Better Incentives: Performant and Private Machine Learning

by

Mimee Xu

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

New York University

August, 2025

Professor Léon Bottou

To Skai, who taught me how to tie shoelaces.

Abstract

Machine learning algorithms benefit from large and diverse datasets. However, business needs and research workflows are potentially at odds with the ownership of private data. Without sharing private data in their raw forms, current privacy-enhancing solutions tend to, instead, compromise on performance (by degrading downstream models) or privacy (by revealing potentially sensitive information).

This thesis addresses gaps between machine learning and data ownership, through modeling a system of three parties: model owners, data owners, and overseers. Incentive issues between the parties are addressed with secure and confidential computation, consisting of Secure-Multiparty Computation (S-MPC) and Homomorphic Encryption (FHE). Though lesser-known to machine learning, these techniques can help support data rights.

1. First, as data used for training tends to be owned by disparate parties, the first sub-problem pertains to whether *unshared* training data's utility can be evaluated without sharing it. This thesis formulates the Data Appraisal Problem, and attacks it with an efficient and accurate privacy-preserving proposal. Specifically, our work implemented influenced-based appraisal functions written to be compatible with efficient S-MPC computation, so that no data needs to be shared for both the model owner and data owner to approximate the relative value of datasets with respect to a model owner's choice of model and test set. It achieves 92.3% correlation with plaintext ground truth ranking for 100 datasets under induced class imbalance, and 96.0% under label-flipping, without the usability challenge of sensitive hyperparameters of training a joint model

under S-MPC [320].

2. Secondly, is it necessary to trade off data utility and privacy in low data domains? As hospitals face severe data availability issues for model training, we seek to ease the privacy tension that thwarts collaborations, specifically before making a commitment. This practical framework, **Secure-KL** (SKL), only releases the output evaluation score while enabling robust evaluation of additional data to combine with. Without making any assumptions about the final downstream model, using only the source and target datasets, SKL is a model-based data-divergence approximation in secure computation, which matches plaintext values by over 90%. For the source hospital, SKL successfully identifies beneficial data partnerships for intensive care unit (ICU) mortality prediction, improving downstream classifier performance. It is more robust and reliable than alternatives of sharing a subset of data (medium leakage), using demographic information (low leakage), or selecting blind (high variance). SKL allows all parties' data to remain *unshared*[1], while *entire* datasets are utilized, effectively eliminating a key roadblock towards orchestrating broader collaborations in healthcare with limited resources (To appear in AAAI AIES 25' [106]).

3. Lastly, seeing the trend of deploying proprietary ML models where the input and output to those models are hidden, can the public audit privately-held data, especially in domains where encryption is the default? Using FHE for auditing triaging fairness in hospitals' emergency department, as an example, this thesis provides a qualitative description of the setup that can be applied to ease the tension between regulators and private data parties, without the need to decrypt private data, utilizing Threshold FHE [78]. (Published as a book chapter [175].)

[1]More precisely, outside the final result both parties intend to compute, no additional data is revealed.

Contents

Dedication iii

Abstract iv

List of Figures xii

List of Tables xxi

1 Introduction 1

1.1 Summary and Overview 5

1.1.1 Summary of Existing Gaps in ML Privacy 5

1.1.2 Contributions Overview 5

1.1.3 Summary of Chapters 6

2 The Three-Actor Privacy Thesis 8

2.1 Chapter Summary 8

2.2 Optimisation Setup 10

2.3 The Three-Actor Ecosystem for Machine Learning 11

2.3.1 Discussion 12

2.4 A Conceptual Framework For Privacy Tensions Through A Sharing Conundrum . 14

2.4.1 Two Approaches: Influence and Divergence 15

2.5 Problematizing Dataset Acquisition Conundrum . . . 18
2.5.1 Baseline Discussion . . . 20
2.5.2 Assumptions . . . 20
2.5.3 Evaluation Model . . . 21
2.6 Significance of The Three Actor Ecosystem . . . 23
2.6.1 Systemic Privacy Tension . . . 23
2.6.2 Evidence for Existing Tension . . . 24
2.6.3 Limitations of Technical Solutions . . . 25
2.6.4 Contribution: Expanding Privacy Definitions . . . 26
2.6.5 Conclusion . . . 27

3 PrivacyML: Building Blocks for Privacy for Machine Learning 28

3.1 Chapter Overview . . . 28
3.2 Introduction: Privacy for Machine Learning . . . 30
3.2.1 Machine Learning with Secure and Confidential Computation . . . 32
3.3 Homomorphic Encryption . . . 35
3.3.1 Homomorphic Encryption, An Informal Primer . . . 36
3.3.2 Distributed FHE For Multiple Parties . . . 37
3.3.3 Engineering Paths . . . 38
3.3.4 Situating Our Contribution: Hospital Fairness Auditing . . . 41
3.4 Secure Multi-party Computation (MPC) . . . 42
3.4.1 Multiparty Computation for Machine Learning . . . 42
3.5 Engineering Secure Computation, In Detail . . . 46
3.5.1 Explaining the Efficacy of Influence-Based Appraisal in MPC . . . 47
3.5.2 Choosing CrypTen For Private Data Appraisal . . . 48
3.5.3 Machine Learning Engineering Discussions . . . 49

3.5.4 Parting Thoughts On Secure Computation for Practical Systems 50
3.5.5 Summary 51
3.6 Privacy-Enhancing Techniques 52
3.6.1 Federated Learning (FL). 52
3.6.2 Differential Privacy (DP). 55
3.6.3 Other Privacy-Enhancing Techniques 56
3.7 Conclusion 60

4 Data Appraisal Without Data Sharing **61**
4.1 Problem Setup 62
4.2 Private Data Appraisal Functions 63
4.2.1 Forward Influence in Multi-party Computation. 64
4.3 Experimental Results 65
4.3.1 In MPC, Forward Influence Functions Are More Usable Than Finetuning . 66
4.3.2 Forward Influence Recovers Ranking Under Noise and Imbalance 68
4.3.3 Applying Influence Appraisal On Corrupted Cancer Patient Data 70
4.4 Related Work 73
4.5 Limitations 75
4.6 Engineering Contributions 76
4.7 Summary 78

5 Dataset-to-Dataset Evaluation Before Data Sharing **79**
5.1 Introduction 80
5.2 Context and Contributions 83
5.3 Setup 87
5.4 Data Acquisition Strategies, in Detail 91
5.4.1 Baseline Strategies Categorized 91

5.4.2 Trivially Private Baseline: Blind Selection . . . 92
5.4.3 Sharing Summary Statistics . . . 92
5.4.4 KL-based Methods, Without Privacy . . . 92
5.4.5 SecureKL: Private KL-based Method . . . 94
5.5 Experimental Setup . . . 96
5.5.1 Experimental Questions . . . 96
5.5.2 MPC implementation. . . . 97
5.5.3 Data and Model Setup . . . 98
5.6 Results and Analysis . . . 99
5.6.1 Consistency Between Plaintext and Encrypted Computations . . . 99
5.6.2 Positivity in Realistic Setup . . . 100
5.6.3 SecureKL Error Analysis . . . 102
5.7 Discussion . . . 105
5.7.1 SecureKL Contribution . . . 105
5.7.2 Potential Challenges to Broader Adoption . . . 106
5.8 Engineering Contribution . . . 108
5.9 Alternative Approaches . . . 110
5.10 Conclusion . . . 114

6 HEalth: Privately Computing on Shared Healthcare Data 115

6.1 Motivation and Background . . . 119
6.2 Summary of Engineering Contribution . . . 122
6.3 Problem Setup . . . 124
6.3.1 Auditing Setups . . . 124
6.4 Algorithms . . . 126

6.4.1 Proposed Applications . . . 130
6.5 Comparing with Differentially Private Model Releases . . . 131
6.6 Overview of the Protocol Setup . . . 133
6.6.1 Setup Assumptions for Hospitals . . . 133
6.7 Conclusion . . . 135

7 Towards ML-Privacy Co-design, A Paradigm Shift 136
7.1 Adapting Data Valuation Techniques for Unseen Data . . . 137
7.2 Gap and Opportunity for MPC for ML . . . 138
7.3 Supporting Optimization–Privacy Co-Design . . . 139

8 Conclusion 142
8.1 Incentive-aligned Privacy for Machine Learning . . . 142
8.1.1 Alternative Views . . . 143
8.2 Key Contributions . . . 144

A Three-Actor Privacy 146
A.1 Dataset Divergence Technical Discussion . . . 146

B PrivacyML 147
B.1 Input-Output Privacy . . . 147
B.2 Contextual Integrity Relation . . . 149
B.3 Further Readings on ML MPC SOTA . . . 150
B.4 Policy Implications for Machine Learning Privacy . . . 151

C Data Appraisal 153
C.1 Forward Influence Details . . . 153

D Dataset-to-Dataset Evaluations **157**

D.1 Methodological Details . . . 158

D.2 Correlation with downstream performance . . . 159

D.3 Hyperparameter Tuning . . . 160

D.4 Correlations between Encrypted Scores and Plaintext Scores . . . 161

D.5 Performance . . . 162

D.6 Folktables Experiments . . . 163

E HEalth **165**

E.1 Privacy Challenges for Auditing Healthcare Data . . . 165

E.2 Protocol Details . . . 166

E.3 Data Model Snippet . . . 168

E.4 Comparing Alternative Auditing Setups . . . 169

Bibliography **171**

List of Figures

1.1 Incentives misalignment creates friction between the three categories of stakeholders. 2

1.2 Individual works in this thesis, visualized within three-actor system. 4

2.1 Model Owner (*MO*) and Data Owner (*DO*) Relationship: Both can benefit from Share(MO, DO). 11

2.2 **Three-Actor Ecosystem** for machine learning clarifies entities with distinct goals, each wanting to gain utility while maintaining privacy. This abstraction forms the foundation of the privacy notions in this thesis; privacy issues are described as data-related tension in this system. 13

2.3 **Secure Computation on Disparately-owned Data for Machine Learning**, using secure inference between data and model as an example. In this thesis, secure computation is the primary approach towards respecting both model utility and privacy among distinct actors. 13

2.4 Model owner wants to perform pairwise evaluations *before* committing to a data partnership. 18

2.5 **Random selection may harm**: Real-world data collaborations are inherently uncertain, necessitating pre-partnership selection. This illustrates the Dataset Combination Problem studied in Chapter 5. AUC change for a source entity (x-axis), after incorporating external data from a different source (y-axis), across hospitals (left) and states (right). **Left**: in mortality prediction using eICU data [244], 10 out of 12 hospitals may see their mortality prediction model degrade for *some* potential hospital partners. **Right**: in income prediction using Folktables [83], combining with random state leads to worse prediction in 7 out of 12 states. (Red indicates bad collaborations.) 19

3.1 **A Privacy Desideratum for Machine Learning, Illustrated.** Performing joint computation $\mathcal{F}$ on data owned by a model owner (MO) and a data owner (DO), in order to facilitate data sharing Share(MO, DO) in Section 2.3. Ideally, each entity does not reveal their data throughout the computation. 30

3.2 **In Secure Machine Learning, Preset Hyperparameters Affect Utility and Computational Overhead.** When fine-tuning in private to acquire additional data, i.e., re-training after folding in $\mathcal{D}_a$, hyperparameters that result in high utility (Beige) tend to be computationally expensive (Red). **Top**: data appraisal correlation with ground truth plaintext training, with respect to different pre-set batch size and epoch (ρ, higher is better). **Bottom**: convergence in log-steps (log SGD steps, smaller is more efficient). Experiments are run on MNIST under secure SGD using CrypTen [164]. 33

3.3 Federated Learning with Data Distribution. Each client owns data, and the server "pools" their data. Naively, all data is sent to the server. In federated learning, to minimize communication, no raw data is sent – instead, parameter updates between the server and the clients are sent intermittently. This respects data locality, but still confers information between the parties. 52

4.1 **Incentive deadlock.** By default, model owner's parameters, θ, $\mathcal{D}_{\text{te}}$ and $\mathcal{D}_{\text{tr}}$, and data owners' $\{\mathcal{D}_{\text{a}}\}$'s are kept private. While both may gain from exchanging data, utility is not realized due to privacy. 61

4.2 **Secure MPC**. Data appraisal without data sharing. model owner and data owner encrypt their respective data. The appraisal function is performed privately, and its result is revealed to both parties. 62

4.3 **Finetuning-based Appraisal Trades Off Efficiency and Accuracy Due to Hyperparameters.**Correlation of appraisal with utility (top; purple is lower) and runtime (bottom; blue is faster) for finetuning hyperparameters batch size configuration (x-axis) and epochs (y-axis; logarithmic). This reproduces Figure 3.2. 67

4.4 **Gradient Norm is an unreliable value proxy.** Gradient norm appraisal and test loss reduction as a function of MNIST label noise (top, $\rho = -1$) and CIFAR-10 plane-to-car class balance (bottom, $\rho = 1$). 68

4.5 **Log Scale Runtimes Spent On Plaintext And Encrypted Computations For All Three Appraisal Methods.** Influence-based appraisal achieves low amortized timing as its plaintext overhead (purple) is fixed. 68

4.6 **Left a-b:** Rank of influence-based appraisal $f_{\text{if}}(\mathcal{D}_{\text{a}})$ (y-axis) as a function of the rank of the utility (a; $\rho = 0.923$) and the test accuracy (b; $\rho = -0.927$) on CIFAR-10's plane-to-car dataset. **Right c-d:** Rank of $f_{\text{if}}(\mathcal{D}_{\text{a}})$ as a function of the rank of the utility on CIFAR-10 dataset for which the rate of cars is in the range $[0, 0.45]$ (c; $\rho = 0.908$) and $[0.55, 1.0]$ (d; $\rho = 0.247$). Each dot is a sampled dataset, colored according to the ratio of the under-sampled class in $\mathcal{D}_{\text{a}}$. 69

4.7 **Influence-based appraisal makes reasonable appraisal ranking compared to ground truth utility (WDBC).** The rank of appraised values (y-axis) as a function of the rank data utility (x-axis) with varying data corruptions. The noiseless datasets (a-b) are benchmarked under 30 features and 20 features. The noisy datasets (c-d) are colored with noise level as a fraction of each dataset's label flips between "Benign" and "Malignant", and retain all features. 71

4.8 **Influence excels random strategy at high noise levels**, when applied sequentially select datasets without replacement for breast cancer diagnostics data. The change in test loss (y-axis) as a function of repeated rounds of data inclusion under varying noise levels. **Random**: choose a random dataset at each round. **Influence**: choose the dataset with the highest influence-based appraisal. For each graph, test loss change is normalized by the maxmimum test reduction in the control group. Averages and variances are taken over 5 runs. 72

5.1 **Privacy can dis-incentivize data collaborations.** Without seeing external data, an organization has two strategies: **i. blind default** π_0: randomly selecting partnerships causes hesitation and hinders partnerships. **ii. Our method** π_p: securely assessing datasets by leveraging MPC *before* commitment. 80

5.2 **Real-world data collaborations are inherently uncertain, necessitating pre-partnership selection** (a *reproduction* of Figure 2.5). AUC change for a source entity (x-axis), after incorporating external data from a different source (y-axis), across hospitals (left) and states (right). **Left**: in mortality prediction using eICU data [244], 10 out of 12 hospitals may see their mortality prediction model degrade for *some* potential hospital partners. **Right**: in income prediction using Folktables [83], combining with random state leads to worse prediction in 7 out of 13 states. (Red is bad). 83

5.3 **Abstraction of non-private evaluation strategies**, following the privacy-preserving methods in Figure 5.1 **iii**. **subset sampling** π_s: a subset of the target's data is shared. **iv**. **demographic-based summaries** π_d: the target entity discloses distributions by protected attributes, i.e., age, gender, or race. In these scenarios, some data is leaked by the target partner (yellow) to assuage the source entity (blue)'s uncertainties about the target data's value, trading off sample utility for limited privacy. 84

5.4 **Our Method** SecureKL($\mathcal{D}_o, \mathcal{D}_i$). Each side encrypts their data. Without assuming the *downstream* model, a dataset comparison model is trained on their joint data, typically a membership inference model using logistic regression (Section 5.3), which enables computing KL-based measures in private. Then their divergence is assessed, and the final result is revealed after both parties participate in decryption. 85

5.5 **AUC change δ_T over all strategies in eICU prediction** (higher is better). Our private dataset evaluation strategy π_p outperforms demographic-based strategy π_d **(left)**, and sub-sampling strategy π_s for $k = 300$ (10%) and $k = 30$ (1%) **(right)**, after combining source data with the top 3 candidates. 100

5.6 **AUC change δ_T over all strategies in Folktables dataset prediction** (higher is better). All strategies exhibit comparable distributions, after combining data from top 3 candidates. In a noisy domain, our method is stable: it neither excels nor penalizes against non-private strategies. 100

5.7 **Correlation Between AUC Change δ_i and $\mathrm{KL}_{\mathcal{XY}}$ Impacts $\mathrm{SecureKL}_{\mathcal{XY}}$'s Efficacy**. For Hospital 420, underlying $\mathrm{KL}_{\mathcal{XY}}$ identifies beneficial data ($\rho < 0$). For Hospital 243, $\mathrm{KL}_{\mathcal{XY}}$ fails to select effective data candidates ($\rho > 0$). 103

5.8 **Left**: $\mathrm{SecureKL}_{\mathcal{XY}}$ outperforms $\pi_s(k = 30)$ and π_0. **Middle**: All strategies perform similarly. **Right**: $\pi_s(k = 300)$ outperforms $\mathrm{SecureKL}_{\mathcal{XY}}$. (Bars represent standard deviations) 103

6.1 **Setup**: Hospital systems will deploy machine learning-based decision models, the outcome of which should be audited. The Overseers from Section 2.4 take on the role of regulatory auditors, who would need to access the input and the decisions made by these models, which are kept private by default at the hospital level. . . 116

6.2 **Fairness-Privacy Trade-off: per hospital group fairness performance under differentially private models with increasing privacy relaxations.** The fairness metric, worst group performance (higher is better), under different privacy parameter $\epsilon = 0.1, 1, 10$ (smaller is more privacy-preserving) setups. $\epsilon = \infty$ refers to no privacy protection (ground truth) numbers. Models are trained with logistic regression under DP-SGD (Sec 3.6.2). Group fairness shown for intersectional groups [48]. *Note*: This experiment is run in 2025 on eICU data for the 24 -hour mortality prediction task in [244]; our data processing is detailed in Appendix D.1. 121

6.3 **Existing: Hospital self-reports final metric.** The hospital computes fairness function $\mathcal{F}$ from raw data, and reports the number to the auditor. 124

6.4 **"Backdoor": Auditor decrypts encrypted data.** The hospital sends encrypted data as well and the key to fully decrypt the data, including sensitive patient data. The auditor then computes $\mathcal{F}$ from raw data. 124

6.5 **Our Method: Homomorphic Encryption-enabled Fairness Audit.** First, the hospital encrypts data (with a special key). The encrypted data is sent to the auditor, who cannot decrypt the data except for the fairness results. 125

6.6 **Regulatory mandates at odds with privacy compliance.** Per HIPAA [53], patient data samples, $\{H_i\}$, should be stored and transported with *industrial-grade encryption*. Regulatory bodies need to ensure fair access to medical resources via auditing the decisions and outputs (methodologies $f_{\text{audit}}, f_{\text{eval}}, \cdots$), but can only do so via a backdoor (*key*), undermining encryption. 127

6.7 **Secure MPC via Threshold FHE for Auditing.** The hospitals' data H_i, each encrypted with a *unique* set of keys, is continuously audited for fairness. The regulator does not hold any of the hospitals' keys, and data is destroyed if a private key is destroyed. The results and only the results, are revealed when an auditor calls for a decryption event (not pictured). 127

6.8 **Preprocessing and Secure Aggregation Overview.** Each hospital preprocesses their data with the *same* binning and quantization function $Q(\cdot)$, and encrypts their resulting histogram H_i to the server. The server performs aggregation of the histograms, outlined in Section 6.4, to obtain aggregate histogram $\mathcal{H}$ through Aggregating Hospital Histograms. Note that H_i's are encrypted before entering auditing compute, so no unencrypted data is shared. 127

6.9 **Fairness Audit Request Overview.** After data has been aggregated, hospitals and overseers get together to answer to a *request-to-decrypt* that is specific to the fairness function in question, such as group fairness difference between races, as outlined in Section 6.4. Note that appropriate decoding, such as rescaling, is needed after decryption. For our protocol, all parties need to agree to decrypt the results. 127

D.1 In Folktables [83], combining with random state leads to worse income prediction in 23 out of 35 states. 163

D.2 **SecureKL: Overall Correctness**. Rank correlation between SecureKL output and ground truth AUC change, δ_i, from acquiring 1 additional dataset for a given source hospital H_o. We propose selecting data partner ranked by our secure system under SecureKL$_{\mathcal{XY}}$ score to reliably reduce downstream AUC downstream task. ($|\mathbf{H}| = 12$ hospitals; colored by source.) 164

E.1 **A Clear-text Example for Hospital Records.** 168

E.2 **Hospital self-reports final metric.** This does not constitute as a third party auditing. 170

E.3 **Hospital uses privacy-preserving data or model releases. From the desensitized data, the auditor approximates $\mathcal{F}$.** This presupposes the release of data. 170

E.4 **Auditor computes $\mathcal{F}$ from raw data, including sensitive patient data.** This level of data access exceeds what is necessary for the sole purpose of fairness metrics. **Left: Auditor goes inside the hospital to audit.** Frequent auditor visits may result in added operational costs for the hospital. **Right: The hospital sends encrypted data and a key to decrypt.** Auditor decrypts encrypted data. This setup introduces novel security risks. Security of key transmission and the data security at the auditing site becomes a concern. When auditing multiple hospitals, the auditor also becomes a central repository of sensitive data (and keys). 170

List of Tables

2.1 Summary of symbols in preliminary. 9

3.1 Summary of symbols in secure computation primer. 29

3.2 **Security Parameters Overview. Left, BFV [96]**: n roughly relates to the length of the integer input, p needs to be large if the input is high precision, and q needs to be large to accommodate computation. **Right, CKKS [62]**: n, q are the same as BFV's. f corresponds to the desired accuracy of the computation, and can be adjusted in a "rescaling" procedure; large f increases accuracy. 38

3.3 Summary of privacy mitigations for machine learning. 57

4.1 Correlation ρ of appraised values and data utility with varying amounts of label noise. Fine-tuning runtimes are limited to 1×, 4× and 16× of influence runtime, each benchmarked on the *best* performances under three learning rates: 0.001, 0.1, and 10. Hyperparameter tuning runtime for fine-tuning is excluded. 66

4.2 Influence Appraisal Correlation $\rho \pm \sigma$ With Data Utility on WDBC Over 10 Runs. 71

5.1 **Dataset divergence is predictive of utility for downstream models.** Pearson correlation (ρ) and p-values between each data addition strategy (π) and the source model's performance after adding selected dataset (AUC drop, δ_i), reported separately for two real-world datasets (eICU [244] and Folktables [83]). The non-secure strategies, including KL_{XY}, are detailed in Section 5.4.4 and in Section 5.4. Statistically significant p-values ($p < 0.05$) are bolded. 89

5.2 **Partnership selection strategies, differentiated by leakage (privacy cost).** A strategy π returns a chosen set of targets T from all candidates. Section 5.4.1 describes the strategies in detail. 91

5.3 **AUC improvements in mean and standard deviation**, across all source regions for each strategy π, for eICU and Folktables setups. n denotes the number of candidate datasets added to the source dataset. The small gains and high variance from adding selected datasets highlight the precarious nature of assessing data value in the real world. **Bold** indicates highest AUC improvement per n. *Note*: Only π_p $\mathrm{SecureKL}_{XY}$ is private. 102

6.1 Summary of symbols in private computation on healthcare data. 118

B.1 A useful dichotomy for machine learning privacy is input- vs. output-private methods. While not mutually exclusive in all scenarios, they *typify* two philosophies towards privacy protection for a system, leading to separate approaches. Section 3.2 argues that purely output-private methods may not ameliorate the privacy conundrum pre-partnership, where data or model releases are not predetermined. 148

D.1 ρ and p-value between AUC drop and plaintext KL using k samples using SGD (left) and LBFGS (right). 159

D.2 Spearman Correlations ρ for encrypted (in CrypTen) and plaintext (in PyTorch) KL-based methods 161

1 Introduction

Data sits at the critical junction of large scale model development and society-wide tension: the development of AI exacerbates and emboldens the loss of individual control over data; yet, at the same time, it requires vast amounts of *new* data in order to scale. Utilizing data that is privately held, from books to images to user behavior, promises better models ahead [134; 155; 46; 280; 331].

While private data can be an important piece in the checks and balances in this future, supporting it – protecting ownership, assigning correct credits, and respecting usage – faces issues.

First, making existing machine learning workflow confidential by direct application of existing general cryptographic – such as training a large model while encrypted – is typically unusable due to large overheads. Model training today requires researchers in the loop [176; 292; 143; 193; 135], so switching entirely to "private mode" where all details are hidden hinders utility. Second, private machine learning technologies are not mature for computations that need high precision and scale [164; 175], without sacrificing privacy itself [31], thus limiting their prospect for supporting the large models in the current markets. Further, a potentially desirable form of privacy people want is the right to not share, selectively share, or rectify their data, applied to machine learning models [52; 51]. Private computations alone cannot solve these issues; for example, supporting "the right to be forgotten" pertains to optimisation problems [221; 225; 304].

While challenges abound, several solutions exist: secure multi-party computation (S-MPC) can be flexibly applied to numerical workflows while allowing the computational methods to be auditable, striking a desirable balance between secrecy and transparency [115; 101; 75]. Moreover,

the machine learning-related computations themselves are also subject to rewriting, as to enable their implementation under homomorphic encryption, or to speed up their S-MPC workloads[1].

First, to address privacy within the context of machine learning, this chapter sets up an abstraction of the three roles in the data ecosystem. Combining secure computation techniques and algorithmic optimisation, tasks supportive of a healthy data ecosystem – training data appraisal (Ch. 4), continuous data auditing (Ch. 6), and private model evaluation (Ch. 5) – become feasible.

Setup As large models are data-hungry, relying on public data – characterized by a haphazard ecosystem of publicly owned, academically curated data – will not be sufficient [135; 306].

Emerging from the tension over data are three categories of stakeholders:

1) model owners and developers such as Google, Meta, and OpenAI; 2) data curators, owners, and creators such as Shutterstock, comic artists, or Reddit users, and 3) oversight entities like governments, corporate oversight boards, medical and other professional associations.

Though data curation and model development can co-exist in tech companies, the activities tend to be in separate divisions [292]. The three-party model remains valuable for analyzing incentives.

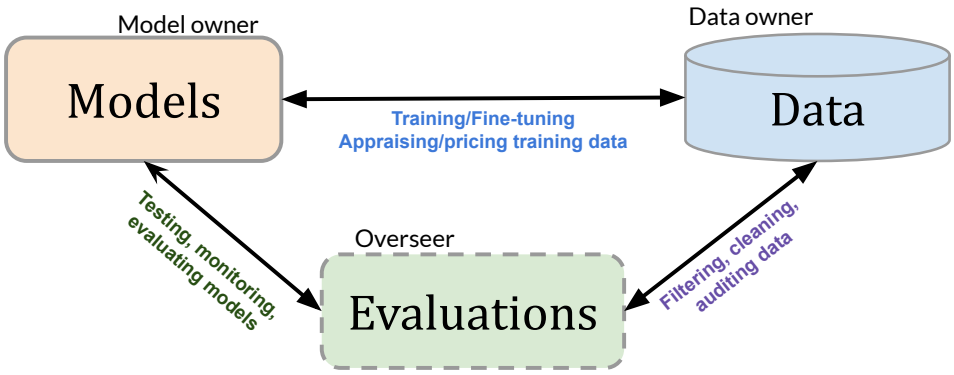


Figure 1.1: Incentives misalignment creates friction between the three categories of stakeholders.

Model makers need to access new data to stay competitive, especially in an economy that

[1]Rewriting refers to both algorithmic rewriting for the setup and making secure computation-friendly approximations, such as in Section 2.4.1.

encourages race-to-market. They typically have resources and talent to train large-scale models, and are often regulated as commercial entities [246; 135; 292]. Potentially, they want a healthy market to ensure their technologies are evaluated fairly, and that their less rule-following competitors are punished.

Data curators know their data is valuable for machine learning, and do not wish to share without being compensated or recognized. This may take the form of payments, equity, or name recognition [191; 3; 140]. Potentially, they want a healthy market that protects some form of control over their proprietary data, which may include the right to rectify data that is shared, and to appropriately detect and discourage stealing data.

Oversight boards are given the mandate to regulate the market or the organization, either with respect to data or with respect to the model. This may be motivated by privacy, safety, integrity, law enforcement needs, societal fairness, and so on [94; 182; 249; 121; 75; 45; 101]. Potentially, they want to be able to audit the training data for integrity issues, the decisions for fairness, or just to keep track of a model's ability on certain tasks over time.

Various forces impact this system of stakeholders. Under intense market competition, businesses are unlikely to make compromises such as reducing model accuracy or training visibility. In addition, the large-scale nature of developing foundational models makes computational powers, engineering talents, and data for training and evaluations all the more precious. These pressure points create friction between the often disparate parties.

Currently, the relationship is antagonistic: governments are suing AI companies and, of course, creators and data platforms suing model owners [122; 230; 296]. This is far from ideal: if private data is locked down, machine learning training may face a bottleneck, creators may not negotiate their fair share, and meaningful evaluations become challenging. Amidst this ever-evolving situation, no one magic solution is clearly in sight.

Nevertheless, despite their complicated relationship, all parties stand to gain from a healthy data ecosystem, through a privacy thesis put forth in Chapter 2. To that end, my work utilizes

practical secure technologies (Chapter 3) to ease the tension between these stakeholders by fostering better communication and easing incentive issues towards equitable data sharing (Chapter 4), demonstrating sensitive data auditing (Chapter 6), and evaluating dataset utility through dataset divergence (Chapter 5). These method works are illustrated in Figure 1.2

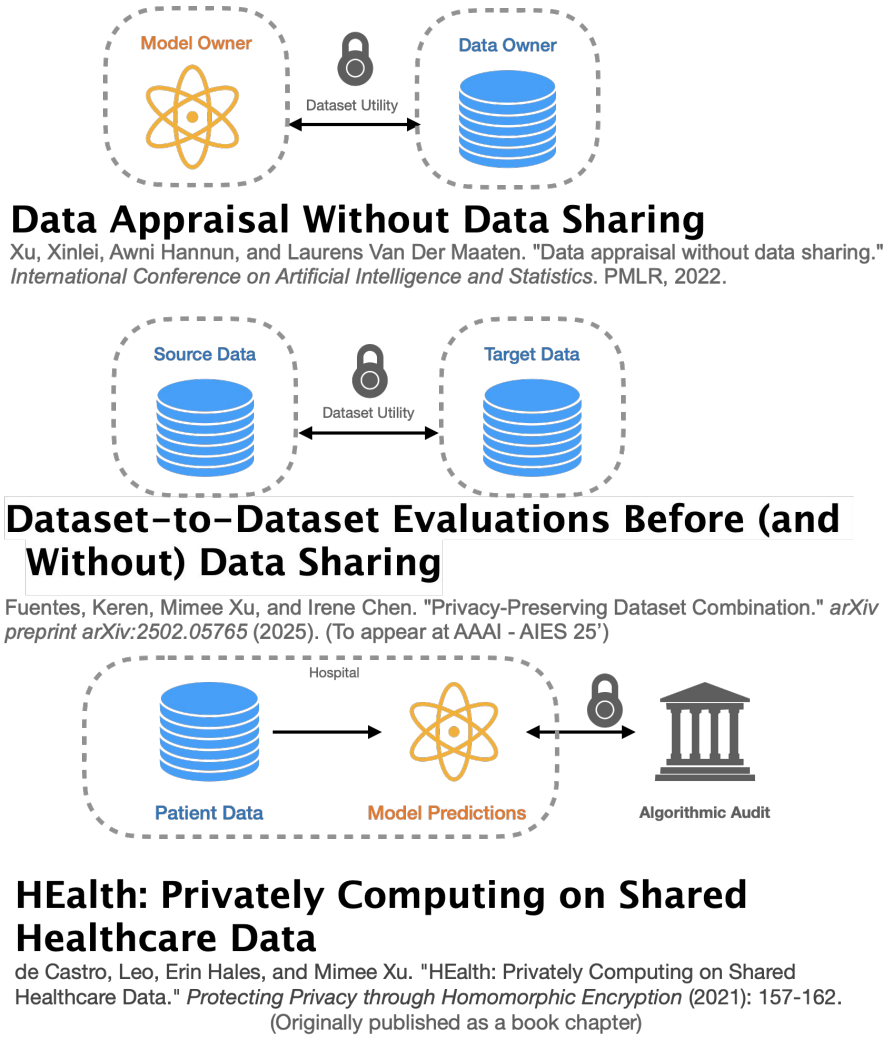


Figure 1.2: Individual works in this thesis, visualized within three-actor system.

1.1 Summary and Overview

1.1.1 Summary of Existing Gaps in ML Privacy

Insufficient recognition for model owners' incentives Privacy-preserving methods that hinders downstream utility is identifying the wrong root cause of "privacy". The model owners do not wish to share their training data (or over-expose their proprietary information), because they do not wish to compromise on their competitiveness in the market.

Insufficient attention on private facilitation towards data sharing While developing performant private models is an important pursuit, facilitating the flow of private data in their respective contexts — training, auditing, and evaluating — are significantly overlooked in examining privacy systemically.

Immature private computation technologies for machine learning 1. Just moving existing machine learning workflow "incognito" hinders utility and usability and 2. Private machine learning technologies are not mature for computations that need high precision and scale, thus limiting their prospect for supporting large models.

1.1.2 Contributions Overview

Focusing machine learning privacy research on the impact of long-term incentives in machine learning development requires both **novel problem definition**, and **effective solution approaches**.

The contribution can be summarized along two axes:

1. Introducing the **Three-Actor Privacy Thesis** for problem definition. This novel model offers a notion of privacy as systemic data tension among the model developers, data owners,

and overseers.

   (a) Positing **Data Acquisition Problem** where data value is uncertain before data sharing, and solve it with two separate methods [320; 106][2].

   (b) Describing the need to audit proprietary models in hospitals for its fairness [78].

2. **Co-designing secure computation alongside algorithmic problems in machine learning** for facilitating data sharing.

   (a) **The Effectiveness of Secure Computation** is presented through the feasibility of training, auditing, and evaluating private data.

   (b) **The Shortcomings of Private ML** are overcome through both clever algorithmic design and overcoming difficult engineering hurdles.

1.1.3 Summary of Chapters

Chapter 2 presents a thesis for framing privacy in machine learning, following through a thread of data-sharing setup, positing the Data Appraisal Problem. Then, this notion of privacy is then discussed in tandem with challenges in machine learning privacy and situated with other definitions of privacy. Along the way, necessary background, terms, and notations are introduced.

Chapter 3 includes a primer of secure computation as a preliminary, and contrasts it with related works of privacy-enhancing techniques in machine learning.

By identifying the gaps in current solutions, it re-states the pressing need to support an effective ecosystem for data sharing, model evaluations, and third-party auditing.

Chapter 4. First, as data used for training tends to be owned by disparate parties, the first subproblem pertains to whether *unshared* training data's utility can be evaluated without sharing

[2] I publish under both names: Xinlei Xu and Mimee Xu.

data. My work implements influence-based appraisal functions that are compatible with efficient MPC computation, so that no data needs to be shared for both the model owner and data owner to approximate the relative value of datasets, achieving 92.3% correlation with plain-text ground truth ranking for 100 datasets under induced class imbalance, and 96.0% under label-flipping, without the usability challenge of sensitive hyperparameters of training under MPC (AISTATS, 2022 [320]).

Chapter 5. Second, would secure computation help small organizations make *realistic* data collaboration decisions, despite the unpredictable nuances of real-world samples? In data-limited, model-agnostic scenario, **SecureKL** (SKL) efficiently computes secure dataset-to-dataset divergence between a source entity and a target entity's respective data, before sharing data. In ICU mortality prediction, it outperforms blind selection, and data-leaking methods – demographic-based selection using age, gender, or race, or using a subset of samples – by reliably selecting positive partners. By utilizing all the data, SKL matches plaintext recommendations by 90% while maintaining minimal leakage of the underlying data. (Will be published with the proceedings for AAAI AIES in 2025 [106])

Chapter 6. Third, can the public audit private data, especially in domains where data is kept private by default? Using HE for auditing triaging fairness in hospitals' emergency department, as an example, this thesis provides a qualitative description of the setup that can be applied to ease the tension between regulators and private data parties, without the need to decrypt private data, expanded upon work from 2019 [78]. (Published as a book chapter in the seminal work [175] in 2022)

2 | The Three-Actor Privacy Thesis

2.1 Chapter Summary

To pinpoint privacy problems in machine learning, we bring forth a shared abstraction, the **Three-Actor Ecosystem** of machine learning, which stems from the roles identified in Chapter 1: model developers, data "owner", and overseer. This chapter is organized along two veins:

1. To identify the emerging incentive issues in machine learning development, which narrow privacy definitions miss. Section 2.2 emphasizes the naturally different roles of ML development and their resulting data sharing tension, due to the necessary data scaling for solving the underpinning optimization problems. To enable modeling general privacy problems from ML development, Section 2.3 posits the Three-Actor Privacy Thesis.

2. To express the novel technical challenges from simultaneous desires for privacy and utility. Respecting model owners' interests to acquire external data, Section 2.4 defines data utility, while Section 2.4.1 tackles this utility computation with influence functions and dataset divergence. Noting that these approaches originate from machine learning, but do not consider privacy. Section 2.5 clarifies the seemingly inherent conundrum – the **Data Acquisition Problem** between privacy and utility – thereby setting up the novel scientific problems throughout this thesis.

Finally, Section 2.6 discusses the implication of our approach, contrasting with other privacy definitions such as contextual integrity. The notations are defined in Table 2.1.

Notation	English Description
MO	Model Owner
DO	Data Owner
OS	Overseer
$\mathcal{D}_{\text{a}}$	Additional Training Data
$\mathcal{M}$	Trained Model
$\boldsymbol{x}$	Input Vector (Features)
y	Scalar Output (Label)
L	Loss Function
K	Number of Data Owners
$\mathcal{D}_{\text{tr}}$	Training Data
$\mathcal{D}_{\text{te}}$	Test Data
$\mathcal{D}_{\text{eval}}$	Evaluation Set
$U(\mathcal{D}_{\text{a}})$	Dataset Utility
f_{priv}	Private Appraisal Function
f_{audit}	Auditing Function
$\mathcal{F}$	Function
E	Encoding Function
D	Decoding Function

Table 2.1: Summary of symbols in preliminary.

2.2 Optimisation Setup

Suppose a learning algorithm outputs model $\mathcal{M}$, parameterized by θ. When trained with the loss function L using the principle of Empirical Risk Minimization [259; 81; 304], the optimal parameters are

$$\hat{\theta} = \arg\min_{\theta} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\text{tr}}} L(\boldsymbol{x}, y; \theta) + \lambda\|\theta\|_2^2. \qquad \text{(Regularized Empirical Risk Minimization)}$$

Concretely, the model developer usually begins with a training set, $\mathcal{D}_{\text{tr}}$; for L2-regularization as illustrated [290], they also set λ.[1]. Their testing set, $\mathcal{D}_{\text{te}}$, is then used for evaluation. Our privacy model centers on a **key observation**: **Model Owners** today drive the development of machine learning, from training to evaluation to deployment. In each stage, the model – as well as its associated training and testing data – is assumed to be proprietary throughout this thesis.

A Natural Outgrowth: Data Acquisition Becomes Necessary. Since Hestness et al. [134], empirical works have predetermined data as a key driver to performance gains, via the so-called "scaling laws" [155; 46]. Yet, dataset size has not always been big; for a given task, they started out small, then grew bigger [280], more diverse [331], sometimes more compute-optimal [135]. Arguably, across all domains, we are still in the growing phase, searching, on one hand, more data [203; 46; 236; 253; 266]; on the other, better data *combinations* [268; 224; 243; 187; 206].

A curious, privacy-relevant dynamic ensues. More data generally benefits optimisation, especially if it is diverse and high quality [134; 155]. Yet, data availability is a problem, as such public datasets are sparse [292; 193]. The model owner's data may be insufficient for their purpose, including for training [134], fine-tuning [146], and evaluations [278]. Thus, it becomes imperative for them to look beyond their organization in order to acquire more data.

[1] Unless otherwise stated, this model is optimized with Stochastic Gradient Descent (SGD) [37].

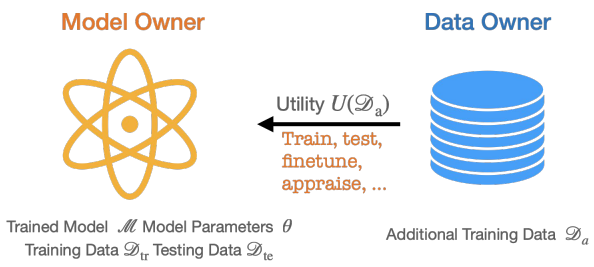


Figure 2.1: Model Owner (*MO*) and Data Owner (*DO*) Relationship: Both can benefit from Share(MO, DO).

2.3 The Three-Actor Ecosystem for Machine Learning

To meaningfully address the needs of model developing parties, this thesis recognizes – and respects – the incentives which are *disparate* from the model owners' economic or other gains.

Following the distinct roles introduced in Chapter 1, this thesis assumes three actors:

1. **Data Owners (*DO*)**: A set of entities where the i-th data owner possesses dataset $\mathcal{D}_a^{[i]}$.
2. **Model Owner (*MO*)**: The entity that owns trained model θ, train set $\mathcal{D}_{tr}$, and test set $\mathcal{D}_{te}$.
3. **Overseer (*OS*)**: An entity responsible for audits and evaluations, potentially holding evaluation data $\mathcal{D}_{eval}$ and a potentially fully-automated auditing algorithm f_{audit}.

Sharing Data Between *MO* And *DO* This conceptual framework is motivated through **Share**(*MO*, *DO*): the action of sharing data between a model owner and a data owner (Figure 2.1).

When additional training data, $\mathcal{D}_a$, is owned externally with respect to the model owner, a privacy tension arises: on one hand, *MO* and *DO* may gain utility from sharing and transacting, yet when faced with uncertainty in this utility, their collaboration may stall.

2.3.1 Discussion

Data "Owner" and Overseer Interests The entity of data owners encompasses a multitude of interests, described as wanting full control over their privately-held data: individuals, creators and platforms ought to know and decide when and how they share – and not share, or even retract – data.[2] This thesis additionally recognizes that the Overseer will take increasing role in regulating and governing the machine learning technologies, such as external auditors and regulators [75]. The interactions of the three actors will be further discussed in Section 2.6.

Notable exceptions to closed weight models. Currently, keeping the model secret is not universal among all major model owners, as several "open-weight" models have emerged as exceptions, such as Meta AI's Llama models [292; 89], Mistral AI's 8x7B models [148]. As this thesis is being hectically revised, at least three other "frontier" open models have been released, by DeepSeek [80], Moonshot [285], and OpenAI [231]. However, we note that as of now, the underlying data used for developing these "open" models are still kept private by default. Interestingly also during revision, Meta suggested reversing course from releasing frontier open-weight models, in a brand new team formation [336] and in an earnings call [210]. It is also worth noting that these companies' best-performing large pre-trained models have remained closed-weight, perhaps out of a need to maintain a competitive edge while entering their smaller models in the "open source" arena. This indicates a need for model owners to maintain control of the access level of their models while retaining proprietary details about the training data – a crucial part of our thesis.

Abstraction Remark Two abstractions are introduced in this thesis. This chapter focuses on the three-actor abstraction, summarized in Figure 2.2. Later Chapters will define and follow the

[2] Our nomenclature centers the *externality* of data to machine learning development, describing Model Owners and Data Owners as distinct entities; yet, we use the term "data owner" without implying the legality or ethicality of "data ownership" [142].

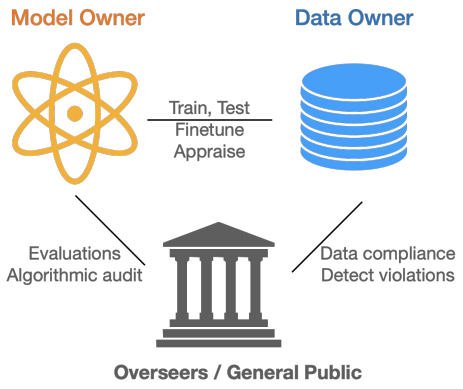


Figure 2.2: Three-Actor Ecosystem for machine learning clarifies entities with distinct goals, each wanting to gain utility while maintaining privacy. This abstraction forms the foundation of the privacy notions in this thesis; privacy issues are described as data-related tension in this system.

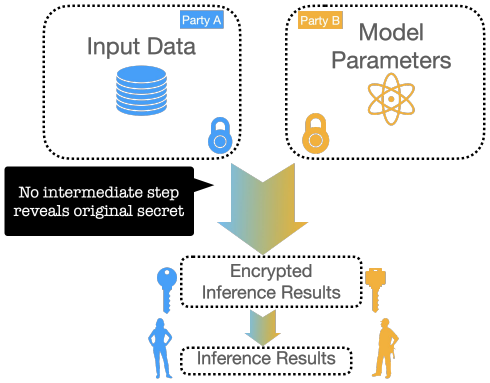


Figure 2.3: Secure Computation on Disparately-owned Data for Machine Learning, using secure inference between data and model as an example. In this thesis, secure computation is the primary approach towards respecting both model utility and privacy among distinct actors.

second abstraction, previewed in Figure 2.3, which is to compute without leaking input[3]. This technical approach is aimed to address privacy tensions arising from the descriptive scenarios within the Three-Actor Ecosystem.

[3]The underlying technologies of this abstraction will be discussed in Section 3.

2.4 A Conceptual Framework For Privacy Tensions Through A Sharing Conundrum

Main Thesis of Three-Actor Ecosystem To meaningfully address the privacy needs of disparate parties: model owners, data owners, and overseers.

Share(*MO*, *DO*) Conundrum When *DO* has ownership over $\mathcal{D}_a$, which *MO* wants to include to train their model, a privacy tension arises: on one hand, both *MO* and *DO* could gain utility from sharing and transacting, potentially. Yet, when this utility is unknown, neither actor wants to share their proprietary data or model to find out. The crux lies in reducing the uncertainty in the utility – denoted as $U(\mathcal{D}_a)$ – with respect to fairly pricing or evaluating $\mathcal{D}_a$.

Data Utility To consider acquiring any given dataset $\mathcal{D}_a$, the model maker *MO* wishes to determine the *utility gain* from updating $\hat{\theta}$ to fit $\mathcal{D}_{tr} \cup \mathcal{D}_a$. A natural definition denotes the difference in test losses via $\mathcal{D}_{te}$:

$$U(\mathcal{D}_a) = \frac{1}{|\mathcal{D}_{te}|} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{te}} L(\boldsymbol{x}, y; \hat{\theta}) - L(\boldsymbol{x}, y; \theta^*) \qquad \text{(Dataset Utility)}$$

where θ^* is the resulting model parameters after including $\mathcal{D}_a$.

Naively computing the Dataset Utility assumes the **MO** to compute the new optimal parameters θ^* upon adding dataset $\mathcal{D}_a$, by minimizing the regularized empirical risk on dataset, $\mathcal{D}_{tr} \cup \mathcal{D}_a$. This requires accessing θ^* from re-training, which presumes data sharing.

Alternative to assuming re-training, two general approaches can be used to evaluate dataset utility. One, using the second-order approximation derived through leave-one-out training, arriving at the forward application of an influence function. The other is through dataset divergence, without assuming $\hat{\theta}$. We now examine their specific procedures.

2.4.1 Two Approaches: Influence and Divergence

Forward Influence Functions The influence function $\mathcal{I}(\boldsymbol{x}, y)$ associates a training sample with the change in the model parameters under an infinitesimal up-weighting of that sample in the risk [72; 165]. We use influence functions to approximate the change on the resulting loss from including the dataset $\mathcal{D}_{\text{a}}$. Denoting the empirical Hessian $\boldsymbol{H}_{\hat{\theta}} = \frac{1}{|\mathcal{D}_{\text{tr}}|} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\text{tr}}} \nabla^2_\theta L(\boldsymbol{x}, y, \hat{\theta})$, the forward influence of sample $(\boldsymbol{x}, y)$ is given by:

$$\mathcal{I}(\boldsymbol{x}, y) = -\boldsymbol{H}_{\hat{\theta}}^{-1} \nabla_\theta L(\boldsymbol{x}, y, \hat{\theta}). \tag{2.1}$$

This function is a first-order approximation of the change in $\hat{\theta}$ for each sample $(\boldsymbol{x}, y) \in \mathcal{D}_{\text{a}}$. In turn, we can use $\Delta\theta \approx \mathcal{I}$ to assess the influence of $(\boldsymbol{x}, y)$ on the test loss of $(\boldsymbol{x}_{\text{te}}, y_{\text{te}})$ via the chain rule:

$$L(\boldsymbol{x}_{\text{te}}, y_{\text{te}}; \theta^*) - L(\boldsymbol{x}_{\text{te}}, y_{\text{te}}; \hat{\theta}) \approx \nabla_\theta L(\boldsymbol{x}_{\text{te}}, y_{\text{te}}; \hat{\theta})^\top \mathcal{I}(\boldsymbol{x}, y). \tag{2.2}$$

Using these observations, we define the influence-based appraisal function to be the sum of each training sample's influence:

$$f_{\text{if}}(\mathcal{D}_{\text{a}}) = -\frac{1}{|\mathcal{D}_{\text{a}}| \cdot |\mathcal{D}_{\text{te}}|} \sum_{(\boldsymbol{x}_{\text{te}}, y_{\text{te}})\in\mathcal{D}_{\text{te}}} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\text{a}}} \nabla_\theta L(\boldsymbol{x}_{\text{te}}, y_{\text{te}}; \hat{\theta})^\top \boldsymbol{H}_{\hat{\theta}}^{-1} \nabla_\theta L(\boldsymbol{x}, y; \hat{\theta}). \tag{Forward Influence Function}$$

Note our influence function applies the approximation *forward* on unseen data, it is differentiated with Koh and Liang [165]'s influence functions which apply to examples that are already seen in training. Our modifications are further explained in Appendix C.

Other definitions are not precluded. Without involving the model itself, a measure on existing and additional training data may be meaningful. This is useful for two separate reasons: one, the model owner may not be sure yet as to what model to use, therefore lacking a θ needed for

computing the Dataset Utility. Two, without training a model, the datasets – $\mathcal{D}_{\text{a}}$ and $\mathcal{D}_{\text{tr}}$ – may be weakly predictive of their combined utility in machine learning without modeling assumptions.

Dataset Divergence Recall KL-divergence, also called *information gain* [170; 245], measures how much a model probability distribution differs from a true probability distribution. If the model owner's in-house data forms the true distribution, where $\mathcal{D}_{\text{te}} \sim P_{MO}$, the model owner may ask the *proxy* question: *does the data owner's distribution conform to my distribution?*

Letting the data owner's additional data be drawn from P_{DO}, or $\mathcal{D}_{\text{a}} \sim P_{DO}$, the *MO-DO* divergence is ideally estimated as

$$\text{KL}(P_{MO}||P_{DO}) = \int_{\boldsymbol{x}\in\mathcal{X}} \log \frac{P_{MO}(\text{d}\boldsymbol{x})}{P_{DO}(\text{d}\boldsymbol{x})} P_{MO}(\text{d}\boldsymbol{x}). \qquad \text{(Ideal } \textit{DO-MO} \text{ Estimator)}$$

Dataset Utility Without Assuming Downstream Models A practical advantage of this simplification is that being agnostic to $\mathcal{M}$ is more appealing than formulations of Dataset Utility in high-stakes, data-constrained settings. Surely, when there is not a lot of data, the model to use may not be pre-determined until more is acquired. Yet, moreover, Shen et al. [267] observes that a new hospital data's usefulness can correlate with how similar the two hospitals' data distributions are. Holding across various models, hyperparameters, and training conditions, Miller et al. [212] demonstrates that a model's in-distribution and out-out-distribution performances are correlated, regardless of model[4]. As such, whether an additional dataset $\mathcal{D}_{\text{a}}^{[i]}$ is useful may be related to its distribution being close to that of source. However, $\mathcal{D}_{\text{a}}$ and $\mathcal{D}_{\text{tr}}$ are datasets, not the distributions P_{MO} and P_{DO} that we need to compute for the Ideal *DO-MO* Estimator [226].

Dataset Divergence $\text{KL}_{\mathcal{XY}}$ One approach is to fit a distribution on the datasets instead. Shen et al. [267] proposes model-based scores to make this divergence approximation tractable from

[4]Intuitively, in-distribution quality is paramount in low-data settings. In contrast, data-rich domains like language modeling more frequently benefit from diverse, specialized data sources.

small samples, which we use in [106] for the first time in secure computation (Chapter 5). We adapt the simple heuristic $\mathrm{KL}_{\mathcal{XY}}$ from [267], where a logistic predictor $\mathrm{Score}(\cdot) : \mathcal{X}, \mathcal{Y} \to [0, 1]$ is fit (privately) in the following procedure: construct a combined dataset $(\mathcal{X}, \mathcal{Y}) : \mathcal{D}_{\mathrm{tr}} \cup \mathcal{D}_{\mathrm{a}}$[5], and create the corresponding membership-labels $\mathcal{I} : \{1 \text{ if } x \in \mathcal{D}_{\mathrm{tr}}; 0 \text{ otherwise}\}$. The score function is the probability score from logistic regression on $(\mathcal{X}, \mathcal{Y}) \to \mathcal{I}$.

Then, the logistic regressor $\mathrm{Score}(\cdot) : \mathcal{X}, \mathcal{Y} \to [0, 1]$ is averaged over *MO*'s samples, obtaining

$$\mathrm{KL}_{\mathcal{XY}} = \mathbb{E}_{(x,y)\sim\mathcal{D}_{\mathrm{tr}}}(\mathrm{Score}(x, y)). \qquad \text{(KL-XY Score)}$$

Summary of Utility Approximation We briefly introduce the two techniques used in the thesis. Both approximate the Dataset Utilitywithout retraining the model. Their choices pertain to their respective setups, and are not exhaustive for all techniques.

As hinted in Figure 2.3, they will be used under secure computation in the later chapters.

Later, we will revisit these approximations:

- Section 3.5 will discuss writing forward influence functions for secure computation.
- Section 5.6 will extensively analyze dataset divergence measure $\mathrm{KL}_{\mathcal{XY}}$'s empirical performance on real-world data and its efficacy in private.
- Appendix C.1 will cover three technical areas of forward influence functions: the mathematical derivation as an approximation of test loss reduction, the underlying assumptions, and their implications.
- Appendix A.1 includes more nuanced discussion on dataset divergence.

[5]The labels in the original dataset, i.e., mortality, is included in the data as $\mathcal{Y}$, hence the name $\mathrm{KL}_{\mathcal{XY}}$.

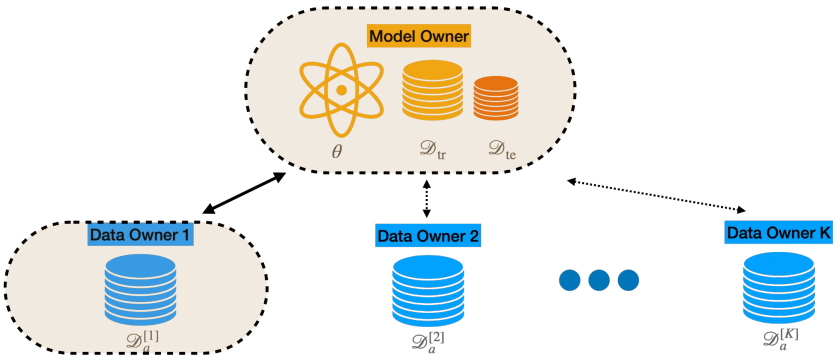


Figure 2.4: Model owner wants to perform pairwise evaluations *before* committing to a data partnership.

2.5 Problematizing Dataset Acquisition Conundrum

Motivation: More Privacy Conundrum In Equations Dataset Utility as well as its approximations Forward Influence Function and KL-XY Score, $\mathcal{D}_{\text{tr}}$ (belonging to *MO*) and $\mathcal{D}_{\text{a}}$ (belonging to *DO*) need to be combined and shuffled. Additionally, both involve model training on the combined data, to compute $L(\boldsymbol{x}, y; \hat{\theta})^\top \boldsymbol{H}_{\hat{\theta}}^{-1} \nabla_\theta L(\boldsymbol{x}, y; \hat{\theta})$, and $\text{Score}(x, y)$, respectively.

Recall the goal of dataset utility computation is to let both model training entities and data owning entities feel comfortable about the benefits *before* sharing. Yet, both definitions of utility entangle datasets that are owned by disparate parties, manufacturing another Catch-22: **solving the data sharing problem requires combining private data from both parties**. How then, do we make this problem tractable?

Problem Statement We suppose each party holds their own data and aims to apply an algorithm such as the Forward Influence Function or the KL-XY Score. As sketched in Figure 2.4, the model owner wishes to evaluate datasets from data owners, but without direct access. We ask,

> To ascertain among candidate data sources, **which one would be sensible to incorporate with my existing data (or model)?**

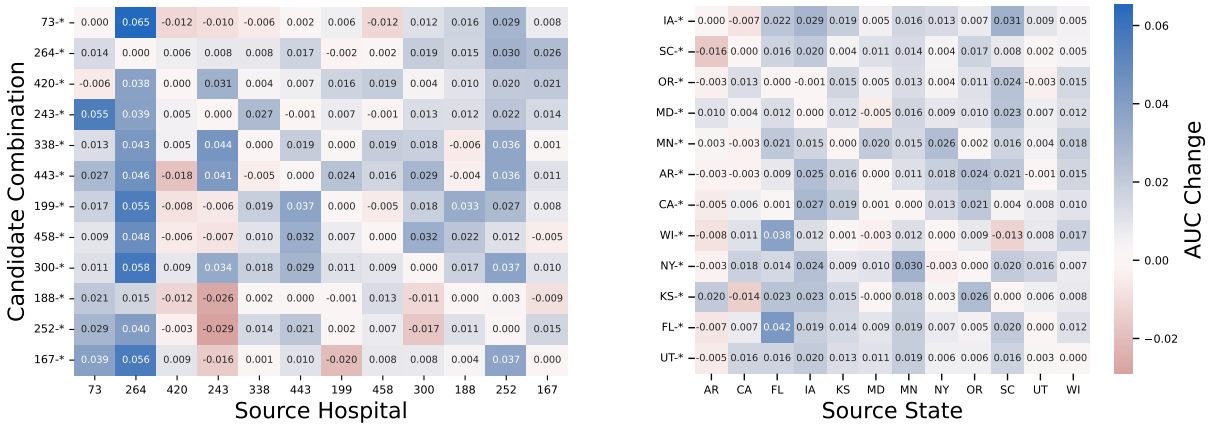


Figure 2.5: Random selection may harm: Real-world data collaborations are inherently uncertain, necessitating pre-partnership selection. This illustrates the Dataset Combination Problem studied in Chapter 5.
AUC change for a source entity (x-axis), after incorporating external data from a different source (y-axis), across hospitals (left) and states (right).
Left: in mortality prediction using eICU data [244], 10 out of 12 hospitals may see their mortality prediction model degrade for *some* potential hospital partners.
Right: in income prediction using Folktables [83], combining with random state leads to worse prediction in 7 out of 12 states. (Red indicates bad collaborations.)

Not Solely About Privacy Examining real-world data sharing, we combine different entities' datasets *pairwise* before training a joint model. Then the joint model is then tested on the source entity. Specifically, this data-adding scenario is simulated using eICU datasets from 12 real hospitals for mortality prediction [244] and using survey datasets across 12 states for income prediction [83][6], with the model improvements produced in Figure 2.5. Perhaps surprisingly, ostensibly "in-domain" data may harm model performance. This mysterious phenomenon appears inherent to optimizing with unseen data (or a part of generalizing under domain shifts), sometimes known as "the dataset combination problem" (Chapter 5) [209; 13; 284; 222; 167; 308; 40; 267]. This suggests that the "commitment issue" extends beyond privacy; instead, it highlights the inability to *privately* assess an external dataset's utility *before* partnerships.

This thesis directly address both by enabling parties with **"intent" to collaborate** to privately

[6] All experiments on downstream models are trained *in the clear* without any encryption, as our encrypted computations are performed before data sharing.

evaluate dataset utility without a pre-commitment. Chapter 3 introduces the underlying private computation solutions, while Chapter 4 and Chapter 5 concretely tackle the challenge where private data cannot be seen/revealed.

2.5.1 Baseline Discussion

This problem gives rise to two natural baselines, which we consider in Chapters 4 and 5.

- **No privacy guarantee** (compute as is, without privacy enhancements). This is to ensure the correctness: as we implement in secure computation, approximations may be introduced (discussed in Section 3.4).
- **Random selection** (no information is provided). This is to ensure demonstration of positive real-world benefits, when approximating the Dataset Utility. Additionally, a random strategy may evade selection biases and help gather diverse data, therefore worthwhile to compare against. However, blind selection likely suffers in domains where data quality is problematic[7].

2.5.2 Assumptions

Privacy Model: Semi-honesty We operate under a semi-honest privacy model—also known as *honest-but-curious* or *passive security*—where parties follow protocols but may probe intermediate values. Parties are "curious", meaning that they may probe into the intermediate values that they are *supposed* to have access to during the protocol execution. Between known parties with some pre-existing trust, preserving privacy under semi-honesty can incentivize collaboration (such as MO and DO in Share(MO, DO)). In our setup, this means each party authentically wants to share their data and set up a collaboration.

[7]For example, even given trust, a hospital's data may be very noisy simply due to the collection process [76].

Integrity Issues Participants are assumed not to tamper with the data they send as input. Sidestepping integrity issues may become increasingly impractical as the number of parties grows, or when the setup pertains to highly competitive industries with less expected trust. However, our scenarios in Chapter 4 and Chapter 5 involve only two parties that want to eventually share data. Chapter 6 also assumes that participating hospitals have consented to the audit, thus simplifying our security model.

Metadata is Not Protected Existing knowledge is assumed to be known, thus not protected by our privacy guarantees. This means the pre-existing "metadata" – including the task at hand, e.g., classification, the size of the additional data, i.e., $|\mathcal{D}_a|$, and the rough idea of how the data comes about, e.g., from hospital emergency room – may be assumed already shared without compromising privacy.

Setup-specific Assumptions In practice, the specific secrecy requirements depend on the setup. For example, between *MO* and *DO*, the desirable outcome of performing Share(*MO*, *DO*) is a fair exchange of data, where data utility is computed *before* data sharing. So $\mathcal{D}_a$ may only be considered private prior to the transaction, while θ is always private [320] (Chapter 4). When evaluating dataset-to-dataset divergence, the source hospital which initiates the collaboration likely has more data to be kept private and a target model that is yet undetermined (or kept secret) (Chapter 5). Additionally, when *OS* audits *DO* to perform $f_{\text{audit}}(\mathcal{D}_a)$, the auditing algorithm f_{audit} may not need to be private such as in [78] (Chapter 6), but does need to be fully-automated, i.e., without human-in-the-loop.

2.5.3 Evaluation Model

To make different methods comparable in our experiments, we further assume that the model owner has a fixed budget acquiring a dataset of a fixed size. Because natural data may be of

variable sizes and prices, evaluating on uniform sized datasets may seem strong. However, we note that both the Forward Influence Function and the KL-XY Score are additive measures that readily scale with dataset size, and are thus applicable to varying sizes.

Applying this assumption does not impact the security properties or the algorithmic applicability of our proposed methods. It is primarily made to streamline our evaluation – to have a unified pipeline for downstream model training, and to ensure a fair comparison between methods. Given that we are primarily concerned with ascertaining the differential dataset utility, relaxing this evaluation assumption, i.e., allowing for varying dataset sizes, may complicate baseline comparisons.

For Influence Functions Equation Forward Influence Function simply divides by the cardinality of candidate dataset, $|\mathcal{D}_a|$, requiring no adjustments to compare against varied dataset sizes. Moreover, evaluating influence-based appraisal against varied dataset sizes may result in noisier outcomes for other metrics, which we had shown to be less robust in in Chapter 4 (Section 4.3).

For Dataset Divergence Equation KL-XY Score also readily scales with dataset size. In Chapter 5, both our method and a baseline method require the training of a model after combining $\mathcal{D}_a$. Ensuring a fair comparison between them already requires two separate sets of hyperparameters even when the dataset size is uniform (Section 5.6). Varying dataset size would entail tuning hyperparameters for each known $|\mathcal{D}_a|$ in order to ensure best model fit. Additionally, Chapter 5 uses this assumption to directly compare private and non-private approaches, where data leakage level alone proxies the cost for setting up collaborations – if dataset size varies, the leakage level of "leaking 1% of all data" will not be as comparable.

2.6 Significance of The Three Actor Ecosystem

So far, the three-actor privacy thesis identified incentive clashes from model scaling and uncovered privacy challenges from natural desires for both privacy and utility during data sharing, thus setting the stage for defining the Data Acquisition Problem.

Notice this thesis uses the expression "private data" in a broad sense, that is, not only sensitive data pertaining to individuals, but more generally any data that its creators, producers, or owners are unwilling to share[8]. Accordingly, we distil systemic data tension amongst the stakeholders into an overlooked technical problem of using private data while controlling its dissemination. This distinction is what makes the three-actor framework unique.

In this section, we expand on the challenges of conceptualizing privacy in machine learning, and discuss where our thesis fits in this pursuit.

2.6.1 Systemic Privacy Tension

Large scale machine learning observes an *inherent* tension with privacy. The historical advancement of technology itself often enhances collection and analyses [228], a trend dramatically more evident in large models [45; 131].

Indeed, as large models increasingly ingest vast amounts of publicly-available data, their ability to retrieve and re-surface data can raise novel privacy issues.[9] Defining privacy merely via the sensitive data framework [229] or as individual freedom [313] are insufficient to capture the systemic nature of these challenges [232]. This poses a complication for machine learning privacy researchers: what problems should computer scientists work on, that can actually mitigate

[8]We broadly use the term "owner" for the data entity to differentiate between model developers and overseers. It is used without implying a legal ownership model of data.

[9]For example, as retrieval improves, data that was previously less sensitive may become sensitive over time, because they become associated with other sensitive information [65; 205]. Moreover, an individual might be willing to share her image for face recognition but not for inferring her location from the background objects. Similarly, focusing only on individual impact may lead to public policies that overlooks broader societal impact [131].

privacy tension [45]?

As a key step towards defining concrete problem, the three-actor abstraction clarifies the interconnected nature of private data flows in machine learning: sharing, auditing, and evaluations are often negotiated between parties with potentially competing interests and varying levels of trust. While these roles may shift over time, they are likely *fenced off* from each by default. Even for overlapping organizations, such as technology companies that own data and train models, the divisions of those functions are nevertheless independently operated. This distinction of the three actors helps identify privacy challenges in this system, while offering unique opportunities for mitigations that lead to a healthier long-term ecosystem.

2.6.2 Evidence for Existing Tension

Data availability issues have always plagued deep learning [151]. However, it is only recently that media platforms no longer wish to provide content data for without compensation [208; 217]. In tandem, as the awareness of algorithmic technologies takes hold, individuals have become more vigilant about their digital sharing [52; 272; 77]. As scaling continues, Villalobos et al. [305] even projects that human-generated data will be insufficient in the next decade for machine learning.

Moreover, the competitive nature among model owners creates an incentive to hoard data. The AI companies' race-to-market dynamic raises concerns for the centralization of power, which can side-step important issues like user privacy [289; 258; 198; 218]. At the same time, the security perimeter of model developers keeps enlarging. Besides training data that is kept private by default, data security now includes securing some proprietary models' weights, sometimes elevated to the importance of national security [11]. This commercial incentive interacts with privacy constraints, where no effective third-party audits – despite being frequently talked about – seem to happen [75].

Lastly, oversight pressure has the potential to increase. Data is already being regulated [94; 182] and the public increasingly worry about potential AI issues [52].

To address these tensions, Chapter 6 shows techniques based on secure computation can support oversight without regulations succumbing to the urge to over-surveil, even as regulatory pressure grows. When these tensions grow, Chapter 4 offers a path forward to acquire data while respecting data owners' rights to their private data for the long term. Chapter 5 studies data acquisition in data-limited domains. Discussion on policy impact is additionally included in Appendix B.4.

2.6.3 Limitations of Technical Solutions

It is important to recognize that security engineering such as secure computation (Chapter 3) is valuable in easing the tension between the different actors, but will not absolve technology providers from all privacy responsibilities. As centralized data collection has become commonplace, meaningfully limiting data shared while maintaining performance is often not attractive enough for technology companies to opt in. Therefore, my works serve as feasibility demonstrations, echoing other beneficial MPC proposals for solving crucial incentives issues, including but not limited to Blumberg and Eckersley [30]'s 2009 EFF proposal on location data for toll collection and Frankle et al. [101]'s work on accountability for secret processes. Without policy enforcement, adopting the more sophisticated implementation – even if proven to be performant – often requires the existing data-collecting parties to voluntarily choose it [125; 22]. Because these relationships are developing, establishing clear "locus of accountability" and minimizing the "need of trust" in this ecosystem remains challenging [159; 125; 22]. For continuous, cross-displinary research is needed, the three-actor framework contributes to modeling systemic tension concentrated at the boundary of sharing among the different actors.

Discussion The concept of privacy has evolved alongside technological advancements [68; 131; 45]. Rather than offering a definitive definition, this section clarifies key technical considerations relevant to its future in ubiquitous machine learning. Specifically, this thesis frames privacy

issues in scientific terms in anticipation of future questions regarding privacy's role arising from the inherent tension of different actors — each of whom may simultaneously seek privacy and benefit from machine learning.

2.6.4 Contribution: Expanding Privacy Definitions

We live in a chaotic era regarding privacy definitions. A large body of legal scholarship debates the nature of privacy for individuals [311; 313; 256]. These debates gain vigor with novel technological advances. After all, it was with the advent of photography in the 19th century that prompted Warren and Brandeis to write *The Right To Privacy* [311], which became the basis of privacy law in America. Even then, they described the "contour" of privacy as unclear. This ambiguity lends a general feeling of underwhelm, where privacy seems to definitionally fluctuate whenever a new technology emerges [67].

Reminding us that privacy itself evades a singular definition [274], Hartzog [131] posits that methods which tackle privacy ought to recognize its plurality and fluidity. Indeed, this attitude is crucial for technology developers. As co-designers for the future, one cannot predict or control it. As science seeks to understand the unknown, technological input is necessitated in many imaginable scenarios, privacy-relevant or not – and this is where I hope my work contributes.

Situating the Three-Actor Privacy Thesis Definitional chaos notwithstanding, the perimeter of my technical work is defined by incentives that *a priori* respect the strong economic interests in developing machine learning [228; 68; 131]. These commercial interests may at times clash with the interests of other participants in the same ecosystem.

Most similar to the conceptualization of privacy as "boundary management" in Cohen [68], role demarcation is a prerequisite to self-determination. Simply put, this thesis does not reactively call for "balance" as mutual compromise between the parties, but rather actively focuses on existing potential for win-win situations.

Yet, relying exclusively on the data owners' self-management leads to fatigue and overwhelm (even if there were a "win") [275]. Privacy is thus a design problem that requires a careful, systematic view to address the so-called privacy paradox [228; 275]. Without technical innovation, relying exclusively on oversight risks increasing surveillance [68].

In design, secure computation aligns with the principle of "data minimization" [50], albeit with the opt-in problem noted in Basu et al. [27] and Gürses et al. [125]: centralized technology developers may not adopt it, due to the associated costs and technical complications.

Lastly, the act of developing data-sharing relationships can be analyzed through Contextual Integrity (CI) by Nissenbaum [228], a framework for evaluating appropriate information flows. CI employs a notion of "appropriate data flows" as privacy, where a descriptive (pre-existing) setup can be analyzed and improved upon with a normative (ethical) flow. We take inspiration from CI to analyze a system of data-flows, simplified to those underlying machine learning development, and focus on grounding concrete technical issues.

This thesis adopts an empowering outlook for machine learning: the language of privacy can be expressed in the three-actor model such that computer scientists can readily understand and identify issues created by ML development, and make effective progress towards solving them.

2.6.5 Conclusion

This section set forth a privacy conceptualization in machine learning that underpins the thesis, and offers preliminary definitions for our problem setup. Observing incentive conundrums that are systemic and predictable in machine learning development, a three-actor model is described. Under this isolation, this thesis describes privacy tensions arising between them as a motivation to model emerging incentive challenges. Lastly, we contrast this novel concept of privacy with existing definitions. Motivated by building a sustainable future of healthy data ecosystems, my works are embedded in areas where practical secure technologies can be applied to great effect: towards solving incentives issues between data owners, model owners, and overseers.

3 PrivacyML: Building Blocks for Privacy for Machine Learning

3.1 Chapter Overview

This chapter aims to achieve two goals: introducing the concepts, denoted as **PrivacyML**, for the machine learning privacy methods used in the following chapters [320; 78; 106]. It also provides references and discussions of implementation limitations regarding the state of the art of secure technologies in the context of machine learning.

Previously, Chapter 1 introduced the problem where a system's inputs, such as training data, must remain private. Section 3.2 introduces the notion of input-privacy, setting forth methods on secure computation for machine learning.

As a preliminary, included in this chapter are introductory primers for homomorphic encryption (Section 3.3) and secure multiparty computation (Section 3.4). Particularly, to engineer secure computation for machine learning, Section 3.5 walks through specific examples. Section 3.6 outlines other specific techniques towards privacy protection for machine learning. Together, these methods form the building blocks of **PrivacyML**. Notations are collected in Table 3.1.

Notation	English Description
$\mathcal{F}$	Function
$\circ$	Operation
δ	Kronecker delta
K	Number of Data Owners
$\mathcal{D}_{MO}$	Model Owner's Data
$\mathcal{D}_{DO}$	Data Owner's Data
$\mathcal{D}_{\text{tr}}$	Training Data
$\mathcal{D}_{\text{te}}$	Test Data
$\mathcal{D}_{\text{a}}$	Additional Training Data
$\mathcal{M}$	Trained Model
Enc	Encryption Function
Dec	Decryption Function
keygen	Key Generation
pk	Public Key
sk	Secret Key
pk^*	Common Public Key
sk^*	Common Secret Key
ct	Ciphertext / Cyphertext
pt	Plaintext / Clear text
ϵ	Privacy Parameter
C	Gradient Clipping Threshold
n	Approximation Rounds
$q(\cdot)$	Polynomial
$\mathcal{N}$	Gaussian Distribution

Table 3.1: Summary of symbols in secure computation primer.

3.2 Introduction: Privacy for Machine Learning

Imagine an ideal system that takes inputs x from distinct entities $A, B, \ldots$, which computes and reveals a scalar output $\mathcal{F}(x_A, x_B, \ldots)$, while keeping the input data perfectly secret. If a malicious actor compromises any of the entities, they would not learn more about other entities' inputs (any more than the result itself). In computing on private data for machine learning across model owner and data owner, a desideratum arises:

$$\mathcal{F}(\mathcal{D}_{MO}, \mathcal{D}_{DO}) = \mathcal{F}(\{\mathcal{D}_{\text{tr}}, \theta, \mathcal{D}_{\text{te}}\}, \mathcal{D}_{\text{a}}^{[i]}) \qquad \text{(Desideratum for Share(MO,DO))}$$

where $\mathcal{F}$ is a data appraisal function to facilitate data sharing between the model owner and the data owner, illustrated in Figure 3.1. In Equation Desideratum for Share(MO,DO), all the data should be private throughout the computations, including training data $\mathcal{D}_{\text{tr}}$, testing data $\mathcal{D}_{\text{te}}$, and model parameters θ for the model owner, as defined in Chapter 2.

More generally, an ideal system computes $\mathcal{F}(x_{MO}, x_{DO}, x_{OS})$ for inputs from MO, DO, OS and would not reveal to any other party the underlying data.

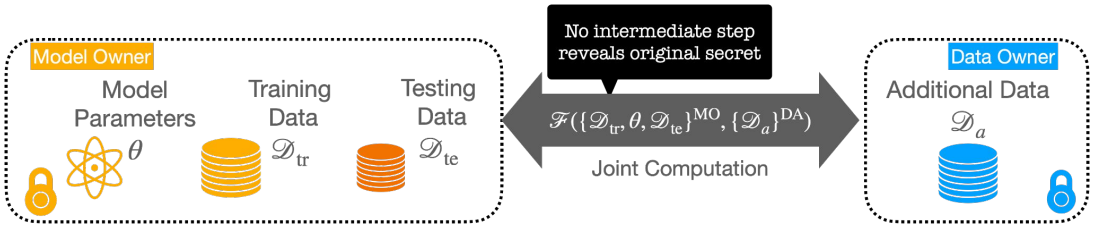


Figure 3.1: A Privacy Desideratum for Machine Learning, Illustrated. Performing joint computation $\mathcal{F}$ on data owned by a model owner (MO) and a data owner (DO), in order to facilitate data sharing Share(MO, DO) in Section 2.3. Ideally, each entity does not reveal their data throughout the computation.

Input- vs. Output- Privacy Prior works identify two privacy protection philosophies [67; 44]:

1. **Input Privacy**, relating to input security [59], describes preserving privacy through se-

curing the *input* to a system, typified by cryptographic methods. This is considered especially relevant when **the input cannot be shared**.

2. **Output Privacy** focuses on the output of a system. "Output-private" methods, such as differential privacy [90], improve the system such that the output does not reveal excessive information about the input (e.g. by controlling privacy leakage). Critically, **the output is assumed released**.

While the philosophies appear symmetric, their associated methods are not directly comparable. As this chapter will later lay out, secure and confidential computation (Input Privacy) offers the guarantee that no additional information is leaked outside the intended output throughout the computation – a cryptographically-protected guarantee – whereas differential privacy (Output Privacy) quantifies privacy loss.

Yet, these philosophies are not exclusive. Differential privacy can strengthen privacy protection for a federated learning system (Section 3.6.1) [321], or synthesize privacy-preserving inputs to a learning system (Section 3.6.3) [153; 322]. Conversely, they can also compete: Chapter 6 compares auditing private models with releasing differentially private models, finding the utility and fairness trade-off of the latter to be undesirable for medical applications. Table B.1 attempts a taxonomy, though it should be noted that not all techniques fit cleanly into this dichotomy.

Tackling Collaboration Bottleneck with **Input-Privacy** Among the two philosophies, input privacy has clear benefits for fostering collaboration: by definition, sensitive data is protected at the source. Where participation may be stalled without the guarantee, this assurance may lead to better trust towards solving incentive conundrums (See Appendix B.1 for further discussion). We now introduce input-private methods leveraged in this thesis for machine learning.

3.2.1 Machine Learning with Secure and Confidential Computation

As an example, consider the appraisal function $f(\cdot)$ from Section 2.4, which requires input from both data owner's data $\mathcal{D}_{DO} = \mathcal{D}_{\text{a}}$ and model owner's data $\mathcal{D}_{MO} = \{\mathcal{D}_{\text{tr}}, \mathcal{D}_{\text{te}}, \hat{\theta}\}$. In an ideal two-party private appraisal, both parties jointly evaluate $f(\cdot)$ on their combined data without revealing that data – including model parameters θ or any intermediate values computed during the function evaluation (Desideratum for Share(MO,DO)). Let $\text{Enc}(\cdot)$ be a generic encoding function with its inverse decoding given by $\text{Dec}(\cdot)$. The private function $f_{\text{priv}}(\cdot)$ performs $f(\cdot)$ (in private) such that:

$$f(\mathcal{D}_{MO}, \mathcal{D}_{DO}) = \text{Dec}(f_{\text{priv}}(\text{Enc}(\mathcal{D}_{MO}), \text{Enc}(\mathcal{D}_{DO}))).$$

Secure and Confidential Computation describes cryptographic techniques that ensure information to be private during a computation. In a simple two-party setup, this "input-private" abstraction allows for joint computation, $\circ$, on disparate inputs A and B. This could be achieved using an encoding/encrypting scheme, $\text{Enc}(\cdot)$, such that

$$\text{Enc}(A) \circ \text{Enc}(B) = \text{Enc}(A \circ B), \qquad \text{(Homomorphism)}$$

The inverse function, $\text{Enc}^{-1}(\cdot)$, is assumed for the eventual output, so $\text{Enc}^{-1}(A \circ B) = A \circ B$.

Given a semi-honest threat model (Section 2.4) where parties are motivated to jointly compute on each party's privately-held data. Two main approaches fit the requirement: **Fully Homomorphic Encryption** (FHE) and **Multiparty Computation** (MPC).

Discussion: **Why Not Make Machine Learning "Incognito"?** While a fully private machine learning development with FHE/SMPC can ensure privacy, its practical deployment remains challenging. Optimizing a model in private hinders visibility of the training curve, which

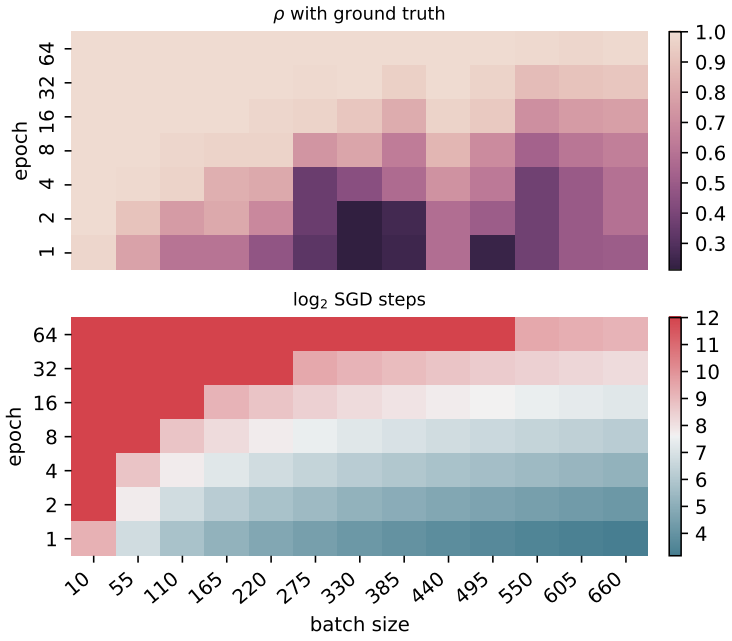


Figure 3.2: In Secure Machine Learning, Preset Hyperparameters Affect Utility and Computational Overhead. When fine-tuning in private to acquire additional data, i.e., re-training after folding in $\mathcal{D}_a$, hyperparameters that result in high utility (Beige) tend to be computationally expensive (Red). **Top**: data appraisal correlation with ground truth plaintext training, with respect to different pre-set batch size and epoch (ρ, higher is better). **Bottom**: convergence in log-steps (log SGD steps, smaller is more efficient). Experiments are run on MNIST under secure SGD using CrypTen [164].

current machine learning workflows require for close monitoring, often with humans in the loop making adjustments. As illustrated in Figure 3.2, hyperparameter choice can affect not only computational cost, but also the accuracy of private training. Lacking the transparency of plaintext machine learning, training parameters may be complex to tune, affecting downstream behavior.

Our work, in contrast, does not rely on private training to stably produce the *final machine learning model*. Instead, when private training is used (Chapter 4 as a baseline and for our method, and as an intermediary step in Chapter 5), it is used to more accurately gauge dataset utility without dictating the downstream model behavior. In data-sharing setups, once the data is exchanged, the training environment is no longer limited to encrypted settings.

General usability problems with using private computation for machine learning are well-documented [334; 180; 329]. Existing work touch on cryptographic primitive design, managing

computational overhead, and closing the expertise gap. Our work extends the discussion to tackling the interfacing of *ML optimisation* and *security engineering*, when applying existing secure techniques to novel ML workloads, with concrete engineering challenges and limitations detailed.

3.3 Homomorphic Encryption

Using confidential computation for machine learning is an active area of research. Chapter 6 describes auditing hospital using Homomorphic Encryption (HE) [78], proposing multiple hospitals to participate via Threshold-FHE [14] in a homomorphically encrypted computation (Desideratum for Share(MO,DO)). Since then, homomorphic encryption has become a prominent approach for enabling other medical applications, such as for precision medicine [104] and on GWAS data [29; 66]. However, even today, Threshold FHE is not implemented in common FHE libraries [127; 7]. Instead of presenting underlying schemas, this section lays out minimal preliminaries for understanding homomorphic encryption from a machine learning perspective.

State of the Art of Pure FHE for ML FHE can support large models in parameter count, but needs significant approximation and can incur large computational overhead. For example, ConcreteML [329], a current leading FHE library that supports large scale machine learning tasks such as transformer inference, requires significant quantization (16bits). It allows fine-tuning on an 8 billion parameter model (LLAMA [292]).[1] When tested on a MacBook Pro, training on 4 examples took 1 Hour, 28 Minutes, and 41 Seconds on CPU. [2]

Our Contribution : Departing from applying FHE to general ML training or finetuning, our work in Chapter 6 [78] paves a novel algorithmic use of FHE: for the practical (and compute-efficient) external auditing of "machine learning fairness" [24] in hospital data, after a model in use. On ConcreteML, over 14 hospitals, a reimplementation of our fairness metric takes 79 seconds to complete on an 2022 MacBook Pro, which is practical for real-world uses.

[1] Note that the weights to update, per LoRA [138], are "in the clear". This means that 0.12% of the weights are not encrypted.

[2] For comparison, in 2014, [36] spent milliseconds to a few seconds per inference for hyperplane decision trees in private. In 2016, cryptonets [112], a single prediction took 250 seconds to complete (however, the same process can make 4096 predictions simultaneously) on MNIST.

3.3.1 Homomorphic Encryption, An Informal Primer

Originally formalized by Gentry [108] in 2009, **Fully Homomorphic Encryption** (FHE) supports both addition and multiplication on encrypted data. Mature FHE schemes, such as BGV [42; 41] and BFV [96], tend to share similar interfaces. Notably, each scheme includes a **keygen** process that takes user-selected parameters to generate keys, akin to generating a password that allows certain operations on encrypted data (where the results can be decrypted). The set of keys is used to encrypt the data.

KeyGen Function In the example computation, BGV/BFV scheme would follow keygen, which produces public key pk, secret key sk. It takes in security parameters n, p, q, noted in Table 3.2. This following procedure is a simplification:

$$\begin{gathered} pk, sk \leftarrow \text{keygen}(n, p, q) \\ \text{Dec}((\text{Enc}(pk, x_1) + \text{Enc}(pk, x_2)) \times \text{Enc}(pk, x_3)) = (x_1 + x_2) \times x_3 \end{gathered} \tag{3.1}$$

Informally, *encoding*, as opposed to *encrypting*, includes data transformations performed locally in plaintext prior to encryption. This "data encoding" is common in FHE literature, which may include formatting data once n is given. These encoding operations are not cryptographically secure, yet happen within each party pre-encryption, therefore private to other parties. We assume that the inputs $\{\boldsymbol{x}\}$ have been transformed. Outputs are also decoded after decryption.

For single-party scenarios, the keygen step can be symmetric where a secret key is used directly, or asymmetric like a "public-key encryption" where a secret key generates public keys [82]. After keygen, data encrypted with these keys now supports homomorphic operations like addition, subtraction, multiplication, and unitary operations like negation and rotation. A scheme that allows both addition and multiplication is called *fully* homomorphic. Additional operations may be supported, depending on the scheme. To manage deep circuits, an *evaluation key* is typi-

cally generated during keygen to keep track of relevant data.

When multiple parties are involved, as described in Desideratum for Share(MO,DO), each party aims to encrypt its input in FHE and perform computations on their combined data without revealing the inputs. Yet, key sharing across different parties is clearly not ideal, as it compromises privacy.

Addition and Multiplication Suppose we want to compute $(x_1+x_2)\times x_3$. Under the principle of homomorphism, we want to have an encoding that also allows for *operations*, like addition and multiplication, to run on encrypted data. We want the homomorphism of

$$\mathrm{Dec}((\mathrm{Enc}(x_1) + \mathrm{Enc}(x_2)) \times \mathrm{Enc}(x_3)) = (x_1 + x_2) \times x_3 \tag{3.2}$$

3.3.2 Distributed FHE For Multiple Parties

Threshold FHE emerges as the most suitable solution for this setup. The main intuition by Asharov et al. [14] is the keygen function (in common setups) being homomorphic under summation:

$$(pk^*, sk^*) = \sum_i (pk_i, sk_i) \qquad \text{(Key Homomorphism (based on BGV))}$$

This enables a design where each party broadcasts its own public-key, pk_i. If all parties are honest, the sum can be computed as $pk^* = \sum_i pk_i$, yielding a common public key. Observe that now each party holds their own secret key, which sums up to the common secret key! Moreover, if each party decrypts a ciphertext c under pk^* with its own secret key, the partially decrypted messages sum up to the complete message. If this over-simplistic system worked, we would have an efficient distributed key generation!

Necessary Adjustments In practice, distributed key generation requires significant adjustments. "Smudging" with noise ensures security of key generation; otherwise, a few sk_i's can

BFV	Parameter Description
n	Ciphertext Dimension
p	Plaintext Modulus
q	(Maximal) Ciphertext Modulus

CKKS	Parameter Description
n	Ciphertext Dimension
f	Fractional Bits
q	(Maximal) Ciphertext Modulus

Table 3.2: Security Parameters Overview. Left, BFV [96]: n roughly relates to the length of the integer input, p needs to be large if the input is high precision, and q needs to be large to accommodate computation. **Right, CKKS [62]**: n, q are the same as BFV's. f corresponds to the desired accuracy of the computation, and can be adjusted in a "rescaling" procedure; large f increases accuracy.

reveal the message. Bootstrapping, an expensive re-evaluation operation in the encrypted setting, is additionally required. In our work engineering an FHE-friendly function in Section 6.4, bootstrapping is undesirable thus avoided.

3.3.3 Engineering Paths

Notably, *engineering FHE for machine learning computation* is a nascent field. At the time of my work in [78] described in Chapter 6, however, BGV had not been implemented in SEAL [60]. Fortunately, relatively mature FHE schemes, including BFV, BGV, and CKKS, are now implemented in many libraries [60; 7; 64]. However, these schemes require users to assign parameters (BFV and CKKS in Table 3.2), similar to C++ flag values or "slurm run" parameters. The security parameters decide program suitability for the hardness of the underlying cryptography. Additionally, performance engineering is crucial due to bit-limits. This section gives guiding principles **for practitioners of machine learning** to get started in programming FHE.

Choosing FHE Schemes Scheme selection is challenging by itself, akin to model selection in machine learning[3]. It involves science, design, and heuristics. Ciphertext space sensitive to the *depth* of the computational circuit [114]. Thus, iterative programs can be memory-intensive, potentially causing issues with small batch sizes without any gradient clipping and normalizations. CKKS [62] is designed for numerical data with some precision loss, which is beneficial for ML op-

[3]That is, before transformer-based architectures ushered us into the era of "foundational models".

erations like *MatMul*. Empirically, some data and computation tolerate overflow and underflow; or, some functions can be re-written to avoid precision loss in CKKS by doing fewer iterations without altering behavior.

Parallel Over Sequential Even for simple plaintext computations, FHE ML programs push computational limits. Nevertheless, parallelization is "free" in FHE, as it does not increase circuit depth, meaning it scales according to the expectation of its plaintext computation when parallelized [175]. In Chapter 6, histogram aggregation takes advantage of parallel operations which scales to multiple hospitals, while decreasing the computation depth of naive implementations, allowing the fairness computation to be practical (1 minute 19 seconds on a laptop).

Parameter Selection Parameter selection for a given scheme is a systems design problem. Small choices could change a program's computability, given *bit-limited* systems. This is analogous to a model training at its memory limit: small tweaks would overflow memory, cause the program to stall or crash, or result in undefined behaviors. Oftentimes, without understanding the scheme, a lookup table is needed [174]. These lookup tables have been recommended, and henceforth proposed, as part of homomorphic encryption standardization [174; 9][4].

For our 2019 work in Chapter 6, we utilized the "automatic" parameter selection function available in SEAL (See Section 8 in [60]). For programmer-supplied summaries about the function and desired security level, it suggested potential sets of parameters for given user-supplied summaries about the function. Even by 2022, this process remained difficult for enterprise users [261].

Framework Considerations and Future Work Of the many FHE frameworks, HElib [128; 127] and OpenFHE [8] focus on supporting a diverse variety of FHE schemes, while SEAL [60], and ConcreteML [329] have stronger design goals to support general machine learning with FHE.

To develop the auditing framework in Chapter 6 [78], we used SEAL in 2019 because it was

[4]See page 18 in [57] for recommended parameter tables.

available. At the time, automatic parameter selection was available but encryption schemes like BGV were not fully available[5]. In 2025, ConcreteML is additionally used to benchmark the fairness algorithms, because it readily supports more deep learning operations with code examples, which we also discuss in Chapter 6. While OpenFHE provides extensive support for cryptographic primitives, its focus on machine learning is less intense, with only bare-bones support for logistic regression (i.e., as an educational code example) [7]. Nevertheless, it has the potential to support more in the future, enabling richer computations for machine learning workloads.

One limitation is that current FHE for ML frameworks can guide parameter selections, yet not fully automate them. For example, Zama's ConcreteML chooses parameters based on user-selected strategies at compile time, which is more streamlined than the programmer choosing from a menu. Yet, distilling the computation's properties is not generally realistic to application programmers without assuming strong algorithmic understanding. Notably, in machine learning, frameworks like TensorFlow [1] and XLA [257] take care of graph-building and figure out the correct compile parameters to support the computation, at the cost of expecting more from the programmer to understand the concept of "dataflow graphs". This points to a future design where computational graphs for FHE are *integrated* with security settings at the compiler level.

Engineering Maturity Programming FHE still requires significantly domain expertise and collaboration with cryptographers, as the level of required knowledge exceeds that of most machine learning researchers [180]. This is starkly contrasted with deep learning, where PyTorch programs have numerous runnable examples. Existing machine learning code in FHE tends to be one-off and research-based [36; 66]. This lack of infrastructure hinders protocol-level programming, as its security is hard to guarantee. Our work relies on framework correctness for security and privacy, and numerically verifies input-output consistency between encrypted and plaintext operations for correctness.

[5]until HElib itself was integrated into SEAL through a collaborative effort [211]

Why Not Move Machine Learning to FHE? Deploying FHE for machine learning, especially at scale, is an active area of research (e.g., [180]). Large models often require aggressive quantization [329], and potential adversarial scenarios can complicate operations, such as for SEAL [242] and CKKS [62]. Specific to machine learning use cases in SEAL, Li and Micciancio [184] notes that an adversary obtaining some results may compromise encryption. Nevertheless, FHE's strong security guarantee is highly appealing and holds great promise for ML. This thesis contributes to a use of FHE in ML in Chapter 6 by exclusively examining machine learning model inputs and outputs, taking advantage of FHE's security guarantees without incurring approximations in model training or inference.

3.3.4 Situating Our Contribution: Hospital Fairness Auditing

Instead of applying FHE to ML training and finetuning, our work in Chapter 6 [78] paves a novel algorithmic application: leveraging FHE for the practical (and compute-efficient) external auditing of "machine learning fairness" [24] in hospital data, after a machine learning model is deployed in use. This responded to the anticipated need for oversight of discriminatory practices in proprietary models [99]. Unlike Kilbertus et al. [161]'s MPC-based machine learning fairness certification that required private training, our proposal enabled post-deployment, external ML fairness audits using FHE, which was not available at the time of the proposal.

Our contribution lies in providing conceptual feasibility for FHE to be practical for machine learning use cases, and for the algorithmic descriptions for continuous and real-time private ML fairness audits, distinct from the exploratory and aggregate data analyses suggested by contemporaneous FHE works [262; 103; 105].

Recently, Park et al. [237] subsumed our working by developing a comprehensive confidential computation stacks for fairness auditing,

3.4 Secure Multi-party Computation (MPC)

Secure Multi-party Computation (S-MPC) or **MPC** [324; 264; 59] allows two or more parties to compute a function over private inputs, without revealing information other than the final output. This section frames our contribution and introduces key concepts for understanding MPC in a machine learning context.

Our Contribution This thesis connects data sharing conundrum to data appraisal problem, and solves it with secure computation. In doing so, it presents the first private implementations of any influence function (Equation Forward Influence Function) and a dataset divergence measure $\mathrm{KL}_{\mathcal{XY}}$ (Equation KL-XY Score) leveraging MPC. Both are novel dataset utility measures that can be used for assessing the value of additional training data. In data-sensitive domains, our work's main impact is the *addition* of privacy, without which collaborations become stalled. Enabling dataset utility to be assessed in private fosters collaboration where privacy is highly valued.

3.4.1 Multiparty Computation for Machine Learning

Our Abstraction In this thesis, data is disparately-owned, and we describe overall MPC system is described in a simple "two-party" setting, meaning that two parties are involved in one collaborative computation. As already put forth in Section 2.5.2, both parties are aware of the scalar function and each side's data types. The MPC system focuses on machine learning:

1. **Preprocessing.** Each party preprocesses their data, such as the machine learning model or training set. This preprocessing may include but not limited to transforming their inputs to reformat, normalize or scale, "encrypting" the computation [324], or breaking the data into shares Shamir [264], before sending it externally.

2. **Computing.** The data is exchanged (after secure transformation) and computed on, but

never revealed to each party until completion. This guarantees that intermediate exchanges of data, if any, do not reveal information about the other party's data throughout the computation. Under the hood, cryptographically-secure protocols are employed, ensuring the security of the computation, which may include homomorphic encryption (Section 3.3).

3. **"Decrypting".** Assuming that the computation successfully completes, then the result – and only the result – is revealed. Broadly speaking, it includes securely reconstructing inputs [264; 114]. Thus, no data additional to the output is shared.

While simplistic, this abstraction recognizes the key demarcation between the different players in machine learning, as put forth in our three-party privacy thesis (Chapter 2.4). The friction between them, reflected in data sharing tension, is directly addressed with MPC guarantees.

Necessary for Machine Learning: Approximated Schemes Machine learning operations that are non-linear are approximated to adapt to MPC while maintaining performance. One example is the exponential function used in SoftMax [32]. By default, it is implemented via

$$e^x = \lim_{n\to\infty}(1+\frac{x}{2n})^{2n}, \qquad \text{(MPC Exponentials)}$$

Reciprocals are also approximated, Knott et al. [164] uses the Newton's method [327]:

$$\frac{1}{x} = \lim_{n\to\infty} y_n = y_{n-1}(2 - xy_{n-1}), \qquad \text{(MPC Reciprocals)}$$

for y_0 set to $^3\sqrt{e}/e^x + 0.003$. As n increases, these operations are more accurate, but less efficient. Typically, $n = 8$ [164] as the default approximation.

Computational Overhead in ML with MPC MPC takes advantage of distributed data but may incur communication rounds. In cross-silo setups where the participants are geographically far, communication costs can dominate. One example is $\text{ReLU}(x) = x \cdot \mathbb{I}(x > 0)$. Typ-

ically, the comparison ">" requires two roundtrips[6]. Henceforth, ReLU is often approximated with GeLU [132] due to ReLU approximation's unfavorable communication-accuracy tradeoff.

The runtime of an algorithm is also a crucial metric, because the MPC version of a program runs differently than its plaintext version. This is usually traded off against approximation accuracies. For example, in a 12-layer transformer, softmax function and takes up > 50% of runtime for each inference while GeLUs [132] take up about 23% [164; 186; 199].

Though our experiments do not pertain to transformers, similar performance engineering principles apply given the shared computational constraints. GeLU optimisation, for example, can preserve the private model's performance with less runtime by reducing the number of rounds of communication (but cannot improve overall performance). Later, Section 3.5 describes a similar optimisation, where influence approximations bypass expensive round trips from private training, resulting in runtime improvements at scale, which we will expand on in Chapter 4.

Relationship With Homomorphic Encryption *Multiparty computation* has a rich history and is thus the umbrella term for distributed secure computations [324; 264; 115; 114; 14; 17], including distributed homomorphic encryption. In MPC, some computation can leverage data that is local[7]. Many components under MPC are made with HE protocols, such as matrix multiplication in transformers [197]; as a result, pure HE researches results often contribute to MPC, such as improving round-efficiency [233].

Comparing with the State of the Art As machine learning models scale, MPC technology is also rapidly improving, albeit with a large computational overhead.

When performing *inference* on large transformer-based models, a LLAMA 7B [292] model takes 5 minutes to generate 1 token in 2023, under the goal of not incurring any model accuracy

[6]Via the "binary share to additive share" conversion, which takes two rounds of communication.

[7]As a result, when data is already at different parties to begin with, some MPC schemes such as secret-sharing [264] can be more performant than pure distributed-FHE methods that are not interactive i.e., when the data is encrypted once and sent for remote computation.

drop [85]. When a small accuracy drop is allowed, the overhead is much reduced. CrypTen [164] inference used to take 71 seconds on $\text{BERT}_{\text{base}}$ in 2021. By 2023 and 2024, this number decreased to just 19 seconds [199; 186], with a small drop in model accuracy compared to plaintext models[8]. However, arranging a data collaboration to build large models increasingly requires a high budget, and is often thwarted by the need to maintain data privacy.

Instead of using MPC to perform training or inference, this thesis proposes using privacy as a direct incentive *towards* data sharing, similar to Azar et al. [17]; Zheng et al. [333]'s work in instrumentalizing privacy to enable data collaboration, our approach specialize in solving the data acquisition problem in modern machine learning. Our specialization paves way to *increase* the amount of (agreed upon) data collected while respecting privacy, without the mandate to scale private training in accordance with non-private state of the art. So although our work does not focus on serving transformer models, our methods readily accommodate different types of data machine learning will require.

Our novel use of machine learning optimization in MPC does not engage or compete directly with private training (or inference). Nevertheless, it stands to benefit from the performance gains developed for them (e.g., activation function approximation by Mohassel and Zhang [215] speeds up the logistic model in Chapter 5's method), while contributing to better data-sharing ecosystem. In particular, Chapter 4 uses private training as a baseline, which our method outperforms[9], while Chapter 5 uses private training as an intermediary step which can be sped up.

In summary, **as larger and larger models become practical in MPC, so does the applicability of dataset utility measures.**

[8] When run on 3 Tesla V100 servers with a 10GB/s bandwidth On $\text{BERT}_{\text{large}}$ and $\text{BERT}_{\text{base}}$, MPC versions of the model drops in accuracy by about 5% with due to SoftMax approximations (Table 2, 3 in [199]).

[9] The performance lag of private training is due to the sequential nature of training, which is likely incorrigible through faster training.

3.5 Engineering Secure Computation, In Detail

Influence Functions with MPC Recall the influence formulation presented in Section 2.4.

$$\mathcal{I}(\mathcal{D}_{\text{a}}) = -\boldsymbol{H}_{\hat{\theta}}^{-1} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\text{a}}} \nabla_\theta L(\boldsymbol{x}, y, \hat{\theta}), \qquad \text{(Forward Influence Functions)}$$

In Chapter 4, it is applied to appraise the value of additional training data, with respect to the Model Owner's existing model and data.

$$f_{\text{if}}(\mathcal{D}_{\text{a}}) = -\frac{1}{|\mathcal{D}_{\text{a}}| \cdot |\mathcal{D}_{\text{te}}|} \sum_{(\boldsymbol{x}_{\text{te}},y_{\text{te}})\in\mathcal{D}_{\text{te}}} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\text{a}}} \nabla_\theta L(\boldsymbol{x}_{\text{te}}, y_{\text{te}}; \hat{\theta})^\top \boldsymbol{H}_{\hat{\theta}}^{-1} \nabla_\theta L(\boldsymbol{x}, y; \hat{\theta}).$$

In the summand, separate the different roles

$$\underbrace{\nabla_\theta L(\boldsymbol{x}_{\text{te}}, y_{\text{te}}; \hat{\theta})^\top \boldsymbol{H}_{\hat{\theta}}^{-1}}_{\text{Model Owners}} \nabla_\theta L(\underbrace{\boldsymbol{x}, y}_{\text{DO}}; \overbrace{\hat{\theta}}^{\text{MO}}). \qquad \text{(Data Ownership)}$$

Notice that the Hessian inverse is computable entirely on Model Owner's data. Avoiding private computation is significantly cheaper than joint computations in MPC. This insight leads to rewriting

$$f_{\text{if}}(\mathcal{D}_{\text{a}}) = -\frac{1}{|\mathcal{D}_{\text{a}}| \cdot |\mathcal{D}_{\text{te}}|} \underbrace{\underbrace{\sum_{(\boldsymbol{x}_{\text{te}},y_{\text{te}})\in\mathcal{D}_{\text{te}}} \nabla_\theta L(\boldsymbol{x}_{\text{te}}, y_{\text{te}}; \hat{\theta})^\top \boldsymbol{H}_{\hat{\theta}}^{-1}}_{\text{Plaintext Computation (MO)}} \underbrace{\left(\sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\text{a}}} \nabla_\theta L(\boldsymbol{x}, y; \hat{\theta}) \right)}_{\text{Joint Computation}}}_{\text{Joint Computation}}.$$

This separation limits the secure computation round trips and overhead, and forms the basis of our runtime gain of influence-based appraisals.

Numerically, it is also a lot more stable than computing Hessian inverses in private. This

approach is contrasted with naively doing model training in MPC. In private training, each SGD step requires a *joint* computation, where the batch data and models are owned by separate parties. This leads to necessary MPC overhead, seen on the bottom of Figure 3.2. Furthermore, method accuracy interacts with hyperparameters (See Figure 3.2 top).

3.5.1 Explaining the Efficacy of Influence-Based Appraisal in MPC

Influence functions are particularly suited for MPC, for two key properties: **computational efficiency** and **numerical stability**. First, it evades the most expensive computation (empirical Hessian inverse $\boldsymbol{H}_{\hat{\theta}}^{-1}$), which can be done in plaintext exclusively by one party (Model Owner), thus 'saving' on private computation, as Equation Data Ownership shows. Second, it evades the numerical instability and high runtime cost of sequential training loops[10].

In terms of adverse downstream impact, "adding" privacy to influence functions does not trade off its accuracy. Equation 3.5 requires only a few private operations that are known to be stable – primarily encrypting/decrypting, taking the gradient of a model, and performing one matrix multiplication on an entire batch of data – while side-stepping the thorny issues of hyperparameter selection by having no extra parameters to tune. In contrast, private training becomes susceptible to yielding more approximate results under different hyperparameter selections (Figure 3.2).

Direct Comparison with Private Training Though private training more closely mirrors what we desire in plaintext, merely *introducing* privacy by emulating model training with private computation hurts its utility through two veins: 1. by obscuring the training process, thus adding uncertainty, and 2. by requiring sequentially approximating numerical operations. By being better-suited for MPC, influence-based appraisal consistently outperforms private training-based appraisal, as Chapter 4 presents in Table 4.1.

[10]This is required of private training followed by private test loss difference in our baseline method called *Finetuning-based Appraisal* in Chapter 4.

Gap and Opportunity Theoretically all computations can be MPC [115], yet, constructing a performant implementation is nontrivial. No MPC production-scale library in Health ML is dominant — existing frameworks, including ours, are research-based [66]. One engineering hurdle of using MPC frameworks is the difficulty of secure computation engineering [215; 157; 333], compounded in our works in Chapter 4 and Chapter 5 with machine learning engineering. Yet, engineered MPC programs can be easily deployed. Even small organizations can deploy MPC without any specialized hardware e.g., secure enclaves. Thus, the algorithms developed and shared in our works can be readily deployed by small organizations today using a laptop computer.

3.5.2 Choosing CrypTen For Private Data Appraisal

When privacy is paramount, secure computation frameworks guarantee provable security of their underlying implementations. This leaves algorithmic design and performance engineering as primary concerns for implementing machine learning tasks. However, compared to standard machine learning systems like PyTorch [239], multiparty computation frameworks tend to be slow to develop [215; 56; 254; 277; 157; 164; 333]. Pre-existing MPC frameworks, such as MP-SPDZ [157] which had just emerged at the time of my work [320], lacked deep learning support[11]. Though CrypTen [164] was still under development, it was chosen for its support of machine learning-related operations, such as gradient descent over floats, while maintaining closeness with PyTorch in both interface and implementation structure. This developmental path was key to its relative ease-of-use, as the coding mimics that of machine learning development itself. Further, it offers an ease of transitioning from training a model (in plain text) to appraising data with the model (in MPC), as posited in our setup in Chapter 4, because trained plaintext PyTorch models can be readily encrypted.

My work realized a novel use case for MPC for ML, which uses influence functions to appraise

[11] Deep learning support refers to performing model training (as opposed to just inference), and using GPUs. See Table 1 in [Knott et al.] for the state of MPC frameworks in 2021.

the relative benefits of *unseen* data before collaboration. Additionally, it was the first forward influence function implementation in MPC. Since then, the CrypTen framework was released as a pure research library, sparking later works [199]. Following this vein, our more recent work detailed in Chapter 5, which proposes the first dataset-to-dataset divergence-based evaluation in MPC, is implemented using CrypTen [106].

3.5.3 Machine Learning Engineering Discussions

Engineering Maturity MPC computations require careful engineering, as naive implementations take days or run out of memory [133; 36]. CrypTen [164], once an early research effort at the time of my work in Chapter 4 [320], has been increasingly leveraged for its basic ML functionalities e.g., SGD, ReLU approximations. Currently, programs using CrypTen face some key engineering hurdles: 1. Precision configurations and normalizations (to avoid blow-ups and underflows), 2. Error control and performance engineering for machine learning in bit-limited space, and 3. Debugging private computations and hyperparameter tuning. These key challenges mean engineering machine learning workflows in private faces different challenges as machine learning engineering in plaintext. For example, in plaintext, the Forward Influence Function is considered expensive and memory-intensive. Yet it gains an advantage in private, as it requires no hyperparameter tuning, while not requiring many sequential operations. Its memory intensity stays, but this plaintext component becomes relatively inexpensive.

Approximations and Experimental Design MPC Tensor Encoding [164] introduces floating point differences between encrypted and plaintext computations, potentially affecting results. For example, the parameters from SGD may be rather different between plaintext and encrypted models, as it is a separate training process, where each iteration slightly differs, even if the batches of data and their random seeds are fixed. Therefore, any MPC program in machine learning needs to assure consistency, where the plaintext function and the encrypted function match.

Research Trends for MPC and Machine Learning Our work uniquely employs machine learning functionalities in CrypTen for the novel purpose of gauging dataset utility, diverging from MPC training and inference. We perform private training (logistic regression using SGD) only as a baseline in Chapter 4, and as an intermediate step in Chapter 5. Since [320], the field of MPC for ML has seen major gains, especially for 2party (2PC) inference [141; 251; 250], with a focus on transformers such as Iron [130] and Bolt [233].

CrypTen's relatively mature framework and its associated developmental ecosystem foster rapid research, as new directions can be easily tested. For example, recent works on engineering transformer-based models in MPC have optimized [85] and modified [199; 186] specific operations, and incorporated distillation [186], largely relying on CrypTen as the "MPC security engine". Our most recent work in Chapter 5 follows this rapid iteration on CrypTen's flexibility for private dataset divergence measurements, tailored for data-limited sensitive domains. Appendix B.3 details *recent trends* in MPC for ML.

3.5.4 Parting Thoughts On Secure Computation for Practical Systems

Cryptographic techniques like hashing and encryption are likely significant, because they can overturn the underlying asymmetry in the world where small players like the data owners may not have much leverage to negotiate against bigger players like technology companies.

Within machine learning, confidential and secure computation generally *preserves*, rather than approximates, the output. In contrast, differentially private training lowers accuracy (Sec 3.6.2), while federated learning (Sec 3.6.1) alters the model behavior for enhanced privacy. However, FHE and SMPC are not magic bullets that solve all confidentiality problems. Embedding machine learning in systems that require strong data security assurances, these techniques should be practised in tandem with complementary privacy-enhancing features, such as differential privacy and federated learning, secure systems solutions like verified hardware and secure enclave, and data security measures like strong passwords, access control lists, and logging.

3.5.5 Summary

1. Fully-homomorphic Encryption (FHE) supports addition and multiplication to be computed on encrypted input. While FHE is the gold standard for computing on encrypted data, it is non-trivial to adapt to modern machine learning due to computational overhead. **Workflow Challenges**. Usual FHE uses lattice-based schemes requires periodic "bootstrapping" (key refreshing and re-encrypting with lower noise terms known as "recrypting"), such as using the CKKS scheme [62]. This adds cryptographic parameters that are difficult for non-experts to set, yet nevertheless crucial for efficient computation. Our work, in contrast, sets up a system that does not require expert intervention.

2. Secure Multi-party Computation (S-MPC) [324; 264] allows two or more parties to compute a function over private inputs, without revealing information other than the final output. In practice, the protocol of key exchanges, encryption schemes, and communication ensures that only encrypted data is transmitted outside its owner's control. S-MPC grants that once the data is transformed, it cannot be recovered while computation is happening. Though faster than FHE, its engineering difficulty and communication overhead may prevent its adoption. **Workflow Challenges**. Traditional private training assumes data is never revealed, hindering model development tasks like inspection, monitoring, debugging, and sometimes parameter-tuning, which often rely on seeing the data. Such a rigid setup for model training is unappealing. Our work, in contrast, aids model owners with appraisal values computed in private, prior to the exchange of data, maximizing flexibility (Section 4).

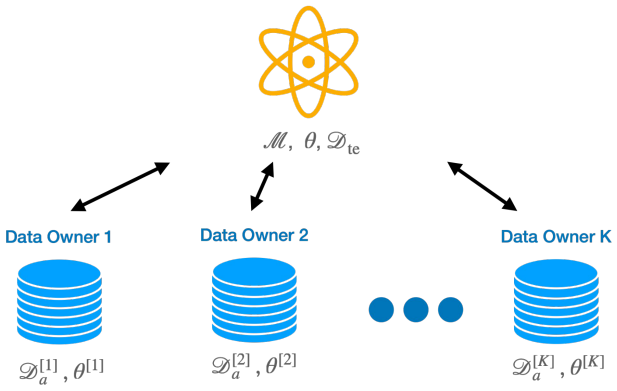


Figure 3.3: Federated Learning with Data Distribution. Each client owns data, and the server "pools" their data. Naively, all data is sent to the server. In federated learning, to minimize communication, no raw data is sent – instead, parameter updates between the server and the clients are sent intermittently. This respects data locality, but still confers information between the parties.

3.6 Privacy-Enhancing Techniques

To mitigate data-related tension between the three entities, various approaches have been considered. This section summarizes high-level approaches towards building machine learning with nontrivial privacy protection, with a deeper focus on discussing Federated Learning (FL) (Sec 3.6.1), and Differential Privacy (DP) (Sec 3.6.2).

3.6.1 Federated Learning (FL).

Since 2016, McMahan et al. [207] sparked a field called **Federated Learning** (FL) [189; 255; 33; 154], which allows siloed data to be computed together in a distributed fashion, often co-designed with systems facets – communication, speed, bandwidth – as well as optimisation challenges. FL relates to our work by supporting collaborations, but incurs an optimisation-privacy trade-off.

For optimisation, [207] belongs to the class of data-pooling methods which entrust each "client" to compute a local model, whose gradients are updated regularly in a trusted central server. Though federated learning has the potential to utilize more data, achieving optimal results through combining local models may be challenging. The obvious drawback, as [189] notes

and [252] theorizes, is that when the data is very heterogeneous, just pooling more data might not help with overall optimisation.

Privacy gain is a bit complicated, however. In our setup, consider jointly training a model on additional data $\{\mathcal{D}_a\}$, a trusted third party $\mathcal{T}$, such as a Google service, would execute the computation without seeing the raw data. In practice, however, the service provider is often incidentally the model owner, such as Google, for their own products that gather data, such as Chrome or YouTube[12]. This means the party from whom the data owners want to be private may be implementing and orchestrating the very system.

Meaningful, but not Privacy-Preserving A purported privacy benefit is that FL bypasses the sending of raw data to the server. That is, only model parameter-relevant information is sent, keeping sensitive data on device (Figure 3.3). Yet this is a misnomer: privacy is improved, but not *preserved*. While FL avoids sharing raw data, it shares critical information, such as gradient updates, that is nonetheless sensitive.

Privacy Discussion In practice, federated learning utilizes more data for optimizing a shared model, which strengthens data governance but harms data propriety – each data owner in the federation does not share data, as is. Because of that, it is sometimes considered "privacy-preserving" and the future of digital health [255; 266; 31]. However, the federation by itself is vulnerable against simple attacks that reverse-engineer information about the data, even if the data is not fully shared [330; 107; 31; 335; 200; 220].

One reason is the inherent difficulty in securing a system with many actors, where traditional security measures are no longer sufficient [291; 97]. This problem is made worse in machine learning, as the information shared across, such as gradients or confidence, are inherently useful to model inversion attacks [102; 100]. Moreover, where federated learning is used, the data tends

[12]Notably despite the suboptimal model performance, federated learning is often deployed by model owners, as the systems gain is meaningful.

to be heterogeneous, thus potentially "easier" for classifier-based attacks to identify [332]. Nevertheless, the abstraction itself still represents a huge step forward from arbitrary sharing and copying raw client data.

Mitigations: *Additional* Privacy Privacy incentivizes federation [297]. Specifically, preserving privacy between the parties under federated learning requires secure computation methods – FHE or MPC [295; 34; 279; 66; 104] – combined with differential privacy [321; 297; 16], trusted execution environments [213] – an approach that is coined "Privacy-in-Depth" in [154]. Cross-siloed federated learning, on the other hand, approaches privacy by getting rid of the assumption of a trusted third party (server), and has been proposed for medical domains [315; 66].

Our Contribution Performing data appraisal with MPC can be seen as an extension of data federation, by adding a separate privacy-preserving component *before* all parties commit to an entire federated learning system. It is low-commitment, as privacy is preserved by default, and there is no requirement to continue in any system; it tackles the incentives' problem blocking collaboration. By demonstrating utility, the two parties foster trust.

As a systems component, it is flexible, because it can respect every role's privacy requirement while avoiding the thorny trade-off that alters the downstream model's behavior [320; 106]. My works complement the line of ambitious systems that incorporate MPC in potential federated scenarios where the participants are semi-honest [323; 34; 66].

Future Work Relating to solving incentives problems in federated learning that includes both input- and output-privacy, Collaborative Machine Learning (CML) [139; 298] emerged as an area of active research. Notably, because the appraisal stage is a separate component, it is adaptable towards many future pre-commitment setups. Pre-commitment represents a different threat model than the model training itself, because after trust is fostered, the threat model may shift, and some prior mistrust may evaporate (or intensify).

3.6.2 Differential Privacy (DP).

While secure computation in Section 3 protects privacy via input privacy, Differential Privacy preserves privacy of the output. Let privacy budget $\epsilon, \delta > 0$, local differential privacy is defined for an algorithm or model $\mathcal{M}$, on datasets D_1 and D_2 which differ by 1 element.

$$\mathbb{P}[\mathcal{M}(D_1) \in S] \leq e^{\epsilon}\mathbb{P}[\mathcal{M}(D_2) \in S] + \delta \qquad ((\epsilon, \delta)\text{-Differential Privacy})$$

where the higher ϵ is, the less private $\mathcal{M}$ is. Here, S are subsets of $\mathcal{M}$'s image.

Originally put forth in 2006 by Dwork [90], Differential Privacy has moved from mathematics to real-world systems and deployment. It is an influential method that was used in the Census collection to limit the information shared and in RAPPOR at Google [92; 93].

DP for machine learning Though this thesis mainly uses secure computation, DP algorithms have mathematical guarantees regarding its privacy leakage, making them narrowly privacy-preserving (See Table B.1). Intuitively, a learning algorithm with DP-guarantees can limit the influence of an individual training example on the output. For machine learning, a common approach to transform a learning algorithm into a differentially-private algorithm add noise to the training example, e.g., perturbing the loss [25; 160]. The most popular method, **DP-SGD**, was put forth in 2016 [2]. It injects noise at the mini-batch at optimisation time with the intuition of bounding the weight influence on every batch update.

Discussion Sharing data with differential privacy preserves privacy to a certain extent, but has significant drawback in performance in machine learning contexts [2; 301]. Namely, [98] suggest that a large model relies on duplicate data to learn, and ignores data that is "long-tail" in terms of frequency. This so-called "privacy-utility trade-off" is verified in healthcare scenarios by Suriyakumar et al. [281], leading additionally to a fairness-privacy trade-off where minority

groups are disparately impacted.

An additional problem is more subtle: because DP requires meaningful privacy parameters, which are set by the data collecting party that also trains the models, its actual privacy is thus hard to audit. Empirically, the final product is often not as private as claimed, as in the case with MacOS [283]. [301] further showed that private prediction often breaks theoretic privacy assumptions. Despite performance hit and their difficulty in auditing, differential privacy generally enhances privacy in many practical scenarios. Local differential privacy is widely used in combination with federated learning for distributed devices, such as cell phones, to not collect raw data on central servers [269; 74]. Additionally, performance does not degrade much, when only fine-tuning data is differentially private [190; 328]; however, it also suffers from linkage attacks, due to data in training not being protected, which may be overlapping with sensitive data [294].

Situating Contributions My works complement DP-based data sharing for machine learning, as secure computation methods can enable continuous use of encrypted data without needing any access to the raw data, preserving individual privacy like DP, while enabling audits, appraisals, and evaluations[13]. My works showcase the potential for secure techniques to ease incentives tension between the disparate parties and present auditable and performant use cases, providing additional enhancements, when differential privacy falls short in either privacy protection or incentives.

3.6.3 Other Privacy-Enhancing Techniques

K-anonymity and Synthetic data. Transforming data into a similar form that desensitizes certain attributes can be desirable [260; 86; 137; 227; 118; 282]. For example, Ng et al. [222] effectively improves out-of-domain generalization with generating synthetic samples from denoising existing ones and reconstructing them, before augmenting these generated samples into train-

[13]Potentially, solving these areas can move towards safe sharing of selective raw data, in encrypted formats, utilized *fully* for the purpose of better optimization.

Privacy Mitigations	Limitation	Seminal methods	Framework
Differential Privacy dampens privacy risks in sensitive data	Performance loss, disparate impact on underrepresented groups.	DP-SGD [2]	OpenDP Opacus [326]
Federated Learning improves data governance and increases data	Governance only, not a formal privacy mechanism. Susceptible to privacy attacks.	FedAvg [207]	OpenFL, Flower
Fully-homomorphic Encryption enables outsourced secure computation	Compute overhead. Usability. Bitwidth-limited → Gradient underflow.	CKKS [62]	SEAL [60] ConcreteML [329]
Multi-party Computation allows model and data owners to work together securely	Communications and compute overhead. Often limited subset of operations.	secret sharing schemes [264]	CrypTen [164]
Trusted Execution Environment supports isolated secure computation through hardware + software	Side channel-attack risks. Trusting the hardware maker / provider.	Confidential-VM [188]	OBLV Deploy
Machine Unlearning rectifies private data from trained model	Conceptually complex; brittle. Risk is a lower bound (membership inference).	Membership Inference [270]	Cleverhans[39] academic code [225]

Table 3.3: Summary of privacy mitigations for machine learning.

ing, potentially side-stepping having to collect extensive private data. Yet, to still preserve the utility of the dataset transformed for analytics or learning tasks is challenging by itself [152]. Additionally, outside the scope of sensitive data that is transformed, little privacy guarantee is available, leading to re-identification risks [219; 152]. Our works in contrast, allows for sharing real data, without sacrificing either privacy or optimization. Additionally, secure computation can be combined with synthetic data augmentation techniques, where the dataset to be shared is synthetically generated (but still protected, as it is based on real data), and only gets shared if it is effective for optimization (Chapter 4).

Learning under data encryption. Simply encrypting data for machine learning tasks is insufficient[14]. Model training often requires human intervention, hindering fully private "black box" approaches. Additionally, current private computation methods struggle with the complexity and scale of modern models [178].

Data valuation. Though my work can be used to "set a price" on data, it does not formally deal with pricing as an explicit goal. Many other works aim to price part of training data for its effect to the resulting model [192; 147; 111]. While data valuation can enable equitable data sharing, Shapley value-based sharing is more suited for data federations where multiple parties have already contributed – presuming, effectively, an existing agreement to collaborate. This does not necessarily solve sharing incentives where data owners do not want to share *before* sharing. My work [320], which uses influence functions in the forward direction, estimates the value of data before sharing occurs.

Data marketplaces describe holistic solutions to data ecosystem issues through a market-driven approach. This field takes into account applicable technologies such as homomorphic

[14]Traditional data encryption limits the utility of data and locks it down. Here, data encryption refers to the encryption used in FHE and MPC.

encryption, multi-party computation, federated learning, and various techniques for incentive-compatible optimisations [19; 18; 171]. Incentives and markets are indeed related. While it includes my work on data appraisal under encrypted computation, and is an important application to the problems we address, it is ultimately out-of-scope for this thesis as we focus on privacy tensions without prescribing market solutions as necessarily constructive for resolving the tension.

Reducing memorization through Machine Unlearning. Unlearning undesirable traits from training data after training is related to fine-tuning with a data-driven focus [39; 221; 225]. In these concept unlearning scenarios, the category of sensitive data serves as a guide to the steering of the model as to what to avoid when sampling [195]. A narrower definition of machine unlearning, which we coin as "exact unlearning" in Xu et al. [317], gives mathematically-sound guarantees on the privacy of withdrawn training data, while keeping the full utilization of the resulting model, to be as good as a retrained baseline without the withdrawn data. This type of forgetting is another direction towards mitigating privacy incentives, especially between individual data owners and model training entities.

3.7 Conclusion

Section 3 introduced Secure and Confidential Computation, as used in this thesis. We especially focused on engineering FHE and MPC for machine learning, contrasted with the state of the art in these emerging fields. These trade-offs are made crucial when additionally walked through the example of implementing the influence functions in private in Section 3.5.

To contextualize our techniques alongside other privacy-preserving methods, we contrasted input privacy with output privacy, where we center on achieving provable input-privacy through secure and confidential computation. Section 3.6 surveys privacy-enhancing techniques related through their use for machine learning, notably Differential Privacy and Federated Learning which do not have input-security guarantees.

Yet, these methods address vastly different needs within machine learning privacy. To better illustrate the practical considerations for choosing different privacy mitigations, Table 3.3 summarized privacy mitigations, their usage, and corresponding implementations as of this writing.

Lastly, our appendices expanded on PrivacyML discussions in contrasting techniques (Table B.1) and their application to policy (Appendix B.4).

4 | Data Appraisal Without Data Sharing

One of the most effective approaches to improving the performance of a machine learning model is to procure additional training data. A model owner seeking relevant data from a data owner needs to appraise the data before acquiring it. However, without a formal agreement, the data owner does not want to share data. The resulting Catch-22 prevents efficient data markets from forming, as illustrated in Figure 4.1.

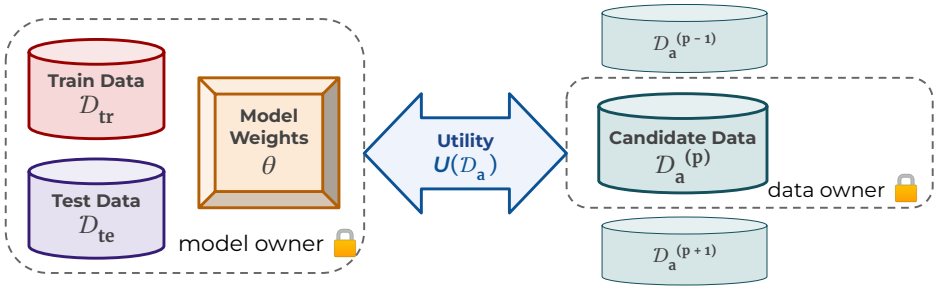


Figure 4.1: Incentive deadlock. By default, model owner's parameters, θ, $\mathcal{D}_{te}$ and $\mathcal{D}_{tr}$, and data owners' $\{\mathcal{D}_a\}$'s are kept private. While both may gain from exchanging data, utility is not realized due to privacy.

To alleviate the paralysis without sharing data, this chapter proposes adding a data appraisal stage (Figure 4.2) that "unlocks" potential data sharing, equitably. Specifically, multi-party computation (Section 3) is used to implement an appraisal function computed on private data. The appraised value $f(\mathcal{D}_a, \mathcal{M})$ proxies utility $U(\mathcal{D}_a)$, which guides data selection and transaction[1].

[1]Code for all the experiments: https://github.com/NorthStar/PrivateDataAppraisal

4.1 Problem Setup

Two-Party Private Appraisal. S-MPC enables two or more parties to jointly evaluate a function on their combined data without revealing that data (which includes model parameters θ) or any intermediate values computed during the function evaluation. The appraisal function $f(\cdot)$ requires as input the data owner's data $\mathcal{D}_a$ and model owner's data $\mathcal{M} = \{\mathcal{D}_{tr}, \mathcal{D}_{te}, \hat{\theta}\}$. Let the E be the encryption function with decryption given by D. The private function $f_{priv}(\cdot)$ performs $f(\cdot)$ with MPC such that:

$$f(\mathcal{D}_a, \mathcal{M}) = D(f_{priv}(E(\mathcal{D}_a), E(\mathcal{M}))).$$

As Figure 4.2 shows, sensitive data does not leave any party's machine without encryption. This lets the appraisal be public and auditable, eliminating the need to trust secure hardware or rely on an intermediate escrow service. Additionally, though every private appraisal is a two-party MPC between a model owner and a data owner, the appraisal methods linearly scales to multiple data owners without repeating the shared computations. In following sections, assume each dataset $\mathcal{D}_a \in \{\mathcal{D}_a^{(p)}\}$ is benchmarked in a private two-party MPC against a fixed model $\mathcal{M}$. In notation, we abbreviate $f(\mathcal{D}_a, \mathcal{M})$ to $f(\mathcal{D}_a)$.

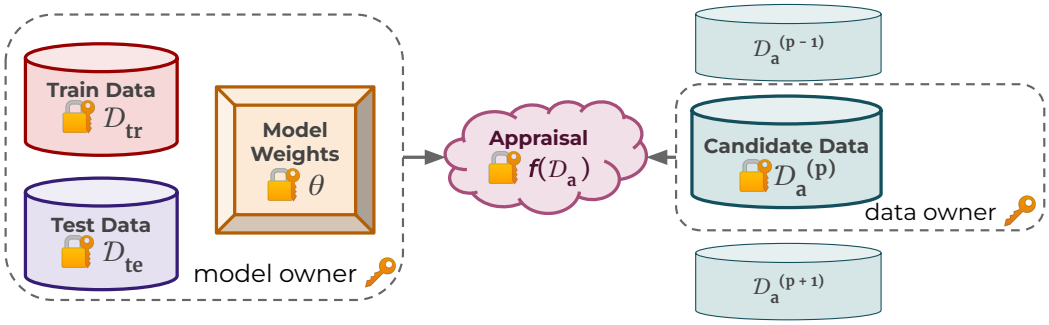


Figure 4.2: Secure MPC. Data appraisal without data sharing. model owner and data owner encrypt their respective data. The appraisal function is performed privately, and its result is revealed to both parties.

4.2 Private Data Appraisal Functions

Gradient Norm. While gradient information sits at the core of influence and fine-tuning, the norm of the gradient itself is a poor approximation for utility. To demonstrate, consider

$$f_{\text{gn}}(\mathcal{D}_{\text{a}}) = \left\| \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\text{a}}} \nabla_\theta L\left(\boldsymbol{x}, y; \hat{\theta}\right) \right\|_2, \tag{4.1}$$

which measures how surprising $\mathcal{D}_{\text{a}}$ is to a model trained on $\mathcal{D}_{\text{tr}}$. Indeed, the gradient norm can be large when the prior distribution of classes in $\mathcal{D}_{\text{a}}$ differs from that of $\mathcal{D}_{\text{tr}}$, as desired when $\mathcal{D}_{\text{tr}}$ is class-imbalanced. Yet, the gradient norm can also be large when $\mathcal{D}_{\text{a}}$ contains unfamiliar but useless or even harmful data. Under a simple formulation of label noise, f_{gn} inverts the desired ranking, as shown in Figure 4.4. More information is needed to reveal relative utility.

Model Fine-tuning. To approximate data utility arbitrarily well, fine-tune a model on $\mathcal{D}_{\text{a}} \cup \mathcal{D}_{\text{tr}}$:

$$f_{\text{ft}}(\mathcal{D}_{\text{a}}) = \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\text{te}}} L(\boldsymbol{x}, y; \hat{\theta}) - L(\boldsymbol{x}, y; \hat{\theta}_{\text{ft}}), \tag{4.2}$$

where $\hat{\theta}_{\text{ft}}$ are the parameters after a fixed number of SGD updates on $\mathcal{D}_{\text{a}} \cup \mathcal{D}_{\text{tr}}$ seeded with $\hat{\theta}$. Despite its success in optimization in plain text, fine-tuning via SGD in private has novel challenges: it can be rather computationally intensive when implemented via MPC, because the number of sequential passes can be large. Moreover, since inspecting the training loss is not possible without leaking additional information, successful SGD optimization in secure MPC requires careful pre-tuning of hyperparameters.

Forward Influence Functions. Here, we use influence functions to approximate the change on the resulting loss from including the dataset $\mathcal{D}_{\text{a}}$. As put forth in Section 2.4.1, the influence function $\mathcal{I}(\boldsymbol{x}, y)$ associates a training sample with the change in the model parameters under an

infinitesimal up-weighting of that sample in the risk [72; 165].

Equation Forward Influence Function defines the influence-based appraisal function to be the sum of each training sample's influence:

$$f_{\text{if}}(\mathcal{D}_{\text{a}}) = -\frac{1}{|\mathcal{D}_{\text{a}}| \cdot |\mathcal{D}_{\text{te}}|} \sum_{(\boldsymbol{x}_{\text{te}}, y_{\text{te}}) \in \mathcal{D}_{\text{te}}} \sum_{(\boldsymbol{x}, y) \in \mathcal{D}_{\text{a}}} \nabla_\theta L(\boldsymbol{x}_{\text{te}}, y_{\text{te}}; \hat{\theta})^\top \boldsymbol{H}_{\hat{\theta}}^{-1} \nabla_\theta L(\boldsymbol{x}, y; \hat{\theta}). \tag{4.3}$$

For interested readers, Appendix C includes a set of key derivations.

4.2.1 Forward Influence in Multi-party Computation.

Computing $f_{\text{if}}(\mathcal{D}_{\text{a}})$ requires computing and inverting empirical Hessian, usually a costly operation. For $\theta \in \mathbb{R}^d$ this requires $O(d^3)$ operations. Prior works suggest employing approximations for Hessian inverse vector product [5; 165; 124]. However, to evaluate multiple candidate datasets for a given model, the inverse Hessian need only be computed once. In this way, the cost of computing and inverting $\boldsymbol{H}_{\hat{\theta}}$ can be amortized over many evaluations. Furthermore, this can be done in the clear by the model owner as it requires only $\hat{\theta}$ and $\mathcal{D}_{\text{tr}}$. Computing the gradient of the loss on the test set can also be done in the clear, as no new data is required. Hence, the term $\frac{1}{n_{\text{te}}} \sum_{(\boldsymbol{x}, y) \in \mathcal{D}_{\text{te}}} \nabla_\theta L(\boldsymbol{x}, y; \hat{\theta})^\top \boldsymbol{H}_{\hat{\theta}}^{-1}$ may be precomputed by the model owner once in the clear and then encrypted. This leaves only a private computation of the loss gradient for each $\mathcal{D}_{\text{a}}$ followed by an inner-product in $\mathbb{R}^d$. Because private computation tends to dominate the overall runtime, this yields considerable computational savings compared to private fine-tuning, as shown in Figure 4.5.

4.3 Experimental Results

We aim to answer the following research questions:

1. In terms of runtime and usability in secure MPC, how do forward influence functions compare with fine-tuning and alternative data appraisal methods?

2. How robust is influence function-based appraisal under data corruption and class imbalance?

3. How effective is a greedy dataset selection strategy in which a model owner sequentially chooses to acquire the dataset with the highest influence function value?

We train and evaluate the model on classification problems using the MNIST [177] and CIFAR-10 [169] datasets: on MNIST, we classify ten digits, and on CIFAR-10, we distinguish planes from cars. Additionally, we verify our findings using Wisconsin diagnostic dataset for breast cancer (WDBC) [87]. The examples consist of features computed from images of breast mass biopsies along with the target benign or malignant cancer diagnosis. The classification problem is solvable when 70% of the data is used for training [4].

The ground truth ranking comes from re-training in the clear, both fine-tuning and influence appraisals are studied using secure MPC implementations in CrypTen. As mentioned in Section 3.5, the numerical precision is set to 24 bits, and influence multiplication is scaled by 1e5 for stability. (Details in Section 4.6.)

In each of the experiments, we fix the initial training model, including $\mathcal{D}_{\text{tr}}$, $\mathcal{D}_{\text{te}}$, and $\hat{\theta}$, and only intervene on the quality of the datasets to construct $\{\mathcal{D}_{\text{a}}^{(p)}\}$, such that their ranking is salient. Prior to evaluating the appraisal functions on $\mathcal{D}_{\text{a}}$, we train the model on the seed training set $\mathcal{D}_{\text{tr}}$ until convergence to obtain $\hat{\theta}$.

We study three types of alterations on the datasets to simulate variations that are likely to arise in an open data market: (1) *label noise* in which the correct label of an example is changed

with some non-zero probability; (2) *class imbalance* in which the marginal frequency of the labels varies between candidate datasets; and (3) *missing features* in which the candidate datasets vary in terms of which features they provide.

To simulate needing additional data, the initial model is trained on 1-10% of the available dataset, further seeded with a 9:1 imbalance in binary classifications. The models are L2-regularized logistic regressors. To best approximate the optimal classifier, the baseline weights are obtained via L-BFGS [194]. For ranking statistics, Spearman's Correlation Coefficient is used, denoted as ρ [84].

	Finetuning			**Influence**
learning rate	1×	4×	16×	1 epoch
0.001	0.61	0.58	0.72	
0.01	0.95	1.0	1.0	0.96
10	0.96	0.59	0.88	

Table 4.1: Correlation ρ of appraised values and data utility with varying amounts of label noise. Finetuning runtimes are limited to 1×, 4× and 16× of influence runtime, each benchmarked on the *best* performances under three learning rates: 0.001, 0.1, and 10. Hyperparameter tuning runtime for fine-tuning is excluded.

4.3.1 In MPC, Forward Influence Functions Are More Usable Than Finetuning

Influence Requires No Additional Hyperparameters. Although finetuning can approximate the test loss arbitrarily well, discovering the hyperparameters that achieve low error requires careful pre-tuning in the clear. In MNIST, small batch sizes and large epochs, as recommended for finetuning, often have high computational runtime (Table 4.1). Figure 4.3 summarizes the effect of finetuning hyperparameters on the correlation of appraisal with utility (top) and runtime (bottom). The hyperparameter selections in green result in few passes, but picking them will lead to sensitive rank correlation, thus requiring extensive tuning or scheduling. Meanwhile, safe

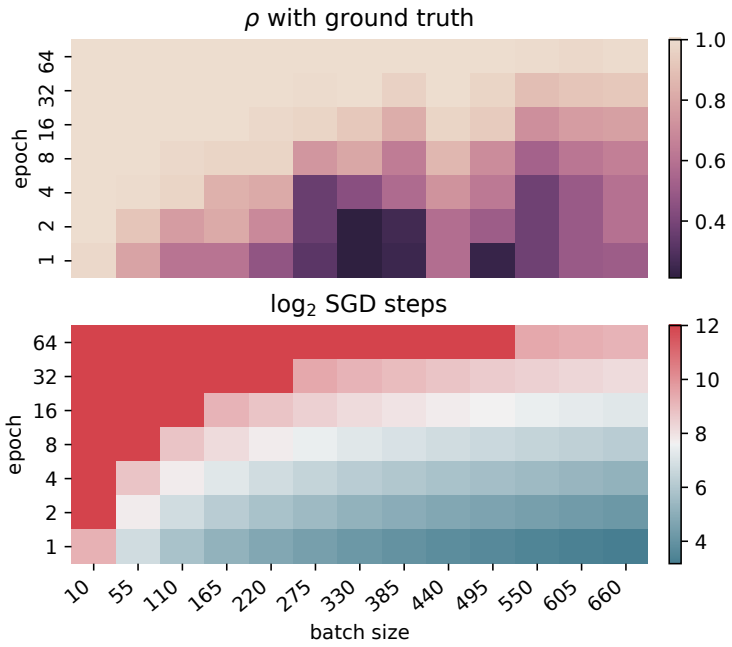


Figure 4.3: Finetuning-based Appraisal Trades Off Efficiency and Accuracy Due to Hyperparameters. Correlation of appraisal with utility (top; purple is lower) and runtime (bottom; blue is faster) for finetuning hyperparameters batch size configuration (x-axis) and epochs (y-axis; logarithmic). This reproduces Figure 3.2.

hyperparameter settings tend to result in relatively large number of SGD passes. Both strategies incur significant computational cost. Lastly, even using the best batch size configurations, finetuning on noisy MNIST can fail to be competitive (Table 4.1).

Influence Has Minimal Private Runtime. For any dataset, private influence performs a full-batch gradient step and a vector multiplication of dimension d for $\theta \in \mathbb{R}^d$. Thus, computing influence in private is comparable to that of finetuning with one SGD pass – the minimal without subsampling. In secure MPC, private runtimes tend to dominate as the number of evaluation grows. For a reasonable hyperparameter setting of 16 steps of full-batch gradient descent for finetuning, Figure 4.5 presents the total runtime of each appraisal function, separating the encrypted from the plaintext runtimes under plane-to-car setup. Due to influence functions' efficient setup with no additional hyperparameter, it trades a high one-time overhead for a convenient imple-

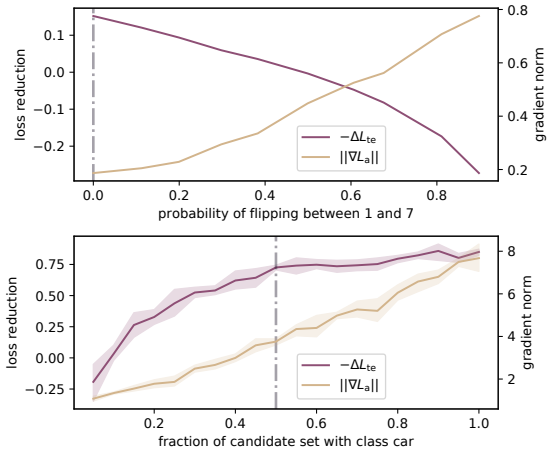


Figure 4.4: Gradient Norm is an unreliable value proxy. Gradient norm appraisal and test loss reduction as a function of MNIST label noise (top, $\rho = -1$) and CIFAR-10 plane- to-car class balance (bottom, $\rho = 1$).

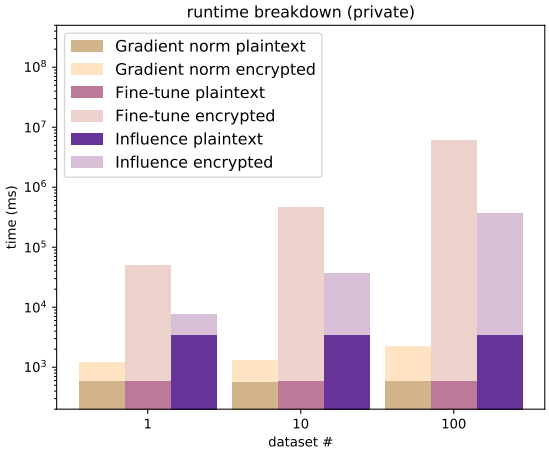


Figure 4.5: Log Scale Runtimes Spent On Plaintext And Encrypted Computations For All Three Appraisal Methods. Influence-based appraisal achieves low amortized timing as its plaintext overhead (purple) is fixed.

mentation that scales well in private.

4.3.2 Forward Influence Recovers Ranking Under Noise and Imbalance

We evaluate the efficacy of our data appraisals in two scenarios: (1) in which the utility of the data varies because of label noise in that data and (2) in which the utility varies because the data distribution does not match the distribution that the model owner is interested in.

Gradient Norm Is Insufficient. Despite their conceptual similarity, label noise and class imbalance are distinct corruptions that challenge naive, gradient-based methods. When gradient norm is used for appraisal, both datasets of poor balance (undesirable) and datasets of low noise (desirable) would obtain similarly low numerical values. As shown in Figure 4.4, the gradient norm appraisal value (y-axis; note that the units vary per method) is monotonic over datasets under our two sets of experiments: label noise (x-axis) on MNIST (top) and data imbalance on CIFAR-10. The gradient norm curve (purple) aligns with risk reduction (yellow) under data im-

balance, but crosses it under labels noise. Using only the norm of the gradient, though fast to compute, is an unreliable predictor for data value.

Label Noise. In our first scenario, we vary the utility of the dataset $\mathcal{D}_a$ by introducing label noise. In particular, we use 1% of the MNIST training data as $\mathcal{D}_{tr}$. The remaining training data is split into 10 candidate datasets $\mathcal{D}_a^{(p)}$ with $p = 1, \ldots, 10$. For each of the candidate sets $\mathcal{D}_a^{(p)}$, we randomly flip labels 1 and 7 with probability $p/10$. We evaluate models on $\mathcal{D}_{te}$. Table 4.1 presents the correlation ρ of the label-noise probabilities with the appraisal value, including under three finetuning learning rates: 0.001, 0.1, and 10. The correlations are high for the model finetuning and influence function methods, suggesting that influence-based appraisal captures data utility.

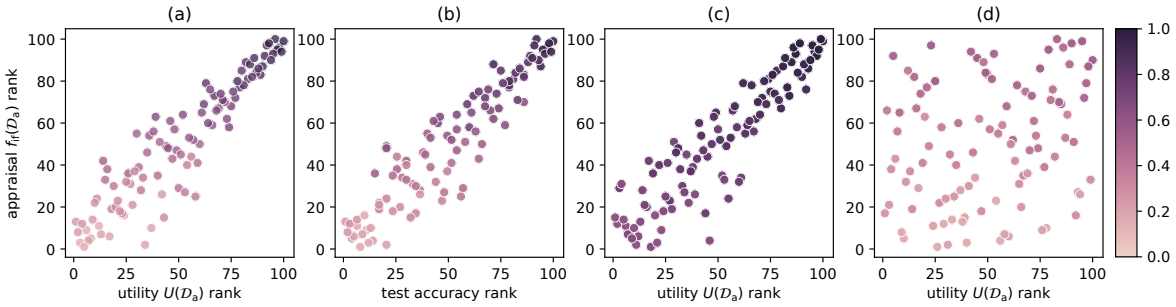


Figure 4.6: Left a-b: Rank of influence-based appraisal $f_{if}(\mathcal{D}_a)$ (y-axis) as a function of the rank of the utility (a; $\rho = 0.923$) and the test accuracy (b; $\rho = -0.927$) on CIFAR-10's plane-to-car dataset. **Right c-d:** Rank of $f_{if}(\mathcal{D}_a)$ as a function of the rank of the utility on CIFAR-10 dataset for which the rate of cars is in the range $[0, 0.45]$ (c; $\rho = 0.908$) and $[0.55, 1.0]$ (d; $\rho = 0.247$). Each dot is a sampled dataset, colored according to the ratio of the under-sampled class in $\mathcal{D}_a$.

Distribution Mismatch. In our second scenario, we focus on influence-based appraisal and study its efficacy under distribution mismatch. We simulate the mismatch between: (1) $\mathcal{D}_{tr}$ and $\mathcal{D}_{te}$ and (2) the candidate datasets $\mathcal{D}_a^{(p)}$ by varying the prior over classes. To do so, we construct a training set from CIFAR-10 with a 10:1 ratio of plane-to-car and a balanced test set with a 1:1 ratio of plane-to-car. We then construct 20 candidate datasets $\mathcal{D}_a^{(p)}$ of which exactly $(5 \cdot p)\%$ are planes and the remainder are cars, with $p = 1, \ldots, 20$. The candidate datasets are of size $|\mathcal{D}_a^{(p)}| = 440$.

We repeat this process five times, sampling the datasets randomly each time.

Figure 4.6 shows scatter plots of: (a) the rank of the influence-based appraisal value, $f_{\text{if}}(\mathcal{D}_{\text{a}})$, of each of the 5×20 candidate datasets and (b) the rank of the utility or test accuracy of those datasets (see caption for details). The experimental results show that the influence-based appraisal value correlates well with gains in utility. Specifically, $f_{\text{if}}(\mathcal{D}_{\text{a}})$ allows the model owner to select a candidate dataset that closely resembles their desired distributions in most situations. However, zooming in on different ranges of class ratios (c-d), influence-based appraisal value $f_{\text{if}}(\mathcal{D}_{\text{a}})$ is becomes less informative when the class ratio deviates far from that of both the training and testing datasets.

4.3.3 Applying Influence Appraisal On Corrupted Cancer Patient Data

Real world applications often use passively gathered data of varying quality. Though the samples are not created for machine learning, they may be included for training. We simulate such a scenario with breast cancer detection from hospital screenings. We corrupt datasets by adding noise or removing features, and then apply influence-based appraisal to rank the datasets.

The first set of experiments concerns the rank correlation of datasets between forward influence functions and the ground truth losses, which trains $\mathcal{D}_{\text{tr}} \cup \mathcal{D}_{\text{a}}^{(p)}$ for all p to convergence. The same data is then corrupted. To simulate missing features, 10 columns are dropped (out of 30). Furthermore, we simulate label noise in candidate set, $\mathcal{D}_{\text{a}}^{(p)}$, benign (positive) and malignant (negative) diagnoses are flipped under a binomial distribution of parameter $p/500$ and $p/200$ for $p = 1, \ldots, 100$.

Influence-based appraisal is able to inform the model owner the relative value in very noisy datasets. Figure 4.7 shows scatterplots of 100 datasets' evaluation (a) when all columns are retained. (b) when 10 feature columns are dropped, and (c-d) when labels are flipped with probability $p/500$ and $p/200$ for $\mathcal{D}_{\text{a}}^{(p)}$. Table 4.2 shows rank correlation consistently above 80%. When all columns are preserved, the trained model can be used to identify helpful datasets. When 10

Corruption	Rank Correlation
None	0.880 ±0.081
Noise (Up to 1/5)	0.863 ±0.064
Noise (Up to 1/2)	0.844 ±0.106
Missing Features	0.810 ±0.213

Table 4.2: Influence Appraisal Correlation $\rho \pm \sigma$ With Data Utility on WDBC Over 10 Runs.

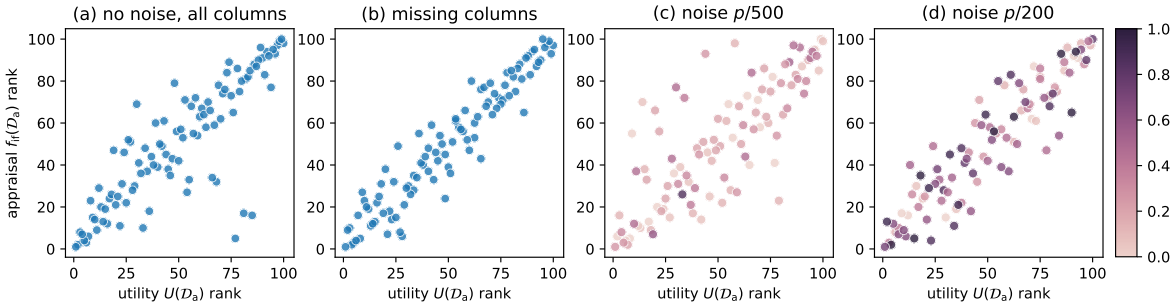


Figure 4.7: Influence-based appraisal makes reasonable appraisal ranking compared to ground truth utility (WDBC). The rank of appraised values (y-axis) as a function of the rank data utility (x-axis) with varying data corruptions. The noiseless datasets (a-b) are benchmarked under 30 features and 20 features. The noisy datasets (c-d) are colored with noise level as a fraction of each dataset's label flips between "Benign" and "Malignant", and retain all features.

columns are missing, performance varies greatly; as the training set has less information about the problem, its second order landscape at convergence is less informative. Nevertheless, influence functions show robust ranking in the presence of missing features and noise.

In the second set of experiments we examine the loss dynamic from repeatedly using influence functions for data selection. Raj et al. [248] proposes a strategy of data inclusion by selecting samples of the highest influence among a set of available candidates. In contrast to their setup where the candidates are existing training sets, samples in an open data markets that we simulate are often farther from the data distribution. Given a base model and 100 candidate datasets, two strategies are used in 15 iterations to select a dataset at a time, without replacement. Figure 4.8 shows the loss in varying noise, with 10 columns randomly dropped at each run. Despite the diverse seed models, the loss curves for greedy strategy based on influence (purple) often drops sooner than that of a random approach to selecting data. As more noise is injected to the candidate

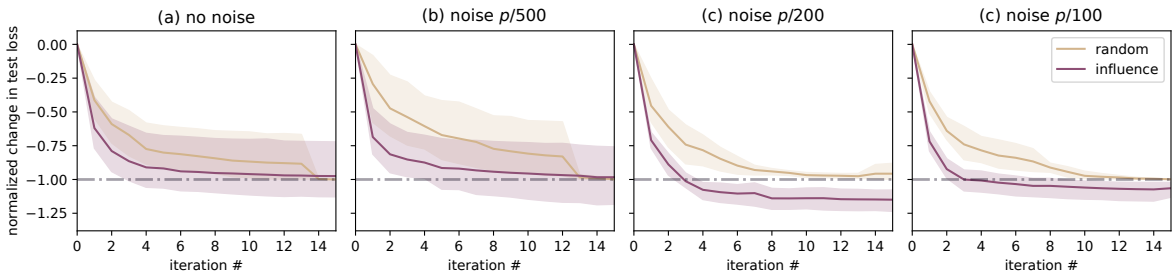


Figure 4.8: Influence excels random strategy at high noise levels, when applied sequentially select datasets without replacement for breast cancer diagnostics data. The change in test loss (y-axis) as a function of repeated rounds of data inclusion under varying noise levels. **Random**: choose a random dataset at each round. **Influence**: choose the dataset with the highest influence-based appraisal. For each graph, test loss change is normalized by the maxmimum test reduction in the control group. Averages and variances are taken over 5 runs.

labels (c-d), influence consistently outperforms random selection, which is a strong baseline.

4.4 Related Work

We present two most similar lines of work from the time of [320], and expand on a recent development.

Data Pricing in Federated Markets. Efficient private appraisals can especially aid federated learning settings where 1. privacy requirements are salient, and 2. the compute resources available *pre-transaction* are limited. In differential privacy and federated learning literature, Li et al. [185]; Song et al. [276] and Wang et al. [307, 310] privately assess sets of data *after* the model is trained on them, while our solution does not require private training. Nevertheless, our approach to craft appraisal functions to suit privacy constraints complements recent works on acquisition strategies and Nash equilibrium in emerging data markets [18; 241]. Also under game-theoretic lens is computing Shapley values [265] to assess training data for machine learning [111; 147; 18; 20]. A primary motivation for using Shapley values is to enable equitable concurrent data assessment, while we focus on a limited scale where datasets are acquired one at a time. Indeed, in sequential acquisition, a dataset acquired at a later stage of research may see its appraisal value lowered, if other datasets had reduced test loss. As a result, our appraisal incentivizes small-scale data owners to join the appraisal as early as possible.

Influence Functions. Measuring the effect of the data under leave-one-out training is known as Cook's distance in linear regression or the influence curve in regression residuals [71; 72]. Many contemporary works employ influence functions to explain existing training examples aposteri, applied to interpretability [166; 124], cross-validation [113], poisoning attacks [145], and training data removal [123; 168]. As a result, influence functions are usually 1. defined with respect to the trained model, 2. used to approximate parameter change under data removal. In contrast, we 1. use forward influence functions where the model has not seen the new data,

concurrent to Raj et al. [248]'s subsampling experiment for model selection and 2. applied to privately recover relative ranking. Incidentally, with the addition of MPC, we demonstrate a use case predicted by Giordano et al. [113], where influence is chosen for our application where the Hessian inverse computation is a worthwhile tradeoff.

Data Valuation Most recently, the field Data Attribution and Data Valuation have emerged as their own fields of research, including but not limited to [144; 238; 172; 149], and an ICML benchmark [309], an LLM benchmark for influence functions [150]. While they are still mainly motivated by explaining model behavior (including test performance) back to training data, it is often implicitly for acquiring new data. In the age of large scale models, such interpretation usually feeds *forward* to knowing what kind of data the model owner ought to commission or purchase. Indeed, [120] evenly explicitly constructs an alternative influence function for this purpose while a 2024 ICML tutorial calls for data attribution with a "predictive" (i.e., predicting *forward*) emphasis [202]. These developments validate our motivation, and echos our use of influence functions for evaluating future data. However, in those works, the privacy issue is yet left open. This offers fertile ground for future work to incorporate privacy-preserving methods for appraising data.

4.5 Limitations

Our procedure shows an appealing trade-off between computation and privacy, with several avenues for future work.

Reconstruction Over Many Queries. While setting a threshold on appraisal scores can limit the information leak to 1 bit, in theory, a strong adversary may reconstruct the data (or model) by observing appraisal values.

Discrimination of Arbitrary Data. Though f_{if} can discriminate quality differences despite corruptions, the choice of the model and $\mathcal{D}_{\text{a}}$ dictates a fundamental limit, e.g., in Figure 4.6, when the class imbalance of $\mathcal{D}_{\text{tr}}$ and $\mathcal{D}_{\text{a}}$ cancels out. Moreover, f_{if} is defined on a limited class of models: twice differentiable and convex in θ. Whether convexity can be relaxed in influence functions is its own active area of research [26; 27]. However, our approach to value data in private benefits from research on data attribution and data valuation, setting the stage for more efficient, accurate, and robust data appraisals without data sharing.

4.6 Engineering Contributions

Previously, Section 3.5 set forth the high level overview of engineering considerations of secure computation. We now summarize the technical challenges in completing our work, including when implementing our baselines.

Implementing Private Training (Ground Truth Baseline) While CrypTen supports private training out-of-the-box, it lacks native support for optimizers like L-BFGS, thus SGD was used in our experiments. Moreover, while the private training-based algorithm (finetuning-based appraisal in Section 4.2) mirrors that of its non-secure counterpart, the numerical approximations at each iteration can cause divergence in model weights. This poses our primary engineering challenge: **achieving an appropriate private model fit in private**. Because hyperparameters are assumed pre-set, comparing private training entails choosing parameters that would achieve a reasonably good model performance. To do so, we swept SGD hyperparameters in plaintext for the specific task, and transferred the best fit as initialization to the private fine-tuning algorithm.[2]

Implementing Private Influence-based Appraisal (Our Method) As illustrated in Section 3.5, influence-based appraisal functions result from a **careful *co-design* of optimization and MPC engineering**. The main algorithmic benefit lies in avoiding the costly computation, numerical instability, and hyperparameter challenges associated with private training.

As mentioned in Section 3.5, influence-based appraisal requires fewer sequential steps. However, my first implementation was not ideal – the matrix multiplication within influence-based appraisal is high-dimensional, where each resulting floating point may be numerically unstable, causing the sum of the influence functions to be erroneous. Despite CrypTen's support for private training, automatic rescaling is not natively supported. Notably, **overflows and under-**

[2]Later work in Chapter 5 used SGD with moment and learning schedule, and swept the hyperparameters in both plaintext and in encrypted setting.

flows are frequent for previously untested workloads such as ours, yet **difficult to debug** due to them occurring "silently" in the encrypted setting, as Knott et al. [164] notes as a CrypTen design limitation.

Three general approaches were considered 1. Higher precision (more bits) for private computation, which was much slower, 2. Normalizing and rescaling inputs that are likely to cause overflows and underflows, which involved more guesswork 3. Jittering the hessian matrix before inverting, which was hard to justify theoretically, as it would slightly modify the plaintext algorithm. Eventually, I arrived at **setting 24 bits for CrypTensor initialization** while **normalizing the inverse hessian sum by 1E-6 before encrypting**. Through extensive testing, I found these to be sufficient for ensuring a stable output that is correct when compared with plaintext results.

Limitations of Using CrypTen The arithmetic secure sharing implementation can "wrap around" silently (when the numbers resulting from a computation over/underflows), which are hard to debug. The default precision (16-bit) is "not enough bits" for modern machine learning — a limitation not sufficiently foreseen at the time of CrypTen development. Working around these design choices can thus be challenging, as they are built into the framework.

Our work uncovers significant room for improvements in the future for MPC for ML frameworks, especially **improved co-design between MPC with high-precision numerical workloads**, including eliminating the guesswork involved for making an algorithm private.

4.7 Summary

We crafted efficient algorithms leveraging secure MPC techniques and forward influence functions to avoid private training. Our implementation is an appealing high quality appraisal with scalable, efficient computation, with few hyperparameters, while being robust to label noise, class imbalance, and missing data. Our empirical results suggest that appraising data using influence function leads to accurate valuations in many scenarios, while requiring limited computation and no hyper-parameter optimization, notably outperforming private fine-tuning.

Lastly, we demonstrated the practical effectiveness of influence-based appraisal in a breast cancer detection task with greedy, sequential data acquisition, which outperforms random selection under data corruptions. Future work focuses on broadening the applications of private data appraisal, including extending private data appraisal to more complex non-linear models with efficient inverse Hessian product approximations.

5 | Dataset-to-Dataset Evaluation Before Data Sharing

Privacy concerns and competitive interests impede data access for machine learning, due to the inability to *privately* assess external data's utility. This dynamic disadvantages smaller organizations that lack resources to aggressively pursue data-sharing agreements. Particularly in data-limited scenarios, not all external data is beneficial [69; 267], further affecting early partnerships. Given these inherent risks, collaborations particularly suffer in heavily-regulated domains: metrics that aim to assess external data given a source e.g., approximating their KL-divergence [267], require accessing samples from both entities pre-collaboration, hence violating privacy. This conundrum disempowers legitimate data-sharing, leading to a false "privacy-utility trade-off". To resolve privacy and uncertainty tensions simultaneously, we introduce `SecureKL`, the first secure protocol for dataset-to-dataset evaluations with no additional leakage outside its score output, to be applied preceding data sharing. `SecureKL` evaluates a source dataset against candidates, performing dataset divergence metrics internally with private computations, all without assuming downstream models. On real-world data, `SecureKL` achieves high consistency ($> 90\%$ correlation with non-secure version's resulting partnership ranking) and successfully identifies beneficial data collaborations in highly-heterogeneous domains (ICU mortality prediction across hospitals and income prediction across states). Our results highlight that secure computation maximizes data utilization, outperforming privacy-agnostic utility assessments that leak information.

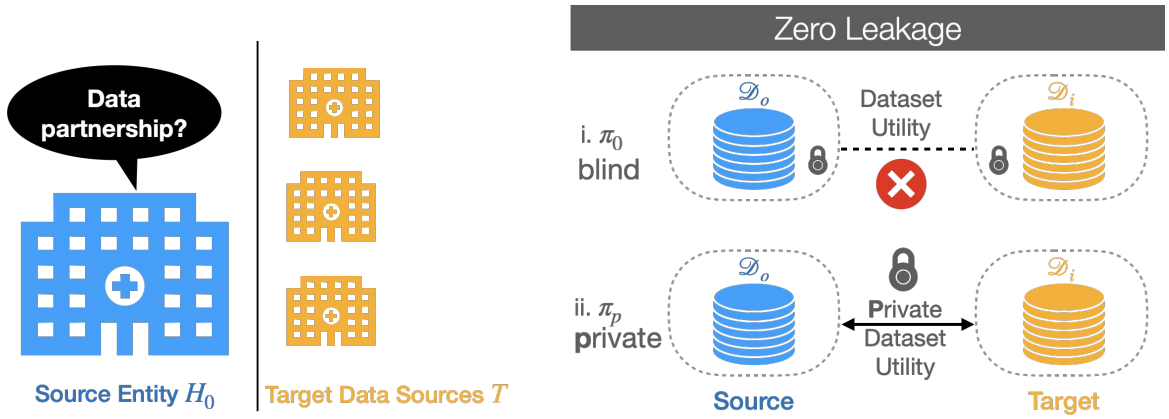


Figure 5.1: Privacy can dis-incentivize data collaborations. Without seeing external data, an organization has two strategies: **i. blind default** π_0: randomly selecting partnerships causes hesitation and hinders partnerships. **ii. Our method** π_p: securely assessing datasets by leveraging MPC *before* commitment.

5.1 Introduction

As brought up in Section 2.2, empirical works have predetermined data as a key driver to performance gains [134; 155; 46]. Yet, accessing and combining datasets is persistently challenging. As datasets have evolved from small to larger [280], more diverse [331], and more compute-optimal [135], the field of machine learning continues to seek more data [203; 46; 236; 253; 266] and better ways to *combine* it [268; 224; 243; 187; 206].

Strategically combining data from different sources promises enhanced models, but disempowers smaller organizations. While diverse, high quality data often improves performance, robustness, and fairness [212], access to such data significantly varies across entities and domains [287; 201]. As domain-specific data becomes increasingly valuable [10; 126; 179], data-owning entities are more reluctant to share it for free, opting instead to sell it in emerging data markets [3; 140; 191]. This dynamic disproportionately handicaps smaller organizations who lack both the resources to purchase data and the leverage to negotiate favorable sharing agreements.

In particular, organizations may hesitate to commit to a potential partnership when unsure

about the benefits. As Figure 5.1 illustrates, this “commitment issue" is not solely a privacy issue; it’s the inability to privately assess an external dataset’s utility *before* partnerships. This evaluation is, however, nontrivial without examining the data. Surprisingly, in-domain data does not necessarily result in more performant models. Due to the precarious nature of domain shifts, machine learning models may behave unpredictably to untested additional training data, especially when the source dataset is small (e.g., a single hospital’s data) – a phenomenon known as “the dataset combination problem" [167; 308; 284; 40; 209; 267].

Ideally, *all* underlying data should be considered to reduce uncertainty in costly data collaborations. Yet, **datasets owned by separate entities cannot be directly and fully accessed**, significantly limiting the practicality of non-private dataset measures [267; 144].

Our work directly addresses this crux by recognizing both privacy and competitive incentives. First, *before* committing to acquiring unseen data, we enable organizations to privately gauge the relative utility of candidate datasets. Second, we provide strong privacy guarantees required of entities operating under stringent regulations – e.g. healthcare providers – to navigate data acquisition. This allows organizations to tackle data availability issues by prioritizing the most relevant potential partnerships, without seeing their data.

Developers are often uncertain about the most effective model before more data becomes available. This renders a secure data appraisal stage introduced in Chapter 4 not applicable, because it requires model parameters to be known ahead [320]. In this more opportunistic setting where the model is not yet developed, we ask:

> *Can we ascertain the differential utility of prospective datasets, without knowing the final model?*

This chapter introduces `SecureKL`, **the first private dataset measure with minimal leakage**, by leveraging dataset-to-dataset divergences. Our key insight is that private, model-agnostic divergence computations via secure multiparty computation (MPC, previously introduced in Sec-

tion 3.4) are more data-efficient than sharing samples, while being just as accurate as sharing full samples without privacy. SecureKL privately computes dataset divergence measures, presenting a compelling guarantee: **for both parties, privacy is fully protected while data utilization is maximized**.

Contributions A novel secure dataset-to-dataset evaluation protocol SecureKL (SKL) that reduces uncertainty in data utility under limited data and budget, producing privacy-preserving measures while using the maximum available samples. SKL achieves a $> 90\%$ correlation with privacy-violating counterparts across two real-world heterogeneous domains. Empirically, on ICU mortality prediction, SKL reliably chooses beneficial hospital(s) to partner with, outperforming data-leaking alternatives, including using demographic summaries or sharing data subsets.[1]

Impact We provide a practical solution for organizations seeking to leverage collective data resources while maintaining privacy and competitive advantages. Our major advantage lies in reliability, especially when small organizations cannot afford to invest in detrimental partnerships. These results demonstrate the potential for wider data collaboration to advance machine learning applications in high-stakes domains while promoting more equitable access to data. Our code is available, and can be readily deployed to demonstrate potential data value preceding collaborations.

[1]Our code: https://github.com/kere-nel/secure-data-eval

5.2 Context and Contributions

The dataset combination problem faces multiple challenges that privacy-preserving dataset evaluation can mitigate. This section contextualizes the appropriate background.

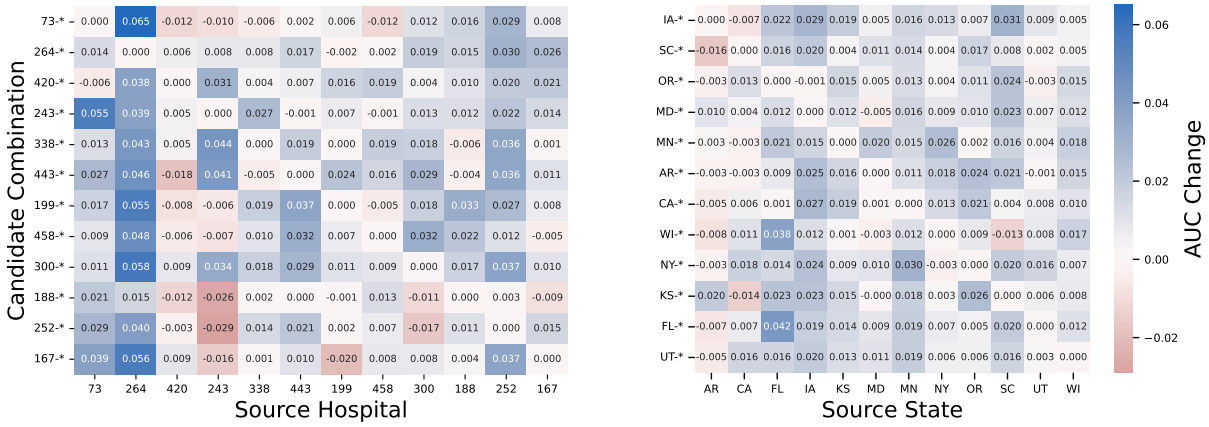


Figure 5.2: Real-world data collaborations are inherently uncertain, necessitating pre-partnership selection (a *reproduction* of Figure 2.5). AUC change for a source entity (x-axis), after incorporating external data from a different source (y-axis), across hospitals (left) and states (right).
Left: in mortality prediction using eICU data [244], 10 out of 12 hospitals may see their mortality prediction model degrade for *some* potential hospital partners.
Right: in income prediction using Folktables [83], combining with random state leads to worse prediction in 7 out of 13 states. (Red is bad).

Dataset Combination Problem In high-stakes domains, additional datasets may *degrade* the model. In healthcare scenarios, both Compton et al. [69] and Shen et al. [267] showed that blindly acquiring new datasets can degrade model performance, especially initially. We confer to findings in [267; 69] and re-run on pairs of hospitals' datatsets and pairs of states' survey data to produce Figure 5.2. Indeed, in both highly heterogeneous domains, opportunistically acquiring *unseen* datasets may even be harmful. This non-monotonic behavior in the Dataset Utility on real-world data means that these risks of "bad data deals" may thwart collaborations, highlighting the need to evaluate data *before* embarking on a full-fledged collaboration. In these scenarios, `SecureKL` validates partnerships robustly and safely before any data is exchanged.

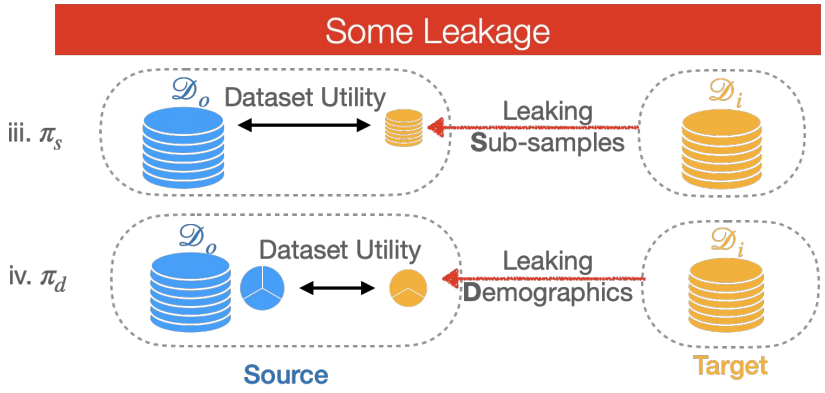


Figure 5.3: Abstraction of non-private evaluation strategies, following the privacy-preserving methods in Figure 5.1 **iii**. **subset sampling** π_s: a subset of the target's data is shared. **iv**. **demographic-based summaries** π_d: the target entity discloses distributions by protected attributes, i.e., age, gender, or race. In these scenarios, some data is leaked by the target partner (yellow) to assuage the source entity (blue)'s uncertainties about the target data's value, trading off sample utility for limited privacy.

Data-leaking Measures Are Inefficient Before an agreement, informed negotiations become impossible when entities do not expose data. Data owners frequently resort to tiny samples or summary statistics (e.g., race, gender, age) for making decisions, illustrated in abstract in Figure 5.3 [54]. Yet, in data-limited settings, model performance is highly sensitive to new input. These heuristics are fickle, as sparse traits or limited samples often fail to capture the entire dataset's nuance and complexity, especially in heterogeneous domains, creating a perceived privacy-utility trade-off. `SecureKL` avoids this by using secure computation to enable measurements over entire datasets, which we show to be more reliable in deciding on hospital collaborations.

Our Method: Private While Flexible This data acquisition scenario needs to ensure privacy: the source data and the data to acquire are kept separate by default, like in Chapter 4. However, Chapter 4 uses the Forward Influence Function, computing on the model parameters θ, which may not be determined. Moreover, empirical Hessian of the Forward Influence Function is not

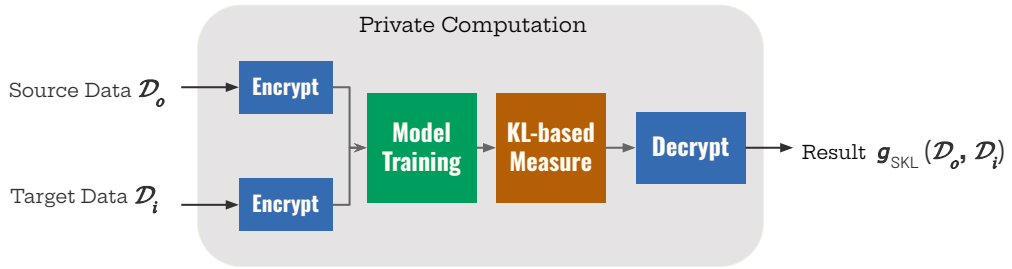


Figure 5.4: Our Method SecureKL$(\mathcal{D}_o, \mathcal{D}_i)$. Each side encrypts their data. Without assuming the *downstream* model, a dataset comparison model is trained on their joint data, typically a membership inference model using logistic regression (Section 5.3), which enables computing KL-based measures in private. Then their divergence is assessed, and the final result is revealed after both parties participate in decryption.

always readily available, and large models do not "fit" well in MPC. [2] A far more flexible approach – **Dataset-to-dataset evaluation** – represents a novel paradigm that gauges the dataset's utility by comparing just the dataset at the source with the dataset at target, securely, without making assumptions about the final model (Figure 5.4).

Summary of Contributions Our method presents an appealing trade-off of privacy and utility: **SecureKL** (Figure 5.4) preserves privacy at the input for both parties, maintains the accuracy of the nonsecure dataset measure for the source entity, while maximizing utility of the samples. Specifically, we present

1. **The first MPC implementation of KL-XY**, the first divergence-based dataset measure without leaking data outside the score output. Computing KL-XY score [267] in this secure multiparty protocol maintains strong privacy guarantees. By utilizing the complete underlying datasets, it achieves $>$ 90% correlation with plaintext computations in downstream dataset ranking.

2. **A methodology to directly compare *private* vs. *non-private* methods** to evaluate potential partnerships. We categorize existing approaches by their privacy leakage risks

[2] As remarked in Section 3.4, current state-of-the-art MPC transformer generates one token in five minutes on an 8B-parameter model [85].

(Section 5.4) and provides a systematic way to assess the trade-offs between data utility and privacy preservation.

3. **An empirically-tested measure for beneficial data-sharing collaborations** in low-data, high-stakes settings. Our method successfully identifies beneficial data partnerships for intensive care unit (ICU) mortality prediction; it reliably improves classifier performance for the source hospital, over multiple runs for all hospitals (Section 5.6)[3]. Detailed benefit analysis advocates for using private dataset combination SKL, as it avoids privacy leakage without utility downside.

Position On Communicating Secure Computation Security researchers tend to misunderstand the lack of adoption of privacy-preserving methods. Surely, when the apparent benefit is the *addition* of privacy, people must not care sufficiently about privacy (Sec 2.6.3) [275; 218; 22]. Yet, the resistance does not stem from a lack of care; instead, it stems from a breakdown of trust. Precisely because data leakage is by far not the *source* of the problem, privacy cannot be the only benefit. While this thesis derives its privacy arguments from identifying the key players of machine learning (Section 2.3), as Balsa et al. [22] aptly puts, cryptographic assumptions are mere "shorthand" for presupposed adherence.

Therefore, communicating to decision makers – the real humans behind each entity that holds data, trains models, or performs oversight – of the benefit from using a technology entails more than presenting a "privacy solution". To that end, this chapter proposes a direct comparison between private, data-efficient but approximate methods and plaintext sharing, demonstrating that **secure computation's value extends beyond privacy**.

[3]Our measure is not symmetrical, meaning that we assume the source hospital purchases and benefits from the hospital providing extra data, but not necessarily the other way around.

5.3 Setup

Consider a binary prediction task for ICU patient mortality based on electronic medical records. A source hospital H_o has historical patient data $\mathcal{D}_o$ containing static past patient characteristics, prior medical records, and ICU outcomes. Other hospitals $\{H_i\}$ each has their patient data: $\{\mathcal{D}_i \mid i \in [1..N]\}$.

For this binary prediction task, hospitals typically optimize for performance metrics, for example the area under the receiver-operating characteristic curve (AUC). Using only their data, H_o can train a model $\mathcal{M}$ with parameters θ to achieve:

$$\text{AUC}_o = \max_{f(\theta)} \text{AUC}(\mathcal{M}, \mathcal{D}_o) \qquad \text{(Baseline AUC)}$$

where f is their chosen algorithm with parameter θ.

When H_o has exhausted their own internal data, they may benefit from incorporating additional target data sources $T \subset [1..N]$. By combining datasets, i.e., $\mathcal{D}_T = \{\mathcal{D}_i \mid i \in T\} \cup \mathcal{D}_o$, H_o can potentially achieve better results:

$$\text{AUC}_T = \max_{f(\theta)} \text{AUC}(\mathcal{M}, \mathcal{D}_T). \qquad \text{(Combined AUC)}$$

We define the potential improvement from data addition as $\delta_T = \delta_{(o,T)} = \text{AUC}_T - \text{AUC}_o$. To add a single additional data source by setting $T = \{i\}$, the improvement is $\delta_i = \delta_{(o,i)} = \text{AUC}_i - \text{AUC}_o$. This leads to our central question:

> ***Without seeing target data, how does a hospital ascertain potential data sources to combine with?***

Formally, given $n \leq N$, we seek a strategy π that selects n target datasets $T = \pi(\mathcal{D}_o, n)$ to

maximize model utility:

$$\pi^*(\mathcal{D}_o, n) = \underset{T \subset \binom{[1\ldots N]}{n}}{\arg\max}\ \text{AUC}_T \qquad \text{(Ideal Combination)}$$

Practical Considerations. Computing every subset $T \subset \binom{[1\ldots N]}{n}$'s associated δ_T is exponential in n. To make this problem tractable, we make two key assumptions. First, we apply strategies greedily, selecting top-ranked target datasets. With the ultimate objective of improving the source hospital's prediction task, we fix H_o; to compare the trade-offs between strategies in Section 5.4, we apply each π greedily to select top-n institution(s) for H_o without replacement. Second, in data-constrained settings where combining datasets only result in small AUC improvements, we may simply aim for more likely occurrences of positive improvement: $P_{H_o \sim \mathbf{H}}(\delta_T > 0)$.

Kullback–Leibler Divergence. Our approach uses Kullback-Leibler (KL)-divergence-based methods to gauge data utility, building on prior work [267]. KL divergence [170], also called *information gain* [245], describes a measure of how much a model probability distribution Q is different from a true probability distribution P:

$$\text{KL}(P||Q) = \int_{x \in \mathcal{X}} \log \frac{P(\mathrm{d}x)}{Q(\mathrm{d}x)} P(\mathrm{d}x) \qquad \text{(KL-Divergence)}$$

Because computing KL-divergence on datasets $\mathcal{D}_o$ and $\mathcal{D}_i$ is non-trivial due to the high dimensionality of the data, Shen et al. [267] proposes groups of scores to approximately differentiate hospital dataset divergences. Specifically, fixing $\mathcal{D}_o$ and $\mathcal{D}_i$, score $\text{KL}_{\mathcal{XY}}$ first trains a logistic regression model on $\mathcal{D}_o \cup \mathcal{D}_i$ – where the labels are folded into the covariates — with the goal of inferring dataset membership. Then, the resulting model's probability score function $\text{Score}(\cdot) : \mathcal{X}, \mathcal{Y} \to [0, 1]$ is averaged over a dataset in H_o, obtaining

$$\text{KL}_{\mathcal{XY}} = \mathbb{E}_{(x,y) \sim \mathcal{D}_o}(\text{Score}(x, y)). \qquad (KL_{XY}\ \text{Score})$$

Dataset	Data Addition π	Pearson ρ	p-value
eICU	SecureKL$_{\mathcal{XY}}$ (**Our Secure Method**)	-0.182	**3.65e-02**
	SecureKL$_{\mathcal{X}}$ (Our Secure Baseline Method)	-0.162	6.27e-02
	KL$_{\mathcal{XY}}$	-0.184	**3.47e-02**
	KL$_{\mathcal{X}}$	-0.162	7.13e-02
	Gender	0.097	2.65e-01
	Race	0.018	8.29e-01
	Age	0.053	5.33e-01
Folktables	SecureKL$_{\mathcal{XY}}$ (**Our Secure Method**)	-0.181	**3.13e-10**
	SecureKL$_{\mathcal{X}}$ (Our Secure Baseline Method)	0.016	5.62e-01
	KL$_{\mathcal{XY}}$	-0.198	**4.58e-12**
	KL$_{\mathcal{X}}$	-0.107	**2.00e-04**
	Gender	0.0002	9.94e-01
	Race	0.026	3.75e-01
	Age	0.012	6.71e-01

Table 5.1: Dataset divergence is predictive of utility for downstream models.
Pearson correlation (ρ) and p-values between each data addition strategy (π) and the source model's performance after adding selected dataset (AUC drop, δ_i), reported separately for two real-world datasets (eICU [244] and Folktables [83]).
The non-secure strategies, including KL$_{\mathcal{XY}}$, are detailed in Section 5.4.4 and in Section 5.4.
Statistically significant p-values ($p < 0.05$) are bolded.

Note: As discussed in Section 2.4.1, ideally the source hospital can compute their KL$_{\mathcal{XY}}$ scores with respect to every $i \in [1...N]$, and chooses, but it is not private. Additionally, dataset divergence heuristics may hinge on model fit (over/underfitting can render the score less effective). Yet it reflects the insight that parametrized distribution estimations are more efficient on finite, *unknown* data [226; 299].

Privacy Model As brought forth in Section 2.5.2, we operate under a semi-honest privacy model. Unlike non-private dataset measures such as [267] and [144], which would require data sharing to participate, this threat model incentivizes collaboration by ensuring data privacy from each other.

Dataset Divergence and Downstream Utility Building on the finding of Miller et al. [212] where a model's in-distribution performance is related to its out-of-distribution performance (across all model choices), we infer that dataset divergence does predict downstream model's performance after combining the data and confirm it in Table 5.1 using metrics introduced in Shen et al. [267]. Intuitively, in-distribution quality is paramount in low-data settings, where dataset divergence can capture greater complexity and nuance than accessing a few traits. In contrast, data-rich domains like language modeling more frequently benefit from diverse, specialized data sources. Yet, privacy is unresolved: divergence measures entail accessing both entities' data [267; 144], posing significant risks for heavily-regulated entities who are liable for any data exposure [53; 109].

Secure Multiparty Computation (MPC) Our divergence computation is cryptographically secured at the input using Secure Multiparty Computation (MPC) [324; 264]. As Chapter 4.4 mentioned, MPC lets two or more parties to compute a function over private inputs, revealing only the final output [115]. For both parties, privacy is guaranteed at the input, while the samples are maximally utilized.

Despite their non-trivial engineering, MPC programs enjoy strong security guarantees and relative ease of deployment (Section 3). Even small organizations can deploy MPC without any specialized hardware. Thus, the algorithms developed and shared in `SecureKL` readily enable dataset-to-dataset evaluations before sharing data.

Strategy	Description	Sub-strategies	Leakage
π_0	Blind (baseline)	n/a	zero
π_p	Private (SecureKL)	n/a	minimal
π_d	Demographics	sex, gender, race distance	moderate
π_s	Sub-Sampling	1%, 10%, 100% shared	high

Table 5.2: Partnership selection strategies, differentiated by leakage (privacy cost). A strategy π returns a chosen set of targets T from all candidates. Section 5.4.1 describes the strategies in detail.

5.4 Data Acquisition Strategies, in Detail

5.4.1 Baseline Strategies Categorized

As depicted by the strategies in Figures 5.1 and 5.3, the potential cost of data acquisition is linked to leakage risks. Therefore, we define three categories of risks and formalize their corresponding strategies, which are summarized in Table 5.2.

A, high leakage, sharing raw data. $\pi_s(n, k)$ supposes each hospital to share a dataset of size k; a default setting of 1% is commonplace practice in some contracts, as a pre-requisite to being considered [54]. Though leakage can be controlled through k, the data is inherently sensitive.

B, moderate leakage, sharing summary statistics. $\pi_d(n)$ uses demographic metadata to guide data selection. This is implemented through ratio distance between source and target distributions, which may be considered aggregates therefore potentially not sensitive, such as when the underlying aggregation function ϕ is differentially private.

C, minimal leakage, sharing no *additional* information besides what is assumed public. There are two methods: a. **Blind selection baseline**: $\pi_0(n)$ randomly selects n disjoint data sources, until data purchasing budget runs out. Prior works suggest that when $n = 1$, randomly selecting a source in hospital ICU may be risky and inefficient. b. **Our method** $\pi_p(n)$ selects data sources based on privacy-preserving measure for data combination, specifically Private KL-XY.

5.4.2 Trivially Private Baseline: Blind Selection

Blind selection refers to the process when no information is provided. This random strategy, $\pi_0(n)$, may evade selection biases and help gather diverse data. Yet, prior work [267] suggests that $\pi_0(1)$ – randomly selecting one source – for ICU is risky and inefficient for mortality prediction.

5.4.3 Sharing Summary Statistics

A relaxation to sharing no sensitive data is to share "metadata". While demographic traits are often *causal* and available, their exact cause in relation to the task is not a priori established (without a highly effective model), therefore their success in distributional-matching is not guaranteed to be strong. Additionally, in practice, the most effective model that results from data combination may or may not be causally-sound. Nevertheless, we posit alternative strategy $\pi_d(n)$ to find the demographically close candidates to guide data selection: Let $\phi : \mathcal{D} \to \mathbb{R}^m$ be an m-dimensional summary statistic of a demographic feature i.e. the racial distribution of patients. Then, we use the distributional distance between $\mathcal{D}_o$ and $\mathcal{D}_i$, characterized by their L_2-distance through ϕ:

$$\pi_d(n=1) = \underset{i \in [1..N]}{\arg\min}\ L_2(\phi(\mathcal{D}_o)||\phi(\mathcal{D}_i)). \qquad \text{(Demographic-based Strategy)}$$

Note: Though alternative features and norms can be considered, our goal is not to feature-engineer (or to compare with the results from the best feature-engineering). Instead, we hope to use this strategy to approximate what hospitals resort to in practice today – using demographic summaries **when the dataset cannot be seen** (a baseline from [267]).

5.4.4 KL-based Methods, Without Privacy

In sub-sampling strategy $\pi_s(n, k)$, each of the candidate entities will leak a set of raw data. π_s is implemented with KL-based measures similar to Shen et al. [267]'s proposal. Recall from

Section 2.4.1, a binary predictor is fit on this combined dataset, predicting $\mathcal{I}$ from $(\mathcal{X}, \mathcal{Y})$ using logistic regression. The model's output $p(x, y)$, also called the probability score, is Score(x, y).

A score of 0.5 or less means the datasets are not distinguishable, making the data potentially useful for being "in-distribution"; besides our own findings, we point out other prior empirical insights: Shen et al. [267] established the insight that in data-limited domains of heterogeneous data sources, domain shifts of the covariates are useful for predicting whether the additional data helps the original task, similar to [212] We note again that even though this model is trained on both parties' data, the final algorithm that the hospital uses to train on combined data is not restricted.

Then, the resulting model's probability score function Score$(\cdot) : \mathcal{X}, \mathcal{Y} \to [0, 1]$ is averaged over a dataset in H_o, obtaining

$$\text{KL}_{\mathcal{X}\mathcal{Y}} = \mathbb{E}_{(x,y)\sim\mathcal{D}_o}(\text{Score}(x, y)). \tag{KL-XY}$$

Let the score function g_{KL} be the approximation of $\text{KL}(\mathcal{D}_o||\mathcal{D}_i)$ (specifically $\text{KL}_{\mathcal{X}\mathcal{Y}}$). The strategy selects the most likely hospital with the closest distance under the measure:

$$\pi_s(n = 1, k = K) = \arg\min_{i\in[1..N]} g_{\text{KL}}(\mathcal{D}_o, \mathcal{D}_i). \tag{KL-based Strategy, *in plaintext*}$$

As mentioned, we reran the strategies on our setups that confirmed [267]'s results that $\text{KL}_{\mathcal{X}\mathcal{Y}}$ is predictive of downstream change in AUC in hospital setting (Table 5.1).

When only a subset is available, this function is adjusted by swapping $\mathcal{D}_i$ for $\mathcal{D}_i' \subseteq \mathcal{D}_i$ where $|\mathcal{D}_i'| = k$. We denote the full dataset size as $K = |\mathcal{D}_i|$. Note while k controls privacy leakage, we acknowledge that the raw data is inherently sensitive (even if only 1%, and anonymized, transferring raw data has security risks), therefore associate π_s with a high privacy risk in Table 5.2.

5.4.5 SecureKL: Private KL-based Method

Using MPC (Section 3), we extend on $\mathrm{KL}_{\mathcal{XY}}$ to require no information sharing (besides what was already assumed public), which is considered minimal in privacy risk in Table 5.2. As illustrated in Figure 5.4, the logistic regression as well as the scoring need to be implemented in private. This infrastructure may be hosted on both sides' machines [164].

Denote the private encoding of x as $[x]$.

$$\mathrm{SecureKL}_{\mathcal{XY}} = \mathbb{E}_{(x,y)\sim\mathcal{D}_o}(\mathrm{Score}([x,y])). \qquad \text{(Secure KLXY)}$$

Let the score function g_{SKL} be the secure approximation of $\mathrm{KL}(\mathcal{D}_o||\mathcal{D}_i)$. The strategy selects the most likely hospital with the closest distance under the measure:

$$\pi_p(n=1) = \underset{i\in[1..N]}{\arg\min}\ g_{\mathrm{SKL}}(\mathcal{D}_o, \mathcal{D}_i). \qquad \text{(SecureKL Strategy, } \textit{encrypted}\text{)}$$

Notably, any KL-based measure g_{SKL} can be adapted to our setup, while we primarily use $\mathrm{SecureKL}_{\mathcal{XY}}$. Additionally, even though our implementation measures distance of data between one source and one target party, the setup readily extends to accommodating multiple parties. Section 5.7.2 discusses potential deployment challenges.

Assumptions Consider high stakes domains where disparate data may have additive benefits to the existing data. We reiterate assumptions laid out in Section 2.5.2 specific to our work.

1. **Existing knowledge** is not private. The hospitals are aware of each other having such data to begin with. The hospitals may know of the available underlying dataset size and format, which is assumed to be uniform across the hospitals in the setup to simulate unit-cost. Hospitals frequently know of each other's resources, such as the scale of their ICU units.

2. **Uniformity** of $|\mathcal{D}_i|$. Though each hospital gets to price their data and set their own budget in practice, pricing is not our primary consideration in this work. For generality in investigating the dataset combination problem, we make two simplifications: 1. every unknown record is "priced" the same (fixed unit-cost), 2. each candidate hospital would supply the same sized dataset (fixed $|\mathcal{D}_i|$ across all potential partnerships). This uniformity assumption allows us to use the number of additional data sources n as the main "budget proxy" across different strategies for a source hospital.

3. **Legal risks** of sharing *any* data are omnipresent in high stakes domains. The risks with sharing sensitive data in data-leaking strategies, which we coin as π_d (demographic distance) and π_s (small sample), are not made explicit, but assumed to be "moderate" and "high" respectively. This abstraction side-steps legal discussion, which would go beyond the scope of our paper.

4. **No malice** is assumed on any of the parties involved, as each hospital wants to authentically sell their data and set up a potential collaboration. This assumption becomes stronger when the number of parties grows or when the setup changes to potentially more competitive industries with less trust.

5.5 Experimental Setup

5.5.1 Experimental Questions

1. **Consistency**: Does MPC degrade the original measure's effectiveness? As MPC implementations introduce approximations, we validate SKL's implementation. The correlation of **private scores** vs. **plaintext scores** (with full data access) needs to be examined. Additionally, consistency of choosing encryption in the hospital domains means examining plaintext and encrypted versions' correlation with downstream ground truth ranking across hospitals.

2. **Positivity**: Does our method pick entities that reliably improve performance? If source dataset D_o can only add data from n more sources, does our measure lead to eventual AUC improvements? Specifically, in multi-dataset combination, we examine whether using SKL can improve the source hospital's downstream outcome. When selecting a single (or a few) additional data source, how many hospitals improve with our method? Additionally, we compare our method with alternative, privacy-leaking strategies.

3. **Error analysis**: When our privacy-preserving method is not the most beneficial strategy against alternatives, such as when our method selects a harmful data partnership for a source hospital, what may be the reason? As we know, small and uncertain improvements for downstream tasks underscore the inherent difficulty of evaluating data utility without seeing the full data. We are especially interested in analyzing (a) hospitals with low $\text{SecureKL}_{\mathcal{XY}}$ and $\text{KL}_{\mathcal{XY}}$ correlations, and (b) hospitals lagging AUC improvements using the random strategy π_0 or limited-sample strategy π_s.

5.5.2 MPC IMPLEMENTATION.

SecureKL$_{\mathcal{XY}}$ includes training logistic regression model in private, following MPC engineering practices in Section 3. We implement measures based on dataset divergence by building a custom logistic regression model over encrypted data leveraging CrypTen. The model parameters and input are encoded as 16-bit MPCTensors, ensuring that all computations, including forward passes, sigmoid activations, and gradient descent updates, are performed in private.

ADDITIONAL BASELINE DETAILS We additionally run our experiments on plaintext methods used in Shen et al. [267], including the KL$_{\mathcal{X}}$ measure, which is similar to KL$_{\mathcal{XY}}$ without using each data souce's labels:

$$\text{KL}_{\mathcal{X}} = \mathbb{E}_{(x)\sim\mathcal{D}_o}(\text{Score}(x)). \tag{KL-X}$$

To compared against KL$_{\mathcal{X}}$ for baseline, we additionally implemented its encrypted version, SecureKL$_{\mathcal{X}}$, as a g_{SKL} candidate.

$$\text{SecureKL}_{\mathcal{X}} = \mathbb{E}_{(x)\sim\mathcal{D}_o}(\text{Score}([x])). \tag{Secure KL-X}$$

OPTIMIZERS Because L-BFGS – the optimizer prior work [267]used in plaintext-only with Scikit-learn [240] – is not available as an encrypted version, our MPC experiments are facilitated with SGD optimizer. A fair comparison between the scores obtained through plaintext and encrypted settings necessitates re-implementing plaintext scores, Score(X) and Score(X, Y), using logistic regression with SGD in PyTorch [239]. The hyperparameter tuning for SGD in private and plain text are performed independently, as they do not transfer. Section 5.8 will discuss hyperparameter-tuning in detail.

5.5.3 Data and Model Setup

eICU dataset. The downstream task is 24-hour mortality prediction from ICU data using the eICU Collaborative Research Dataset [244]. This dataset contain $> 200,000$ real-world admission records from 208 hospitals across the United States.

A note on data filtering. Because medical research is inherently complex, ensuring reproducibility on statistical methods can be significantly challenging. Water et al. [312] proposes that the research community follow a shared set of tasks with fixed preprocessing pipelines that are clinically informed – essentially a machine learning training and evaluation protocol on eICU – in order to facilitate method verification on the same benchmark. Therefore, our work follows their data cleaning criteria and the evaluation protocol. Additionally, we use the hospital exclusion criteria in [267] to obtain top 12 hospitals, where the most patient visits are collected (each with > 2000 patients).

Downstream model and baselines for eICU. Recall that each strategy uses the same K number of records per hospital – in our experiment, $K = 3000$. For π_s which leaks a subset k of all samples, a default $k = \lfloor 1/100 |\mathcal{D}_i| \rfloor$ randomly drawn samples are shared. In ICU data, we run experiments on $\{0.1\%, 1\%, 10\%, 100\%\}$.

Following holistic benchmarking tools in [312], our strategy comparisons take 1500 samples and the downstream model performance – AUC_o, AUC_T – uses 400 samples (unless otherwise noted). Specifically, the AUC change, δ_i or δ_T, comes from 1. combining 1500 random samples from each selected dataset and 2. combining it with 1500 samples from $\mathcal{D}_o$, and 3. subtracting the baseline model's AUC[4].

Folktables dataset Though we primarily focus on hospital domain, we additionally validate using Folktables [83], predicting across 35 states an individual's annual income exceeds \$50,000. The details of our processing, which diverges from that of eICU, is included in Appendix D.6.

[4]The samples are fixed across all experiments, the sample numbers are chosen to match the setup in [267].

5.6 Results and Analysis

Our goal is to investigate whether using multiparty implementation sacrifice the original $\mathrm{KL}_{\mathcal{XY}}$'s efficacy (**consistency**; Section 5.6.1), whether our method reliably picks hospitals that improve performance (**positivity**; Section 5.6.2), and where our method may fail (**error analysis and limitations**; Section 5.6.3).

5.6.1 Consistency Between Plaintext and Encrypted Computations

Because our encrypted computation is the first implementation of dataset divergence in MPC, we ought to show that $\mathrm{SecureKL}_{\mathcal{XY}}$ and $\mathrm{SecureKL}_{\mathcal{X}}$ lead to highly comparable behaviors as $\mathrm{KL}_{\mathcal{XY}}$ and $\mathrm{KL}_{\mathcal{X}}$.

Spearman's Rank Correlation Coefficient for Underlying Scores For each source hospital H_o, use all full samples for $\mathcal{D}_i$. Between $\mathrm{KL}_{\mathcal{XY}}$ and $\mathrm{SecureKL}_{\mathcal{XY}}$ on $\mathcal{D}_o$ and $\mathcal{D}_i$ for all remaining hospitals H_i, $\mathbb{E}_{H_o \sim \mathbf{H}}[\rho] = 0.908$ with a range of $[0.691, 1.0]$, obtaining $p < 0.02$ across all hospitals. Between $\mathrm{SecureKL}_{\mathcal{X}}$ and $\mathrm{KL}_{\mathcal{X}}$, $\mathbb{E}_{H_o \sim \mathbf{H}}[\rho] = 0.9303$ with a range of $[0.455, 0.991]$, with 11 of 12 hospitals achieving p-values below 0.05. After applying Hochberg false discovery rate correction [28], our p-values remain significant. This range may be an artifact of sweeping hyperparameters independently in plaintext and encrypted optimisations, because sharing the same SGD hyperparameters would result in a tighter range. For all 12 hospitals, see appended Appendix D.4 for details.

Impact of Adding Security on AUC Correlation We further examine the effect by *adding* encryption through its impact on the downstream AUC, using how AUC improvements are ranked. This rank is compared with how secure measures (i.e., $\mathrm{SecureKL}_{\mathcal{XY}}$) and plaintext measures (i.e. $\mathrm{KL}_{\mathcal{XY}}$) rank hospitals. This comparison investigates the extent of the shift in the full

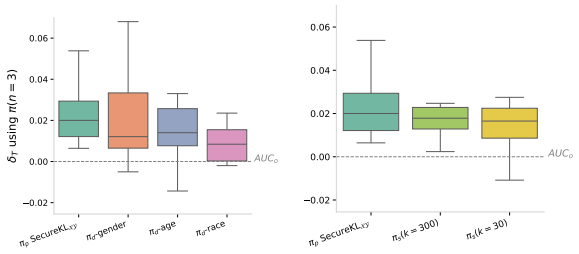


Figure 5.5: AUC change δ_T over all strategies in eICU prediction (higher is better). Our private dataset evaluation strategy π_p outperforms demographic-based strategy π_d **(left)**, and subsampling strategy π_s for $k = 300$ (10%) and $k = 30$ (1%) **(right)**, after combining source data with the top 3 candidates.

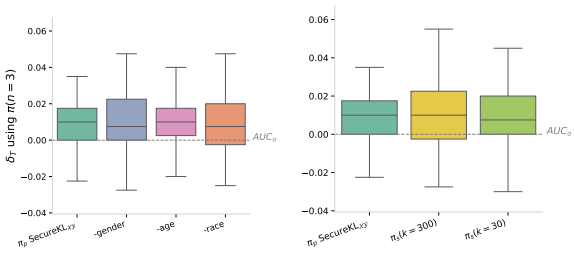


Figure 5.6: AUC change δ_T over all strategies in Folktables dataset prediction (higher is better). All strategies exhibit comparable distributions, after combining data from top 3 candidates. In a noisy domain, our method is stable: it neither excels nor penalizes against non-private strategies.

hospital ranking, when we switch from a plaintext setup to encrypted. For $H_o \sim \mathbf{H}$, we measure δ_i that results from adding $\mathcal{D}_i$ to $\mathcal{D}_o$ for all i. This correlates all target hospitals $\{H_i\}$ with their ground truths $\{\delta_i\}$ in the case of picking a single target hospital. We find the linear coefficient for encrypted SecureKL$_{\mathcal{XY}}$ to be -0.182 and plaintext KL$_{\mathcal{XY}}$ to be -0.184 (99% matching). Both SecureKL$_{\mathcal{X}}$ and KL$_{\mathcal{X}}$ have a linear coefficient of -0.164 with δ_i. For all strategies' correlations with ground truth at $n = 1$, see Appendix D.2.

5.6.2 Positivity in Realistic Setup

Overall Positivity. We apply `SecureKL` in a pragmatic multi-source data combination problem, where each strategy acquires n datasets for $n \in \{1, 2, 3\}$. For $n = 1$, we find that π_p improves AUC in 10 out of the 12 hospitals. When $n = 2$ and $n = 3$, we find that using π_p consistently improves AUC for all hospitals. Overall, 34 out of the 36 dataset combinations we evaluate on have an AUC improvement $\delta_T > 0$, suggesting that π_p is a reasonable strategy for selecting hospital dataset combinations with a high expected return $\mathbb{E}[P_{H_o \sim \mathbf{H}}(\delta_T > 0)]$ for the source hospital from using our strategy.

Comparing With Alternative Strategies Other strategies – π_0, π_d, and π_s – can also arrive at "positive" datasets. Comparing private method to other strategies at $n = 3$, i.e, $\pi_p(n = 3)$, we describe our results in Figure 5.5

1. π_p (our method based on SecureKL$_{\mathcal{XY}}$) has a median δ_T of 0.020, and a standard deviation of 0.015. Our results indicate that for 50% of the hospitals, π_p gives a $\delta_T >= .02$. Compared to other strategies, π_p has the highest median, the lowest standard deviation, and it is one of two strategies that improves performance for all hospitals.

2. Demographic-based strategies underperform compared to π_p on average. However, we observe that π_d-gender can be highly effective for a subset of hospitals, as it achieves the highest 75th percentile (Q3) of 0.033 among all strategies. This indicates that for 25% of hospitals, $\delta_T \geq 0.033$. Despite this, π_d-gender has a lower median value of 0.012 compared to π_p, exhibits a high standard deviation (0.022), and degrades the performance for certain hospitals. Similarly, π_d-age has a median of 0.014, and π_d-race has a median of 0.008, both lower than π_p's median.

3. Plaintext small-sample strategies, π_s, outperform all demographic-based methods but slightly underperform relative to π_p. For instance, $\pi_s(k = 300)$ has a median δ_T of 0.0178, and although it achieves $\delta_T > 0$ across all hospitals, it performs worse on average compared to π_p and exhibits a higher standard deviation (0.017). $\pi_s(k = 30)$ has a median δ_T of 0.0165. Compared to other strategies, it has the largest standard deviation (0.024), and it degrades the performance for some hospitals.

In summary, our method π_p achieves the highest AUC improvement on average with the lowest standard deviation, demonstrating **more consistent improvements** for all hospitals. While the average improvement of π_p is small, demographic-based and plaintext small-sample strategies exhibit greater variability, with some strategies improving performance for specific subsets of hospitals but underperforming or degrading results in others.

n	π_d-gender	π_d-age	π_d-race	$\pi_s(k=300)$	$\pi_s(k=30)$	π_p SecureKL$_{\mathcal{XY}}$
			eICU			
1	**0.020** ± 0.023	0.012 ± 0.016	0.014 ± 0.015	0.012 ± 0.014	0.010 ± 0.017	0.011 ± 0.018
2	0.016 ± 0.016	0.017 ± 0.017	0.015 ± 0.013	0.024 ± 0.020	0.017 ± 0.019	**0.027** ± 0.022
3	0.020 ± 0.023	0.016 ± 0.019	0.011 ± 0.021	0.021 ± 0.017	0.021 ± 0.024	**0.024** ± 0.015
			Folktables			
1	0.009 ± 0.011	0.011 ± 0.008	**0.011** ± 0.011	**0.011** ± 0.008	0.008 ± 0.009	0.008 ± 0.009
2	0.014 ± 0.011	**0.015** ± 0.011	0.014 ± 0.012	0.014 ± 0.011	0.012 ± 0.011	0.011 ± 0.011
3	0.015 ± 0.011	0.016 ± 0.010	**0.017** ± 0.012	**0.017** ± 0.012	0.015 ± 0.010	0.015 ± 0.010

Table 5.3: AUC improvements in mean and standard deviation, across all source regions for each strategy π, for eICU and Folktables setups. n denotes the number of candidate datasets added to the source dataset. The small gains and high variance from adding selected datasets highlight the precarious nature of assessing data value in the real world. **Bold** indicates highest AUC improvement per n. *Note*: Only π_p SecureKL$_{\mathcal{XY}}$ is private.

5.6.3 SecureKL Error Analysis

Underlying Data Limitations In high-stakes domains, data partnerships are expensive, but potentially detrimental – this forms a challenging landscape for evaluating methods on real-world data. Indeed, as shown in Table 5.3, the AUC gain is small across all strategies, and the variance is high. This suggests that 3 hospitals' data is likely still too small for the general task to the robust explains limited AUC gains, highlighting the need to maximize samples for informative decisions. The key distinction, however, is that privacy-leaking methods (demographic, small sample) and blind baseline risk performance declines in many hospitals while SKL consistently improves downstream tasks **more reliably** than alternatives, across all hospitals, over multiple runs.

Underlying Score Limitations Data addition algorithms underpin the effectiveness of our method. Even if $\mathcal{D}_o$ obtains access to all the plaintext data, there is no guarantee that π_p can correctly predict whether the data is useful. As seen in Figure 5.7, Hospital 243's utility when acquiring another data set is badly correlated with plaintext and encrypted KL-XY scores. This leads to its bad strategy for acquiring the top 3 hospitals, as seen in the middle pane of Figure 5.8.

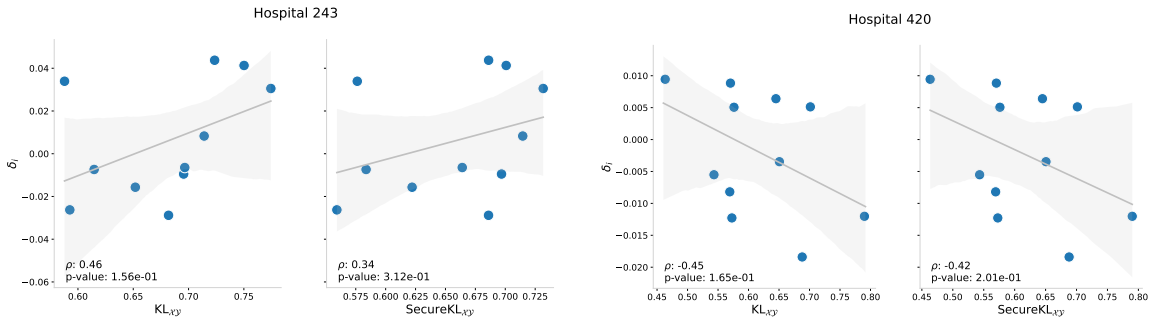


Figure 5.7: Correlation Between AUC Change δ_i and $\mathrm{KL}_{\mathcal{XY}}$ Impacts $\mathrm{SecureKL}_{\mathcal{XY}}$'s Efficacy. For Hospital 420, underlying $\mathrm{KL}_{\mathcal{XY}}$ identifies beneficial data ($\rho < 0$). For Hospital 243, $\mathrm{KL}_{\mathcal{XY}}$ fails to select effective data candidates ($\rho > 0$).

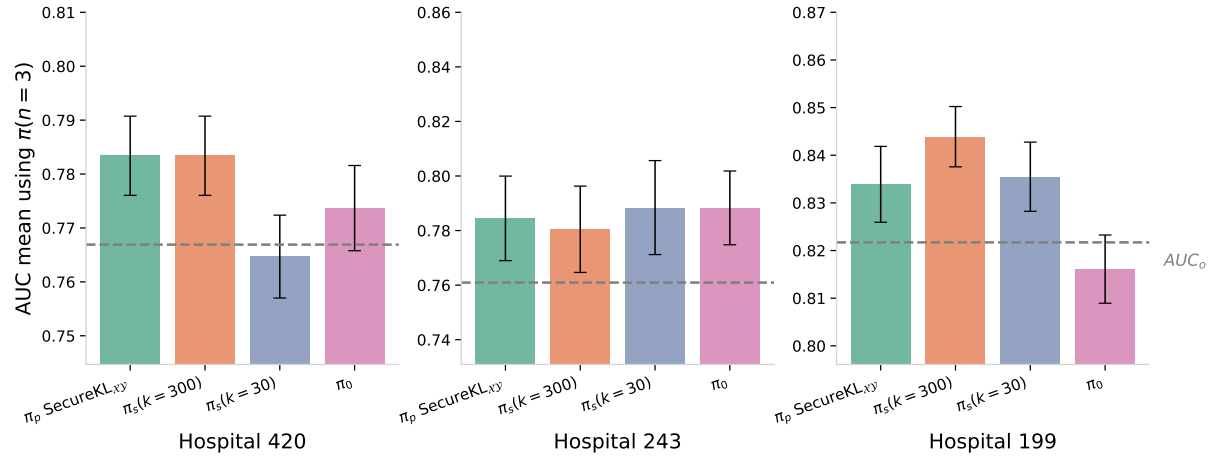


Figure 5.8: Left: $\mathrm{SecureKL}_{\mathcal{XY}}$ outperforms $\pi_s(k = 30)$ and π_0. **Middle**: All strategies perform similarly. **Right**: $\pi_s(k = 300)$ outperforms $\mathrm{SecureKL}_{\mathcal{XY}}$. (Bars represent standard deviations)

Interestingly, for this hospital, no other informational strategy excels, either, so choosing a random 3 may be preferred.

This behavior stems from the underlying measure, not from adding secure computation: in Figure 5.7 and Figure 5.8, the encrypted performance closely follows that of plaintext performance, for both good and bad downstream correlations.

Sometimes, Not All Underlying Data Is Needed Relatedly, when seeing a few samples can successfully identify useful candidate hospitals, π_s (which is on small samples) outperforms π_p (which is on full samples).

In the right panel of Figure 5.8, hospital 199, the smaller sample sizes achieve a score that better reflects ground truth as a data addition strategy. In that case, the hospital may not need the full sample to know which target hospitals to collaborate with.

This behavior is specific to the interaction of the data and the underlying score, and does not affect the general insight that adding private computation preserves privacy (and eases privacy-related risks that hinder data sharing). We further note that our method still clearly applies to encrypted computation on a smaller data set under data minimization.

5.7 Discussion

5.7.1 SecureKL Contribution

After establishing that π_p with SecureKL$_{\mathcal{XY}}$is a robust strategy in practical downstream performance, we hereby summarize the benefits of SecureKL and elaborate on their practical implications.

Matching Plaintext Performance in Downstream Tasks. Our major contribution is to match plaintext performance with no data sharing. Using MPC provides *input privacy*, meaning that if both hospitals only want to know the resulting score, the computation can be done without leaking original data. This strong guarantee can significantly ease the tension related to privacy and compliance in setting up a collaboration, leading to a practical "data appraisal stage" in data-limited high stakes domains.

Gain from Data Availability. In contrast to limited-sample approaches, a key advantage for our method π_p is that it takes advantage of all the underlying data – generally impossible with non-secure methods for private data in heavily regulated domains. The general intuition is that data is localized; therefore, once a good target hospital is identified, we would prefer to acquire all of the data. It may be tempting to assert that we prefer the highest k for data addition algorithms as well. In our experiments, while this is generally true, the smaller k sometimes outperform larger k in plaintext strategy π_s, which we investigate in Section 5.6.3 and in Figure 5.8. This occasionally non-monotonic behavior mirrors the challenge of data combination itself: even within one source dataset for the same estimator, more data is not necessarily better. This suggests the potential for a hospital-specific alternative to sharing a large amount of data for some source hospital, and points to future directions to using secure computation on a minimal-sized sample dataset for minimal performance overhead while remaining private.

Potential Improvements to SecureKL In the case where that output can be sensitive, i.e.,

when a source hospital queries a target hospital multiple times and accrues information through the score function, the *output* can also be made privacy-preserving through differentially private data releases, such as using randomized response [92].

5.7.2 Potential Challenges to Broader Adoption

Our code is readily usable by small organizations. While our approach generalizes to model-based measures (by substituting g_{SKL}) and scales to multiple parties, our work also uncovered deployment limitations.

1. **Operational**: engineering personnel limitations. While our implementation requires little cryptographic knowledge to deploy, it still needs technically-trained staff at each participating hospital to collaborate and maintain. This skill is similar to using pre-packaged software, cleaning data, and setting up network calls.

2. **Engineering Extensions**: Extending any MPC protocol is non-trivial, as security engineering is a specialized skill. While SecureKL applies broadly to other underlying scores in multi-party setups, *validating* a new MPC algorithm requires software engineering – prototyping, tuning, debugging – and numerical verification – akin to data analytics and research - likely requiring technical talents who can be especially costly for hospitals to retain in-house.

3. **Framework Limitation**: While CrypTen is designed to accommodate PyTorch, it is a research tool where not all plain text functionalities are implemented. As mentioned in Section 5.8, for example, writing optimizers – such as L-BFGS – and custom operators that are not readily available requires both machine learning and cryptography knowledge. Moreover, the protocol incurs additional computational overhead, especially if hyper-parameters become more complex to sweep[5]. This will likely improve with time, as new frameworks

[5]For our work, the performance metrics are provided in Appendix D.5 for reference

can address design shortcomings efficiently.

4. **Inherent to Secure Computation**: When the method requires significant hyper-parameter tuning, such as using SGD on small batch data with learning rate schedules, plaintext tuning may not transfer perfectly. As detailed in Appendix D.3, our hyperparameters for SGD differ in encrypted and plaintext settings. Thus, as encrypted computation *hides* loss curves and training details by default, development is expected to be complex. This is because both hospitals want to ensure model fit with secure evaluation, but may not want to expend the computational cost of private hyperparameter sweeping.

5.8 Engineering Contribution

Similar to Chapter 4, `SecureKL` is limited to using CrypTen with SGD for private training, which can lead to model fitting challenges discussed in Section 4.6 under "Implementing Private Training (Ground Truth Baseline)". Key to `SecureKL`'s contributions is reliably training in private for method validation. This entails fairly comparing `SecureKL` with non-secure $\mathrm{KL}_{\mathcal{XY}}$ with extensive experimentation.

Hyperparameter Tuning in Private became a significant bottleneck for our research. For both secure and non-secure performance comparisons, we tuned **separate sets of hyperparameters with and without encryption**[6]. Our baseline $\mathrm{KL}_{\mathcal{XY}}$ design had applied SGD with momentum and learning rate schedule for small batch data with early stopping, requiring six parameters to tune[7]. While straightforward in plaintext, hyperparameter tuning was complex in the encrypted setting. Because each encrypted run needed to complete, this process was clearly computationally-intensive. The sweep was often rerun when we had to debug issues in the encrypted setting (i.e., bad model fits).

Framework Improvements Acknowledging the need for both numerical stability and training fit when developing MPC-for-ML, we identify two improvement areas for future frameworks:

1. Support more private operations. As put forth in Section 3.5 and Section 4.6, CrypTen does not support all PyTorch functionalities. In particular, writing custom operators — including optimizers such as L-BFGS used in [267] —- is inaccessible without understanding both deep learning and secure engineering.

2. Make hidden curves more visible. As encrypted computation hides loss curves and training details by default in CrypTen, algorithmic MPC development is expected to be complex.

[6] This design decision is further validated in practice, as we found that plaintext tuning indeed does not transfer in the secure setting. For the hyperparameter comparison, see Table D.3 in the Appendix.

[7] In comparison, the baseline method in Chapter 4 used vanilla SGD with fixed learning rate and weight decay.

Model-based algorithms that involve joint training like ours especially suffer — when encrypted hyperparameter tuning is not part of the budget, requiring debugging over obscured data can hinder adoption. This indicates the potential for adjustable, privacy-preserving training statuses for MPC-for-ML frameworks to make visible.

5.9 Alternative Approaches

To achieve data combination has explored several approaches to 1. mitigate data sharing constraints while maintaining model performance, or 2. achieve secure data combination among multiple parties. We briefly discuss our survey and thinking along these two directions specifically for the medical domain. This section is adapted from [106].

Augmenting existing data with synthetic data in medical domains Synthetic data generation has emerged as a promising approach to expand training datasets while preserving privacy. Generative adversarial networks (GANs) have shown success in generating realistic cancer incidence data [117], medical imaging data [288], and electronic health records [23]. These methods can preserve statistical properties of the original data while providing differential privacy guarantees. Transforming data into a similar form that desensitizes certain attributes can be desirable [86; 137; 227; 118; 282]. Yet, to still preserve the utility of the dataset transformed for analytics or learning tasks is challenging by itself [152]. Additionally, outside the scope of sensitive data that is transformed, little privacy guarantee is available, leading to re-identification risks [219; 152].

In addition, evaluation of synthetic medical data reveals challenges in capturing rare conditions and maintaining consistent relationships between multiple health variables [117]. For tabular data, methods like CTGAN and TVAE [316] have demonstrated ability to learn complex distributions while preserving correlations between features. However, these approaches often struggle with high-dimensional data and can introduce subtle biases that impact downstream model performance [15]. Recent work has also explored combining synthetic data with differential privacy to provide formal privacy guarantees [153]. While these methods offer stronger privacy protection, they often face significant utility loss, particularly for rare but important cases in the original dataset [322].

Secure Data Combination Recent work has explored methods for securely combining datasets while preserving privacy and improving model performance. Early approaches focused on using secure multi-party computation to enable multiple parties to jointly train models without sharing raw data [12]. However, these methods often struggled with computational overhead and communication costs when dealing with large-scale datasets [207]. More recent techniques have introduced frameworks for evaluating potential data partnerships before commitment. These approaches use privacy-preserving protocols to estimate the compatibility and complementarity of different datasets [183; 55]. Some methods focus specifically on measuring distribution shifts between datasets without revealing sensitive information [88]. Others trained the downstream model in private, but limited to LASSO [303]. Several systems have been developed to facilitate secure data combination in specific domains. In healthcare, methods have been proposed for securely combining patient records across institutions while maintaining HIPAA compliance [247; 303]. Financial institutions have explored similar approaches for combining transaction data while preserving client confidentiality [196].

Federated Learning. Cross-silo federated, decentralized, and collaborative machine learning [207; 189; 33; 154] focus on acquiring more data through improved data governance and efficient system design. Healthcare machine learning is considered especially suitable, as health records are often isolated [255; 315; 223]. Yet, even though no raw data is shared, model parameters or gradients flow through the system. As the federated computing paradigm offer no privacy guarantee, the system is vulnerable to model inversion [107] and gradients leakage attacks [31; 335]. A subtle but urgent concern is that privacy risks discourage the very formation of the federation when optimisation is traded off with privacy [200; 252]. Building on the insight that useful data is often disparately owned, we tackle the specific incentive problem between pairs of data players where one side trains the model, instead of scaling up a federation (number of parties) to address data access issues. We thus focus on making this exchange efficient, accurate,

and private.

Compared to vanilla Federated Learning, an MPC system [264; 324; 34; 164] provides stronger guarantee in terms of input security. Model owners and data owners can potentially federate their proprietary data, including model weights, training, and testing data, can work together under stringent privacy requirements. Our work extends the line of works [320; 323; 34] that demonstrates the potential of incorporating MPC in various federated scenarios. On the practical side, unlike mobile-based networks for secure federated learning protocols [33], our system assumes a smaller number of participants, where communication cost and runtime are not dominant concerns.

Differential Privacy for Data Sharing Differential privacy (DP) [90] offers formal privacy guarantees for sharing data and training machine learning models. While DP mechanisms can protect individual privacy when releasing model outputs or aggregated statistics, they face significant limitations for interorganizational data sharing. The primary challenge is that DP operates on already-pooled data, but organizations are often unwilling to share their raw data in the first place [92]. Even when organizations are willing to share data, the privacy guarantees of DP come at a substantial cost to utility, particularly in machine learning applications. DP-SGD, the standard approach for training deep neural networks with differential privacy, significantly degrades model performance compared to non-private training [2]. This performance impact is especially pronounced in data-constrained settings, where recent work has shown that large models rely heavily on memorization of rare examples that DP mechanisms tend to obscure [98]. The privacy-utility trade-off becomes even more challenging when dealing with high-dimensional data or complex learning tasks. Studies have demonstrated that achieving meaningful privacy guarantees while maintaining acceptable model performance requires prohibitively large datasets [21]. This limitation is particularly problematic in specialized domains like healthcare, where data is inherently limited and performance requirements are stringent [110]. Recent work has attempted to

improve the privacy-utility trade-off through advanced composition theorems and adaptive privacy budget allocation [235]. However, these approaches still struggle to match the performance of non-private training, especially when working with modern deep learning architectures [293]. While differential privacy offers important theoretical guarantees, our work focuses on the practical challenge of enabling data owners to evaluate potential partnerships before sharing any data, addressing a key barrier to collaboration that DP alone cannot solve.

5.10 Conclusion

Our work demonstrates that privacy-preserving data valuation can help organizations identify beneficial data partnerships while maintaining data sovereignty. Through `SecureKL`, we showed that entities can make informed decisions about data sharing without compromising privacy or requiring complete dataset access. As the AI community continues to grapple with data access challenges, particularly in regulated domains like healthcare, methods that balance privacy and utility will become increasingly critical for responsible advancement of the field. As noted in Section 5.6.3, our approach has several limitations, including the fact that, despite impressive aggregate results, our method is less effective for individual hospitals; this finding is fertile ground for future work. Additionally, our work present opportunities for follow-up research. Our method assumes static datasets and may not generalize well to scenarios where data distributions evolve rapidly over time. A sequential version of our framework may more closely model dynamic data collaborations. Future work should explore extending these techniques to handle more complex data types and dynamic distribution shifts while maintaining strong privacy guarantees.

Lastly, extending our setup to other data-evaluation measures, any data combination method (if Turing-complete) can be made private [115]; yet, in practice, balancing the right trade-off of utility and privacy is non-trivial. Barring engineering difficulties, not all algorithms readily adapt efficiently in private. Previously, Chapter 4 [320], which assumed the trained model and test data, achieved relatively exact results; however, complex methods would exacerbate the operational challenges discussed in Section 5.7.2, as the source hospital now needs to prepare more data and development. Future work should explore the practical mileage of bridging MPC deployments to the stakeholders without extensive resources.

6 | HEalth: Privately Computing on Shared Healthcare Data

While machine learning models can operate on sensitive data in private, *auditing* these models, such as for decision fairness [263], becomes exceedingly challenging when the data is encrypted.

For example, when a patient arrives at the emergency room in a hospital, they may get admitted, turned away, or put in the waiting room — a sorting and organizing procedure known as *triaging*. Using ML for triaging can save cost, reduce emergency department overcrowding, thus allocating scarce healthcare resources to save the most lives; however, it is crucial that this potentially life-or-death decision process — automated or not — is fair and non-discriminative.

In this future powered by machine learning advances, auditing tools nevertheless lag behind. At the center of the challenge is data privacy: health regulations ensure that digital records are encrypted by default, so hospital decisions – automated or not – rely on protected attributes of patient data; yet, auditing requires fairness evaluations on those exact attributes. In the setup, sketched in Figure 6.1, we ask

Can existing homomorphic encryption help overseers audit private model decisions?

This chapter qualitatively describes a solution with novel protocol **HEalth**: Privately Computing on Shared Healthcare Data, included in [78]. It was an early document on applying Fully-Homomorphic Encryption (FHE) (Section 3.3) to machine learning.

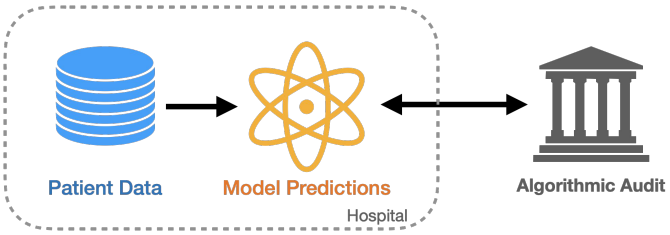


Figure 6.1: Setup: Hospital systems will deploy machine learning-based decision models, the outcome of which should be audited.
The Overseers from Section 2.4 take on the role of regulatory auditors, who would need to access the input and the decisions made by these models, which are kept private by default at the hospital level.

In 2019, we demonstrated through auditing triaging fairness that FHE can be feasible as a *privacy-preserving oversight solution*. Our setup and algorithm allow for efficiently auditing attributes of decisions, without decrypting the data-sensitive outputs of machine learning models.

By 2025, besides enhancing the writing with background and contexts, we have additionally made diagrams and clarified algorithmic descriptions, and contextualize our work to prior publication. Additionally, the overall system is compared with auditing differentially private models.

Contributions We posit the hospital fairness problem for external regulators to audit hospitals' proprietary ML models, which can be solved efficiently. This enables novel use cases of homomorphic encryption for machine learning. Specifically, we present

1. A multi-stakeholder, incentive-compatible key exchange, which allows for continuous uploads of healthcare records without frequent key refreshes.

2. Descriptions of secure fairness auditing algorithms that are practical for existing computational systems, without relying on expensive non-linearities.

3. Descriptions of a system that, once set up, does not require expert intervention.

State of the Art (2025) The framework proposed in [237] in 2022 directly subsumed the original 2019 contribution that was published in [175]: it achieves private fairness auditing in

a holistic system, while our work focuses on algorithmic feasibility in FHE. It "builds a chain of trust through enclave attestation primitives, combined with public scrutiny and state-of-the-art software-based security techniques, enabling fair ML models to be securely certified and clients to verify a certified one."

Notes The security protocol proposed is a combination of pre-existing ideas that were theoretically explored for, but not applied to, healthcare use cases. Threshold cryptography has yet to be standardized, and it is unclear if it can be adeptly governed [43]. Through the early demonstration of fairness auditing in hospital data, this work presents a potential future where public key infrastructure underlying the modern internet can extend to machine learning-related tasks. The notations are presented in Table 6.1.

Notation	English Description
$\mathcal{F}$	Function
$\mathbb{P}$	Probability
$\mathbb{N}$	Natural Numbers
$\hat{Y}$	Prediction
$Q(\cdot)$	Quantization
A	Protected Attributes
N	Feature(/Bin) Cardinality
K	Hospital Cardinality
p	Class Cardinality
c	Feature Category Cardinality
$\mathcal{M}$	Trained Model
Enc	Encryption Function
Dec	Decrypt Function
$\mathcal{H}$	Aggregate Histogram
H_i	i-th Hospital(/Week)'s Local Histogram
$H_i[j]$	j-th Bin of Histogram H_i
keygen	Key Generation
pk	Public Key
sk	Secret Key
pk^*	Common Public Key
sk^*	Common Secret Key
ct	Ciphertext / Cyphertext
pt	Plaintext / Clear text
$q(\cdot)$	Polynomial
qs	Scaling Factor
$\mathbf{T}$	Integer Threshold
k	Threshold, (Threshold-FHE)

Table 6.1: Summary of symbols in private computation on healthcare data.

6.1 Motivation and Background

As the work was originally completed in 2019, we give a historical context as motivation for our conceptual work.

Fairness and Machine Learning. Testing and modifying machine learning to be fair was a burgeoning field under the umbrella term *ML Fairness* [24; 48]. Our work is based on one example of statistical fairness criteria, "group fairness", which is also called the Equal Outcome Criteria or *independence*. Rooted in the pre-existing legal theory of disparate treatment [70], group fairness is formulated as

$$\mathbb{P}(\hat{Y} = 1|A = a) = \mathbb{P}(\hat{Y} = 1|A = b) \quad \forall a, b, \qquad \text{(Equal Outcome)}$$

where $\hat{Y}$ is the predicted label and A is a Protected **A**ttribute. When A refers to race, gender, or age[1], the criteria aims to have each subgroup under this attribute receive a positive outcome at equal rates (demographic disparity).

Privacy Challenges for Auditing Healthcare Data. While regulatory agencies mandate fairness, hospital machine learning requires protected attributes, which are inherently sensitive. If auditing requires data to be presented in clear text (without encryption), privacy may be undermined. Appendix E.1 includes detailed write-ups.

Fairness Risks With Switching to Differentially Private Models. The regulator-hospital privacy issue can be resolved if the model can be safely shared, such as when the model is trained in with differential privacy [90] (DP, Section 3.6.2). However, under DP, fairness suffers.

While releasing models trained with DP can protect the privacy of the individual records, [281]'s extensive testing found that DP-SGD [2] in particular gives unfair influence to majority members. Specifically, when a model is trained under DP-SGD, a member of the population

[1] In our original work A is one-dimensional i.e. A is race, and a is Caucasian. We can extend A to accommodate a collection of different features.

majority (under A) would have a higher impact on the differentially private models' decisions.

This phenomenon, aptly coined as the Privacy-Fairness trade-off [281], can be problematic for auditing: when the very goal of auditing is to encourage fairer machine learning models, auditing differentially private models may lead to hospitals training *less fair* models.

This trade-off may relate to learning under imbalanced groups; heavy-tail data tends to negatively impact classification, as studied theoretically by Chaudhuri et al. [58], and in practice by Suriyakumar et al. [281] — both inspired by healthcare domains. To test this out with real hospital data, we present Figure 6.2. Our fairness analysis uses models trained DP-SGD for mortality prediction from ICU data. We confirmed that model fairness (using "worst group fairness") is negatively affected when privacy protection is meaningfully strong ($\epsilon = 1$). For further analysis, see Section 6.5.

Homomorphic Encryption-Friendly Audits. If fairness functions can be *rewritten* without expensive non-linearities, then auditors could assess "secret" black box models via homomorphic encryption.

Need-to-Know Audits. Proposing FHE for auditing can improve efficiency on data that is already encrypted – auditing automated decisions that are done in bulk can also be achieved in real time, without relying on hyperparameter choices or needing backdoors. Thus, our solution improves upon decrypting private data to enable auditing (Figure 6.7, also argued in Raisaro et al. [247] for general medical audits). While generic FHE schemes are not *ad hoc*, ours scales to multiple audits per encryption, so regulators can make modify audits without needing key refreshes.

Opportunities in PrivacyML Incentives. At the time of the project, traditional private ML assumed training-related workloads [36; 112; 162]. In incorporating encryption technologies to machine learning, one class of problems was overlooked: the incentives pertaining to *using* confidential computation to build trust with stakeholders, as discussed in Chapter 3. Our work tackles the auditing of ML: apply practical private FHE to help with regulatory duties associated

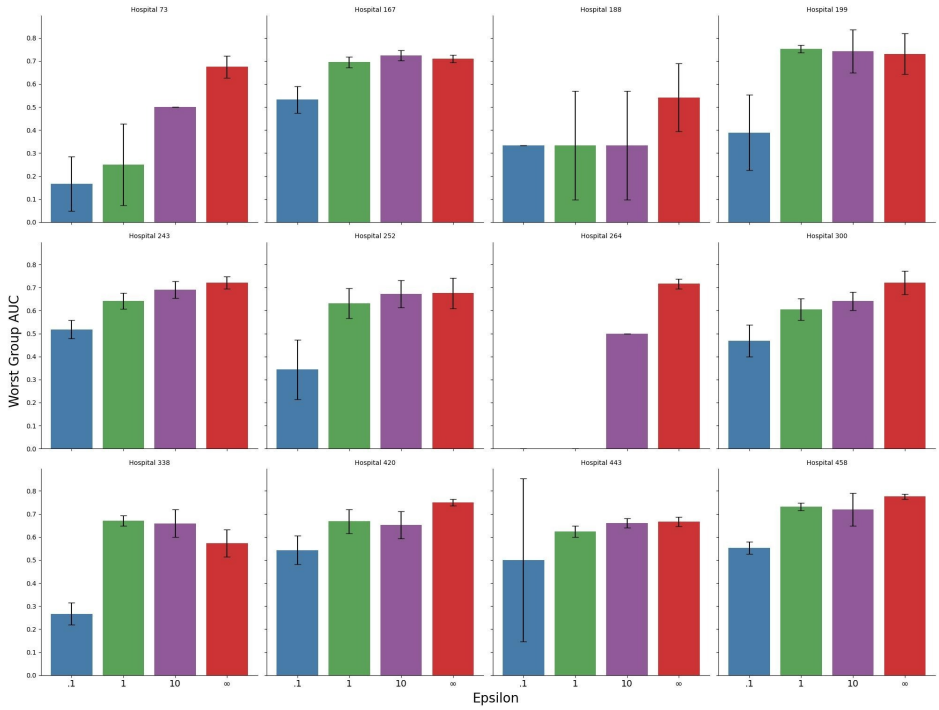


Figure 6.2: Fairness-Privacy Trade-off: per hospital group fairness performance under differentially private models with increasing privacy relaxations. The fairness metric, worst group performance (higher is better), under different privacy parameter $\epsilon = 0.1, 1, 10$ (smaller is more privacy-preserving) setups. $\epsilon = \infty$ refers to no privacy protection (ground truth) numbers. Models are trained with logistic regression under DP-SGD (Sec 3.6.2). Group fairness shown for intersectional groups [48]. *Note*: This experiment is run in 2025 on eICU data for the 24 -hour mortality prediction task in [244]; our data processing is detailed in Appendix D.1.

with future machine learning models, without leaking data.

Proof-of-Concept for Using FHE/HE for ML As brought up in Section 3.3, homomorphic encryption has been applied for single-stakeholder tasks, with significant computational overhead, and need extensive expertise to write and configure. Moreover, many costly operations become strenuous in FHE, seemingly incompatible with current machine learning paradigms. The original publication of [175] therefore focused on the conceptual validation of using FHE for a useful ML workload, and had only presented the qualitative descriptions.

6.2 Summary of Engineering Contribution

The original publication [175] focused on the potentiality of FHE for ML and presented the project's qualitative descriptions only. In 2025, we supplement this thesis with an expansive algorithmic description and new diagrams that clarify how expensive non-linearities are avoided. We also computed group fairness (by trait) on 3000 data points in 79 seconds on an 2019 MacBook Pro without further optimisation[2]. We hereby summarize the engineering hurdles experienced in both the initial development in 2019 [175], and our enhanced implementation in 2025.

Challenges in 2019 In 2019, an experimental prototype of the work was written in SEAL [60] to demonstrate the feasibility of the FHE use case; however, some core functionalities, such as BGV [42], were not fully available within the framework [211]. Moreover, parameter selection was a known challenge (Section 3.3.3), as SEAL's parameter setting defaults required both cryptography knowledge and a close understanding of the computation.[3] Lastly, for pure conceptual feasibility, our tests were written largely with converting made-up data arrays into integer arrays, not linked to real hospital data; as a result, real-world numerical stability was not tested.

Challenges in 2025 Our updated implementation leveraged Zama's FHE library, which includes ConcreteML [329]. We found the learning curve of using the FHE library to have greatly decreased compared to 2019, with engineering focusing more on utilizing existing library features.[4] The remaining engineering hurdles, which are largely functional, pertain to runtime optimisation (performance) and integer encoding normalization (correctness and numerical stability). Notably, Zama still requires integer arrays as the default input. Lastly, our new implementation uses real hospitals' datasets (eICU [244]) — a significant step forward from previous simulations.

[2]For comparison, this is the amount of data a typical hospital system's ICU unit generates in a year.

[3]Unfortunately, our code artifact, including our security parameters, was not carefully preserved.

[4]For example, the max operator for "worst-group" performance was not part of the original algorithm we supported, but can now be enabled out-of-the-box by Zama, albeit with a performance penalty (a non-linear approximation).

Limitations FHE libraries offer support for specific operations, including documenting sample code for a variety of efficient algorithms. While these operations can support general computations, not all machine learning tasks are supported out of the box. To complete our work, which is a novel ML workload on FHE, we are essentially writing tokenizers, quantization, as well as kernels for machine learning.

Frameworks can potentially do more – for example, categorical data that has many categories can be encoded as a one-hot vector, or it can be represented as a hashmap. These choices are thus similar to tokenization in natural language processing [273; 116]. Moreover, the pre-processing step has the liberty to transform data not only into an FHE-friendly format, but also to aid later computations on the encrypted data. Quantization can be automated in FHE frameworks, but only on low-level data types such as floats or strings, whereas modern machine learning operates at a much higher level on more generic data. This suggests the potential for private ML frameworks to co-design machine learning tasks with homomorphic encryption.

6.3 Problem Setup

Who To Trust Following semi-honesty in Section 2.5.2, willing participation is assumed in this setup. Concretely, we assume that all hospitals involved wish to be audited through our system, and that the regulatory auditors are trusted. Also mentioned in Section 2.5.2 is that all parties know what is being audited. As such, **the data formats and preprocessing steps are not secret**. The setup for hospitals is expanded on in Section 6.6.1.

6.3.1 Auditing Setups

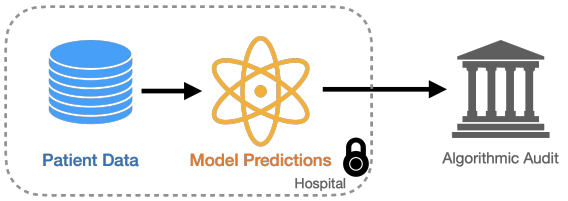


Figure 6.3: Existing: Hospital self-reports final metric. The hospital computes fairness function $\mathcal{F}$ from raw data, and reports the number to the auditor.

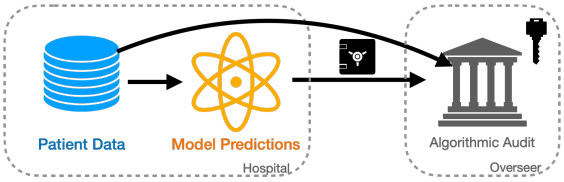


Figure 6.4: "Backdoor": Auditor decrypts encrypted data. The hospital sends encrypted data as well and the key to fully decrypt the data, including sensitive patient data. The auditor then computes $\mathcal{F}$ from raw data.

Existing approach: self-reporting. As illustrated in Figure 6.3, hospitals computing their own metrics does not constitute as a third party audit.

Our approach: FHE-friendly fairness audits. As illustrated in Figure 6.5, the hospital preprocesses and encrypts data. Then, the encrypted data is sent to the auditor, who cannot decrypt the raw data. The auditor, potentially with hospital agreement, decrypts only the auditing results. The raw data is never exposed.

Alternative approach: encrypt, but pass the key. Encrypting and decrypting all raw data by sending the auditor the encrypted data as well as the secret key introduces novel security risks, illustrated in Figure 6.4; under multiple hospitals, the algorithmic auditor now holds the

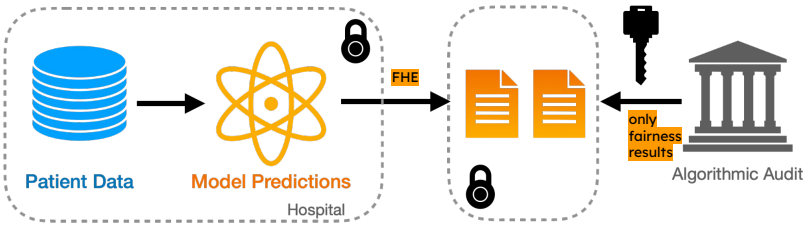


Figure 6.5: Our Method: Homomorphic Encryption-enabled Fairness Audit. First, the hospital encrypts data (with a special key). The encrypted data is sent to the auditor, who cannot decrypt the data except for the fairness results.

keys to all participating hospitals' raw data (Figure 6.6), and becomes a risky central repository of everyone's raw data.

UNDERMINING ENCRYPTION IS COSTLY, SOCIALLY The level of data access in Figure 6.4 far exceeds what is necessary for fairness auditing. This approach also has a social cost: undermining encryption by *bypassing* regulations gives regulators a backdoor to [53], disempowering individual patient's privacy over their health records [65]. Our approach, on the other hand, limits the data access of the regulator to what is necessary for fairness metrics to be computed on the encrypted data. At decryption time, only aggregate results are decrypted (Figure 6.7 under multiple hospitals). Appendix E.4 discusses more auditing setups.

6.4 Algorithms

Shared Knowledge Assumptions Because common to all parties is the knowledge **what is being audited** and **how the record arises**, such as using intake form data for emergency room triaging or using ICU patient data for mortality prediction. Thus, the associated known information is not private, as all parties are aware of the type of data.

This shared knowledge includes the number of predictive classes p, the number of features N[5]. Additionally, all hospitals' records are assumed to **contain all protected attributes**, and each attribute A's cardinality is known, e.g., how many values can race take.

Binning, Quantization, and Normalization We additionally assume a **pre-determined data format**, including quantization and normalization schemes.

Because each trait of the data needs to be converted into integer arrays for most protocols, we assume binning (for continuous data) and quantization, denoted as $Q(\cdot) : \mathbb{R} \to \mathbf{N}^m$. Namely, when the data type associated with data field D is too large to fit in one bit-limited integer, this function allocates m ints instead. As a result, each trait A may be associated with multiple indices. Moreover, we also assume that appropriate normalization Normalize$(\cdot)$ is applied to adjust the input for the specific formats each FHE encryption function take.

Multiple Hospital Assumption In our initial work in 2019, the hospitals' data is aggregated for one fairness metric. This is due to several considerations: 1. there is mandate to know the industry-wide statistic, so a joint statistic makes sense for all hospitals deploying the same model, as pictured in Figure 6.7. 2. there is more data when multiple hospitals upload to the system. Because hospital data can be sparse, this joint operation may be more realistic for real-time monitoring. As a result, the histogram description is for the aggregation of disparate hospitals.

[5] In our original work, a "trait" was used instead of a feature.

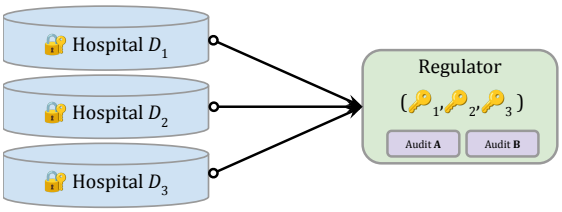


Figure 6.6: Regulatory mandates at odds with privacy compliance. Per HIPAA [53], patient data samples, $\{H_i\}$, should be stored and transported with *industrial-grade encryption*. Regulatory bodies need to ensure fair access to medical resources via auditing the decisions and outputs (methodologies $f_{\text{audit}}, f_{\text{eval}}, \cdots$), but can only do so via a backdoor (*key*), undermining encryption.

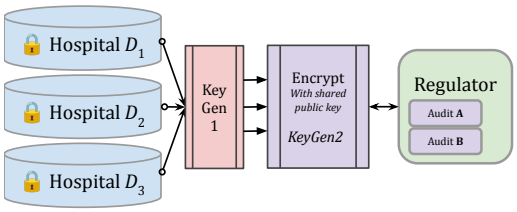


Figure 6.7: Secure MPC via Threshold FHE for Auditing. The hospitals' data H_i, each encrypted with a *unique* set of keys, is continuously audited for fairness. The regulator does not hold any of the hospitals' keys, and data is destroyed if a private key is destroyed. The results and only the results, are revealed when an auditor calls for a decryption event (not pictured).

However, the algorithm readily adjusts to parallel operation, where fairness metrics in each hospital is computed at once. More assumptions are addressed in Section 6.6.1.

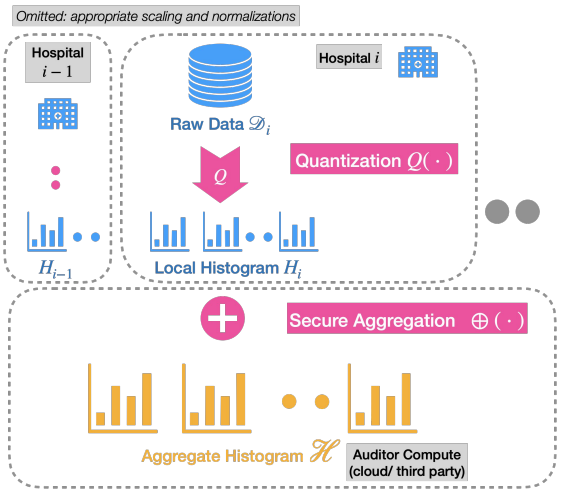


Figure 6.8: Preprocessing and Secure Aggregation Overview. Each hospital preprocesses their data with the *same* binning and quantization function $Q(\cdot)$, and encrypts their resulting histogram H_i to the server. The server performs aggregation of the histograms, outlined in Section 6.4, to obtain aggregate histogram $\mathcal{H}$ through Aggregating Hospital Histograms. Note that H_i's are encrypted before entering auditing compute, so no unencrypted data is shared.

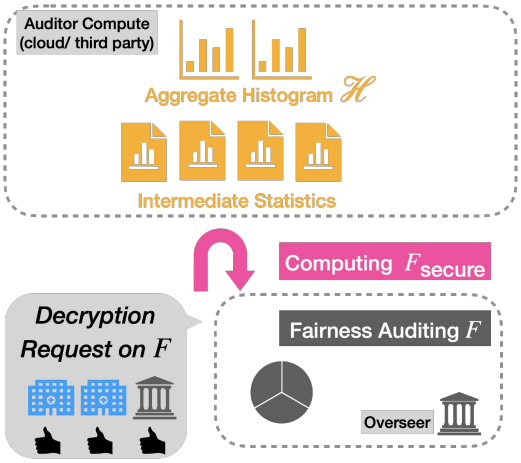


Figure 6.9: Fairness Audit Request Overview. After data has been aggregated, hospitals and overseers get together to answer to a *request-to-decrypt* that is specific to the fairness function in question, such as group fairness difference between races, as outlined in Section 6.4. Note that appropriate decoding, such as rescaling, is needed after decryption. For our protocol, all parties need to agree to decrypt the results.

Encoding and Summing Histograms With Binning and Quantization Bin all the data into N bins. Let $H_i[j]$ represent hospital i's j-th bin. This histogram's representation is discrete, and each value $H_i[j]$ is bounded by $\mathbf{T} \in \mathbb{N}$.

Representing each local histogram as $H_i \in \mathbb{N}^N$, we have aggregate histogram $\mathcal{H}$, summed over respective indices across hospitals[6]:

$$\mathcal{H}(K) = \sum_{i=0}^{n} H_i = \sum_{i=0}^{K} \begin{pmatrix} H_{i1} \\ H_{i2} \\ \vdots \\ H_{iN} \end{pmatrix} = \begin{pmatrix} \sum_{i=0}^{K} H_i[1] \\ \sum_{i=0}^{K} H_i[2] \\ \vdots \\ \sum_{i=0}^{K} H_i[N] \end{pmatrix}. \qquad \text{(Aggregating Hospital Histograms)}$$

The overview of the process is illustrated in Figure 6.8. Most FHE operations require integer inputs, such as an array of ints[7]. For data of known range, indexing or binning would suffice. For data spanning extreme values, such as blood test results, additional pre-processing may be required, by applying a quantization layer $Q(x)$. This quantization function is known, and uniformly applied, to all hospitals. Then each hospital computes their local histogram over an appropriate period of time[8]. The overseer can utilize the aggregate histogram for auditing needs, on a need-only basis. This aggregation is provided in all HE schemes.

Encoding Labels Because the outcomes are the features to audit, for each outcome $Y = y$, we have a corresponding histogram $\mathcal{H}(Y = y)$. They are assumed to be a few classes (such as admit, wait, turn away). This class information is not private.

Aggregative Property of Histograms Equation Aggregating Hospital Histograms, $\mathcal{H}(K + 1) = \mathcal{H}(K) + H_{K+1}$. This means the data can be continuously uploaded in the form of local

[6]Omitted: Normalize$(\cdot)$ is applied at the end to $\mathcal{H}$ to adjust for FHE-friendly input.

[7]This is usually not automatically done. For instance, the FHE compiler Concrete [329] asks for quantized values instead of taking floats, though some quantization operations are provided in the library.

[8]For triaging, this frequency would be approximately a week; for ICU mortality, approximately a month.

histograms. In a single hospital setting, H_i represents data from different times, such as every week.

Fairness Function Within $\mathcal{H}$, consider a protected attribute A represented with histogram indices J_A, meaning that $H_i[j]$ for all $j \in J_A$ represents A in Hospital i. (Due to quantizing or binning, one attribute's value may be spread over multiple indices.) Let J_Y be the collection of indices that represent model prediction $\hat{Y}$ in the data. Without loss of generality, overload $\mathcal{H}[j = a] = \sum_{a \in J_A} \mathcal{H}[a]$

Given group fairness notion Equal Outcome, suppose $y \in \{0, 1, 2\}$, we can compute $\mathcal{H}_y$'s corresponding outcome by attribute.

$$\begin{aligned} H_0[j = a] &= H_0[j = b] && \text{turned away;} \\ H_1[j = a] &= H_1[j = b] && \text{told to wait;} \\ H_2[j = a] &= H_2[j = b] \quad \forall a, b \in A && \text{admitted.} \end{aligned} \tag{Equal Outcome}$$

A metric of interest is the admission gap between the highest and lowest admitted groups, such as by race [99]. Given the limited number of groups, and that parallelization is "free" in FHE, we can simply compute them through adding up pair-wise differences. We arrive at the set of fairness values

$$F = \{H_y[j = a] - H_y[j = b] \mid \forall a, b \in A, \forall y \in Y\}. \tag{Fairness Values}$$

These additions and subtractions are also supported by most FHE schemes.

Request to Decrypt While F can be returned as is, it may be too much data. Alternatively, a max operation can be applied, and the indices can be return as "the most unfair groups for a label". However, computing max (or arg max) efficiently remains an open problem in FHE [63]. Nevertheless, in fairness audits, the questions are known *a priori*. A pressing concern may be: *Are*

black patients turned away more often than Caucasian patients? To which the auditor submits a **Request to Decrypt** in the form of retrieving the decrypted value of the corresponding difference in F. Because the metadata is assumed to be public, the indices are not private, yet the encryption of F enforces a request.

Scaling Intermediates The auditing function is now written without using any non-linearities, thus bounding the error and ciphertext length without needless bootstrapping. With K hospitals, N traits, and c classes i.e., how many values for race, a predictor for p outcomes would give us $O(pcNK)$ number of values to compute. This memory growth is manageable, and does not scale with the number of users. On the other hand, time wise, most operations can be parallelized; while the key refreshes are not needed, the operations are efficient.

6.4.1 Proposed Applications

We propose a holistic monitoring system for transparent audit, anomaly detection, and insight discovery that benefit hospitals and regulatory bodies.

Fairness over any population feature. Select one particular statistic: e.g. admission rates of groups by protected traits (gender, race). The goal of the computation is to compute an industry-wide aggregate statistics of admission rates by class with the goal of providing an accurate benchmark and identify outliers. This involves careful assumptions that we specify in our document.

Anomaly detection and pandemic discovery. Surface anomalous traits such as patient history, location, and patient traits in regard to their health (e.g., age). This helps detect and prevent breakouts of epidemics by discovering trends on time, and has the potential to generate insights for rare diseases and chronic illnesses. These tasks serve public interest, but are very hard for doctors to do by themselves in a regional hospital.

6.5 Comparing with Differentially Private Model Releases

Fairness suffers under DP. As mentioned in Section 6.1, DP-SGD gives unfair influence to majority members in medical data, which can be hard to detect with standard measures of group fairness [281], suggesting that DP releases may undermine the coverage of auditing. Figure 6.2 shows our analysis using differential privacy for mortality prediction from ICU data from real hospitals [244], based on data cleaning procedures outlined in Appendix D.1.

While released DP models protect the privacy of the individual records (Section 3.6.2), model fairness is negatively affected when privacy protection is strict. Moreover, each hospital would need different parameter settings to achieve better fairness, introducing another operational cost to audit[9]. Auditing the model effectively still requires anonymizing patient data, which is known to degrade representation of minority statistics [158]. Additionally, DP-treated data disclosures do not give the data parties a chance to re-negotiate against what is computed on their data. In our full-party protocol, if the audit function is not agreed upon, depending on the threshold, any party can refuse to participate in the decryption, thus nullifying the request.

Comparing DP and FHE for Fairness Algorithms Though the histogram statistic in Section 6.4 is an aggregate one, reminiscent of DP-based aggregation like RAPPOR [93], doing so with FHE is very different. Coarsely speaking, DP algorithms for *model releases* are **less accurate** than non-private models. Even though the leakage to patient data is bounded, releasing a hospital model (even if just to share with the auditing agency) is still **more public**; thus, as a system for the sole goal of enabling auditing, DP-releases risk privacy loss, and tends to degrade utility [281].

[9]For reference, typically, in training private models on patient-level data like ours, a privacy parameter ϵ is set between 1 and 10 [298].

Utility degradation. In healthcare domains, worsened model performance is the primary reason to not deploy DP-models, as hospitals prioritize model utility. The undesirable trade-offs under DP, for both fairness and utility, is extensively documented in [281].

Nevertheless, as argued in Section 3.2, both input- and output-privacy are crucial considerations in designing new systems. In fairness audits, in particular, our method does not require any model release (privacy) and does not require in model degradation (utility).

6.6 Overview of the Protocol Setup

6.6.1 Setup Assumptions for Hospitals

Hospital preprocessing assumptions. Data is assumed to be uploaded using the key *pk* corresponding to each hospital, taking the same tabular format ({features} → decisions), encapsulating the doctors' observations (which often include the protected attributes), as well as the resulting decision. Figure E.1 includes a snippet of example data. We assume known data ranges for real number representation and tokenization before encryption e.g., having 1e2 fields, each taking up 1e3 values. **Encoding Basic Statistics.** Averages can be computed locally, such as average age for each visit (binned by timestamp). The weighted average is the overall average, surfacing "average hospital occupancy". **Fairness Analysis.** Each hospital constructs histogram data on encrypted data regarding a specific feature, e.g. admission rate by gender, uploads it to the server, which aggregates them, computing the mean and standard deviation, then return the outlier identification for joint decryption.

Procedure Multiple Hospitals With one hospital, the protocol is two-party with hospital and auditor as the two parties. The secret key is used to decrypt the result, with the auditor requesting the hospital through a formal request. Before decryption, the hospital passively uploads data that is encrypted with their public key.

When multiple hospitals are involved, the key exchange is Threshold FHE [14], an idealized version of multiparty extension of FHE. The hospitals are required to be in sync: at the key generating stage, they are all "online" at the same time. As a default, we require **full-party** in order to decrypt, to adhere to the Need-to-Know principle that is in accordance with classified data access, where a justification and a clearance are both needed [67].

Privacy Model The system is built under the assumption of an honest-but-curious threat model (Section 3). When both the hospitals and the regulators are motivated to participate in the audit, the hospitals serve as trusted clients. A third party, such as a secure cloud provider, an overseer, or a leader hospital, serves as the server (the protocol adjusts for these variants).

The parties jointly generate public and private keys, which they use to upload data, perform computation, which are done continuously. Eventually – or periodically, in the case of monitoring or regular compliance checks – all parties jointly decrypt the result, thus concluding an auditing event. A simplified diagram is shown in Figure 6.7, with additional notes in Appendix E.2.

Long-term Benefits A particular benefit of our approach is that as long as the hospitals keep their keys, the centralized auditing party can potentially keep developing (aggregative and linear) algorithms to compute collective statistics, including implementing causal discovery algorithms on historical data without revealing secrets.

When computational resources allow, we can take combinations of multiple traits, and output anomalies continuously. This can be made interactive where the threshold changes, but there is no need to rehash keys when the computation changes.

Multiparty Key Exchange Limitations This work assume hospitals storing their data in the same format, when in reality any data unification is non-trivial. Deployment across the country is clearly challenging, yet the method supports a regulatory body to start from a small trial, such as from hospitals in the same region. The limitation of the protocol is that error is slightly bigger compared to the regular BFV, which requires larger parameters to accommodate.

6.7 Conclusion

Confidential computation mitigates incentives problems. This chapter demonstrates that the triaging fairness – an auditing workload – can be efficiently and accurately implemented using FHE, contributing to better trust in proprietary machine learning models by operating on encrypted data.

Heavily regulated domains may especially benefit from our method, because it does not require a risky decryption of sensitive data in order to enable auditing. Additionally, the system points to a paradigm of encrypted medical records, which may be used to enable automated decisions, to also be auditable *real-time*.

Since the drafting of [78], efficient implementations of homomorphic encryption has exploded rapidly. BGV [42], leveled homomorphic encryption, became a main feature of SEAL in 2022 [211]. Set intersections, chest x-ray classification for pneumonia risks, and many other ML operations are now computable in FHE settings [128; 38; 216].

On the other hand, usability and trustworthiness remain a challenge for their adoption in healthcare domains. Supporting auditing on invisible data and hard-to-grasp models can alleviate the privacy fears regulators may have with respect to private machine learning.

7 | Towards ML-Privacy Co-design, A Paradigm Shift

Machine learning and privacy are both experiencing their real-world moments, where academic advances have the potential to be particularly influential very fast. How we frame privacy as a community impacts how we grow as a field. Future work on private ML needs a mindset change, where model optimization and privacy engineering are not designated to their mutually exclusive lanes. One direction of future work thus pertains to aligning the goal of machine learning with that of privacy and vice versa. Specifically, I argue for a co-design that refers to developing optimization algorithms to be more privacy-friendly, such as for secure technologies, akin to how we have been developing deep learning for specific hardware [136].

7.1 Adapting Data Valuation Techniques for Unseen Data

In optimization, evaluating training data – existing [120; 172] or potential [202; 150] – has become increasingly relevant for large scale language modeling tasks. These are sometimes explicitly formulated as influence functions like Koh and Liang [165]'s, where a scalar value is associated with training examples. Novel influence functions for data can potentially be better suited for LLMs through optimizers [314], speed [124], and scale [120], with potential room for improvement in terms of robustness [26]. These optimization techniques may be adapted for acquiring new data using methods outlined in our work (Chapter 4), so that data appraisal can apply without data sharing on *unseen* data.

Potentially, a well-crafted benchmark suite on data appraisal without data sharing can encourage scalable, efficient computation, with few hyperparameters, in order to proxy their usefulness in the real world. To that end, existing tutorials and competition on data valuation [202; 150] can be extended to acquiring unseen real-world data where the distribution may not match such as for domain generalization [167]. Adding private computation can encourage visibility of systems parameters – computational cost, throughput, and execution speed – while ensuring high accuracy across the privacy-optimization stack.

7.2 Gap and Opportunity for MPC for ML

Early efforts in supporting machine learning in private point to massive potential for building towards mature robust systems. Knowing that privacy is a complex problem, we propose maturing privacy techniques as standard tool-kits for general machine learning computation.

Even if absolutely minimizing the downside to model performance and privacy feels impossible, the battle is far from "lost". When machine learning systems faced similar blockers, the interdisciplinary community came through and tackled them, from frameworks [239; 1] to their compilers [61; 173; 257], or even designing completely new programming languages [214; 6].

This thesis identified a few gaps in the existing landscape with machine learning used in secure computation, summarized below.

1. **Algorithmic development** Constructing a performant implementation of a machine learning computation is nontrivial.

2. **Framework readiness** The lack of production-scale library for MPC (defined as being more mature than CrypTen [164]).

3. **Communicating to machine learning research** The lack of awareness in machine learning community and accessible information, such as runnable examples and engineering documentation.

4. **Missing shared engineering practices** Secure engineering and machine learning engineering are disparate mini-communities.

5. **Selling MPC for the real world** Stakeholders, especially those in data-sensitive domains, have limited conceptual knowledge on secure computation, such as its ease of deployment for small organizations without needing specialized hardware.

7.3 Supporting Optimization–Privacy Co-Design

Akin to "hardware-software co-design" [79; 286], privacy and optimization also have a symbiotic relation where progress in one can feed into, and improve upon the other, in a virtuous cycle. A broad definition of co-design already includes works that formulate a relationship between utility and privacy. We are far more optimistic, however.

We dream up a landscape of co-design that puts both goals as paramount, where trade-offs that matter are avoided. This section introduces some concrete thoughts towards this field, in method, experimentation, benchmarks, and communication.

Designing Optimization Methods to Minimize Privacy Trade-off The design of optimization techniques, which has been key to advancing modern machine learning [37], can branch off towards more explicit privacy goals. DP-SGD [2] (introduced in Section 3.6.2) has become a popular method for ensuring differential privacy in machine learning. It successfully optimizes deep learning models in an SGD-like interface while bounding privacy budget. In a similar approach, I changed the optimization process in recommendation models to enable machine unlearning [317]. It aims to "exactly unlearn" recommendation data without degrading the recommendation model, while also avoiding the cost of re-training [317].

The requirements of privacy and utility can be preserved through the consistency of the resulting loss function, which matches that of retraining without the removal data. This framing lets privacy *participate* in optimization. Besides machine unlearning, such a requirement can be extended to include different models, and to include myriad privacy goals.

Understanding Optimization Trade-offs In Privacy Design Knowing that privacy is a complex problem, securing inputs [115], bounding the output's leakage [91; 326], learning on-device [302; 119], or using secure hardware [237; 213] are not necessarily competing solutions –

we will likely need many of them. While the early efforts in supporting machine learning in private may be limited in adoption, they nevertheless point to potential for building towards mature, robust systems (such as FL, as introduced in Section 3.6.1).

Benchmarking privacy with respect to other trade-offs can appeal to communities that already care deeply about privacy, such as clinical healthcare. Suriyakumar et al. [281] benchmarked privacy, utility, and fairness on real-world trade-offs, which our work in Chapter 6 extends to practical audit setups. Expanding on them can clarify the narrative, empowering these communities to advance the symbiosis of privacy and optimization.

Mastering Interdisciplinary Communication. The engineering hurdles notwithstanding, a pressing "people issue" lies in communication between different communities that lack a common language. Specifically, secure ML solutions lie between those who can use Secure ML to solve real-world problems they face and those who can contribute algorithmically.

In our work we found that abstraction-mismatch prevents effective communication between disciplines, and the lack of shared tooling makes practical collaboration strenuous. To that end, I have co-led a NeurIPS tutorial on PrivacyML: Meaningful Privacy-Preserving Machine Learning and How To Evaluate AI Privacy [318]. This tutorial marked a first stab at addressing the knowledge gap between secure ML engineering and machine learning safety. Moreover, we published in interdisciplinary channels that spoke to model owners' interest [320], to societal needs for healthcare data sharing [106], and to cryptographic techniques' applications [78]. Though these efforts may not result in near term citations, they nevertheless build up momentum for projects that blur the line of each discipline to solve problems they respectively see.

Just like the integration of systems and machine learning, bridging the communication chain "closer" involves reaching communities where they are. In the past six years, I have been fortunate to organize Machine Learning for Systems, a workshop at NeurIPS [319]. This thesis marks the beginning of similar efforts to foster a healthy community of effective in-person discussions,

grounded in scientific methods and evaluations.

8 Conclusion

8.1 Incentive-aligned Privacy for Machine Learning

This thesis introduces the Three-Actor Ecosystem that captures the complex incentive tensions between model developers, data owners, and overseers in machine learning. Rather than restricting machine learning progress with privacy constraints, the thesis reframes "privacy issues" as emerging incentive problems, which computer scientists can pinpoint and mitigate, perhaps through thoughtfully engineered secure systems. In doing so, we may progress towards a more sustainable future – better machine learning that is also more privacy-respecting.

As machine learning scales, so does the applicability of this thesis. Key to our work is the feasibility of respecting both privacy and utility[1]. As the world adapts to AI technologies, we will face many similar growing pains, as though progress and agency must conflict; regardless whether these issues explicitly take on the word "privacy", understanding and addressing the underlying incentives can substantially move us towards healthy progress.

Though privacy and utility cannot always be without trade-off, we nevertheless argue they can be *co-designed*. Specifically, this thesis concludes that demonstrating utility outside privacy, such as resulting in more performant models, will be crucial for convincing stakeholders to adopt privacy-preserving methods. We presented the first fast and equitable data appraisal (and dataset

[1] In the narrow domain of this document, utility pertained to model performance, while privacy referred to the data owning parties' control over their private data.

evaluation) without data sharing, where a model owner can appraise another party's data without requiring any data (or model) sharing between the two parties.

For secure and confidential technologies, in particular, we demonstrate their capacity to ease incentive issues by virtue of utilizing all the underlying data. Our engineering insights uncover several shortcomings with present secure ML technologies, which we had overcome to enable data appraisal, fairness auditing, and dataset-to-dataset evaluations. Despite the difficulty in engineering secure systems for ML workloads, we hope to gain much more from relentlessly trying: once engineered, these systems can be readily deployed (just like machine learning [176]!).

Lastly, privacy is not the villain of our story. When it appears like an obstacle, we argue it is, perhaps, simply a sign of maturity in our relationship with data. This thesis advocates for a private data economy where individuals retain control, where data collaborations can be equitable and fair, and where proprietary models are governable without over-surveillance. Through demonstrating secure, incentive-respecting methods, we seek to inspire an open market that incentivizes efficient, equitable exchange of data, which, in turn, fuels AI development. But those are not the only sustainable futures we can build; if we can reflect on privacy pains to discover a better and *less restricted* machine learning, machine learning can better grow as a field, too.

8.1.1 Alternative Views

AI Privacy framing. Framed under AI privacy, my work specifically develops secure techniques to preserve data rights: to share it equitably, to rectify it after sharing, and to be able to examine models with respect to privacy and security.

Healthy data ecosystem. My research explores the concept of a private data economy, where individuals retain control over their data, while fostering a healthy market environment. By empowering individuals with the right to be forgotten and other data rights, we can establish sustainable AI oversight for the future.

8.2 Key Contributions

I. Definitional. Privacy is not about preventing data sharing – rather, it is about respecting each player's simultaneous desires for privacy and utility.

   (a) Formulated the **Three-Actor Privacy Model** that characterizes long term incentives between the stakeholders of machine learning development. Crucially, this framework acknowledges the model owners' interests to keep their models and data proprietary.

   (b) Introduced the **Data Appraisal Problem** of enabling model owners to evaluate potential data partnerships before any data is shared — a key barrier to collaboration, underserved by preexisting Privacy ML techniques.

II. Engineering. We presented secure and confidential computations that avoided the current major pitfalls of supporting ML workloads in private, while serving to mitigate growing tension that arise from AI progress.

   (a) Novel, empirically-tested measures for data partnerships. We presented the first MPC-based Influence Function, the first MPC-based dataset divergence. They are accurate, practical, with minimal leakage (no data is leaked except for the score/price output).

   (b) Novel FHE-based ML fairness audit for models. We encrypted algorithms to audit the fairness of proprietary ML deployments in hospitals.

III. Methodological. Co-designing privacy and optimization can build a deeper relationship between private computation and machine learning, with the potential for a virtuous cycle. We demonstrated privacy advancing utility, compared private methods directly with non-private methods, and argued that secure computation is effective through its utilization of all the underlying data – just like machine learning.

Data Appraisal Without Data Sharing [320]. Research performed at Facebook AI Research, New York. The authors thank Brian Knott and Alex Melville for helpful discussions.

Dataset-to-Dataset Evaluations Without (and Before) Data Sharing [106]. Research done with Keren Fuentes and Irene Chen. I thank Kerrick Staley for helpful discussions.

HEalth: Privately Computing on Shared Healthcare Data [78]. Research done at Microsoft Private AI Bootcamp. The authors thank Wei Dai and Kristin Lauter for helpful discussions.

A | Three-Actor Privacy

A.1 Dataset Divergence Technical Discussion

$\mathrm{KL}_{\mathcal{XY}}$ Discussion One key assumption that is satisfied by the hospital setup is that the in-distribution data is *non-overlapping* across hospitals. Suppose $\mathcal{D}_{\mathrm{a}} = \mathcal{D}_{\mathrm{tr}}$, $\mathrm{KL}_{\mathcal{XY}}$ would produce dangerously small numbers, as the logistic regression would struggle to distinguish the two datasets. This turn this would fool the source hospital into acquiring $\mathcal{D}_{\mathrm{a}}$, only to realize it adds nothing. When the datasets are very different, the fit of $\mathrm{Score}(x, y)$ would matter, too, because an overfit model would produce high scores consistently, rendering the measure not useful. For that reason, we reimplemented the original [267]'s implementation to induce more reliable early stop.

In our experiments $\mathrm{KL}_{\mathcal{XY}}$ is flexible and stable, though not directly predictive of downstream model loss. We tested it for guiding dataset combinations in ICU data across 13 hospitals. Initially inspired by the ratio $\frac{P_{MO}(\mathrm{d}x)}{P_{DO}(\mathrm{d}x)}$, this heuristic score outperforms ratio-approximations where the score is substituted[1]. Though not a direct estimation of KL-divergence, it is an MLE-based method for "estimating the unseen" where data is limited, and the underlying domain is "broad" [300; 234]; certainly, more research is needed to carefully set up the assumptions needed.

[1] A major reason is that the ratio substitutions render the KL approximation sensitive to noise in low-data regimes. See later results for its unstable p-values from ratio scores on hospital ICU data.

B | PrivacyML

B.1 Input-Output Privacy

Chicken-and-Egg Problem: Output-Privacy Discussion Output private techniques may not be ideal. As a philosophy, output privacy emphasizes that, once a collaboration took place, the resulting product, such as a machine learning model $\mathcal{M}$ or the output of $\mathcal{F}$, would not leak too much input information. While potentially assuring, its privacy guarantees *presuppose* a known system that both parties *would have* committed to. Privacy implicitly depends on data sharing or model release agreements in the first place. In prevalent economic theory on technology adoption in the early stages, [156] suggests that when the participants are not well-incentivized, adoption stalls. While not specific to PrivacyML as a technology to adopt, it is nevertheless useful to consider. Relying solely on the natural adoption of output private technologies, such as differential privacy (Section 3.6.2), will be insufficient for entities that are still exploring intent, as it is prone to this chicken-and-egg causality conundrum, particularly used for modeling establishing early partnerships [49] as well as infrastructural technical adoption [47].

Notes: Our Usage of Input-Private Methods In this thesis, the privacy setup, by design, protects model owner, i.e., without sharing $\mathcal{D}_{\text{tr}}, \mathcal{D}_{\text{te}}, \theta$. Therefore, purely output-private setups, where model weights θ are shared externally, fall out-of-scope. However, recognizing that models eventually make their way into the world through a service or a product, without protecting

	Input Privacy	Output Privacy
Privacy goal	**The input to a process is not observed.**	**The output of a computation does not reveal the input.**
Scenario	Joint training, post-training, and validation of models. Data appraisal, audit, and evaluation (Chapters 4, 6, and 5).	Open-weight (closed data) releases. ML-as-a-service such as chatbots, APIs, and embeddings.
Challenge	Distinct entities, such as model and data owners, do not want to share data with each other.	The released model, $\mathcal{M}$, may reveal details about training data $\mathcal{D}_{\text{tr}}$[1].
Privacy-preserving mitigations	Secure Computations: SMPC (Sec 3.4), FHE (Sec 3.3), and TEEs [188].	Differential Privacy (Sec 3.6), *Exact* Machine Unlearning [317]
Empirical mitigations	Federated learning[2] (Sec 3.6.1), using synthetic data (Sec 3.6.3).	Approximate unlearning [195], k-anonymity [260]
Related ideas	Data marketplaces, data governance [159]	Memorization and copyright [73], robustness and replicability

Table B.1: A useful dichotomy for machine learning privacy is input- vs. output-private methods. While not mutually exclusive in all scenarios, they *typify* two philosophies towards privacy protection for a system, leading to separate approaches. Section 3.2 argues that purely output-private methods may not ameliorate the privacy conundrum pre-partnership, where data or model releases are not predetermined.

both *input*- and *output*-privacy, trust in data sharing for machine learning nevertheless erodes. For example, output-private methods lend nicely to the release of the results from private data appraisal functions.

B.2 Contextual Integrity Relation

This thesis argues that privacy issues in machine learning often represent breaches of contextual integrity: if common implementations of these sharing are not privacy-preserving, each party‘s normative privacy expectations *are* violated. My works therefore propose alternative implementations to achieve the same goal that better respect these expectations.

B.3 Further Readings on ML MPC SOTA

Notably, SecretFlow [85] has emerged as a promising production-supported framework, though efficient private functionalities are still in beta. On the other hand, modifying transformers' modules may result in huge efficiency gain, but risks some drop in accuracy. This empirical trade-off is, however, distinct from bit-limits in FHE, or optimisation limits like differentially private training (see Section 3.6.2). Knowledge distillation for the feed-forward layers is introduced in MPCFormer [186] in 2023, upon which SecFormer [199] improved in 2024 with faster and more accurate GeLU and layer norm approximations.

Quadratic approximations for GeLU and their subsequent Fourier approximations drove recent developments in improving secure transformer inference, consisting of Cheetah [141] in 2022, MPCFormer [186], PUMA [85] and MPCFormer [186] in 2023, Bolt [233] and SecFormer [199] in 2024.

B.4 Policy Implications for Machine Learning Privacy

Practical considerations of machine learning affect both existing and emerging policies [256]. This section discusses policy applications of the technical contribution.

Existing privacy laws for high-stakes domains. Many aforementioned privacy-enhancing techniques are motivated by existing legislations [95; 94; 182] and standards regarding high-stake domains like healthcare [53]. However, enforcing these policies responsibly is tricky, as critical tasks often require accessing sensitive data [325; 35]. To that end, Chapter 4 offers a data-compliant way to efficiently allocate high-stakes resources without trading off privacy and utility, and Chapter 6 offers the governing bodies a feasible approach to securely uphold accountability without introducing unwanted trade-offs.

Copyright in relation to privacy. Copyright law has emerged as a pressure point between model owners and data platforms [122]. While copyright does not directly address privacy policy, the supply chain of generative AI [181] surfaces policy directions to regulate large scale machine learning development such as data auditing [271], hardware, hosting, scraping, and deployment; oftentimes, privacy is a major, related consideration. Additionally, copyright's applicability to machine learning models relates to the extent to which the model 'memorizes' or reproduces training data, which relates to Chapter 5 where privacy evaluation is supported.

Independent, Third-party Audits for data and models can broaden data access and uphold public good values [159; 249]. Mandates such as bias auditing in New York City [121] highlight its potential for regulating AI.

Secure computation can lower the risks for data access across the board for audits, not only when data is a bottleneck. For instance, the legal liability of the auditing party [121] may be reduced, as the analysis can still be performed without accessing data or model [78].

Contributions This thesis demonstrates to policy stakeholders that auditing data is not necessarily a compromise to data propriety (Chapter 6), and evaluating models does not need to come at the cost of scientific soundness (Chapter 5). Secure computation offers an attractive alternative to all-in, all-closed, or complex data sharing contracts [325], allowing more effective governance for high stakes data and application.

C | Data Appraisal

C.1 Forward Influence Details

Influence Setup. Recall that the data is owned by two disparate parties: a *model owner*, who is developing the model, and a *data owner*, who possesses the dataset $\mathcal{D}_a$ to be appraised. The model owner begins with a test set $\mathcal{D}_{te}$ and their initial training set $\mathcal{D}_{tr}$. Before acquiring the data $\mathcal{D}_a$, the model owner wants a peek at the utility gain from updating θ to fit $\mathcal{D}_{tr} \cup \mathcal{D}_a$. The initial model parameters $\hat{\theta}$ are obtained by minimizing the regularized empirical risk on $\mathcal{D}_{tr}$:

$$\hat{\theta} = \arg\min_{\theta} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{tr}} L(\boldsymbol{x}, y; \theta) + \lambda\|\theta\|_2^2. \tag{C.1}$$

If the dataset $\mathcal{D}_a$ were included, new parameters θ^* would be obtained by minimizing risk on dataset $\mathcal{D}_{tr} \cup \mathcal{D}_a$ instead. The value of concern is the utility of $\mathcal{D}_a$, as evaluated on test loss:

$$U(\mathcal{D}_a) := \frac{1}{|\mathcal{D}_{te}|} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{te}} L(\boldsymbol{x}, y; \hat{\theta}) - L(\boldsymbol{x}, y; \theta^*). \tag{C.2}$$

Influence Derivation. Given Equation Dataset Utility, we make a linear extrapolation by supposing that the resulting parameters (from an infinitesimal upweighing of a set of existing data)

are very close to the original:

$$U(\mathcal{D}_{\text{a}}) \approx \frac{1}{|\mathcal{D}_{\text{te}}|} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\text{te}}} \nabla_\theta L(\boldsymbol{x}, y; \hat{\theta}) \cdot (\hat{\theta} - \theta^*). \tag{C.3}$$

The model owner can compute the gradient of the model on the test set in plaintext. Because $L(\cdot)$ is twice differentiable, we have the empirical Hessian matrix associated with the training samples

$$\boldsymbol{H}_{\hat{\theta}} \coloneqq \frac{1}{|\mathcal{D}_{\text{tr}}|} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\text{tr}}} \nabla^2_\theta L(\boldsymbol{x}, y, \hat{\theta}). \tag{C.4}$$

This Hessian and its associative inverse can also be computed in plaintext.

Suppose we upweigh a sample, $(\boldsymbol{x}_0, y_0)$, by an infinitesimal amount ϵ, and study the effect of this perturbation on the resulting model parameters. The associated loss is thus formulated as $\epsilon L(\boldsymbol{x}_0, y_0, \theta) + \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\text{tr}}} L((\boldsymbol{x}, y, \theta)$. Training the new model till convergence to get new parameter θ^*, we can assume that the gradient of its loss is 0, or

$$\epsilon \nabla_\theta L(\boldsymbol{x}_0, y_0, \theta^*) + \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\text{tr}}} \nabla_\theta L(\boldsymbol{x}, y, \theta^*) = 0. \tag{C.5}$$

We write the left hand side as an function of the new parameters, where

$$f(\theta^*) \coloneqq \epsilon \nabla_\theta L(\boldsymbol{x}_0, y_0, \theta^*) + \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\text{tr}}} \nabla_\theta L(\boldsymbol{x}, y, \theta^*). \tag{C.6}$$

We wish to find a relation between the parameters before and after the perturbation. To that end, denote the parameter difference $\Delta_\theta \coloneqq \theta^* - \hat{\theta}$. The goal is to find a closed expression for Δ_θ, given the approximation that $f(\theta^*) \approx 0$.

As $\epsilon \to 0$, the new training set is just the original training data, or $\mathcal{D} \to \mathcal{D}_{\text{tr}}$. The resulting model (from the non-perturbation), as we know, is optimal at $\hat{\theta}$. Therefore, the first two terms in the Taylor expansion of $f(\theta^*)$ around $\Delta_\theta = 0$ is $f(\theta^*) \approx f(\hat{\theta}) + f'(\hat{\theta}) \cdot \Delta_\theta$. We write

$$0 = f(\theta^*) \approx f(\hat{\theta}) + f'(\hat{\theta}) \cdot \Delta_\theta$$

Additionally, Equation C.6 gives us

$$f(\hat{\theta}) := \epsilon \nabla_\theta L(\boldsymbol{x}_0, y_0, \hat{\theta}) + \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\text{tr}}} \nabla_\theta L(\boldsymbol{x}, y, \hat{\theta}).$$

We thus obtain the approximation

$$\sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\text{tr}}} \nabla_\theta L(\boldsymbol{x}, y, \hat{\theta}) + \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\text{tr}}} \nabla^2_\theta L(\boldsymbol{x}, y, \hat{\theta}) \cdot \Delta_\theta + \epsilon \nabla_\theta L(\boldsymbol{x}_0, y_0, \hat{\theta}) + \epsilon \nabla^2_\theta L(\boldsymbol{x}_0, y_0, \hat{\theta}) \cdot \Delta_\theta \approx 0. \tag{C.7}$$

Recall that on the original seed dataset $\mathcal{D}_{\text{tr}}$, parameter $\hat{\theta}$ is optimal, so $\sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\text{tr}}} \nabla_\theta L((\boldsymbol{x}, y, \hat{\theta}) = 0$. This allows for a simplification:

$$\sum_{(\boldsymbol{x},y)\in\mathcal{D}_{\text{tr}}} \nabla^2_\theta L(\boldsymbol{x}, y, \hat{\theta}) \cdot \Delta_\theta + \epsilon \nabla_\theta L(\boldsymbol{x}_0, y_0, \hat{\theta}) + \epsilon \nabla^2_\theta L(\boldsymbol{x}_0, y_0, \hat{\theta}) \cdot \Delta_\theta \approx 0. \tag{C.8}$$

Solving for Δ_θ approximately requires taking the inverse of the empirical Hessian (see discussion below).

$$\left(|\mathcal{D}_{\text{tr}}| \boldsymbol{H}_{\hat{\theta}} + \epsilon \nabla^2_\theta L(\boldsymbol{x}_0, y_0, \hat{\theta})\right) \cdot \Delta_\theta = -\epsilon \nabla_\theta L(\boldsymbol{x}_0, y_0, \hat{\theta}). \tag{C.9}$$

Multiply both sides with the scaled Hessian inverse

$$\left(1 + \frac{\epsilon}{|\mathcal{D}_{\text{tr}}|} \boldsymbol{H}^{-1}_{\hat{\theta}} \nabla^2_\theta L(\boldsymbol{x}_0, y_0, \hat{\theta})\right) \cdot \Delta_\theta = -\frac{\epsilon}{|\mathcal{D}_{\text{tr}}|} \boldsymbol{H}^{-1}_{\hat{\theta}} \nabla_\theta L(\boldsymbol{x}_0, y, \hat{\theta}). \tag{C.10}$$

Drop the term $\epsilon \nabla^2_\theta L(\boldsymbol{x}_0, y_0, \hat{\theta})$ (see discussion notes), and take the derivate of both sides with respect to ϵ and write

$$\frac{\delta \Delta_\theta}{\delta \epsilon} = -\frac{1}{|\mathcal{D}_{\text{tr}}|} \boldsymbol{H}^{-1}_{\hat{\theta}} \nabla_\theta L(\boldsymbol{x}_0, y_0, \hat{\theta}). \tag{C.11}$$

We thus obtain our influence formulation or $\mathcal{I}(\boldsymbol{x}, y) = -\boldsymbol{H}^{-1}_{\hat{\theta}} \nabla_\theta L(\boldsymbol{x}, y, \hat{\theta})$. Forward influence refers to its application on unseen data (see discussion for more). Applying it to evaluate the change of loss given a

particular dataset $\mathcal{D}_\text{a}$ gives us the key appraisal component:

$$\mathcal{I}(\mathcal{D}_\text{a}) = -\boldsymbol{H}_{\hat{\theta}}^{-1} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_\text{a}} \nabla_\theta L(\boldsymbol{x}, y, \hat{\theta}), \tag{C.12}$$

before scaling (by the cardinality of the datasets) to approximate $(\hat{\theta} - \theta^*)$ in Equation C.3.

Influence Discussion First, strong convexity is usually assumed [165], so that the Hessian matrix is positive definite. This is a stronger assumption than necessary, as only the empirical Hessian with respect to the combined dataset needs to be positive-definite. In practice, we assume convexity and use regularization when inverting the Hessian [1], so the method can be potentially applied to problems when the Hessian is not positive definite.

Machine learning literature typically assumes $(\boldsymbol{x}_0, y_0)$ to be part of the training data when applying influence functions. Here we are using the numerical form of the result, but applying the extrapolation to new data $\mathcal{D}_\text{a}$, hence it is referred to as a forward influence. A mismatched data construction is standard technique in the construction of influence functions [129; 113]. The impact of this mismatch is studied in our experiments.

Thirdly, the Taylor Expansions' validity likely matters little in application, but it is worth mentioning that the loss function is preferred to be second-order smooth. The truncation error is studied in Basu et al. [27] for its interaction with non-convexity.

Additionally, dropping the term $\epsilon\nabla^2 L(\boldsymbol{x}_0, y, \hat{\theta})$ from the first order expansion is effectively approximating the gradient on the new data point with the gradient of the previous model, which may not be bounded. This approximation is also present in the usual influence definition.

[1] Alternatively, a pracitioner may implement the numerical function to avoid inverting the Hessian altogether. See Gauss-Newton approximation [204].

D | Dataset-to-Dataset Evaluations

D.1 Methodological Details

Datasets We use two datasets: 1. eICU Collaborative Research Dataset [244] contains over 200,000 admissions from 208 hospitals across the United States. Following the data cleaning and exclusion criteria outlined by [312] and [267], We select 12 hospitals with the highest number of patient visits (each with at least 2000 patients) as our entire set of hospitals **H**.

In Chapter 5, each strategy would compute with a max $K = 3000$ records, as the total available data per hospital.

2. To evaluate broader applicability, we replicate a portion of our experiments on the Folktables [83] dataset on income prediction is additionally used, which provides rich demographic and socioeconomic information on individuals across U.S. states. We predict whether an individual's annual income exceeds $50,000.

Data Treatment For each strategy, the same records available per hospital are used, with $K = 3000$. Performance – AUC_o, AUC_T – uses 400 samples (unless otherwise noted) [1] The AUC change, δ_i or δ_T, comes from 1. combining 1500 random samples from each selected dataset and 2. combine it with 1500 samples from $\mathcal{D}_o$, and 3. subtracting the baseline model's AUC[2]. The downstream task is the 24-hour mortality prediction.Strategy comparisons take 1500 samples. We simulate the problem setup for each hospital with the 24-hour mortality prediction task. Unless otherwise specified, all experiments follow the training and evaluation protocol in Yet Another ICU Benchmark [312], using 1,500 training samples and 400 test samples per hospital. For the data combination experiments that compute AUC change δ_i or δ_T, to match [267], we take 1500 random samples from each selected dataset and combine it with 1500 samples from $\mathcal{D}_o$. To match [267], each hospital experiment was carried out using 5-fold cross-validation, repeated 5 times with different random seeds. AUC results are averaged first across folds, then across repetitions.

The strategy comparisons described are implemented using 1500 samples for our training set and 400 samples for our test set per hospital for all of our experiments unless otherwise noted. This follows training and evaluation protocols in Yet Another ICU Benchmark [312].

[1] This follows training and evaluation protocols in Yet Another ICU Benchmark [312].

[2] The samples are fixed across all experiments, the sample numbers are chosen to match the setup in [267].

k	ρ	**p-value**
3	-0.063	4.70e-01
30	-0.082	3.47e-01
300	-0.059	5.00e-01
3000	-0.184	**3.47e-02**

k	ρ	**p-value**
3	-0.158	7.02e-02
30	0.167	5.60e-02
300	-0.097	2.70e-01
3000	-0.284	**9.47e-04**

Table D.1: ρ and p-value between AUC drop and plaintext KL using k samples using SGD (left) and LBFGS (right).

D.2 Correlation with downstream performance

On Table D.1, we report the Pearson correlations between $\pi_s(k = K)$ for $k \in \{3, 30, 300, 3000\}$ and δ_i. On Table 5.1, we report the Pearson correlations between different strategies and δ_i.

D.3 Hyperparameter Tuning

We obtain Score(X,Y) by training a Logistic Regression model using SGD. We find that SGD requires hyperparameter tuning in order to perform well when evaluated on Brier Score Loss. Optuna is used to perform hyperparameters search. The hyperparameters we use for plaintext scores are:

1. learning rate: 0.0795
2. patience: 2
3. tolerance: 0.000117
4. momentum: 0.886
5. weight decay: 1.81e-9
6. dampening: .0545

The hyperparameters for the encrypted model:

1. learning rate: 0.0974
2. patience: 5
3. tolerance: 0.000132
4. momentum: 0.907
5. weight decay: 8.14e-7
6. dampening: .0545

Hospital	$\rho(\text{KL}_{\mathcal{X}}, \text{SecureKL}_{\mathcal{X}})$	p-value	$\rho(\text{KL}_{\mathcal{XY}}, \text{SecureKL}_{\mathcal{XY}})$	p-value
73	0.945	1.118e-05	1.000	0.0
264	0.973	5.142e-07	0.945	1.118e-05
420	0.982	8.403e-08	0.991	3.763e-09
243	0.973	5.142e-07	0.909	1.056e-04
338	0.973	5.142e-07	0.982	8.403e-08
443	0.964	1.852e-06	0.882	3.302e-04
199	0.991	3.763e-09	0.973	5.142e-07
458	0.873	4.546e-04	0.964	1.852e-06
300	0.455	1.601e-01	0.691	1.857e-02
188	0.718	1.280e-02	0.864	6.117e-04
252	0.873	4.546e-04	0.809	2.559e-03
167	0.764	6.233e-03	0.891	2.335e-04

Table D.2: Spearman Correlations ρ for encrypted (in CrypTen) and plaintext (in PyTorch) KL-based methods

D.4 Correlations between Encrypted Scores and Plaintext Scores

On Table D.2, we measure the Spearman correlations between $\text{KL}_{\mathcal{X}}$ and $\text{SecureKL}_{\mathcal{X}}$, and between $\text{KL}_{\mathcal{XY}}$ and $\text{SecureKL}_{\mathcal{XY}}$ for all hospitals. We find that all hospitals have statistically significant correlations with the exception of hospital 300's $\rho(\text{KL}_{\mathcal{X}},\text{SecureKL}_{\mathcal{X}})$

D.5 Performance

Computing SecureKL$_{\mathcal{XY}}$ scores for 144 hospital pairs, each with at most 3000 samples, took 317 seconds, which is 6.6X longer than KL$_{\mathcal{XY}}$.

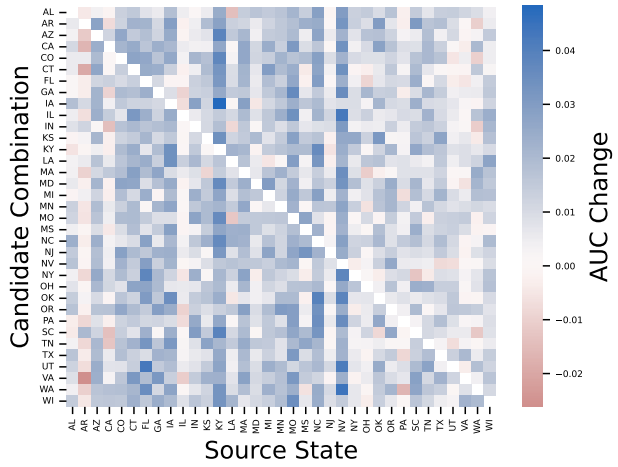


Figure D.1: In Folktables [83], combining with random state leads to worse income prediction in 23 out of 35 states.

D.6 Folktables Experiments

Dataset We use the Folktables dataset, a benchmark derived from the U.S. Census American Community Survey (ACS), which provides rich demographic and socioeconomic information on individuals across U.S. states. We focus on income prediction, which classifies whether an individual's annual income exceeds $50,000$ based on 10 features. We focus on states with $12,000$ individuals, for a total of 35 states.

For our baseline, we train a XGB model on $4,000$ training samples and 400 test samples per state. For the data combination experiments, we combine 4000 randomly selected samples from each selected dataset to combine with the 4000 samples in D_o.

For eICU data, used Score$(\cdot)$ to estimate data density due to its high dimensionality. In contrast, Folktables only contains 10 features, allowing us to compute KL divergence directly using kernel density estimation. We apply a Gaussian kernel to the top 3 scaled principal components.

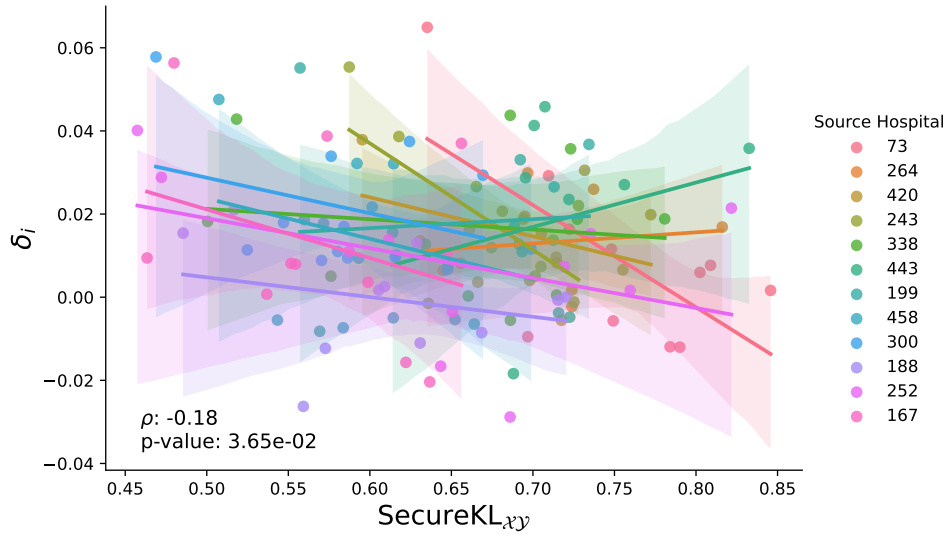


Figure D.2: SecureKL: Overall Correctness. Rank correlation between SecureKL output and ground truth AUC change, δ_i, from acquiring 1 additional dataset for a given source hospital H_o. We propose selecting data partner ranked by our secure system under SecureKL$_{\mathcal{XY}}$ score to reliably reduce downstream AUC downstream task. ($|\mathbf{H}| = 12$ hospitals; colored by source.)

E | HEalth

E.1 Privacy Challenges for Auditing Healthcare Data

Regulatory Agencies Mandate Fairness. The Agency for Healthcare Research and Quality (AHRQ) is working to produce evidence to make healthcare safer, more accessible and more affordable. One of their goals is to make sure that the evidence is understood and used. The Health Resources and Services Administration (HRSA) provides health care to people who are geographically isolated, economically or medically vulnerable.

Hospital Decisions Require Protected Attributes, Inherently Sensitive. US federal law protects 9 characteristics: race, religion, national origin, age, sex (incl. sexual orientation and gender identity), pregnancy, familial status, disability status, veteran status, genetic information. Not all of this information is collected by hospitals, nor would all hospitals have the same categorizations. However, most record some sensitive information that is critical for decision-making. For example, race, age, sex, pregnancy. This makes the records of which the agencies need to audit inherently sensitive.

Auditing in Non-encrypted Forms Undermines Privacy. In theory, using synthetic or anonymized data, or differentially-aggregated data, can be alternatives to encryption. Yet privacy wise, they are prone to linkage attacks that connect the obscured data with clear data, in order to de- or re-anonymize individuals. In the medical domain, decrypting at set intervals for trusted regulatory bodies (or otherwise curating data) brings about significant operational overhead, not to mention significant social cost for degrading encryption (Figure 6.6).

E.2 Protocol Details

The KeyGen function assumes a full-party variant of Threshold FHE, where each party can destroy data by destroying their key. An alternative implementation would use a threshold, where only some parties are needed to decrypt.

Employing a variant of BFS (star topology) with a full-party threshold.

The scheme comes with a qs (scaling factor $\lfloor qs/t \rfloor$). Plain text modulus $t << qs$.

For hospital i, generate a fresh s, e, and construct a public key.

$$\text{pk}_i = (a, as_i + e_i)$$

where

$$s_i, e_i \sim \chi(\vec{0}, \sigma^2)\sigma = 3.2$$

To generate a shared public key, we assume that all parties have a

1. Shared random polynomial is broadcasted $\rightarrow a$ 2. Each has a secret key, which have to go together, s_i's. This is equivalent to BFV with $s \leftarrow \sum s_i$, and $e \leftarrow \sum e_i$.

$$\mathcal{R}_{\text{II}} = \mathbb{Z}_q[X]/(x^n + 1)$$

3. Validation Phase (optional): sum of all pk's. Everyone broadcasts it back to everyone else. 4. Decryption

$$\text{Ciphertext ct} = (a, as + \Delta m + e) \bmod R_q \text{where } \Delta = \lfloor qs/t \rfloor, m \in R_t$$

We want to apply BFV's Decoding

$$(\text{ct}, \sum s_i)$$

Each hospital outputs decryption share

$$d_i = \mathrm{ct}[1] - k(as_i + \tilde{e}_i)$$

The sum is therefore

$$\sum d_i = k(\mathrm{ct}[1]) - k(a \sum s_i + \sum \tilde{e}_i) \approx km + 2k(\bar{e})$$

The rest follows rotation key and multiplication key generation.

E.3 Data Model Snippet

```
{
    patient_id: 0
    timestamp: 0
    age: 37
    gender: M
    race: Caucasian
    medical_history:
        'severe abdominal pain'
    occupancy_at_admission: 70%
    reason_for_visit: high fever
    .
    .
    .
    decision : admit
}
```

Figure E.1: A Clear-text Example for Hospital Records.

E.4 Comparing Alternative Auditing Setups

We discuss other setups to achieve auditing fairness at hospitals. **Existing Practice.** In Figure E.2, the hospital would compute $\mathcal{F}_{\text{fairness}}$ on its premise, and report the relevant statistics to the regulators. This is currently part of the existing paradigm for algorithmic fairness. Our solution allows a third party, such as the government regulator, to compute $\mathcal{F}_{\text{fairness}}$ independently of the hospital, and from raw data.

Output-private Audits Are Approximate. In Figure E.3, auditors use released data or model to construct their auditing, typically approximating $\mathcal{F}$ from "fuzzy" versions of the patient data or model predictions used in practice, meaning that auditors are approximating the metric to audit. This limits the precision of external auditing, which is crucial for independent verification.

We summarize challenges with output-private techniques, already laid out in Section 3.6.2 and Section 3.2. 1) compromising data accuracy reduces its usefulness, 2) re-identification risks from released data and models, when outside information is used, and 3) meaningfully setting up the infrastructure, including meaningful privacy parameter selection is nontrivial, making the method in practice difficult to adopt for small organizations.

We clarify data usefulness challenge in the fairness auditing setup. [281] finds that the more privacy-preserving the models are, the less useful the data becomes, which disempowers auditing.

Yet, ensuring hospitals to use the same model as they released is detrimental to patient outcome, as protecting privacy from the output often changes the original model's behavior; in healthcare domain, differentially private models exhibit degraded performance, in the so-called Privacy-Utility Trade-off [281]. Our solution does not make this trade-off, as data is kept private by default, and the audit uses exact raw data, thus matching the fairness output as Setup E.2.

The second challenge of privacy is also notable. When the approximation is on the raw data, data anonymization techniques like k-anonymity [282] risk re-identification. In contrast, an input-private method like encryption, does not have re-identification risk, as no individual data is released in the first place.

Undermining encryption is undesirable for all. When a hospital invites the auditor inside the

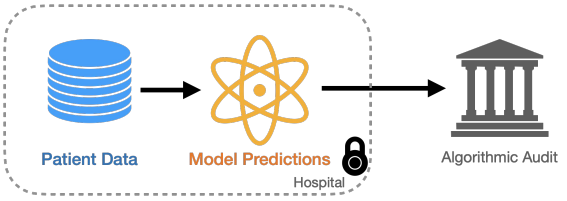


Figure E.2: Hospital self-reports final metric. This does not constitute as a third party auditing.

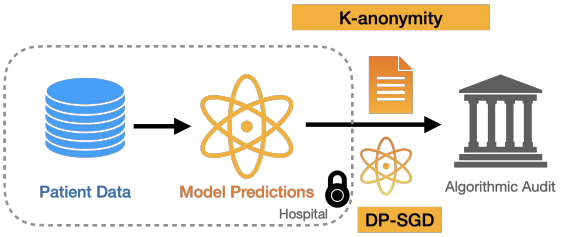
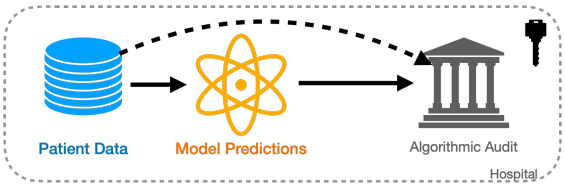


Figure E.3: Hospital uses privacy-preserving data or model releases. From the de-sensitized data, the auditor approximates $\mathcal{F}$. This presupposes the release of data.



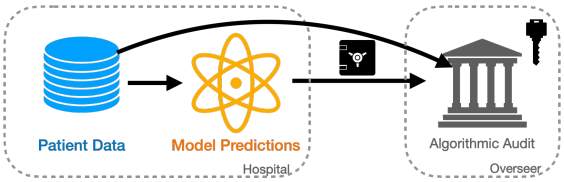


Figure E.4: Auditor computes $\mathcal{F}$ from raw data, including sensitive patient data. This level of data access exceeds what is necessary for the sole purpose of fairness metrics. **Left: Auditor goes inside the hospital to audit.** Frequent auditor visits may result in added operational costs for the hospital. **Right: The hospital sends encrypted data and a key to decrypt.** Auditor decrypts encrypted data. This setup introduces novel security risks. Security of key transmission and the data security at the auditing site becomes a concern. When auditing multiple hospitals, the auditor also becomes a central repository of sensitive data (and keys).

compound, and allows the auditor to see all raw data. As illustrated in Figure ??, a fairness audit is feasible without violating data privacy regulations, assuming the government auditor can be trusted. However, accommodating auditors adds operational friction to hospitals.

If the data is sent encrypted, and then a key is also sent to the auditor, the auditor can then decrypt all raw data on their side, avoiding transporting unencrypted data. Yet this approach centralizes data security risks to the auditor, who may now hold many keys to many hospitals' raw data.

Nevertheless, neither setup considers what the audit *ought to access* [228]. We ask, should algorithmic fairness audit require access to *all* the hospital records? While most group fairness metrics necessitates protected attributes as input, having the auditing party access all raw data violates the data minimization principle of privacy-by-design. Undermining encryption is also against the spirit of data protection regulations.

Bibliography

[1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

[2] Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. (2016). Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318.

[3] Acemoglu, D., Makhdoumi, A., Malekian, A., and Ozdaglar, A. (2022). Too much data: Prices and inefficiencies in data markets. *American Economic Journal: Microeconomics*, 14(4):218–256.

[4] Agarap, A. F. M. (2018). On breast cancer detection: An application of machine learning algorithms on the wisconsin diagnostic dataset. In *Proceedings of the 2nd International Conference on Machine Learning and Soft Computing*, ICMLSC '18, pages 5–9, New York, NY, USA. ACM.

[5] Agarwal, N., Bullins, B., and Hazan, E. (2017). Second-order stochastic optimization for machine learning in linear time. *Journal of Machine Learning Research*, 18(116):1–40.

[6] Akre, P. D. and Pacharaney, U. (2025). A comprehensive review of mojo: A high-performance programming language. In *2025 6th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI)*, pages 861–867. IEEE.

[7] Al Badawi, A., Bates, J., Bergamaschi, F., Cousins, D. B., Erabelli, S., Genise, N., Halevi, S., Hunt, H., Kim, A., Lee, Y., et al. (2022a). Openfhe: Open-source fully homomorphic encryption library. In *proceedings of the 10th workshop on encrypted computing & applied homomorphic cryptography*, pages 53–63.

[8] Al Badawi, A., Bates, J., Bergamaschi, F., Cousins, D. B., Erabelli, S., Genise, N., Halevi, S., Hunt, H., Kim, A., Lee, Y., Liu, Z., Micciancio, D., Quah, I., Polyakov, Y., R.V., S., Rohloff, K., Saylor, J., Suponitsky, D., Triplett, M., Vaikuntanathan, V., and Zucca, V. (2022b). Openfhe: Open-source fully homomorphic encryption library. In *Proceedings of the 10th Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, WAHC'22, pages 53–63, New York, NY, USA. Association for Computing Machinery.

[9] Albrecht, M., Chase, M., Chen, H., Ding, J., Goldwasser, S., Gorbunov, S., Halevi, S., Hoffstein, J., Laine, K., Lauter, K., et al. (2021). Homomorphic encryption standard. *Protecting privacy through homomorphic encryption*, pages 31–62.

[10] Alsentzer, E., Murphy, J., Boag, W., Weng, W.-H., Jindi, D., Naumann, T., and McDermott, M. (2019). Publicly available clinical bert embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78.

[11] Anthropic (2023). Frontier model security.

[12] Aono, Y., Hayashi, T., Wang, L., Moriai, S., et al. (2017). Privacy-preserving deep learning via additively homomorphic encryption. *IEEE transactions on information forensics and security*, 13(5):1333–1345.

[13] Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. (2019). Invariant risk minimization. *arXiv preprint arXiv:1907.02893*.

[14] Asharov, G., Jain, A., López-Alt, A., Tromer, E., Vaikuntanathan, V., and Wichs, D. (2012). Multiparty computation with low communication, computation and interaction via threshold fhe. In *Advances in Cryptology–EUROCRYPT 2012: 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings 31*, pages 483–501. Springer.

[15] Assefa, S. A., Dervovic, D., Mahfouz, M., Tillman, R. E., Reddy, P., and Veloso, M. (2020). Generating synthetic data in finance: opportunities, challenges and pitfalls. In *Proceedings of the First ACM International Conference on AI in Finance*, pages 1–8.

[16] Avent, B., Korolova, A., Zeber, D., Hovden, T., and Livshits, B. (2017). {BLENDER}: Enabling local search with a hybrid differential privacy model. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 747–764.

[17] Azar, P. D., Goldwasser, S., and Park, S. (2016). How to incentivize data-driven collaboration among competing parties. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, pages 213–225.

[18] Azcoitia, S. A. and Laoutaris, N. (2020). Try before you buy: A practical data purchasing algorithm for real-world data marketplaces. *arXiv preprint arXiv:2012.08874*.

[19] Azcoitia, S. A. and Laoutaris, N. (2022). A survey of data marketplaces and their business models. *ACM SIGMOD Record*, 51(3):18–29.

[20] Azcoitia, S. A., Paraschiv, M., and Laoutaris, N. (2020). Computing the relative value of spatio-temporal data in wholesale and retail data marketplaces. *arXiv preprint arXiv:2002.11193*.

[21] Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., and Shmatikov, V. (2020). How to backdoor federated learning. In *International conference on artificial intelligence and statistics*, pages 2938–2948. PMLR.

[22] Balsa, E., Nissenbaum, H., and Park, S. (2022). Cryptography, trust and privacy: It's complicated. In *Proceedings of the 2022 Symposium on Computer Science and Law*, pages 167–179.

[23] Baowaly, M. K., Lin, C.-C., Liu, C.-L., and Chen, K.-T. (2019). Synthesizing electronic health records using improved generative adversarial networks. *Journal of the American Medical Informatics Association*, 26(3):228–241.

[24] Barocas, S., Hardt, M., and Narayanan, A. (2023). *Fairness and machine learning: Limitations and opportunities*. MIT press.

[25] Bassily, R., Smith, A., and Thakurta, A. (2014). Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th annual symposium on foundations of computer science*, pages 464–473. IEEE.

[26] Basu, S., Pope, P., and Feizi, S. (2020a). Influence functions in deep learning are fragile. *arXiv preprint arXiv:2006.14651*.

[27] Basu, S., You, X., and Feizi, S. (2020b). On second-order group influence functions for black-box predictions. In *International Conference on Machine Learning*, pages 715–724. PMLR.

[28] Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the royal statistical society series b-methodological*, 57:289–300.

[29] Blatt, M., Gusev, A., Polyakov, Y., and Goldwasser, S. (2020). Secure large-scale genome-wide association studies using homomorphic encryption. *Proceedings of the National Academy of Sciences*, 117(21):11608–11613.

[30] Blumberg, A. J. and Eckersley, P. (2009). On locational privacy, and how to avoid losing it forever. *Electronic frontier foundation*, 10(11):1–7.

[31] Boenisch, F., Dziedzic, A., Schuster, R., Shamsabadi, A. S., Shumailov, I., and Papernot, N. (2023). When the curious abandon honesty: Federated learning is not private. In *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*, pages 175–199. IEEE.

[32] Boltzmann, L. (1868). Studien uber das gleichgewicht der lebenden kraft. *Wissenschafiliche Abhandlungen*, 1:49–96.

[33] Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konecnỳ, J., Mazzocchi, S., McMahan, H. B., et al. (2019). Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*.

[34] Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A.,

and Seth, K. (2017). Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191.

[35] Borgesius, F. Z., Gray, J., and Van Eechoud, M. (2015). Open data, privacy, and fair information principles: Towards a balancing framework. *Berkeley Technology Law Journal*, 30(3):2073–2131.

[36] Bost, R., Popa, R. A., Tu, S., and Goldwasser, S. (2014). Machine learning classification over encrypted data. *Cryptology ePrint Archive*.

[37] Bottou, L. and Bousquet, O. (2007). The tradeoffs of large scale learning. *Advances in neural information processing systems*, 20.

[38] Boulila, W., Ammar, A., Benjdira, B., and Koubaa, A. (2022). Securing the classification of covid-19 in chest x-ray images: A privacy-preserving deep learning approach. In *2022 2nd International Conference of Smart Systems and Emerging Technologies (SMARTTECH)*, pages 220–225. IEEE.

[39] Bourtoule, L., Chandrasekaran, V., Choquette-Choo, C., Jia, H., Travers, A., Zhang, B., Lie, D., and Papernot, N. (2021). Machine unlearning. In *Proceedings of the 42nd IEEE Symposium on Security and Privacy*.

[40] Bradley, V. C., Kuriwaki, S., Isakov, M., Sejdinovic, D., Meng, X.-L., and Flaxman, S. (2021). Unrepresentative big surveys significantly overestimated us vaccine uptake. *Nature*, 600(7890):695–700.

[41] Brakerski, Z. (2012). Fully homomorphic encryption without modulus switching from classical gapsvp. In *Annual cryptology conference*, pages 868–886. Springer.

[42] Brakerski, Z., Gentry, C., and Vaikuntanathan, V. (2014). (leveled) fully homomorphic encryption without bootstrapping. *ACM Transactions on Computation Theory (TOCT)*, 6(3):1–36.

[43] Brandao, L. and Peralta, R. (2023). Nist first call for multi-party threshold schemes. *doi*, 10:6028.

[44] Bresson, E., Catalano, D., Fazio, N., Nicolosi, A., and Yung, M. (2006). Output privacy in secure multiparty computation. *Proc. YACC*.

[45] Brown, H., Lee, K., Mireshghallah, F., Shokri, R., and Tramèr, F. (2022). What does it mean for a language model to preserve privacy? In *Proceedings of the 2022 ACM conference on fairness, accountability, and transparency*, pages 2280–2292.

[46] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

[47] Brozynski, M. T. and Leibowicz, B. D. (2022). A multi-level optimization model of infrastructure-dependent technology adoption: Overcoming the chicken-and-egg problem. *European Journal of Operational Research*, 300(2):755–770.

[48] Buolamwini, J. and Gebru, T. (2018). Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91. PMLR.

[49] Caillaud, B. and Jullien, B. (2003). Chicken & egg: Competition among intermediation service providers. *RAND journal of Economics*, pages 309–328.

[50] Cavoukian, A. et al. (2009). Privacy by design: The 7 foundational principles. *Information and privacy commissioner of Ontario, Canada*, 5(2009):12.

[51] Center, P. R. (2020). Most americans support right to have some personal info removed from online searches.

[52] Center, P. R. (2023). Growing public concern about the role of artificial intelligence in daily life.

[53] Centers for Medicare & Medicaid Services (1996). The Health Insurance Portability and Accountability Act of 1996 (HIPAA). Online at http://www.cms.hhs.gov/hipaa/.

[54] Centers for Medicare & Medicaid Services (2025). Limited data set (lds) files | cms.

[55] Chakraborty, N., Sharma, A., Dutta, J., and Kumar, H. D. (2024). Privacy-preserving data quality assessment for time-series iot sensors. In *2024 IEEE International Conference on Internet of Things and Intelligence Systems (IoTaIS)*, pages 51–57. IEEE.

[56] Chandran, N., Gupta, D., Rastogi, A., Sharma, R., and Tripathi, S. (2017). Ezpc: Programmable, efficient, and scalable secure two-party computation. *IACR Cryptol. ePrint Arch.*, 2017:1109.

[57] Chase, M., Chen, H., Ding, J., Goldwasser, S., Gorbunov, S., Hoffstein, J., Lauter, K., Lokam, S., Moody, D., Morrison, T., et al. (2017). Security of homomorphic encryption. *HomomorphicEncryption. org, Redmond WA, Tech. Rep.*

[58] Chaudhuri, K., Ahuja, K., Arjovsky, M., and Lopez-Paz, D. (2023). Why does throwing away data improve worst-group error? In *International Conference on Machine Learning*, pages 4144–4188. PMLR.

[59] Chaum, D., Damgård, I. B., and Van de Graaf, J. (1988). Multiparty computations ensuring privacy of each party's input and correctness of the result. In *Advances in Cryptology—CRYPTO'87: Proceedings 7*, pages 87–119. Springer.

[60] Chen, H., Laine, K., and Player, R. (2017). Simple encrypted arithmetic library-seal v2. 1. In *Financial Cryptography and Data Security: FC 2017 International Workshops, WAHC, BITCOIN, VOTING, WTSC, and TA, Sliema, Malta, April 7, 2017, Revised Selected Papers 21*, pages 3–18. Springer.

[61] Chen, T., Moreau, T., Jiang, Z., Zheng, L., Yan, E., Shen, H., Cowan, M., Wang, L., Hu, Y., Ceze, L., et al. (2018). {TVM}: An automated {End-to-End} optimizing compiler for deep learning. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*, pages 578–594.

[62] Cheon, J. H., Kim, A., Kim, M., and Song, Y. (2017). Homomorphic encryption for arithmetic of approximate numbers. In *Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I 23*, pages 409–437. Springer.

[63] Cheon, J. H., Kim, D., and Kim, D. (2020). Efficient homomorphic comparison methods with optimal complexity. In *Advances in Cryptology–ASIACRYPT 2020: 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7–11, 2020, Proceedings, Part II 26*, pages 221–256. Springer.

[64] Chillotti, I., Gama, N., Georgieva, M., and Izabachène, M. (2020). Tfhe: fast fully homomorphic encryption over the torus. *Journal of Cryptology*, 33(1):34–91.

[65] Chiruvella, V., Guddati, A. K., et al. (2021). Ethical issues in patient data ownership. *Interactive journal of medical research*, 10(2):e22269.

[66] Cho, H., Froelicher, D., Chen, J., Edupalli, M., Pyrgelis, A., Troncoso-Pastoriza, J. R., Hubaux, J.-P., and Berger, B. (2025). Secure and federated genome-wide association studies for biobank-scale datasets. *Nature Genetics*, pages 1–6.

[67] Clifton, C., Kantarcioglu, M., and Vaidya, J. (2002). Defining privacy for data mining. In *National science foundation workshop on next generation data mining*, volume 1, page 1. Citeseer.

[68] Cohen, J. E. (2012). What privacy is for. *Harv. L. Rev.*, 126:1904.

[69] Compton, R., Zhang, L., Puli, A., and Ranganath, R. (2023). When more is less: Incorporating additional datasets can hurt performance by introducing spurious correlations. In Deshpande, K., Fiterau, M., Joshi, S., Lipton, Z., Ranganath, R., Urteaga, I., and Yeung, S., editors, *Proceedings of the 8th Machine Learning for Healthcare Conference*, volume 219 of *Proceedings of Machine Learning Research*, pages 110–127. PMLR.

[70] Congress, U. (1964). Title vii of the civil rights act of 1964. 42 U.S.C. § 2000e et seq.

[71] Cook, R. D. (1977). Detection of influential observation in linear regression. *Technometrics*, 19(1):15–18.

[72] Cook, R. D. and Weisberg, S. (1982). *Residuals and influence in regression*. New York: Chapman and Hall.

[73] Cooper, A. F. and Grimmelmann, J. (2025). The files are in the computer: Copyright, memorization, and generative ai.

[74] Cormode, G., Jha, S., Kulkarni, T., Li, N., Srivastava, D., and Wang, T. (2018). Privacy at scale: Local differential privacy in practice. In *Proceedings of the 2018 International Conference on Management of Data*, pages 1655–1658.

[75] Costanza-Chock, S., Raji, I. D., and Buolamwini, J. (2022). Who audits the auditors? recommendations from a field scan of the algorithmic auditing ecosystem. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1571–1583.

[76] Daneshjou, R., Vodrahalli, K., Novoa, R. A., Jenkins, M., Liang, W., Rotemberg, V., Ko, J., Swetter, S. M., Bailey, E. E., Gevaert, O., et al. (2022). Disparities in dermatology ai performance on a diverse, curated clinical image set. *Science advances*, 8(31):eabq6147.

[77] Das, S. and Kramer, A. (2013). Self-censorship on facebook. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 7, pages 120–127.

[78] de Castro, L., Hales, E., and Xu, M. (2021). Health: Privately computing on shared healthcare data. *Protecting Privacy through Homomorphic Encryption*, pages 157–162.

[79] De Michell, G. and Gupta, R. K. (1997). Hardware/software co-design. *Proceedings of the IEEE*, 85(3):349–365.

[80] DeepSeek-AI (2024). Deepseek-v3 technical report.

[81] Dennis Jr, J. E. and Schnabel, R. B. (1996). *Numerical methods for unconstrained optimization and nonlinear equations*. SIAM.

[82] Diffie, W. and Hellman, M. E. (2022). New directions in cryptography. In *Democratizing Cryptography: The Work of Whitfield Diffie and Martin Hellman*, pages 365–390.

[83] Ding, F., Hardt, M., Miller, J., and Schmidt, L. (2021). Retiring adult: New datasets for fair machine learning. In *Neural Information Processing Systems*.

[84] Dodge, Y. (2008). *Spearman Rank Correlation Coefficient*, pages 502–505. Springer New York, New York, NY.

[85] Dong, Y., Lu, W.-j., Zheng, Y., Wu, H., Zhao, D., Tan, J., Huang, Z., Hong, C., Wei, T., and Chen, W. (2023). Puma: Secure inference of llama-7b in five minutes. *arXiv preprint arXiv:2307.12533*.

[86] Drechsler, J. (2011). *Synthetic datasets for statistical disclosure control: theory and implementation*, volume 201. Springer Science & Business Media.

[87] Dua, D. and Graff, C. (2017). UCI machine learning repository.

[88] Duan, M., Liu, D., Ji, X., Wu, Y., Liang, L., Chen, X., Tan, Y., and Ren, A. (2021). Flexible clustered federated learning for client-level data distribution shift. *IEEE Transactions on Parallel and Distributed Systems*, 33(11):2661–2674.

[89] Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. (2024). The llama 3 herd of models. *CoRR*.

[90] Dwork, C. (2006). Differential privacy. In *International colloquium on automata, languages, and programming*, pages 1–12. Springer.

[91] Dwork, C. (2008). Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer.

[92] Dwork, C., Roth, A., et al. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407.

[93] Erlingsson, Ú., Pihur, V., and Korolova, A. (2014). Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 1054–1067.

[94] European Parliament and Council of the European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council.

[95] European Parliament and Council of the European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council.

[96] Fan, J. and Vercauteren, F. (2012). Somewhat practical fully homomorphic encryption. *Cryptology ePrint Archive*.

[97] Fang, M., Cao, X., Jia, J., and Gong, N. (2020). Local model poisoning attacks to {Byzantine-Robust} federated learning. In *29th USENIX security symposium (USENIX Security 20)*, pages 1605–1622.

[98] Feldman, V. and Zhang, C. (2020). What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems*, 33:2881–2891.

[99] for Developers, G. (2025). Fairness: Demographic parity | machine learning.

[100] Fowl, L., Geiping, J., Czaja, W., Goldblum, M., and Goldstein, T. (2021). Robbing the fed: Directly obtaining private data in federated learning with modified models. *arXiv preprint arXiv:2110.13057*.

[101] Frankle, J., Park, S., Shaar, D., Goldwasser, S., and Weitzner, D. (2018). Practical accountability of secret processes. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 657–674.

[102] Fredrikson, M., Jha, S., and Ristenpart, T. (2015). 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.

[103] Froelicher, D., Troncoso-Pastoriza, J. R., Pyrgelis, A., Sav, S., Sousa, J. S., Bossuat, J.-P., and Hubaux, J.-P. (2020a). Scalable privacy-preserving distributed learning. *arXiv preprint arXiv:2005.09532*.

[104] Froelicher, D., Troncoso-Pastoriza, J. R., Raisaro, J. L., Cuendet, M. A., Sousa, J. S., Cho, H., Berger, B., Fellay, J., and Hubaux, J.-P. (2021). Truly privacy-preserving federated analytics for precision medicine with multiparty homomorphic encryption. *Nature communications*, 12(1):5910.

[105] Froelicher, D., Troncoso-Pastoriza, J. R., Sousa, J. S., and Hubaux, J.-P. (2020b). Drynx: Decentralized, secure, verifiable system for statistical queries and machine learning on distributed datasets. *IEEE Transactions on Information Forensics and Security*, 15:3035–3050.

[106] Fuentes, K., Xu, M., and Chen, I. (2025). Privacy-preserving dataset combination. *arXiv preprint arXiv:2502.05765*.

[107] Geiping, J., Bauermeister, H., Dröge, H., and Moeller, M. (2020). Inverting gradients-how easy is it to break privacy in federated learning? *Advances in neural information processing systems*, 33:16937–16947.

[108] Gentry, C. (2009). Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 169–178.

[109] Gervasi, S. S., Chen, I. Y., Smith-McLallen, A., Sontag, D., Obermeyer, Z., Vennera, M., and Chawla, R. (2022). The potential for bias in machine learning and opportunities for health insurers to address it. *Health Affairs*, 41(2):212–218.

[110] Geyer, R. C., Klein, T., and Nabi, M. (2017). Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*.

[111] Ghorbani, A. and Zou, J. (2019). Data shapley: Equitable valuation of data for machine learning. In *International conference on machine learning*, pages 2242–2251. PMLR.

[112] Gilad-Bachrach, R., Dowlin, N., Laine, K., Lauter, K., Naehrig, M., and Wernsing, J. (2016). Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In *International conference on machine learning*, pages 201–210. PMLR.

[113] Giordano, R., Stephenson, W., Liu, R., Jordan, M., and Broderick, T. (2019). A swiss army infinitesimal jackknife. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1139–1147.

[114] Goldreich, O., Micali, S., and Wigderson, A. (2019). How to play any mental game, or a completeness theorem for protocols with honest majority. In *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*, pages 307–328.

[115] Goldwasser, S. and Micali, S. (2019). Probabilistic encryption & how to play mental poker keeping secret all partial information. In *Providing sound foundations for cryptography: on the work of Shafi Goldwasser and Silvio Micali*, pages 173–201.

[116] Golkar, S., Pettee, M., Eickenberg, M., Bietti, A., Cranmer, M., Krawezik, G., Lanusse, F., McCabe,

M., Ohana, R., Parker, L., Blancard, B. R.-S., Tesileanu, T., Cho, K., and Ho, S. (2024). xval: A continuous numerical tokenization for scientific language models.

[117] Goncalves, A., Ray, P., Soper, B., Stevens, J., Coyle, L., and Sales, A. P. (2020). Generation and evaluation of synthetic patient data. *BMC medical research methodology*, 20:1–40.

[118] Gonzales, A., Guruswamy, G., and Smith, S. R. (2023). Synthetic data in health care: A narrative review. *PLOS Digital Health*, 2(1):e0000082.

[119] Grangier, D., Katharopoulos, A., Ablin, P., and Hannun, A. (2024). Specialized language models with cheap inference from limited domain data. *arXiv preprint arXiv:2402.01093*.

[120] Grosse, R., Bae, J., Anil, C., Elhage, N., Tamkin, A., Tajdini, A., Steiner, B., Li, D., Durmus, E., Perez, E., et al. (2023). Studying large language model generalization with influence functions. *arXiv preprint arXiv:2308.03296*.

[121] Groves, L., Metcalf, J., Kennedy, A., Vecchione, B., and Strait, A. (2024). Auditing work: Exploring the new york city algorithmic bias audit regime. In *The 2024 ACM Conference on Fairness, Accountability, and Transparency*, pages 1107–1120.

[122] Grynbaum, M. M. and Mac, R. (2023). The times sues openai and microsoft over ai use of copyrighted work. *The New York Times*, 27.

[123] Guo, C., Goldstein, T., Hannun, A., and van der Maaten, L. (2020a). Certified data removal from machine learning models. In *International Conference on Machine Learning*.

[124] Guo, H., Rajani, N. F., Hase, P., Bansal, M., and Xiong, C. (2020b). Fastif: Scalable influence functions for efficient model interpretation and debugging. *arXiv preprint arXiv:2012.15781*.

[125] Gürses, S., Troncoso, C., and Diaz, C. (2015). Engineering privacy by design reloaded. In *Amsterdam Privacy Conference*, volume 21.

[126] Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., and Smith, N. A. (2020). Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360.

[127] Halevi, S. and Shoup, V. (2014). Algorithms in helib. In *Advances in Cryptology–CRYPTO 2014: 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I 34*, pages 554–571. Springer.

[128] Halevi, S. and Shoup, V. (2020). Design and implementation of helib: a homomorphic encryption library. *Cryptology ePrint Archive*.

[129] Hampel, F. R. (1974). The influence curve and its role in robust estimation. *Journal of the american statistical association*, 69(346):383–393.

[130] Hao, M., Li, H., Chen, H., Xing, P., Xu, G., and Zhang, T. (2022). Iron: Private inference on transformers. *Advances in neural information processing systems*, 35:15718–15731.

[131] Hartzog, W. (2021). What is privacy? that's the wrong question. *U. Chi. L. Rev.*, 88:1677.

[132] Hendrycks, D. and Gimpel, K. (2016). Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

[133] Henecka, W., K ögl, S., Sadeghi, A.-R., Schneider, T., and Wehrenberg, I. (2010). Tasty: tool for automating secure two-party computations. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 451–462.

[134] Hestness, J., Narang, S., Ardalani, N., Diamos, G. F., Jun, H., Kianinejad, H., Patwary, M. M. A., Yang, Y., and Zhou, Y. (2017). Deep learning scaling is predictable, empirically. *ArXiv*, abs/1712.00409.

[135] Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., de Las Casas, D., Hendricks, L. A., Welbl, J., Clark, A., et al. (2022). Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pages 30016–30030.

[136] Hooker, S. (2021). The hardware lottery. *Communications of the ACM*, 64(12):58–65.

[137] Howe, B., Stoyanovich, J., Ping, H., Herman, B., and Gee, M. (2017). Synthetic data for social good. *arXiv preprint arXiv:1710.08874*.

[138] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al. (2022). Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

[139] Hu, Y., Niu, D., Yang, J., and Zhou, S. (2019). Fdml: A collaborative machine learning framework for distributed features. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2232–2240.

[140] Huang, L., Dou, Y., Liu, Y., Wang, J., Chen, G., Zhang, X., and Wang, R. (2021). Toward a research framework to conceptualize data as a factor of production: The data marketplace perspective. *Fundamental Research*, 1(5):586–594.

[141] Huang, Z., Lu, W.-j., Hong, C., and Ding, J. (2022). Cheetah: Lean and fast secure {Two-Party} deep neural network inference. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 809–826.

[142] Hummel, P., Braun, M., and Dabrock, P. (2021). Own data? ethical reflections on data ownership. *Philosophy & Technology*, 34(3):545–572.

[143] Hurst, A., Lerer, A., Goucher, A. P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., et al. (2024). Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

[144] Ilyas, A., Park, S. M., Engstrom, L., Leclerc, G., and Madry, A. (2022). Datamodels: Predicting predictions from training data. *arXiv preprint arXiv:2202.00622*.

[145] Jagielski, M., Severi, G., Pousette Harger, N., and Oprea, A. (2021). Subpopulation data poisoning attacks. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 3104–3122.

[146] Ji, W., Yuan, W., Getzen, E., Cho, K., Jordan, M. I., Mei, S., Weston, J. E., Su, W. J., Xu, J., and Zhang, L. (2025). An overview of large language models for statisticians. *arXiv preprint arXiv:2502.17814*.

[147] Jia, R., Dao, D., Wang, B., Hubis, F. A., Hynes, N., Gürel, N. M., Li, B., Zhang, C., Song, D., and Spanos, C. J. (2019). Towards efficient data valuation based on the shapley value. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1167–1176. PMLR.

[148] Jiang, A. Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D. S., Casas, D. d. l., Hanna, E. B., Bressand, F., et al. (2024). Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

[149] Jiang, K., Liang, W., Zou, J. Y., and Kwon, Y. (2023). Opendataval: a unified benchmark for data valuation. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S., editors, *Advances in Neural Information Processing Systems*, volume 36, pages 28624–28647. Curran Associates, Inc.

[150] Jiao, C., Pan, Y., Xiao, E., Sheng, D., Jain, N., Zhao, H., Dasgupta, I., Ma, J. W., and Xiong, C. (2025). Date-lm: Benchmarking data attribution evaluation for large language models. *arXiv preprint arXiv:2507.09424*.

[151] Jordan, M. I. and Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260.

[152] Jordon, J., Jarrett, D., Saveliev, E., Yoon, J., Elbers, P., Thoral, P., Ercole, A., Zhang, C., Belgrave, D., and van der Schaar, M. (2021). Hide-and-seek privacy challenge: Synthetic data generation vs. patient re-identification. In *NeurIPS 2020 Competition and Demonstration Track*, pages 206–215. PMLR.

[153] Jordon, J., Yoon, J., and Van Der Schaar, M. (2018). Pate-gan: Generating synthetic data with differential privacy guarantees. In *International conference on learning representations*.

[154] Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. (2021). Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2):1–210.

[155] Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. (2020). Scaling laws for neural language models. *ArXiv*, abs/2001.08361.

[156] Katz, M. L. and Shapiro, C. (1986). Technology adoption in the presence of network externalities. *Journal of political economy*, 94(4):822–841.

[157] Keller, M. (2020). Mp-spdz: A versatile framework for multi-party computation. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, pages 1575–1590.

[158] Kenny, C. T., Kuriwaki, S., McCartan, C., Rosenman, E. T., Simko, T., and Imai, K. (2021). The use of differential privacy for census data and its impact on redistricting: The case of the 2020 us census. *Science advances*, 7(41):eabk3283.

[159] Khatri, V. and Brown, C. V. (2010). Designing data governance. *Communications of the ACM*, 53(1):148–152.

[160] Kifer, D., Smith, A., and Thakurta, A. (2012). Private convex empirical risk minimization and high-dimensional regression. In *Conference on Learning Theory*, pages 25–1. JMLR Workshop and Conference Proceedings.

[161] Kilbertus, N., Gascon, A., Kusner, M., Veale, M., Gummadi, K., and Weller, A. (2018). Blind justice: Fairness with encrypted sensitive attributes. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2630–2639. PMLR.

[162] Kim, M., Song, Y., Wang, S., Xia, Y., Jiang, X., et al. (2018). Secure logistic regression based on homomorphic encryption: Design and evaluation. *JMIR medical informatics*, 6(2):e8805.

[Knott et al.] Knott, B., Venkataraman, S., Hannun, A., Sengupta, S., Ibrahim, M., and van der Maaten, L. Supplemental material for crypten: Secure multi-party computation meets machine learning.

[164] Knott, B., Venkataraman, S., Hannun, A., Sengupta, S., Ibrahim, M., and van der Maaten, L. (2021). Crypten: Secure multi-party computation meets machine learning. *Advances in Neural Information Processing Systems*, 34:4961–4973.

[165] Koh, P. W. and Liang, P. (2017a). Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR.

[166] Koh, P. W. and Liang, P. (2017b). Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pages 1885–1894.

[167] Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., et al. (2021). Wilds: A benchmark of in-the-wild distribution shifts. In *International conference on machine learning*, pages 5637–5664. PMLR.

[168] Koh, P. W. W., Ang, K.-S., Teo, H., and Liang, P. S. (2019). On the accuracy of influence functions for measuring group effects. In *Advances in Neural Information Processing Systems*, pages 5254–5264.

[169] Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.

[170] Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

[171] Kumar, A., Finley, B., Braud, T., Tarkoma, S., and Hui, P. (2020). Marketplace for ai models. *arXiv preprint arXiv:2003.01593*.

[172] Kwon, Y., Wu, E., Wu, K., and Zou, J. (2023). Datainf: Efficiently estimating data influence in lora-tuned llms and diffusion models. *arXiv preprint arXiv:2310.00902*.

[173] Lattner, C. and Pienaar, J. (2019). Mlir primer: A compiler infrastructure for the end of moore's law.

[174] Lauter, K., López-Alt, A., and Naehrig, M. (2014). Private computation on encrypted genomic data. In *International Conference on Cryptology and Information Security in Latin America*, pages 3–27. Springer.

[175] Lauter, K. E., Dai, W., and Laine, K. (2022). *Protecting privacy through homomorphic encryption*. Springer.

[176] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (2002). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[177] LeCun, Y. and Cortes, C. (1998). The MNIST database of handwritten digits.

[178] Lee, J., Lee, E., Lee, J.-W., Kim, Y., Kim, Y.-S., and No, J.-S. (2023a). Precise approximation of convolutional neural networks for homomorphically encrypted data. *IEEE Access*, 11:62062–62076.

[179] Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., and Kang, J. (2020). Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

[180] Lee, J.-W., Kang, H., Lee, Y., Choi, W., Eom, J., Deryabin, M., Lee, E., Lee, J., Yoo, D., Kim, Y.-S., and No, J.-S. (2021). Privacy-preserving machine learning with fully homomorphic encryption for deep neural network.

[181] Lee, K., Cooper, A. F., and Grimmelmann, J. (2023b). Talkin"bout ai generation: Copyright and the generative-ai supply chain. *arXiv preprint arXiv:2309.08133*.

[182] Legislature, C. (2018). California consumer privacy act. Cal. Civ. Code § 1798.100 et seq.

[183] Leung, C., Law, A., and Sima, O. (2019). Towards privacy-preserving collaborative gradient boosted decision trees. *UC Berkeley*.

[184] Li, B. and Micciancio, D. (2021). On the security of homomorphic encryption on approximate numbers. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 648–677. Springer.

[185] Li, C., Li, D. Y., Miklau, G., and Suciu, D. (2014). A theory of pricing private data. *ACM Transactions on Database Systems (TODS)*, 39(4):1–28.

[186] Li, D., Shao, R., Wang, H., Guo, H., Xing, E. P., and Zhang, H. (2022). Mpcformer: fast, performant and private transformer inference with mpc. *arXiv preprint arXiv:2211.01452*.

[187] Li, J., Fang, A., Smyrnis, G., Ivgi, M., Jordan, M., Gadre, S. Y., Bansal, H., Guha, E., Keh, S. S., Arora, K., et al. (2024a). Datacomp-lm: In search of the next generation of training sets for language models. *Advances in Neural Information Processing Systems*, 37:14200–14282.

[188] Li, M., Yang, Y., Chen, G., Yan, M., and Zhang, Y. (2024b). Sok: Understanding design choices and pitfalls of trusted execution environments. In *Proceedings of the 19th ACM Asia Conference on Computer and Communications Security*, pages 1600–1616.

[189] Li, T., Sahu, A. K., Talwalkar, A., and Smith, V. (2020). Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37(3):50–60.

[190] Li, X., Tramer, F., Liang, P., and Hashimoto, T. (2021). Large language models can be strong differentially private learners. *arXiv preprint arXiv:2110.05679*.

[191] Liang, F., Yu, W., An, D., Yang, Q., Fu, X., and Zhao, W. (2018). A survey on big data market: Pricing, trading and protection. *Ieee Access*, 6:15132–15154.

[192] Lin, J., Zhang, A., Lécuyer, M., Li, J., Panda, A., and Sen, S. (2022). Measuring the effect of training data on deep learning predictions via randomized experiments. In *International Conference on Machine Learning*, pages 13468–13504. PMLR.

[193] Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., et al. (2024). Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

[194] Liu, D. C. and Nocedal, J. (1989). On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528.

[195] Liu, K. Z. (2024). Machine unlearning in 2024.

[196] Liu, Z., Huang, D., Huang, K., Li, Z., and Zhao, J. (2021). Finbert: A pre-trained financial language representation model for financial text mining. In *Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence*, pages 4513–4519.

[197] Lu, W.-j., Huang, Z., Gu, Z., Li, J., Liu, J., Hong, C., Ren, K., Wei, T., and Chen, W. (2023). Bumblebee: Secure two-party inference framework for large transformers. *Cryptology ePrint Archive*.

[198] Luitse, D. and Denkena, W. (2021). The great transformer: Examining the role of large language models in the political economy of ai. *Big Data & Society*, 8(2):20539517211047734.

[199] Luo, J., Zhang, Y., Zhang, Z., Zhang, J., Mu, X., Wang, H., Yu, Y., and Xu, Z. (2024). Secformer: Fast and accurate privacy-preserving inference for transformer models via smpc. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 13333–13348.

[200] Lyu, L., Yu, H., Ma, X., Chen, C., Sun, L., Zhao, J., Yang, Q., and Philip, S. Y. (2022). Privacy and robustness in federated learning: Attacks and defenses. *IEEE transactions on neural networks and learning systems*.

[201] L'Heureux, A., Grolinger, K., Elyamany, H. F., and Capretz, M. A. M. (2017). Machine learning with big data: Challenges and approaches. *IEEE Access*, 5:7776–7797.

[202] Madry, A., Ilyas, A., Engstrom, L., Park, S. M. S., and Georgiev, K. (2024). Data attribution at scale. Tutorial presented at the International Conference on Machine Learning (ICML). ICML 2024, Vienna, Austria.

[203] Mahajan, D., Girshick, R., Ramanathan, V., He, K., Paluri, M., Li, Y., Bharambe, A., and Van Der Maaten, L. (2018). Exploring the limits of weakly supervised pretraining. In *Proceedings of the European conference on computer vision (ECCV)*, pages 181–196.

[204] Martens, J. et al. (2010). Deep learning via hessian-free optimization. In *Icml*, volume 27, pages 735–742.

[205] McCoy, M. S., Allen, A. L., Kopp, K., Mello, M. M., Patil, D., Ossorio, P., Joffe, S., and Emanuel, E. J. (2023). Ethical responsibilities for companies that process personal data. *The American Journal of Bioethics*, 23(11):11–23.

[206] McKinney, S. M., Sieniek, M., Godbole, V., Godwin, J., Antropova, N., Ashrafian, H., Back, T., Chesus, M., Corrado, G. S., Darzi, A., et al. (2020). International evaluation of an ai system for breast cancer screening. *Nature*, 577(7788):89–94.

[207] McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR.

[208] Mehta, I. (2024). Social networks are getting stingy with their data, leaving third-party developers in the lurch.

[209] Meng, X.-L. (2018). Statistical paradises and paradoxes in big data (i) law of large populations, big data paradox, and the 2016 us presidential election. *The Annals of Applied Statistics*, 12(2):685–726.

[210] Meta Platforms Inc. (2025). META Q2 2025 Earnings Call Transcript. Available from https://s21.q4cdn.com/399680738/files/doc_financials/2025/q2/META-Q2-2025-Earnings-Call-Transcript.pdf. Accessed: August 3, 2025.

[211] Microsoft (2022). SEAL: Update to Release 3.7.0. urlhttps://github.com/microsoft/SEAL/commit/6e6292ef93d1ae73d96473ccd4f6b2fb0c4472b9. Accessed: 2025-08-04.

[212] Miller, J. P., Taori, R., Raghunathan, A., Sagawa, S., Koh, P. W., Shankar, V., Liang, P., Carmon, Y., and Schmidt, L. (2021). Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization. In *International conference on machine learning*, pages 7721–7735. PMLR.

[213] Mo, F., Haddadi, H., Katevas, K., Marin, E., Perino, D., and Kourtellis, N. (2021). Ppfl: Privacy-preserving federated learning with trusted execution environments. In *Proceedings of the 19th annual international conference on mobile systems, applications, and services*, pages 94–108.

[214] Modular Inc. (2023). Mojo Programming Language. Software. Developed by Chris Lattner and Tim Davis.

[215] Mohassel, P. and Zhang, Y. (2017). Secureml: A system for scalable privacy-preserving machine learning. In *2017 IEEE symposium on security and privacy (SP)*, pages 19–38. IEEE.

[216] Munjal, K. and Bhatia, R. (2023). A systematic review of homomorphic encryption and its contributions in healthcare industry. *Complex & Intelligent Systems*, 9(4):3759–3786.

[217] Murtfeldt, R., Alterman, N., Kahveci, I., and West, J. D. (2024). Rip twitter api: A eulogy to its vast research contributions. *arXiv preprint arXiv:2404.07340*.

[218] Narayanan, A. (2013). What happened to the crypto dream?, part 2. *IEEE Security & Privacy*, 11(3):68–71.

[219] Narayanan, A. and Shmatikov, V. (2006). How to break anonymity of the netflix prize dataset. *arXiv preprint cs/0610105*.

[220] Nasr, M., Shokri, R., and Houmansadr, A. (2019). Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE symposium on security and privacy (SP)*, pages 739–753. IEEE.

[221] Neel, S., Roth, A., and Sharifi-Malvajerdi, S. (2021). Descent-to-delete: Gradient-based methods for machine unlearning. In *Algorithmic Learning Theory*, pages 931–962. PMLR.

[222] Ng, N., Cho, K., and Ghassemi, M. (2020). Ssmba: Self-supervised manifold based data augmentation for improving out-of-domain robustness. *arXiv preprint arXiv:2009.10195*.

[223] Nguyen, D. C., Pham, Q.-V., Pathirana, P. N., Ding, M., Seneviratne, A., Lin, Z., Dobre, O., and Hwang, W.-J. (2022a). Federated learning for smart healthcare: A survey. *ACM Computing Surveys (Csur)*, 55(3):1–37.

[224] Nguyen, T., Van Nguyen, C., Lai, V. D., Man, H., Ngo, N. T., Dernoncourt, F., Rossi, R. A., and Nguyen, T. H. (2023). Culturax: A cleaned, enormous, and multilingual dataset for large language models in 167 languages. *arXiv preprint arXiv:2309.09400*.

[225] Nguyen, T. T., Huynh, T. T., Ren, Z., Nguyen, P. L., Liew, A. W.-C., Yin, H., and Nguyen, Q. V. H. (2022b). A survey of machine unlearning. *arXiv preprint arXiv:2209.02299*.

[226] Nguyen, X., Wainwright, M. J., and Jordan, M. I. (2010). Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861.

[227] Nikolenko, S. I. (2021). *Synthetic data for deep learning*, volume 174. Springer.

[228] Nissenbaum, H. (2004). Privacy as contextual integrity. *Wash. L. Rev.*, 79:119.

[229] Ohm, P. (2014). Sensitive information. *S. Cal. L. Rev.*, 88:1125.

[230] Ontario Superior Court of Justice (2024). Toronto star newspapers ltd. et al. v. openai, inc. et al. Statement of Claim, Court File No. CV-24-00732231-00CL.

[231] OpenAI (2025). Introducing gpt-oss. https://openai.com/index/introducing-gpt-oss/. Accessed: August 6, 2025.

[232] Panda, A., Tang, X., Nasr, M., Choquette-Choo, C. A., and Mittal, P. (2024). Privacy auditing of large language models. In *ICML 2024 Next Generation of AI Safety Workshop*.

[233] Pang, Q., Zhu, J., Möllering, H., Zheng, W., and Schneider, T. (2024). Bolt: Privacy-preserving, accurate and efficient inference for transformers. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 4753–4771. IEEE.

[234] Paninski, L. (2003). Estimation of entropy and mutual information. *Neural computation*, 15(6):1191–1253.

[235] Papernot, N., Thakurta, A., Song, S., Chien, S., and Erlingsson, Ú. (2021). Tempered sigmoid activations for deep learning with differential privacy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9312–9321.

[236] Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D., and Le, Q. V. (2019). Specaugment: A simple data augmentation method for automatic speech recognition. *Interspeech 2019*, page 2613.

[237] Park, S., Kim, S., and Lim, Y.-s. (2022). Fairness audit of machine learning models with confidential computing. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 3488–3499, New York, NY, USA. Association for Computing Machinery.

[238] Park, S. M., Georgiev, K., Ilyas, A., Leclerc, G., and Madry, A. (2023). Trak: Attributing model behavior at scale. In *International Conference on Machine Learning (ICML)*.

[239] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

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

[241] Pejó, B., Tang, Q., and Biczók, G. (2018). The price of privacy in collaborative learning. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 2261–2263.

[242] Peng, Z. (2019). Danger of using fully homomorphic encryption: A look at microsoft seal. *arXiv preprint arXiv:1906.07127*.

[243] Petty, J., van Steenkiste, S., and Linzen, T. (2024). How does code pretraining affect language model task performance? *arXiv preprint arXiv:2409.04556*.

[244] Pollard, T. J., Johnson, A. E. W., Raffa, J. D., Celi, L. A., Mark, R. G., and Badawi, O. (2018). The eicu collaborative research database, a freely available multi-center database for critical care research. *Scientific Data*, 5.

[245] Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1:81–106.

[246] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

[247] Raisaro, J. L., Troncoso-Pastoriza, J. R., Misbach, M., Sousa, J. S., Pradervand, S., Missiaglia, E., Michielin, O., Ford, B., and Hubaux, J.-P. (2018). M ed c o: Enabling secure and privacy-preserving exploration of distributed clinical and genomic data. *IEEE/ACM transactions on computational biology and bioinformatics*, 16(4):1328–1341.

[248] Raj, A., Musco, C., Mackey, L., and Fusi, N. (2020). Model-specific data subsampling with influence functions. *arXiv preprint arXiv:2010.10218*.

[249] Raji, I. D. and Buolamwini, J. (2019). Actionable auditing: Investigating the impact of publicly naming biased performance results of commercial ai products. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 429–435.

[250] Rathee, D., Rathee, M., Goli, R. K. K., Gupta, D., Sharma, R., Chandran, N., and Rastogi, A. (2021). Sirnn: A math library for secure rnn inference. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 1003–1020. IEEE.

[251] Rathee, D., Rathee, M., Kumar, N., Chandran, N., Gupta, D., Rastogi, A., and Sharma, R. (2020). Cryptflow2: Practical 2-party secure inference. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 325–342.

[252] Raynal, M. and Troncoso, C. (2024). On the conflict of robustness and learning in collaborative machine learning. *arXiv preprint arXiv:2402.13700*.

[253] Reed, S., Zolna, K., Parisotto, E., Colmenarejo, S. G., Novikov, A., Barth-Maron, G., Gimenez, M., Sulsky, Y., Kay, J., Springenberg, J. T., et al. (2022). A generalist agent. *arXiv preprint arXiv:2205.06175*.

[254] Riazi, M. S., Weinert, C., Tkachenko, O., Songhori, E. M., Schneider, T., and Koushanfar, F. (2018). Chameleon: A hybrid secure computation framework for machine learning applications. In *Proceedings of the 2018 on Asia conference on computer and communications security*, pages 707–721.

[255] Rieke, N., Hancox, J., Li, W., Milletari, F., Roth, H. R., Albarqouni, S., Bakas, S., Galtier, M. N., Landman, B. A., Maier-Hein, K., et al. (2020). The future of digital health with federated learning. *NPJ digital medicine*, 3(1):1–7.

[256] Rubinstein, I. S. (2013). Big data: The end of privacy or a new beginning? *Int'l Data Priv. L.*, 3:74.

[257] Sabne, A. (2020). Xla : Compiling machine learning for peak performance.

[258] Sadowski, J. (2019). When data is capital: Datafication, accumulation, and extraction. *Big data & society*, 6(1):2053951718820549.

[259] Sain, S. R. (1996). The nature of statistical learning theory.

[260] Samarati, P. and Sweeney, L. (1998). Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression.

[261] Sapir, B. (2022). Data Science without Seeing Data: How to Set Microsoft Open Source SEAL Parameters. Intuit Engineering, https://medium.com/intuit-engineering/data-science-without-seeing-data-how-to-set-microsoft-open-source-seal-parameters-72929b184058. Accessed: 2025-08-04.

[262] Sav, S., Pyrgelis, A., Troncoso-Pastoriza, J. R., Froelicher, D., Bossuat, J.-P., Sousa, J. S., and Hubaux, J.-P. (2020). Poseidon: Privacy-preserving federated neural network learning. *arXiv preprint arXiv:2009.00349*.

[263] Seyyed-Kalantari, L., Zhang, H., McDermott, M. B., Chen, I. Y., and Ghassemi, M. (2021). Underdiagnosis bias of artificial intelligence algorithms applied to chest radiographs in under-served patient populations. *Nature medicine*, 27(12):2176–2182.

[264] Shamir, A. (1979). How to share a secret. *Communications of the ACM*, 22(11):612–613.

[265] Shapley, L. S. (1952). A value for n-person games. Technical report, Rand Corp Santa Monica CA.

[266] Sheller, M. J., Edwards, B., Reina, G. A., Martin, J., Pati, S., Kotrotsou, A., Milchenko, M., Xu, W., Marcus, D., Colen, R. R., et al. (2020). Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. *Scientific reports*, 10(1):12598.

[267] Shen, J. H., Raji, I. D., and Chen, I. Y. (2024). The data addition dilemma. In *Machine Learning for Healthcare Conference*. PMLR.

[268] Shen, Z., Tao, T., Ma, L., Neiswanger, W., Liu, Z., Wang, H., Tan, B., Hestness, J., Vassilieva, N., Soboleva, D., et al. (2023). Slimpajama-dc: Understanding data combinations for llm training. *arXiv preprint arXiv:2309.10818*.

[269] Shokri, R. and Shmatikov, V. (2015). Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1310–1321.

[270] Shokri, R., Stronati, M., Song, C., and Shmatikov, V. (2017). Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE.

[271] Siddiqui, S. A., Rajkumar, N., Maharaj, T., Krueger, D., and Hooker, S. (2022). Metadata archaeology: Unearthing data subsets by leveraging training dynamics. *arXiv preprint arXiv:2209.10015*.

[272] Siibak, A. and Traks, K. (2019). The dark sides of sharenting. *Catalan journal of communication & cultural studies*, 11(1):115–121.

[273] Singh, A. K. and Strouse, D. (2024). Tokenization counts: the impact of tokenization on arithmetic in frontier llms. *arXiv preprint arXiv:2402.14903*.

[274] Solove, D. J. (2005). A taxonomy of privacy. *U. Pa. l. Rev.*, 154:477.

[275] Solove, D. J. (2021). The myth of the privacy paradox. *Geo. Wash. L. Rev.*, 89:1.

[276] Song, T., Tong, Y., and Wei, S. (2019). Profit allocation for federated learning. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 2577–2586. IEEE.

[277] Srinivasan, W. Z., Akshayaram, P., and Ada, P. R. (2019). Delphi: A cryptographic inference service for neural networks. In *Proc. 29th USENIX secur. symp*, volume 3.

[278] Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M., Abid, A., Fisch, A., Brown, A. R., Santoro, A., Gupta, A., Garriga-Alonso, A., et al. (2022). Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

[279] Stripelis, D., Saleem, H., Ghai, T., Dhinagar, N., Gupta, U., Anastasiou, C., Ver Steeg, G., Ravi, S., Naveed, M., Thompson, P. M., et al. (2021). 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.

[280] Sun, C., Shrivastava, A., Singh, S., and Gupta, A. (2017). Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852.

[281] Suriyakumar, V. M., Papernot, N., Goldenberg, A., and Ghassemi, M. (2021). Chasing your long tails: Differentially private prediction in health care settings. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 723–734.

[282] Sweeney, L. (2002). k-anonymity: A model for protecting privacy. *International journal of uncertainty, fuzziness and knowledge-based systems*, 10(05):557–570.

[283] Tang, J., Korolova, A., Bai, X., Wang, X., and Wang, X. (2017). Privacy loss in apple's implementation of differential privacy on macos 10.12. *arXiv preprint arXiv:1709.02753*.

[284] Taori, R., Dave, A., Shankar, V., Carlini, N., Recht, B., and Schmidt, L. (2020). Measuring robustness to natural distribution shifts in image classification. *Advances in Neural Information Processing Systems*, 33:18583–18599.

[285] Team, K., Bai, Y., Bao, Y., Chen, G., Chen, J., Chen, N., Chen, R., Chen, Y., Chen, Y., Chen, Y., Chen, Z., Cui, J., Ding, H., Dong, M., Du, A., Du, C., Du, D., Du, Y., Fan, Y., Feng, Y., Fu, K., Gao, B., Gao, H., Gao, P., Gao, T., Gu, X., Guan, L., Guo, H., Guo, J., Hu, H., Hao, X., He, T., He, W., He, W., Hong, C., Hu, Y., Hu, Z., Huang, W., Huang, Z., Huang, Z., Jiang, T., Jiang, Z., Jin, X., Kang, Y., Lai, G., Li, C., Li, F., Li, H., Li, M., Li, W., Li, Y., Li, Y., Li, Z., Li, Z., Lin, H., Lin, X., Lin, Z., Liu, C., Liu, C., Liu, H., Liu, J., Liu, J., Liu, L., Liu, S., Liu, T. Y., Liu, T., Liu, W., Liu, Y., Liu, Y., Liu, Y., Liu, Y., Liu, Z., Lu, E., Lu, L., Ma, S., Ma, X., Ma, Y., Mao, S., Mei, J., Men, X., Miao, Y., Pan, S., Peng, Y., Qin, R., Qu, B., Shang, Z., Shi, L., Shi, S., Song, F., Su, J., Su, Z., Sun, X., Sung, F., Tang, H., Tao, J., Teng, Q., Wang, C., Wang, D., Wang, F., Wang,

H., Wang, J., Wang, J., Wang, J., Wang, S., Wang, S., Wang, Y., Wang, Y., Wang, Y., Wang, Y., Wang, Y., Wang, Z., Wang, Z., Wang, Z., Wei, C., Wei, Q., Wu, W., Wu, X., Wu, Y., Xiao, C., Xie, X., Xiong, W., Xu, B., Xu, J., Xu, J., Xu, L. H., Xu, L., Xu, S., Xu, W., Xu, X., Xu, Y., Xu, Z., Yan, J., Yan, Y., Yang, X., Yang, Y., Yang, Z., Yang, Z., Yang, Z., Yao, H., Yao, X., Ye, W., Ye, Z., Yin, B., Yu, L., Yuan, E., Yuan, H., Yuan, M., Zhan, H., Zhang, D., Zhang, H., Zhang, W., Zhang, X., Zhang, Y., Zhang, Y., Zhang, Y., Zhang, Y., Zhang, Y., Zhang, Y., Zhang, Z., Zhao, H., Zhao, Y., Zheng, H., Zheng, S., Zhou, J., Zhou, X., Zhou, Z., Zhu, Z., Zhuang, W., and Zu, X. (2025). Kimi k2: Open agentic intelligence.

[286] Teich, J. (2012). Hardware/software codesign: The past, the present, and predicting the future. *Proceedings of the IEEE*, 100(Special Centennial Issue):1411–1430.

[287] Tenopir, C., Allard, S., Douglass, K., Aydinoglu, A. U., Wu, L., Read, E., Manoff, M., and Frame, M. (2011). Data sharing by scientists: practices and perceptions. *PloS one*, 6(6):e21101.

[288] Thambawita, V., Salehi, P., Sheshkal, S. A., Hicks, S. A., Hammer, H. L., Parasa, S., Lange, T. d., Halvorsen, P., and Riegler, M. A. (2022). Singan-seg: Synthetic training data generation for medical image segmentation. *PloS one*, 17(5):e0267976.

[289] Thatcher, J., O'Sullivan, D., and Mahmoudi, D. (2016). Data colonialism through accumulation by dispossession: New metaphors for daily data. *Environment and Planning D: Society and Space*, 34(6):990–1006.

[290] Tikhonov, A. N. (1977). Solutions of ill posed problems.

[291] Tolpegin, V., Truex, S., Gursoy, M. E., and Liu, L. (2020). Data poisoning attacks against federated learning systems. In *Computer security–ESORICs 2020: 25th European symposium on research in computer security, ESORICs 2020, guildford, UK, September 14–18, 2020, proceedings, part i 25*, pages 480–501. Springer.

[292] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. (2023). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

[293] Tramer, F. and Boneh, D. (2020). Differentially private learning needs better features (or much more data). *arXiv preprint arXiv:2011.11660*.

[294] Tramèr, F., Kamath, G., and Carlini, N. (2022). Position: Considerations for differentially private learning with large-scale public pretraining. In *Forty-first International Conference on Machine Learning*.

[295] Truex, S., Baracaldo, N., Anwar, A., Steinke, T., Ludwig, H., Zhang, R., and Zhou, Y. (2019). A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM workshop on artificial intelligence and security*, pages 1–11.

[296] United States District Court Northern District of California (2025). Bartz et al. v. anthropic pbc. Docket No. 3:24-cv-05417.

[297] Usynin, D., Rueckert, D., and Kaissis, G. (2024). Incentivising the federation: gradient-based metrics for data selection and valuation in private decentralised training. In *European Interdisciplinary Cybersecurity Conference*, pages 179–185.

[298] Usynin, D., Ziller, A., Makowski, M., Braren, R., Rueckert, D., Glocker, B., Kaissis, G., and Passerat-Palmbach, J. (2021). Adversarial interference and its mitigations in privacy-preserving collaborative machine learning. *Nature Machine Intelligence*, 3(9):749–758.

[299] Valiant, G. and Valiant, P. (2017). Estimating the unseen: improved estimators for entropy and other properties. *Journal of the ACM (JACM)*, 64(6):1–41.

[300] Valiant, P. and Valiant, G. (2013). Estimating the unseen: Improved estimators for entropy and other properties. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

[301] van der Maaten, L. and Hannun, A. (2020). The trade-offs of private prediction. *arXiv preprint arXiv:2007.05089*.

[302] van der Ouderaa, T. F., Baioumy, M., Beton, M., Howes, S., Vrabie, G., and Cheema, A. (2025). Towards large scale training on apple silicon. In *ES-FoMo III: 3rd Workshop on Efficient Systems for Foundation Models*.

[303] van Egmond, M. B., Spini, G., van der Galien, O., IJpma, A., Veugen, T., Kraaij, W., Sangers, A., Rooijakkers, T., Langenkamp, P., Kamphorst, B., et al. (2021). Privacy-preserving dataset combination and lasso regression for healthcare predictions. *BMC medical informatics and decision making*, 21:1–16.

[304] Vapnik, V. (2006). *Estimation of dependences based on empirical data*. Springer Science & Business Media.

[305] Villalobos, P., Ho, A., Sevilla, J., Besiroglu, T., Heim, L., and Hobbhahn, M. (2022). Will we run out of data? limits of llm scaling based on human-generated data. *arXiv preprint arXiv:2211.04325*.

[306] Villalobos, P., Ho, A., Sevilla, J., Besiroglu, T., Heim, L., and Hobbhahn, M. (2024). Will we run out of data? limits of llm scaling based on human-generated data.

[307] Wang, G., Dang, C. X., and Zhou, Z. (2019). Measure contribution of participants in federated learning. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 2597–2604. IEEE.

[308] Wang, J., Lan, C., Liu, C., Ouyang, Y., Qin, T., Lu, W., Chen, Y., Zeng, W., and Philip, S. Y. (2022). Generalizing to unseen domains: A survey on domain generalization. *IEEE transactions on knowledge and data engineering*, 35(8):8052–8072.

[309] Wang, J. T. and Jia, R. (2023). Data banzhaf: A robust data valuation framework for machine learning. In *International Conference on Artificial Intelligence and Statistics*, pages 6388–6421. PMLR.

[310] Wang, T., Rausch, J., Zhang, C., Jia, R., and Song, D. (2020). A principled approach to data valuation for federated learning. *Federated Learning: Privacy and Incentive*, pages 153–167.

[311] Warren, S. and Brandeis, L. (1989). The right to privacy. In *Killing the Messenger: 100 Years of Media Criticism*, pages 1–21. Columbia University Press.

[312] Water, R. V. D., Schmidt, H., Elbers, P., Thoral, P. J., Arnrich, B., and Rockenschaub, P. (2023). Yet another icu benchmark: A flexible multi-center framework for clinical ml. *ArXiv*, abs/2306.05109.

[313] Westin, A. F. (1968). Privacy and freedom. *Washington and Lee Law Review*, 25(1):166.

[314] Xia, M., Malladi, S., Gururangan, S., Arora, S., and Chen, D. (2024). Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*.

[315] Xu, J., Glicksberg, B. S., Su, C., Walker, P., Bian, J., and Wang, F. (2021). Federated learning for healthcare informatics. *Journal of healthcare informatics research*, 5:1–19.

[316] Xu, L., Skoularidou, M., Cuesta-Infante, A., and Veeramachaneni, K. (2019). Modeling tabular data using conditional gan. *Advances in neural information processing systems*, 32.

[317] Xu, M., Sun, J., Yang, X., Yao, K., and Wang, C. (2023a). Netflix and forget: Efficient and exact machine unlearning from bi-linear recommendations. *arXiv preprint arXiv:2302.06676*.

[318] Xu, M., Usynin, D., and Barez, F. (2024). PrivacyML: Meaningful Privacy-Preserving Machine Learning and How To Evaluate AI Privacy. In *Proceedings of the NeurIPS 2024 Tutorials*, Vancouver, Canada. NeurIPS 2024 Tutorial.

[319] Xu, M., Zhang, D., Phothilimthana, P. M., Mahajan, D., Qiu, H., and Musau, P. (2025). NeurIPS 2025 Workshop on Machine Learning for Systems. NeurIPS 2025 Workshop: MLForSys.

[320] Xu, X., Hannun, A., and Van Der Maaten, L. (2022). Data appraisal without data sharing. In *International Conference on Artificial Intelligence and Statistics*, pages 11422–11437. PMLR.

[321] Xu, Z., Zhang, Y., Andrew, G., Choquette-Choo, C. A., Kairouz, P., McMahan, H. B., Rosenstock, J., and Zhang, Y. (2023b). Federated learning of gboard language models with differential privacy. *arXiv preprint arXiv:2305.18465*.

[322] Yang, M., Chi, C.-H., Lam, K.-Y., Feng, J., Guo, T., and Ni, W. (2024a). Tabular data synthesis with differential privacy: A survey. *arXiv preprint arXiv:2411.03351*.

[323] Yang, Z., Zhang, Y., Zheng, Y., Tian, X., Peng, H., Liu, T., and Han, B. (2024b). Fedfed: Feature distillation against data heterogeneity in federated learning. *Advances in Neural Information Processing Systems*, 36.

[324] Yao, A. C. (1982). Protocols for secure computations. In *23rd annual symposium on foundations of computer science (sfcs 1982)*, pages 160–164. IEEE.

[325] Young, M., Rodriguez, L., Keller, E., Sun, F., Sa, B., Whittington, J., and Howe, B. (2019). Beyond open vs. closed: Balancing individual privacy and public accountability in data sharing. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 191–200.

[326] Yousefpour, A., Shilov, I., Sablayrolles, A., Testuggine, D., Prasad, K., Malek, M., Nguyen, J., Ghosh, S., Bharadwaj, A., Zhao, J., et al. (2021). Opacus: User-friendly differential privacy library in pytorch. *arXiv preprint arXiv:2109.12298*.

[327] Ypma, T. J. (1995). Historical development of the newton–raphson method. *SIAM review*, 37(4):531–551.

[328] Yu, D., Naik, S., Backurs, A., Gopi, S., Inan, H. A., Kamath, G., Kulkarni, J., Lee, Y. T., Manoel, A., Wutschitz, L., et al. (2021). Differentially private fine-tuning of language models. *arXiv preprint arXiv:2110.06500*.

[329] Zama (2022). Concrete ML: a privacy-preserving machine learning library using fully homomorphic encryption for data scientists. https://github.com/zama-ai/concrete-ml.

[330] Zhang, Y., Jia, R., Pei, H., Wang, W., Li, B., and Song, D. (2020a). The secret revealer: Generative model-inversion attacks against deep neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 253–261.

[331] Zhang, Y., Qin, J., Park, D. S., Han, W., Chiu, C.-C., Pang, R., Le, Q. V., and Wu, Y. (2020b). Pushing the limits of semi-supervised learning for automatic speech recognition. *arXiv preprint arXiv:2010.10504*.

[332] Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., and Chandra, V. (2018). Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*.

[333] Zheng, W., Deng, R., Chen, W., Popa, R. A., Panda, A., and Stoica, I. (2021). Cerebro: A platform for {Multi-Party} cryptographic collaborative learning. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2723–2740.

[334] Zhou, I., Tofigh, F., Piccardi, M., Abolhasan, M., Franklin, D., and Lipman, J. (2024). Secure multi-party computation for machine learning: A survey. *IEEE Access*, 12:53881–53899.

[335] Zhu, L., Liu, Z., and Han, S. (2019). Deep leakage from gradients. *Advances in neural information processing systems*, 32.

[336] Zuckerberg, M. (2025). Personal Superintelligence. https://web.archive.org/web/20250731085000/https://www.meta.com/superintelligence/. Accessed: August 3, 2025.