}O_i = [{}_{\mathscr{P}}v_i^{1\,T}\ldots{}_{\mathscr{P}}v_i^{k_i\,T}], $$

for $\mathscr{P}\in\{\mathscr{A},\mathscr{B},\mathscr{C}\}$. Further, for any $p\le j$ and $\mathscr{P},\mathscr{Q}\in\{\mathscr{A},\mathscr{B},\mathscr{C}\}$, it holds that

$$ \left({}_{\mathscr{P}}O_p^{\dagger}\right)\left({}_{\mathscr{Q}}O_p\right) = I, $$

where $\dagger$ denotes the conjugate transpose. Each ${}_{\mathscr{P}}v_i^l$ is an element of a corresponding vector space V_i defined over $\mathbb{C}$ such that $k_i \le dim(V_i)$. Further, $V_1\oplus\ldots\oplus V_j = V$, $dim(V) = f+k$, and $k_1+\ldots+k_j = f$. Then we define ${}_{\mathscr{P}}\Gamma$ to be the direct sum

$$ {}_{\mathscr{P}}\Gamma = {}_{\mathscr{P}}O_1\oplus\cdots\oplus{}_{\mathscr{P}}O_j. $$

To enhance privacy, a random permutation matrix $\sigma\in S_f$ is generated using the shared seed by all three participants. Here S_f denotes the symmetric group of degree f. They then permute the rows of their matrices ${}_{\mathscr{A}}\Gamma, {}_{\mathscr{B}}\Gamma$, and ${}_{\mathscr{C}}\Gamma$ according to this permutation, and subsequently, encode their data to derive $A' = A({}_{\mathscr{A}}\Gamma\circ\sigma)^{\dagger}\in\mathbb{C}^{n_{\mathscr{A}}\times(f+k)}$, $B' = B({}_{\mathscr{B}}\Gamma\circ\sigma)^{\dagger}\in\mathbb{C}^{n_{\mathscr{A}}\times(f+k)}$, and $C' = C({}_{\mathscr{C}}\Gamma\circ\sigma)^{\dagger}\in\mathbb{C}^{n_{\mathscr{C}}\times(f+k)}$.

Alice, Bob and Charlie then send A', B' and C' to the central server. Denoting the i^{th} row of a matrix as A_i, which denotes an image in our setting, we propose the following.

Theorem 1. Given A', B' and C', the central server can correctly compute the Linear, Gaussian, the Polynomial and the Rational Quadratic Kernels for the distributed input data.

Proof. Without loss of generality, let us consider the kernels to be computed between Alice and Bob. Let us denote an image from Alice as $a = A_i$, and Bob as $b = B_j$. Then representing the encoded images as $a' = A'_i$ and $b' = B'_j$, we propose that the functions to be computed by the central server are as follows.

$$ K^{\dagger}_{lin}(a',b') = [(A'_i)(B'_j)^{\dagger}],\quad K^{\dagger}_{RBF}(a',b') = \exp\big(-\gamma d_{ij}(A',B')\big), $$
$$ K^{\dagger}_{poly}(a',b') = \Big(1+[(A'_i)(B'_j)^{\dagger}]\Big)^{d},\quad K^{\dagger}_{RQ}(a',b') = \gamma^2\left(1+\frac{d_{ij}(A',B')}{2\alpha l^2}\right)^{-\alpha}, $$

where

$$ d_{ij}(A',B') := \big([(A'_i)(A'_i)^{\dagger}]+[(B'_j)(B'_j)^{\dagger}]-[(A'_i)(B'_j)^{\dagger}]-[(B'_j)(A'_i)^{\dagger}]\big). \tag{I.1} $$

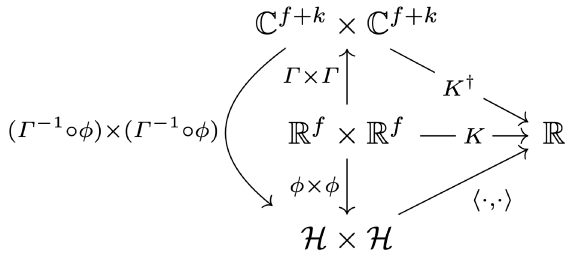
To show that these are well defined kernels, we prove the correctness of the computed functions. Without loss of generality for $P, Q \in \{A, B\}$ and any p, q -

$$[(P'_p)(Q'_q)^\dagger] = \left[(P')\left(_{\mathscr{Q}}\Gamma\sigma(Q_q)^\dagger\right)\right]_p = \left[P\sigma^\dagger\left(\bigoplus_{k=1}^{q} {}_{\mathscr{P}}O^\dagger_{kQ}O_k\right)\sigma Q_q\right]_p = \left[P_p Q_q\right]. \tag{I.2}$$

Hence, it follows that

$$\begin{aligned}
[(A'_i)(B'_j)^\dagger] = ab^T, \quad [(B'_j)(A'_i)^\dagger] = ba^T,\\
[(A'_i)(A'_i)^\dagger] = aa^T, \quad [(B'_j)(B'_j)^\dagger] = bb^T,\\
K^\dagger_{lin}(a', b') = K_{lin}(a, b), \quad K^\dagger_{RBF}(a', b') = K_{RBF}(a, b),\\
K^\dagger_{poly}(a', b') = K_{poly}(a, b), \quad K^\dagger_{RQ}(a', b') = K_{RQ}(a, b).
\end{aligned}$$

This shows the correctness of the computed functions, hence theoretically satisfying our ***Correctness*** requirement. Further, to show that these functions are well defined kernels, we note that the map $\Gamma : \mathbb{R}^f \to \mathbb{C}^{f+k}$ defined by ${}_{\mathscr{P}}\Gamma(v) = v'$ for $v \in \mathbb{R}^f$ and $P \in \{\mathscr{A}, \mathscr{B}, \mathscr{C}\}$ is a bijection, and since K_{lin}, K_{RBF}, K_{poly} and K_{RQ} are well defined kernels, so are $K^\dagger_{lin} = K_{lin} \circ (\Gamma^{-1} \times \Gamma^{-1})$, $K^\dagger_{RBF} = K_{RBF} \circ (\Gamma^{-1} \times \Gamma^{-1})$, $K^\dagger_{poly} = K_{poly} \circ (\Gamma^{-1} \times \Gamma^{-1})$, and $K^\dagger_{RQ} = K_{RQ} \circ (\Gamma^{-1} \times \Gamma^{-1})$ on the range of $\Gamma \times \Gamma$.



Hence we have the above commutative diagram.

Data Adaptability

In our methodology, where Alice, Bob and Charlie contribute datasets A, B, and C, introducing additional data X from Charlie or an additional participant is executed naturally. The shared seed is used to generate X' which can be used for further kernel computations with the existing data. Only specific portions of the kernel are updated, thus saving computational resources. When the data is dynamic and changes, OKRA requires new masks to be created, represented by Γ_t for training iteration t. This secures the privacy of data alterations from each contributing party during further model training. Additionally, if a participant withdraws data, the appropriate contribution is removed, aligning with standard data protection standards. Hence we satisfy the ***Data Adaptability*** requirement.

I.3.3 Privacy Analysis

As stated earlier, OKRA operates under the environment consisting of a proper subset of semi-honest participants, and/or a non-colluding semi-honest central server. For this purpose, we will show that it guarantees security in both settings.

It is essential to highlight that our privacy model excludes certain extreme adversarial conditions. We exclude data distributions where the number of features or the training data of one or more input parties can be guessed, and protocols with only one input party.

Theorem 2. The proposed methodology is secure against a semi-honest adversary who corrupts the central server.

Proof. A semi-honest central server is only the receiver of the encoded data from the input parties, and follows the protocol as intended. Without loss of generality, let there be two input parties Alice and Bob with input data $A \in \mathbb{R}^{n_{\mathscr{A}} \times f}$ and $B \in \mathbb{R}^{n_{\mathscr{B}} \times f}$, respectively, where $n_{\mathscr{P}}$ is the number of samples with the corresponding party $\mathscr{P} \in \{\mathscr{A}, \mathscr{B}\}$ and f is the number of features in each image. The semi-honest function party receives the projected data from Alice and Bob. These consist of $A' = A\gamma^{\dagger} \in \mathbb{C}^{n_{\mathscr{A}} \times (f+k)}$ and $B' = B\gamma^{\dagger} \in \mathbb{C}^{n_{\mathscr{B}} \times (f+k)}$ where $k > 0$. Then, it computes the kernel functions mentioned above in Theorem 1. The data to which the function party has access then includes

(a) A' and analogously, B',
(b) $AB^T = (BA^T)^T$, AA^T and analogously BB^T.

Regarding (a), it is evident that A' does not reveal the number of features of A. We now show that A' is not produced by a unique pair A and ${}_{\mathscr{A}}\Gamma \circ \sigma$.

Given a unitary matrix $U \in \mathbb{C}^{f \times f}$ with $f > 1$, for $\tilde{A} = AU$ and ${}_{\mathscr{A}}\tilde{\Gamma} = \Gamma \circ \sigma \circ U$, we have $A' = A({}_{\mathscr{A}}\Gamma \circ \sigma)^{\dagger} = \tilde{A}\,{}_{\mathscr{A}}\tilde{\Gamma}^{\dagger}$. In the context of reconstruction attacks using analogous data, consider a sample dataset $\hat{A}$ that is drawn from a distribution similar to A. In the context of reconstructing A, the goal is to find a transformation matrix R such that $A'R = \hat{A}$ approximates A. However, since $\hat{A}$ and A do not exist within the same feature space, this leads to loss of information, rendering the approximation ineffective. Further, since Γ consists of orthogonal vectors, the inability to approximate A can also be attributed to the unbalanced orthogonal procrustes problem which is unsolved [126].

Regarding (b), we adopt the proof from [55]. The matrices that produce these symmetric matrices are not unique, since for any unitary matrix $U \in \mathbb{C}^{(f+k)\times(f+k)}$ where $f > 1$, labeling $\tilde{A} = A'U$ and $\tilde{B} = B'U$, we have

$$\tilde{A}\tilde{A}^T = AA^T, \quad \tilde{B}\tilde{B}^T = BB^T, \quad \tilde{A}\tilde{B}^T = AB^T.$$

Similar to the prior argument, these results can also be derived from data in a different feature space. As a result, the function party is limited to learning only the singular values and vectors of the matrices. This means it can derive U and S from the singular value decomposition $A = USV^T$ by eigen-decomposing AA^T. Yet, this information is not adequate

to deduce A since V is unknown. Hence, the function party does not gain insights into (i) the input data or (ii) the number of the features.

Theorem 3. The proposed methodology is secure against a proper subset of semi-honest participants.

Proof. Since the proposed methodology follows one-shot federated learning with no communication back from the central server, the uni-directional flow of data prohibits the corrupt participants from learning anything about the data of the non-corrupt participants.

Therefore, we've shown that the data of non-corrupt participants can not be learned by corrupt semi-honest participants, and that the central server doesn't learn the image sizes, satisfying our ***Privacy*** requirement.

I.4 Results

This section presents details about the implementation of OKRA. Additionally, the evaluation of OKRA's accuracy with regards to model quality, performance alongside a run-time analysis is reported. Experiments were run on an AMD 7713 at 2.0GHZ. We allocated one CPU core and 16 GB of RAM. This reflects our envisioned scenario: multiple hospitals collaboratively training a classifier with modest resources available. The communication settings encompassed a bandwidth of 1.25MBps, with an average latency of $0.1s$ and a packet loss of 2%. Unless otherwise stated, we implemented OKRA for a scenario involving three participants and a central server. To mimic the network communication between the participants and the central server, we implemented each party as an isolated process communicating with others via TCP connections. Our experiments utilize multiple threads to encode the data according to the respective protocol and forward it to the central server. Finally, the central server computes the kernel functions on the whole data to train a machine learning model.

We consider the experiments successful if OKRA's accuracy is exact to non-private classification methods and the run-times at each stage surpass those of other state-of-the-art randomized encoding algorithms, ESCAPED [55] and FLAKE [56].

I.4.1 Datasets

To demonstrate OKRA's applicability to various datasets, we experimented with different medical datasets, which have strong privacy concerns.

We used two clinical image datasets with different label types and data distributions. Given the sensitivity of medical data, these datasets have inherent privacy concerns. The first dataset consists of preprocessed MRI (Magnetic Resonance Imaging) images capturing various stages of Alzheimer's disease [78]. This dataset is instrumental in either predicting

the disease's stages or determining its presence. The second dataset features augmented images of white blood cells, representing four distinct cell types [79]. The primary objective of this dataset is the accurate classification of cell types. Both datasets are suitable for binary or multi-class classification tasks. Details about the data statistics are available in Table I.1. For run-time analysis, we used **synthetic datasets** created with a random number generator. These synthetic datasets are crucial for benchmarking purposes without the influence of real-world data irregularities. The generated data is balanced, encompassing multi-class clusters of data points drawn from a multivariate normal distribution with a mean of 0 and a variance of 1 for each attribute. We generated 400 samples per participant with 1000 features.

I.4.2 Correctness of the Model

Dataset (Samples × Features)	Label Type	Kernel	Naive		OKRA	
			F1	ROC AUC	F1	ROC AUC
Alzheimer's Disease	Binary	Gaussian	0.91 ± 0.03	0.97 ± 0.01	0.91 ± 0.03	0.97 ± 0.01
$(6400 \times 256 \cdot 256)$	Multi-class	Gaussian	0.97 ± 0.04	1.00 ± 0.00	0.97 ± 0.04	1.00 ± 0.00
Blood Cell						
$(12500 \times 100 \cdot 100 \cdot 3)$	Multi-class	Gaussian	0.88 ± 0.01	0.94 ± 0.01	0.88 ± 0.01	0.94 ± 0.01
Synthetic						
$(200 - 51200 \times 12800)$	Multi-class	Linear	0.89 ± 0.03	0.95 ± 0.02	0.89 ± 0.03	0.95 ± 0.02

Table I.1: Dataset statistics and performance metrics for both Naive SVM and OKRA classification methods.

Before beginning the run-time experiments, we aimed to validate OKRA's correctness on kernel learning methods. In our experiments, we explored classification using SVM and dimensionality reduction with Kernel-PCA. We tested various combinations of datasets, methods, and kernels. The kernels employed were Gaussian, Polynomial, and Linear kernels. Each experiment was conducted 10 times, with the averaged results and respective error margins reported. Note that seeds were set to ensure deterministic behavior and reproducible results.

For the classification task, OKRA-SVM was tested against a Naive SVM. Both implementations shared identical hyperparameters $C \in \{2^{-4}, ..., 2^{10}\}$ (misclassification penalty) and $p \in \{1, ..., 5\}$ (degree), optimized using Grid Search. Both were trained with a 5-fold cross-validation. For this implementation, we used Sequential Minimal Optimization (libsvm) provided by scikit-learn [127]. We applied Macro Averaging when the datasets had an unbalanced distribution of classes. Correspondingly, we have evaluated the balanced datasets with Micro Averaging. Table I.1 shows that OKRA-SVM and the naive classifier produce the same F1 and ROC AUC scores.

For the dimensionality reduction, we compared a naive Kernel-PCA and OKRA-PCA. A

Dataset	Kernel	OKRA	
		MSE	R^2
Alzheimer's Disease	Polynomial	0.0	1.0
Blood Cell	Gaussian	0.0	1.0

Table I.2: Performance metrics for OKRA-PCA

kernel is computed, followed by eigenvalue decomposition to sort eigenvalues and eigenvectors by magnitude. The top n eigenvectors are then chosen to transform the data into the new principal component space. To assess utility, the principal components of both Kernel-PCA and OKRA-PCA, utilizing the same kernel, were compared by computing the Mean Squared Error (MSE) and R-squared (R2). This provides a quantitative comparison of the two PCA methods in terms of their ability to capture the data's variance and structure. Table I.2 indicates that both OKRA-PCA and naive PCA are highly consistent and produce identical results, explaining the same patterns in the data. As expected, the encoding has no influence on the results. Hence we experimentally satisfy the ***Correctness*** requirement.

I.4.3 Performance Analysis

To ensure that OKRA's computational overhead does not limit its applicability, especially with high-dimensional datasets like images, we compared OKRA's run-time with that of FLAKE and ESCAPED.

We are interested in two types of run-time: The *Encoding time*, which represents the overhead incurred by encoding data using OKRA for an individual participant. Additionally, we aimed to measure the *Total time*, including communication, from when the server initiates connections to the completion of the kernel function computation.

Our run-time analysis explored the impact of varying the number of participants and image sizes.

Scaling up the Number of Participants

Evaluating our method's adaptability across multiple participants is imperative. In Figure I.2, we present the cumulative time required for multiple participants to perform data encoding, transmission, and linear kernel function computation. Each participant encodes a dataset consisting of 400 samples with 1000 features. FLAKE, despite operating under a one-shot federated learning scheme, lags in efficiency due to its encoding process. Because ESCAPED requires peer-to-peer communication among all clients, its performance also deteriorates due to the increasing overhead with each additional party. OKRA's performance is superior: Even with 10 participants, the execution takes only 16 seconds on average.

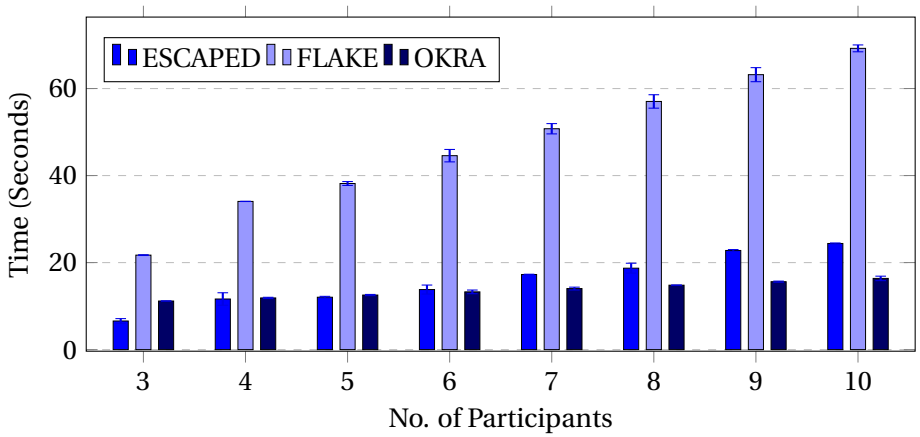


Figure I.2: Runtime comparison with increasing participants.

Scaling up the Image Size

Medical image data is typically captured at high resolutions, essential for medical professionals to accurately diagnose and make treatment decisions. In preprocessing, these images are usually scaled down due to memory requirements or computational efficiency. Rescaled images typically range from (32×32) or (64×64) up to (256×256) [128, 129].

To find out if OKRA can encode preprocessed medical images without burdening the data holder, we experiment with increasing image sizes - (8×8), (16×16), (32×32), (64×64), (128×128), (256×256), and (512×512). For ML tasks on grayscale images, this results in 128, 512, 2048, 8192, 32768, 131072, 524288 features ($h \times w \times c$). Figure I.3 reports the encoding time of OKRA, FLAKE and ESCAPED for one participant with 400 images. OKRA encodes images of size (128×128) in 2.57 seconds, while ESCAPED takes 12.76 seconds. FLAKE is not capable to encode anything beyond (64×64) within 8 hours of the allocated resources, while ESCAPED encodes images up to (256×256). This leads to the conclusion, that OKRA is applicable to medical imaging, while state-of-the-art methods are not. We, therefore, meet both requirements ***Data Adaptability*** and ***Efficiency***.

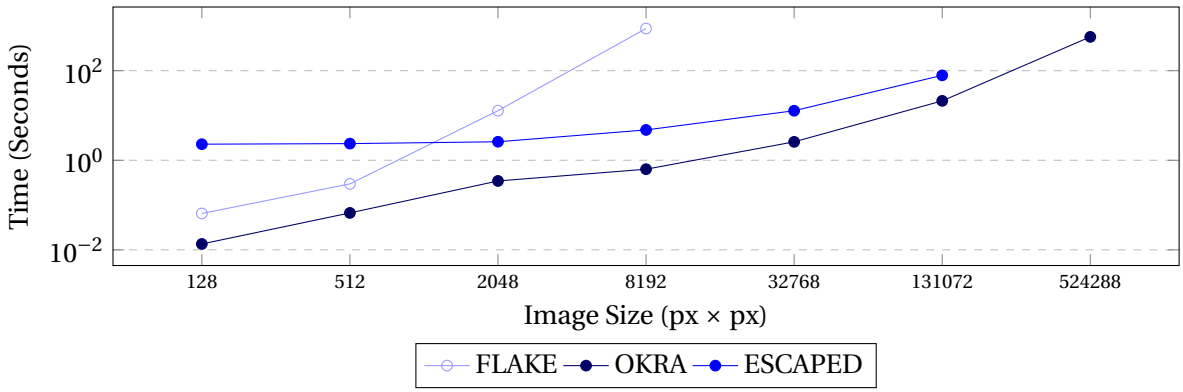


Figure I.3: Encoding times against image sizes.

I.4.4 Discussion

Randomized encoding methods, used for privacy-preserving kernel learning, offer the advantage of computational efficiency while maintaining model accuracy. In the field of randomized encoding methods, our proposed solution, OKRA, demonstrates significant advantages. It consistently outperforms both FLAKE and ESCAPED without compromising utility. Notably, OKRA minimizes the need for extended communication rounds and supports data adaptability, issues primarily encountered with ESCAPED. Additionally, OKRA achieves more efficient data encoding than FLAKE and can handle larger image sizes. With these strengths, OKRA emerges as a robust and efficient option within the domain of randomized encoding techniques for image data.

In subsequent work, we aim to explore the challenges associated with the colluding central server scenario. Further, we intend to address malicious participant behavior, where encoded data is altered to cause inaccuracies in the server's results. Additionally, we plan to explore the applicability of OKRA to other kernel methods, given its promising potential.

I.5 Conclusion

Privacy preservation for the analysis of distributed medical images remains a critical challenge. While randomized encoding offers a promising avenue, existing methodologies often grapple with the trade-off between accuracy and efficiency. Addressing this gap with regards to kernel functions applied to high-dimensional data, we introduced OKRA, a novel algorithm operating within a one-shot federated learning paradigm. OKRA not only privately computes exact kernel functions but also ensures that kernel-based machine learning algorithms are trained efficiently. Empirical evaluations further validate the superiority of OKRA, demonstrating its potential to achieve accuracy levels comparable to centralized machine learning models while minimizing computational burdens. As the landscape of machine learning continues to evolve, OKRA's robustness and efficiency position it as a pivotal tool in privacy-preserving endeavors.

Code availability

Our code is available on GitHub at the following URL: https://github.com/mdppml/Okra [130].

Acknowledgements

This study is supported by the German Federal Ministry of Research and Education (BMBF), under project number 01ZZ2010.

II PP-GWAS: Privacy Preserving Multi-Site Genome-wide Association Studies

Arjhun Swaminathan Anika Hannemann Ali Burak Ünal Nico Pfeifer Mete Akgün

Abstract

Genome-wide association studies help uncover genetic influences on complex traits and diseases. Importantly, multi-site data collaborations enhance the statistical power of these studies but pose challenges due to the sensitivity of genomic data. Existing privacy-preserving approaches to performing multi-site genome-wide association studies rely on computationally expensive cryptographic techniques, which limit applicability. To address this, we present PP-GWAS, a privacy-preserving algorithm that improves efficiency and scalability while maintaining data privacy. Our method leverages randomized encoding within a distributed framework to perform stacked ridge regression on a linear mixed model, enabling robust analysis of quantitative phenotypes. We show experimentally using real-world and synthetic data that our approach achieves twice the computational speed of comparable methods while reducing resource consumption.

II.1 Introduction

Genome-wide association studies (GWAS) have emerged as a critical instrument for discerning the genetic components that underlie complex biological traits and diseases. By investigating differences in allele frequencies of genetic variants, particularly single-nucleotide polymorphisms (SNPs), between ancestrally similar individuals exhibiting distinct phenotypic traits, GWAS have highlighted numerous genomic risk loci associated with a variety of diseases and characteristics [131, 132, 133]. The power of these studies is realized especially when multiple datasets are collaboratively analyzed, as such joint efforts have consistently revealed a broader spectrum of associations than when individual datasets are studied in isolation [134, 135].

Nevertheless, despite the potential advantages, multi-site dataset collaborations in the realm of GWAS are rarely pursued. This can be attributed predominantly to stringent institutional policies and regulations, such as the General Data Protection Regulation (GDPR) in the European Union, which act as obstacles to the sharing of sensitive genetic data [93]. The emphasis on privacy is not exclusive to the European Union. Other jurisdictions, including in Africa, have started to bolster privacy protections as a response to the growing awareness of the potential misuse of sensitive data [136]. This global move towards stringent data protection presents an important discussion: on one hand, there is the undeniable potential of collaborative GWAS in advancing medical science, and on the other, there is the indispensable need to safeguard individual privacy [137].

A well-established technique for multi-site data collaborations in the context of genomic studies is meta-analysis [138], which combines summary statistics from independent GWAS to identify associations in the total combined sample. Although it can mitigate some privacy concerns by avoiding the direct exchange of individual-level data, meta-analysis

is susceptible to biases arising from heterogeneous cohorts, varying sample sizes, and differing imputation or phenotyping strategies [139, 140]. These discrepancies can impair the consistency of estimated genetic effects, highlighting the need for approaches that analyze data jointly while still preserving privacy.

Thus, a growing interest [141, 142, 143, 144] in secure computation for collaborative multi-site GWAS has led to solutions such as [141], S-GWAS [80], FAMHE [81], and SF-GWAS [82]. S-GWAS [80] was one of the first practically feasible frameworks designed for large-scale data. It relies on a secure multiparty computation (MPC) [94] backbone: multiple computational nodes hold secret shares of the original data and cooperate in such a way that no individual's genetic or phenotypic information is exposed. A key factor in S-GWAS's efficiency is its adaptation of Beaver triples, a widely used multiplication technique in MPC, generalized to handle exponentiation and other higher-order operations essential to genomic analyses. Further, S-GWAS uses pseudo-random generators to help mitigate the typical communication overhead associated with MPC. To address population stratification, S-GWAS employs random projection methods that reduce the dimension of genotype matrices before running principal component analysis (PCA). Operating under a non-colluding semi-honest model, S-GWAS is best suited for quantitative traits; for binary traits, the authors propose a two-stage procedure: first, Cochran-Armitage trend tests narrow down candidate variants, then logistic regression is applied only to that reduced subset.

FAMHE [81] subsequently explored the use of homomorphic encryption (HE) [145] to achieve privacy-preserving GWAS. In FAMHE, each computational node can run operations locally on its unencrypted data before encrypting intermediate results with HE and sharing them. These intermediate encrypted values are then aggregated and redistributed for further computation. While FAMHE eliminates many of MPC's communication hurdles and excels at additive and multiplicative operations, it contends with considerable computational overhead and must approximate non-linear operations (such as those required for logistic regression), diminishing precision.

Building on both S-GWAS and FAMHE, SF-GWAS [82] strengthens the architecture further by integrating federated learning principles alongside MPC and MHE. It addresses one of the main drawbacks of purely homomorphic strategies—namely, the difficulty of non-linear operations like division and comparisons—by partitioning the analysis pipeline. Homomorphic encryption handles additions and multiplications on encrypted data, while dedicated MPC routines perform divisions, comparisons, and other operations more cumbersome for MHE alone. SF-GWAS also provides two key workflows: a PCA-based approach that uses linear regression for quantitative traits and logistic regression for binary traits, and a linear mixed model (LMM)-based workflow inspired by REGENIE [59], which relies exclusively on linear regression for quantitative traits. As a result, SF-GWAS offers improvements in terms of practical performance and versatility compared to earlier methods, while still preserving data privacy under an all-but-one semi-honest adversarial model.

However, despite these advances in privacy-preserving multi-site GWAS, methods relying on MPC and MHE still pose practical challenges, which become especially pronounced

when handling large-scale datasets. MPC often requires frequent communication among participants and may need reconfiguration when new data providers join, whereas MHE demands specialized on-premise computational resources that many healthcare institutions may lack [100, 101].

With the challenges presented, our work seeks to integrate GWAS into a distributed architecture where a single third-party helper node is tasked with helping data providers carry out multi-site GWAS in a privacy-preserving manner. Hence, we introduce PP-GWAS as an alternative to state-of-the-art solutions, that aims to perform association tasks for quantitative traits with high accuracy and reduced computational strain. We evaluate our method against S-GWAS [80] and its more powerful successor, SF-GWAS [82]. Unlike S-GWAS and SF-GWAS that utilize MPC and MHE to perform secure multi-site GWAS, our method relies on randomized encoding in a distributed architecture, resulting in improved efficiency and lower computational demands.

Randomized encoding [146, 147] achieves privacy-preservation by obfuscating data in a lower/higher dimensional space. The encoding is dependent upon the analysis performed on the data, and hence establishing security depends on the encoding used [52]. This translates into a dynamic challenge of identifying potential vulnerabilities and attacks rather than proving robustness from the outset. In our work, we use randomized encoding to obfuscate the data, as in other applications such as [148, 149, 56, 1]. By employing this approach, we shift the computational burden away from the intensive multi-round communication or specialized hardware requirements typical of MPC and MHE, making our approach more accessible to resource-limited healthcare institutions.

We adapt a well-established centralized GWAS algorithm based on Linear Mixed Models, REGENIE [59] into a distributed and privacy-preserving setting. We do this since REGENIE is particularly adept at managing large-scale datasets. It employs a two-step methodology, where one first performs ridge regression on the whole genome data to arrive at a smaller space of predictions. Subsequently, another round of ridge regression is performed on these predictions in a stacked fashion, and the SNPs are individually tested.

In this work, we evaluate PP-GWAS against its alternatives on both synthetic data, generated using pysnptools [150], and two real world datasets: a Bladder Cancer Risk dataset, and an Age-Related Macular Degeneration (AMD) dataset. Our empirical findings highlight a notable advancement in both scalability and execution speed, with PP-GWAS performing nearly twice as fast as SF-GWAS. Most importantly, these speeds are achieved utilizing computational resources that are considerably lesser than what SF-GWAS necessitates, making our approach more pertinent to real-world scenarios. Moreover, the accuracy of our GWAS results are validated against REGENIE and meta-analysis, ensuring comprehensive evaluation. Further, in terms of the adversarial conditions, SF-GWAS operates within an all-but-one semi-honest adversarial model, and incorporates an external node designated as helper. However, the potential for malicious intent from this external node remains ambiguous. In contrast, our approach distinctly outlines the role of the external node, categorizing it as both non-colluding and semi-honest. This clear description not only ensures the method's preciseness, but also aligns with standard privacy enhancing techniques in

distributed frameworks [151, 137].

II.2 Results

II.2.1 Experimental Setup

Most of our experiments, unless mentioned otherwise, were conducted on a state-of-the-art high-performance computing (HPC) cluster. Each node within this HPC environment was equipped with an Intel XEON CPU E5-2650 v4, complemented by 256 GB of memory and a 2 TB SSD storage capacity. We employed Python as the primary programming language, taking advantage of Intel's Math Kernel Library (MKL) for high-demand computational tasks. A dynamic core allocation strategy was utilized for MKL-based operations, enhancing computational efficiency and throughput. To ensure the robustness and reproducibility of our experimental findings, each experiment was conducted five times, and the results were averaged. Error bars in the runtime figures represent deviations from these multiple iterations, reflecting the consistency of our measurements. Our runtime comparisons prominently include SF-GWAS (PCA-based), with the reported execution times for SF-GWAS sourced directly from their original publication [82].

The architecture of our experimental system was distributed across multiple nodes of the HPC cluster. The server was allowed to access 128GB of memory for all experiments while the other individual nodes utilized memory variably, utilizing up to a maximum of 32GB unless specified otherwise. Such a configuration is reminiscent of real-world scenarios where computational tasks are commonly outsourced by medical and research institutions (Figure II.1). This design also mirrors the setup described in SF-GWAS, though with more constrained memory allocations for the nodes.

Communication between the server and the nodes was facilitated through socket programming, implemented using TCP connections. Each node established a connection to the server through a unique port. The server, leveraging multiprocessing capabilities, managed simultaneous data exchanges with multiple nodes. This approach ensured real-time interactions and minimized node idleness. The communication was characterized by a round-trip latency of 0.249ms, with the TCP window size set at the default 128 KByte. For matrix operations, especially involving large sparse datasets, we integrated the sparse-dot-mkl library [152].

In summary, our experimental design was crafted to leverage the capabilities of the HPC cluster, drawing parallels with the setup detailed in SF-GWAS. By optimizing computational resources and ensuring efficient communication protocols, our aim was to create a versatile system, adept at addressing the stringent demands of privacy-preserving genome-wide association studies.

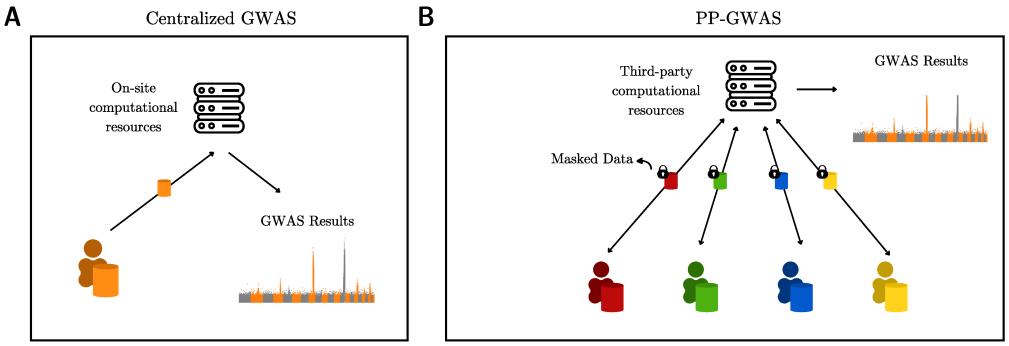


Figure II.1: **Comparison of centralized GWAS and PP-GWAS. (A)** A centralized approach is depicted where a single institution, such as a hospital or research institute, utilizes on-site computational resources to conduct GWAS on its local data. **(B)** A distributed model is illustrated, in which multiple entities collaborate to perform GWAS on a combined dataset. This is achieved without sharing the local data, and by leveraging a third-node service to facilitate computations.

II.2.2 Synthetic Data Generation

Synthetic data for our experiments was generated using the pysnptools library [150] and was simulated to resemble quantitative traits. The population structure was set at 0.1, and the degree of family relatedness was fixed at 0.25. This synthetic data was horizontally partitioned across the nodes. The synthetic datasets varied widely in size, with sample sizes ranging from 9,178 to 275,000, SNP counts ranging from 580,000 to 2,451,176, and the number of covariates ranging from 2 to 40.

II.2.3 Real Datasets

Two real genomic datasets were used for the experiments: A Bladder Cancer Risk dataset (13,060 Samples, 467,172 SNPs) (dbGaP Study Accession: phs000346.v2.p2) [153, 154, 155], and an Age-Related Macular Degeneration dataset (22,683 Samples, 508,740 SNPs) (dbGaP Study Accession: phs001039.v1.p1) [156]. Access to these datasets was secured through the dbGaP platform, adhering to the necessary procedural requirements. These datasets were further imputed for missing data using Beagle [157]. Since our GWAS algorithm is tailored for quantitative data, we treat these real datasets as if they were quantitative. Both dbGaP releases include standard subject-level covariates. For the Bladder Cancer Risk dataset, age, sex, and study-center indicators (capturing platform information) were available; for the AMD dataset, age and sex were available across cohorts. We included these as covariates in our analyses.

Both the synthetic and real datasets were stored blockwise in the .npz format, which our code is designed to read.

II.2.4 Quality Control

For both the synthetic and real datasets, a series of preprocessing steps were performed to ensure appropriate data quality. These are a part of the algorithm, and are included in the runtime analysis in the subsequent section. Genotypes with a missing rate exceeding 0.1 were filtered out. Further, only alleles with a minor allele frequency greater than 0.05 were retained. Lastly, a Hardy-Weinberg equilibrium chi-squared test statistic threshold of 23.928 (corresponding to a p-value of 10^{-6}) was applied. These preprocessing measures were securely executed on the whole data by the nodes using standard addition-based randomized encoding techniques.

II.2.5 Accuracy Analysis

To rigorously evaluate the accuracy of PP-GWAS, it was essential to conduct a comparative analysis against the well-established unencrypted plaintext GWAS algorithm, REGENIE. Using Pearson's r-square coefficient as a measure for accuracy between the negative log of the p-values, PP-GWAS demonstrated robust performance on real-world datasets. Specifically, the Pearson correlation of $-\log_{10}(p)$ between our method and REGENIE across both datasets was $r^2 = 0.999999 \sim 1.00$ (df $= M-2$), $P \sim 0$, 95% CI [0.999999, 1.00], where M is the number of SNPs. These outcomes, illustrating high correlation with the plaintext benchmarks, are detailed in Figure II.2. This comparison highlights the capability of PP-GWAS to maintain genetic association analysis accuracy while ensuring data privacy.

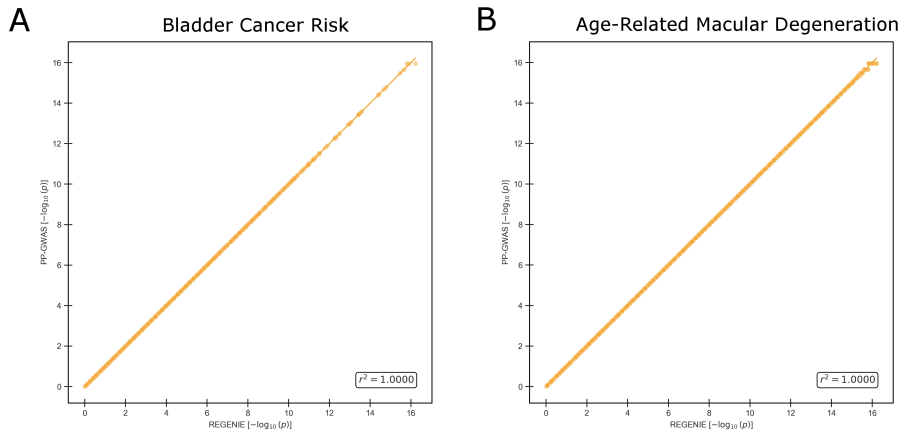


Figure II.2: **Accuracy of PP-GWAS against REGENIE (centralized) on real-world datasets (A)** Scatter (correlation) plot for the Bladder Cancer Risk dataset, showing the SNP-wise agreement between $-\log_{10}(p)$ from PP-GWAS and REGENIE. Pearson correlation (two-sided) between $-\log_{10}(p)$ across SNPs ($M = 467{,}172$): $r = 1.000000$ (95% CI [0.999999, 1.000000]); $t(467{,}170) = 677{,}871.82$, $P < 10^{-6}$; $R^2 = 0.999999$. **(B)** Scatter (correlation) plot for the Age-Related Macular Degeneration (AMD) dataset, depicting the analogous correlation. Pearson correlation (two-sided) between $-\log_{10}(p)$ across SNPs ($M = 508{,}740$): $r = 1.000000$ (95% CI [0.999999, 1.000000]); $t(508{,}738) = 577{,}736.14$, $P < 10^{-6}$; $R^2 = 0.999998$. No multiple-testing correction was applied to these correlation tests.

II.2.6 Scalability Analysis of PP-GWAS with Simulated Data

The ability to maintain both computational efficiency and accuracy with large-scale data is a critical challenge in genome-wide association studies. This section provides a comparative analysis between PP-GWAS and SF-GWAS, focusing on performance under various conditions.

To facilitate a direct comparison, we utilized a simulated dataset designed similar to those in SF-GWAS's scalability analysis. We consider four primary factors in our scalability analysis: the number of computational nodes, the SNP count within the genomic data, the number of covariates within the genomic data and the sample sizes managed by each node. Incremental increases in each of these factors allow us to observe and quantify the performance implications on PP-GWAS.

Our initial evaluation focuses on the algorithm's performance in response to an increasing number of nodes. With a test dataset comprising 9,178 samples and 612,794 SNPs, we assess the algorithm's distributed computation capabilities. Performance outcomes, as shown in Figure II.3 (A), indicate that PP-GWAS maintains a linear performance with an increasing number of nodes.

We then explore the scalability in relation to SNP counts, with a fixed configuration of two nodes and 9,178 samples. Addressing the large-scale nature of many genomic datasets, PP-GWAS's performance remains superior to that of SF-GWAS, as depicted in Figure II.3 (B).

Next, we explore the scalability in relation to the number of covariates, with a fixed genomic dataset size of 9,178 samples and 612,794 SNPs. We note in Figure II.3 (C) that the runtime is unaffected by an increase in the number of covariates since projecting out covariates is done early in our methodology, and is a cheaper operation as opposed to working with the whole genomic dataset.

Lastly, we examine how sample size affects PP-GWAS's scalability. Keeping the number of nodes at two and SNPs constant at 612,794, we increment the sample size and analyze the impact. The performance of PP-GWAS against increasing sample sizes is demonstrated in Figure II.3 (D).

In conclusion, the scalability analysis underscores PP-GWAS's capability to efficiently manage increased computational demands across various dimensions. This is instrumental for its application in extensive genetic association studies.

II.2.7 Adaptability to Large-Scale Data

To address the challenge of scaling PP-GWAS for large-scale genomic analyses, we conducted experiments using synthetic datasets, given the inaccessibility of datasets such as the UK Biobank and eMERGE. For simulations other than the UK Biobank scale, the system was configured with the central server being allocated 256 GB of RAM and six participant nodes each provided with 56 GB of RAM. In contrast, for the UK Biobank-sized

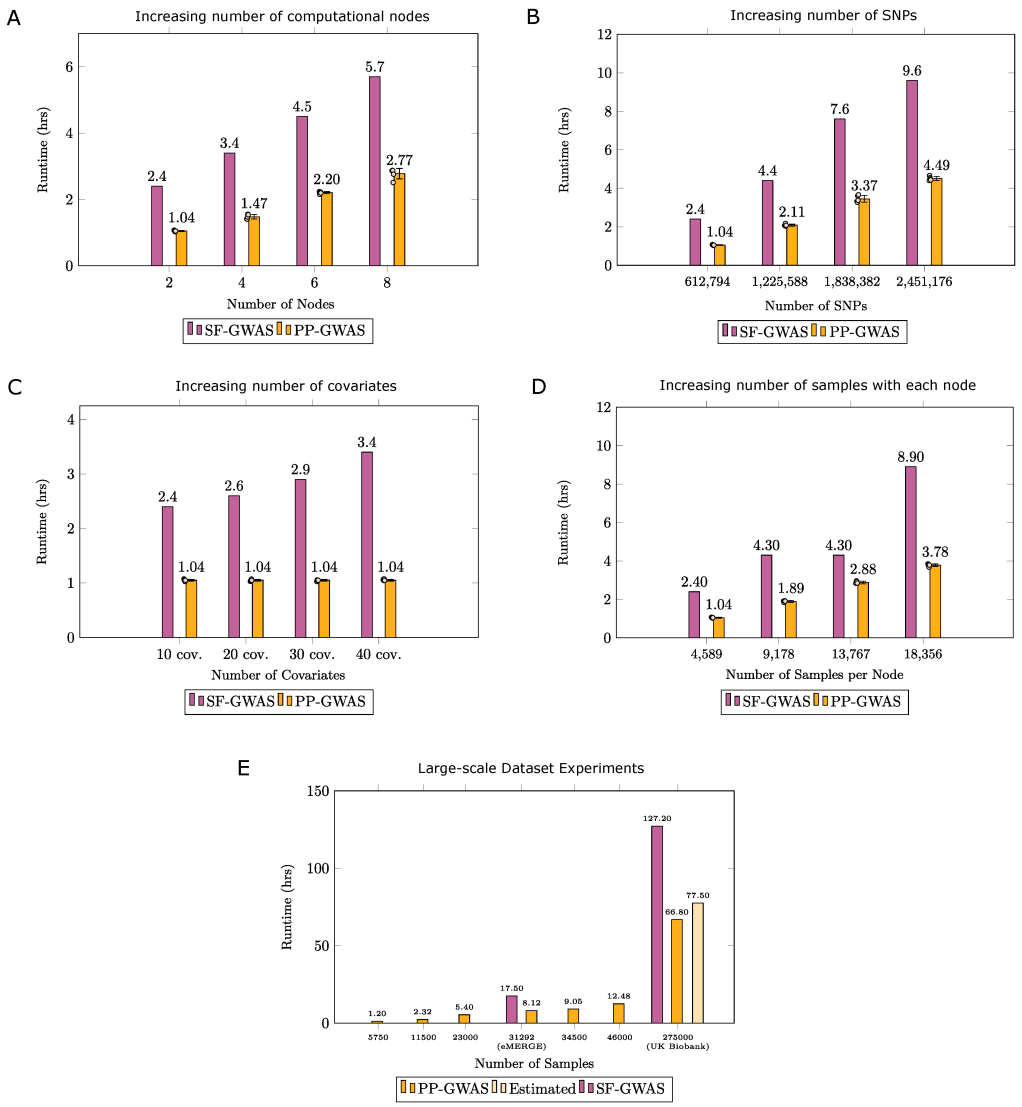


Figure II.3: **Scalability Analysis of PP-GWAS (A)** Comparison of total computational times for SF-GWAS and PP-GWAS, analysing a dataset (9,178 samples × 612,794 SNPs) across varying number of participating institutions. **(B)** Comparison of total computational times for SF-GWAS and PP-GWAS, analyzing a dataset with 9,178 samples across two participating institutions, and an increasing number of SNPs. **(C)** Comparison of total computational times for SF-GWAS and PP-GWAS, analyzing a dataset with 9,178 samples and 612,794 SNPs across two participating institutions, and an increasing number of covariates. **(D)** Comparison of total computational times for SF-GWAS and PP-GWAS, analyzing a dataset with 612,794 SNPs across two participating institutions, and an increasing number of samples. **(E)** Comparison of total computational times for PP-GWAS and SF-GWAS when applied to large-scale datasets equivalent in size to the eMERGE and the UK Biobank datasets. Source data are provided as a Source Data file. *Data presentation and statistics (3a–d):* Bars show mean values and error bars show ± standard deviation of five independent runs of PP-GWAS; all individual runs are overlaid as jittered dots to display the distribution. Statistical summaries are derived from technical (not biological) replicates because the objective is to quantify computational runtime variability.

experiments—which comprised 275,000 samples and 580,000 SNPs—we leveraged deNBI Cloud resources, which consisted of vastly different hardware as compared with what is offered by Google Cloud. Due to the technical constraints, our configuration employed a modified setup with four client nodes, each assigned 256 GB of RAM, alongside a central server equipped with 700 GB of RAM.

Under the deNBI Cloud setup simulating the UK Biobank configuration, PP-GWAS completed the analysis in 2 days 18 hours and 49 minutes, while the simulation configured to represent the eMERGE dataset finished in 8 hours and 7 minutes, as illustrated in Figure II.3(E). These results provide a clear assessment of PP-GWAS's scalability across large scale dataset sizes and different computational environments. Moreover, under linear interpolation, we expect PP-GWAS to complete the UK Biobank dataset sized experiments in 3 days 5 hours and 30 minutes if we had the same computational resources and six-node configuration as SF-GWAS.

II.2.8 Memory Efficiency and Communication Cost Analysis

In the realm of privacy-preserving GWAS, PP-GWAS algorithm presents a notable shift from SF-GWAS, especially in terms of memory efficiency and communication costs. This section goes into how these two critical factors play out in the implementation and scalability of PP-GWAS.

Memory Efficiency: A key strength of PP-GWAS lies in its significantly reduced RAM requirements compared to SF-GWAS, as discussed in Figure II.4 (B). This aspect is particularly advantageous for settings with limited computational resources, such as smaller research institutions or medical facilities. By lowering the memory demands, PP-GWAS enables these organizations to partake in large-scale genetic studies without the need for extensive hardware upgrades. This improvement in memory efficiency is instrumental in democratizing GWAS, allowing for wider and more inclusive research participation.

Communication Costs: As seen in Figure II.4 (A), while PP-GWAS requires higher communication overhead than SF-GWAS when the number of computational nodes is low, this increase is a strategic trade-off. Specifically, the communication demands in PP-GWAS rise linearly and predictably, in contrast to the exponential growth experienced by SF-GWAS as the number of nodes increases. This makes PP-GWAS a more accessible option for many institutions, especially in an era where digital connectivity often surpasses the availability of advanced computational resources. Furthermore, the distributed nature of the PP-GWAS algorithm reduces the number of communication rounds, alleviating some of the burdens seen in SF-GWAS.

II.2.9 Performance in LAN and WAN Settings

Evaluating the performance of PP-GWAS across different network configurations is essential to its applicability in real-world scenarios. Using simulated data, we compared the performance of PP-GWAS to SF-GWAS in both local-area network (LAN) and wide-area

network (WAN) settings using Google Cloud.

For these experiments, we replicated the network setup from SF-GWAS. In the WAN configuration, three computational nodes were distributed across geographically distant regions: two clients located in Iowa (us-central1) and London (europe-west2), and the server in North Virginia (us-east4). For the LAN configuration, all nodes were placed in North Virginia (us-east4). We progressively scaled the dataset size, using sample sizes ranging from 9,178 to 36,712, with 612,794 SNPs. The round-trip latency matched the SF-GWAS setup, measuring 0.3ms in the LAN and up to 100ms in the WAN.

In addition to runtime, we measured the total volume of data transferred between a client and the server in each experiment to understand the communication efficiency of PP-GWAS. The total data transferred increased with sample size: 9,178 samples (188.9.GB), 18,356 samples (377.6.GB), 27,534 samples (566.5.GB), and 36,712 samples (755.6.GB). These values provide an estimate of the communication overhead in general.

Figure II.4 (C) illustrates the runtime performance of PP-GWAS in both LAN and WAN settings relative to SF-GWAS, highlighting its adaptability to varying network conditions.

II.2.10 Performance Evaluation against Meta-Analysis

Here, we evaluate the performance of meta-analysis, which relies on combining individual node association results, and compare it to both centralized GWAS (REGENIE) and PP-GWAS. The comparison is conducted using two real-world datasets, both treated like quantitative data: the Bladder Cancer dataset (Figure II.5) and the AMD dataset (Figure II.6).

For meta-analysis, we utilized PLINK with configurations involving 2, 3, 4, 5, and 6 parties, while PP-GWAS was evaluated with 6 parties. Unlike meta-analysis, PP-GWAS's performance is independent of the number of parties and consistently achieves an r^2 accuracy of 1, demonstrating its robustness.

Our findings highlight that as data becomes more fragmented across an increasing number of parties, the performance of meta-analysis deteriorates. This decline occurs because each node works with progressively smaller sample sizes, leading to worse individual-level summary statistics. In contrast, PP-GWAS maintains high accuracy regardless of the degree of data partitioning.

To further illustrate these performance differences, we conducted additional experiments using a simulated dataset comprising $20,000$ samples and $500,000$ SNPs, distributed across 6 computational nodes. We applied REGENIE, PP-GWAS, and meta-analysis to this dataset and generated the resulting Manhattan plots. We note in Figure II.7 that REGENIE serves as the reference. PP-GWAS exhibits a near-identical distribution. Minor variations in peak cut-offs can be attributed to numerical differences introduced by floating-point arithmetic, which do not impact overall accuracy. In contrast, meta-analysis exhibits weaker association signals and increased variance across detected loci.

These results further validate the advantages of PP-GWAS, demonstrating its ability to

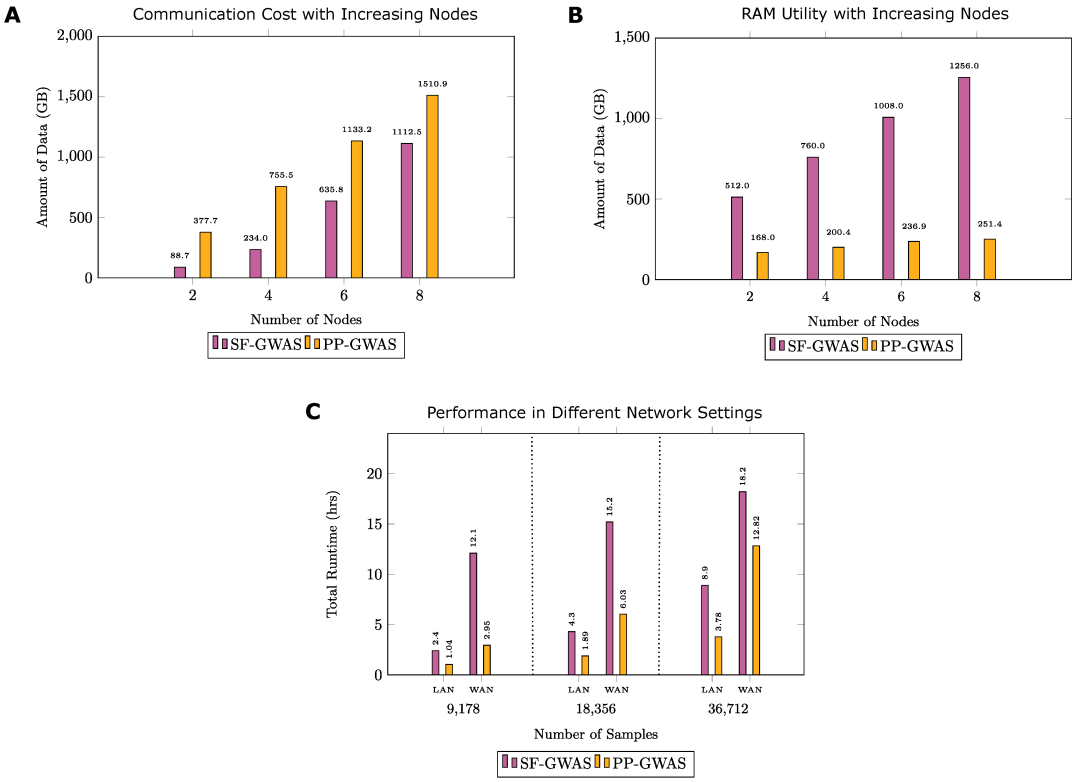


Figure II.4: **Communication Cost, Memory Usage, and Performance Across Network Settings (A)** Comparison of communication cost (in GB) across an increasing number of computational nodes, analysing a genetic dataset consisting of 9,178 samples and 612,794 SNPs. **(B)** Comparison of RAM utility (in GB) across an increasing number of computational nodes, analysing a genetic dataset consisting of 9,178 samples and 612,794 SNPs. **(C)** Comparison of total runtimes of PP-GWAS and SF-GWAS under both LAN and Trans-Atlantic WAN settings, with varying sample sizes. Source data are provided as a Source Data file.

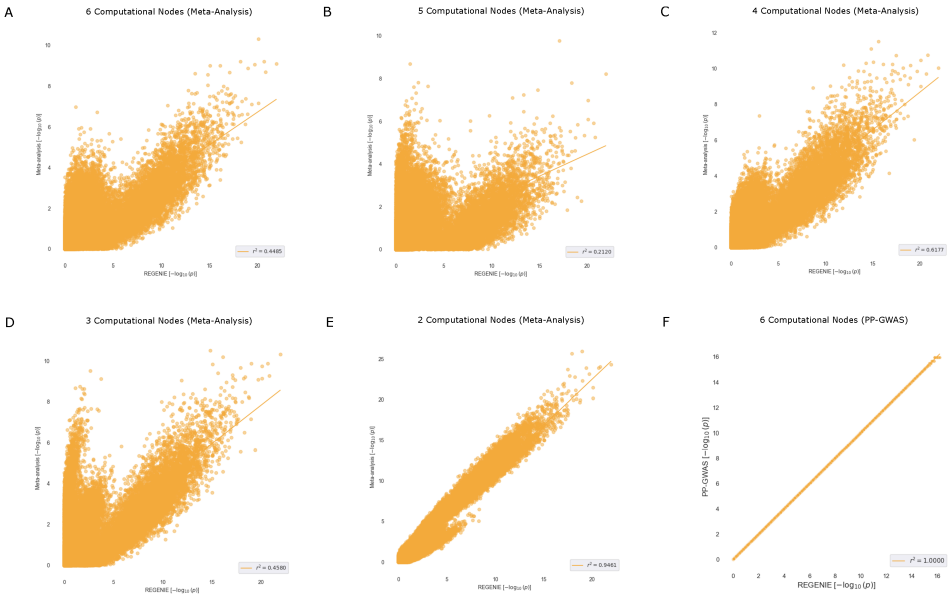


Figure II.5: **Accuracy Comparison of PP-GWAS against Meta-Analysis (A-F)** Scatter correlation plots to compare performance of meta-analysis with varying computational nodes against PP-GWAS on the Bladder Cancer Risk dataset. Pearson correlation (two-sided) between the $-\log_{10}(p)$ values is reported. No multiple-testing correction was applied to these correlation tests.

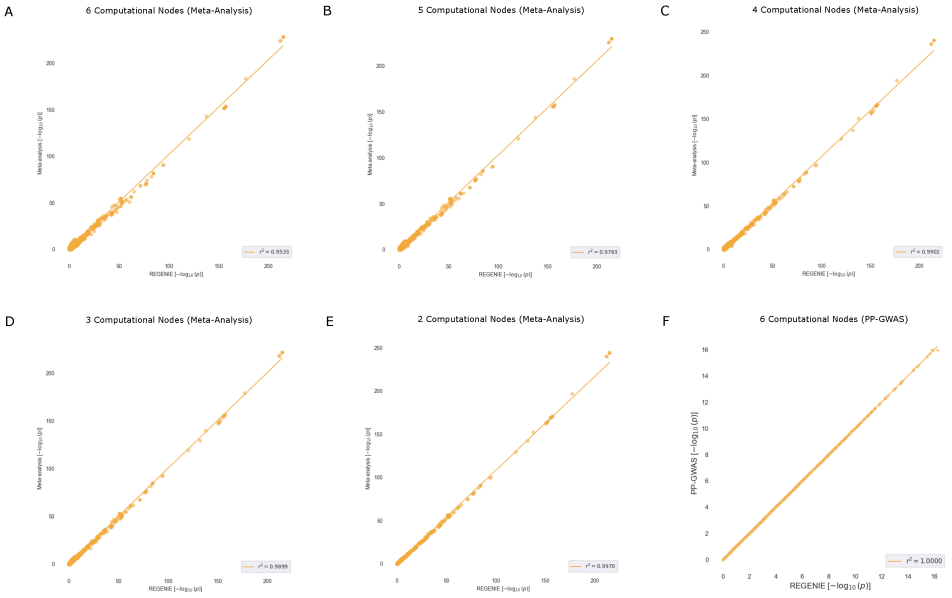


Figure II.6: **Accuracy Comparison of PP-GWAS against Meta-Analysis (A-F)** Scatter correlation plots to compare performance of meta-analysis with varying computational nodes against PP-GWAS on the AMD dataset. Pearson correlation (two-sided) between the $-\log_{10}(p)$ values is reported. No multiple-testing correction was applied to these correlation tests.

achieve accuracy comparable to centralized GWAS while preserving data privacy. Importantly, its robustness to data partitioning highlights its suitability for collaborative genomic studies.

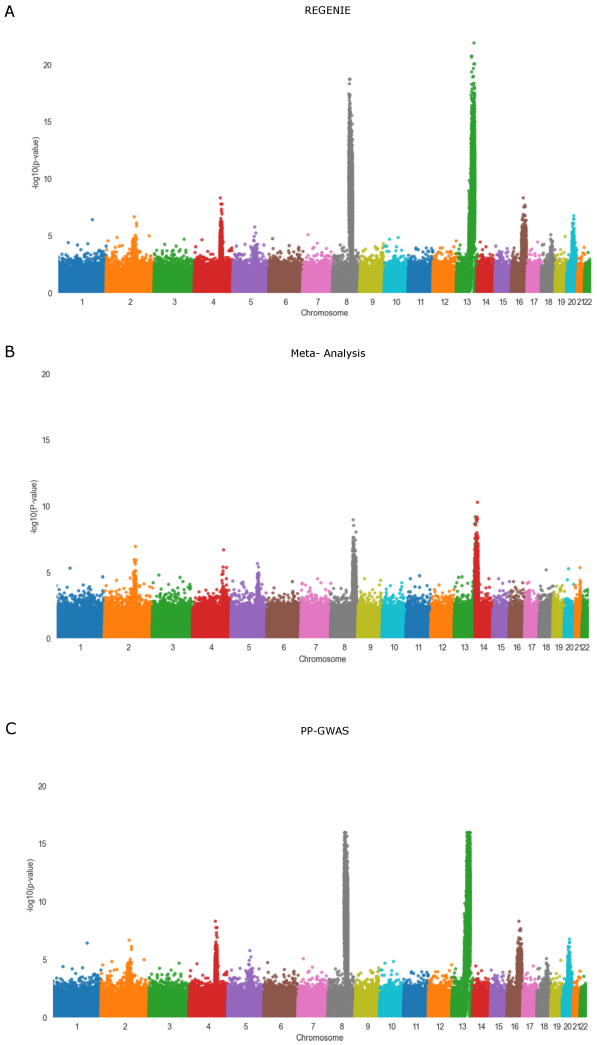


Figure II.7: **Manhattan Plots of REGENIE, Meta-Analysis and PP-GWAS** Manhattan plots display $-\log_{10}(p)$ for single-SNP additive association tests in simulated data ($N = 20{,}000$ samples, $M = 500{,}000$ SNPs). **(A)**REGENIE. p-values arise from the single-SNP association testing as implemented in REGENIE; the null hypothesis is $\boldsymbol{\beta} = 0$, the test statistic follows χ^2 with df = 1, and two-sided p-values are reported. **(B)** Meta-analysis. Per-site SNP effects and standard errors are combined by fixed-effect inverse-variance meta-analysis to a pooled Z statistic (with $Z^2 \sim \chi^2$ under the null hypothesis $H_0 : \boldsymbol{\beta} = 0$); two-sided p-values are reported. **(C)** PP-GWAS. p-values are computed from the distributed single-SNP association test (Box II.3); the reported statistic is χ^2 with df = 1 under the null hypothesis $H_0 : \boldsymbol{\beta} = 0$, yielding two-sided p-values. For all panels, p-values are exact and unadjusted across SNPs.

II.3 Discussion

In this study, we introduced PP-GWAS, a privacy-preserving distributed framework designed to perform multi-site genome-wide association studies on quantitative data. Our extensive comparative analysis demonstrates that PP-GWAS maintains genetic association analysis accuracy equivalent to traditional centralised methods, in the analysis of real-world datasets such as the Bladder Cancer Risk dataset, and the Age-Related Macular Degeneration (AMD) dataset.

PP-GWAS excels in scalability and adaptability when tested against the state-of-the-art privacy-preserving GWAS algorithm SF-GWAS [82]. Through evaluations with varying number of computational nodes, SNP counts, and sample sizes, our framework demonstrated a consistent linear performance increase, proving its effectiveness in multi-site GWAS. This scalability is essential for accommodating the expanding size and diversity of genomic datasets in real-world scenarios, making PP-GWAS a stable solution even under the constraints of limited computational resources. Furthermore, the adaptability of PP-GWAS was tested using synthetic datasets as proxies for large-scale real datasets, predicting feasible processing times for extensive databases such as the eMERGE and the UK Biobank datasets.

Another significant advancement is in memory efficiency and communication costs. PP-GWAS considerably reduces the RAM requirements, enabling institutions with constrained computational resources to participate in genomic research. While it necessitates higher communication overhead than SF-GWAS with less nodes, this overhead progresses in a predictable and manageable linear fashion, which is a strategic compromise for achieving greater computational and memory efficiency. Further, since the communication overhead for SF-GWAS increases rather exponentially, we expect to perform better with more nodes. This trade-off ensures applicability across a broader spectrum of research environments, from hospitals to smaller research institutions.

In addition, our experiments investigating network performance further highlight the strengths of PP-GWAS. Using both local-area network (LAN) and wide-area network (WAN) settings on Google Cloud, we observed that PP-GWAS maintains competitive performance across varying network conditions. These findings confirm the potential for deployment in diverse real-world settings, from localized institutional networks to globally distributed research collaborations.

Our performance evaluation against traditional meta-analysis approaches highlights the superiority of PP-GWAS in terms of accuracy and reliability. While meta-analysis suffers from deteriorating performance as the number of collaborating parties increases, owing to progressively smaller sample sizes per node, PP-GWAS consistently retains accuracy. This performance, even under substantial data fragmentation, underscores the efficacy of PP-GWAS as a powerful solution for collaborative genomic research.

II.3.1 Limitations

PP-GWAS operates on datasets that may be generated by different sites without joint-

genotyping. In such settings, platform- and pipeline–specific biases can induce variant-level discrepancies. We mitigate global batch effects via harmonization (shared positions, alleles, strand, and rsIDs), perform global quality control that retains rare variants present at any participating site, and remove covariate effects using covariate projection which includes site, platform/pipeline, and batch indicators. These steps, which are standard even in centralized analyses where data is pooled from different sources [59, 158, 159, 160, 161, 162, 163], are effective for single-variant association but do not eliminate all effects of technical heterogeneity.

PP-GWAS, as well as other state-of-the-art privacy-preserving distributed GWAS would be most effective when upstream variant calls are produced within a unified framework. The privacy-preserving way to achieve this is a distributed joint-genotyping layer that accounts for platform differences during variant calling without centralising raw data. Designing such a layer e.g., using secure aggregation, multi-party computation, homomorphic encryption, or trusted hardware remains an important direction for future research.

Finally, we do not advocate centralising or sharing raw genotypes for joint genotyping and then returning to a privacy-preserving distributed GWAS workflow. Were genotypes to be shared, the core rationale for privacy-preserving analyses would be undermined. PP-GWAS is therefore intended either (i) for non-jointly genotyped settings with the above mitigations and explicit technical covariates, acknowledging residual confounding may persist, or (ii) to be composed with a privacy-preserving distributed joint-genotyping layer.

II.4 Methods

This research complies with all relevant ethical regulations. Access to dbGaP datasets used in this study, phs000346.v2.p2 and phs001039.v1.p1, was authorized by the NIH dbGaP Data Access Committees (NCI DAC for phs000346; NEI DAC for phs001039).

II.4.1 Linear Mixed Models in Genome-Wide Association Studies

With regards to GWAS, the application of linear mixed models (LMMs) has emerged as a fundamental approach for deciphering the intricate genetic underpinnings of various phenotypes. A standard linear mixed model used for GWAS is described below.

$$\mathbf{y} = \beta_{\text{test}}\mathbf{x}_{\text{test}} + \mathbf{Z}\boldsymbol{\alpha} + \mathbf{g} + \mathbf{e}. \tag{II.1}$$

Here $\mathbf{y}$ represents the phenotype vector of N individuals while $\mathbf{x}_{\text{test}}$ encapsulates the minor allele dosages of the variant being tested, represented as 0, 1, or 2, signifying reference-homozygous, heterozygous, and alternate homozygous alleles, respectively. This is represented as a column vector, similar to $\mathbf{y}$, which are both standardized initially to have mean zero and unit standard deviation. An $N \times C$ matrix $\mathbf{Z}$ accounts for other confounding factors. The polygenic effect $\mathbf{g}$ includes multiple small-effect size variants. Specifically, $\mathbf{g} = \mathbf{X}\boldsymbol{\beta}$, with $\mathbf{X}$ representing the standardized genotypes of m variants. Environmental effects denoted by $\mathbf{e}$, is modeled as Gaussian noise.

Both $\mathbf{x}_{\text{test}}$ and $\mathbf{y}$ are standardized to have zero mean and unit variance. The model incorporates fixed effects (β_{test} and $\boldsymbol{\alpha}$) and random effects ($\mathbf{g}$ and $\mathbf{e}$). The genetic effect uses what is called the kinship matrix $\mathbf{K} = \frac{1}{m}\mathbf{X}\mathbf{X}^\top$, with $\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{0}, (\sigma_g^2/m)\mathbf{I}_{m\times m})$, leading to $\mathbf{g} \sim \mathcal{N}(\mathbf{0}, \sigma_g^2\mathbf{K})$. The environmental effect is modeled as $\mathbf{e} \sim \mathcal{N}(\mathbf{0}, \sigma_e^2\mathbf{I}_{n\times n})$. The variance components σ_g^2 and σ_e^2 represent the polygenic and environmental variances, respectively.

The model's validity is assessed by testing the null hypothesis $H_0 : \beta_{\text{test}} = 0$ for each variant, thus identifying significant associations with the phenotype under study. A pivotal aspect of LMM implementation is the projection of covariates from phenotypes and genotypes, a technique used to remove any confounding effects. This is done by projecting the genomic matrix and the phenotype data to the null space of Z. The projection matrix is formalized as:

$$\mathbf{P} = \mathbf{I}_n - \mathbf{Z}\left(\mathbf{Z}^\top\mathbf{Z}\right)^{-1}\mathbf{Z}^\top \tag{II.2}$$

Post-projection, the model assumes the form:

$$\tilde{\mathbf{y}} = \beta_{\text{test}}\tilde{\mathbf{x}}_{\text{test}} + \tilde{\mathbf{X}}\boldsymbol{\beta} + \mathbf{e} \tag{II.3}$$

where $\tilde{\mathbf{y}} = \mathbf{P}\mathbf{y}$, $\tilde{\mathbf{x}}_{test} = \mathbf{P}\mathbf{x}_{test}$ and $\tilde{\mathbf{X}} = \mathbf{P}\mathbf{X}$. This approach effectively removes the influence of covariates, yielding residuals that more accurately reflect the relevant genetic associations. The LMM-based χ^2 test statistic, central to hypothesis testing, is given by:

$$\chi^2 = \frac{\left(\tilde{\mathbf{x}}_{\text{test}}^\top\mathbf{V}^{-1}\tilde{\mathbf{y}}\right)^2}{\tilde{\mathbf{x}}_{\text{test}}^\top\mathbf{V}^{-1}\tilde{\mathbf{x}}_{\text{test}}} \tag{II.4}$$

where $\mathbf{V} = \hat{\sigma}_g^2\mathbf{K} + \hat{\sigma}_e^2\mathbf{I}_{n\times n}$ given the Maximum Likelihood Estimates $\hat{\sigma}_g$ and $\hat{\sigma}_e$ of the variance parameters σ_g and σ_e.

II.4.2 Stacked Ridge Regression for LMM-based GWAS

The computation of association statistics within the framework of LMMs presents a significant computational challenge. This arises primarily due to the necessity of maximum likelihood estimation of the variance parameter σ_g, which involves large matrix operations. This complexity escalates dramatically for large-scale datasets, often making the computations prohibitively resource-intensive. Traditional efforts in algorithmic development have primarily focused on optimizing the utilization of the kinship matrix, for instance, through matrix factorization methods.

REGENIE [59] employs a stacked ridge regression strategy and achieves an accuracy comparable to established tools such as BOLT-LMM [158], fastGWA [164], SAIGE [165] and FaST-LMM [166]. Since REGENIE is more friendly to distributed datasets, SF-GWAS [82] employed methods from MHE and MPC to build upon the algorithm. We similarly work with REGENIE in a distributed setting.

REGENIE executes its analysis in two phases. The initial phase involves a regression of the contributions from $\tilde{\mathbf{X}}$ out of $\tilde{\mathbf{y}}$, followed by fitting β_{test} on these adjusted residuals to

ascertain associations. To mitigate the computational demands posed by the extensive genome-wide matrix $\tilde{\mathbf{X}}$, REGENIE implements a stacked ridge regression, executed in two distinct phases: Level 0 and Level 1. This approach significantly enhances computational efficiency and adaptability for large-scale genomic datasets, marking a notable progression in the field of genetic association studies.

At *Level 0*, the genotype matrix $\mathbf{X}$ is partitioned into B vertical blocks, denoted as $\tilde{\mathbf{X}} = \left(\tilde{\mathbf{X}}^1, \ldots, \tilde{\mathbf{X}}^B\right)$. A set of R distinct ridge parameters $\{\lambda_1, \ldots, \lambda_R\}$ are then chosen, where

$$\lambda_r := \frac{M(1-h_r^2)}{h_r^2}, \quad h_r := \frac{0.01\,(R-1) + 0.98\,(r-1)}{R-1}. \tag{II.5}$$

Here, M is the number of SNPs in the study. Consequently, R ridge estimators are computed for each block:

$$\hat{\boldsymbol{\beta}}^b_{\lambda_r} = \left(\tilde{\mathbf{X}}^{b\top}\tilde{\mathbf{X}}^b + \lambda_r\,\mathbf{I}_{n\times n}\right)^{-1}\tilde{\mathbf{X}}^{b\top}\tilde{\mathbf{y}}, \tag{II.6}$$

$$\hat{\mathbf{y}}^{(b,r)} := \tilde{\mathbf{X}}^b\,\hat{\boldsymbol{\beta}}^b_{\lambda_r}. \tag{II.7}$$

These intermediate predictors $\hat{\mathbf{y}}^{(b,r)}$ for each block are then aggregated into a global feature matrix: $\mathbf{W}^b := \left(\hat{\mathbf{y}}^{(b,1)}, \ldots, \hat{\mathbf{y}}^{(b,R)}\right)$, $\quad \mathbf{W} := \left(\mathbf{W}^1, \ldots, \mathbf{W}^B\right)$. This is implemented in a k-fold cross validation framework, and hence we denote the kth folds of data as $\tilde{\mathbf{X}}^b_{(LOCO,k)}$ and $\tilde{\mathbf{y}}_{(k)}$, and the data without the kth fold as $\tilde{\mathbf{X}}^b_{(LOCO,k-1)}$ and $\tilde{\mathbf{y}}_{(k-1)}$. Hence, we have

$$\hat{\boldsymbol{\beta}}^b_{(\lambda_r,k-1)} = \left(\tilde{\mathbf{X}}^b_{(\mathrm{LOCO},k-1)}{}^{\top}\tilde{\mathbf{X}}^b_{(\mathrm{LOCO},k-1)} + \lambda_r\,\mathbf{I}_{n\times n}\right)^{-1}\tilde{\mathbf{X}}^b_{(\mathrm{LOCO},k-1)}{}^{\top}\tilde{\mathbf{y}}_{(k-1)}, \tag{II.8}$$

$$\hat{\mathbf{y}}^{(b,r)}_{(\mathrm{LOCO},k)} := \tilde{\mathbf{X}}^b_{(\mathrm{LOCO},k)}\,\hat{\boldsymbol{\beta}}^b_{(\lambda_r,k-1)}. \tag{II.9}$$

At *Level 1*, a subsequent round of ridge regression is conducted on the intermediate feature matrix of size $N \times BR$, using R parameters

$$\{\omega_1, \ldots, \omega_R\} = \left\{(BR/M)\,\lambda_1, \ldots, (BR/M)\,\lambda_R\right\}. \tag{II.10}$$

The ridge estimators are thus $\hat{\boldsymbol{\eta}}_r = \left(\mathbf{W}^\top\mathbf{W} + \omega_r\,\mathbf{I}_{BR\times BR}\right)^{-1}\mathbf{W}^\top\tilde{\mathbf{y}}$. The optimal ridge parameter r^* is selected by minimizing the residual sum of squares:

$$r^* = \arg\min_r \left\|\tilde{\mathbf{y}} - \mathbf{W}\hat{\boldsymbol{\eta}}_r\right\|^2. \tag{II.11}$$

Phenotype predictions by the stacked regression model are defined as $\hat{\mathbf{y}} = \mathbf{W}\hat{\boldsymbol{\eta}}_{r^*}$. Notably, these two levels of ridge regression are implemented within a k-fold cross-validation framework. The predictions for the kth fold $\hat{\mathbf{y}}_k$ are aggregated, where

$$\hat{\mathbf{y}}_k := \mathbf{W}_k\,\hat{\boldsymbol{\eta}}_{(k-1,r^*)}, \tag{II.12}$$

$$\hat{\boldsymbol{\eta}}_{(k-1,r)} := \left(\mathbf{W}^\top_{k-1}\mathbf{W}_{k-1} + \omega_r\,\mathbf{I}_{BR\times BR}\right)^{-1}\mathbf{W}^\top_{k-1}\tilde{\mathbf{y}}_{k-1}, \tag{II.13}$$

$$r^* = \arg\min_r \sum_{k=1}^{K}\left\|\tilde{\mathbf{y}}_k - \mathbf{W}_k\hat{\boldsymbol{\eta}}_{(k-1,r)}\right\|^2. \tag{II.14}$$

The global predictor $\hat{\mathbf{y}} := \sum_{k=1}^{K} \hat{\mathbf{y}}_k$ facilitates the calculation of the associated χ^2 statistic with one degree of freedom for the variant being tested:

$$\chi^2 = \frac{\left(\tilde{\mathbf{x}}_{\text{test}}^\top (\tilde{\mathbf{y}} - \hat{\mathbf{y}})\right)^2}{\hat{\sigma}_e^2 \left(\tilde{\mathbf{x}}_{\text{test}}^\top \tilde{\mathbf{x}}_{\text{test}}\right)}, \quad \hat{\sigma}_e^2 := \frac{\left\|\tilde{\mathbf{y}} - \hat{\mathbf{y}}\right\|_2^2}{(N-C)}. \tag{II.15}$$

The SNPs that have a χ^2 value of above a significant threshold are taken to be associated with the phenotype. The exact threshold depends on the study [167], with a conventional threshold being a p-value of 5×10^{-8}.

II.4.3 Randomized Encoding

Randomized Encoding is central to our approach for computing a function's outcome while masking its underlying inputs. Formally, given a function

$$f : \mathscr{X} \to \mathscr{Y}, \tag{II.16}$$

a randomized encoding of f is defined by two components:

- A randomized function $\hat{f}\colon \mathscr{X} \times \mathscr{R} \to \hat{\mathscr{Y}}$ where $\mathscr{R}$ represents the randomness space.
- A deterministic decoder $\text{Dec}\colon \hat{\mathscr{Y}} \to \mathscr{Y}$.

A randomized encoding of f then satisfies:

$$\text{Dec}\big(\hat{f}(x;r)\big) = f(x) \tag{II.17}$$

with high probability, yet $\hat{f}(x;r)$ reveals no more information about x than $f(x)$ does. In other words, $\hat{f}$ injects structured noise r that conceals the input x, while still allowing a valid output $f(x)$ to be recovered by the decoder. Specific instances of this concept can preserve additional relationships (such as dot products) if required by tasks. Having introduced RE, we now describe the overall PP-GWAS protocol, beginning with a distributed quality control step that leverages an addition based randomized encoding scheme.

II.4.4 Quality Control

In our protocol, the initial stage involves rigorous quality control (QC) checks on the genetic data. This is crucial to ensure the data's integrity and reliability, which are foundational for the accuracy of any subsequent analyses. We adhere to stringent criteria for these checks: a missing rate below 0.1, a minor allele frequency (MAF) above 0.05, and a Hardy-Weinberg equilibrium (HWE) chi-squared test statistic threshold of 23.928. These thresholds are aligned with established GWAS standards, allowing us to filter Single Nucleotide Polymorphisms (SNPs) effectively. Consistent with existing policies, for instance by the National Institutes for Health (NIH) [168], our process includes sharing the total counts of reference homozygous, heterozygous, and alternate homozygous alleles for each SNP with each

participating node, a practice also mirrored in SF-GWAS. To preserve data confidentiality during the QC phase (Figure II.8 A-B) in our distributed environment, since we only need to sum the total counts amongst all nodes, we implement simple addition based randomized encoding in a server-assisted manner. To compute the sum $f(x) = \sum_{i=1}^{P} x_i$, party i in possession of r_i and $\sum_i^P r_i$ generated using the shared seed, sends to the server $\hat{f}(x_i; r_i) = x_i + r_i$ which the server sums as $\hat{f}(x; r) = \sum_{i=1}^{P} \hat{f}(x_i; r_i)$ and returns to all the nodes. They then remove $\sum_i^P r_i$ to obtain

$$\mathrm{Dec}\big(\hat{f}(x,r)\big) = \sum_{i=1}^{P} \hat{f}(x_i, r_i) - \sum_{i=1}^{P} r_i = \sum_{i=1}^{P} x_i. \tag{II.18}$$

PP-GWAS does not necessitate traditional joint genotyping with centralized data [58]. For common-variant single-SNP association on well-imputed or high-coverage datasets, modern variant-calling and imputation pipelines achieve high accuracy limiting the benefits of joint genotyping [169, 170, 171]. Second, our globally performed QC retains variants present in any participating site, so rare variants that might otherwise be discarded by site-specific QC are preserved. This realizes a principal benefit of joint genotyping where rare SNPs absent at a cohort will be "rescued" [169]. When using our method in collaborative settings, in the absence of joint genotyping, data harmonization is required to identify a common set of SNPs across sites. This can be achieved by sharing only non-private information such as genomic positions, reference, alternate alleles, strand information and when available, the rsID, so an aggregator can build a common SNP list without exposing individual level data. These steps, together with covariate projection mitigate technical artefacts, but do not eliminate all effects of technical heterogeneity. We note that a privacy-preserving distributed joint-genotyping layer could further reduce such heterogeneity without centralizing raw data and is complementary to PP-GWAS, but outside the scope of this work.

II.4.5 Distributed Projection of Covariates and Standardizing

In our framework, the genomic information $\mathbf{X}$, covariate information $\mathbf{Z}$, and phenotype information y are horizontally partitioned across P computational nodes, with each node p holding $\mathbf{X}_p$, $\mathbf{Z}_p$, and $\mathbf{y}_p$. Each node maintains a count of the total number of samples added to the study prior to their inclusion and the overall sample count. This information is conveyed through a sequential onboarding process. At the outset of the study, all the nodes establish a shared secret key using established cryptographic techniques, k_{seed}, unknown to the server. This secret key serves as the seed for generating subsequent shared keys. We denote that we have N samples in total, M SNPs, C covariates, and B blocks, which can be inferred by the server.

Subsequently, we standardize the genomic matrix $\mathbf{X}$ and phenotype matrix $\mathbf{y}$, and project out covariate information $\mathbf{Z}$ in the same computation (Figure II.8 C). We do this by appending $\mathbf{Z}$ with a column of ones to mean-center $\mathbf{X}$ and $\mathbf{y}$. We shall denote the updated covariate matrix as $\mathbf{Z}_1$. We also pre-compute the standard deviation matrix $\mathbf{S_X}$ of $\mathbf{X}$ and the standard

deviation $s_{\mathbf{y}}$ of phenotype information $\mathbf{y}$ using the same addition-based randomized encoding approach as before, since we only need to sum up relevant allele counts from each node. We can then project out covariates in a single computation since we know that

$$\tilde{\mathbf{X}} := \left(\mathbf{I}_N - \mathbf{Z}(\mathbf{Z}^\top\mathbf{Z})^{-1}\mathbf{Z}^\top\right)\mathbf{X}_S = \left(\mathbf{I}_N - \mathbf{Z}_1(\mathbf{Z}_1^\top\mathbf{Z}_1)^{-1}\mathbf{Z}_1^\top\right)\mathbf{X}\mathbf{S}_{\mathbf{X}}. \tag{II.19}$$

Here $\mathbf{X}_S$ denotes $\mathbf{X}$ after standardization. We do this since covariate projection inherently corrects for site-level batch effects by adjusting for technical covariates in the model. In settings where cohorts differ by sequencing platform, or variant-calling pipeline, each node can encode platform, pipeline, batch indicators as covariates to correct for potential artefacts as is the standard across various studies [59, 158, 159, 160, 161, 162, 163].

To perform Equation (II.19) in a distributed and privacy-preserving manner, we treat the computation as a randomized encoding task, i.e, $f(\mathbf{Z},\mathbf{X}) = \tilde{\mathbf{X}}$. We adopt methods based on randomized projection from [146, 172, 146, 56, 1], where we achieve data obfuscation as described below. We first construct rectangular matrices $O_{\mathbf{X}}, O_{\mathbf{y}}$, $O_{\mathbf{Z}}$ and $O_{\mathbf{Z}'}$ that satisfy $\mathbb{E}[O_{\mathbf{X}}^\dagger O_{\mathbf{X}}] = \mathbb{E}[O_{\mathbf{y}}^\dagger O_{\mathbf{y}}] = \mathbb{E}[O_{\mathbf{Z}}^\dagger O_{\mathbf{Z}}] = \mathbb{E}[O_{\mathbf{Z}'}^\dagger O_{\mathbf{Z}'}] = \mathbf{I}$.. Each node p prepares encoded data in the form of $O_{\mathbf{Z}}\mathbf{Z}_p O_{\mathbf{Z}'}^\dagger$, $O_{\mathbf{Z}}\mathbf{X}_p O_{\mathbf{X}}^\dagger$, and $O_{\mathbf{Z}}[\mathbf{y}_p, \mathbf{M}_{\mathbf{y}}]\,\rho\, O_{\mathbf{y}}^\dagger$ and sends them to the server. Here $M_{\mathbf{y}}$ is a random matrix with N rows, and ρ a permutation matrix. We note that all the random matrices here are prepared with the help of the shared seed k_{seed}. The server then computes for each node,

$$O_{\mathbf{Z}}\tilde{\mathbf{X}}_p\,\mathbf{S}_{\tilde{\mathbf{X}}}^{-1}O_{\tilde{\mathbf{X}}}^\dagger = O_{\mathbf{Z}}\mathbf{X}_p O_{\mathbf{X}}^\dagger - O_{\mathbf{Z}}\mathbf{Z}_p O_{\mathbf{Z}'}^\dagger\left(\mathbf{T}^\dagger\mathbf{T}\right)^{-1}\mathbf{T}^\dagger\sum_{p=1}^{P}\left(O_{\mathbf{Z}}\mathbf{X}_p O_{\mathbf{X}}^\dagger\right), \tag{II.20}$$

$$\mathbf{T} := O_{\mathbf{Z}}\mathbf{Z}O_{\mathbf{Z}'} = \sum_{p=1}^{P}\left(O_{\mathbf{Z}}\mathbf{Z}_p O_{\mathbf{Z}'}^\dagger\right), \tag{II.21}$$

and sends these to the appropriate nodes. The nodes can then compute $\mathbb{E}[\tilde{\mathbf{X}}_p] = O_{\mathbf{Z}}^\dagger\left(O_{\mathbf{Z}}\tilde{\mathbf{X}}_p\mathbf{S}_{\tilde{\mathbf{X}}}^{-1}O_{\tilde{\mathbf{X}}}^\dagger\right)O_{\tilde{\mathbf{X}}}\mathbf{S}_{\mathbf{X}}$. Analogously, the nodes compute $\mathbb{E}[[\tilde{\mathbf{y}}_p, \mathbf{M}_{\mathbf{y}}]\rho]$ and retrieve $\mathbb{E}[\tilde{\mathbf{y}}_p]$ by undoing the permutation. Hence, we have estimated our computation $f(\mathbf{Z},\mathbf{X})$ with $f(\mathbf{Z},\mathbf{X})$ with $\hat{f}(\mathbf{Z},\mathbf{X};\, O_{\mathbf{Z}}, O_{\mathbf{X}})$ using $O_{\mathbf{Z}}$ and $O_{\mathbf{X}}$ that act as structured noise. Similarly, we have computed $f(\mathbf{Z},\mathbf{y})$ with $\hat{f}(\mathbf{Z},\mathbf{y};\, O_{\mathbf{Z}}, O_{\mathbf{y}}, \mathbf{M}_{\mathbf{y}}, \rho)$.

II.4.6 Level 0 Ridge Regression using distributed ADMM

Next, we would like to perform the first level of ridge regression on the genotypes against the phenotypes, using R parameters $(\lambda_1, \ldots, \lambda_R)$ given by Equation (II.5) (Figure II.8 D). We now estimate $\hat{\boldsymbol{\beta}}_{\lambda_r}^b$ for all blocks b from Equation (II.6). For this purpose, we adopt the distributed Alternate Direction Method of Multipliers [173] to jointly estimate the level 0 predictions. Note that on a centralized dataset, the ridge regression problem can be formulated as the following optimization problem for a given ridge parameter λ_r: $\hat{\boldsymbol{\beta}}_{\lambda_r}^b = \arg\min_{\boldsymbol{\beta}}\left(\left\|\tilde{\mathbf{X}}^b\boldsymbol{\beta} - \tilde{\mathbf{y}}\right\|_2^2 + \lambda_r\left\|\boldsymbol{\beta}\right\|_2^2\right)$. We introduce a variable $\mathfrak{b}$ to rewrite the equation as a constraint problem below.

$$\hat{\boldsymbol{\beta}}_{\lambda_r}^b = \arg\min_{\boldsymbol{\beta},\mathbf{b}}\left(\left\|\tilde{\mathbf{X}}^b\boldsymbol{\beta} - \tilde{\mathbf{y}}\right\|_2^2 + \lambda_r\left\|\mathbf{b}\right\|_2^2\right), \quad \text{s.t. } \boldsymbol{\beta} - \mathbf{b} = 0. \tag{II.22}$$

Since the data in our setting is horizontally partitioned, we can rewrite Equation (II.22) as follows, where we also horizontally partition β.

$$\hat{\boldsymbol{\beta}}^{b}_{\lambda_r} = \arg\min_{\{\boldsymbol{\beta}_p\},\mathbf{b}} \left(\sum_{p=1}^{P} \left\| \tilde{\mathbf{X}}^{b}_{p} \boldsymbol{\beta}_p - \tilde{\mathbf{y}} \right\|_2^2 + \lambda_r \left\| \mathbf{b} \right\|_2^2 \right), \quad \boldsymbol{\beta}_p - \mathbf{b} = 0 \ \ \forall p. \tag{II.23}$$

We detail our distributed approach to use randomized encoding to compute Equation (II.23) in Box II.1 below. The computational nodes use their shared seed to consistently segregate their data into B blocks. They also then use the seed to determine how they split their data vertically into K folds, such that every node has some data in every fold. They then denote the kth fold as $\tilde{\mathbf{X}}^{b}_{(p,k)}$ and the data without the kth fold as $\tilde{\mathbf{X}}^{b}_{(p,k-1)}$. Similarly, they have $\tilde{\mathbf{y}}_{(p,k)}$ and $\tilde{\mathbf{y}}_{(p,k-1)}$.

In this distributed ADMM framework, each computational node independently updates its local estimate β_p by minimizing its respective objective, while a central variable $\mathfrak{b}$ is iteratively updated to enforce consensus among the nodes. The method involves alternating updates of the local variables and dual variables, ensuring that the global constraint $\beta_p - \mathfrak{b} = 0$ is satisfied as the algorithm converges. In the algorithm, the local ADMM updates $\mathscr{X}_p^{(i)}$ correspond to the variables β_p from Equation (II.23), and the consensus variable $\mathfrak{b}$ is represented by $\mathscr{Z}^{(i)}$.

II.4.7 Level 1 Ridge Regression using CGD

Now, like before in the centralized formulation as in Equation (II.4.2), we have reduced our problem to lower dimensionality. We then perform Conjugate Gradient descent (Figure II.8 E), however on the server's side on the obfuscated data, as described in Box II.2 below. For this, the server prepares R ridge regression parameters $(\omega_1, \ldots, \omega_R)$ given by Equation (II.10). In this CGD framework, the variable $\mathscr{X}^{(i)}$ represents the current estimate of the lower-dimensional solution (analogous to the parameter vector in Equation (II.4.2)), while $\mathscr{Z}^{(i)}$ and $\mathscr{Y}^{(i)}$ correspond to the residual and conjugate direction vectors, respectively. These mappings ensure that the iterative updates converge to the optimal ridge regression solution on the obfuscated data.

II.4.8 Distributed Single SNP Association Testing

For the next stage of the analyses, the nodes engage in a one-off communication with the server, helping them retrieve the χ^2 values associated with each SNP (Figure II.8 F). This is outlined in Box II.3 below. Note that the server sees the final χ^2 values, but has no direct access to underlying genotype or phenotype data. Furthermore, in case this also needs to be hidden, one can shuffle the ordering of SNPs in the study preventing the server from linking specific χ^2 statistics to identifiable SNP positions. The computational nodes can apply thresholds using standard criteria on these p-values locally.

Box II.1 Distributed ADMM Algorithm for Level 0 Ridge Regression

Input: Each node p in $\{1,\dots,P\}$ knows matrices $\tilde{\mathbf{X}}^b_{(p,k)}$, $\tilde{\mathbf{X}}^b_{(p,k-1)}$, column vector $\tilde{\mathbf{y}}_{(p,k-1)}$, learning rate ℓ, and the number of ridge regression parameters R. The server requires learning rate ℓ, ridge regression parameters $\{\lambda_1,\dots,\lambda_R\}$, and number of iterations n.
Output: Obfuscated Level 0 predictions for block b, regression parameter λ_r, and fold k given by $O^{(k,r)}_{\tilde{\mathbf{y}}}\hat{\mathbf{y}}^{(b,r)}_k$.

1. Each node, using a shared seed, prepares a non-zero constant $k_{\tilde{\mathbf{y}}}$, rectangular matrices $O^{(k,r)}_{\tilde{\mathbf{y}}}$, $O^{(k,b,r)}_{\tilde{\mathbf{X}}}$ for all k,b,r, ensuring $\mathbb{E}\big[(O^{(k,r)}_{\tilde{\mathbf{y}}})^\dagger O^{(k,r)}_{\tilde{\mathbf{y}}}\big] = \mathbf{I}_N$, and $\mathbb{E}\big[(O^{(k,b,r)}_{\tilde{\mathbf{X}}})^\dagger O^{(k,b,r)}_{\tilde{\mathbf{X}}}\big] = \mathbf{I}_{N_b}$..
2. For each combination of k,b,r:
   (a) Server initializes $\mathscr{X}^{(0)}, \mathscr{Y}^{(0)}, \mathscr{Z}^{(0)}$ to 0.
   (b) Nodes compute and share with the server:
   - $\mathscr{R}^{(p,k,b,r)} := O^{(k,b,r)}_{\tilde{\mathbf{X}}}\Big[(\tilde{\mathbf{X}}^b_{(p,k-1)})^\top \tilde{\mathbf{X}}^b_{(p,k-1)} + \ell\,\mathbf{I}_{N_b}\Big](O^{(k,b,r)}_{\tilde{\mathbf{X}}})^\dagger$,
   - $\mathscr{X}^{(p,k,b,r)} := \frac{1}{k_{\tilde{\mathbf{y}}}}(O^{(k,r)}_{\tilde{\mathbf{y}}})^\dagger \tilde{\mathbf{X}}^b_{(p,k)}\, O^{(k,b,r)}_{\tilde{\mathbf{X}}}$,
   - $\mathscr{Y}^{(p,k,r)} := O^{(k,r)}_{\tilde{\mathbf{y}}}\tilde{\mathbf{y}}_{(k)}$,

   (c) Server computes:
   - $\mathscr{X}^{(1)}_p = \mathscr{R}^{(p,k,b,r)}\Big(\sum_{\hat{k}\neq k}\mathscr{X}^{(p,\hat{k},b,r)}\Big)^\dagger \mathscr{Y}^{(p,k,r)}$,
   - $\mathscr{X}^{(1)} = \sum_p \mathscr{X}^{(1)}_p / P$,
   - $\mathscr{Z}^{(1)} = \ell\,\mathscr{X}^{(1)}/\lambda_r$,
   - $\mathscr{Y}^{(1)}_p = \ell\Big(\mathscr{X}^{(1)}_p - \mathscr{Z}^{(1)}\Big)$,
   - $\mathscr{Y}^{(1)} = \sum_p \mathscr{Y}^{(1)}_p$.

   (d) For i in $\{1,\dots,n-1\}$:
   - Server updates as follows:
     – $$\mathscr{X}^{(i+1)}_p = \mathscr{R}^{(p,k,b,r)}\Big(\Big(\sum_{\hat{k}\neq k}\mathscr{X}^{(p,\hat{k},b,r)}\Big)^\dagger\Big(\sum_{\hat{k}\neq k}\mathscr{Y}^{(p,\hat{k},r)}\Big) + \ell\,\mathscr{Z}^{(i)} - \mathscr{Y}^{(i)}_p\Big),$$
     – $\mathscr{X}^{(i+1)} = \sum_p \mathscr{X}^{(i+1)}_p / P$,
     – $\mathscr{Z}^{(i+1)} = \ell\,\mathscr{X}^{(i+1)} + \mathscr{Y}^{(i)}/\lambda_r$,
     – $\mathscr{Y}^{(i+1)}_p = \mathscr{Y}^{(i)} + \ell(\mathscr{X}^{(i+1)}_p - \mathscr{Z}^{(i+1)})$,
     – $\mathscr{Y}^{(i+1)} = \sum_p \mathscr{Y}^{(i+1)}_p$.

   (e) Server computes $\mathscr{X}^{(p,k,b,r)}\,\mathscr{Z}^{(n)} = O^{(k,r)}_{\tilde{\mathbf{y}}}\hat{\mathbf{y}}^{(b,r)}_k$.

Return: $O^{(k,r)}_{\tilde{\mathbf{y}}}\hat{\mathbf{y}}^{(b,r)}_k$.

Box II.2 CGD Algorithm for Level 1 Ridge Regression

Input: The server requires a number of iterations n, ridge regression parameters $\omega_1, \ldots, \omega_R$, pre-received $\mathscr{Y}^{(p,k,r)}$ from the level 0 computation, and the precomputed $O_{\tilde{\mathbf{y}}}^{(k,r)} \hat{\mathbf{y}}_k^{(b,r)}$ for all k, b, r.
Output: Obfuscated Level 1 predictions for regression parameter ω_r and fold k given by $O_{\tilde{\mathbf{y}}}^{(k,r)} [\hat{\mathbf{y}}_k^{(1,r)}, \ldots, \hat{\mathbf{y}}_k^{(B,r)}] \hat{\boldsymbol{\eta}}_{(k-1,r)}$.

1. For each combination of k, r:
   (a) Server initializes $\mathscr{X}^{(0)} = 0$.
   (b) Server initializes $\mathscr{Y}^{(0)}$, $\mathscr{Z}^{(0)} = \sum_{\hat{k} \neq k} \left(\left(O_{\tilde{\mathbf{y}}}^{(\hat{k},r)} \hat{\mathbf{y}}_{\hat{k}}^{(b,r)} \right)^\dagger \left(\sum_{p=1}^{P} \mathscr{Y}^{(p,\hat{k},r)} \right) \right)$..
   (c) Server precomputes $\mathscr{W} = \sum_{\hat{k} \neq k} \left(O_{\tilde{\mathbf{y}}}^{(\hat{k},r)} \hat{\mathbf{y}}_{\hat{k}}^{(b,r)} \right)^\dagger \left(O_{\tilde{\mathbf{y}}}^{(\hat{k},r)} \hat{\mathbf{y}}_{\hat{k}}^{(b,r)} \right)$.
   (d) For i in $0, \ldots, n-1$:
      - $\alpha = \mathscr{W} \mathscr{Y}^{(i)}$,,
      - $\alpha + \omega_r \mathscr{Y}^{(i)}$,
      - $\gamma = (\mathscr{Z}^{(i)})^\dagger \mathscr{Z}^{(i)} / (\mathscr{Y}^{(i)})^\dagger \alpha$,
      - $\mathscr{X}^{(i+1)} = \mathscr{X}^{(i)} + \gamma \mathscr{Y}^{(i)}$,
      - $\mathscr{Z}^{(i+1)} = \mathscr{Z}^{(i)} - \gamma \alpha$,
      - $\delta = (\mathscr{Z}^{(i+1)})^\dagger \mathscr{Z}^{(i+1)} / (\mathscr{Z}^{(i)})^\dagger \mathscr{Z}^{(i)}$,
      - $\mathscr{Y}^{(i+1)} = \mathscr{Z}^{(i+1)} + \delta \mathscr{Y}^{(i)}$.
   (e) Server computes
   $r^* := \sum_k \arg\min_r \| \sum_p \bar{\mathscr{Y}}^{(p,k,r)} - O_{\tilde{\mathbf{y}}}^{(k,r)} [\hat{\mathbf{y}}_k^{(1,r)}, \ldots, \hat{\mathbf{y}}_k^{(B,r)}] \hat{\boldsymbol{\eta}}_{(k-1,r)} \|_2^2$.

Return:
$O_{\tilde{\mathbf{y}}}^{(k,r^*)} [\hat{\mathbf{y}}_k^{(1,r^*)}, \ldots, \hat{\mathbf{y}}_k^{(B,r^*)}] \mathscr{X}^{(n)} = O_{\tilde{\mathbf{y}}}^{(k,r^*)} [\hat{\mathbf{y}}_k^{(1,r^*)}, \ldots, \hat{\mathbf{y}}_k^{(B,r^*)}] \hat{\boldsymbol{\eta}}_{(k-1,r^*)}$.

Box II.3 Distributed Association Testing

Input: The server requires pre-received $\bar{\mathscr{Y}}^{(p,k,r)}$ from the level 0 computation and the precomputed $\mathscr{K}_k := O_{\tilde{\mathbf{y}}}^{(k,r^*)}[\hat{\mathbf{y}}_k^{(1,r^*)},\ldots,\hat{\mathbf{y}}_k^{(B,r^*)}]\,\hat{\boldsymbol{\eta}}_{(k-1,r^*)}$ from the level 1 computation for all k.
Output: χ^2 value associated to SNP $\tilde{\mathbf{x}}_{\text{test}}$.

1. Nodes compute and share $\left(O_{\tilde{\mathbf{y}}}^{(k,r^*)}\right)^{\dagger}\tilde{\mathbf{x}}_{(\text{test},k,p)}$ for all k.
2. Server computes

$$\chi^2_{\tilde{\mathbf{x}}_{\text{test}}} = \frac{\left[\sum_{k=1}^{K}\left(\sum_{p=1}^{P}\left(O_{\tilde{\mathbf{y}}}^{(k,r^*)}\right)^{\dagger}\tilde{\mathbf{x}}_{(\text{test},k,p)}\right)^{\dagger}\left(\sum_{p=1}^{P}\mathscr{Y}^{(p,k,r^*)}-\mathscr{K}_k\right)\right]^2}{\hat{\sigma}^2\sum_{k=1}^{K}\sum_{p=1}^{P}\left(\left(O_{\tilde{\mathbf{y}}}^{(k,r^*)}\right)^{\dagger}\tilde{\mathbf{x}}_{(\text{test},k,p)}\right)^{\dagger}\left(\left(O_{\tilde{\mathbf{y}}}^{(k,r^*)}\right)^{\dagger}\tilde{\mathbf{x}}_{(\text{test},k,p)}\right)}, \tag{II.24}$$

where

$$\hat{\sigma}^2 := \frac{1}{N-C}\sum_{k=1}^{K}\Big\|\sum_{p=1}^{P}\mathscr{Y}^{(p,k,r^*)}-\mathscr{K}_k\Big\|_2^2. \tag{II.25}$$

Return: $\chi^2_{\tilde{\mathbf{x}}_{\text{test}}}$.

II.4.9 Privacy Analysis

We now describe the privacy guarantees of our algorithm within an adversarial framework, comprising a subset of semi-honest computational nodes and/or a semi-honest non-colluding central server. We show that a corrupted participant is unable to extract any information about the data of other non-corrupt nodes, and similarly, a corrupted server is incapable of deducing any node-specific information. It is important to clarify that our analysis does not cover extreme data scenarios, that automatically enable the prediction of block sizes. Our proof methodology aligns with the approaches documented in prior works [146, 56, 172, 55, 1].

Theorem 4. PP-GWAS is secure against a semi-honest adversary who corrupts the central server.

Proof. We define a semi-honest central server to be a third-party server that adheres to the prescribed protocol, but attempts to learn the private data. In PP-GWAS, the server receives encoded data

$$O_{\mathbf{Z}}\mathbf{Z}_p O_{\mathbf{Z}'}^{\top} \in \mathbb{C}^{(N+k_{\mathbf{Z}})\times(C+k_{\mathbf{Z}'})}, \quad O_{\mathbf{Z}}\mathbf{X}_p O_{\mathbf{X}}^{\dagger} \in \mathbb{C}^{(N+k_{\mathbf{Z}})\times(N+k_{\mathbf{X}})}, \tag{II.26}$$

$$O_{\mathbf{Z}}\,[\mathbf{y}_p,\mathbf{M}_{\mathbf{y}}]\,\rho\,O_{\mathbf{y}} \in \mathbb{C}^{(N+k_{\mathbf{Z}})\times(1+k_{M_{\mathbf{y}}}+k_{\mathbf{y}})}, \quad \left(O_{\tilde{\mathbf{y}}}^{(k,r^*)}\right)^{\dagger}\tilde{\mathbf{x}}_{(\text{test},k,p)} \in \mathbb{C}^{(N+k_{\tilde{\mathbf{y}}})\times 1}, \tag{II.27}$$

from each input node where N is the number of samples and C is the number of covariates. The data the central server then has access to includes

$$O_{\mathbf{Z}}\mathbf{Z}_p O_{\mathbf{Z}'}^{\top}, \quad O_{\mathbf{Z}}\mathbf{X}_p O_{\mathbf{X}}^{\dagger}, \tag{II.28}$$

$$O_{\mathbf{Z}}\left[\mathbf{y}_p, \mathbf{M}_{\mathbf{y}}\right] O_{\mathbf{y}}, \quad \left(O_{\tilde{\mathbf{y}}}^{(k,r^*)}\right)^{\dagger} \tilde{\mathbf{x}}_{(\text{test},k,p)}. \tag{II.29}$$

It is evident that the block sizes are hidden from the central server. The first three quantities are obfuscated on both sides and provide sufficient privacy [172, 146, 56, 1]. Now we show that $\left(O_{\tilde{\mathbf{y}}}^{(k,r^*)}\right)^{\dagger} \tilde{\mathbf{x}}_{(\text{test},k,p)}$ is not produced by a unique pair $\left(O_{\tilde{\mathbf{y}}}^{(k,r^*)}\right)^{\dagger}$ and $\tilde{\mathbf{x}}_{(\text{test},k,p)}$. For simplicity, we denote the quantities as $O_{\tilde{\mathbf{x}}}$ and $\tilde{\mathbf{x}}$. Given an orthogonal matrix $U \in \mathbb{R}^{N\times N}$ with $U\mathbf{1} = \mathbf{1}$, $\check{\mathbf{x}}_p = U\tilde{\mathbf{x}}_p$ and $\check{O}_{\tilde{\mathbf{x}}} = O_{\tilde{\mathbf{x}}}U^{\top}$, we have $O_{\tilde{\mathbf{x}}}\tilde{\mathbf{x}}_p = \check{O}_{\tilde{\mathbf{x}}}\check{\mathbf{x}}_p$. Further, since $\tilde{\mathbf{x}}$ is standardized, so is $\check{\mathbf{x}}$, and hence the structure of $\tilde{\mathbf{x}}$ provides no additional information for the server.

Theorem 5. PP-GWAS is secure against a proper subset of semi-honest nodes.

Proof. We define a proper subset of corrupt nodes as any subset excluding at least one honest node. We assume corrupt nodes are semi-honest, meaning they follow the protocol but may attempt to learn additional information from the accessible data. Each node p, only receives the relevant p'th partitions, such as $\mathbb{E}[\tilde{\mathbf{X}}_p]$ and $\mathbb{E}[\tilde{\mathbf{y}}_p]$. Therefore if a proper subset of the corrupt nodes collude, they cannot access or infer information beyond their encoded partitions. Since our adversarial setting considers a non-colluding semi-honest central server, the server will not deviate from the protocol and share information pertaining to non-corrupt nodes with the corrupt nodes.

Therefore, we have shown that the data of non-corrupt nodes remains private, and secure from a proper subset of semi-honest nodes, and/or a non-colluding semi-honest central server. Further, the central server does not at any point of the protocol learn the block sizes utilized.

Data availability

The real-world datasets analysed here are available via controlled access from the NCBI database of Genotypes and Phenotypes (dbGaP). The bladder cancer risk dataset (n=13,060; phs000346.v2.p2 [https://www.ncbi.nlm.nih.gov/projects/gap/cgi-bin/study.cgi?study_id=phs000346.v2.p2]) and the age-related macular degeneration dataset (n=22,683; phs001039.v1.p1 [https://www.ncbi.nlm.nih.gov/projects/gap/cgi-bin/study.cgi?study_id=phs001039.v1.p1]) contain individual-level genomic and phenotypic information collected under informed consent and are therefore available only to qualified researchers under the Data Use Limitations specified in each dbGaP record. Access requests should be submitted through the dbGaP Authorized Access system, citing the accession numbers above and including an institutional Data Use Certification and, where applicable, IRB/ethics approval. Requests are reviewed by the appropriate NIH dbGaP Data Access Committee; the authors are not involved in approval decisions. Further details on the original study protocols, including participant recruitment and sample collection, are provided in the respective

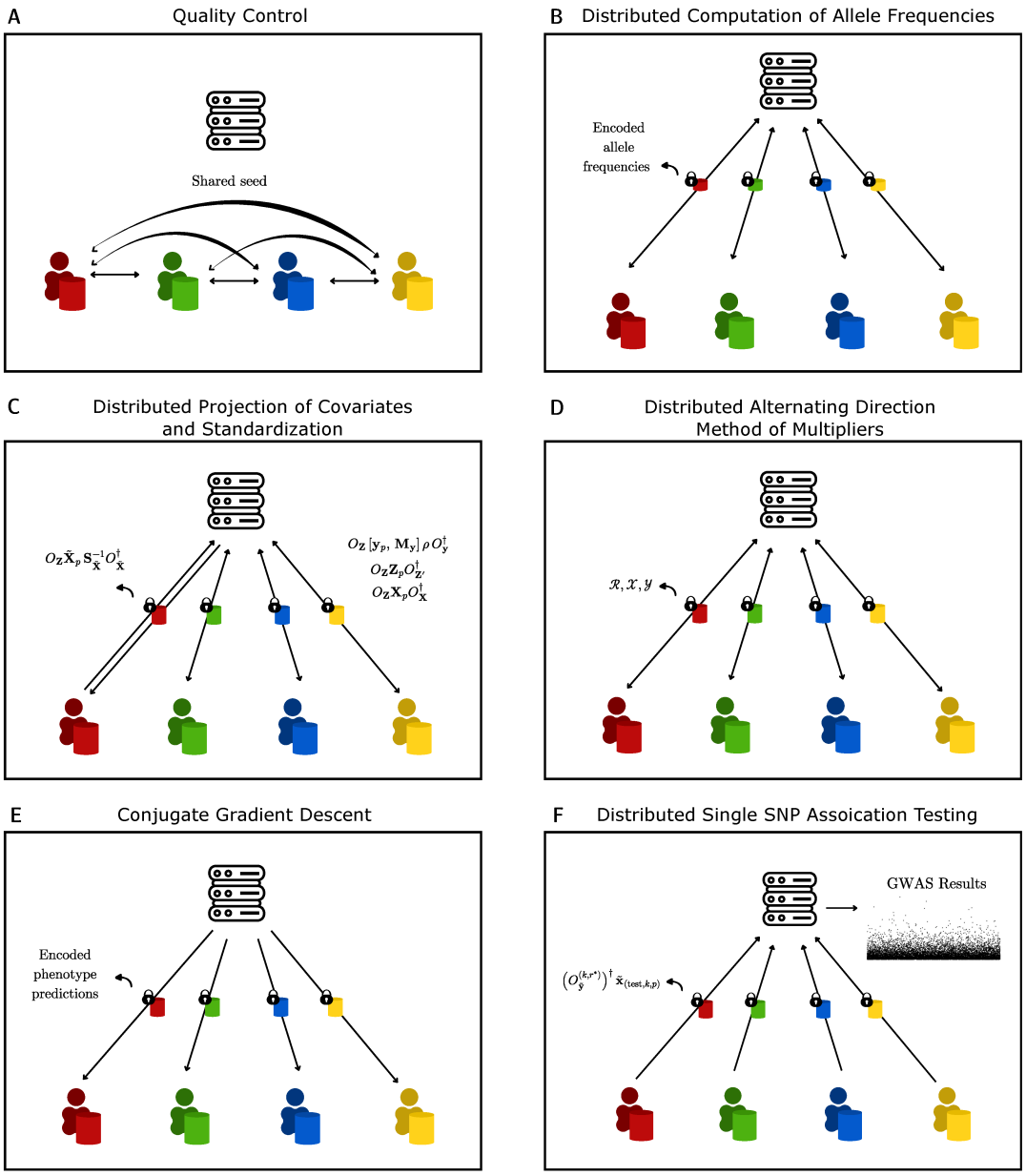


Figure II.8: **Step-by-step illustration of PP-GWAS (A)** Quality control and initialization: a common random seed is generated and securely shared among the computational nodes. **(B)** Allele-frequency estimation: with server coordination, nodes compute allele frequencies as in Eq. II.18. **(C)** Covariate projection: nodes and server remove covariate effects as described in Eq. II.20. **(D)** Level 0 model fitting: nodes transmit the aggregated quantities in Box II.1 to enable distributed ADMM ridge regression. **(E)** Level 1 model fitting: the server performs ridge regression via conjugate gradient descent (CGD) on the ADMM outputs. **(F)** Single-SNP testing: nodes provide the quantities in Box II.3; the server computes, for each SNP, a χ^2 statistic (df = 1) and the corresponding two-sided p-value. At no point does the server access raw genotypes or phenotypes; only obfuscated intermediate values are exchanged.

dbGaP records. Access requests are typically reviewed by the NIH Data Access Committee in about two weeks on average; if approved, dataset access is granted for one year and may be renewed. Synthetic data were generated using pysnptools. Instructions and scripts for generating these synthetic datasets are publicly available in our GitHub repository. No other custom datasets were generated for this study. Source data are provided with this paper for all figures and tables derived from testing on the synthetic data.

Code availability

Our code is available on GitHub at the following URL: https://github.com/mdppml/PP-GWAS [174].

Acknowledgements

This research was supported by the German Federal Ministry of Education and Research (BMBF) (project 01ZZ2010; A.S., M.A., and N.P.) and, in part, by the PrivateAIM project (01ZZ2316D; M.A. and N.P.). We express our gratitude to Prof. Dr. Sven Nahnsen for providing access to the real-world datasets utilized in this study. Our gratitude also goes to Dr. Carl Kadie for their assistance in generating synthetic data. We acknowledge the usage of the Training Center for Machine Learning (TCML) cluster at the University of Tübingen. This work was further supported by the de.NBI Cloud within the German Network for Bioinformatics Infrastructure (de.NBI) and ELIXIR-DE (Forschungszentrum Jülich and W-de.NBI-001, W-de.NBI-004, W-de.NBI-008, W-de.NBI-010, W-de.NBI-013, W-de.NBI-014, W-de.NBI-016, W-de.NBI-022). We also thank Cem Ata Baykara, Larissa Reichart and Lukas Böhm for their help with debugging code errors. We acknowledge support from the Open Access Publication Fund of the University of Tübingen.

II.5 Supplementary Material

To further assess the accuracy of PP-GWAS on simulated data, we report the squared Pearson correlation coefficient (r^2) between the negative logarithm of p-values obtained from PP-GWAS and those computed by REGENIE pertaining to the experiments whose runtimes are illustrated in Figure 3 in the main manuscript. We demonstrate that PP-GWAS can reliably reproduce the results from plaintext computation under diverse settings (Supplementary Table II.1). The minimal discrepancies observed are restricted to a small number of SNPs with very high $-\log_{10}(p)$ values. These deviations are attributed to floating-point arithmetic errors, and do not affect the significance ranking of top associations, as further seen in Figure II.7.

N	M	C	P	r^2
9 178	612 794	10	2	0.999999
9 178	612 794	10	4	0.999999
9 178	612 794	10	6	0.999999
9 178	612 794	10	8	0.999999
9 178	1 225 588	10	2	0.999999
9 178	1 838 382	10	2	1.000000
9 178	2 451 176	10	2	1.000000
9 178	612 794	20	2	0.999999
9 178	612 794	30	2	0.999999
9 178	612 794	40	2	0.999999
18 356	612 794	10	2	0.999999
27 534	612 794	10	2	0.999999
36 712	612 794	10	2	0.999999

$^{\dagger}r^2$ values are rounded to six decimal places (precision 10^{-6}).

Table II.1: Agreement between PP-GWAS and REGENIE on synthetic datasets varying sample size (N), number of SNPs (M), covariates (C), and computational parties (P). Across all SNPs, the correlation of $-\log_{10}(p\text{-values})$ was $r^2 = 0.999999 \sim 1.00$ (df $= M-2$), $p < 10^{-6}$, 95% CI [0.999999, 1.000000]; all scenarios showed near-perfect agreement ($r^2 \approx 1$)†.

III Distributed and Secure Kernel-Based Quantum Machine Learning

Arjhun Swaminathan Mete Akgün

Abstract

Quantum computing promises to revolutionize machine learning, offering significant efficiency gains for tasks such as clustering and distance estimation. Additionally, it provides enhanced security through fundamental principles like the measurement postulate and the no-cloning theorem, enabling secure protocols such as quantum teleportation and quantum key distribution. While advancements in secure quantum machine learning are notable, the development of secure and distributed quantum analogs of kernel-based machine learning techniques remains underexplored.

In this work, we present a novel approach for securely computing three commonly used kernels: the polynomial, radial basis function (RBF), and Laplacian kernels, when data is distributed, using quantum feature maps. Our methodology formalizes a robust framework that leverages quantum teleportation to enable secure and distributed kernel learning. The proposed architecture is validated using IBM's Qiskit Aer Simulator on various public datasets.

III.1 Introduction

Quantum computing is set to revolutionize machine learning (ML) by leveraging its capability to encode high-dimensional data into quantum bits, or qubits. These qubits exist in a superposition of states, enabling quantum data to represent data exponentially more efficiently than classical computing—data represented using N classical bits can equivalently be represented by $log_2 N$ qubits. Further, in addition to superposition, quantum entanglement enables qubits to exhibit strong correlations that persist regardless of spatial separation. These correlations are essential for achieving exponential speedups in specific quantum algorithms [175]. Although practical quantum computers are still in their infancy, various quantum machine learning (QML) techniques have been proposed [176, 177, 178].

Notably, quantum computing has exhibited substantial efficiency gains in some computational tasks compared to classical computing [179, 180]: the estimation of distances and inner products between post-processed N-dimensional vectors is achieved in $\mathcal{O}(log(N))$ as compared to $\mathcal{O}(N)$. Similarly, clustering N-dimensional vectors into M clusters is expedited to $\mathcal{O}(log(MN))$ using quantum data, compared to $\mathcal{O}(poly(MN))$.

Quantum computers are not only efficient at handling high-dimensional data, but are inherently secure. This stems from two fundamental principles of quantum mechanics: the measurement postulate and the no-cloning theorem [181]. Quantum data collapse upon measurement and cannot be copied without destroying the original data, offering absolutely secure communication. Secure quantum computing is well studied and includes protocols such as quantum teleportation [62, 182], quantum key distribution [183, 184], and quantum secure direct communication [185, 186, 187, 83].

Additionally, quantum computers utilizing various technologies, such as trapped ions, photons, superconducting circuits, and so forth, are actively being developed, enhancing the practical implementation of these secure communication protocols.

In parallel, kernel-based ML methods have emerged as effective tools for classification and regression tasks. These methods compute similarities between data points in high-dimensional spaces, making them particularly suited for problems where data is not trivially separable. In contrast to more advanced counterparts, such as deep learning, many kernel-based ML techniques often offer greater interpretability [188, 189] and better accuracy when high-dimensional data is limited, which is often the case in many real-world applications [190, 104]. Although much of the focus in QML has been on developing quantum or hybrid (quantum-classical) deep learning and neural networks [191, 192, 193, 194], quantum analogs of kernel-based ML techniques are important alternatives, with landmark studies focusing on centralized data [195, 61].

However, real-world scenarios often involve distributed data [196, 56], in which multiple parties wish to collaboratively train a model while ensuring the privacy of their datasets. Designing QML techniques that work securely in such distributed settings remains a critical challenge, and current research on distributed kernel-based QML methods is still sparse. In particular, Sheng and Zhou [197] introduced a distributed framework that computes the distance between two data points for classification tasks. Later, Schuld and Killoran [61] demonstrated that this approach is a specific instance of a more general principle: encoding data into an infinite-dimensional Hilbert space as quantum data is equivalent to mapping data into a higher-dimensional feature space for kernel computation. This insight establishes a fundamental link between quantum encoding and kernel feature maps, revealing that Sheng and Zhou [197]'s distance computation is the quantum analog of the linear kernel within a broader theoretical context.

Building on the works of Sheng and Zhou [197] and Schuld and Killoran [61], our research investigates this intriguing relationship between quantum feature maps and kernel functions in the context of distributed kernel computations. Specifically, we identify appropriate quantum encodings for commonly used kernels and explore their applications in distributed learning settings. Our approach encodes classical data into quantum states using Random Fourier Features (RFF) [84] and uses a robust architecture for secure and distributed kernel-based learning. This architecture leverages quantum teleportation to ensure data security and employs a semi-honest server to compute kernels and train ML models.

We used publicly available datasets and Qiskit's Aer Simulator [198] to validate our architecture. Our evaluation demonstrates that the proposed approach ensures data security and achieves performance comparable to centralized classical and quantum methods. However, it is important to acknowledge that achieving identical or superior accuracy to classical methods is not anticipated in our study. This is not only due to widely recognized challenges in quantum computing, such as quantum noise [199], approximations in quantum state preparation, and hardware constraints, but also due to the inherent probabilistic nature of quantum states. We make the following three contributions:

1. **Introduction of Quantum Feature Maps:** We introduce quantum feature maps for the polynomial, Radial Basis Function (RBF), and Laplacian kernels and theoretically prove the correctness of these feature maps.
2. **Architecture for Secure Kernel Computation:** We formalize a secure architecture to compute linear, polynomial, RBF, and Laplacian kernels using quantum encoding in a federated manner on distributed datasets.
3. **Implementation and Validation:** We theoretically validate our architecture, and for empirical validation, we implement it for linear kernels on publicly available datasets using Qiskit's Aer Simulator. Due to the limitations of the simulator and our lack of access to a real quantum computer, we were unable to test other kernels at this stage.

III.2 Background

III.2.1 Kernel-based Machine Learning

In machine learning, one typically works with a dataset $\mathscr{X}$ consisting of data points $\{x_1, x_2, \ldots, x_N\} \in \mathscr{X}$, where the goal is to identify patterns to evaluate previously unseen data. Kernel-based techniques employ a similarity measure, called the kernel function, between two inputs to construct models that effectively capture the underlying properties of the data distribution. This kernel function is often an inner product in a feature space, typically of higher dimensionality, where nonlinear relationships between data points become more apparent. Various kernel functions are used in practice, such as linear, RBF, polynomial, and Laplacian kernels. These functions are designed to accommodate diverse data characteristics, making them suitable for various applications. Beyond their many applications, these methods have a rich theoretical foundation, which we briefly explore below.

Definition 9. *Kernel function [45]*
A *kernel function* K is a map $K : \mathscr{X} \times \mathscr{X} \to \mathbb{C}$ that satisfies $K(x,y) = \langle \phi(x), \phi(y) \rangle$, where $\phi : \mathscr{X} \to \mathscr{H}$ is a map from the input space $\mathscr{X}$ to a Hilbert space $(\mathscr{H}, \langle \cdot, \cdot \rangle)$.

One refers to ϕ as a feature map. Since for any unitary operator $U : \mathscr{H} \to \mathscr{H}$, $\langle \phi(x), \phi(y) \rangle = \langle U\phi(x), U\phi(y) \rangle$, a kernel can be related to many different feature maps. However, kernel theory defines a unique Hilbert space associated with a kernel, called the Reproducing Kernel Hilbert Space (RKHS) as follows.

Definition 10. *Reproducing Kernel Hilbert Space (RKHS) [46]*
Let $\mathscr{X}$ be an input space and $(\mathscr{R}, \langle \cdot, \cdot \rangle)$ the Hilbert space of functions $f : \mathscr{X} \to \mathbb{C}$. Then $\mathscr{R}$ is an RKHS if there exists a function $K : \mathscr{X} \times \mathscr{X} \to \mathbb{C}$ such that for all $x \in \mathscr{X}$ and $f \in \mathscr{R}$, the following holds:

$$f(x) = \langle f, K(x, \cdot) \rangle. \tag{III.1}$$

Alternatively, considering an associated feature map, $\phi : \mathscr{X} \to \mathscr{H}$, then $\mathscr{R}$ is the space of functions $f : \mathscr{X} \to \mathbb{C}$ such that for all $x \in \mathscr{X}$ and $v \in \mathscr{H}$,

$$f(x) = \langle v, \phi(x) \rangle_{\mathscr{H}}. \tag{III.2}$$

Typically, a large family of machine learning problems aims to compute a prediction function $f : \mathscr{X} \to \mathbb{C}$ that takes training or test data and predicts the corresponding label. This is often formulated as the solution to the following optimization problem:

$$\min_{f \in \mathscr{R}} \left(\sum_{j=1}^{n} L(y_j, f(x_j)) + \lambda \Omega(f) \right), \tag{III.3}$$

where L is a loss function, x_j are training data points, y_j the corresponding labels, λ a regularization parameter controlling the trade-off between the loss and the complexity of the function f, and $\Omega(f)$ a general regularization term that penalizes the complexity of f. This prediction function generally lives in an RKHS. The representer theorem [47] then states that the solution to this optimization problem can be formulated as follows.

$$f^*(x) = \sum_{j=1}^{n} \alpha_i K(x, x_j), \tag{III.4}$$

where K is the corresponding kernel in the RKHS. Hence, optimization in an infinite-dimensional space is reduced to a finite-dimensional problem of solving for α_i by computing the kernel values at the training data points.

In summary, kernel functions are fundamental in machine learning, as they enable the transformation of data into higher-dimensional spaces where nontrivial relationships between data can be studied. By leveraging the theoretical framework of RKHS and the representer theorem, kernels facilitate efficient computations for a large class of machine learning models, such as Support Vector Machines (SVM), Gaussian Processes, and Principal Component Analysis (PCA) [103].

Random Fourier Features

Kernel methods often face significant computational challenges, particularly with large datasets. To address this issue, Rahimi and Recht [84] introduced Random Fourier Features (RFF) as an effective approach to estimate kernel functions using finite-dimensional feature maps. RFF enables efficient computation of kernel approximations by leveraging the Fourier transform properties of shift-invariant kernels. One defines RFF as below:

Definition 11. *Random Fourier Features (RFF) [84]*
Given a shift-invariant kernel $k(x-y)$ that is the fourier transform of a probability distribution χ, the corresponding lower dimensional feature map $z : \mathbb{R}^D \to \mathbb{R}^d$ defined by

$$z(x) := \left(\sqrt{2} cos(w_1 x + b_1), \ldots, \sqrt{2} cos(w_d x + b_d) \right), \tag{III.5}$$

with $w_i \sim \chi((w_1, \ldots, w_d))$ and b_i are independent samples from the uniform distribution $U[0, 2\pi]$, satisfies the following inequality for all ϵ:

$$\mathbb{P}\left(|z^T(x) z(y) - k(x,y)| \geq \epsilon \right) \leq 2 \exp\left(\frac{-D\epsilon^2}{4} \right). \tag{III.6}$$

The feature map z is called an RFF.

Quantum Encoding

Quantum encoding techniques are crucial for translating classical data into quantum states. There are various methods to do so, such as basis encoding, angle encoding, amplitude encoding, and Hamiltonian evolution ansatz encoding, each with its own distinct advantages and disadvantages. For example, one such encoding is defined below.

Definition 12. *Amplitude Encoding [200]*
Given classical data $x = (x_1, x_2, \ldots, x_N)^T$, where $N = 2^n$, the amplitude encoding of the data is defined as the quantum state:

$$|\psi(x)\rangle := \sum_{j=1}^{N} \frac{x_j}{\|x\|} |j\rangle, \tag{III.7}$$

where $|j\rangle$ represents the computational basis states of an n-qubit system.

The computational basis here refers to the set of basis states that span the state space of an n-qubit quantum system. These states are represented with $|j\rangle$ where $j \in \{0, 1, \ldots, 2^n - 1\}$, and are expressed as tensor products of individual qubit states $|0\rangle$ and $|1\rangle$. For example, the computational basis in a 2-qubit system consists of states:

$$\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}. \tag{III.8}$$

In the context of our work, we will adopt RFF to determine the quantum encodings necessary for the computation of different kernels.

III.3 Related Work

III.3.1 Quantum Feature Maps

Building on the concept of encoding classical data into quantum states, Schuld and Killoran [61] formalized the idea of a *quantum feature map* by noting that any quantum encoding $x \mapsto |\psi(x)\rangle$ behaves like a feature map and maps to a complex Hilbert space $\mathscr{H}$, hence naturally inducing a kernel. This opens up the possibility of utilizing the rich theory of kernel methods alongside quantum computing. Of particular interest is when two input vectors x and y are embedded in an N-dimensional Hilbert space via amplitude encoding. The resulting inner product of the encodings corresponds to the linear kernel.

$$\langle \psi(x) \rangle \psi(y) = x^T y = k(x, y). \tag{III.9}$$

Beyond discussing the linear kernel, the work introduced additional quantum feature maps that correspond to other well-known kernels. For instance, the *Copies of Quantum States* map given by

$$x = (x_1, \ldots, x_N) \mapsto |\psi(x)\rangle = \left(\sum_{j}^{N} \frac{x_j}{\|x_j\|} |j\rangle \right)^{\otimes d}. \tag{III.10}$$

is associated with the homogeneous polynomial kernel, expressed as

$$k(x, y) = (x^T y)^d, \tag{III.11}$$

under the inner product $\langle \psi(x) \rangle \psi(y)$. Similarly, the work proposed the following feature map -

$$x = (x_1, \ldots, x_N) \mapsto \psi(x) = \frac{1}{\sqrt{N}} \sum_{j=1}^{N} \left(\cos(x_{2j-2}) |2j-2\rangle + \sin(x_{2j-1}) |2j-1\rangle \right), \tag{III.12}$$

which is associated with the cosine kernel, expressed as

$$k(x, y) = \prod_{i=1}^{N} cos(x_i - y_i), \tag{III.13}$$

under the inner product $\langle \psi(x) \rangle \psi(y)$.

As discussed earlier, a large family of ML algorithms optimize the functional in (III.3) to obtain a prediction function. There are two primary approaches to this optimization in the context of QML: the implicit approach and the explicit approach. The implicit approach uses the representer theorem and computes kernels while offloading the remaining tasks to classical computing, as demonstrated by Rebentrost et al. [201], Schuld and Killoran [61], Schuld [202]. The explicit approach uses variational circuits to solve the optimization problem in the infinite-dimensional RKHS, as discussed by Havlíček et al. [195], Schuld and Killoran [61], Cerezo et al. [203]. Our work follows the implicit approach in a distributed setting where quantum states are used for kernel computation, and the modeling is offloaded to classical computing.

III.3.2 Distributed Secure Quantum Machine Learning (DSQML)

To the best of our knowledge, only one study, Sheng and Zhou [197], has implemented kernel-based techniques using quantum computing within a distributed framework. Their work introduced the Distributed Secure Quantum Machine Learning (DSQML) algorithm, which facilitates distance computation using a polarization-based quantum system. The setup is designed to ensure security against potential eavesdropping or interference during the computation, as any disturbance by an adversary can be detected.

The DSQML framework offers two operational modes: Client-Server and Client-Server-Database. In the Client-Server model, a client with basic quantum technology aims to classify a single data point into one of two clusters, A and B by computing its distance from the reference vectors v_A and v_B. The client uses amplitude encoding to quantum encode the data point and employs quantum teleportation to delegate the inner product computation to the server. This can be interpreted as a computation of the linear kernel between the data point and the reference vectors as though in a distributed setting.

In the more complex Client-Server-Database model, the client lacks significant quantum resources and can only perform single-qubit preparation and measurement. In this setup,

multiple databases encode their respective data points using amplitude encoding and transfer them to the server via quantum teleportation. The server utilizes a Fredkin gate with the client-prepared ancilla qubit as a control, performs a Hadamard gate, and returns the ancilla qubit to the client. The client then measures the qubit to extract the inner product, which is computed over multiple repetitions.

Although DSQML essentially computes the linear kernel in a distributed setting using amplitude encoding, it does not explicitly acknowledge or leverage the intrinsic relationship between quantum encoding and kernel methods as proposed later by Schuld and Killoran [61]. Consequently, it overlooked the broader kernel framework that can be leveraged for various supervised and unsupervised machine learning tasks across different types of data, including images, text, and numeric data.

In contrast, our research substantially broadens these initial concepts by facilitating the computation of encoding-induced kernels and other standard kernels such as polynomial, RBF, and Laplacian kernels for data of any dimensionality. This generalization to other widely used kernels is much stronger and is important for future study. Further, our method follows a simple Client-Server model and can easily be adapted by relabeling to a Client-Server-Database model.

III.4 Methods

III.4.1 Quantum Feature Maps

Although Schuld and Killoran [61] pointed out the implicit connection between quantum encoding techniques and feature maps, they devised feature maps only for the linear kernel, the homogeneous polynomial kernel, and the cosine kernel. We extend this by defining the quantum feature maps associated with three widely used kernels in ML: the polynomial, RBF, and Laplacian kernels.

Polynomial Kernel

Given classical data $x = (x_1, x_2, \ldots, x_N)^T$, we define the following quantum feature map:

$$x \mapsto \psi(x) = \bigotimes_{j=1}^{\binom{N+d}{d}} \frac{\sqrt{a}\sqrt{d!}}{\sqrt{k_1!k_2!\ldots k_{N+1}!}} x_1^{k_1} \ldots x_N^{k_N} \sqrt{c}^{k_{N+1}} |j-1\rangle\,, \tag{III.14}$$

where the multi-index $k = (k_1, \ldots, k_{N+1})$ runs over all combinations such that $\sum_{l=1}^{N+1} k_l = d$, and $c = 1 - a\|x\|$ if $d \in \mathbb{N}$, or $c = -1 - a\|x\|$ if d $\in 2\mathbb{N}$.

Theorem 6. The quantum feature map above is a well-defined quantum state.

Proof. To be well defined, we require the map to be normalizable. Consider the map

$x \mapsto \psi(x)$. Then, using the multinomial theorem [45, 85], we have that

$$\begin{aligned}\|\psi(x)\| &= \sum_{\sum_l k_l = d} \left(\frac{\sqrt{a}\sqrt{d!}}{\sqrt{k_1! k_2! \dots k_{N+1}!}} \right)^2 (x_1^{k_1})^2 \dots (x_N^{k_N})^2 c^{k_{N+1}}, \\ &= (a\|x\| + c)^d = 1.\end{aligned}$$

This completes the proof.

Theorem 7. The quantum feature map defined above yields the polynomial kernel [47],

$$K_{poly} = (ax^T y + c)^d, \tag{III.15}$$

under an inner product.

Proof. This follows directly from the multinomial theorem. Let $\phi(x)$ and $\phi(y)$ be two quantum feature maps of classical data x and y, defined as above. Then,

$$\begin{aligned}\langle \phi(x) \rangle \phi(y) &= \sum_{\sum_l k_l = d} \left(\frac{\sqrt{a}\sqrt{d!}}{\sqrt{k_1! k_2! \dots k_{N+1}!}} \right)^2 x_1^{k_1} \dots x_N^{k_N} y_1^{k_1} \dots y_N^{k_N} c^{k_{N+1}}, \\ &= (ax^T y + c)^d, \\ &= K_{poly}(x, y).\end{aligned}$$

This completes the proof.

RBF Kernel

Using RFF, given classical data $x = (x_1, x_2, \dots, x_N)^T$, we define the following quantum feature map:

$$x \mapsto \psi(x) = \frac{1}{\sqrt{D}} \sum_{j=1}^{D} \Big(\cos(w_j^T x) |2j-2\rangle + \sin(w_j^T x) |2j-1\rangle \Big), \tag{III.16}$$

where $\lceil \log_2(2D) \rceil$ determines the number of qubits used and the approximation quality, and w_i are independent samples from the normal distribution $\mathcal{N}(0, \sigma^{-2} I)$.

Theorem 8. The quantum feature map above is a well-defined quantum state.

Proof. To be well defined, we require the map to be normalizable. Consider the map $x \mapsto \psi(x)$. Then, we have that

$$\|\psi(x)\| = \frac{1}{D} \sum_{j=1}^{D} cos^2(w_j^T x) + sin^2(w_j^T x) = 1. \tag{III.17}$$

This completes the proof.

Theorem 9. The quantum feature map defined above yields the RBF kernel [204],

$$K_{RBF}(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right), \tag{III.18}$$

under an inner product.

Proof. Let $\phi(x)$ and $\phi(y)$ be two quantum feature maps of classical data x and y, defined as above. It follows that

$$\begin{aligned}\mathbb{E}[\langle\phi(x)\rangle\,\phi(y)] &= \frac{1}{D}\sum_{j=1}^{D}\mathbb{E}\left[cos(w_j^T x)cos(w_j^T y) + sin(w_j^T x)sin(w_j^T y)\right], \\ &= \frac{1}{D}\sum_{j=1}^{D}\mathbb{E}[\cos(w_j^T(x-y))].\end{aligned} \tag{III.19}$$

Using Euler's formula, this can be rewritten as

$$\mathbb{E}[\langle\phi(x)\rangle\,\phi(y)] = \frac{1}{2D}\sum_{j=1}^{D}\left(\mathbb{E}[\exp(i w_j^T(x-y))] + \mathbb{E}[\exp(-i w_j^T(x-y))]\right). \tag{III.20}$$

Since normal distributions remain closed under linear transformations [205],

$$w_j^T(x-y) = \sum_{k=1}^{N} w_{jk}(x_k - y_k) \sim N\left(0, \frac{1}{\sigma^2}\sum_{k=1}^{N}(x_k-y_k)^2\right) \sim \mathcal{N}\left(0, \frac{1}{\sigma^2}\|x-y\|^2\right).$$

Hence $w_j^T(x-y)$ is a normal distribution. Then (III.20) can be rewritten as

$$\mathbb{E}[\langle\phi(x)\rangle\,\phi(y)] = \frac{1}{2D}\sum_{j=1}^{D}\left(M_{w_j^T(x-y)}(i) + M_{w_j^T(x-y)}(-i)\right),$$

where $M_Z(t) = \mathbb{E}[\exp(tZ)]$ is the moment generating function of a random variable Z. Hence, since the moment generating function of a normal distribution $Z \sim \mathcal{N}(\mu,\gamma^2)$ is given by $M_Z(t) = \exp\left(t\mu + \frac{1}{2}\gamma^2 t^2\right)$, we have

$$\begin{aligned}\mathbb{E}[\langle\phi(x)\rangle\,\phi(y)] &= \frac{1}{2D}\sum_{j=1}^{D}\left[\exp\left(-\frac{1}{2\sigma^2}\|x-y\|^2\right) + \exp\left(-\frac{1}{2\sigma^2}\|x-y\|^2\right)\right], \\ &= \exp\left(-\frac{1}{2\sigma^2}\|x-y\|^2\right) = K_{RBF}(x,y).\end{aligned} \tag{III.21}$$

This completes the proof.

Laplacian Kernel

Using RFF, given classical data $x = (x_1, x_2, \ldots, x_N)^T$, we define the following quantum feature map:

$$x \mapsto \psi(x) = \frac{1}{\sqrt{D}} \sum_{j=1}^{D} \Big(\cos(w_j^T x + \alpha_j) \, |2j-2\rangle + \sin(w_j^T x + \alpha_j) \, |2j-1\rangle \Big), \tag{III.22}$$

where $\lceil \log_2(2D) \rceil$ determines the number of qubits used and the approximation quality, w_j are independent samples from the Cauchy distribution $\mathscr{C}(0, \alpha^{-1} I)$, and α_j are independent samples from the uniform distribution $\mathscr{U}(0, 2\pi)$.

Theorem 10. The quantum feature map above is a well-defined quantum state.

Proof. To be well defined, we require the map to be normalizable. Consider the map $x \mapsto \psi(x)$. Then, we have that

$$\|\psi(x)\| = \frac{1}{D} \sum_{j=1}^{D} cos^2(w_j^T x + \alpha_j) + sin^2(w_j^T x + \alpha_j) = 1. \tag{III.23}$$

This completes the proof.

Theorem 11. The quantum feature map defined above yields the Laplacian kernel [206],

$$K_L(x, y) = \exp\left(-\frac{\|x - y\|_1}{\alpha} \right), \tag{III.24}$$

under an inner product.

Proof. Let $\phi(x)$ and $\phi(y)$ be two quantum feature maps of classical data x and y, defined as above. It follows like in Theorem 9 that

$$\mathbb{E}[\langle \phi(x) \rangle \phi(y)] = \frac{1}{2D} \sum_{j=1}^{D} \Big(\mathbb{E}[\exp(i w_j^T (x - y))] + \mathbb{E}[\exp(-i w_j^T (x - y))] \Big). \tag{III.25}$$

Since Cauchy distributions remain closed under linear transformations [207],

$$w_j^T (x - y) = \sum_{k=1}^{N} w_{jk}(x_k - y_k) \sim \mathscr{C}\left(0, \frac{1}{\alpha} \sum_{k=1}^{N} |x_k - y_k| \right) \sim \mathscr{C}\left(0, \frac{\|x - y\|_1}{\alpha} \right). \tag{III.26}$$

Hence, $w_j^T (x - y)$ is a Cauchy distribution. We rewrite (III.25) as

$$\mathbb{E}[\langle \phi(x) \rangle \phi(y)] = \frac{1}{2D} \sum_{j=1}^{D} \Big(\phi_{w_j^T (x-y)}(1) + \phi_{w_j^T (x-y)}(-1) \Big),$$

where $\phi_Z(t) = \mathbb{E}[\exp(itZ)]$ is the characteristic function of a random variable Z. Hence, since the characteristic function of a Cauchy distribution $Z \sim \mathscr{C}(\mu, \gamma)$ is given by $\phi_Z(t) = \exp\left(it\mu - \gamma|t|\right)$, we have

$$\begin{aligned}\mathbb{E}[\langle\phi(x)\rangle\,\phi(y)] &= \frac{1}{2D}\sum_{j=1}^{D}\left[\exp\left(-\frac{\|x-y\|_1}{\alpha}\right) + \exp\left(-\frac{\|x-y\|_1}{\alpha}\right)\right], \\ &= \exp\left(-\frac{\|x-y\|_1}{\alpha}\right) = K_L(x,y). \end{aligned} \tag{III.27}$$

This completes the proof.

III.4.2 Computational Complexity

Classical Setting

In classical computation, kernel evaluation involves performing pairwise operations on N-dimensional vectors. For example:

- The polynomial kernel requires computing the inner product - $\mathscr{O}(N)$ - and raising the result to the power d - ($\mathscr{O}(d)$) - leading to a total complexity of $\mathscr{O}(N+d)$.
- The RBF kernel requires computing the squared norm of the difference between two vectors - $\mathscr{O}(N)$ - and applying the exponential function - $\mathscr{O}(1)$ - leading to a total complexity of $\mathscr{O}(N)$.
- The Laplacian kernel involves the L_1-norm computation ($\mathscr{O}(N)$) - and applying the exponential function - $\mathscr{O}(1)$ - leading to a total complexity of $\mathscr{O}(N)$.

This indicates that classical methods face challenges when N and d are large.

Quantum Setting

The quantum approach involves two main steps: (1) state preparation and (2) computing inner products in the corresponding Hilbert space.

State Preparation: Preparing the quantum feature map for an N-dimensional vector scales with $\mathscr{O}(N)$. This step is a bottleneck in the quantum pipeline. However, using quantum feature maps offers implicit security guarantees through the no-cloning theorem. Secure classical systems require additional overhead, such as homomorphic encryption or secure multi-party computation, to achieve comparable privacy, which can be computationally more expensive [208].

Inner Product Computation: Once states are prepared, computing the inner product scales logarithmically with the dimension of the computational basis, $\bar{D}$, which determines the number of qubits required. Specifically, the number of qubits used is $\lceil \log_2 \bar{D} \rceil$, and the

computational complexity is therefore $\mathcal{O}(\lceil \log_2 \bar{D} \rceil)$ for one shot. The dimension $\bar{D}$ depends on the kernel being computed:

- For polynomial kernels: $\bar{D} = \binom{N+d}{d}$, as seen in (III.14). Using Stirling's approximation [209], the complexity can be approximated as $\mathcal{O}(d \log N)$ for large N and d. In Appendix III.7.2, we show that to achieve an additive approximation error of ϵ, one must use $M = \mathcal{O}(1/\epsilon^2)$ number of shots, and hence the overall complexity becomes $\mathcal{O}((d \log N) \cdot M) = \mathcal{O}((d \log N)/\epsilon^2)$.
- For RBF and Laplacian kernels: $\bar{D} = 2D$, where D is the number of random Fourier features used to approximate the kernel. The number of qubits required in this case scales as $\lceil \log_2(2D) \rceil$, leading to a complexity of $\mathcal{O}(\lceil \log_2(2D) \rceil)$. In Appendices III.7.2 and III.7.2, we show that to achieve an additive approximation error of ϵ with high probability, one must choose $D = \mathcal{O}(1/\epsilon^2)$ and use $M = \mathcal{O}(1/\epsilon^2)$ shots. Consequently, the overall computational complexity for the estimation of the inner product in these cases becomes $\mathcal{O}(\lceil \log_2(2D) \rceil \cdot M) = \mathcal{O}(\lceil \log_2(2/\epsilon^2) \rceil / \epsilon^2)$.

III.4.3 Distributed Secure Computation of Kernels

Architecture

Our architecture builds on foundational concepts in quantum computing, including quantum teleportation [62] and Fredkin gates (controlled-SWAP) [210], to enable secure and distributed kernel computation. While prior work in the field [197] explored similar ideas within a single-qubit/single-client framework, we extend and generalize these principles to an n-qubit system, k-client system, and thus are capable of handling high-dimensional data in diverse settings.

Our architecture comprises multiple clients, a central server, and a helper entity. The clients hold sensitive data from which they want to learn privately and collaboratively. The central server is tasked with computing the kernel securely and privately. The helper prepares entangled quantum states to facilitate quantum communication. All entities in this setup are capable of performing the necessary quantum operations. The architecture is depicted in Figure III.1.

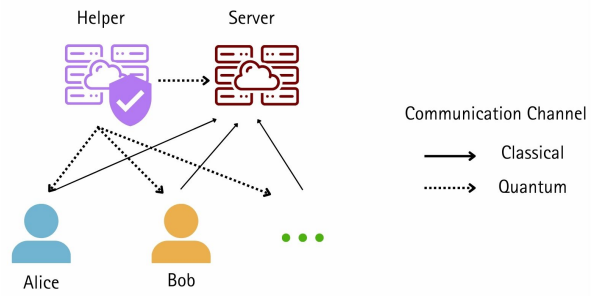


Figure III.1: Visualization of our architecture consisting of a helper, a server, and multiple clients.

We employ an infrastructure in which clients are initially provided with their shared seeds securely, using established cryptographic primitives. The helper party ensures the fair distribution of seeds, adhering to standard privacy-preserving protocols.

Protocol Description

Without loss of generality, we describe our protocol with two participants. Our method naturally extends to any number of participants. To begin the protocol, Alice and Bob declare the size of their classical data in bits, denoted by N. The helper entity then computes the number of qubits, n, needed to encode the data for a single participant based on the chosen encoding technique. The protocol is established within a total system of $(6n+1)$ qubits.

The protocol's circuit diagram is detailed in Figure III.2 below. The correctness of the protocol is theoretically shown in Appendix III.7.1.

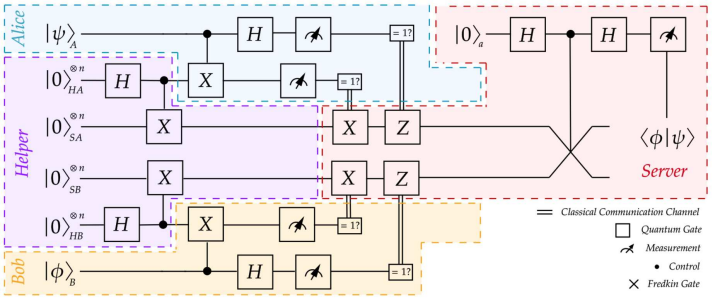


Figure III.2: Quantum circuit diagram associated with our secure and distributed quantum-based kernel computation architecture.

Helper: Quantum State Preparation for Teleportation

The helper generates $2n$ Bell states, which are maximally entangled two-qubit states, to facilitate quantum teleportation. The helper begins by distributing the qubits between Alice, Bob, and the server as follows:

1. In the first set of n Bell states, one qubit from each entangled pair represented by $|0\rangle_{HA}$ is sent to Alice, and the other represented by $|0\rangle_{SA}$ to the server.
2. In the remaining n Bell states, one qubit from each entangled pair represented by $|0\rangle_{HB}$ is sent to Bob, and the other represented by $|0\rangle_{SB}$ to the server.

This enables the quantum teleportation of Alice's and Bob's encoded data to the server for secure computation.

Clients: Data Encoding and Measurement

Alice and Bob determine the encoding of their data represented by $|\psi\rangle_A$ and $|\phi\rangle_B$ respectively, with multiple encodings for every data point based on the required model accuracy. The encoding sequence is derived from the initial shared seed. Subsequently, Alice and Bob execute the following steps:

1. Apply a Controlled-X gate to the qubits they received from the helper using their original quantum data as control.
2. Apply a Hadamard gate to their original data.
3. Measure their data and the received qubits in the computational basis.
4. Communicate the results to the server through an encrypted classical communication channel.

The server applies the appropriate X and Z gates to adjust the qubits it holds.

Server: Inner Product Measurement

The server then executes a standard swap test [211]. It prepares an ancilla qubit in the zero state, applies a Hadamard gate, and uses it to conditionally swap the two sets of qubits received from Alice and Bob. After reverting the ancilla qubit with another Hadamard and measuring it, the output helps determine the required inner product [212]. This measurement process is repeated p times to enhance accuracy.

III.4.4 Security of Protocol

In our proposed adversarial model, clients, including Alice and Bob, as well as the server, are semi-honest. They adhere to the defined protocol, yet may attempt to infer additional information from the data they handle. A helper entity, deemed a semi-honest third-party, guarantees the integrity of the quantum states used in communication, similar to protocols implementing secure multi-party computation (SMPC) [94]. The server is explicitly characterized as non-colluding with the clients, consistent with established norms in distributed and federated architectures [**?** 1].

In our setup, each client processes exclusively their own data and cannot access information from other clients, effectively mitigating the risk of adversarial clients learning the data from honest input parties. The non-colluding server does not know the series of encodings applied to the original data and hence cannot reconstruct the original classical data from the quantum data it receives since it does not know how to measure it. It only learns the kernel matrix reflecting similarities between participants' data. However, since the labels are obfuscated and are irrelevant to model training, the server gains no knowledge beyond the similarity distribution pertaining to obscured labels. Further, following the same argument,

we note that in the case of the presence of colluding malicious clients and a non-colluding malicious server, the non-adversarial clients' data remain private. However, malicious participants can affect the accuracy of the model.

An adversarial third-party attempting to eavesdrop on the quantum data would face significant challenges due to the no-cloning theorem [181], which prohibits the duplication of quantum information without destroying the original information. In the event of interception, the adversarial entity would need to generate and transmit its own quantum data to the server. This can be effectively detected if clients and the server periodically exchange predetermined random quantum states, allowing the server to check for any discrepancies indicative of interference [197]. Additionally, the utility of intercepted data is limited for the third-party as the encoding of data for transmission is randomized and only known to the clients through the pre-shared seed.

III.5 Results

All the proof-of-concept experiments in our evaluation were carried out using classical computing resources in a High-Performance Computing (HPC) cluster. Each node within this HPC environment was equipped with an Intel XEON CPU E5-2650 v4, complemented by 256 GB of memory and 2 TB of SSD storage capacity. We used the Qiskit Aer Simulator to run the program offline due to limited access to IBM's quantum resources. As a result, we were restricted to simulating only 31 qubits in our environment. Note that we do not report any timings since the experiments are run on a simulator.

Our experiments focused on computing the linear kernel. Given the limitation of simulating only 31 qubits, which confines us to 2^7 features, we adopted this approach and assigned $n = 7$ qubits to each party in our distributed setup. While implementing other kernels, such as encoding-induced kernels, RBF kernels, polynomial kernels, and Laplacian kernels, would require more qubits than available, our primary goal is to validate the architecture rather than exhaustively test every encoding. Since the validity of these encodings has already been theoretically established, our focus is on demonstrating that our architecture functions as expected within this framework.

In addition, we tested our methodology in a two-party configuration. This can be easily expanded, as the data can be redistributed to additional participants while maintaining consistent results. Due to the constraints on qubit simulation, a two-party configuration is employed, allocating 14 qubits for the two data providers, 14 for the helper, and 1 for the server.

III.5.1 Accuracy Analysis

We present a comparative analysis of our distributed quantum kernel learning setup against centralized quantum kernel computation and centralized classical kernel computation. Centralized quantum kernel computation only performs the swap test and does not constitute quantum teleportation. The datasets used for this analysis are widely used and publicly

available. These include the Wine dataset (178 samples, 13 features) [213], the Parkinson's disease dataset (197 samples, 23 features) [214], and the Framingham Heart Study dataset (4238 samples, 15 features) [215]. Kernel-based training was performed using SVM for all datasets, and PCA was applied to the binary datasets (Parkinson's and Framingham Heart Study) to reduce dimensionality. After applying PCA, SVM was used on the transformed data to obtain accuracy metrics. All SVM training and evaluation were performed using stratified 5-fold cross-validation to ensure unbiased accuracy metrics. The accuracies of the different models on the datasets are summarized in Table III.1 below.

Dataset (Samples × Features)	**Method**	**Accuracy**		
		Centralised Classical	Centralised Quantum	Distributed Quantum
Wine (178 × 13)	kernel-SVM	0.9860 ± 0.0172	0.8805	0.8874 ± 0.0259
Parkinsons (197 × 23)	kernel-SVM	0.8196 ± 0.0644	0.7875	0.7983 ± 0.0798
	kernel-PCA	0.7872 ± 0.0716	0.7451	0.7660 ± 0.0744
Framingham Heart Study (4238 × 15)	kernel-SVM	0.6788 ± 0.0108	0.6308	0.6340 ± 0.0143
	kernel-PCA	0.6788 ± 0.0095	0.6249	0.6422 ± 0.0092

Table III.1: Comparison of accuracies across different methods and datasets.

As expected, centralized classical methods generally achieve the highest accuracy, serving as a baseline. Centralized quantum methods show competitive performance, although slightly lower than their classical counterparts, due to the inherent characteristics of quantum data and quantum simulators. Our distributed quantum architecture exhibits comparable but not the same accuracy as the centralized quantum architecture because of the complexity introduced by additional gates in the quantum circuit. All experiments were conducted with 1024 shots of the quantum circuit to ensure reliable accuracy. Here, shots refers to the number of times, p, a circuit is repeated.

III.5.2 Effect of Noise

Quantum computing is susceptible to various types of errors due to environmental interactions and imperfections in quantum gate implementations. Our objective is to evaluate the performance of distributed kernel-based QML under different noise conditions on Qiskit and compare it with a classical SVM. We employed three noise models:

No Noise: This model assumes an ideal environment without noise. It serves as a baseline for evaluating the performance of our protocol in the absence of errors.

Noise Level 1: This model introduces a depolarizing error with an error rate of 0.1% for single-qubit gates and two-qubit gates. The depolarizing error is a type of quantum error in which a qubit with a certain probability is replaced by a completely mixed state, losing all of its original information.

Noise Level 2: This model simulates a more challenging environment with a depolarizing error rate of 1%.

Our results reported in Figure III.3 show that increasing noise had a negative impact on model performance.

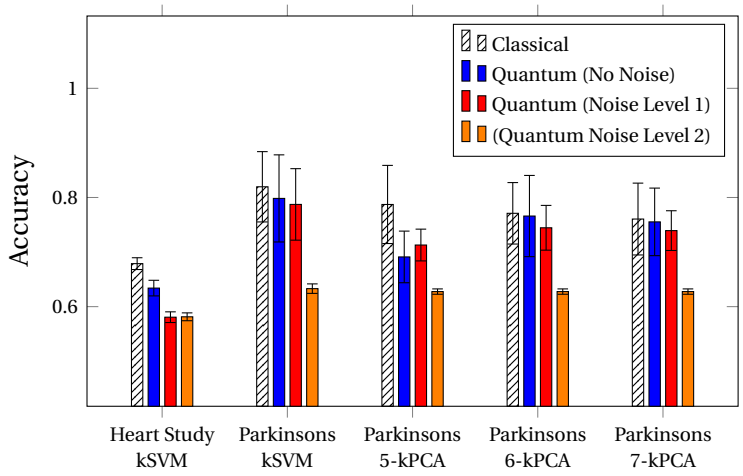


Figure III.3: Comparison of accuracy scores across different noise levels. The baseline includes centralized classical kernel computation and our distributed quantum kernel computation with no noise. We incrementally introduce noise, using depolarizing error at Level 1 and Level 2, to evaluate and report the corresponding accuracy loss.

III.5.3 Effect of Shots

Here, we detail the impact of varying the number of shots used in Qiskit to repeat a quantum circuit on the performance of our proposed algorithm. We used a subset of the Digits dataset containing 100 samples [216]. The objective was to classify these samples into 10 labels (0-9) and to evaluate the classification accuracy using linear kernel-based SVM.

We varied the number of shots, specifically using 128, 256, 512, and 1024 shots, to observe the effect on the classification accuracy. The results, depicted in Figure III.4, indicate improved performance with an increased number of shots.

III.6 Conclusion

In this study, we introduced a novel kernel-based QML algorithm that operates within a distributed and secure framework. Using the implicit connection between quantum encoding and kernel computation, our method supports the calculation of encoding-induced and traditional kernels. To that end, we extended the existing body of knowledge by introducing three novel quantum feature maps designed to compute the polynomial, RBF, and Laplacian kernels, building upon previous studies that primarily focused on linear and homogeneous polynomial kernels. Furthermore, we have theoretically validated that the proposed quantum feature maps help to compute the concerned kernels.

Using a hybrid quantum-classical architecture, our approach functions in a distributed

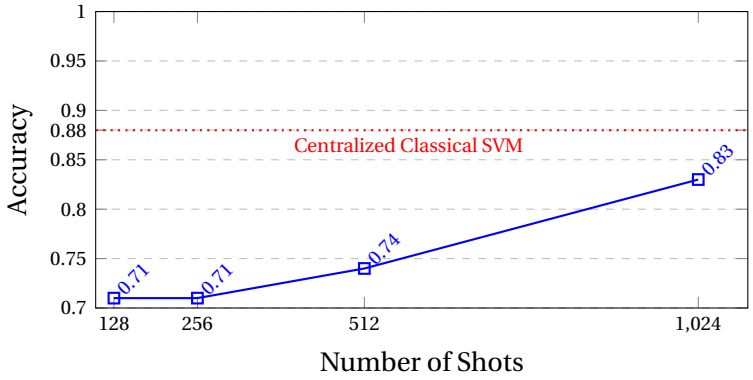


Figure III.4: Accuracy of linear kernel-based SVM on a subset of the Digits dataset featuring 100 samples and 10 labels, compared against the number of shots run by the simulator.

environment where a central server assists data providers in processing their data collaboratively, akin to a centralized model. This setup primarily addresses the case of a semi-honest scenario—a common consideration in studies involving distributed architectures. We have demonstrated that our proposed framework upholds security against semi-honest parties as well as external eavesdroppers.

The application of our architecture to compute the linear kernel for publicly available datasets using Qiskit's Aer Simulator validates our distributed framework, yielding accuracies comparable to those achieved in centralized classical and quantum frameworks.

In the future, our aim is to adapt our methodology to scenarios that involve actively malicious entities. Additionally, given the rich potential of kernel theory, further theoretical and practical exploration of quantum feature maps represents a promising direction for future research in the field.

Code Availability

Our code is available on GitHub at the following URL:
https://github.com/mdppml/distributed-secure-kernel-based-QML [**?**].

Acknowledgements

This research was supported by the German Federal Ministry of Education and Research (BMBF) under project number 01ZZ2010 and partially funded through grant 01ZZ2316D (PrivateAIM). The authors acknowledge the usage of the Training Center for Machine Learning (TCML) cluster at the University of Tübingen.

Broader Impact Statement and Ethical Concerns

Our work focuses on computing commonly used kernels in machine learning using quantum computing. As a piece of fundamental research, our contribution is theoretical in nature and does not, in itself, pose any direct negative societal impacts or ethical concerns. The methods and datasets involved do not contain sensitive personal data, nor do they target or adversely affect any vulnerable populations.

While it is acknowledged that kernel-based machine learning methods can, in certain applications, be associated with issues such as bias amplification or limited interpretability, especially when applied to non-curated datasets or in safety-critical systems, our work builds upon established techniques without introducing fundamentally new methods that would exacerbate these concerns. We recognize that downstream applications of kernel methods may encounter ethical challenges. However, such considerations are intrinsic to the broader field rather than a consequence of our specific contribution.

Given the theoretical scope of this work, no further mitigation strategies are necessary. Nevertheless, we urge practitioners to consider the ethical implications in any concrete application of these methods.

III.7 Supplementary Material

III.7.1 Correctness of Architecture

In this section, we provide a theoretical proof of correctness for our architecture, demonstrating that it accurately computes the kernel matrix for given input data. Without loss of generality, consider an n-qubit system, where Alice's encoded data is represented by $|\psi\rangle_A$ and Bob's data by $|\psi\rangle_B$. As illustrated in Figure III.1, the Helper initializes the system by preparing $2n$ Bell states. The initial state consists of $|0\rangle_{HA}^{\otimes n}$, $|0\rangle_{SA}^{\otimes n}$, $|0\rangle_{SB}^{\otimes n}$, and $|0\rangle_{HB}^{\otimes n}$, where the superscript denotes the qubits in each subsystem, e.g., the i-th qubit of Alice's data is represented as $|\psi\rangle_A^i$.

Let $|\psi\rangle$ be written as $\alpha|0\rangle+\beta|1\rangle$ and $|\phi\rangle$ as $\delta|0\rangle+\gamma|1\rangle$ in the computational basis. Initially, the entire system is in the state:

$$|\psi\rangle_A^{\otimes n}\otimes|0\rangle_{HA}^{\otimes n}\otimes|0\rangle_{SA}^{\otimes n}\otimes|0\rangle_{SB}^{\otimes n}\otimes|0\rangle_{HB}^{\otimes n}\otimes|\phi\rangle_B^{\otimes n}. \tag{III.28}$$

For simplicity, we track only the i-th qubit of each n-qubit state:

$$|\psi\rangle_A^i\otimes|0\rangle_{HA}^i\otimes|0\rangle_{SA}^i\otimes|0\rangle_{SB}^i\otimes|0\rangle_{HB}^i\otimes|\phi\rangle_B^i. \tag{III.29}$$

After applying Hadamard gates to the Helper qubits, the system evolves to the following state:

$$|\psi\rangle_A^i\otimes\frac{1}{\sqrt{2}}\left(|0\rangle_{HA}^i+|1\rangle_{HA}^i\right)\otimes|0\rangle_{SA}^i\otimes|0\rangle_{SB}^i\otimes\frac{1}{\sqrt{2}}\left(|0\rangle_{HB}^i+|1\rangle_{HB}^i\right)\otimes|\phi\rangle_B^i. \tag{III.30}$$

Upon applying the Controlled-X gates, the system is entangled, preparing it for quantum teleportation:

$$\frac{1}{2}\left(|\psi\rangle_A^i \otimes \left(|00\rangle_{HA,SA}^i + |11\rangle_{HA,SA}^i\right) \otimes \left(|00\rangle_{SB,HB}^i + |11\rangle_{SB,HB}^i\right) \otimes |\phi\rangle_B^i\right). \tag{III.31}$$

Next, Alice and Bob perform Controlled-X gates, resulting in the state:

$$\begin{aligned}&\frac{1}{2}\left(\alpha|000\rangle_{A,HA,SA}^i + \beta|110\rangle_{A,HA,SA}^i + \alpha|011\rangle_{A,HA,SA}^i + \beta|101\rangle_{A,HA,SA}^i\right)\\ &\otimes\frac{1}{2}\left(\gamma|000\rangle_{SB,HB,B}^i + \delta|011\rangle_{SB,HB,B}^i + \gamma|110\rangle_{SB,HB,B}^i + \delta|101\rangle_{SB,HB,B}^i\right).\end{aligned}$$

After applying Hadamard gates, the system evolves to:

$$\begin{aligned}&\frac{1}{4}\Big(\alpha|000\rangle_{A,HA,SA}^i + \alpha|100\rangle_{A,HA,SA}^i + \beta|010\rangle_{A,HA,SA}^i - \beta|110\rangle_{A,HA,SA}^i\\ &\quad+\alpha|011\rangle_{A,HA,SA}^i + \alpha|111\rangle_{A,HA,SA}^i + \beta|001\rangle_{A,HA,SA}^i - \beta|101\rangle_{A,HA,SA}^i\Big)\\ &\otimes\frac{1}{4}\Big(\gamma|000\rangle_{SB,HB,B}^i + \gamma|001\rangle_{SB,HB,B}^i - \delta|011\rangle_{SB,HB,B}^i + \delta|010\rangle_{SB,HB,B}^i\\ &\quad+\gamma|110\rangle_{SB,HB,B}^i + \gamma|111\rangle_{SB,HB,B}^i - \delta|101\rangle_{SB,HB,B}^i + \delta|100\rangle_{SB,HB,B}^i\Big).\end{aligned}$$

Once the classical bits are communicated to the server, the server applies the appropriate X and Z gates, resulting in the system state:

$$|0\rangle_a(\alpha|0\rangle_{SA} + \beta|1\rangle_{SA}) \otimes (\gamma|0\rangle_{SB} + \delta|1\rangle_{SB}). \tag{III.32}$$

After applying a Hadamard gate, the server obtains:

$$\frac{1}{\sqrt{2}}(|0\rangle_a + |1\rangle_a)(|\psi\rangle_{SA}) \otimes (|\phi\rangle_{SB}). \tag{III.33}$$

The final step involves applying Fredkin gates, with the ancilla qubit as the control:

$$\frac{1}{\sqrt{2}}\left(|0\psi\phi\rangle_{a,SA,SB} + |1\phi\psi\rangle_{a,SA,SB}\right). \tag{III.34}$$

Upon applying a Hadamard gate to the ancilla qubit, we obtain:

$$\frac{1}{2}\left(|0\rangle_a \otimes (|\psi\phi\rangle_{SA,SB} + |\phi\psi\rangle_{SA,SB}) + |1\rangle_a \otimes (|\psi\phi\rangle_{SA,SB} - |\phi\psi\rangle_{SA,SB})\right). \tag{III.35}$$

Measuring the ancilla qubit along the computational basis yields:

$$Pr(0)_a = \frac{1}{4}\left(\langle\psi|\langle\phi| + \langle\phi|\langle\psi|\right)\left(|\psi\rangle|\phi\rangle + |\phi\rangle|\psi\rangle\right) = \frac{1}{2} + \frac{1}{2}\|\langle\psi|\phi\rangle\|^2, \tag{III.36}$$

where, $P(0)_a$ is the probability that the anscilla qubit is in the $|0\rangle$ state. Rearranging, this then determines the inner product of $|\psi\rangle$ and $|\phi\rangle$.

$$\|\langle\psi|\phi\rangle\| = \sqrt{2Pr(0)_a - 1}, \tag{III.37}$$

III.7.2 Error Analysis and Measurement Complexity

Shot Complexity for Inner Product Estimation

In this section, we derive the relationship between the number of measurement shots, M, and the additive error ϵ in estimating the inner product of quantum states.

As discussed earlier in (III.37), the probability of obtaining the outcome $|0\rangle$ in the inner product measurement is

$$p = \frac{1 + \|\langle\psi|\phi\rangle\|^2}{2}.$$

For each shot, we define the Bernoulli random variable X_i by

$$X_i = \begin{cases} 1, & \text{if the outcome is } |0\rangle, \\ 0, & \text{if the outcome is } |1\rangle. \end{cases} \tag{III.38}$$

We know $\mathbb{E}[X_i] = p$ and $\mathrm{Var}(X_i) = p(1-p)$. After M independent shots, the sample mean is $\bar{X} = \sum_{i=1}^{M} X_i/M$, with $\mathbb{E}[\bar{X}] = p$ and $\mathrm{Var}(\bar{X}) = (p(1-p)/M)$. Labeling $(1 + \|\langle\psi|\phi\rangle\|)^2/2 = l$, an estimator for l is $\hat{l} = 2\bar{X} - 1$. Its variance is $\mathrm{Var}(\hat{l}) = 4\,\mathrm{Var}(\bar{X}) = (4p(1-p))/M)$, and the standard deviation is

$$\sigma_{\hat{c}} = 2\sqrt{\frac{p(1-p)}{M}}.$$

The standard deviation attains its maximum at $p = 1/2$. In that case, $p(1-p) = 1/4$, and hence $\sigma_{\hat{c}} \leq 1/\sqrt{M}$. To achieve an additive error ϵ in estimating l, we thus require $1/\sqrt{M} \leq \epsilon$, which implies

$$M = \mathcal{O}\left(\frac{1}{\epsilon^2}\right).$$

Determining Number of Qubits for the RBF Kernel

In this section, we derive the relationship between the number of qubits required, given by $\lceil \log_2(2D) \rceil$, and the additive error ϵ in approximating the RBF kernel. We consider the quantum feature map corresponding to the RBF kernel defined in (III.16).

$$x \in \mathbb{R}^d \mapsto \psi(x) = \frac{1}{\sqrt{D}} \sum_{j=1}^{D} \Big(\cos(w_j^T x)\,|2j-2\rangle + \sin(w_j^T x)\,|2j-1\rangle\Big).$$

Here, w_j are drawn independently from the multivariate normal distribution $\mathcal{N}(0, \sigma^{-2} I)$. The associated kernel approximation is then given by

$$\hat{K}(x,y) = \langle\psi(x), \psi(y)\rangle = \frac{1}{D} \sum_{j=1}^{D} \cos\Big(w_j^T (x-y)\Big).$$

As shown in (III.21), the expectation of this approximation is

$$\mathbb{E}\left[\hat{K}(x,y)\right] = \exp\left(-\frac{1}{2\sigma^2}\|x-y\|^2\right) = K_{\mathrm{RBF}}(x,y).$$

Define for each $j = 1, \ldots, D$, the random variables $X_j = \cos\left(w_j^T (x-y)\right)$. Then, the kernel approximation can be written as $\hat{K}(x,y) = \sum_{j=1}^{D} X_j / D$. Next, we compute the variance of each X_j. Let $Z_j = w_j^T (x-y)$. Because $w_j \sim \mathcal{N}(0, \sigma^{-2} I)$, it follows that $Z_j \sim \mathcal{N}\left(0, \|x-y\|^2/\sigma^2\right)$. Standard results for the cosine of a Gaussian random variable yield

$$\mathbb{E}[\cos(Z_j)] = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right), \quad \mathbb{E}[\cos^2(Z_j)] = \frac{1}{2}\left(1 + \exp\left(-\frac{\|x-y\|^2}{\sigma^2}\right)\right).$$

Thus, the variance of X_j is

$$v := \mathrm{Var}(X_j) = \mathbb{E}[\cos^2(Z_j)] - \left(\mathbb{E}[\cos(Z_j)]\right)^2 = \frac{1}{2}\left[1 - \exp\left(-\frac{\|x-y\|^2}{\sigma^2}\right)\right].$$

Since the X_j are i.i.d., the variance of the average $\hat{K}(x,y)$ is $\mathrm{Var}\left(\hat{K}(x,y)\right) = v/D$.

We then define the centered random variables $Y_j := X_j - \mathbb{E}[X_j]$. Each Y_j then has zero mean and variance v. Since $\cos(\cdot)$ is bounded in $[-1,1]$, it follows that $|X_j| \le 1$ and $|X_j - \mathbb{E}[X_j]| = |Y_j| \le 2$,. We then apply Bernstein's inequality [217] to the sum $\sum_{j=1}^{D} Y_j$, whose summands are bounded by M (here $M = 2$), and whose total variance is $\sigma^2 = vD$. We have for any $t > 0$,

$$\Pr\left(\left|\sum_{j=1}^{D} Y_j\right| \ge t\right) \le 2\exp\left(-\frac{t^2}{2vD + \frac{2}{3}Mt}\right).$$

Setting $t = D\epsilon$, we see

$$\Pr\left(\left|\hat{K}(x,y) - \mathbb{E}[\hat{K}(x,y)]\right| \ge \epsilon\right) \le 2\exp\left(-\frac{D\epsilon^2}{2v + \frac{4}{3}\epsilon}\right).$$

For sufficiently small ϵ (i.e. when the term $\frac{4}{3}\epsilon$ is negligible relative to $2v$), this simplifies to

$$\Pr\left(\left|\hat{K}(x,y) - K_{\mathrm{RBF}}(x,y)\right| \ge \epsilon\right) \le 2\exp\left(-\frac{D\epsilon^2}{2v}\right).$$

To ensure that the additive error is bounded by ϵ with probability at least $1-\delta$, we require

$$2\exp\left(-\frac{D\epsilon^2}{2v}\right) \le \delta.$$

Taking natural logarithms and rearranging, we obtain

$$D \ge \frac{2v}{\epsilon^2}\ln\left(\frac{2}{\delta}\right).$$

Thus, to guarantee an additive error of at most ϵ with confidence $1-\delta$, the number of random features must satisfy

$$D = \mathcal{O}\left(\frac{v}{\epsilon^2}\right).$$

For large $\|x-y\|^2/\sigma^2$, we have $\nu = 1/2$, and for small $\|x-y\|^2/\sigma^2$, we can approximate the exponential term in ν with the first order of the Taylor expansion. Hence

$$D = \begin{cases} \mathcal{O}\left(\frac{1}{\epsilon^2}\right), & \text{for large } \|x-y\|^2/\sigma^2, \\ \mathcal{O}\left(\frac{\|x-y\|^2}{\sigma^2\epsilon^2}\right), & \text{for small } \|x-y\|^2/\sigma^2. \end{cases} \tag{III.39}$$

It is important to note that the constant ν depends on the ratio $\|x-y\|^2/\sigma^2$. For a fixed pair (x, y), increasing σ results in a smaller ν. Consequently, for larger σ (i.e., for a wider kernel), a smaller number of random features D is required to achieve a given error tolerance ϵ. Conversely, for smaller σ (i.e., for a narrower kernel), the ratio $\|x-y\|^2/\sigma^2$ increases, leading to a larger ν and thus a larger D is necessary. This can be seen in Figure III.5(a).

Moreover, when σ becomes very small (for example, when $\sigma < 0.5$) or when $\|x-y\|^2$ is very large (i.e., $x, y \in \mathbb{R}^d$, and d is very large), the exponential term in ν rapidly approaches zero, and ν asymptotically approaches its maximum value of $1/2$. In this regime, the relationship between D and ϵ becomes independent of σ. Consequently, the error curves will converge as we observe in Figure III.5(c).

Determining Number of Qubits for the Laplacian Kernel

In this section, we derive the relationship between the number of qubits required, given by $\lceil \log_2(2D) \rceil$, and the additive error ϵ in approximating the Laplacian kernel. We consider the quantum feature map corresponding to the Laplacian kernel defined in (III.22) as

$$x \in \mathbb{R}^d \mapsto \psi(x) = \frac{1}{\sqrt{D}} \sum_{j=1}^{D} \left(\cos\left(w_j^T x + \alpha_j\right) |2j-2\rangle + \sin\left(w_j^T x + \alpha_j\right) |2j-1\rangle \right),$$

where w_j are drawn independently from the Cauchy distribution $\mathcal{C}(0, \alpha^{-1} I)$ and the phase shifts α_j are independent samples from the uniform distribution $\mathcal{U}(0, 2\pi)$. The associated kernel approximation is then given by

$$\hat{K}(x, y) = \langle \psi(x), \psi(y) \rangle = \frac{1}{D} \sum_{j=1}^{D} \cos\left(w_j^T (x-y)\right).$$

As shown in (III.27), the expectation of this approximation is

$$\mathbb{E}\left[\hat{K}(x, y)\right] = \exp\left(-\frac{\|x-y\|_1}{\alpha}\right) = K_L(x, y).$$

As before, define for each $j = 1, \ldots, D$, $X_j = \cos\left(w_j^T (x-y)\right)$. Then, the kernel approximation can be written as $\hat{K}(x, y) = \sum_{j=1}^{D} X_j / D$. Next, we compute the variance of each X_j. Let $Z_j = w_j^T (x-y)$. Since w_j is drawn from $\mathcal{C}(0, \alpha^{-1} I)$ and Cauchy distributions remain closed

under linear transformations, we have

$$Z_j \sim \mathscr{C}\left(0,\gamma\right), \quad \gamma = \frac{\|x-y\|_1}{\alpha}.$$

The characteristic function of a Cauchy random variable $Z_j \sim \mathscr{C}(0,\gamma)$ is given by $\phi_{Z_j}(t) = \exp\left(-\gamma|t|\right)$. Thus, setting $t=1$ we obtain

$$\mathbb{E}[\cos(Z_j)] = \exp(-\gamma) = \exp\left(-\frac{\|x-y\|_1}{\alpha}\right).$$

Moreover, using the trigonometric identity $\cos^2(Z_j) = (1+\cos(2Z_j))/2$, we have

$$\mathbb{E}[\cos^2(Z_j)] = \frac{1+\exp(-2\gamma)}{2} = \frac{1+\exp\left(-\frac{2\|x-y\|_1}{\alpha}\right)}{2}.$$

Thus, the variance of X_j is

$$v := \mathrm{Var}(X_j) = \mathbb{E}[\cos^2(Z_j)] - \left(\mathbb{E}[\cos(Z_j)]\right)^2 = \frac{1-\exp\left(-\frac{2\|x-y\|_1}{\alpha}\right)}{2}.$$

Now, following the same procedure as in III.7.2 by defining $Y_j = X_j - \mathbb{E}[X_j]$ and applying Bernstein's inequality on $\sum_{j=1}^{D} Y_j$, we find that

$$D = \begin{cases} \mathscr{O}\left(\frac{1}{\epsilon^2}\right), & \text{for large } \|x-y\|_1/\alpha, \\ \mathscr{O}\left(\frac{\|x-y\|_1}{\alpha\epsilon^2}\right), & \text{for small } \|x-y\|_1/\alpha. \end{cases} \tag{III.40}$$

It is important to note that the constant v depends on the ratio $\|x-y\|_1/\alpha$. For a fixed pair (x,y), increasing α results in a small v. Consequently, for larger α (i.e., for a wider Laplacian kernel), the required D to achieve a given error tolerance ϵ is smaller. Conversely, for a smaller α (i.e., for a narrower kernel), v increases. Thus a larger D is necessary. This can be seen in Figure III.5(b). Similar to before, when α becomes very small (for example, when $\alpha < 1$) or when $\|x-y\|_1$ is very large (i.e., $x, y \in \mathbb{R}^d$, and d is very large), the exponential term steadily approaches zero, and v asymptotically approaches its maximum value of $1/2$. In this regime, the relationship between D and ϵ becomes independent of α as we observe in Figure III.5(d).

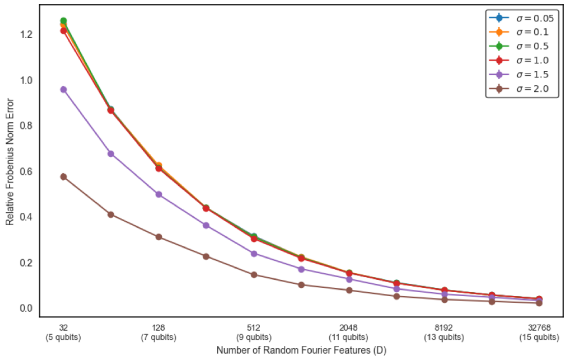
Classical Simulation of Kernel Approximation Error

In this section, we describe experiments run on a classical computer to validate the theoretical claims made in Sections III.7.2 and III.7.2. Our goal is to assess the quality of the kernel approximations obtained via our feature maps for both the RBF and Laplacian kernels. To quantify the approximation quality, we compute the relative Frobenius norm

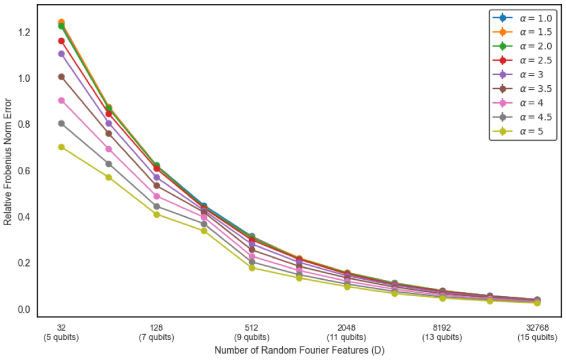
error between the estimated kernel matrix and the exact kernel matrix. The error metric is defined as

$$\frac{\|K_{\text{exact}} - K_{\text{approx}}\|_F}{\|K_{\text{exact}}\|_F},$$

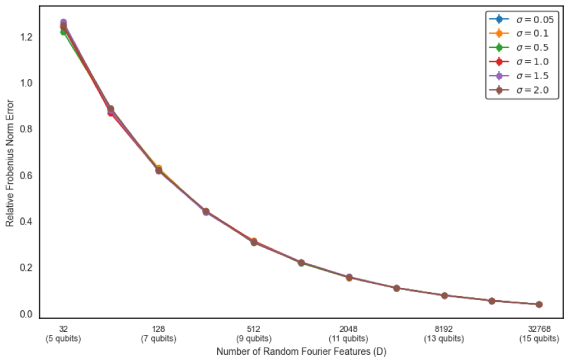
where $\|\cdot\|_F$ is the Frobenius norm.



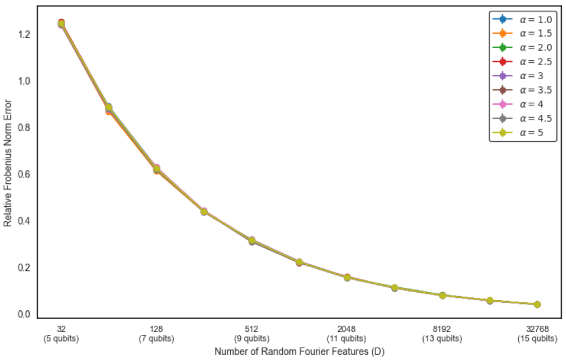
(a) RBF Kernel, $d = 10$. Different curves correspond to various values of σ, with the x-axis showing the number of random features D (and corresponding qubits, $\lceil \log_2(2D) \rceil$).



(b) Laplacian Kernel, $d = 10$. Different curves correspond to various values of α, with the x-axis showing the number of random features D (and corresponding qubits, $\lceil \log_2(2D) \rceil$).



(c) RBF Kernel, $d = 100,000$. In this regime the $\|x-y\|^2$ term dominates, causing the variance to approach $1/2$ and all curves to coincide.



(d) Laplacian Kernel, $d = 100,000$. Again, the $\|x-y\|_1$ term dominates, causing the variance to approach $1/2$ and all curves to coincide.

Figure III.5: Kernel approximation error curves for the RBF and Laplacian kernels under two data regimes. All experiments were performed on simulated datasets with 100 samples, and each experiment was repeated 5 times with the errors reported as averages. **Top:** Experiments with a small feature dimension ($d = 8$) show clear dependence on σ (RBF) or α (Laplacian). **Bottom:** Experiments with a large feature dimension ($d = 100,000$) exhibit nearly overlapping curves regardless of kernel parameters, as the large $\|x-y\|$ values force the variance to saturate at $1/2$, resulting in $D = \mathcal{O}(1/\epsilon^2)$.

Our empirical results as seen in Figure III.5 confirm that the kernel approximation error decreases as the number of random features D increases. In both the RBF and Laplacian

kernel experiments, the observed decay rate aligns with the theoretical $\mathcal{O}(1/\sqrt{D})$ behavior predicted earlier. These findings, albeit using classical resources provide strong evidence that the proposed feature maps produce reliable approximations to exact kernel matrices.

IV Accelerating Targeted Hard-Label Adversarial Attacks in Low-Query Black-Box Settings

Arjhun Swaminathan **Mete Akgün**

Abstract

Deep neural networks for image classification remain vulnerable to adversarial examples — small, imperceptible perturbations that induce misclassifications. In black-box settings, where only the final prediction is accessible, crafting targeted attacks that aim to misclassify into a specific target class is particularly challenging due to narrow decision regions. Current state-of-the-art methods often exploit the geometric properties of the decision boundary separating a source image and a target image rather than incorporating information from the images themselves. In contrast, we propose Targeted Edge-informed Attack (TEA), a novel attack that utilizes edge information from the target image to carefully perturb it, thereby producing an adversarial image that is closer to the source image while still achieving the desired target classification. Our approach consistently outperforms current state-of-the-art methods across different models in low query settings (nearly 70% fewer queries are used), a scenario especially relevant in real-world applications with limited queries and black-box access. Furthermore, by efficiently generating a suitable adversarial example, TEA provides an improved target initialization for established geometry-based attacks.

IV.1 Introduction

Deep neural networks have achieved remarkable performance in image classification tasks, powering applications from autonomous systems [218, 219] to medical diagnostics [220, 221]. However, they have repeatedly been shown to be vulnerable to adversarial examples [22, 222]. These are small, often imperceptible perturbations to a correctly classified image that cause a misclassification. Although many of these attacks assume white-box access to a model's internals, hard-label (decision-based) attacks offer a more challenging yet practical setting where only the top-1 predicted label is observed. This limited-feedback scenario commonly arises in commercial APIs [223]. In this realm, targeted attacks, which push the model's prediction to a specific target class, are inherently more difficult since the decision regions corresponding to the specific target classes are usually narrower and more isolated.

In black-box settings, targeted hard-label attacks have become an active area of research, with several techniques proposed in the literature [224, 225, 226, 227, 228], with state-of-the-art methods relying on the geometry of the decision boundary separating the source image from the target class. Geometry-informed attacks traverse in a lower-dimensional space and fall into two categories: *Boundary Tracing Attacks* and *Gradient Estimation Attacks*. Boundary Tracing Attacks ([63, 229, 26, 230]) perform walks along the decision boundary while Gradient Estimation Attacks ([67, 231, 232, 233, 24, 234, 235]) perform the same walks but using information about the approximate tangent/normal to the decision boundary in a local neighborhood to their adversarial image. Although powerful for local refinement, both approaches tend to burn through queries when the source image lies far from a given adversarial image in a target class, wasting many queries before reaching a narrow region

where local geometry can more effectively be leveraged.

Further, in practice, many real-world scenarios such as commercial pay-per-query APIs, impose severe constraints on the number of queries that can be made to a target model in a specified time. Often practical query limits may be on the order of a few hundred to fewer than a few thousand queries. Under these conditions, the limited feedback available makes it difficult to effectively use information about the local decision boundary geometry in the early stages of an attack. When the decision space is still wide, movement along restricted lower dimensional spaces leads to limited incremental progress, and gradient estimation methods often use queries in estimating local gradients by sampling predictions in a local neighborhood instead of moving. This raises a pressing need to develop methods that are efficient in the low-query regime. In a high query setting, this would also lead us to a good starting point to employ the geometry-informed methods since we arrive at a good adversarial point quickly and can use the geometry-informed information more effectively. To this end, we propose Targeted Edge-Informed Attack (TEA), a novel targeted adversarial attack designed for hard-label black-box scenarios within a restricted query budget. Rather than depending on the local properties of the decision boundary, TEA leverages the intrinsic features of the target image itself - specifically, the edge information obtained using Sobel filters [236] help identify prominent structural features in the target image. Edges encode high-magnitude spatial gradients that delineate object boundaries [236]. Research has shown that early layers fire on oriented edge filters [237, 238], and that shape/edge cues remain predictive even when textures are suppressed [239]. The core idea is to preserve these low-level features, while applying perturbations to the non-edge regions of an image, allowing us to stay in the target class while pushing the adversarial image towards the source image. When our progress plateaus, evidenced by a series of consecutive queries that fail to achieve further reduction in distance while maintaining target class prediction, one can switch to current state-of-the-art geometry-informed methods for a local refinement procedure. Hence we make the following contributions:

TEA: We introduce an edge-informed perturbation strategy, enabling rapid progress toward the source image in the early stages of an adversarial attack when limited queries are available. TEA involves a two-step process: First, a global edge-informed search is performed, and then, edge-informed updates are applied to small patches using Gaussian weights.

Empirical validation under strict query budgets: We perform extensive evaluations on the ImageNet validation dataset [240] across four architectures (ResNet-50 [221], ResNet-101 [221], VGG16 [241], and ViT [242]) and an adversarially trained architecture (ResNet-50 [243]). Our attack consistently outperforms existing state-of-the-art hard-label methods, including HSJA, Adaptive History-driven Attack (AHA) [25], CGBA, and CGBA-H, under realistic query budgets (fewer than 1000 queries). To achieve a 60% reduction in distance from a target image to a source image, TEA required on average 251 queries across the four models - 70% fewer than AHA (the second fastest), which required 845 queries.

The rest of this paper is structured as follows. Section IV.2 reviews related work on targeted hard-label adversarial attacks. Section IV.3 describes our methodology, while Section IV.4

presents experimental results. Finally, Section IV.5 outlines directions for future research.

IV.2 Related Work

Early work in the realm of targeted hard-label adversarial attacks consisted of seminal work: Boundary Attack (BA) [63], which proposed a method of traversing the decision boundary that separates an adversarial image from a source image. Building on this framework, BiasedBA [229] incorporated directional priors, such as perceptual and low-frequency biases, to restrict the search to more promising regions. BA with Output Diversification Strategy (BAODS) [234] integrates diverse gradient-like signals into the exploration process of BA, thereby strengthening the original method.

Following these foundational methods, Hybrid Attack (HA) [230] employed a combination of heuristic search strategies to balance global exploration and local refinement. Advancing toward more gradient-centric techniques, qFool [232] leverages the local flatness of decision boundaries to streamline the attack process. Meanwhile, HopSkipJumpAttack (HSJA) [67] locally approximates the normal to the decision boundary to "jump off" from it before progressing toward the source image. Complementarily, Tangent Attack exploits locally estimated tangents of the decision surface to steer the adversarial perturbation toward the source image. Addressing the challenges posed by high-dimensional input spaces, Query Efficient Boundary Attack (QEBA) [231] projects gradient estimation into lower-dimensional subspaces, such as the frequency domain, thus significantly reducing the number of required queries.

Incorporating historical query information, Adaptive History-driven Attack (AHA) [25] adapts its search trajectory based on previous successes and failures, while Decision-based query Efficient Adversarial Attack based on boundary Learning (DEAL) [67] employs an evolutionary strategy that concentrates queries on promising regions of the input space. SurFree [26] demonstrated that using 2D planes and semicircular trajectories toward the source image was an effective strategy. This was based on the fact that under the assumption of a flat decision boundary, the point on the boundary that is closest to the source image is precisely where the semicircular trajectory intersects it. Building on this idea, Curvature-Aware Geometric black-box Attack (CGBA) and its variant that is more suited for targeted attacks - CGBA-H [24], use normal estimation at the decision boundary to select the traversal plane. To the best of our knowledge, CGBA-H serves as the current state of the art in decision-based targeted adversarial attacks.

IV.3 Methods

In this section, we formalize the hard-label attack setting and detail our attack, which drastically reduces early query cost, especially when the target image is far from the source image, and exists in a wider decision space. Once the adversarial image achieves a good distance and reaches a narrow decision space, one can use the image as an initialization and continue refining with existing geometric-based methods.

Problem Statement. We consider a hard-label image classifier modeled by

$$f : [0,1]^{C\times H\times W} \to \mathbb{R}^K, \tag{IV.1}$$

where C denotes the number of color channels, H and W are the image height and width, and the classifier distinguishes among K classes. For any query image x, we do not observe its continuous output (e.g., logits or probabilities) but only the predicted label index

$$\hat{y}(x) = \arg\max_{1\le k\le K} [f(x)]_k. \tag{IV.2}$$

Let x_s be a *source* image correctly classified as y_s. In a *targeted* attack, we start with a *target* image x_t (correctly classified as y_t). Our goal is to find an adversarial image x_{adv} that is as close as possible to x_s (in the ℓ_2 norm) while maintaining the target classification, i.e.,

$$x_{\text{adv}} = \arg\min_x \|x - x_s\|_2, \quad \text{subject to} \quad \hat{y}(x) = y_t. \tag{IV.3}$$

We propose a two-part procedure for perturbing the target image x_t to get closer to x_s while maintaining adversarial requirements. First, a *global* edge-informed search coarsely aligns major image regions. Second, a *patch-based* edge-informed search further perturbs local regions in our adversarial image. Note that neither step leverages local decision boundary geometry or gradient estimation, which are typically beneficial only when the adversarial image lies near a narrow decision space and when movements towards the source image are unlikely to maintain classification. This approach avoids burning queries in a wider decision space.

IV.3.1 Global Edge-Informed Search

The global perturbation step aims to coarsely align x_t towards x_s by modifying predominantly smooth, non-edge areas, thereby preserving crucial edge structures that help maintain target classification.

Soft Edge Mask. To this end, we detect edges in x_t using the Sobel operator [236], as depicted in Figure IV.1. A subsequent blurring operation yields a *soft edge mask* M_{edge}, where $M_{\text{edge}}(i,j) \in [0,1]$ has values close to 1 at edge locations and gradually transitions to 0 in smoother regions. We describe this in Box IV.1.

Global Interpolation. Starting from $x_0 = x_t$, we perform iterative updates

$$x_{k+1} \leftarrow x_k + \alpha(x_s - x_k) \odot \left(I - M_{\text{edge}}\right), \tag{IV.4}$$

where $\odot$ denotes the element-wise (Hadamard) product. The scaling factor α is optimized via a momentum based search. Masking out the edge regions in this update ensures that the interpolation primarily affects smooth areas, thus preserving the overall structure necessary for target classification. The complete procedure is summarized in Box IV.2. Once the improvements begin to stagnate, we transition to a local, patch-based refinement.

Box IV.1 Soft Edge Mask

Input: Image x, edge thresholds $\{T_\ell, T_h\}$, Gaussian blur kernel size b, intensity factor γ, small constant ϵ.
Output: Soft edge mask M_{edge}.

1. $x^{\text{gray}} \leftarrow \text{GrayScale}(x)$.
2. $(s_x, s_y) \leftarrow (\text{Sobel}(x^{\text{gray}}, 0), \text{Sobel}(x^{\text{gray}}, 1))$.
3. $G \leftarrow (s_x^2 + s_y^2)^{1/2}$.
4. $G \leftarrow 255 \cdot \big(G/(max_{i,j}(G) + \epsilon)\big)$.
5. $\text{edge\_mask}(i, j) \leftarrow \begin{cases} 255, & \text{if } T_\ell \le G(i, j) \le T_h, \\ 0, & \text{otherwise.} \end{cases}$
6. $\text{blurred} \leftarrow \text{GaussianBlur}(\text{edge\_mask}, (b, b))$.
7. $\text{norm} \leftarrow \text{Normalize}(\text{blurred})$.
8. $M_{\text{edge}} \leftarrow \gamma \cdot \text{norm}$.

Return: M_{edge}.

IV.3.2 Patch-Based Edge-Informed Search

After the global interpolation, some subregions of the image may still display significant discrepancies from x_s. In the patch-based refinement, we then partition the image into randomly selected patches of random sizes

$$\mathscr{P} \subseteq \{1, \dots, H\} \times \{1, \dots, W\}, \tag{IV.5}$$

and for each patch, construct a local soft edge mask $M_{\mathscr{P}}$ by restricting M_{edge} to $\mathscr{P}$. Next, starting from our adversarial image x_k, we apply a local interpolation

$$\widetilde{x}(\beta) = x_k + \beta(x_s - x_k) \odot G \odot (I - M_{\mathscr{P}}), \tag{IV.6}$$

and search for the largest β such that the classifier still predicts the target label, $\hat{y}(\widetilde{x}(\beta)) = y_t$. Here, G is a Gaussian weighting function over the patch that smoothly downweights updates near patch borders, helping to avoid artificial edges introduced by patch boundaries. We repeat this for different patches until a termination criterion is met (e.g., 25 consecutive iterations with no further improvement). The corresponding illustration is depicted in Figure IV.1 and the relevant pseudocode is provided in Box IV.3.

Figure IV.2 offers a visual overview of TEA. The figure depicts the adversarial image throughout our method, and displays a hotspot of individual pixel differences from the source image.

Box IV.2 Global Edge-Informed Search

Input: Source image x_s, target image x_t, soft edge mask M_{edge}, tolerance τ, maximum queries $qc_{\max}$, initial step factor η, momentum μ.
Output: Adversarial image x_{adv} such that $\hat{y}(x_{\text{adv}}) = y_t$.

1. Initialize $x_{\text{current}} \leftarrow x_t$, $v \leftarrow 0$, and $d \leftarrow x_s - x_t$.
2. Set step size $s \leftarrow \|d\|_2 \cdot \eta$, and initialize query count $qc \leftarrow 0$.
3. **while** $qc < qc_{\max}$ **and** $\|s\| \geq \tau$:
   (a) $v \leftarrow \mu \cdot v + (1-\mu) \cdot d$.
   (b) $x_{\text{next}} \leftarrow x_{\text{current}} + s \cdot (v \odot (I - M_{\text{edge}}))$.
   (c) $qc \leftarrow qc + 1$.
   (d) **if** $\hat{y}(x_{\text{next}}) = y_t$:
      i. $x_{\text{current}} \leftarrow x_{\text{next}}$.
      ii. $s \leftarrow 1.1 \cdot s$.
   (e) **else**:
      i. **break**.

Return: x_{current}.

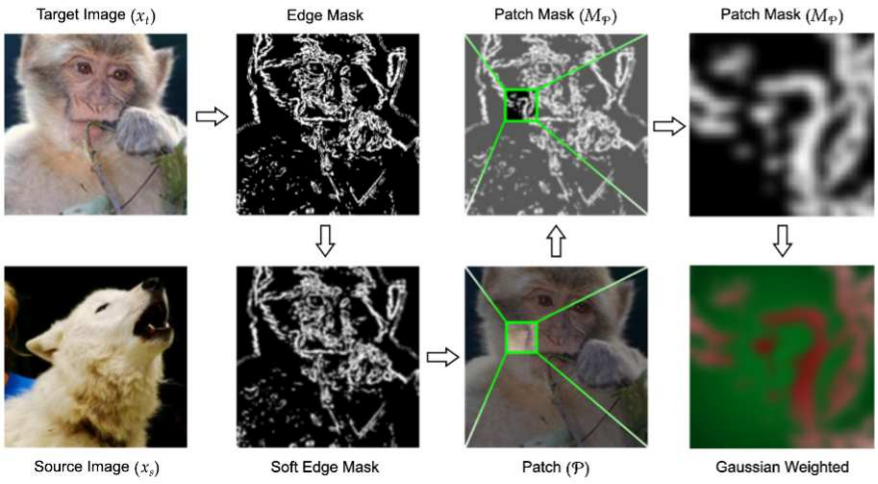


Figure IV.1: **Overview of Patch-Based Edge-Informed Search.** Edge information from the target image, obtained via the Sobel operator, is first blurred to generate a soft edge mask. A square patch is then selected and a Gaussian weighting function is applied. In the bottom right panel, the intensity of the modification is illustrated: dark red regions remain largely unchanged, while light green regions receive a more pronounced update. The lack of changes near the patch borders helps prevent the introduction of artificial edges.

Box IV.3 Patch-Based Edge-Informed Search

Input: Source image x_s, current adversarial image x_{adv}, soft edge mask M_{edge} (from `create_soft_edge_mask`), minimum patch size $p_{\min}$, maximum patch size $p_{\max}$, step factor η, momentum μ, maximum calls $N_{\max}$.
Output: Refined adversarial image x_{adv}.

1. $n_{\text{break}} \leftarrow 0$.
2. **while** $n_{\text{break}} < 25$:
   (a) $D \leftarrow \text{AvgPool}(|x_s - x_{\text{adv}}|)$.
   (b) Select high-difference indices from D and choose a random center (i_c, j_c).
   (c) $p \leftarrow \text{randInt}(p_{\min}, p_{\max})$; $\mathscr{P} \leftarrow \{(i,j) \mid |i - i_c| \le \lfloor p/2 \rfloor, |j - j_c| \le \lfloor p/2 \rfloor\}$.
   (d) $M_{\mathscr{P}}(i,j) \leftarrow \begin{cases} 1, & (i,j) \in \mathscr{P}, \\ 0, & \text{otherwise.} \end{cases}$
   (e) $x_{\text{patch}} \leftarrow \text{Patch}(x_{\text{adv}}, \mathscr{P})$.
   (f) $m_{\text{patch}} \leftarrow 0, \quad d_{\text{base}} \leftarrow \|x_s - x_{\text{adv}}\|_2$.
   (g) **for** iteration = 1 to $N_{\max}$:
      i. $d_{\text{local}} \leftarrow \text{Patch}(x_s, \mathscr{P}) - x_{\text{patch}}$.
      ii. $m_{\text{patch}} \leftarrow \mu \cdot m_{\text{patch}} + (1-\mu) \cdot d_{\text{local}}$.
      iii. $s_{\text{patch}} \leftarrow \eta \cdot \|x_s - x_{\text{adv}}\|_2$.
      iv. $G \leftarrow \text{GaussianWeight}((i_c, j_c), \sigma = p/3)$.
      v. $\Delta x \leftarrow s_{\text{patch}} \cdot m_{\text{patch}} \odot \big(G \odot \big(1 - (M_{\text{edge}} \odot M_{\mathscr{P}})\big)\big)$.
      vi. $x^+_{\text{patch}} \leftarrow x_{\text{patch}} + \Delta x$.
      vii. $x_{\text{temp}} \leftarrow x_{\text{adv}} - x_{\text{adv}} \odot M_{\mathscr{P}} + x^+_{\text{patch}} \odot M_{\mathscr{P}}$.
      viii. $d_{\text{new}} \leftarrow \|x_s - x_{\text{temp}}\|_2$.
      ix. **if** $d_{\text{new}} \ge 0.999 \cdot d_{\text{base}}$:
         A. **break**.
      x. **if** $\hat{y}(x_{\text{temp}}) = y_t$:
         A. $x_{\text{patch}} \leftarrow x^+_{\text{patch}}$.
         B. $x_{\text{adv}} \leftarrow \text{Replace}(x_{\text{adv}}, \mathscr{P}, x_{\text{patch}})$.
         C. $d_{\text{base}} \leftarrow d_{\text{new}}$.
         D. $n_{\text{break}} \leftarrow 0$.
      xi. **else**:
         A. $n_{\text{break}} \leftarrow n_{\text{break}} + 1$.
         B. **break**.

Return: x_{adv}.

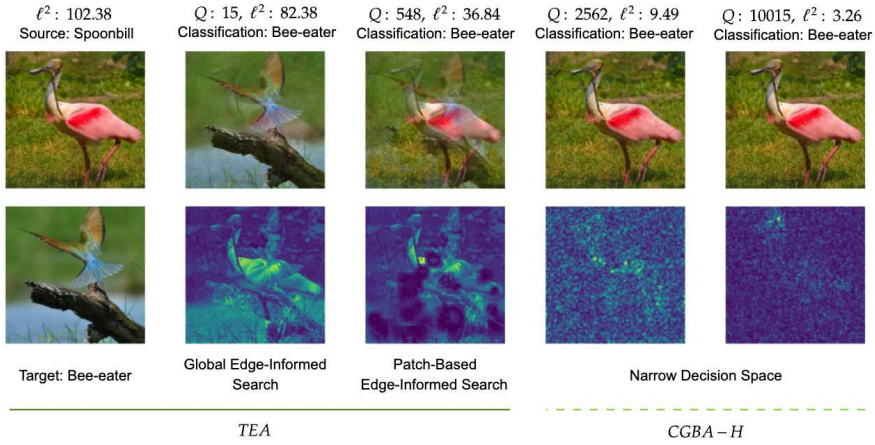


Figure IV.2: **Visualization of TEA on a source–target image pair.** The target image (initially classified as *Bee-eater*) is perturbed to resemble the source image (classified as *Spoonbill*), while preserving its original *Bee-eater* label. Global Edge-Informed Search efficiently applies edge-aware perturbations using only 15 queries to achieve a ≈20% reduction in distance to the source. Patch-Based Edge-Informed Search introduces localized, edge-aware modifications to small image regions, as seen in the hotspot of changes. Further refinement utilizing CGBA-H is illustrated in the narrow decision space.

IV.4 Results

In this section, we present our empirical evaluation benchmarking *TEA* against existing targeted hard-label attacks. In what follows, we describe our experimental setup and metrics, then detail the performance of each method under a range of query budgets. Our findings indicate that TEA achieves efficient distance reduction to the source image, particularly in the early query regime, before switching to the current state-of-the-art geometry-based attack CGBA-H for further refinement in a narrow decision space.

IV.4.1 Setup and Metrics

Computational Resources. The experiments were executed on a High-performance computing (HPC) Cluster. Each node on the cluster consisted of four NVIDIA GeForce GTX 1080 Ti GPUs (one GPU was allocated per source-target pair for a given attack).

Dataset and Image Pairs. We randomly sample 1000 source-target image pairs from the ImageNet ILSVRC2012 validation set, ensuring each pair contains images from distinct classes. All images are resized to $3 \times 224 \times 224$. Each pair (x_s, x_t) contains a source image x_s, correctly classified under its label, and a target image x_t, also correctly classified under a different label. The goal is to modify x_t to approach x_s under the ℓ_2 norm while keeping the prediction unchanged.

Target Models. We evaluate our approach on four well-known classifiers: ResNet50 and ResNet101 (CNNs of varying depth), VGG16 (a CNN composed of repeated convolutional

blocks), and ViT (a vision transformer that processes images as sequences of patches). These models represent a diverse set of architectures.

Compared Methods. We compare our method to four targeted hard-label attacks: HSJA, AHA, CGBA, and CGBA-H.For each method, the ℓ_2 distance to the source image is recorded after each set of queries, and along with the classification label of the perturbed target image.

Evaluation Metrics. Performance is quantified using three metrics. First, we compute the ℓ_2 distance from x_s to the adversarial example generated for each of the 1000 image pairs as queries progress. Second, the attack success rate (ASR) is defined as the fraction of image pairs for which an α% reduction in ℓ_2 distance between adversarial image and source image is achieved as compared to the target image and source image. Third, we also measure the performance of each method when there is a set fixed low-query budget by measuring the ASRs for all α when each method has used up 500 queries. Finally, we integrate the ℓ_2 distance versus queries curve to obtain the area under the curve (AUC), which provides an aggregated measure of how rapidly the distance is reduced.

Implementation Details. Our implementation employs a two-stage process. In the first stage, we perform our proposed methodology TEA, where edge-aware distortions are applied to the target image while preserving its target classification, allowing a rapid reduction in distance to the source image. In the second stage, once the perturbations have brought the image into a narrow decision space, the method switches to CGBA-H for further refinement (denoted as TEA* throughout the study). The last query performed with TEA is referred to as the *turning point* throughout the study. We switch to CGBA-H since it consistently performs better than other methods in a high query setting across different architectures.

Query	ResNet50					Query	ResNet101				
	HSJA	CGBA	CGBA-H	AHA	TEA		HSJA	CGBA	CGBA-H	AHA	TEA
100	92.797	88.857	83.104	80.438	**75.2155**	100	90.962	85.623	81.376	82.574	**74.5036**
200	88.715	87.165	79.348	75.344	**62.0338**	200	88.194	84.246	74.738	77.869	**61.3794**
300	88.604	86.236	74.168	71.054	**55.5589**	300	87.954	83.750	71.874	70.676	**54.9321**
400	88.199	85.579	71.479	67.091	**51.9530**	400	87.356	82.483	68.803	67.051	**51.3780**
500	87.724	84.897	68.671	64.172	**49.8352**	500	87.222	81.543	66.169	63.717	**49.3863**
602	87.203	83.937	66.774	61.168	**45.6000**	592	87.133	80.339	64.528	61.134	**45.1390**
Query	VGG16					Query	ViT				
	HSJA	CGBA	CGBA-H	AHA	TEA		HSJA	CGBA	CGBA-H	AHA	TEA
100	93.966	92.154	85.805	84.770	**77.2608**	100	80.454	76.885	73.074	68.863	**67.5654**
200	89.884	90.899	81.599	78.445	**63.6530**	200	79.981	74.702	69.511	65.501	**56.1255**
300	89.665	90.516	74.359	73.174	**56.9954**	300	79.606	74.139	63.655	63.951	**50.8059**
400	88.943	89.880	70.559	68.538	**53.4018**	400	79.599	72.839	61.300	59.857	**47.8666**
500	88.919	89.563	67.745	65.569	**51.3763**	500	78.761	71.291	58.598	56.572	**46.2528**
588	88.389	88.362	66.578	62.126	**47.4840**	554	78.626	69.954	57.710	54.494	**43.4220**

Table IV.1: Median ℓ_2 distances computed until the turning point across different architectures. Lower ℓ_2 values denote faster adversarial example generation.

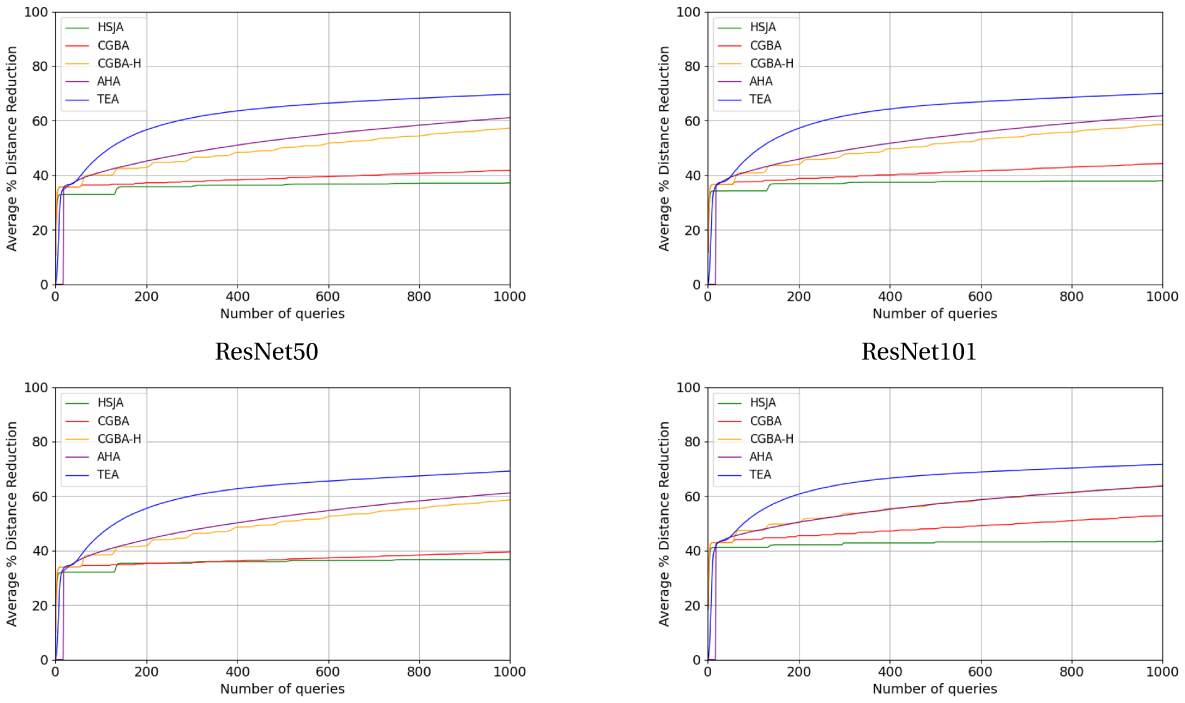


Figure IV.3: Average ℓ_2 distance reduction across different architectures in a low-query regime. Higher values indicate improved performance.

Average ℓ_2-Distance vs. Queries. Table IV.1 presents a comparative analysis of the median ℓ_2 distances achieved in the low-query regime—specifically, within the range defined by the average turning point computed over one thousand image pairs. This analysis underscores the rapid decrease in perturbation norm facilitated by TEA during the early query stages. Figure IV.3 extends this evaluation by depicting the average percentage reduction in ℓ_2 distance against queries used. In the plots, the initial inflection point marks TEA's transition from global exploration to patch-based refinement.

Attack Success Rate. In Figures IV.4 and IV.5, we show how many images reach at least 50% and 75% distance reduction over queries respectively. A sharper increase in this fraction signifies that more pairs experience substantial improvement more quickly. Again, once the turning point is reached between a pair, CGBA-H is implemented for further refinement.

Success at a Fixed Low-Query Budget. Figure IV.6 reports, at a low-query budget of 500 queries, the proportion of image pairs that reach or exceed various distance-reduction thresholds. TEA maintains a consistently higher proportion of successful pairs across nearly all thresholds, suggesting that its early-stage perturbations secure significant distance reductions more rapidly.

AUC Comparisons. We assess the efficiency of our aproach by computing the area under the median ℓ_2-distance vs queries curve (AUC). Table IV.2 presents the AUC values for the low-query regime, up to the turning point. Across all architectures, TEA consistently yields

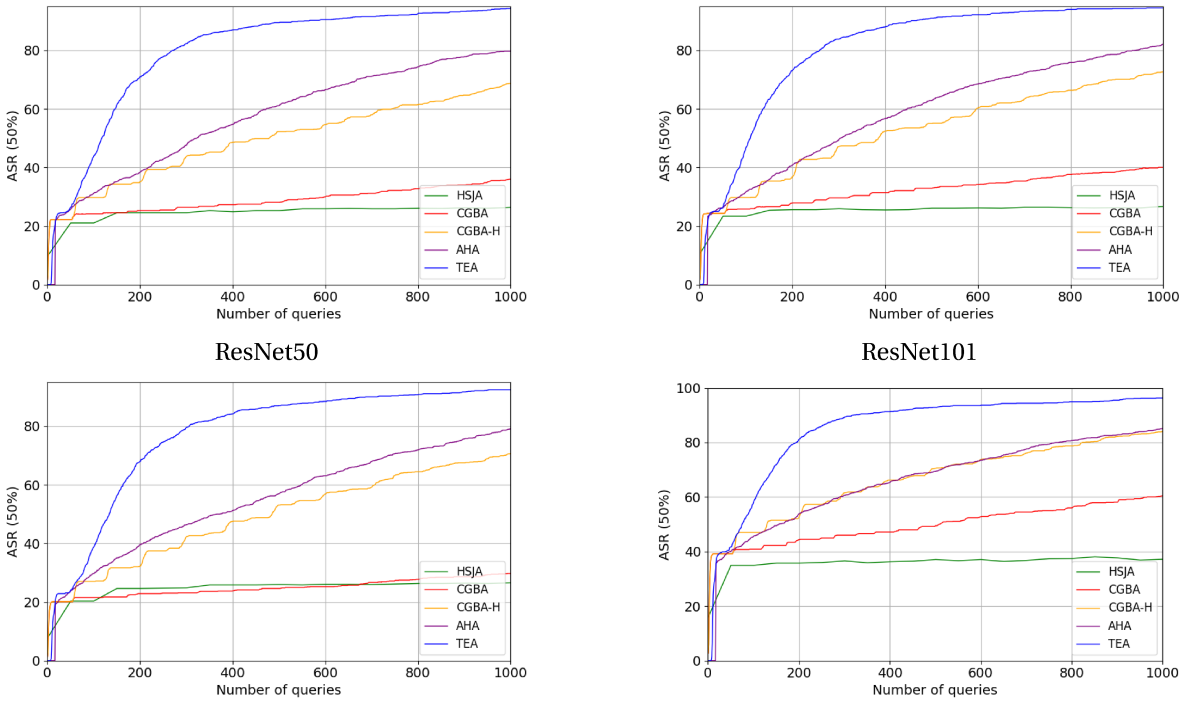


Figure IV.4: Comparison of ASR of 50% distance reduction. Higher values indicate that a higher proportion of images reach a distance reduction of 50% sooner.

lower AUC values.

Model	HSJA	CGBA	CGBA-H	AHA	TEA
ResNet50	53575.41	52130.74	45745.09	44332.55	**35133.14**
ResNet101	52269.70	49487.56	43865.80	43542.48	**34189.94**
VGG16	53053.06	53226.27	45261.02	44756.50	**35790.23**
ViT	44377.75	41097.13	36788.92	37356.25	**30337.35**

Table IV.2: Average AUC values computed until the turning point across different architectures. Lower AUC values denote more effective early-stage distance reduction.

Evaluation on an Adversarially Trained Model. In Figure IV.7, we present a comparative analyses of performance against an adversarial trained model (Resnet50 from MardyLab [243]) in the low-query region. We see that TEA consistently achieves more distortion, while also noting that all methods perform significantly worse compared to the standard Resnet50 model as seen in Figure IV.3.

Edge Ablation Study

To better understand the role of edge preservation in our method, we perform an ablation study by comparing three variants of TEA under identical settings. TEA perturbs non-edge pixels, preserving edge regions with a soft edge mask. INV-TEA inverts TEA's soft

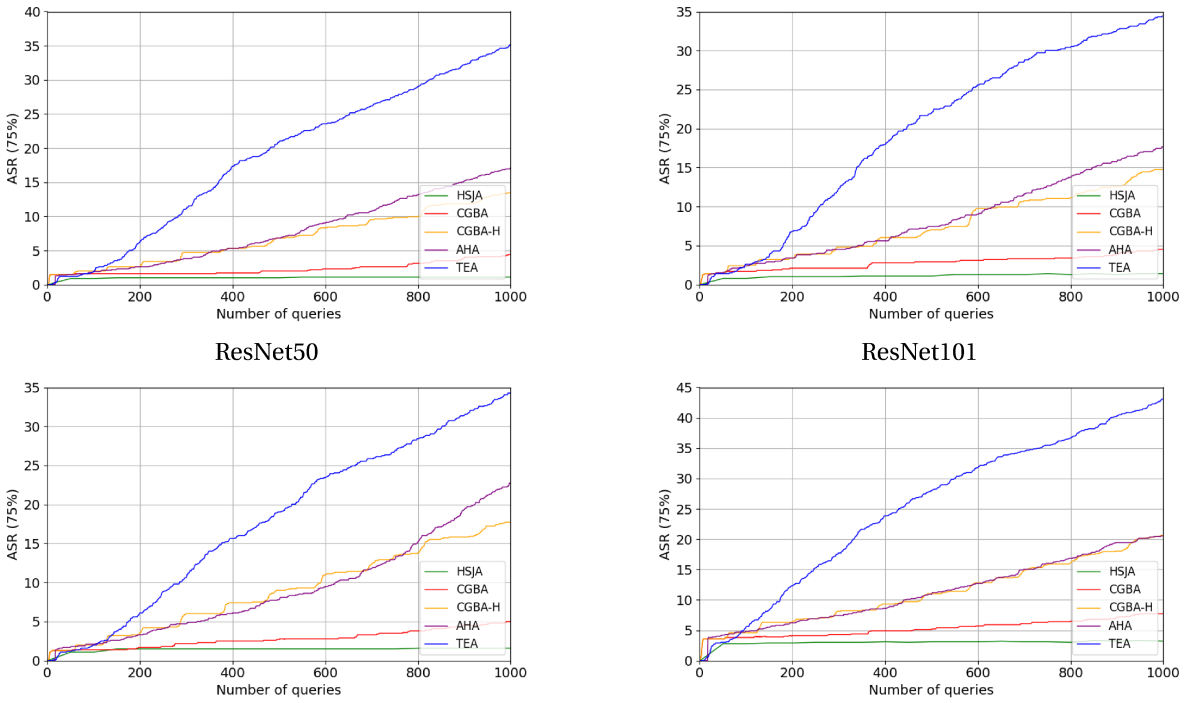


Figure IV.5: Comparison of ASR of 75% distance reduction. Higher values indicate that a higher proportion of images reach a distance reduction of 75% sooner.

edge mask to perturb primarily edge-like pixels. HALF-TEA perturbs non-edge and edge regions equally, using the same editable-pixel budget as TEA, and serves as a simple control baseline. Table IV.3 summarizes results in the low-query regime: median ℓ_2 distance, ASR at 50% distance reduction, and ASR at 75% distance reduction. Across all four architectures, TEA achieves the lowest median ℓ_2 and the highest success rates overall; HALF-TEA is consistently in between, and INV-TEA performs the worst, indicating that prioritizing non-edge regions proves crucial for early attack progress while preserving the semantic structure.

Randomness and Stability Across Patch Selection Our patch-based refinement uses random patch locations and sizes, introducing stochasticity. To quantify this effect, we executed five independent runs per model. For each run and query budget, we computed the average ℓ_2 distance across all pairs; we then summarized the five runs by reporting mean $\pm$ standard deviation (std) of the average ℓ_2 distance across all pairs. As seen in Table IV.4, across all models and budgets, variability is small relative to the mean. The standard deviation is typically below 0.3 (well under 1% of the mean).

Lower Resolution Performance We evaluate TEA, CGBA, CGBA-H and AHA at two lower input sizes, 128×128 and 64×64. The Tables IV.5 and IV.6 report the median ℓ_2 distance to the source image achieved at fixed query budgets; lower is better. VGG-16 and ViT required standard architectural adjustments at reduced dimensions.

As observable in the results, at lower input resolution sizes, each patch modification

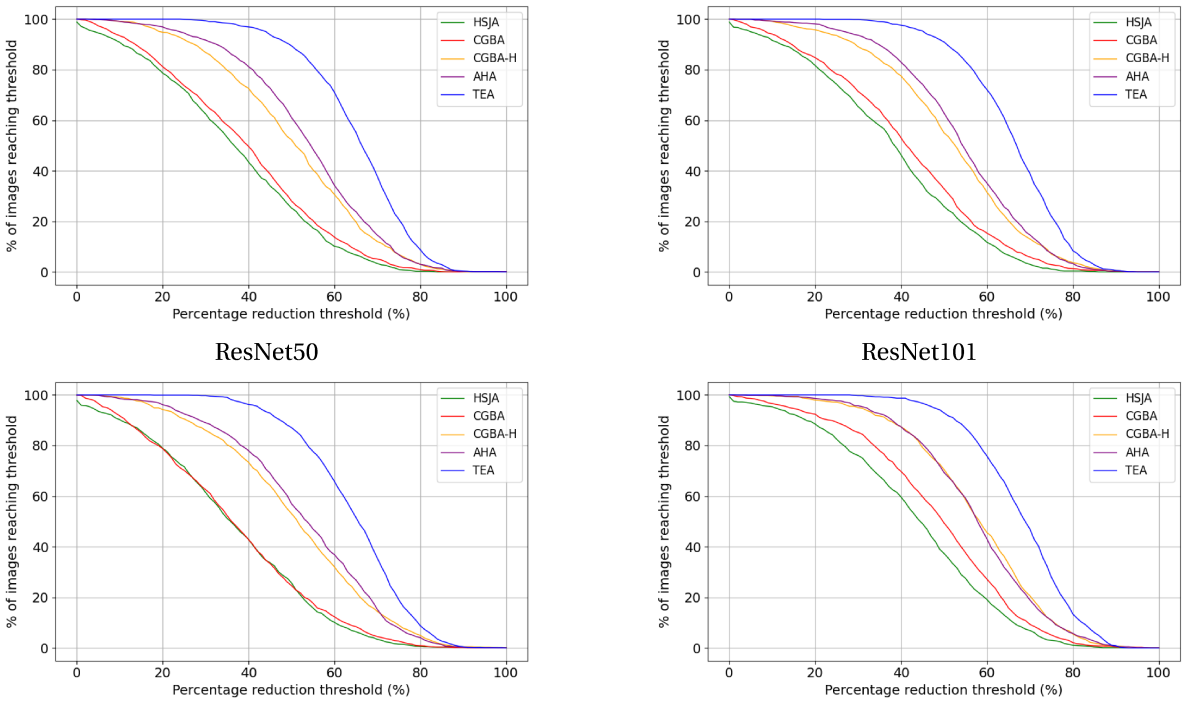


Figure IV.6: Cumulative distribution functions (CDFs) of distance reduction at 500 queries. Each curve represents the fraction of image pairs that achieve a given percentage reduction in the ℓ_2 distance, with higher values indicating a more effective reduction method.

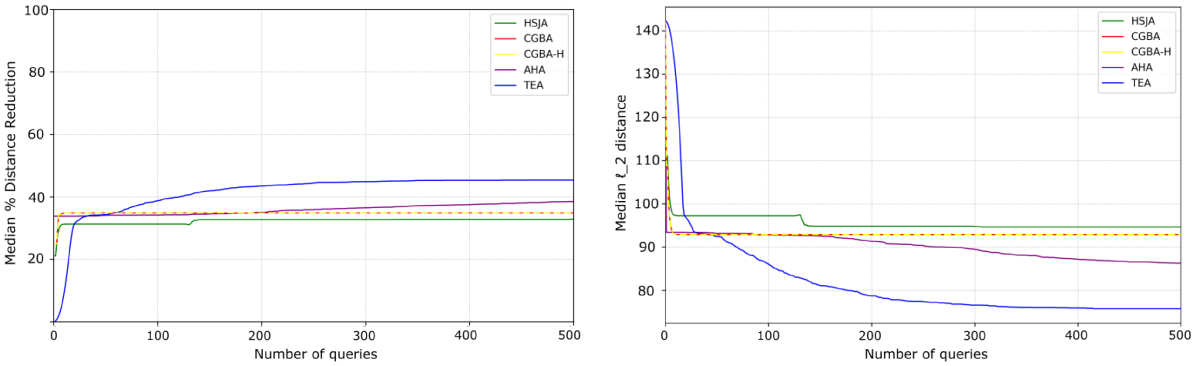


Figure IV.7: Comparison on an adversarially trained model. Left: median percentage decrease in ℓ_2 distance against number of queries, with higher values indicating a more effective reduction method. Right: median ℓ_2 distance against number of queries, with lower values indicating a more effective reduction method.

Query	ℓ_2			ASR@50%			ASR@75%		
	TEA	Half	Inv	TEA	Half	Inv	TEA	Half	Inv
ResNet-50									
100	**75.1377**	87.3465	87.5367	**44.0**	23.6	22.8	**2.0**	0.2	1.3
200	**61.8307**	73.1697	73.3559	**70.5**	49.7	45.2	**6.5**	0.7	2.6
300	**55.3574**	65.0474	66.5448	**81.8**	67.8	61.8	**10.7**	3.3	4.4
400	**51.8838**	60.0257	63.2027	**86.0**	77.6	69.0	**15.6**	5.9	6.1
500	**49.8877**	56.8181	61.6506	**87.5**	81.8	71.3	**19.2**	9.5	6.9
ResNet-101									
100	**74.4637**	86.1526	86.8610	**44.4**	24.1	23.2	**2.8**	0.3	1.0
200	**61.3517**	71.7322	72.6071	**72.4**	53.3	47.3	**6.9**	0.9	3.1
300	**54.9709**	63.5007	65.8087	**83.9**	69.7	62.0	**11.8**	3.2	4.7
400	**51.3084**	58.5084	62.4517	**87.5**	77.6	69.0	**17.1**	6.7	6.0
500	**49.3153**	55.3806	60.8018	**89.6**	82.6	72.2	**19.7**	9.5	6.5
VGG16									
100	**77.4206**	90.7514	89.8802	**38.1**	17.6	20.8	**2.0**	0.3	0.8
200	**63.7679**	75.3909	75.5661	**66.3**	45.0	40.8	**5.8**	0.7	2.3
300	**57.0986**	66.5727	68.5222	**78.7**	64.8	56.9	**10.8**	2.9	3.9
400	**53.4352**	61.0717	65.1434	**83.3**	74.9	62.9	**16.8**	7.4	5.0
500	**51.4759**	57.5518	63.4061	**84.7**	79.2	66.6	**19.5**	11.4	6.8
ViT									
100	**67.4874**	76.4322	77.6473	**57.8**	43.4	40.4	**5.2**	0.9	2.8
200	**56.2801**	63.8529	66.0835	**80.0**	69.8	60.8	**11.8**	3.6	6.7
300	**50.8057**	57.2091	60.8559	**88.1**	82.8	71.0	**17.5**	7.7	9.7
400	**47.9956**	53.3073	58.3993	**90.6**	88.1	74.5	**22.9**	12.7	12.1
500	**46.3913**	51.0218	57.2895	**92.1**	90.6	76.0	**26.1**	16.2	13.0

Table IV.3: Edge ablation in the low-query regime. Per architecture and budget we report median ℓ_2 (lower is better), ASR at 50%, and ASR at 75% (higher is better).

Budget	ResNet50	ResNet101	VGG16	ViT
50	88.1415 ± 0.0254	87.0992 ± 0.0637	90.6675 ± 0.0553	78.7152 ± 0.0277
100	75.2155 ± 0.2338	74.5036 ± 0.0676	77.2608 ± 0.1085	67.5654 ± 0.0652
150	67.3196 ± 0.2301	66.6368 ± 0.1022	69.1007 ± 0.1479	60.6342 ± 0.1187
200	62.0338 ± 0.2607	61.3794 ± 0.1318	63.6530 ± 0.1896	56.1255 ± 0.0829
250	58.2941 ± 0.2458	57.6515 ± 0.1356	59.7896 ± 0.2565	53.0414 ± 0.0537
300	55.5589 ± 0.1995	54.9321 ± 0.1397	56.9954 ± 0.2654	50.8059 ± 0.0627
400	51.9530 ± 0.1907	51.3780 ± 0.1564	53.4018 ± 0.2513	47.8666 ± 0.0654
500	49.8352 ± 0.1522	49.3863 ± 0.1306	51.3763 ± 0.2361	46.2528 ± 0.0899

Table IV.4: Randomness analysis across five runs: ℓ_2 (mean ± std).

Query	ResNet50				Query	ResNet101			
	AHA	CGBA	CGBA-H	TEA		AHA	CGBA	CGBA-H	TEA
50	**52.812610**	56.892204	56.892204	53.891792	50	**52.325765**	55.829930	55.830184	52.836270
100	49.792591	52.638543	52.610928	**44.656296**	100	49.256166	51.690833	52.297024	**43.908848**
150	47.580821	48.751302	48.900604	**40.200085**	150	47.165379	47.665892	48.747030	**39.356266**
200	45.816245	47.670539	47.708520	**37.588080**	200	45.418667	46.499661	48.731472	**36.744068**
250	44.242092	47.291906	47.416906	**35.637354**	250	43.904684	45.875023	48.286582	**34.803333**
300	42.705044	45.310037	45.394565	**34.083014**	300	43.015997	43.947135	46.251110	**33.364790**
350	41.704023	45.917060	46.053402	**32.915845**	350	42.285534	45.637712	48.112825	**32.363468**
400	40.606535	43.876574	44.083020	**31.711626**	400	41.279571	43.208419	47.001603	**31.200894**
Query	VGG16				Query	ViT			
	AHA	CGBA	CGBA-H	TEA		AHA	CGBA	CGBA-H	TEA
50	**53.742676**	57.595045	57.595044	54.653168	50	**49.717520**	61.999604	61.999605	51.518260
100	50.687319	53.132007	53.124912	**45.101308**	100	46.560914	56.845693	56.760291	**42.468795**
150	48.452131	48.769476	48.688020	**40.641963**	150	44.133746	51.938031	51.849316	**37.765469**
200	46.376375	47.648656	47.548296	**37.729822**	200	42.135314	52.195369	52.190025	**34.882641**
250	44.719990	46.665404	46.692002	**35.775914**	250	40.692779	50.410471	50.458582	**32.847132**
300	43.824315	44.433300	44.557708	**34.574623**	300	39.469095	47.456320	47.224463	**31.225525**
350	42.589694	45.370771	45.405220	**33.251792**	350	38.537397	49.233241	49.101166	**30.127403**
400	41.597263	43.446274	43.461510	**32.151386**	400	37.460420	47.532119	47.482755	**29.117229**

Table IV.5: Median ℓ_2 distances at 128×128.

Query	ResNet50				Query	ResNet101			
	AHA	CGBA	CGBA-H	TEA		AHA	CGBA	CGBA-H	TEA
50	**28.562640**	31.263057	31.263184	29.709433	50	**28.527201**	31.240249	31.240327	29.828940
100	26.583674	28.666067	28.734703	**25.884418**	100	26.545651	28.491136	28.543173	**26.137666**
150	25.094041	26.041276	26.107465	**23.768843**	150	25.101155	25.824065	25.971276	**23.934921**
200	23.449901	24.898503	25.231694	**21.971898**	200	23.850520	24.798062	25.058821	**22.132178**
250	22.191710	24.117243	24.478960	**20.671545**	250	22.805084	23.948980	24.090509	**20.863024**
Query	VGG16				Query	ViT			
	AHA	CGBA	CGBA-H	TEA		AHA	CGBA	CGBA-H	TEA
50	**28.318550**	30.844857	30.837613	29.530338	50	**30.081730**	32.758422	32.758422	31.046581
100	26.390916	28.302722	28.346767	**25.606639**	100	27.818672	29.806229	29.807765	**27.077744**
150	24.731502	25.578313	25.651390	**23.405260**	150	26.072833	26.316862	26.340219	**24.609781**
200	23.409042	24.719593	24.810609	**21.847981**	200	24.510567	24.761248	24.786895	**22.262164**
250	22.453166	23.539063	23.687779	**20.527478**	250	23.248782	23.717796	23.829428	**20.461638**

Table IV.6: Median ℓ_2 distances at 64×64.

distorts a larger fraction of the target image and thus contributes more to the overall ℓ_2-norm reduction. Consequently, maintaining the adversariality of the image becomes much harder, and TEA therefore terminates earlier than for higher resolutions.

Perceptual Metrics While adversarial examples are generated to trick ML models, it is worth considering their impact on human perception [65, 244]. We report SSIM [245] and FSIM [246] alongside distortion at a fixed budget of 400 queries. As shown in Table IV.7, TEA achieves high distortion while slightly improving FSIM. A small drop in SSIM is expected, since TEA's perturbations are more structured. Prior image quality assessment (IQA) studies [246, 247] indicate FSIM correlates more strongly with human perception than SSIM, which helps contextualise these differences.

	ResNet50				ResNet101			
Metric	CGBA	CGBA-H	AHA	TEA	CGBA	CGBA-H	AHA	TEA
ℓ_2	85.579	71.479	64.172	**51.9530**	82.483	68.803	67.051	**51.3780**
SSIM	**0.531765**	0.530459	0.231784	0.478806	0.525847	**0.528782**	0.223511	0.491950
FSIM	0.251337	0.252585	0.232117	**0.294672**	0.252746	0.251235	0.223831	**0.288153**
	VGG16				ViT			
Metric	CGBA	CGBA-H	AHA	TEA	CGBA	CGBA-H	AHA	TEA
ℓ_2	89.880	70.559	68.538	**53.4018**	72.839	61.300	59.857	**47.8666**
SSIM	**0.534783**	0.532189	0.231784	0.457879	0.580527	**0.582544**	0.201767	0.548110
FSIM	0.251129	0.251239	0.232117	**0.305919**	0.220927	0.220213	0.202171	**0.257571**

Table IV.7: SSIM and FSIM at 400 queries on 1000 source—target pairs.
Lower ℓ_2 indicates smaller perturbations; higher SSIM/FSIM indicates greater perceptual similarity to the source image.

High-Query Performance with CGBA-H refinement. Our hybrid TEA+CGBA-H strategy consistently matches or surpasses the state-of-the-art. Table IV.8 lists the median ℓ_2 dis-

tances upto 20000 queries. Note that since CGBA-H is randomness-dependent, it doesn't re-explore when trapped in a narrow decision space and its stability is subject to local decision boundary geometry. Figure IV.8 illustrates the median percentage reduction in the distance between the images.

Query	ResNet50					Query	ResNet101				
	HSJA	CGBA	CGBA-H	AHA	TEA*		HSJA	CGBA	CGBA-H	AHA	TEA*
1000	86.800	80.709	58.815	52.069	**41.108**	1000	86.162	76.005	55.824	51.642	**40.090**
2500	85.262	69.031	39.701	33.165	**29.747**	2500	85.314	63.477	38.043	33.259	**29.965**
5000	83.345	48.703	22.835	18.394	**18.338**	5000	84.202	41.437	22.470	**18.780**	18.932
10000	81.623	22.094	10.214	**8.597**	8.798	10000	82.443	17.623	10.341	**9.021**	9.864
15000	80.642	10.999	5.908	6.757	**5.429**	15000	81.873	8.936	6.341	6.937	**6.281**
20000	79.605	6.594	4.185	6.449	**4.011**	20000	80.886	5.689	**4.507**	6.644	4.560
Query	VGG16					Query	ViT				
	HSJA	CGBA	CGBA-H	AHA	TEA*		HSJA	CGBA	CGBA-H	AHA	TEA*
1000	87.451	85.772	56.388	52.361	**41.578**	1000	79.212	64.516	49.394	49.541	**38.482**
2500	86.346	74.963	35.686	30.429	**27.984**	2500	78.739	44.789	32.573	34.762	**28.134**
5000	85.219	53.869	19.753	16.076	**16.050**	5000	78.170	25.118	18.566	22.197	**18.507**
10000	85.219	20.792	8.558	**7.456**	7.626	10000	77.786	10.319	**9.299**	11.267	9.416
15000	85.219	9.455	5.306	6.238	**4.979**	15000	77.087	**5.920**	5.987	8.180	6.301
20000	85.219	5.687	3.985	6.042	**3.823**	20000	76.727	**4.258**	4.635	7.404	4.955

Table IV.8: Median ℓ_2 distances across different architectures.
Here, TEA* indicates using TEA until the turning point, and refining further with CGBA-H.

Source-Target Similarity Analysis. To examine how the performance of TEA depends on the similarity between source and target images, we group the 1000 source-image pairs by structural similarity. For each pair (x_s, x_t) we compute the SSIM score and sort all pairs by SSIM, partitioning them into ten equally sized bins. Within each bin, and model, we measure the percentage ℓ_2 reduction at a fixed budget of 400 queries. We then average this quantity over all pairs in the same bin and plot the resulting curves (Figure IV.9). We see that all methods have relatively identical performance variability, with improved performance across structurally similar source-target image pairs. This indicates that structural similarity does not favor TEA over the other methods.

Edge–Density Analysis. To assess how TEA behaves under different levels of structural detail in the source and target images, we perform an edge–density based stratification of the source–target pairs. For each image, we convert it to grayscale, compute Sobel gradients, and form the normalized gradient magnitude $\tilde{g}(p) \in [0,1]$ for each pixel p. Pixels with $\tilde{g}(p) > 0.2$ are treated as edge pixels, and the edge density $d(x)$ is defined as the fraction of pixels classified as edges. We collect $d(x_s)$ and $d(x_t)$ for all source-target image pairs and set a global dense/sparse threshold τ_{dense} to be the median density over all 2000 images, yielding $\tau_{\text{dense}} \approx 0.110242$ in our case. Images with $d(x) > \tau_{\text{dense}}$ are labeled *dense*, and the rest *sparse*. Each pair (x_s, x_t) is thus assigned to one of four edge–pattern regimes: dense–sparse, dense–dense, sparse–sparse, or sparse–dense. For each regime, and model, we report the average percentage ℓ_2 reduction (in %) at a fixed budget of 400 queries. The resulting curves in Figure IV.10 show that performance variability is similar across methods,

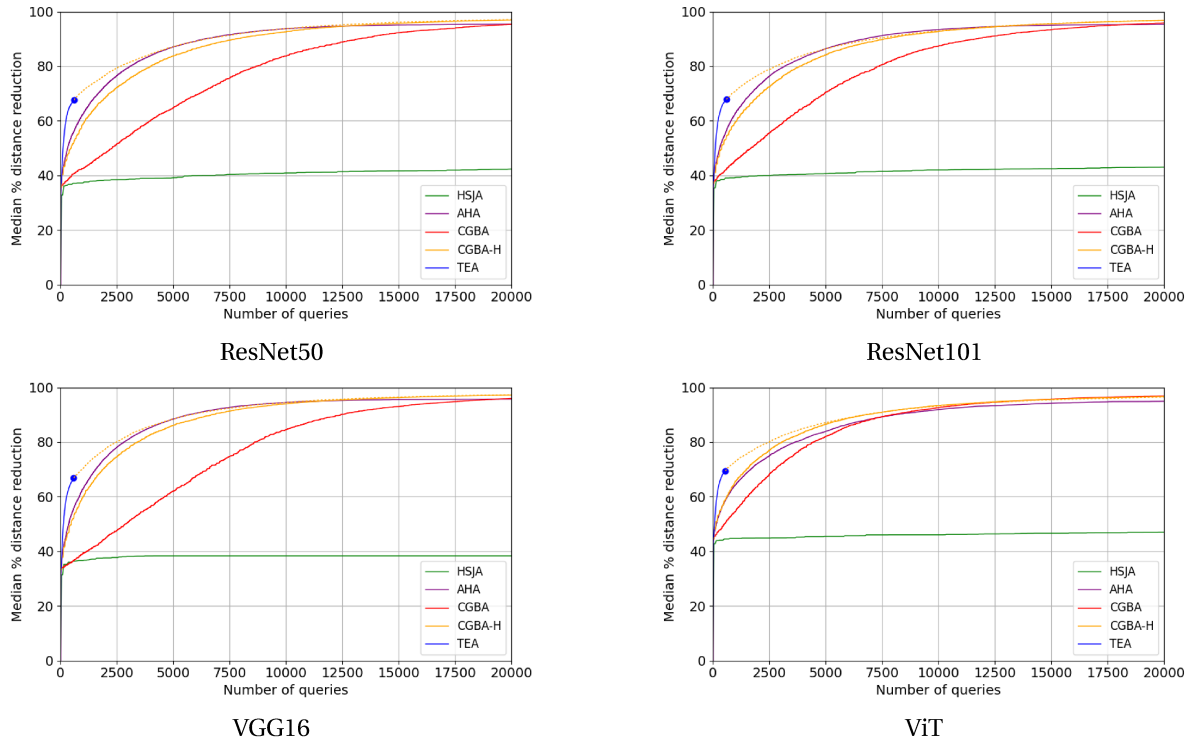


Figure IV.8: Comparison of median percentage decrease in ℓ_2 distance across different architectures — higher values indicate a more effective reduction method.

with higher success when moving towards a sparse source image than a dense one, and that TEA maintains a consistent advantage across all regimes.

Zero-Shot CLIP. To assess whether TEA also transfers to more modern vision-language models, we additionally evaluate it against a zero-shot CLIP classifier on the ImageNet validation set. We use the ViT-B/32 variant of CLIP [248, 242] in its standard zero-shot configuration. Following the usual protocol, we construct a linear zero-shot head by encoding multiple natural-language templates for each ImageNet class (e.g., “a photo of a {}.”, “a close-up photo of a {}.”) with the CLIP text encoder, normalizing and averaging the resulting embeddings to obtain a single prototype per class. Stacking these prototypes yields a fixed weight matrix, and logits are obtained by taking scaled dot products between normalized image features and these class prototypes. Our implementation relies on the official CLIP PyTorch code-base and its recommended preprocessing.

In Figure IV.11, we show the average percentage decrease in ℓ_2 distance and the average ℓ_2 distance as a function of the query budget. TEA retains a clear advantage, consistently achieving larger perturbation reductions than the baselines within the same limited query budget, indicating that its strategy remains effective even for modern zero-shot vision-language models.

Other Datasets. To further assess the robustness and generality of TEA beyond ImageNet, we additionally evaluate all tested methods on two standard classification benchmark

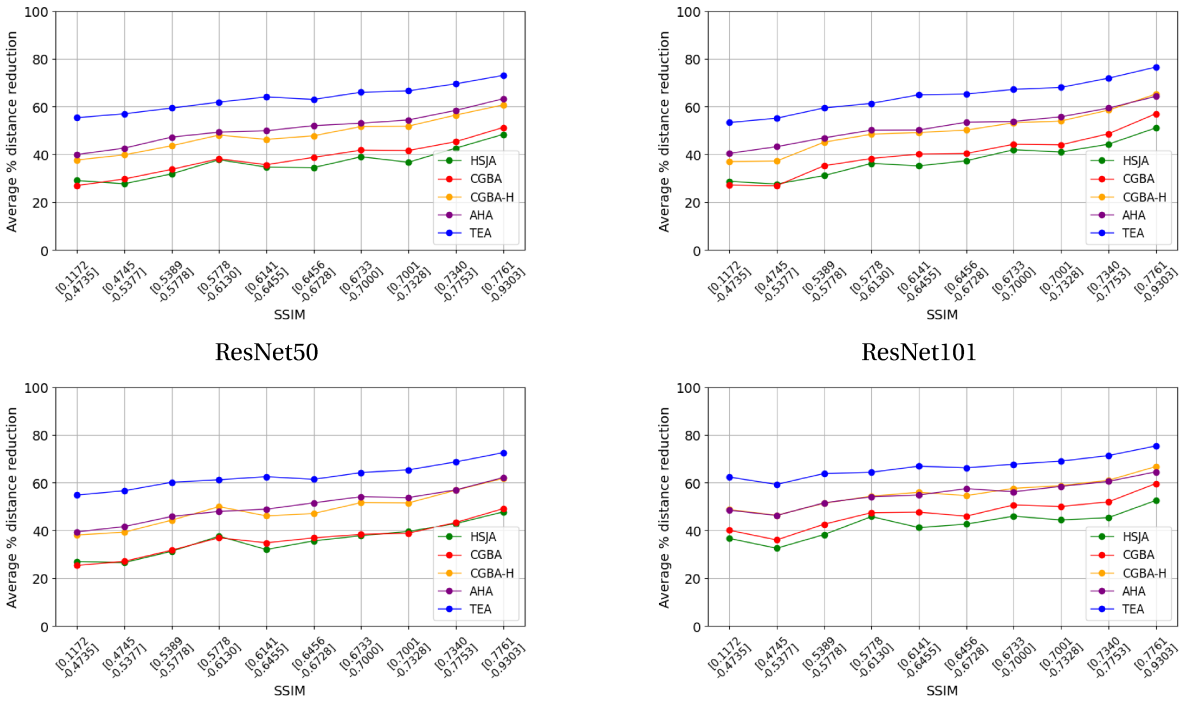


Figure IV.9: Effect of source–target structural similarity on attack performance. We group image pairs into ten bins by SSIM and report the average percentage decrease in ℓ_2 distance after 400 queries. All methods benefit similarly from higher structural similarity.

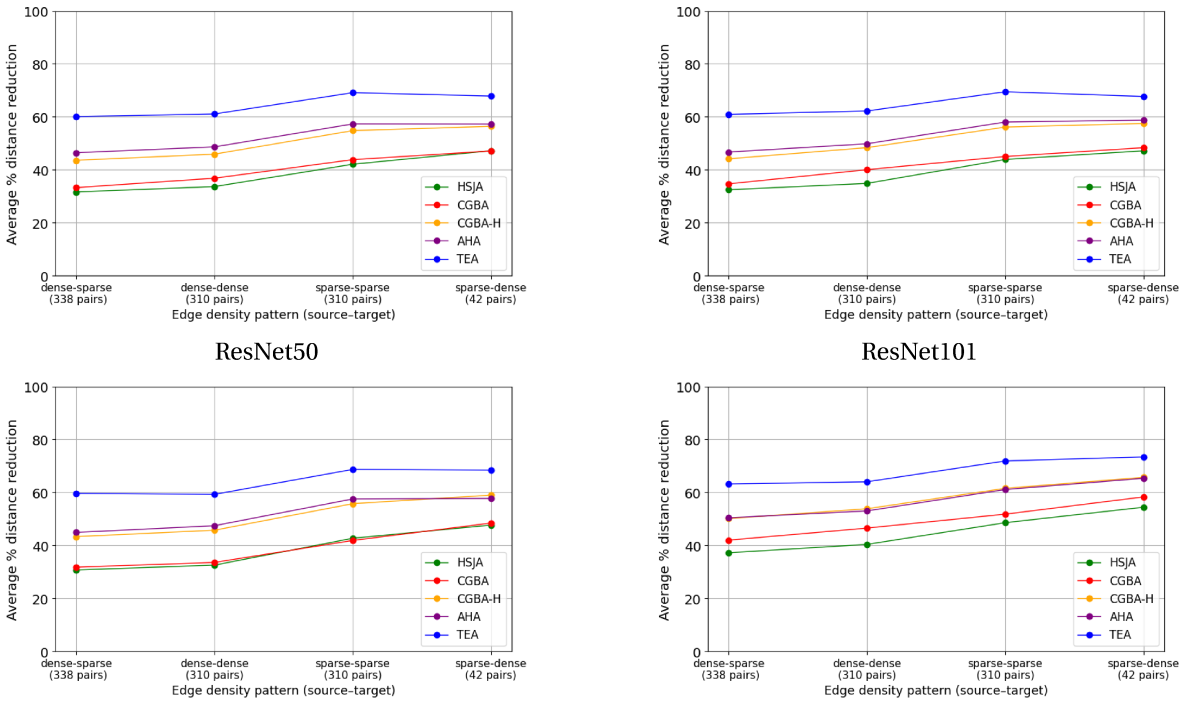


Figure IV.10: Effect of source–target edge density on attack performance. Each image is classified as *dense* or *sparse* based on a Sobel edge–density threshold, inducing four edge–pattern regimes (dense–dense, dense–sparse, sparse–dense, sparse–sparse). For each regime, and model, we report the average percentage decrease in ℓ_2 distance after 400 queries, illustrating how performance varies with the edge richness of the source–target pairs.

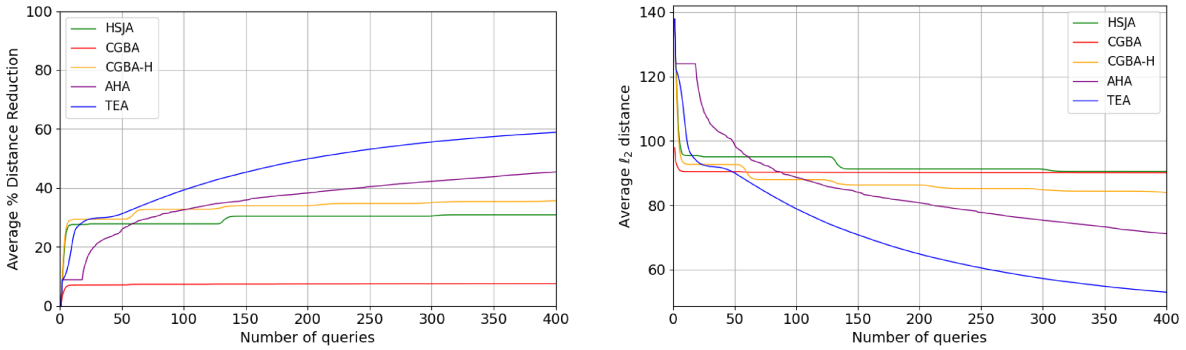


Figure IV.11: Comparison on a zero-shot CLIP (ViT-B/32) classifier on ImageNet. Left: average percentage decrease in ℓ_2 distance against number of queries, with higher values indicating a more effective reduction method. Right: average ℓ_2 distance against number of queries, with lower values indicating a more effective reduction method.

datasets: CIFAR-100 [249] and the Intel Image Classification dataset [250]. CIFAR-100 contains 100 object categories with lower-resolution images while the Intel dataset consists of six scene categories (e.g., buildings, forest, sea). In both cases, we again attack the four models, ResNet-50, ResNet-101, VGG-16, and ViT, and measure performance in terms of the relative ℓ_2-distance reduction between the source and target images. Figures IV.12 and IV.13 report the average ℓ_2-distance reduction at a fixed low-query budget of 400 across all four models. Consistent with our results on the ImageNet dataset, TEA achieves larger perturbation reductions on both datasets, indicating that its edge-aware strategy remains effective across diverse data distributions and resolutions.

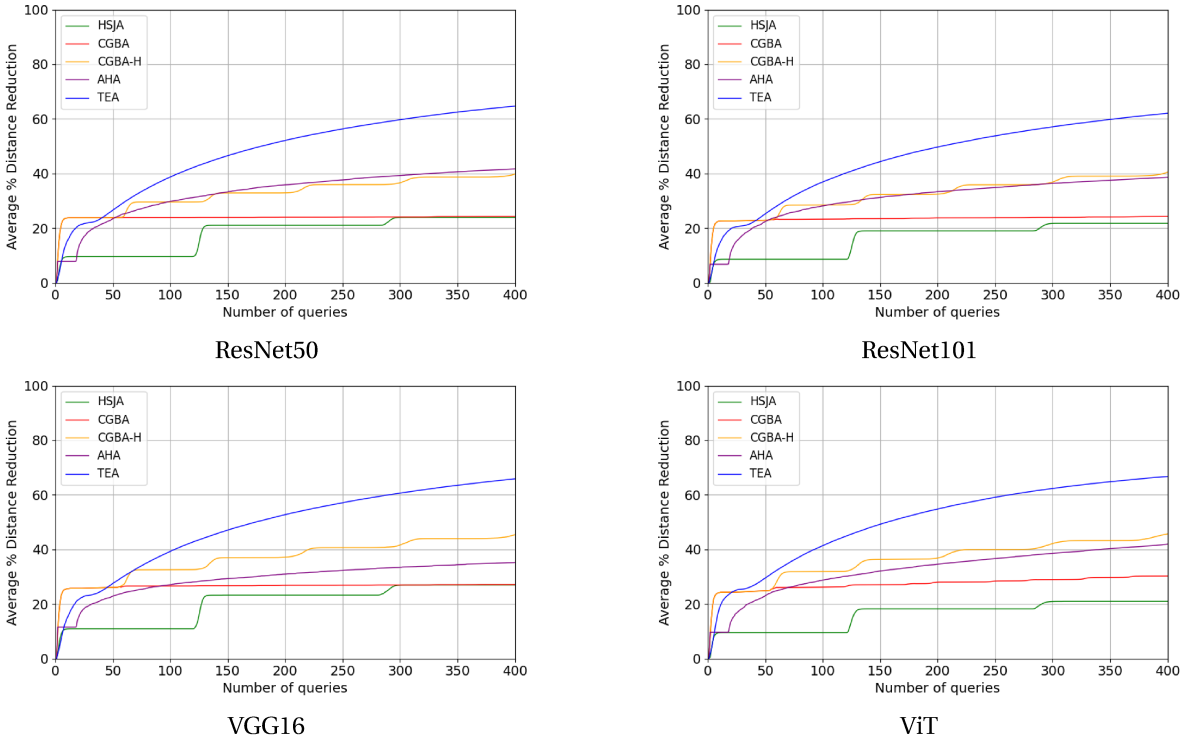


Figure IV.12: Average ℓ_2 distance reduction across different architectures in a low-query regime on the CIFAR-100 dataset.

IV.5 Conclusion

In this work, we introduced *TEA*, a targeted, hard-label, black-box adversarial attack that leverages edge information from a target image to efficiently produce adversarial examples perceptually closer to a source image in low-query settings. TEA initially employs a global search that preserves prominent edge structures across the target image, subsequently refining the perturbations via patch-wise updates. Empirical results demonstrate that TEA significantly reduces the ℓ_2 distance between adversarial and source images, requiring over *70% fewer queries* compared to current state-of-the-art methods. In addition, it also achieves reduced AUC and high ASR scores, indicating a consistently rapid reduction in distance to the source image across different models.

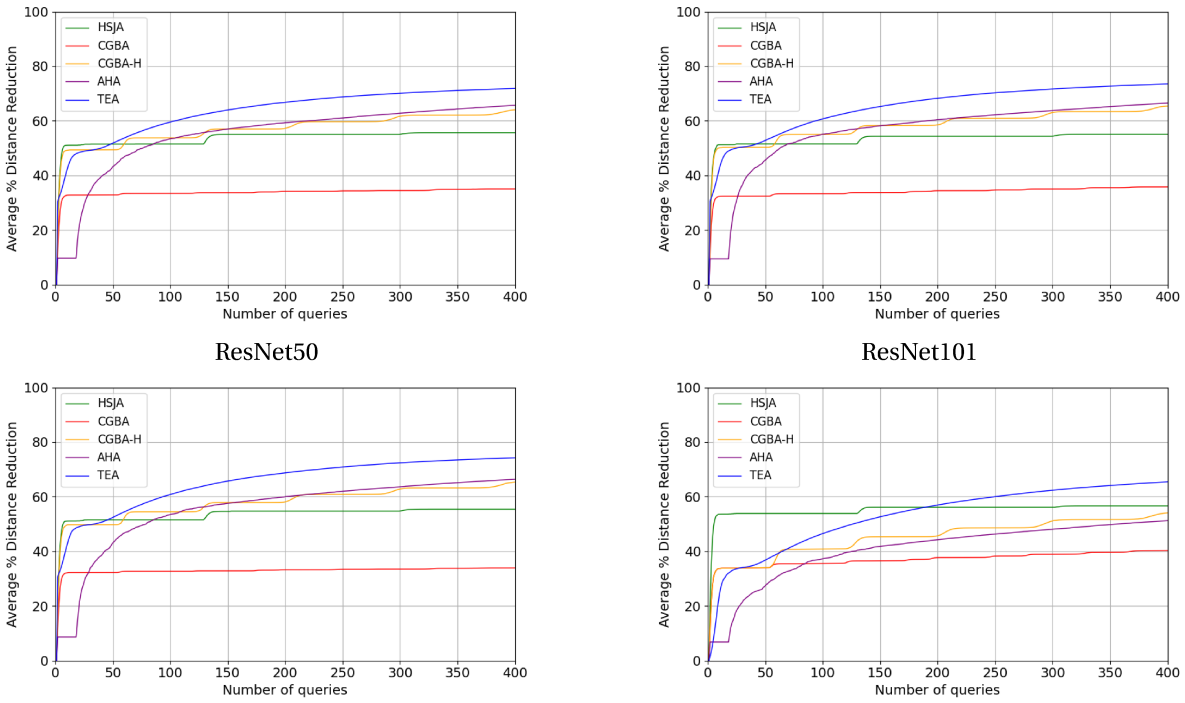


Figure IV.13: Average ℓ_2 distance reduction across different architectures in a low-query regime on the Intel Image Classification dataset.

Limitations. TEA is specifically designed for the targeted, hard-label, low-query black-box setting, and its main advantage lies in rapidly improving performance under tight query budgets. When TEA is used merely as an initialization and followed by stronger high-query attacks in regimes where many queries are available, we observe that the final performance is comparable to that of the underlying state-of-the-art methods, rather than strictly better. In addition, while TEA preserves global edge structure and achieves low ℓ_2 distances, the patch-wise updates can still introduce localized artifacts that are visible to human observers. This limitation is shared with other norm-bounded attacks in general, but in TEA the patch-based nature of the perturbations can make these local changes particularly noticeable.

Future Work. Several extensions could enhance TEA: (a) substituting edge information with other features such as textures, color distributions, or high-frequency components; (b) training a surrogate model based on query data already collected to guide subsequent perturbations and (c) developing defense mechanisms capable of mitigating attacks based on structure-preserving perturbations.

Code Availability

Our code is available at the following URL: https://github.com/mdppml/TEA [251].

Acknowledgements

This research was supported by the German Federal Ministry of Education and Research (BMBF) under project number 01ZZ2010 and partially funded through grant 01ZZ2316D (PrivateAIM). The authors acknowledge the usage of the Training Center for Machine Learning (TCML) cluster at the University of Tübingen.

Code of Ethics and Broader Impact Statement

We evaluate TEA exclusively on the publicly available ImageNet-1K validation set, which is distributed for non-commercial research and educational use under the ImageNet access agreement. The dataset contains no personally identifiable information, and was used in accordance with the license terms. As TEA reduces the number of required queries for targeted hard-label attacks in a low-query setting, it could be leveraged to more rapidly craft adversarial inputs against commercial or safety-critical vision systems. We encourage practitioners to adopt defense mechanisms that go beyond detecting frequency-based perturbations, explicitly incorporating checks for edge-informed distortions, and defenses attuned to other structural features, to guard against similar attacks.

V Dynamic k-anonymity for Electronic Health Records: A Topological Framework

Arjhun Swaminathan **Mete Akgün**

Abstract

With the rapid digitization of Electronic Health Records (EHRs), fast and adaptive data anonymization methods have become increasingly important. While tools from topological data analysis (TDA) have been proposed to anonymize static datasets—allowing the creation of multiple generalizations for different anonymization needs from a single computation—the application to dynamic datasets remains unexplored. To address this, our work adapts existing methodologies to the dynamic setting. We develop an improved version of weighted persistence barcodes that track higher-dimensional holes in data, allowing us to edit persistence information on the fly. Additionally, we introduce filtration trimming, a novel technique designed to update persistence data quickly with minimal computing effort when data is added. Our work represents a significant advancement in healthcare data privacy, offering a refined approach to protecting highly sensitive and evolving patient data through dynamic k-anonymity.

V.1 Introduction

In the era of digital transformation, the healthcare sector has accumulated vast amounts of patient data through EHRs, encompassing demographics, medications, diagnoses, progress notes, and medical history. By 2019, 96% of general practitioners in the EU had adopted EHRs[*]. This shift coincides with strict privacy regulations like the GDPR[*], which mandate de-identification or anonymization of sensitive data before processing.

Healthcare providers face the challenge of utilizing this rich data for improved healthcare while ensuring patient privacy. Anonymizing EHRs before public or third-party access enables researchers to conduct large-scale studies, aid in public health monitoring and develop crucial medical treatments without compromising privacy. The primary technique for anonymizing EHRs has been k-anonymity [252, 253], but it is vulnerable to attacks like background knowledge attacks [69, 254], attribute disclosure attacks [70], membership inference attacks, and linkage attacks [255], as highlighted by Sweeney's analysis of the 1990 U.S. Census data [29]. To address these limitations, l-diversity [69, 256] and t-closeness [70] were developed to enhance k-anonymity. Despite its computational challenges and NP-hardness [257], k-anonymity remains widely used [258, 259, 260, 261, 262, 263, 264].

Differential privacy, another key privacy technique, adds noise to data to ensure that the omission of a single entry does not significantly alter analysis results [38]. It protects against linkage and dataset reconstruction attacks [51] and is used in various fields, such as privacy-preserving data sharing in GWAS studies [265, 137] and health recommender systems [266]. However, differential privacy is analysis-sensitive — the amount of noise added to the data

[*] eHealth adoption in primary healthcare in the EU is on the rise, European Commission

[*] General Data Protection Regulation ((EU) 2016/679)

depends on the type of queries or computations executed on it. k-anonymity often emerges as a more general alternative for various applications.

Amongst numerous techniques that achieve k-anonymity, [74] stands out, being the only topologically informed method. It is particularly notable for its unique ability to compute multiple generalizations for different k values without the need to rerun the algorithm for each generalization or each k. This in turn allows for stronger privacy measures such as l-diversity and t-closeness to be applied. This remarkable efficiency is achieved through a single computation of a weighted persistence barcode, using scalable algorithms [267, 268, 269]. However, the method has a critical limitation - it is limited to static datasets. This is a significant drawback in time sensitive fields such as healthcare where data is frequently republished, often with alterations. When handling the dynamic nature of data in such cases, [74] would require a complete re-computation of the weighted persistence barcode for the dataset, leading to significant computational strain.

V.1.1 Our Contribution and Paper Structure

In our work, we address the major limitation identified in [74] by developing a method that removes the need for complete re-computation of persistent homology when the data changes. We introduce the concept of *hole-weighted persistence barcodes*, a new approach to track the evolution of higher-dimensional holes in the data. This innovation allows us to manage data removal, addition, and updating in dynamic datasets efficiently.

For data removal and updating, our algorithm changes the existing hole-weighted persistence barcodes directly, thus avoiding the need to recalculate persistent homology with every change in the dataset. When new data is added, we use *filtration trimming*, a novel technique which significantly reduces the number of persistent homology re-computations needed. Our proof of concept experimental evaluation using [268] on simulated data exhibits a significant reduction in the number of re-computations needed for data additions compared to [74].

We chose to evaluate our method against Speranzon et al. [74] because of its pioneering role in topology-informed k-anonymity and its unique advantages. To the best of our knowledge, our work is the first topologically-informed anonymization method for dynamic data.

The remainder of the paper is structured as follows. In Section V.2, we review k-anonymity and other preliminaries. In Section V.3, we discuss related work, with a focus on [74]. In Section V.4, we propose our methodology, proposing hole-weighted persistence barcodes, and inductively describing handling of addition, deletion and updating of data. We conclude in Section V.5.

V.2 Preliminaries

V.2.1 k-anonymity

When employing k-anonymity, one aims to work with datasets whose attributes are categorized into identifiers, quasi-identifiers, and sensitive data, as described in Table V.1.

- **Identifiers:** These are attributes like name or Social Security number that can directly identify an individual. These are typically de-identified before data publishing to protect the privacy of individuals.
- **Quasi-identifiers:** These are attributes $A_1, A_2, \ldots, A_M$ such as age, gender, or zip code that cannot individually identify an individual but can do so when combined together.
- **Sensitive data:** These are attributes like medical conditions or salary, which are sensitive information one intends to keep private.

Name	Admission Date	Age	Blood Pressure	Diagnosis
Maria	02.10.2022	23	121 mm Hg	Anxiety
Priya	05.10.2022	44	97 mm Hg	UTI
Ahmed	03.01.2023	21	95 mm Hg	–
Aiden	05.02.2023	41	100 mm Hg	Asthma
Identifiers		Quasi-identifiers		Sensitive Data

Table V.1: Table illustrating the classification of data attributes into identifiers (to be de-identified prior to publication), quasi-identifiers, and sensitive data.

T		
02.10.2022	23	121
05.10.2022	44	97
03.01.2023	21	95
05.02.2023	41	100

$\bar{T}$		
2022	$20-50$	$95-125$
2022	$20-50$	$95-125$
2023	$*1$	$70-100$
2023	$*1$	$70-100$

$\bar{T}^*$		
$****$	$**$	$***$
$****$	$**$	$**$
$****$	$**$	$**$
$****$	$**$	$***$

Table V.2: Here, T represents the original data table, $\bar{T}$ is the 2-anonymous generalization of T, while $\bar{T}^*$ is an over-generalization of T and $\bar{T}$.

The table of quasi-identifiers is denoted by $T(A_1, A_2, \ldots, A_M)$ consisting of N rows, where A_i are attributes that can take numeric or categorical value. Then the i^{th} sample's j^{th} quasi-identifier data is denoted as T_{ij}. Another table $\bar{T} = (\bar{A}_1, \bar{A}_2, \ldots, \bar{A}_M)$ consisting of N rows is said to be a generalization $\bar{T}$ of T if for all i, j, $T_{ij} \subset \bar{T}_{ij}$.

Definition 13 (k-anonymity [252])**.** Consider a generalization $\bar{T}$ of T. $\bar{T}$ is said to have the k-anonymity property if every row in $\bar{T}$ appears at least k times in $\bar{T}$.

An illustration of k-anonymity is described in Table V.2. The objective of k-anonymity is to transform a given table T with quasi-identifiers into a generalized table that satisfies the k-anonymity property. There exist multiple generalizations that achieve k-anonymity, but we aim to minimize data loss, so the anonymized data is private, but also usable for further analysis. A common first step used to achieve k-anonymity, is mapping data to a metric space, and forming a point cloud with the data. This allows for geometric structures to be defined to create generalizations.

Definition 14 (Embedded Point Cloud [74])**.** Consider a table of quasi-identifiers T containing M attributes and N samples. An embedded point cloud $P_T \in \mathbb{R}^M$ is formed by treating each row of T as a point in $\mathbb{R}^M$, such that $P_T = \{p_1, p_2, \ldots, p_N\}$, where p_i corresponds to the i^{th} row of T. This data is generally standardized along every attribute.

V.3 Related Work

V.3.1 k-anonymity in practice

The implementation of k-anonymity involves techniques like data generalization, fragmentation, and microaggregation, each with its own distinct advantages. Generalization can use predefined intervals (hierarchy-based) such as [252, 270] or runtime-determined intervals (recoding-based) like Mondrian [271, 74], and our approach. Data fragmentation separates quasi-identifier data from sensitive information [256], while microaggregation forms clusters with a minimum of k-similar records [272, 273, 274, 264].

k-anonymity is applied in various fields. During COVID-19, contact-tracing apps used Bluetooth Low Energy (BLE) to maintain user privacy [258, 259, 260, 261]. These contract tracing applications anonymize data via encrypted beacon IDs or hash-based representations of user contacts, ensuring users were indistinguishable from at least *k-1* others. In location-based services (LBS), spatial cloaking [275, 276, 277] reduces location accuracy to ensure each request is indistinguishable from *k-1* others, using trusted anonymizers. Other applications include road networks [278], autonomous vehicles [262], crowd-sensing [279], and data publishing for research [263].

The utility of k-anonymity to safeguard dynamic datasets has been explored in various studies, each offering vastly different methodologies to ours limiting direct comparability. The seminal work by [280] established a method for efficiently updating k-anonymized frameworks with the addition of new data entries, although it does not address deletions or modifications. In contrast, m-invariance [281] utilizes counterfeit data to preserve privacy when data is either added or removed. Alternatively, [282] employs micro-aggregation to manage changes including additions, deletions, and updates, which is fundamentally different from generalization-based approaches.

Unique in its application, [74] employs topological information to achieve k-anonymity. This method not only supports the generation of multiple generalizations but also caters to varied k-anonymity requirements in a single computation.

V.3.2 Topology-based k-anonymity

We now briefly discuss the theoretical framework developed by [74], which enables the application of persistent homology to the k-anonymity problem to derive an optimal generalization for a database with minimal loss. We will discuss application to numerical data, as in the foundational work, and this can be extended to categorical data using *generalization trees*.

Definition 15 (Anonymity Complex [74])**.** Given an embedded point cloud P_T derived from a table T of quasi-identifiers, an anonymity complex $\mathscr{C}^{\epsilon}(P_T)$ is the Cêch complex [283] defined over P_T comprising of simplices [283], each having vertices that correspond to points in P_T. A k-simplex is included in $\mathscr{C}^{\epsilon}(P_T)$ iff the ϵ-ball neighborhoods around its $k+1$ vertices share at least one common point.

Definition 16 (k-anonymity Complex [74])**.** An anonymity complex $\mathscr{C}^{\epsilon}(P_T)$ is termed a k-anonymity complex if the following conditions are met:

1. $\bigcup S_l = P_T$, where S_l is an l-simplex with $l \geq k-1$.
2. $S_{l_1} \cap S_{l_2} = \emptyset$ for all $l_1 \neq l_2$.
3. For every p_i in P_T, p_i belongs to some S_l.

The smallest ϵ for which $\mathscr{C}^{\epsilon}(P_T)$ satisfies these conditions is termed the global generalization strategy for parameter k, and is denoted by ϵ_k. Other ϵ that satisfies these conditions are other generalizations. We call the sequence of subcomplexes $\phi \subseteq \mathscr{C}^{\epsilon_1}(P_T) \subseteq \ldots \subseteq \mathscr{C}^{\epsilon_n}(P_T)$, where $\epsilon_i < \epsilon_j$ if $i < j$, a *filtration*.

Notably, it was demonstrated in [74] that an anonymity complex $\mathscr{C}^{\epsilon_k}(P_T)$ is a k-anonymity complex iff it has trivial homology groups $H_n(\mathscr{C}^{\epsilon_k}(P_T)) = 0$ for all $n > 0$.

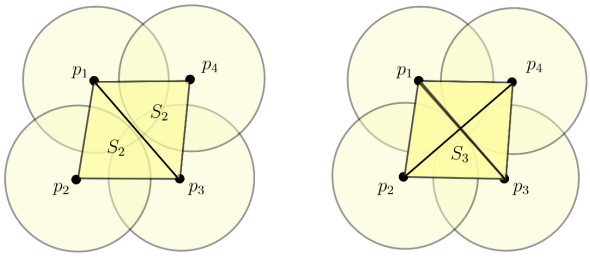


Figure V.1: A comparative visualization of two filtration levels, ϵ_1 and ϵ_2, showcasing the formation of a 3-simplex, S_3, at $\epsilon_2 > \epsilon_1$. While the points do not form a 3-anonymity complex at ϵ_1, they successfully form a 4-anonymity complex at ϵ_2.

By mapping T into an embedded point cloud P_T and constructing a Čech filtration as illustrated in Figure V.1, we aim to find the smallest ϵ-value, ϵ_k, that forms a k-anonymity

complex $\mathscr{C}^{\epsilon_k}(P_T)$. Simplices S_l in the complex form when their $l+1$ vertices have overlapping ϵ_k-ball neighborhoods. However, these simplices do not represent the anonymized equivalence classes. Instead, equivalence classes are defined by computing the circumcenter and circumcircle radius r of each simplex. The equivalence class for a simplex is the higher-dimensional hypercube centered at the circumcenter with side length $2r$. This method generalizes the data, distinguishing equivalence classes from the simplices.

Weighted Persistence Barcodes

To identify the optimal generalization strategy ϵ_k for a specific k-anonymity problem, [74] introduced weighted persistence barcodes, as shown in Figure V.2. We replicated this using *PHAT* [268]. Unlike traditional persistence barcodes, the weighted versions feature bars with varying thickness, determined by the number of vertices in each H_0 component.

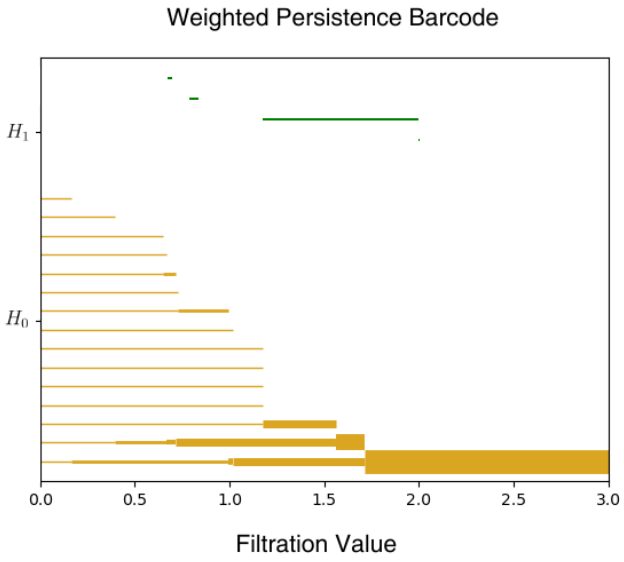


Figure V.2: The weighted persistence barcode describing the merging of components and formation of holes on a simulated dataset consisting of 15 samples and 2 quasi-identifiers.

The goal is to find filtration values where all H_0 bars have a thickness of k, and higher-order homology groups are trivial, indicating a viable k-anonymity complex. These regions are potential generalization values ϵ_k. The number of equivalence classes in a generalization relates to the maximal simplices [283] covering the complex at specific filtration values. Thus, a single persistence barcode calculation for a dataset can identify multiple generalizations for any k, which other algorithms cannot achieve [252, 270, 271, 256, 272, 273, 274, 264, 282].

The smallest viable generalization is optimal, and for stronger properties like l-diversity and t-closeness, other generalizations can be used.

Computational Constraints

Although [74] stands out amongst other methods of anonymization with its unique advantages, it comes at a cost since computing persistent homology is a costly operation. When limited to static datasets, the method needs to recompute the persistent homology for the entire data when any change occurs in the dataset. In the worst case scenario, this comes at a computational cost of $\mathcal{O}(\sum_i^M ({}^N C_i)^3)$ where N denotes the number of points, and M the number of quasi-identifiers [284, 285, 286].

V.4 Methods

In this section, we introduce our methodology to utilize persistence information to anonymize dynamic datasets. Recalculating persistent homology with every dataset change is computationally impractical. Our goal is to efficiently use existing persistence barcode information and extract insights as the data evolves. We address data removal, addition, and updating scenarios inductively.

First, we introduce hole-weighted persistence barcodes that help avoid recomputing persistent homology when data is removed. Next, we introduce *filtration trimming*, a technique that significantly reduces the number of persistent homology recomputations made when additional data is added. Finally we discuss data updates, where the stability of persistence diagrams help avoid recomputations.

V.4.1 Hole-weighted Persistence Barcodes

To capture information about holes across the filtration, we enhance weighted persistence barcodes from [74] to include both component and cycle information. This non-trivial task is detailed in Box V.1, which tracks the birth and progression of topological holes. Specifically, we track a k-dimensional hole from its birth ϵ_{birth} to its death ϵ_{death}. We do this by constructing a graph of simplices and using breadth-first search (BFS) [287] to find loops. Persistence data $\mathscr{P}$ will represent simplex formations and corresponding filtration values.

Figure V.3 simulated using persistence information with the help of [268] demonstrates that in contrast to the growing thickness of bars in H_0, bars for holes decrease in weight over time due to the ongoing filling of these holes.

We now establish our notation to describe hole-weighted persistence barcodes as visualized in Figure V.4. Let's represent each component in H_n as ${}_nB^i$. Each component ${}_nB^i$ comprises a series of bars represented as $({}_nB^i_1, {}_nB^i_2, \ldots, {}_nB^i_{n_i})$. Here, ${}_nB^i_j$ denotes the j^{th} bar in the component ${}_nB^i$. n_i represents the number of bars in the i^{th} n-dimensional component.

Further deconstructing, each bar, ${}_nB^i_j$, is represented as a tuple: ${}_nB^i_j = ({}_nb^i_j, {}_nd^i_j, {}_nw^i_j, {}_nV^i_j)$, where:

Box V.1 Tracking k-dimensional holes in a filtration

Input: Persistence data $\mathscr{P}$, $\epsilon_{birth}, \epsilon_{death}$.
Output: Updated hole-weighted persistence barcode $\mathscr{B}$.

1. Let $V_I \leftarrow [v_1, v_2, \ldots, v_{k+1}]$ be the simplex forming at ϵ_{birth}.
2. Initialize $S_k \leftarrow$ all k-simplices from $\mathscr{P}$ formed at or before ϵ_{birth}.
3. Initialize an undirected graph $G(V, E)$ where $V = S_k$.
4. For each pair of simplices $s_1, s_2 \in S_k$:
   (a) **if** $s_1 \cap s_2$ has k vertices:
      i. Add edge (s_1, s_2) to E.
5. $G' \leftarrow$ component of G containing V_I.
6. $L \leftarrow \text{BFS}(G')$ to identify all loops.
7. For each loop $l \in L$:
   (a) $\Delta_l \leftarrow \bigcup_{s \in l} s$.
   (b) **if** NOT $\exists s \in \mathscr{P}$ such that s formed before ϵ_{birth} and $s = \Delta_l$:
      i. Mark Δ_l as a hole.
8. For each k-simplex s formed after ϵ_{birth} and before ϵ_{death}:
   (a) **if** $s \subseteq \Delta_l$:
      i. Add s as a vertex to G'.
      ii. For each $s_{G'} \in G'$:
         A. **if** $s \cap s_{G'}$ has k vertices:
         B. Add edge $(s, s_{G'})$ to G'.
      iii. $L' \leftarrow$ loops in Δ_L using $\text{BFS}(G')$ such that $s \notin \Delta_{L'}$.
      iv. Update Δ_L with $\Delta_{L'}$.

Return: $\mathscr{B}$ updated based on changes to Δ_L.

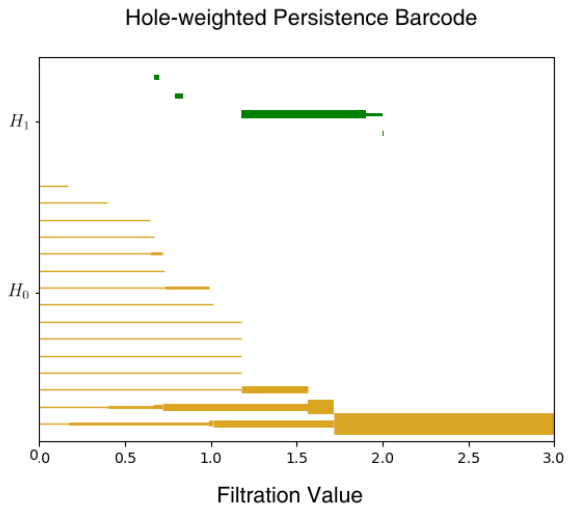


Figure V.3: Hole-weighted persistence barcode describing the merging of components and evolution of holes demonstrated on simulated data comprising 15 samples and 2 quasi-identifiers.

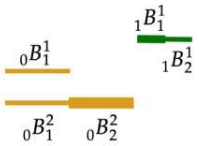


Figure V.4: Visual representation of the notation used.

- ${}_nb_j^i$ denotes the birth of a topological hole.
- ${}_nd_j^i$ denotes the death of a topological hole.
- ${}_nw_j^i$ represents the number of vertices forming the hole.
- ${}_nV_j^i$ is the set of vertices that form the hole.

To ensure chronological coherence, we maintain that for every $j < n$, the relation ${}_nd_j^i = {}_nb_{j+1}^i$ holds true. This alignment presents a sequential understanding of the evolution of topological holes over time.

V.4.2 Data Removal

Healthcare data management is subject to strict regulations, often requiring time-bound consent for patient information. As a result, healthcare institutions periodically delete patient data from their databases. To handle this removal of data efficiently, we directly modify hole-weighted persistence barcodes instead of recomputing persistent homology across the filtration of the updated data. Without loss of generality, we detail the removal of vertex v_r from the data in Box V.2. It is worth noting that we wouldn't need to track holes for further updates in the data since the algorithm inherently produces the hole-weighted persistence barcode associated with the updated data.

Box V.2 Data removal

Input: Persistence data $\mathscr{P}$, vertex v_r to be removed.
Output: Updated hole-weighted persistence barcode.

1. Collect all ${}_0B^r$ where ${}_0d_1^r$ is the shortest distance between v_r and any other point.
2. Remove any one component ${}_0B^{\hat{r}}$ that contains only one bar ${}_0B_1^{\hat{r}}$.
3. **if** ${}_0V_1^{\hat{r}} = [v_l] \neq [v_r]$:
   (a) For ${}_0B^{\tilde{r}}$ with ${}_0V_1^{\tilde{r}} = [v_r]$:
      i. Replace every occurrence of v_r with v_l and vice versa.
4. For each bar ${}_0B_j^i$ in every component:
   (a) **if** v_r is a member of ${}_0V_j^i$:
      i. **if** 1-simplex $[v_t, v_r]$ forms at b_j^i and $v_t \in {}_0V_j^i$:
         A. $\mathbf{d} \leftarrow \min\{\|v_r - v_s\| \mid v_s \notin {}_0V_j^i\}$.
         B. Adjust ${}_0b_j^i$ by adding $\mathbf{d}$.
      ii. Remove v_r from ${}_0V_j^i$.
      iii. Update ${}_0w_j^i = {}_0w_j^i - 1$.
5. For each component ${}_{k\geq 1}B^i$:
   (a) **if** $v_r \in {}_kV_j^i$:
      i. Remove ${}_kB_j^i$.
6. For each component without bars:
   (a) Remove the component.
7. Relabel the persistence barcode to reflect changes.

Return: Updated hole-weighted persistence barcode.

Algorithmic Complexity

In the worst-case scenario, where the dataset forms a fully connected graph, the BFS algorithm employed for generating the hole-weighted persistence barcode incurs a computational complexity of $\mathcal{O}(\sum_i^M ({}^N C_i)^3)$, where N represents the number of data points and M the dimension of the space the points reside in. This complexity is analogous to that of computing persistent homology for the dataset as discussed earlier in Section V.3. Consequently, the cumulative complexity for both computing the persistence information and generating the hole-weighted persistence barcode is $\mathcal{O}(2\sum_i^M ({}^N C_i)^3)$, in comparison to the $\mathcal{O}(\sum_i^M ({}^N C_i)^3)$ required solely for computing persistence information.

However, an important distinction arises in the context of data removal. While recalculating persistent homology after each data point removal remains computationally intensive, modifying the existing hole-weighted persistence barcode is considerably more efficient, with a complexity of only $\mathcal{O}(N)$. After undergoing K successive data removal operations, [74] would incur a computational complexity of $\mathcal{O}(\sum_{J=N-K}^N \sum_i^M ({}^J C_i)^3)$. In contrast, our approach maintains a more manageable complexity of $\mathcal{O}(2\sum_i^M ({}^N C_i)^3 + KN)$.

V.4.3 Data Addition

The continuous flow of new data in the healthcare sector highlights the need to regularly update medical datasets. Regular updates help capture changes in patient profiles and medical treatments, improving diagnostic and therapeutic strategies. However, this continuous addition of data presents a challenge: maintaining the privacy of individuals while ensuring k-anonymity. Each new entry can potentially compromise existing anonymizations.

Addressing the complexities of data addition, our concern is the recalibration of persistence barcodes without the need to compute the persistence information for the entire Cêch filtration. We demonstrate that the addition of a single point does not demand a comprehensive homology computation in Algorithm V.3. Instead, updates primarily occur in the local vicinity of the added point. By leveraging the persistent homology computations from the static data, we only recompute specific trimmed segments influenced by the new data. This methodology promotes computational efficiency by reducing redundant calculations.

When introducing a new point, v_{N+1}, we first identify the circumcenters of all simplices that include v_{N+1} up to dimension M within its local neighbourhood. Here, M represents the number of quasi-identifiers. Subsequently, we calculate the distances between these circumcenters and v_{N+1}, sorting them in $\delta = \{\delta_1, \delta_2, \ldots,\}$ where, for $i < j$, $\delta_i < \delta_j$.

Box V.3 Data addition

Input: δ, persistence data $\mathscr{P}$.
Output: Updated persistence barcode.

1. Add new component ${}_0B_1^{N+1} = (0, \min(\delta), 1, [v_{N+1}])$.
2. For any one v_l such that $\|v_l - v_{N+1}\| = \min(\delta)$:
   (a) Find bar ${}_0B_j^l$ such that ${}_0b_j^l \leq \min(\delta) \leq {}_0d_j^l$ where $v_l \in {}_0V_j^l$.
   (b) Set ${}_0B_j^l = ({}_0b_j^l, \min(\delta), {}_0w_j^l, {}_0V_j^l)$.
   (c) Set ${}_0B_{j_+}^l = (\min(\delta), {}_0d_j^l, {}_0w_j^l + 1, {}_0V_j^l \cup [v_{N+1}])$.
3. Identify all δ_i with $i > 1$ and compute persistence in ascending order of δ_i.
4. Determine homology changes as compared to previous persistence information and denote as $\tilde{\delta} = \{\tilde{\delta}_1, \ldots, \tilde{\delta}_k\}$.
5. For each $0 \leq i < k$:
   (a) Compute persistent homology using the updatable SNF for the filtrations:
   $$\mathscr{C}^{\tilde{\delta}_i} \to \mathscr{C}^{\epsilon_i^1} \to \ldots \to \mathscr{C}^{\epsilon_i^k} \to \mathscr{C}^{\tilde{\delta}_{i+1}}.$$
   (b) **if** $H_k(\mathscr{C}^{\epsilon_i^t})$ matches the previous persistence information for any k at ϵ_i^t:
      i. Move to next filtration.
6. Relabel indices as per component and bar order.

Return: Updated persistence barcode.

Data Points	Added Points	Filtration Length	Trimmed Length
10	1	231	19
10	2	298	20
10	5	575	45
20	1	1561	347
20	5	2625	386
20	10	4525	1115
50	1	22151	3301
50	5	27775	3792
50	10	36050	3374
100	1	171801	15379
100	5	193025	17263
100	10	221925	18760
100	25	325625	19242

Table V.3: Filtration Lengths and Trimmed Filtration Lengths for Simulated Data with 2 Quasi-identifiers.

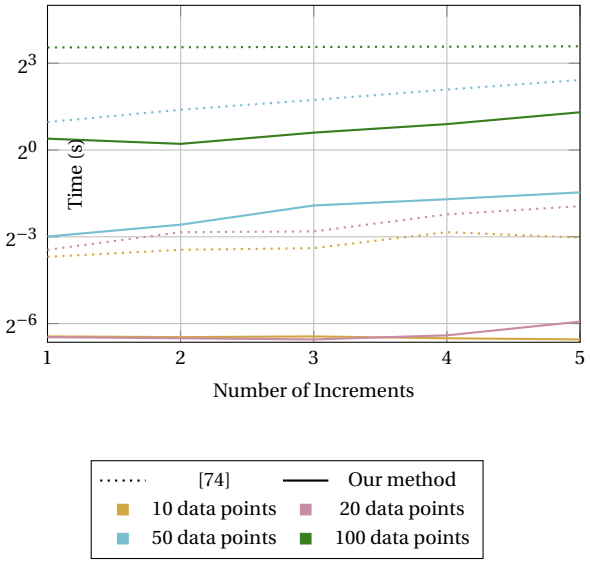


Figure V.5: Comparison of methods when data points are increased by 10% of the original dataset at each step. The time required to compute persistent homology on full and trimmed filtration lengths is plotted.

Algorithmic Complexity

Calculating a simplex's circumcenter is an $\mathcal{O}(t)$ operation using Welzl's recursive algorithm [288], where t is the dimensionality of the simplex. Assuming we have $\bar{T}$ simplices around the added point, calculating the circumcenters of $\bar{T}$ t-dimensional simplices has a complexity of only $\mathcal{O}\left({}^{\bar{T}}C_{\bar{T}/2}(t/2)\right)$. A key insight emerges from this: computing circumcenters around our new data point is considerably less resource-intensive than computing the persistent homology for an entire filtration.

The parameter δ and its reduction $\tilde{\delta}$ has proven instrumental in our algorithmic approach to trimming the filtration length in our Cêch filtration, as shown in Table V.3 and Figure V.5, generated using [268]. Here we measure the filtration lengths when additional data is added, and after we perform filtration trimming. The runtimes reported are total persistent homology computation times for the respective filtration lengths. This streamlined process, even with the addition of extra data points, dramatically cuts down on computational overhead, especially when considering larger datasets. For our experiments, we worked with data simulated using two quasi-identifiers to maintain simplicity.

V.4.4 Data Updatability

In rare cases when EHR data entries are modified, whether to refine diagnoses or correct data, it raises challenges for maintaining k-anonymity.

Based on the findings from [289], minor data perturbations have minimal impact on persistent homology. This suggests a practical approach: for slight data modifications, first check if the current anonymized dataset meets k-anonymity standards. If not, manage changes using protocols for data removal and addition. If the data is minimally altered and k-anonymity is satisfied, updating the hole-weighted persistence barcode is straightforward. If a component or hole arises due to the formation of a simplex $[\ldots, v_u, \ldots]$ with v_u being the adjusted data, we modify the associated bar by computing the circumcenter of the simplex.

V.4.5 Comparison with related literature

Below, in Table V.4, we present a comprehensive comparison between our method and established k-anonymity techniques against key criteria. These were chosen to highlight the strengths and potential areas for improvement of our approach relative to the current State-of-the-art. Our work, and [74] stand out as the only two topology-informed methods, that can generate multiple generalizations for multiple anonymization criteria with a single computation. Additionally, our method's ability to handle dynamic data allows us to adapt not only to changing data but also to evolving privacy requirements.

Method	Technique Applied	Handles Categorical Data	Multiple Generalisations	Multiple Anonymizations	Adaptability to Dynamic Databases
[271]	Recoding-based	×	×	×	×
[270]	Recoding/Hierarchy-based	✓	×	×	×
[290]	Recoding/Hierarchy-based	✓	×	×	×
[280]	Recoding-based	×	×	×	partially
[281]	Recoding/Hierarchy-based	✓	×	×	✓
[282]	Microaggregation	✓	×	×	✓
[74]	Recoding-based	×	✓	✓	×
Our Method	Recoding-based	×	✓	✓	✓

Table V.4: Comparison of various k-anonymity methods.

V.5 Conclusion

In conclusion, our work substantially advances the application of TDA techniques for privacy-preserving dynamic data publishing in the context of EHRs. Previous work in the field focused on static datasets [74] and required computation of persistent homology on the whole dataset whenever data is updated. Our methodology extends this by addressing key challenges in dynamic databases, including data removal, addition, and updatability, enabling rapid adaptability without the need for extensive recomputation of persistent homology for updated data. Our solution can adapt not only to dynamic data, but dynamic privacy demands without requiring additional computational strain.

For data removal, we have innovatively utilized hole-weighted persistence barcodes, constructed through a breadth-first search algorithm on a graph derived from existing persistence data. This approach allows for systematic editing of the barcodes upon data removal. In the case of data addition, we have refined the process by trimming the Cêch filtration, thus significantly reducing the number of persistent homology recomputations required, as our experiments have demonstrated. Regarding data updatability, we focused on the stability of persistence diagrams and computed a limited number of simplex circumcenters, avoiding expensive persistent homology re-computations.

Our approach enhances the balance between data utility, anonymization flexibility and patient privacy, which is crucial in the context of rapid EHR adoption and stringent regulations like the GDPR. Although our work primarily addresses numerical data, extending these methodologies to include categorical data using zigzag persistent homology is a promising direction for future research. Moreover, incorporating more robust privacy measures such as l-diversity and t-closeness could enhance the algorithm. As tools for persistence computations continue to evolve, the scalability and practical applicability of our approach are expected to increase, providing greater utility across various applications.

Code Availability

Our code is available at the following URL:
https://github.com/mdppml/dynamic-topological-k-anonymity [291].

Acknowledgements

This research was supported by the German Ministry of Research and Education (BMBF), project number 01ZZ2010. We thank Saradha Senthil Velu for their invaluable discussions that helped in formulating the ideas presented in the paper.

Bibliography

[1] Anika Hannemann, Arjhun Swaminathan, Ali Burak Ünal, and Mete Akgün. Private, efficient and scalable kernel learning for medical image analysis. In *International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics*, volume 15276 of *Lecture Notes in Computer Science*, pages 81–95. Springer Nature Switzerland, May 2025. doi: 10.1007/978-3-031-89704-7_7.

[2] Arjhun Swaminathan, Anika Hannemann, Ali Burak Ünal, Nico Pfeifer, and Mete Akgün. Pp-gwas: Privacy preserving multi-site genome-wide association studies. *Nature Communications*, 16(1):11030, December 2025. ISSN 2041-1723. doi: 10.1038/s41467-025-66771-z.

[3] Arjhun Swaminathan and Mete Akgün. Distributed and secure kernel-based quantum machine learning. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=3jdI0aEW3k.

[4] Arjhun Swaminathan and Mete Akgün. Accelerating targeted hard-label adversarial attacks in low-query black-box settings. *arXiv preprint arXiv:2505.16313*, 2025.

[5] Arjhun Swaminathan and Mete Akgün. Dynamic k-anonymity for electronic health records: A topological framework. In *European Symposium on Research in Computer Security*, volume 15263 of *Lecture Notes in Computer Science*, pages 137–152. Springer Nature Switzerland, April 2025. doi: 10.1007/978-3-031-82349-7_10.

[6] Danah Boyd and Kate Crawford. Six provocations for big data. In *A decade in internet time: Symposium on the dynamics of the internet and society*, 2011.

[7] David Reinsel-John Gantz-John Rydning, John Reinsel, and John Gantz. The digitization of the world from edge to core. *Framingham: International Data Corporation*, 16:1–28, 2018.

[8] Adrien Marie Legendre. *Nouvelles méthodes pour la détermination des orbites des comètes: avec un supplément contenant divers perfectionnemens de ces méthodes et leur application aux deux comètes de 1805*. Courcier, 1806.

[9] Thomas H Davenport and John C Beck. The attention economy. *Ubiquity*, 2001(May): 1–es, 2001.

[10] Linyuan Lü, Matúš Medo, Chi Ho Yeung, Yi-Cheng Zhang, Zi-Ke Zhang, and Tao Zhou. Recommender systems. *Physics reports*, 519(1):1–49, 2012.

[11] Shoshana Zuboff. The age of surveillance capitalism. In *Social theory re-wired*, pages 203–213. Routledge, 2023.

[12] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pages 191–198, 2016.

[13] Matus Tomlein, Branislav Pecher, Jakub Simko, Ivan Srba, Robert Moro, Elena Stefancova, Michal Kompan, Andrea Hrckova, Juraj Podrouzek, and Maria Bielikova. An audit of misinformation filter bubbles on youtube: Bubble bursting and recent behavior changes. In *Proceedings of the 15th ACM Conference on Recommender Systems*, pages 1–11, 2021.

[14] Jim Isaak and Mina J Hanna. User data privacy: Facebook, cambridge analytica, and privacy protection. *Computer*, 51(8):56–59, 2018.

[15] THE EUROPEAN PARLIAMENT and THE COUNCIL OF THE EUROPEAN UNION. Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation), 2016. URL https://eur-lex.europa.eu/eli/reg/2016/679/oj/eng.

[16] International Organization for Standardization. Iso/iec 27701:2025 information security, cybersecurity and privacy protection — privacy information management systems — requirements and guidance. https://www.iso.org/standard/27701, 2025. Accessed 2026-03-08.

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[18] Ann Cavoukian et al. Privacy by design: The 7 foundational principles. *Information and privacy commissioner of Ontario, Canada*, 5(2009):12, 2009.

[19] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. *Advances in neural information processing systems*, 32, 2019.

[20] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients-how easy is it to break privacy in federated learning? *Advances in neural information processing systems*, 33:16937–16947, 2020.

[21] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. Deep models under the gan: Information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 603–618, 2017.

[22] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[23] Minhao Cheng, Thong Le, Pin-Yu Chen, Jinfeng Yi, Huan Zhang, and Cho-Jui Hsieh. Query-efficient hard-label black-box attack: An optimization-based approach. *arXiv preprint arXiv:1807.04457*, 2018.

[24] Md Farhamdur Reza, Ali Rahmati, Tianfu Wu, and Huaiyu Dai. Cgba: Curvature-aware geometric black-box attack. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 124–133, 2023.

[25] Jie Li, Rongrong Ji, Peixian Chen, Baochang Zhang, Xiaopeng Hong, Ruixin Zhang, Shaoxin Li, Jilin Li, Feiyue Huang, Shaoqing Ren, and Yongjian Sun, JiWu. Aha! adaptive history-driven attack for decision-based black-box models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16168–16177, 2021.

[26] Thibault Maho, Teddy Furon, and Erwan Le Merrer. Surfree: a fast surrogate-free black-box attack. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 770–778, 2016.

[27] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 111–125. IEEE, 2008.

[28] International Conference of Data Protection and Privacy Commissioners. Resolution on privacy by design, 2010. URL https://www.edps.europa.eu/sites/default/files/publication/10-10-27_jerusalem_resolutionon_privacybydesign_en.pdf.

[29] Latanya Sweeney. Simple demographics often identify people uniquely. *Health (San Francisco)*, 671(2000):1–34, 2000.

[30] Wajih Ul Hassan, Saad Hussain, and Adam Bates. Analysis of privacy protections in fitness tracking social networks-or-you can run, but can you hide? In *27th USENIX Security Symposium (USENIX Security 18)*, pages 497–512, 2018.

[31] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[32] Organisation for Economic Co-operation and Development (OECD). The oecd privacy framework: Guidelines governing the protection of privacy and transborder flows of personal data, 2013. URL https://legalinstruments.oecd.org/public/doc/114/114.en.pdf.

[33] Council of Europe. Protocol amending the convention for the protection of individuals with regard to automatic processing of personal data, 2018. URL https://rm.coe.int/16808ac918.

[34] International Organization for Standardization. ISO 31700-1:2023 consumer protection — privacy by design for consumer goods and services — part 1: High-level requirements. ISO, 2023. URL https://www.iso.org/standard/84977.html. Standard.

[35] European Union Agency for Cybersecurity (ENISA). Data protection engineering: From theory to practice, 2022. URL https://www.enisa.europa.eu/publications/data-protection-engineering.

[36] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191, 2017.

[37] Yehuda Lindell. Secure multiparty computation. *Communications of the ACM*, 64(1): 86–96, 2020.

[38] Cynthia Dwork. Differential privacy. In *International colloquium on automata, languages, and programming*, pages 1–12. Springer, 2006.

[39] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 169–178, 2009.

[40] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333, 2015.

[41] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.

[42] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction {APIs}. In *25th USENIX security symposium (USENIX Security 16)*, pages 601–618, 2016.

[43] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *International conference on artificial intelligence and statistics*, pages 2938–2948. PMLR, 2020.

[44] Trevor Hastie, Robert Tibshirani, Jerome Friedman, et al. The elements of statistical learning, 2009.

[45] A Aizerman. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and remote control*, 25:821–837, 1964.

[46] Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404, 1950.

[47] Bernhard Schölkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[48] Stefanie Warnat-Herresthal, Hartmut Schultze, Krishnaprasad Lingadahalli Shastry, Sathyanarayanan Manamohan, Saikat Mukherjee, Vishesh Garg, Ravi Sarveswara, Kristian Händler, Peter Pickkers, N Ahmad Aziz, et al. Swarm learning for decentralized and confidential clinical machine learning. *Nature*, 594(7862):265–270, 2021.

[49] David Evans, Vladimir Kolesnikov, and Mike Rosulek. A pragmatic introduction to secure multi-party computation. *Foundations and Trends® in Privacy and Security*, 2 (2-3):70–246, 2018.

[50] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. Homomorphic encryption for arithmetic of approximate numbers. In *International conference on the theory and application of cryptology and information security*, pages 409–437. Springer, 2017.

[51] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[52] Yuval Ishai and Eyal Kushilevitz. Randomizing polynomials: A new representation with applications to round-efficient secure computation. *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 294–304, 2000.

[53] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Computationally private randomizing polynomials and their applications. *computational complexity*, 15(2):115–162, 2006.

[54] Ali Burak Ünal, Mete Akgün, and Nico Pfeifer. A framework with randomized encoding for a fast privacy preserving calculation of non-linear kernels for machine learning applications in precision medicine. In *Cryptology and Network Security: 18th International Conference, CANS 2019, Fuzhou, China, October 25–27, 2019, Proceedings 18*, pages 493–511. Springer, 2019.

[55] Ali Burak Ünal, Mete Akgün, and Nico Pfeifer. Escaped: Efficient secure and private dot product framework for kernel-based machine learning algorithms with applications in healthcare. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9988–9996, 2021.

[56] Anika Hannemann, Ali Burak Ünal, Arjhun Swaminathan, Erik Buchmann, and Mete Akgün. A privacy-preserving framework for collaborative machine learning with kernel methods. In *2023 5th IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, pages 82–90. IEEE, 2023.

[57] Shai Halevi, Yuval Ishai, Eyal Kushilevitz, and Tal Rabin. Additive randomized encodings and their applications. *Cryptology ePrint Archive*, 2023.

[58] Emil Uffelmann, Qin Qin Huang, Nchangwi Syntia Munung, Jantina De Vries, Yukinori Okada, Alicia R Martin, Hilary C Martin, Tuuli Lappalainen, and Danielle Posthuma. Genome-wide association studies. *Nature Reviews Methods Primers*, 1(1):59, 2021.

[59] Joelle Mbatchou, Leland Barnard, Joshua Backman, Anthony Marcketta, Jack A Kosmicki, Andrey Ziyatdinov, Christian Benner, Colm O'Dushlaine, Mathew Barber, Boris

Boutkov, et al. Computationally efficient whole-genome regression for quantitative and binary traits. *Nature genetics*, 53(7):1097–1103, 2021.

[60] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010.

[61] Maria Schuld and Nathan Killoran. Quantum machine learning in feature hilbert spaces. *Physical review letters*, 122(4):040504, 2019.

[62] Charles H Bennett, Gilles Brassard, Claude Crépeau, Richard Jozsa, Asher Peres, and William K Wootters. Teleporting an unknown quantum state via dual classical and einstein-podolsky-rosen channels. *Physical review letters*, 70(13):1895, 1993.

[63] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017.

[64] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[65] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017.

[66] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

[67] Jianbo Chen, Michael I Jordan, and Martin J Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. In *2020 ieee symposium on security and privacy (sp)*, pages 1277–1294. IEEE, 2020.

[68] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International journal of uncertainty, fuzziness and knowledge-based systems*, 10(05):557–570, 2002.

[69] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3–es, 2007.

[70] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *2007 IEEE 23rd international conference on data engineering*, pages 106–115. IEEE, 2006.

[71] Brian Lee, Brandi Dupervil, Nicholas P Deputy, Wil Duck, Stephen Soroka, Lyndsay Bottichio, Benjamin Silk, Jason Price, Patricia Sweeney, Jennifer Fuld, et al. Protecting privacy and transforming covid-19 case surveillance datasets for public use. *Public Health Reports*, 136(5):554–561, 2021.

[72] Afra J Zomorodian. *Topology for computing*, volume 16. Cambridge university press, 2005.

[73] Herbert Edelsbrunner, John Harer, et al. Persistent homology-a survey. *Contemporary mathematics*, 453(26):257–282, 2008.

[74] Alberto Speranzon and Shaunak D Bopardikar. An algebraic topological perspective to privacy. In *2016 American Control Conference (ACC)*, pages 2086–2091. IEEE, 2016.

[75] Keke Chen and Ling Liu. Privacy preserving data classification with rotation perturbation. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 4–pp. IEEE, 2005.

[76] Keke Chen, Gordon Sun, and Ling Liu. Towards attack-resilient geometric data perturbation. In *proceedings of the 2007 SIAM international conference on Data mining*, pages 78–89. SIAM, 2007.

[77] Keng-Pei Lin, Yi-Wei Chang, and Ming-Syan Chen. Secure support vector machines outsourcing with random linear transformation. *Knowledge and Information Systems*, 44:147–176, 2015.

[78] Pamela J LaMontagne, Tammie LS Benzinger, John C Morris, Sarah Keefe, Russ Hornbeck, Chengjie Xiong, Elizabeth Grant, Jason Hassenstab, Krista Moulder, Andrei G Vlassenko, et al. Oasis-3: longitudinal neuroimaging, clinical, and cognitive dataset for normal aging and alzheimer disease. *MedRxiv*, pages 2019–12, 2019.

[79] Shenggan. Bccd dataset. https://github.com/Shenggan/BCCD_Dataset, 2017. Accessed on 27.07.2023.

[80] Hyunghoon Cho, David J Wu, and Bonnie Berger. Secure genome-wide association analysis using multiparty computation. *Nature biotechnology*, 36(6):547–551, 2018.

[81] David Froelicher, Juan R Troncoso-Pastoriza, Jean Louis Raisaro, Michel A Cuendet, Joao Sa Sousa, Hyunghoon Cho, Bonnie Berger, Jacques Fellay, and Jean-Pierre Hubaux. Truly privacy-preserving federated analytics for precision medicine with multiparty homomorphic encryption. *Nature communications*, 12(1):5910, 2021.

[82] Hyunghoon Cho, David Froelicher, Jeffrey Chen, Manaswitha Edupalli, Apostolos Pyrgelis, Juan R Troncoso-Pastoriza, Jean-Pierre Hubaux, and Bonnie Berger. Secure and federated genome-wide association studies for biobank-scale datasets. *Nature Genetics*, pages 1–6, 2025.

[83] Yu-Bo Sheng, Lan Zhou, and Gui-Lu Long. One-step quantum secure direct communication. *Science Bulletin*, 67(4):367–374, 2022.

[84] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20, 2007.

[85] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992.

[86] Hyunseok Seo, Masoud Badiei Khuzani, Varun Vasudevan, Charles Huang, Hongyi Ren, Ruoxiu Xiao, Xiao Jia, and Lei Xing. Machine learning techniques for biomedical image segmentation: an overview of technical aspects and introduction to state-of-art applications. *Medical physics*, 47(5):e148–e167, 2020.

[87] Basna Mohammed Salih Hasan and Adnan Mohsin Abdulazeez. A review of principal component analysis algorithm for dimensionality reduction. *Journal of Soft Computing and Data Mining*, 2(1):20–30, 2021.

[88] Nisar Wani and Khalid Raza. Multiple kernel-learning approach for medical image analysis. In *Soft computing based medical image analysis*, pages 31–47. Elsevier, 2018.

[89] Xue Ran, Junyi Shi, Yalan Chen, and Kui Jiang. Multimodal neuroimage data fusion based on multikernel learning in personalized medicine. *Frontiers in Pharmacology*, 13:947657, 2022.

[90] Bozheng Dou, Zailiang Zhu, Ekaterina Merkurjev, Lu Ke, Long Chen, Jian Jiang, Yueying Zhu, Jie Liu, Bengong Zhang, and Guo-Wei Wei. Machine learning methods for small data challenges in molecular science. *Chemical Reviews*, 123(13):8736–8780, 2023.

[91] Khaled El Emam, Elizabeth Jonker, Luk Arbuckle, and Bradley Malin. A systematic review of re-identification attacks on health data. *PloS one*, 6(12):e28071, 2011.

[92] Boris Lubarsky. Re-identification of "anonymized data". *Georgetown Law Technology Review. Available online: https://www. georgetownlawtechreview. org/re-identification-of-anonymized-data/GLTR-04-2017 (accessed on 10 September 2021)*, 2010.

[93] David Peloquin, Michael DiMaio, Barbara Bierer, and Mark Barnes. Disruptive and avoidable: Gdpr challenges to secondary research uses of data. *European Journal of Human Genetics*, 28(6):697–705, 2020.

[94] Andrew C Yao. Protocols for secure computations. In *23rd annual symposium on foundations of computer science (sfcs 1982)*, pages 160–164. IEEE, 1982.

[95] Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. Differential privacy has disparate impact on model accuracy. *Advances in neural information processing systems*, 32, 2019.

[96] Jawad Ahmad and Fawad Ahmed. Efficiency analysis and security evaluation of image encryption schemes. *computing*, 23(04):25, 2010.

[97] Anika Hannemann, Benjamin Friedl, and Erik Buchmann. Differentially private multi-label learning is harder than you'd think. In *2024 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 40–47. IEEE, 2024.

[98] Jameela Al-Jaroodi, Nader Mohamed, and Eman Abukhousa. Health 4.0: on the way to realizing the healthcare of the future. *Ieee Access*, 8:211189–211210, 2020.

[99] Subrato Bharati, Prajoy Podder, M. Rubaiyat Hossain Mondal, and Pinto Kumar Paul. Applications and challenges of cloud integrated iomt. In *Cognitive Internet of Medical Things for Smart Healthcare: Services and Applications*, pages 67–85. Springer, 2021.

[100] Mu-Hsing Kuo et al. Opportunities and challenges of cloud computing to improve health care services. *Journal of medical Internet research*, 13(3):e1867, 2011.

[101] Lena Griebel, Hans-Ulrich Prokosch, Felix Köpcke, Dennis Toddenroth, Jan Christoph, Ines Leb, Igor Engel, and Martin Sedlmayr. A scoping review of cloud computing in healthcare. *BMC medical informatics and decision making*, 15(1):1–16, 2015.

[102] Thomas Hofmann, Bernhard Schölkopf, and Alexander J Smola. Kernel methods in machine learning. 2008.

[103] J Shawe-Taylor. Kernel methods for pattern analysis. *Cambridge University Press google schola*, 2:181–201, 2004.

[104] Abelardo Montesinos-López, Osval Antonio Montesinos-López, José Cricelio Montesinos-López, Carlos Alberto Flores-Cortes, Roberto de la Rosa, and José Crossa. A guide for kernel generalized regression methods for genomic-enabled prediction. *Heredity*, 126(4):577–596, 2021.

[105] Runzhao Yao, Shaoyi Du, Teng Wan, Wenting Cui, Yang Yang, Yang Jing, and Ce Li. Robust registration algorithm based on rational quadratic kernel for point sets with outliers and noise. *Multimedia Tools and Applications*, 80:27925–27945, 2021.

[106] Minta Thomas, Kris De Brabanter, and Bart De Moor. New bandwidth selection criterion for kernel pca: Approach to dimensionality reduction and classification problems. *BMC bioinformatics*, 15:1–12, 2014.

[107] Jun Cheng, Zhi Han, Rohit Mehra, Wei Shao, Michael Cheng, Qianjin Feng, Dong Ni, Kun Huang, Liang Cheng, and Jie Zhang. Computational analysis of pathological images enables a better diagnosis of tfe3 xp11. 2 translocation renal cell carcinoma. *Nature communications*, 11(1):1778, 2020.

[108] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[109] Viraaji Mothukuri, Reza M Parizi, Seyedamin Pouriyeh, Yan Huang, Ali Dehghantanha, and Gautam Srivastava. A survey on security and privacy of federated learning. *Future Generation Computer Systems*, 115:619–640, 2021.

[110] Neel Guha, Ameet Talwalkar, and Virginia Smith. One-shot federated learning. *arXiv preprint arXiv:1902.11175*, 2019.

[111] Jaideep Vaidya, Hwanjo Yu, and Xiaoqian Jiang. Privacy-preserving svm classification. *Knowledge and Information Systems*, 14:161–178, 2008.

[112] Febrianti Wibawa, Ferhat Ozgur Catak, Murat Kuzlu, Salih Sarp, and Umit Cali. Homomorphic encryption and federated learning based privacy-preserving cnn training: Covid-19 detection use-case. In *Proceedings of the 2022 European Interdisciplinary Cybersecurity Conference*, pages 85–90, 2022.

[113] Dimitris Stripelis, Hamza Saleem, Tanmay Ghai, Nikhil Dhinagar, Umang Gupta, Chrysovalantis Anastasiou, Greg Ver Steeg, Srivatsan Ravi, Muhammad Naveed, Paul M Thompson, et al. Secure neuroimaging analysis using federated learning with homomorphic encryption. In *17th International Symposium on Medical Information Processing and Analysis*, volume 12088, pages 351–359. SPIE, 2021.

[114] Vaikkunth Mugunthan, Antigoni Polychroniadou, David Byrd, and Tucker Hybinette Balch. Smpai: Secure multi-party computation for federated learning. In *Proceedings of the NeurIPS 2019 Workshop on Robust AI in Financial Services*, 2019.

[115] Chi Zhang, Sotthiwat Ekanut, Liangli Zhen, and Zengxiang Li. Augmented multi-party computation against gradient leakage in federated learning. *IEEE Transactions on Big Data*, 2022.

[116] Huajie Chen, Ali Burak Ünal, Mete Akgün, and Nico Pfeifer. Privacy-preserving svm on outsourced genomic data via secure multi-party computation. In *Proceedings of the Sixth International Workshop on Security and Privacy Analytics*, pages 61–69, 2020.

[117] Bjarne Pfitzner, Nico Steckhan, and Bert Arnrich. Federated learning in a medical context: a systematic literature review. *ACM Transactions on Internet Technology (TOIT)*, 21(2):1–31, 2021.

[118] Mohammed Adnan, Shivam Kalra, Jesse C Cresswell, Graham W Taylor, and Hamid R Tizhoosh. Federated learning and differential privacy for medical image analysis. *Scientific reports*, 12(1):1–10, 2022.

[119] Mohammad Malekzadeh, Burak Hasircioglu, Nitish Mital, Kunal Katarya, Mehmet Emre Ozfatura, and Deniz Gündüz. Dopamine: Differentially private federated learning on medical data. *arXiv preprint arXiv:2101.11693*, 2021.

[120] Keke Chen and Ling Liu. Geometric data perturbation for privacy preserving outsourced data mining. *Knowledge and information systems*, 29:657–695, 2011.

[121] Keng-Pei Lin. Privacy-preserving kernel k-means outsourcing with randomized kernels. In *2013 IEEE 13th International Conference on Data Mining Workshops*, pages 860–866. IEEE, 2013.

[122] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20: 273–297, 1995.

[123] Christopher Williams and Carl Rasmussen. Gaussian processes for regression. *Advances in neural information processing systems*, 8, 1995.

[124] Ole Christensen et al. *An introduction to frames and Riesz bases*, volume 7. Springer, 2003.

[125] Chengliang Zhang, Suyi Li, Junzhe Xia, Wei Wang, Feng Yan, and Yang Liu. Batchcrypt: Efficient homomorphic encryption for cross-silo federated learning. In *Proceedings of the 2020 USENIX Annual Technical Conference (USENIX ATC 2020)*, 2020.

[126] John C Gower and Garmt B Dijksterhuis. *Procrustes problems*, volume 30. OUP Oxford, 2004.

[127] Zhi-Qiang Zeng, Hong-Bin Yu, Hua-Rong Xu, Yan-Qi Xie, and Ji Gao. Fast training support vector machines using parallel sequential minimal optimization. In *2008 3rd international conference on intelligent system and knowledge engineering*, volume 1, pages 997–1001. IEEE, 2008.

[128] Morteza Heidari, Seyedehnafiseh Mirniaharikandehei, Abolfazl Zargari Khuzani, Gopichandh Danala, Yuchen Qiu, and Bin Zheng. Improving the performance of cnn to predict the likelihood of covid-19 using chest x-ray images with preprocessing algorithms. *International journal of medical informatics*, 144:104284, 2020.

[129] Rubina Sarki, Khandakar Ahmed, Hua Wang, Yanchun Zhang, Jiangang Ma, and Kate Wang. Image preprocessing in classification and identification of diabetic eye diseases. *Data Science and Engineering*, 6(4):455–471, 2021.

[130] A. Hannemann, A. Swaminathan, A. B. Ünal, and M. Akgün. Private, efficient and scalable kernel learning for medical image analysis - code, 2023. URL https://doi.org/10.5281/zenodo.18743744.

[131] The Wellcome Trust Case Control Consortium. Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature*, 447(7145): 661–678, 2007.

[132] Mark I McCarthy, Gonçalo R Abecasis, Lon R Cardon, David B Goldstein, Julian Little, John PA Ioannidis, and Joel N Hirschhorn. Genome-wide association studies for complex traits: consensus, uncertainty and challenges. *Nature reviews genetics*, 9(5): 356–369, 2008.

[133] Teri A Manolio, Francis S Collins, Nancy J Cox, David B Goldstein, Lucia A Hindorff, David J Hunter, Mark I McCarthy, Erin M Ramos, Lon R Cardon, Aravinda Chakravarti, et al. Finding the missing heritability of complex diseases. *Nature*, 461(7265):747–753, 2009.

[134] Sarah E Graham, Shoa L Clarke, Kuan-Han H Wu, Stavroula Kanoni, Greg JM Zajac, Shweta Ramdas, Ida Surakka, Ioanna Ntalla, Sailaja Vedantam, Thomas W Winkler, et al. The power of genetic diversity in genome-wide association studies of lipids. *Nature*, 600(7890):675–679, 2021.

[135] Vassily Trubetskoy, Antonio F Pardiñas, Ting Qi, Georgia Panagiotaropoulou, Swapnil Awasthi, Tim B Bigdeli, Julien Bryois, Chia-Yen Chen, Charlotte A Dennison, Lynsey S Hall, et al. Mapping genomic loci implicates genes and synaptic biology in schizophrenia. *Nature*, 604(7906):502–508, 2022.

[136] Ciara Staunton, Rachel Adams, Dominique Anderson, Talishiea Croxton, Dorcas Kamuya, Marianne Munene, and Carmen Swanepoel. Protection of personal information act 2013 and data protection for health research in south africa. *International Data Privacy Law*, 10(2):160–179, 2020.

[137] Mete Akgün, A Osman Bayrak, Bugra Ozer, and M Şamil Sağıroğlu. Privacy preserving processing of genomic data: A survey. *Journal of biomedical informatics*, 56:103–111, 2015.

[138] Elizabeth K Speliotes, Cristen J Willer, Sonja I Berndt, Keri L Monda, Gudmar Thorleifsson, Anne U Jackson, Hana Lango Allen, Cecilia M Lindgren, Jian'an Luan, Reedik Mägi, et al. Association analyses of 249,796 individuals reveal 18 new loci associated with body mass index. *Nature genetics*, 42(11):937–948, 2010.

[139] Evangelos Evangelou and John PA Ioannidis. Meta-analysis methods for genome-wide association studies and beyond. *Nature Reviews Genetics*, 14(6):379–389, 2013.

[140] Dan-Yu Lin and Daniel Zeng. On the relative efficiency of using summary statistics versus individual-level data in meta-analysis. *Biometrika*, 97(2):321–332, 2010.

[141] Scott D Constable, Yuzhe Tang, Shuang Wang, Xiaoqian Jiang, and Steve Chapin. Privacy-preserving gwas analysis on federated genomic datasets. *BMC medical informatics and decision making*, 15:1–9, 2015.

[142] Charlotte Bonte, Eleftheria Makri, Amin Ardeshirdavani, Jaak Simm, Yves Moreau, and Frederik Vercauteren. Towards practical privacy-preserving genome-wide association study. *BMC bioinformatics*, 19:1–12, 2018.

[143] Can Kockan, Kaiyuan Zhu, Natnatee Dokmai, Nikolai Karpov, M Oguzhan Kulekci, David P Woodruff, and S Cenk Sahinalp. Sketching algorithms for genomic data analysis and querying in a secure enclave. *Nature methods*, 17(3):295–301, 2020.

[144] Wentao Li, Han Chen, Xiaoqian Jiang, and Arif Harmanci. Federated generalized linear mixed models for collaborative genome-wide association studies. *Iscience*, 26 (8), 2023.

[145] Adriana López-Alt, Eran Tromer, and Vinod Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 1219–1234, 2012.

[146] Kun Liu, Hillol Kargupta, and Jessica Ryan. Random projection-based multiplicative data perturbation for privacy preserving distributed data mining. *IEEE Transactions on knowledge and Data Engineering*, 18(1):92–106, 2005.

[147] Ricardo Mendes and João P Vilela. Privacy-preserving data mining: methods, metrics, and applications. *IEEE Access*, 5:10562–10582, 2017.

[148] Stanley RM Oliveira and Osmar R Zaiane. Privacy preserving clustering by data transformation. *Journal of Information and Data Management*, 1(1):37–37, 2010.

[149] Tapan K Nayak, Bimal Sinha, and Laura Zayatz. Statistical properties of multiplicative noise masking for confidentiality protection. *Journal of Official Statistics*, 27(3):527, 2011.

[150] Carl Kadie and David Heckerman. Ludicrous speed linear mixed models for genome-wide association studies. *BioRxiv*, page 154682, 2017.

[151] Kaihe Xu, Hao Yue, Linke Guo, Yuanxiong Guo, and Yuguang Fang. Privacy-preserving machine learning algorithms for big data systems. *2015 IEEE 35th international conference on distributed computing systems*, pages 318–327, 2015.

[152] Chris Jackson. sparse-dot-mkl: Intel mkl wrapper for sparse matrix multiplication. https://github.com/flatironinstitute/sparse_dot, 2023. Accessed: September 2023.

[153] Montserrat Garcia-Closas, Yuanqing Ye, Nathaniel Rothman, Jonine D Figueroa, Núria Malats, Colin P Dinney, Nilanjan Chatterjee, Ludmila Prokunina-Olsson, Zhaoming Wang, Jie Lin, et al. A genome-wide association study of bladder cancer identifies a new susceptibility locus within slc14a1, a urea transporter gene on chromosome 18q12. 3. *Human molecular genetics*, 20(21):4282–4289, 2011.

[154] Nathaniel Rothman, Montserrat Garcia-Closas, Nilanjan Chatterjee, Nuria Malats, Xifeng Wu, Jonine D Figueroa, Francisco X Real, David Van Den Berg, Giuseppe Matullo, Dalsu Baris, et al. A multi-stage genome-wide association study of bladder cancer identifies multiple susceptibility loci. *Nature genetics*, 42(11):978–984, 2010.

[155] Jonine D Figueroa, Yuanqing Ye, Afshan Siddiq, Montserrat Garcia-Closas, Nilanjan Chatterjee, Ludmila Prokunina-Olsson, Victoria K Cortessis, Charles Kooperberg, Olivier Cussenot, Simone Benhamou, et al. Genome-wide association study identifies multiple loci associated with bladder cancer risk. *Human molecular genetics*, 23(5): 1387–1398, 2014.

[156] Lars G Fritsche, Wilmar Igl, Jessica N Cooke Bailey, Felix Grassmann, Sebanti Sengupta, Jennifer L Bragg-Gresham, Kathryn P Burdon, Scott J Hebbring, Cindy Wen, Mathias Gorski, et al. A large genome-wide association study of age-related macular degeneration highlights contributions of rare and common variants. *Nature genetics*, 48(2):134–143, 2016.

[157] Daniel L Ayres, Aaron Darling, Derrick J Zwickl, Peter Beerli, Mark T Holder, Paul O Lewis, John P Huelsenbeck, Fredrik Ronquist, David L Swofford, Michael P Cummings, et al. Beagle: an application programming interface and high-performance computing library for statistical phylogenetics. *Systematic biology*, 61(1):170–173, 2012.

[158] Po-Ru Loh, Gleb Kichaev, Steven Gazal, Armin P Schoech, and Alkes L Price. Mixed-model association for biobank-scale datasets. *Nature genetics*, 50(7):906–908, 2018.

[159] Alkes L Price, Nick J Patterson, Robert M Plenge, Michael E Weinblatt, Nancy A Shadick, and David Reich. Principal components analysis corrects for stratification in genome-wide association studies. *Nature genetics*, 38(8):904–909, 2006.

[160] Thomas W Winkler, Felix R Day, Damien C Croteau-Chonka, Andrew R Wood, Adam E Locke, Reedik Mägi, Teresa Ferreira, Tove Fall, Mariaelisa Graff, Anne E Justice, et al. Quality control and conduct of genome-wide association meta-analyses. *Nature protocols*, 9(5):1192–1212, 2014.

[161] Oriol Canela-Xandri, Konrad Rawlik, and Albert Tenesa. An atlas of genetic associations in uk biobank. *Nature genetics*, 50(11):1593–1599, 2018.

[162] Momoko Horikoshi, Robin N Beaumont, Felix R Day, Nicole M Warrington, Marjolein N Kooijman, Juan Fernandez-Tajes, Bjarke Feenstra, Natalie R Van Zuydam, Kyle J Gaulton, Niels Grarup, et al. Genome-wide associations for birth weight and correlations with adult disease. *Nature*, 538(7624):248–252, 2016.

[163] Joanne B Cole, Jose C Florez, and Joel N Hirschhorn. Comprehensive genomic analysis of dietary habits in uk biobank identifies hundreds of genetic associations. *Nature communications*, 11(1):1467, 2020.

[164] Longda Jiang, Zhili Zheng, Ting Qi, Kathryn E Kemper, Naomi R Wray, Peter M Visscher, and Jian Yang. A resource-efficient tool for mixed model association analysis of large-scale data. *Nature genetics*, 51(12):1749–1755, 2019.

[165] Wei Zhou, Jonas B Nielsen, Lars G Fritsche, Rounak Dey, Maiken E Gabrielsen, Brooke N Wolford, Jonathon LeFaive, Peter VandeHaar, Sarah A Gagliano, Aliya Gifford, et al. Efficiently controlling for case-control imbalance and sample relatedness in large-scale genetic association studies. *Nature genetics*, 50(9):1335–1341, 2018.

[166] Christoph Lippert, Jennifer Listgarten, Ying Liu, Carl M Kadie, Robert I Davidson, and David Heckerman. Fast linear mixed models for genome-wide association studies. *Nature methods*, 8(10):833–835, 2011.

[167] KNUT M Wittkowski, VIKAS Sonakya, BENEDETTA Bigio, MONA KATHARINA Tonn, FREDERICK Shic, M Ascano, CARLA Nasca, Gold-Von Simson, et al. A novel computational biostatistics approach implies impaired dephosphorylation of growth factor receptors as associated with severity of autism. *Translational psychiatry*, 4(1): e354–e354, 2014.

[168] National Institutes of Health (NIH). Genomic Data Sharing (GDS) Policy. https://grants.nih.gov/grants/guide/notice-files/NOT-OD-14-124.html, 2014. Guide Notice NOT-OD-14-124. Effective January 25, 2015.

[169] Ryan Poplin, Valentin Ruano-Rubio, Mark A DePristo, Tim J Fennell, Mauricio O Carneiro, Geraldine A Van der Auwera, David E Kling, Laura D Gauthier, Ami Levy-Moonshine, David Roazen, et al. Scaling accurate genetic variant discovery to tens of thousands of samples. *biorxiv*, page 201178, 2017.

[170] Ryan Poplin, Pi-Chuan Chang, David Alexander, Scott Schwartz, Thomas Colthurst, Alexander Ku, Dan Newburger, Jojo Dijamco, Nam Nguyen, Pegah T Afshar, et al. A universal snp and small-indel variant caller using deep neural networks. *Nature biotechnology*, 36(10):983–987, 2018.

[171] Sairam Behera, Severine Catreux, Massimiliano Rossi, Sean Truong, Zhuoyi Huang, Michael Ruehle, Arun Visvanath, Gavin Parnaby, Cooper Roddey, Vitor Onuchic, et al. Comprehensive genome analysis and variant detection at scale using dragen. *Nature Biotechnology*, 43(7):1177–1191, 2025.

[172] A Adam Ding, Guanhong Miao, and Samuel S Wu. On the privacy and utility properties of triple matrix-masking. *The Journal of privacy and confidentiality*, 10(2), 2020.

[173] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[174] Arjhun Swaminathan, Anika Hannemann, Ali Burak Ünal, Nico Pfeifer, and Mete Akgün. Pp-gwas: Privacy preserving multi-site genome-wide association studies — code, 2025. URL https://doi.org/10.5281/zenodo.17580283.

[175] Richard Jozsa and Noah Linden. On the role of entanglement in quantum-computational speed-up. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 459(2036):2011–2032, 2003.

[176] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum algorithms for supervised and unsupervised machine learning. *arXiv preprint arXiv:1307.0411*, 2013.

[177] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum principal component analysis. *Nature physics*, 10(9):631–633, 2014.

[178] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017.

[179] Maria Schuld and Francesco Petruccione. *Supervised learning with quantum computers*, volume 17. Springer, 2018.

[180] Liming Zhao, Zhikuan Zhao, Patrick Rebentrost, and Joseph Fitzsimons. Compiling basic linear algebra subroutines for quantum computers. *Quantum Machine Intelligence*, 3(2):21, 2021.

[181] William K Wootters and Wojciech H Zurek. A single quantum cannot be cloned. *Nature*, 299(5886):802–803, 1982.

[182] Dik Bouwmeester, Jian-Wei Pan, Klaus Mattle, Manfred Eibl, Harald Weinfurter, and Anton Zeilinger. Experimental quantum teleportation. *Nature*, 390(6660):575–579, 1997.

[183] Charles H Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. *Theoretical computer science*, 560:7–11, 2014.

[184] Charles H Bennett, François Bessette, Gilles Brassard, Louis Salvail, and John Smolin. Experimental quantum cryptography. *Journal of cryptology*, 5:3–28, 1992.

[185] Gui-Lu Long and Xiao-Shu Liu. Theoretically efficient high-capacity quantum-key-distribution scheme. *Physical Review A*, 65(3):032302, 2002.

[186] Fu-Guo Deng, Gui Lu Long, and Xiao-Shu Liu. Two-step quantum direct communication protocol using the einstein-podolsky-rosen pair block. *Physical Review A*, 68(4): 042317, 2003.

[187] Chuan Wang, Fu-Guo Deng, Yan-Song Li, Xiao-Shu Liu, and Gui Lu Long. Quantum secure direct communication with high-dimension quantum superdense coding. *Physical Review A*, 71(4):044305, 2005.

[188] Manuel Eugenio Morocho-Cayamcela, Haeyoung Lee, and Wansu Lim. Machine learning for 5g/b5g mobile and wireless communications: Potential, limitations, and future directions. *IEEE access*, 7:137184–137206, 2019.

[189] Pedro Ponte and Roger G Melko. Kernel methods for interpretable machine learning of order parameters. *Physical Review B*, 96(20):205146, 2017.

[190] Xiaojian Ding, Jian Liu, Fan Yang, and Jie Cao. Random radial basis function kernel-based support vector machine. *Journal of the Franklin Institute*, 358(18):10121–10140, 2021.

[191] Siddhant Garg and Goutham Ramakrishnan. Advances in quantum deep learning: An overview. *arXiv preprint arXiv:2005.04316*, 2020.

[192] Yunseok Kwak, Won Joon Yun, Jae Pyoung Kim, Hyunhee Cho, Jihong Park, Minseok Choi, Soyi Jung, and Joongheon Kim. Quantum distributed deep learning architectures: Models, discussions, and applications. *ICT Express*, 9(3):486–491, 2023.

[193] Yijie Dang, Nan Jiang, Hao Hu, Zhuoxiao Ji, and Wenyin Zhang. Image classification based on quantum k-nearest-neighbor algorithm. *Quantum Information Processing*, 17:1–18, 2018.

[194] Afrad Basheer, A Afham, and Sandeep K Goyal. Quantum k-nearest neighbors algorithm. *arXiv preprint arXiv:2003.09187*, 2020.

[195] Vojtěch Havlíček, Antonio D Córcoles, Kristan Temme, Aram W Harrow, Abhinav Kandala, Jerry M Chow, and Jay M Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747):209–212, 2019.

[196] Hwanjo Yu, Xiaoqian Jiang, and Jaideep Vaidya. Privacy-preserving svm using nonlinear kernels on horizontally partitioned data. In *Proceedings of the 2006 ACM symposium on Applied computing*, pages 603–610, 2006.

[197] Yu-Bo Sheng and Lan Zhou. Distributed secure quantum machine learning. *Science Bulletin*, 62(14):1025–1029, 2017.

[198] Robert Wille, Rod Van Meter, and Yehuda Naveh. Ibm's qiskit tool chain: Working with and developing for real quantum computers. In *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1234–1240. IEEE, 2019.

[199] Kishor Bharti, Alba Cervera-Lierta, Thi Ha Kyaw, Tobias Haug, Sumner Alperin-Lea, Abhinav Anand, Matthias Degroote, Hermanni Heimonen, Jakob S Kottmann, Tim Menke, et al. Noisy intermediate-scale quantum algorithms. *Reviews of Modern Physics*, 94(1):015004, 2022.

[200] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. An introduction to quantum machine learning. *Contemporary Physics*, 56(2):172–185, 2015.

[201] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. *Physical review letters*, 113(13):130503, 2014.

[202] Maria Schuld. Supervised quantum machine learning models are kernel methods. *arXiv preprint arXiv:2101.11020*, 2021.

[203] Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, et al. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, 2021.

[204] David S. Broomhead and David Lowe. Radial basis functions, multi-variable functional interpolation and adaptive networks. 1988. URL https://api.semanticscholar.org/CorpusID:117200472.

[205] Dennis D Wackerly, William Mendenhall, and Richard L Scheaffer. *Mathematical statistics with applications*, volume 7. Thomson Brooks/Cole Belmont, CA, 2008.

[206] Alexander J Smola and Risi Kondor. Kernels and regularization on graphs. In *Learning Theory and Kernel Machines: 16th Annual Conference on Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24-27, 2003. Proceedings*, pages 144–158. Springer, 2003.

[207] John P Nolan. *Stable distributions*. 2012.

[208] Junfeng Fan and Frederik Vercauteren. Somewhat practical fully homomorphic encryption. *Cryptology ePrint Archive*, 2012.

[209] James Stirling. *Methodus differentialis: sive tractatus de summatione et interpolatione serierum infinitarum*. 1730.

[210] Edward Fredkin and Tommaso Toffoli. Conservative logic. *International Journal of theoretical physics*, 21(3):219–253, 1982.

[211] Adriano Barenco, Andre Berthiaume, David Deutsch, Artur Ekert, Richard Jozsa, and Chiara Macchiavello. Stabilization of quantum computations by symmetrization. *SIAM Journal on Computing*, 26(5):1541–1557, 1997.

[212] Harry Buhrman, Richard Cleve, John Watrous, and Ronald De Wolf. Quantum fingerprinting. *Physical review letters*, 87(16):167902, 2001.

[213] Arthur Asuncion, David Newman, et al. Uci machine learning repository, 2007.

[214] C Okan Sakar, Gorkem Serbes, Aysegul Gunduz, Hunkar C Tunc, Hatice Nizam, Betul Erdogdu Sakar, Melih Tutuncu, Tarkan Aydin, M Erdem Isenkul, and Hulya Apaydin. A comparative analysis of speech signal processing algorithms for parkinson's disease classification and the use of the tunable q-factor wavelet transform. *Applied Soft Computing*, 74:255–263, 2019.

[215] Ashish Bhardwaj. Framingham heart study dataset, 2022. URL https://www.kaggle.com/dsv/3493583.

[216] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[217] Sergei Bernstein. On a modification of chebyshev's inequality and of the error formula of laplace. *Ann. Sci. Inst. Sav. Ukraine, Sect. Math*, 1(4):38–49, 1924.

[218] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[219] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *Proceedings of the IEEE international conference on computer vision*, pages 2722–2730, 2015.

[220] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *nature*, 542(7639):115–118, 2017.

[221] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[222] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[223] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *International conference on machine learning*, pages 2137–2146. PMLR, 2018.

[224] Minhao Cheng, Simranjit Singh, Patrick Chen, Pin-Yu Chen, Sijia Liu, and Cho-Jui Hsieh. Sign-opt: A query-efficient hard-label adversarial attack. *arXiv preprint arXiv:1909.10773*, 2019.

[225] Shuyu Cheng, Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Improving black-box adversarial attacks with a transfer-based prior. *Advances in neural information processing systems*, 32, 2019.

[226] Viet Quoc Vo, Ehsan Abbasnejad, and Damith C Ranasinghe. Ramboattack: A robust query efficient deep neural network decision exploit. *arXiv preprint arXiv:2112.05282*, 2021.

[227] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pages 15–26, 2017.

[228] Minhao Cheng, Thong Le, Pin-Yu Chen, Huan Zhang, JinFeng Yi, and Cho-Jui Hsieh. Query-efficient hard-label black-box attack: An optimization-based approach. In *International Conference on Learning Representations*, 2018.

[229] Thomas Brunner, Frederik Diehl, Michael Truong Le, and Alois Knoll. Guessing smart: Biased sampling for efficient black-box adversarial attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4958–4966, 2019.

[230] Fnu Suya, Jianfeng Chi, David Evans, and Yuan Tian. Hybrid batch attacks: Finding black-box adversarial examples with limited queries. In *29th USENIX security symposium (USENIX Security 20)*, pages 1327–1344, 2020.

[231] Huichen Li, Xiaojun Xu, Xiaolu Zhang, Shuang Yang, and Bo Li. Qeba: Query-efficient boundary-based blackbox attack. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1221–1230, 2020.

[232] Yujia Liu, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. A geometry-inspired decision-based attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4890–4898, 2019.

[233] Chen Ma, Xiangyu Guo, Li Chen, Jun-Hai Yong, and Yisen Wang. Finding optimal tangent points for reducing distortions of hard-label attacks. *Advances in Neural Information Processing Systems*, 34:19288–19300, 2021.

[234] Yusuke Tashiro, Yang Song, and Stefano Ermon. Diversity can be transferred: Output diversification for white-and black-box attacks. *Advances in neural information processing systems*, 33:4536–4548, 2020.

[235] Xiaosen Wang, Zeliang Zhang, Kangheng Tong, Dihong Gong, Kun He, Zhifeng Li, and Wei Liu. Triangle attack: A query-efficient decision-based adversarial attack. In *European conference on computer vision*, pages 156–174. Springer, 2022.

[236] Irvin Sobel. Neighborhood coding of binary images for fast contour following and general binary array processing. *Computer graphics and image processing*, 8(1):127–135, 1978.

[237] Robert Bassett, Mitchell Graves, and Patrick Reilly. Color and edge-aware adversarial image perturbations. *arXiv preprint arXiv:2008.12454*, 2020.

[238] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pages 818–833. Springer, 2014.

[239] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *International conference on learning representations*, 2018.

[240] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[241] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[242] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[243] Logan Engstrom, Andrew Ilyas, Hadi Salman, Shibani Santurkar, and Dimitris Tsipras. Robustness (python library), 2019. URL https://github.com/MadryLab/robustness.

[244] Qi-An Fu, Yinpeng Dong, Hang Su, Jun Zhu, and Chao Zhang. {AutoDA}: Automated decision-based iterative adversarial attacks. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 3557–3574, 2022.

[245] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[246] Lin Zhang, Lei Zhang, Xuanqin Mou, and David Zhang. Fsim: A feature similarity index for image quality assessment. *IEEE transactions on Image Processing*, 20(8): 2378–2386, 2011.

[247] Sid Ahmed Fezza, Yassine Bakhti, Wassim Hamidouche, and Olivier Déforges. Perceptual evaluation of adversarial attacks for cnn-based image classification. In *2019 eleventh international conference on quality of multimedia experience (QoMEX)*, pages 1–6. IEEE, 2019.

[248] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[249] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[250] Intel and Analytics Vidhya. Intel image classification (scene classification challenge). https://www.kaggle.com/datasets/puneet6060/intel-image-classification, 2018. Originally released as the Intel Scene Classification Challenge on Analytics Vidhya.

[251] A. Swaminathan and M. Akgün. Accelerating targeted hard-label adversarial attacks in low-query black-box settings - code, 2026. URL https://doi.org/10.5281/zenodo.18744010.

[252] Pierangela Samarati. Protecting respondents identities in microdata release. *IEEE transactions on Knowledge and Data Engineering*, 13(6):1010–1027, 2001.

[253] Latanya Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):571–588, 2002.

[254] Yan Sun, Lihua Yin, Licai Liu, and Shuang Xin. Toward inference attacks for k-anonymity. *Personal and ubiquitous computing*, 18:1871–1880, 2014.

[255] Kenta Kitamura, Mhd Irvan, and Rie Shigetomi Yamaguchi. Disclosure of multiple" patient characteristics" format statistics leaks quasi-identifier linkage. In *Proceedings of the 9th ACM International Workshop on Security and Privacy Analytics*, pages 15–25, 2023.

[256] Xiaokui Xiao and Yufei Tao. Anatomy: Simple and effective privacy preservation. In *Proceedings of the 32nd international conference on Very large data bases*, pages 139–150, 2006.

[257] Adam Meyerson and Ryan Williams. On the complexity of optimal k-anonymity. In *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 223–228, 2004.

[258] Jinkyung Park, Eiman Ahmed, Hafiz Asif, Jaideep Vaidya, and Vivek Singh. Privacy attitudes and covid symptom tracking apps: Understanding active boundary management by users. In *International Conference on Information*, pages 332–346. Springer, 2022.

[259] Junade Ali and Vladimir Dyo. Cross hashing: Anonymizing encounters in decentralised contact tracing protocols. In *2021 International Conference on Information Networking (ICOIN)*, pages 181–185. IEEE, 2021.

[260] Pietro Tedeschi, Spiridon Bakiras, and Roberto Di Pietro. Iotrace: a flexible, efficient, and privacy-preserving iot-enabled architecture for contact tracing. *IEEE Communications Magazine*, 59(6):82–88, 2021.

[261] Serge Vaudenay. Analysis of dp3t-between scylla and charybdis. 2020.

[262] Jinbao Wang, Zhipeng Cai, and Jiguo Yu. Achieving personalized k-anonymity-based content privacy for autonomous vehicles in cps. *IEEE Transactions on Industrial Informatics*, 16(6):4242–4251, 2019.

[263] Jenno Verdonck, Kevin De Boeck, Michiel Willocx, Jorn Lapon, and Vincent Naessens. A hybrid anonymization pipeline to improve the privacy-utility balance in sensitive datasets for ml purposes. In *Proceedings of the 18th International Conference on Availability, Reliability and Security*, pages 1–11, 2023.

[264] Rudolf Mayer, Alicja Karlowicz, and Markus Hittmeir. K-anonymity on metagenomic features in microbiome databases. In *Proceedings of the 18th International Conference on Availability, Reliability and Security*, pages 1–11, 2023.

[265] Stephen E Fienberg, Aleksandra Slavkovic, and Caroline Uhler. Privacy preserving gwas data sharing. In *2011 IEEE 11th International Conference on Data Mining Workshops*, pages 628–635. IEEE, 2011.

[266] André Calero Valdez and Martina Ziefle. The users' perspective on the privacy-utility trade-offs in health recommender systems. *International Journal of Human-Computer Studies*, 121:108–121, 2019.

[267] Clément Maria, Jean-Daniel Boissonnat, Marc Glisse, and Mariette Yvinec. The gudhi library: Simplicial complexes and persistent homology. In *Mathematical Software–ICMS 2014: 4th International Congress, Seoul, South Korea, August 5-9, 2014. Proceedings 4*, pages 167–174. Springer, 2014.

[268] Ulrich Bauer, Michael Kerber, Jan Reininghaus, and Hubert Wagner. Phat–persistent homology algorithms toolbox. *Journal of symbolic computation*, 78:76–90, 2017.

[269] David Cohen-Steiner, Herbert Edelsbrunner, and Dmitriy Morozov. Vines and vineyards by updating persistence in linear time. In *Proceedings of the twenty-second annual symposium on Computational geometry*, pages 119–126, 2006.

[270] Kristen LeFevre, David J DeWitt, and Raghu Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 49–60, 2005.

[271] Kristen LeFevre, David J DeWitt, and Raghu Ramakrishnan. Mondrian multidimensional k-anonymity. In *22nd International conference on data engineering (ICDE'06)*, pages 25–25. IEEE, 2006.

[272] Josep Domingo-Ferrer and Vicenç Torra. Ordinal, continuous and heterogeneous k-anonymity through microaggregation. *Data Mining and Knowledge Discovery*, 11: 195–212, 2005.

[273] Jordi Soria-Comas and Josep Domingo-Ferrer. Probabilistic k-anonymity through microaggregation and data swapping. In *2012 IEEE International Conference on Fuzzy Systems*, pages 1–8. IEEE, 2012.

[274] Jordi Soria-Comas, Josep Domingo-Ferrer, David Sánchez, and Sergio Martínez. Enhancing data utility in differential privacy via microaggregation-based k-anonymity. *The VLDB Journal*, 23(5):771–794, 2014.

[275] Bugra Gedik and Ling Liu. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *IEEE Transactions on Mobile Computing*, 7(1):1–18, 2007.

[276] Marco Gruteser and Dirk Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the 1st international conference on Mobile systems, applications and services*, pages 31–42, 2003.

[277] Mohamed F Mokbel, Chi-Yin Chow, and Walid G Aref. The new casper: Query processing for location services without compromising privacy. In *VLDB*, volume 6, pages 763–774, 2006.

[278] Kyriakos Mouratidis and Man Lung Yiu. Anonymous query processing in road networks. *IEEE Transactions on Knowledge and Data Engineering*, 22(1):2–15, 2009.

[279] Xiong Wang, Zhe Liu, Xiaohua Tian, Xiaoying Gan, Yunfeng Guan, and Xinbing Wang. Incentivizing crowdsensing with location-privacy preserving. *IEEE Transactions on Wireless Communications*, 16(10):6940–6952, 2017.

[280] Ji-Won Byun, Yonglak Sohn, Elisa Bertino, and Ninghui Li. Secure anonymization for incremental datasets. In *Secure Data Management: Third VLDB Workshop, SDM 2006, Seoul, Korea, September 10-11, 2006. Proceedings 3*, pages 48–63. Springer, 2006.

[281] Xiaokui Xiao and Yufei Tao. M-invariance: towards privacy preserving re-publication of dynamic datasets. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 689–700, 2007.

[282] Julián Salas and Vicenç Torra. A general algorithm for k-anonymity on dynamic databases. In *Data Privacy Management, Cryptocurrencies and Blockchain Technology: ESORICS 2018 International Workshops, DPM 2018 and CBT 2018, Barcelona, Spain, September 6-7, 2018, Proceedings 13*, pages 407–414. Springer, 2018.

[283] Allen Hatcher. *Algebraic topology*. 2005.

[284] Edelsbrunner, Letscher, and Zomorodian. Topological persistence and simplification. *Discrete & Computational Geometry*, 28:511–533, 2002.

[285] Gunnar Carlsson, Vin De Silva, and Dmitriy Morozov. Zigzag persistent homology and real-valued functions. In *Proceedings of the twenty-fifth annual symposium on Computational geometry*, pages 247–256, 2009.

[286] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Extending persistence using poincaré and lefschetz duality. *Foundations of Computational Mathematics*, 9 (1):79–103, 2009.

[287] Alan Bundy and Lincoln Wallen. Breadth-first search. *Catalogue of artificial intelligence tools*, pages 13–13, 1984.

[288] Emo Welzl. Smallest enclosing disks (balls and ellipsoids). In *New Results and New Trends in Computer Science: Graz, Austria, June 20–21, 1991 Proceedings*, pages 359–370. Springer, 2005.

[289] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. In *Proceedings of the twenty-first annual symposium on Computational geometry*, pages 263–271, 2005.

[290] Latanya Sweeney. Guaranteeing anonymity when sharing medical data, the datafly system. In *Proceedings of the AMIA Annual Fall Symposium*, page 51. American Medical Informatics Association, 1997.

[291] A. Swaminathan and M. Akgün. Dynamic k-anonymity for electronic health records: A topological framework - code, 2024. URL https://doi.org/10.5281/zenodo.18744146.