Privacy–Utility Optimization for Training Data Minimization

Zhendong Guo[1], Shuhao Liu[2], Wenfei Fan[2,3], Jiahui Jin[1]
Southeast University[1], China Shenzhen Institute of Computing Sciences[2], China University of Edinburgh[3], UK
zdguo@seu.edu.cn, shuhao@sics.ac.cn, wenfei@inf.ed.ac.uk, jjin@seu.edu.cn

ABSTRACT

This paper studies compliance-aware data minimization for machine learning training. We formulate the problem as deriving a minimized dataset that reduces residual privacy risk while preserving model utility within a specified tolerance, and show that the problem is NP-complete. To address the challenge, we propose TRIM, a framework that jointly optimizes vertical and horizontal minimization. The key idea is a novel *retention view*: starting from a fully suppressed dataset, TRIM progressively restores information by attribute refinement and retention-class inclusion. Guided by a ratio-marginal objective, TRIM explicitly balances utility recovery against privacy leakage. To efficiently evaluate candidate operations, TRIM employs a two-stage utility estimator, avoiding retraining the target model. We prove a bounded approximation guarantee and generate auditable records for DPIA compliance. Using real-world datasets, we experimentally show that TRIM reduces utility degradation by 84.9% over the best baselines at matched privacy level, scales to million-row data, and supports DPIA compliance.

PVLDB Reference Format:
Zhendong Guo[1], Shuhao Liu[2], Wenfei Fan[2,3], Jiahui Jin[1] . Privacy–Utility Optimization for Training Data Minimization. PVLDB, 20(1): X-X, 2027.
doi:XX.XX/XXX.XX

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/gogodong/TRIM.

1 INTRODUCTION

Modern machine learning (ML) models are routinely trained on datasets containing rich personal information [52]. Such information may be disclosed through released data, *e.g.,* via data reconstruction [23], attribute-inference attacks [25], or leaked by trained models [28, 30], *e.g.,* through memorization [12]. Thus, minimizing unnecessary personal data in training datasets is essential for reducing privacy risk and limiting potential disclosure.

Prior approaches typically perform either *vertical minimization*, which removes or generalizes attributes [21, 50], or *horizontal minimization*, which removes records through sampling, coresets, or record selection [35, 38, 39, 43, 45, 57]. Neither is sufficient alone: at a given utility tolerance, vertical methods may leave sensitive attributes insufficiently generalized and retain low-utility records that increase re-identification risk, whereas horizontal methods may preserve highly identifying attribute combinations in the dataset.

Proceedings of the VLDB Endowment, Vol. 20, No. 1 ISSN 2150-8097.
doi:XX.XX/XXX.XX

Table 1: Illustrative records adapted from the `Diabetes` dataset [51].

tuple	age†	diagnosis†	#meds	readmit$_{30}$
t_1	79	410.41 (Ischemic / Heart attack)	18	Yes
t_2	72	414.01 (Ischemic / Atherosclerosis)	15	No
t_3	86	428.00 (Other heart / Heart failure)	21	Yes
t_4	82	402.00 (Hypertensive / Heart disease)	9	No
t_5	88	410.41 (Other heart / Arrhythmia)	20	Yes

Vertical minimization via coarsening (age→decade, `diagnosis`→ICD-9 family):
$t_1, t_2 \rightarrow$ (70s, Ischemic), $t_3, t_5 \rightarrow$ (80s, Other heart) become alike;
yet $t_4 \rightarrow$ (80s, Hypertensive) remains uniquely identifiable
Horizontal minimization via deletion (*e.g.,* keep t_1, t_2, t_3): each kept record is unique

Example 1: Table 1 shows medical records adapted from the (130-US Hospitals) `Diabetes` dataset [51] for training a 30-day readmission classifier. An adversary who knows a patient's age and `diagnosis` can pinpoint that patient's record, as no two records share the same pair. Every patient is thus directly identifiable, and the model may memorize these one-of-a-kind patterns.

Vertical minimization coarsens attributes, *e.g.,* age to decades and `diagnosis` to ICD-9 families, which leaves t_1 and t_2 indistinguishable, as well as t_3 and t_5. Yet t_4, the only patient with a hypertensive disease, is still unique; hiding it needs the far coarser *circulatory disease* level, which blurs every other record with the same value and hurts the classifier utility. Horizontal minimization fares no better: deleting records leaves whoever is kept just as identifiable. □

Effective training-data minimization must therefore jointly determine (a) which records to retain and (b) at what level of detail their attributes should be represented. This raises a key question: what constitutes a "good" minimized training dataset?

Practical training-data minimization should satisfy three requirements: (1) *utility preservation*, maintaining the utility required by the target ML task; (2) *privacy minimization*, minimizing the remaining risk of re-identification; and (3) *decision accountability*, providing auditable justifications for each retained attribute or record.

These requirements closely align with Data Protection Impact Assessments (DPIAs) under regulations such as the GDPR [20]. A DPIA requires that the personal data used for a given processing activity is necessary for its intended purpose, that the resulting privacy risks are proportionate to the expected benefits, and that the rationale behind data-retention decisions is properly documented.

These requirements raise three challenges: (1) jointly optimizing privacy risk and model utility; (2) efficiently estimating utility over a large space of record-retention and attribute-generalization decisions without repeated retraining; and (3) making minimization decisions transparent and auditable for regulatory and DPIA review. These call for an efficient, privacy-aware, and auditable framework.

Contributions & Organization. To address these challenges, we develop TRIM, a compliance-aware TRaIning data Minimization framework that jointly optimizes privacy and utility, and produces auditable retention decisions for DPIA-oriented review.

(1) Compliance-aware training-data minimization (Section 3). We

formulate training-data minimization as an optimization problem that minimizes privacy risk with a utility guarantee. Given a model $\mathcal{M}$ and relational dataset $\mathcal{D}$, it aims to derive a minimized dataset $\mathcal{D}_M$ that minimizes information leakage while preserving $\mathcal{M}$'s utility within a tolerance ϵ. We show that the problem is NP-complete.

(2) Retention-view optimization (Section 4). We present TRIM, our framework that jointly integrates vertical and horizontal minimization. It proposes a *retention view*: starting from a fully suppressed dataset, TRIM progressively restores information via attribute refinement and retention-class inclusion. This enables a *ratio-marginal optimization strategy* that prioritizes the operation with the largest utility gain per unit privacy risk, explicitly balancing utility and privacy while offering a provable approximation guarantee.

(3) Provable and auditable minimization (Sections 5–6). Section 5 develops the retention-based search space of TRIM, which unifies horizontal and vertical minimization through two novel operation families: retention-class inclusion and attribute refinement. This search space is bounded, semantically meaningful, and auditable.

Section 6 develops the evaluation and selection core. TRIM quantifies *residual privacy risk*, *i.e.,* the remaining re-identification exposure after minimization, and balances it against model utility using ratio-marginal optimization. To avoid repeatedly retraining the target model, TRIM further introduces a two-stage utility estimator for ranking and certifying candidate operations.

Together, these components yield a bounded approximation guarantee and auditable records that justify every retention decision.

(4) Experimental study (Section 7). Using real-world and synthetic datasets, we evaluate TRIM's effectiveness, efficiency, robustness, and auditability: TRIM (a) improves the privacy-utility frontier, reducing utility degradation by up to 84.9% over the best baselines at matched privacy, with consistent gains in tail risk and across downstream models; (b) is 1.66-28.82× faster than baselines that retrain the model and scales sublinearly to one million rows; (c) produces auditable DPIA traces, with stable default hyperparameters; and (d) derives its gains from joint horizontal-vertical minimization, ratio-marginal selection, and two-stage utility estimation.

We discuss related work in Section 2 and summarize the novelty of the work in Section 8. We defer detailed proofs to [1].

VLDB relevance. Training-data minimization builds a task-specific retained view of a relational dataset by jointly selecting tuples and controlling attribute granularity. It requires efficient search over data transformations, evaluation of candidate views under privacy-utility constraints, and provenance tracking for audit. These are core challenges in privacy-aware and accountable data management.

2 BACKGROUND AND RELATED WORK

This section outlines regulatory concepts relevant to data minimization and reviews existing techniques that are potentially applicable.

Key concepts. We start with relevant regulatory concepts.

Personal data. Under modern privacy regulations, personal data refers to any information relating to an identified or identifiable natural person. This includes both *direct identifiers* (*e.g.,* names or IDs) and *quasi-identifiers (QIs)* (*e.g.,* demographic attributes or locations) whose combinations may enable re-identification. Consequently, both features and labels in ML training datasets may constitute personal data, even after direct identifiers have been removed.

Table 2: Training-data minimization methods (△: partial).

Method	Hor.	Ver.	Explicit Risk	Utility Constraint	Auditable
Horizontal minimization	✓	✗	✗	✗	✗
Vertical minimization	✗	✓	△	✗	✓
Feature selection	✗	✓	✗	✗	✓
Differential privacy	△	△	✓	✗	✗
Dataset distillation	△	△	✗	✗	✗
TRIM	✓	✓	✓	✓	✓

DPIAs. A Data Protection Impact Assessment (DPIA) is an *ex ante* assessment process for high-risk processing of personal data [20]. It requires organizations to justify the necessity and proportionality of their use of personal data and to document the resulting privacy risks and mitigation measures (see Section 1 for the requirements). In this paper, training-data minimization is treated as a practical mechanism for supporting these DPIA obligations.

Example 2: Continuing with Example 1, a DPIA for the 30-day readmission classifier must produce three artifacts.

- *A necessity-and-proportionality record.* For every retained attribute (*e.g.,* `age`, `diagnosis`) and every retained tuple in Table 1, the controller must justify the necessity, *i.e.,* why the chosen granularity is the least intrusive form that remains adequate.
- *A quantified residual-risk evaluation.* The DPIA must report a quantitative measure of re-identification risk for the retained dataset (*e.g.,* the singling-out probability for tuples such as t_4), not merely a qualitative claim that risk is "low".
- *An auditable mitigation log.* The DPIA must also record each minimization decision with its utility impact and privacy-risk change to support independent review of the final dataset. □

Datasets. Consider a relational dataset $\mathcal{D}$ over schema $\mathcal{R} = \{R_k\}_{k=1}^{K}$. For ML training, we assume *w.l.o.g.* that $\mathcal{R}$ contains a single relation R. We treat all personal-data attributes of R as QIs, with direct identifiers removed before minimization. Let $A = \{A_1, \ldots, A_m\} \subseteq \text{attr}(R)$ be the QIs. For each tuple $t_i \in \mathcal{D}$, $A(t_i)$ is the QI record associated with person i in population $\mathcal{I}$.

Data minimization. Article 5(1)(c) of the GDPR [20] requires personal data to be limited to what is necessary for a specified purpose. For ML pipelines, this means deriving a minimized training dataset that preserves the required model utility while reducing the amount and granularity of retained personal data. We model minimization as a sequence of *minimization operations* on $\mathcal{D}$, in two categories:

- *Horizontal minimization*, which reduces the set of retained records by removing tuples not required for the target utility;
- *Vertical minimization*, which reduces the amount of information retained per record by removing attributes or replacing fine-grained values with coarser generalizations.

It aims to identify a minimized dataset that balances utility and privacy risk while providing auditable justifications for retained data.

Related work. Existing approaches reduce training data by horizontal minimization, vertical minimization, feature selection, differential privacy, and dataset distillation, as summarized in Table 2.

Data minimization for ML pipelines. Previous approaches typically focus on either *horizontal* or *vertical* minimization. Horizontal ap-

proaches [9, 21, 26, 38, 45, 48] often adopt sampling or coreset selection, but may leave highly identifying attributes intact and provide limited support for auditability. Vertical approaches [21, 33, 46, 50] typically optimize heuristic privacy-utility objectives. Across both categories, privacy risk and model utility are rarely enforced jointly through explicit objectives and constraints (see Example 1).

In contrast, TRIM (1) minimizes an explicit privacy-risk objective subject to a utility guarantee, (2) combines horizontal and vertical minimization, (3) efficiently estimates the utility impact of candidate operations without repeated retraining, and (4) provides auditable justifications for data-retention decisions.

Training data reduction. Several related techniques can reduce training data, although they were designed for different objectives.

(1) Data de-identification and sanitization. Classical k-anonymity generalizes QIs so that each record shares its QI values with at least $k-1$ other records [44]. Although such group-size guarantee is interpretable, it neither accounts for downstream model utility nor establishes the necessity and proportionality of individual retention decisions. Data de-identification and sanitization techniques [13, 31, 32, 42, 53] detect and remove personally identifiable information (PII), including direct identifiers and sensitive attributes. Industrial tools (*e.g.,* [34, 49]) can effectively reduce privacy exposure, but are largely task-agnostic: they sanitize data without considering downstream model utility. Consequently, they may remove information that is necessary for the target ML task. In contrast, TRIM explicitly balances privacy risk against model utility and retains only the data justified by the intended use.

(2) Dataset distillation and condensation. Dataset distillation and condensation [54, 59, 60] construct a small synthetic dataset from an original dataset while preserving model performance. Although the resulting surrogate may reduce direct exposure to the original data, its residual privacy risk is often unclear. Furthermore, existing methods optimize task-specific surrogate objectives (*e.g.,* training gradients) rather than explicit privacy and utility metrics. In contrast, TRIM directly optimizes the privacy-utility tradeoff and provides auditable justifications for its retention decisions.

(3) Feature selection. Feature selection [3, 19, 58] retains or discards features based primarily on predictive utility. As a result, it neither explicitly models privacy risk nor supports intermediate levels of disclosure between full retention and removal. In contrast, TRIM enables hierarchical feature generalization, allowing finer-grained and more interpretable privacy-utility tradeoffs.

(4) Differential privacy (DP). DP [17, 18] can protect ML training against membership-inference attacks [47], *e.g.,* via DP-SGD [2] and its variants [4, 7, 10]. Complementary to data minimization, DP limits the influence of any individual record, often by adding calibrated noise during training, and thus mitigates membership-inference risks outside TRIM's data-level leakage model. TRIM and other data-minimization techniques can therefore serve as preprocessing steps for DP, providing additional protection against accidental training-data leakage while preserving auditability.

3 COMPLIANCE-AWARE ML TRAINING

This section formalizes training-data minimization for ML pipelines. We first introduce the threat model and the privacy and utility metrics that capture the objectives of minimization (Section 3.1). We then formulate the optimization problem, establish its complexity, and identify the key challenges that motivate TRIM (Section 3.2).

3.1 Privacy and Utility Foundations

We define the threat model and the metrics to quantify privacy risk and task utility, which form the basis of our optimization problem.

Threat model. We model privacy exposure as *probabilistic re-identification* by an adversary that observes a minimized dataset $\mathcal{D}_M$ and possesses auxiliary information about target individuals. Given $\mathcal{D}_M$ derived from dataset $\mathcal{D}$, an adversary $\mathcal{A}$ combines $\mathcal{D}_M$ with auxiliary information about target individuals to assign posterior re-identification probabilities. This is related to the three attacks identified in [6], and subsumes more powerful attacks [15, 25, 37]:

- *Singling out*: The adversary isolates a record or small set of records corresponding to a target individual in $\mathcal{D}_M$;
- *Linkability*: The adversary determines whether multiple records refer to the same individual, *e.g.,* by correlating attributes; and
- *Inference*: The adversary infers an individual's sensitive attribute value with non-negligible probability.

These attacks involve different posterior hypotheses: record identity for singling out, shared identity for linkability, and sensitive values for inference. Our metric formalizes singling out, while experiments show that TRIM also mitigates linkability and inference. Formalizing these attacks requires different posterior spaces, especially when memorization reveals only partial record information [12].

Background knowledge. We assume that adversary $\mathcal{A}$ possesses auxiliary information about a set $\mathcal{I}$ of target individuals. For a subset of QIs $A \subseteq \mathrm{attr}(\mathcal{R})$, let $\mathrm{dom}(A)$ denote the space of possible assignments to A. For each target individual $i \in \mathcal{I}$, π_i is a prior distribution over $\mathrm{dom}(A)$ that represents $\mathcal{A}$'s uncertainty about the QI values of i before observing $\mathcal{D}_M$. The more concentrated π_i is, the stronger the adversary's background knowledge becomes.

Posterior analysis. Suppose that the adversary observes the minimized dataset $\mathcal{D}_M$ and attempts to identify a target individual $i \in \mathcal{I}$. Let $A(t) \in \mathrm{dom}(A)$ denote the QI values of t. For a record $t \in \mathcal{D}_M$, we define the posterior probability that t corresponds to i as

$$q_i(t \mid \mathcal{D}_M) := \Pr[t \mid \mathcal{D}_M] = \frac{\Pr(\mathcal{D}_M \mid t) \cdot \pi_i(A(t))}{\sum_{s \in \mathcal{D}_M} \Pr(\mathcal{D}_M \mid s) \cdot \pi_i(A(s))}, \quad (1)$$

where $\Pr(\mathcal{D}_M \mid t)$ is the likelihood of observing $\mathcal{D}_M$ given that t is the true record associated with i. Records with the same QI projection receive equal likelihoods and are indistinguishable; thus they have a uniform posterior within their equivalence class.

A common worst-case assumption is that $\mathcal{A}$ knows each target's exact QI values. Then π_i is a point mass on the true assignment $a_i \in \mathrm{dom}(A)$, and the posterior is supported on records with $A(t) = a_i$. Under equal likelihoods, a target in an equivalence class of size k is re-identified with probability $1/k$, recovering the intuition behind k-anonymity [44]. Proposition 1 formalizes this correspondence.

Privacy risk. We quantify the privacy risk of a minimized dataset $\mathcal{D}_M$ using a background-adjusted entropy-gap score. For a target individual $i \in \mathcal{I}$, let $H_i := -\sum_{a \in \mathrm{dom}(A)} \pi_i(a) \log \pi_i(a)$ and $H_i'(\mathcal{D}_M) := -\sum_{t \in \mathcal{D}_M} q_i(t \mid \mathcal{D}_M) \log q_i(t \mid \mathcal{D}_M)$ denote the entropy of the QI background knowledge and the posterior record-

identity entropy, respectively. The *privacy-risk score of* i is

$$\Delta H_i(\mathcal{D}_M) := H_i - H'_i(\mathcal{D}_M).$$

The score increases with H_i and decreases with the posterior record-identity entropy $H'_i(\mathcal{D}_M)$; larger values indicate higher disclosure risk under this definition. Because H_i and $H'_i(\mathcal{D}_M)$ describe different hypothesis spaces, $\Delta H_i(\mathcal{D}_M)$ is a composite risk score rather than a literal Shannon information gain.

At the dataset level, we define *privacy risk* as the maximum score over all target individuals:

$$\text{leak}(\mathcal{D}_M) := \max_{i \in \mathcal{I}} \Delta H_i(\mathcal{D}_M). \tag{2}$$

This worst-case formulation measures the exposure of the most vulnerable individual and serves as our privacy objective.

Proposition 1: *In the exact-QI setting, the risk score* $\text{leak}(\mathcal{D}_M)$ *is monotonically equivalent to k-anonymity-style singling-out risk.* □

Proof sketch: For a target i with known QI values, let $C_i(\mathcal{D}_M)$ be its matching equivalence class and $k_i = |C_i(\mathcal{D}_M)|$. The point-mass prior has $H_i = 0$, while indistinguishability within $C_i(\mathcal{D}_M)$ gives $H'_i(\mathcal{D}_M) = \log k_i$. Hence, $\Delta H_i(\mathcal{D}_M) = -\log k_i = \log(1/k_i)$, a monotone transformation of the singling-out probability $1/k_i$. Maximizing over targets yields $\text{leak}(\mathcal{D}_M) = -\log \min_{i \in \mathcal{I}} k_i$, establishing the equivalence at the dataset level. □

Utility. We quantify *the utility of a minimized dataset* $\mathcal{D}_M$ with respect to a fixed model $\mathcal{M}$ using the negative Expected Generalization Error (EGE) of the model $\mathcal{M}_{\mathcal{D}_M}$ trained on $\mathcal{D}_M$:

$$U(\mathcal{M}_{\mathcal{D}_M}) := -\mathbb{E}_{(x,y)\sim\mathcal{P}}\left[\ell(\mathcal{M}_{\mathcal{D}_M}(x), y)\right], \tag{3}$$

where $\ell(\cdot,\cdot)$ is the loss function and $\mathcal{P}$ is the underlying distribution over feature-label pairs. The EGE measures the expected predictive loss on unseen data; a higher U indicates better preservation of task-relevant information, *i.e.,* higher model utility.

To quantify the utility impact of minimization, we compare target model $\mathcal{M}_{\mathcal{D}_M}$ trained on $\mathcal{D}_M$ with the model $\mathcal{M}_{\mathcal{D}}$ trained on the original dataset $\mathcal{D}$. We define *the utility degradation of* $\mathcal{D}_M$ as

$$\Delta U(\mathcal{D}_M) := U(\mathcal{M}_{\mathcal{D}}) - U(\mathcal{M}_{\mathcal{D}_M}). \tag{4}$$

A minimized dataset is considered *utility-preserving* if $\Delta U(\mathcal{D}_M)$ does not exceed a user-specified tolerance.

3.2 Problem Formulation and Challenges

The privacy and utility metrics above provide a quantitative foundation for training-data minimization. We now formulate the problem.

Problem statement. Given a relational dataset $\mathcal{D}$ with schema $\mathcal{R}$ and a target model $\mathcal{M}$, our goal is to derive a *minimized dataset* $\mathcal{D}_M$ by applying a sequence of minimization operations $t_1, \ldots, t_n$ to $\mathcal{D}$. We state the Data Minimization Problem (DMP) as follows:

- *Input*: A dataset $\mathcal{D}$, a model $\mathcal{M}$, and a utility tolerance $\epsilon > 0$.
- *Output*: A minimized dataset $\mathcal{D}_M = t_n \circ \cdots \circ t_1(\mathcal{D})$.
- *Objective*: Minimize the residual privacy risk $\text{leak}(\mathcal{D}_M)$ while bounding the utility degradation $\Delta U(\mathcal{D}_M)$:

$$\min_{\mathcal{D}_M = t_n \circ \cdots \circ t_1(\mathcal{D})} \quad \text{leak}(\mathcal{D}_M) \quad \text{s.t.} \quad \Delta U(\mathcal{D}_M) \le \epsilon.$$

This formulation captures the goal of training-data minimization: finding the lowest-risk dataset that preserves the required utility.

DMP is nontrivial. Consider the decision version of DMP: given $\mathcal{D}$, $\mathcal{M}$, and thresholds B and ϵ, decide whether there exists a minimized dataset $\mathcal{D}_M$ such that $\text{leak}(\mathcal{D}_M) \le B$ and $\Delta U(\mathcal{D}_M) \le \epsilon$. We assume that, for the model class and evaluation protocol considered, both $\text{leak}(\mathcal{D}_M)$ and $\Delta U(\mathcal{D}_M)$ are computable in PTIME.

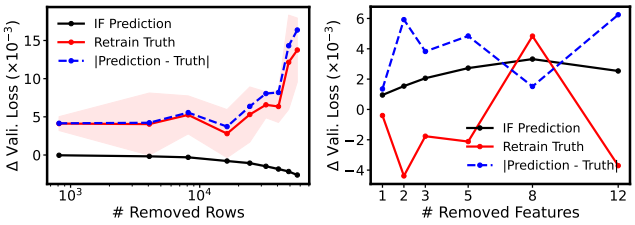


Figure 1: Influence-function estimates vs. actual utility degradation.

Theorem 2: *The decision version of* DMP *is* NP*-complete.* □

Proof sketch: A candidate solution can be verified in PTIME due to PTIME $\text{leak}(\mathcal{D}_M)$ and $\Delta U(\mathcal{D}_M)$ evaluation; hence the problem is in NP. For the lower bound, we reduce from NP-complete Knapsack (cf. [22]): each minimization operation corresponds to an item, its privacy cost to the item weight, and its utility contribution to the item value. Selecting a minimized dataset that satisfies the utility constraint while minimizing privacy exposure is equivalent to selecting a feasible knapsack solution. Thus DMP is NP-complete. □

Challenges. Solving DMP raises three challenges.

(1) Privacy-utility optimization. Privacy risk and utility are tightly coupled: removing or generalizing data may reduce disclosure risk, but can also lose information needed by the model. Moreover, their effects may be non-monotonic: an operation that appears beneficial in isolation may become redundant or harmful after other operations are applied. Hence, finding a Pareto-optimal minimized dataset requires searching a large, interdependent privacy-utility space.

(2) Efficient utility estimation. Retraining the target model for every candidate is prohibitively expensive. Existing influence-based methods are cheaper, but estimate utility locally around a fixed training configuration and become inaccurate when reused across successive operations that change the training data and model.

Example 3: Figure 1 revisits Example 1, comparing influence-function (IF) estimates with the actual utility degradation observed after retraining on `Diabetes`. The left and right panels report cumulative row and feature removals, respectively. In both cases, the estimated and true degradations diverge as operations accumulate, demonstrating the limitations of local influence scores. □

(3) Auditability. Data minimization is not merely an optimization task. Each decision should record its privacy cost, utility contribution, and role in the final privacy-utility tradeoff.

4 TRIM: DATA MINIMIZATION FRAMEWORK

The NP-hardness of DMP motivates the approximate optimization framework TRIM. TRIM combines a retention view, which treats data minimization as progressive information admission, and a ratio-marginal greedy rule, which prioritizes information with the highest utility gain per unit of privacy risk. Together, they support efficient, auditable, and DPIA-aligned training-data minimization.

Retention view. Existing minimization methods follow a *removal view* [21, 45, 46, 50]: starting from the original dataset $\mathcal{D}$, they iteratively remove records or generalize QIs, treating minimization as deletion. Instead, TRIM treats it as a retention problem by adopting

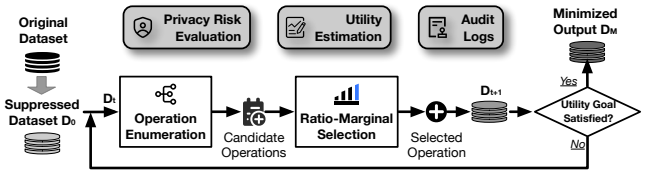


Figure 2: Overview of the TRIM workflow.

a *retention view*. It begins with a minimally exposed dataset $\mathcal{D}_0$, in which only a small label-covering seed of records is retained and all QIs are suppressed. It then progressively restores information whose utility contribution justifies its privacy cost.

The retention view offers two advantages. First, it prioritizes information with the highest utility recovery per unit of privacy cost, building the dataset around data necessary for the task. Second, early iterations operate on small retained datasets, making TRIM orders of magnitude faster than removal-view search (see [1]).

The retention and removal views are equivalent.

Proposition 3: *The retention view of* DMP *is equivalent to the removal view: both admit the same feasible minimized datasets and attain the same optimal objective value.* □

Proof sketch: Each minimized dataset $\mathcal{D}_M$ can be obtained either by removing from $\mathcal{D}$ exactly the information absent from $\mathcal{D}_M$, or by restoring to $\mathcal{D}_0$ exactly the information retained in $\mathcal{D}_M$. Hence, both views induce the same feasible datasets. Since the objective and constraint depend only on $\mathcal{D}_M$, they also have the same optimum. □

Ratio-marginal optimization. Under the retention view, each operation restores information and therefore increases both utility and privacy exposure. To satisfy the data-minimization principle, additional personal data should be retained only when justified by its contribution to the target task. TRIM therefore prioritizes operations according to their *ratio-marginal score*, defined as the utility gain achieved per unit of additional privacy risk.

Let $\mathcal{V}_t$ denote the set of candidate operations applicable to the current dataset $\mathcal{D}_{t-1}$. For each candidate operation $o \in \mathcal{V}_t$, let $\Delta U(o) \geq 0$ denote its marginal utility recovery and $\Delta\mathsf{leak}(o) > 0$ its marginal privacy cost. The *ratio-marginal score of* o is defined as:

$$\rho(o) := \frac{\Delta U(o)}{\Delta\mathsf{leak}(o)} = \frac{U(\mathcal{D}_{t-1} \oplus o) - U(\mathcal{D}_{t-1})}{\mathsf{leak}(\mathcal{D}_{t-1} \oplus o) - \mathsf{leak}(\mathcal{D}_{t-1})}. \tag{5}$$

where $\mathcal{D}_{t-1} \oplus o$ denotes applying o to $\mathcal{D}_{t-1}$. TRIM then selects the operation with the largest ratio-marginal score:

$$o_t^* := \arg\max_{o \in \mathcal{V}_t} \rho(o). \tag{6}$$

The selected operation is used to obtain $\mathcal{D}_t = \mathcal{D}_{t-1} \oplus o_t^*$. The process continues until the utility constraint $\Delta U(\mathcal{D}_t) \leq \epsilon$ is satisfied.

Framework overview. Figure 2 illustrates the workflow of TRIM. Starting from the suppressed dataset $\mathcal{D}_0$, TRIM maintains a sequence of retained datasets $\mathcal{D}_0, \mathcal{D}_1, \ldots, \mathcal{D}_t$, where $\mathcal{D}_t$ denotes the dataset after t retention operations. At each round, TRIM enumerates candidate retention operations, estimates their privacy and utility impacts, selects the best operation based on the ratio-marginal greedy rule, and applies it to obtain $\mathcal{D}_t$ from $\mathcal{D}_{t-1}$. The process terminates once the utility constraint $\Delta U(\mathcal{D}_t) \leq \epsilon$ is satisfied.

Suppressed dataset. Following the retention view, TRIM starts from a suppressed dataset $\mathcal{D}_0$: every QI is generalized to its coarsest level, and only a small label-covering seed is retained. Thus, $\mathcal{D}_0$ has near-minimal privacy exposure, and subsequent information is admitted only when it improves utility. The nonempty seed is the minimal initialization needed to train the current model and estimate candidate utility from the first round.

After $\mathcal{D}_0$ is obtained, TRIM iterates the following three steps.

(1) Operation enumeration. Given the current dataset $\mathcal{D}_{t-1}$, TRIM enumerates candidate retention operations from a bounded search space consisting of retention-class inclusions (horizontal) and attribute refinements (vertical). These operations define the candidate set used to construct $\mathcal{D}_t$ (see Section 5 for details).

(2) Ratio-marginal evaluation. For each candidate o, TRIM calculates its marginal privacy $\Delta\mathsf{leak}(o)$ directly. Estimating the marginal utility $\Delta U(o)$ is much more difficult, which would require retraining the target model for every candidate operation. To avoid this cost, TRIM employs a two-stage utility estimator: a lightweight ranking stage that prioritizes promising candidates, followed by a certification stage that derives tighter utility estimates for the top-ranked operations. This limits expensive certification to the top-ranked operations and substantially reduces evaluation cost (Section 6).

(3) Selection and application. TRIM next applies the ratio-marginal greedy rule to select the most promising operation from the set $\mathcal{V}_t$. Let $o_t^* \in \mathcal{V}_t$ denote the selected operation. TRIM applies o_t^* to the current dataset $\mathcal{D}_{t-1}$, yielding the updated dataset $\mathcal{D}_t = \mathcal{D}_{t-1} \oplus o_t^*$.

The process then proceeds to the next round. When the estimator indicates that $\Delta U(\mathcal{D}_t) \leq \epsilon$, TRIM fully retrains the target model on $\mathcal{D}_t$ to confirm the constraint before terminating. This check adds little overhead because it is triggered only near termination and requires at most one full retraining per round.

Addressing DMP challenges. TRIM directly addresses the three challenges identified in Section 3.

(1) Privacy-utility optimization. TRIM optimizes DMP via the ratio-marginal greedy rule, retaining information only when its utility gain justifies the additional privacy exposure. This directly reflects the DPIA principles of necessity and proportionality. Despite the NP-hardness of DMP, the greedy algorithm runs in PTIME and achieves a $(1 - e^{-(1-\gamma)})$-approximation guarantee (Section 5).

(2) Efficient utility estimation. TRIM avoids exhaustive search and repeated retraining through structured candidate enumeration and a two-stage utility estimator. A lightweight ranking stage filters candidates, while a certification stage derives tighter utility estimates for a small subset of promising operations (Section 6).

(3) Auditability. TRIM logs each committed operation with its utility gain and privacy cost. The former documents necessity, while ratio-marginal selection documents proportionality by showing that the operation offers the greatest utility recovery per unit privacy cost among the candidates considered. These support DPIA review.

5 RETENTION-BASED SEARCH SPACE

Section 4 introduced the retention view and ratio-marginal optimization. This section develops the search space over which these ideas operate. Specifically, we define retention classes and the retention operations that progressively construct the minimized dataset.

Overview. At round t, TRIM enumerates a set $\mathcal{V}_t$ of candidate

retention operations applicable to the current dataset $\mathcal{D}_{t-1}$. These operations define the search space explored by ratio-marginal optimization. To support DPIA-compliant minimization, a key challenge is to make the search space both tractable and auditable.

(1) Tractability. Enumeration must avoid the exponential space of record subsets and arbitrary QI groupings. To address this challenge, TRIM exposes a bounded set $\mathcal{V}_t$ of structured local operations, each restoring an interpretable unit of information.

(2) Semantic coherence. Each operation must have a clear semantic meaning that can later be justified in the audit log. Horizontal operations admit records that naturally belong together, while vertical operations follow predefined generalization hierarchies such as disease taxonomies, rather than ad hoc partitions.

Because suppression affects both record coverage and attribute granularity, TRIM supports two complementary operation families: retention-class inclusion and attribute refinement. Together they define a bounded and auditable search space for ratio-marginal optimization. We first introduce retention classes, then present the two operation families and the resulting enumeration procedure.

Retention classes. To coordinate horizontal and vertical retention, TRIM groups records by their generalized QIs. For each record $r \in \mathcal{D}$, let $A_{t-1}(r)$ denote its *QI signature*, *i.e.,* the generalized QI values under the representation used by the current dataset $\mathcal{D}_{t-1}$. The *retention class* of r is $C_{\text{ret}}(r) = \{\, r' \in \mathcal{D} \mid A_{t-1}(r') = A_{t-1}(r) \,\}$, *i.e.,* the maximal set of records in $\mathcal{D}$ that share this signature $C_{\text{ret}}(r)$. Hence, at round t, retention classes partition $\mathcal{D}$ into equivalence classes by equality of current QI signatures.

Each class contains records indistinguishable under the current representation. It is analogous to a k-anonymity equivalence class [44], but evolves as the QI representation changes. Since class sizes directly affect re-identification risk (Proposition 1), retention classes form the basic units of both horizontal and vertical retention.

Retention-class inclusion. This is the horizontal operation in TRIM. At a fixed QI granularity, the current dataset $\mathcal{D}_{t-1}$ may omit regions of the data distribution that are important for the target task. The goal of inclusion is to restore such missing coverage. To construct inclusion candidates, TRIM proceeds in two steps.

(1) Filtering. TRIM first identifies not-yet-admitted records that a reference model $\mathcal{M}^*$ classifies correctly but that the current model, trained on $\mathcal{D}_{t-1}$, supports poorly. Such records are identified, *e.g.,* from small prediction margins, high uncertainty, or disagreement between the current model and $\mathcal{M}^*$. These records indicate regions where additional coverage may improve utility.

(2) Retention-class enumeration. For each qualified record passing the filter, TRIM retrieves its current QI signature and enumerates the not-yet-admitted members of the corresponding retention class as a candidate operation. Admission thus occurs at the retention-class level rather than the record level. If multiple qualified records belong to the same class, the class is enumerated only once.

This preserves the anonymity of the current QI representation while avoiding an exponential search over record subsets. Since retention classes are defined by exact equality of current QI signatures, the qualified records partition into disjoint classes that can be enumerated in one linear scan. Denote by F_t the filtered, not-yet-admitted records, and by $\Pi_t^{\text{filt}} = \{A_{t-1}(x) \mid x \in F_t\}$ their distinct current QI signatures. TRIM considers at most one inclusion operation per signature, yielding $|\Pi_t^{\text{filt}}|$ candidates instead of the $2^{|\mathcal{D}|}$ subsets induced by arbitrary record-level admission. Moreover, $|\Pi_t^{\text{filt}}| \le |F_t| \le |\mathcal{D}|$ typically because filtering retains only a small subset of records, while QI generalization maps many of them to the same signature. The resulting candidates are evaluated by the utility estimator and the ratio-marginal selection rule.

For a DPIA, the qualifying record identifies where additional support is needed, while class-level admission documents what personal data is retained and why it is necessary and proportionate.

Attribute refinement. This is the vertical operation in TRIM. Each operation advances one QI in $\mathcal{D}_{t-1}$ by a single level in a precomputed generalization hierarchy and restores a finer-grained representation. Unlike retention-class inclusion, refinement may increase re-identification risk by splitting existing classes without adding records. To address this, TRIM couples each refinement with retention-class swaps in a *refine-and-swap* operation.

Offline hierarchy construction. Before refinement, TRIM constructs a generalization hierarchy for each QI in the schema $\mathcal{R}$ of $\mathcal{D}$. Given an attribute's name, type, observed domain, and optional metadata (*e.g.,* categorical dictionaries), an LLM generates semantically meaningful generalization levels from the finest values to full suppression, denoted by '$*$'. The levels are organized as a rooted tree, with '$*$' as the root and the finest observed values as leaves.

This one-time offline preprocessing requires $O(|\mathcal{R}|)$ LLM calls, and the resulting hierarchies are reused throughout the online workflow. Their finite fan-out bounds the candidate space, while domain-recognized levels make refinements interpretable and avoid opaque statistical partitions. Unlike performance-driven hierarchies [50], TRIM uses only semantically meaningful levels, improving auditability with negligible impact on privacy and utility (Section 7).

Refine-and-swap. TRIM enumerates refinement targets directly from the generalization hierarchies. At round t, each retained QI q in $\mathcal{D}_{t-1}$ is represented at some level ℓ of its tree. If ℓ is not the finest level, a *refinement target* $q : \ell \mapsto \ell'$ advances it to a child level ℓ'.

A bare refinement may violate the structural properties required by ratio-marginal optimization: splitting retention classes can create privacy spikes, trigger utility-reducing repairs, and make privacy and utility changes non-monotone. For example, refining `diagnosis` from the section to the family level in Table 1 may isolate the only patient in their 80s with a hypertensive disease (t_4).

TRIM thus uses *refine-and-swap*, an atomic operation that couples refinement with class-level repair. It removes newly exposed high-risk fragments and admits low-risk classes to keep utility non-decreasing. A resulting operation that strictly reduces leak without utility loss is committed immediately as a *Pareto improvement*, without consuming privacy budget. Algorithm 1 gives the details.

Algorithm. Algorithm 1 constructs a refine-and-swap candidate for a refinement target. It first advances QI q from level ℓ to its child level ℓ', fragmenting the affected retention classes (line 1). Classes whose risk exceeds τ_t are then identified and evicted (lines 2–3). To recover the resulting utility loss, it admits the lowest-risk eligible classes until utility returns to at least its pre-refinement level (line 4). The resulting refine-and-swap operation is then formed

Algorithm 1: TRIM: refine-and-swap at round t.

Input: Retained dataset $\mathcal{D}_{t-1}$, original dataset $\mathcal{D}$, a refinement target (one tree step $q : \ell \mapsto \ell'$), a risk ceiling $\tau_t \geq \mathsf{leak}(\mathcal{D}_{t-1})$.
Output: A refine-and-swap o for $\mathcal{V}_t^{\text{attr}}$, or a committed Pareto improvement.

1 $\mathcal{D}' \leftarrow \mathcal{D}_{t-1}$ with QI q advanced from ℓ to ℓ';
2 $H \leftarrow \{C \mid C$ is a retained class in $\mathcal{D}',\ \mathsf{leak}(C) > \tau_t\}$; /* Flag risky classes */
3 $\mathcal{D}' \leftarrow \mathcal{D}' \setminus H$; /* evict, capping risk at τ_t */
4 **while** $U(\mathcal{D}') < U(\mathcal{D}_{t-1})$ and some retention class fits within τ_t **do**
5 $\quad C^+ \leftarrow$ a lowest-risk not-yet-admitted class with $\mathsf{leak}(\mathcal{D}' \cup C^+) \leq \tau_t$;
6 $\quad \mathcal{D}' \leftarrow \mathcal{D}' \cup C^+$; /* admit a low-risk substitute */
7 $o \leftarrow (\mathcal{D}_{t-1} \mapsto \mathcal{D}')$; /* the composite candidate */
8 **if** $\Delta\mathsf{leak}(o) < 0$ **then return** committed o;
9 **return** o ;

(line 7) and its marginal leakage evaluated. If it strictly reduces leak without reducing utility, TRIM commits it immediately as a Pareto improvement (line 8); otherwise, it is emitted into $\mathcal{V}_t^{\text{attr}}$ (line 9).

Setting τ_t. The ceiling τ_t is a free parameter of refine-and-swap. It controls the maximum residual re-identification risk permitted at round t. To preserve monotonicity, $\tau_t \geq \mathsf{leak}(\mathcal{D}_{t-1})$ is required. By default, we set $\tau_t = \mathsf{leak}(\mathcal{D}_{t-1})$, evicting every fragment whose risk exceeds the current worst case. A higher DPIA-approved threshold permits limited additional risk in exchange for fewer evictions and compensating admissions. Section 7 evaluates this tradeoff.

Example 4: Figure 3 illustrates the operations examined on `Diabetes` of Example 1. Suppose earlier operations have retained `age` at the decade level and `diagnosis` at the section level, where codes 390–459 are generalized to *Circulatory*. At this granularity, t_4 is retained and shares the signature (80s, Circulatory) with t_3 and t_5.

(1) Retention-class inclusion. Suppose filtering identifies t_1 and t_5. Rather than admitting them individually, TRIM enumerates their not-yet-admitted retention classes as candidates: $\{t_1, t_2\}$ with signature (70s, Circulatory), and $\{t_3, t_5\}$ with signature (80s, Circulatory). The former is selected and admitted in this round.

(2) Attribute refinement. In the next round, TRIM refines `diagnosis` from the section to the family level. Records t_1 and t_2 remain grouped as (70s, Ischemic heart diseases), while t_4 becomes the singleton (80s, Hypertensive diseases), whose risk exceeds τ_t. Refine-and-swap therefore evicts $\{t_4\}$ and admits the low-risk class $\{t_3, t_5\}$, with signature (80s, Other heart diseases), to recover utility.

The swap from $\{t_4\}$ to $\{t_3, t_5\}$ is committed directly as a Pareto improvement. The resulting dataset retains t_1, t_2, t_3, t_5 with decade-level ages, family-level diagnoses, and no distinctive record. Thus, t_4 is retained while protected by a larger class and evicted once refinement makes it unique, an outcome that neither vertical nor horizontal minimization alone achieves in Example 1. □

Properties. The ratio-marginal framework requires candidate operations to exhibit diminishing utility returns and increasing marginal privacy cost. Proposition 4 shows that the operations enumerated by TRIM satisfy these properties, thus establishing the conditions needed for the approximation guarantee of Theorem 5.

Proposition 4: *Let $\mathcal{V}_t = \mathcal{V}_t^{\text{attr}} \cup \mathcal{V}_t^{\text{cls}}$ be the operations enumerated by* TRIM *at round t. Then every candidate $o \in \mathcal{V}_t$ satisfies $\Delta U(o) \geq 0$ and $\Delta\mathsf{leak}(o) \geq 0$. Moreover, over sets of operations, the marginal utility recovery $\Delta U(\cdot)$ is a monotone submodular function and the marginal privacy cost $\Delta\mathsf{leak}(\cdot)$ is a monotone supermodular function.* □

Proof sketch: Retention-class inclusion only admits information, yielding $\Delta U(o) \geq 0$ and $\Delta\mathsf{leak}(o) \geq 0$. For refine-and-swap, the admission phase guarantees $\Delta U(o) \geq 0$, while leakage-reducing candidates are committed as Pareto improvements and excluded from $\mathcal{V}_t$. Thus every remaining candidate satisfies $\Delta\mathsf{leak}(o) \geq 0$.

Submodularity of $\Delta U(\cdot)$ follows from diminishing utility returns, while supermodularity of $\Delta\mathsf{leak}(\cdot)$ follows from progressive fragmentation of retention classes (see [1] for a full proof). □

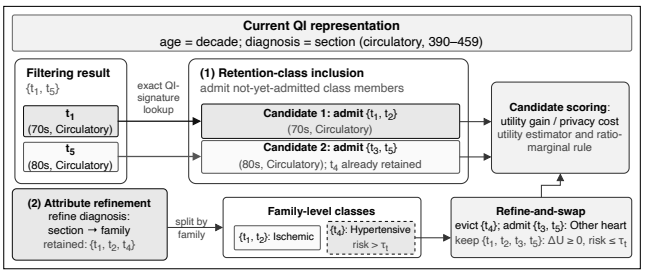


Figure 3: Enumeration example.

These structural properties allow the retention view to be cast as a submodular optimization problem, yielding the guarantee below, which provides the theoretical basis for ratio-marginal optimization.

Theorem 5: *Let $\mathcal{D}_M$ be an optimal solution to* DMP *and let $\mathcal{D}_T$ be the dataset returned by* TRIM. *Then $\mathcal{D}_T$ achieves a constant-factor approximation:* $\Delta\mathsf{leak}(\mathcal{D}_T) \leq \frac{\log(1/\delta_U)+1/\delta_U}{1-\gamma}\Delta\mathsf{leak}(\mathcal{D}_M)$, *where $\gamma \in [0, 1)$ is the curvature of the privacy-risk function and $\delta_U \in (0, 1]$ is the resolution of the normalized utility metric.* □

Proof sketch: By Proposition 4, the retention view is a submodular utility-cover problem with a supermodular privacy cost. The optimal operations recover the current utility gap at total marginal privacy cost at most $\Delta\mathsf{leak}(\mathcal{D}_M)/(1-\gamma)$. The ratio-marginal charges then telescope to $\log(1/\delta_U)/(1-\gamma)$, while practical utility costs bound the final charge by $1/((1-\gamma)\delta_U)$. Pareto-improving swaps have nonpositive privacy cost, yielding the stated factor (see [1]). □

The remaining challenge is to estimate $\Delta U(o)$ efficiently for the candidates in $\mathcal{V}_t$. Thus, Section 6 develops a two-stage estimator.

6 EFFICIENT UTILITY ESTIMATION

This section develops an efficient two-stage estimator.

Overview. Computing the utility gain $\Delta U(o)$ exactly would require retraining the target model for every candidate operation $o \in \mathcal{V}_t$, which is prohibitively expensive. To avoid this cost, TRIM employs a two-stage estimator. The first stage ranks candidates using gradient information alone and retains only the top-k. The second stage certifies their utility recovery using a lightweight proxy model. Together, the two stages provide the utility estimates needed for ratio-marginal optimization while avoiding repeated retraining.

Estimating $\Delta U(o)$. The greedy rule does not require exact utility values for all candidates. Instead, it requires estimates that (a) evaluate both retention-class inclusions and attribute refinements under a unified measure, and (b) preserve the ordering needed for greedy selection while providing a certified bound for the final choice.

Existing estimators satisfy neither requirement. Influence functions and approximate unlearning [8, 29] rely on local linearity assumptions that break down for refinements and class-level swaps. Per-record valuation methods such as Data Shapley [24] cannot naturally evaluate operations that rewrite existing records or capture interactions among jointly admitted data. Moreover, they provide no certified bound for the final selection. TRIM addresses these

limitations with the following two-stage estimator.

A two-stage estimator. TRIM satisfies both requirements without retraining the target model $\mathcal{M}$ for every candidate.

(1) *Ranking.* Learning Gradient Attention (LGA) scores each candidate from training gradients alone and retains the top-k. The gradient change between $\mathcal{D}_{t-1}$ and $\mathcal{D}_{t-1} \oplus o$ provides a unified score for both inclusion and refinement operations.
(2) *Certification.* For each top-k candidate, a capacity-constrained *proxy model* is retrained in place of $\mathcal{M}$ to estimate its utility recovery. The bound established below quantifies the error of this estimate, which is then used for the final greedy selection.

We first present the ranking stage based on Learning Gradient Attention (LGA), followed by the proxy-based certification stage, and then establish their ranking and estimation guarantees.

(1) Ranking. We rank candidates based on a gradient-based score.

LGA score. Fix the parameters θ of $\mathcal{M}$ and the current dataset $\mathcal{D}$ (round subscripts omitted). For a candidate operation o, let $\mathcal{D} \oplus o$ denote the provisional dataset obtained by applying o. LGA estimates the utility contribution of o without retraining by aligning the validation gradient with *the marginal training gradient* induced by o:

$$\mathsf{LGA}_\theta(\mathcal{D}, o) = \left\langle \nabla_\theta L_{\mathsf{v}}(\theta),\ \nabla_\theta L_{\mathsf{train}}(\theta, \mathcal{D} \oplus o) - \nabla_\theta L_{\mathsf{train}}(\theta, \mathcal{D}) \right\rangle.$$

The first term is the direction that reduces validation loss. The second is the gradient gap induced by o, which isolates its marginal learning signal. Consequently, $-\eta\, \mathsf{LGA}_\theta(\mathcal{D}, o)$ approximates the first-order change in validation loss after applying o: a large positive score indicates greater utility recovery.

LGA is inspired by gradient matching [54, 60], but uses the dataset-level gradient *gap* induced by an operation rather than the gradient of a sample or dataset. This marginal formulation provides a unified score for both retention-class inclusion and attribute refinement, whereas record-level influence estimators [41, 55] naturally handle record additions but not attribute rewrites.

Top-k selection. LGA induces a total order on the candidates in $\mathcal{V}_t$, with higher scores indicating greater utility recovery. TRIM retains only the top-k candidates and discards the rest before any retraining. The retained candidates proceed to certification, which focuses its computation on the operations most likely to be admitted.

(2) Proxy certification. For each of the top-k candidates retained by LGA, TRIM estimates $\Delta U(o)$ using a proxy model.

Proxy model. A *capacity-constrained proxy* $\bar{\mathcal{M}}$ belongs to the same model family as $\mathcal{M}$ but has reduced capacity, *e.g.*, fewer layers or narrower width. This is intended to preserve comparable learning behavior while making retraining much cheaper. TRIM retrains $\bar{\mathcal{M}}$ on both $\mathcal{D}_{t-1}$ and $\mathcal{D}_{t-1} \oplus o$, and uses the resulting utility recovery

$$\overline{\Delta U}(o) = \bar{U}(\mathcal{D}_{t-1} \oplus o) - \bar{U}(\mathcal{D}_{t-1})$$

as the estimate of $\Delta U(o)$. The greedy rule then operates on these estimates without retraining $\mathcal{M}$. As shown next, the estimation error decreases as the proxy capacity and retained dataset size increase.

Greedy selection. The two stages produce a certified estimate $\overline{\Delta U}(o)$ for each top-k candidate. TRIM then applies the ratio-marginal rule using these estimates, committing the candidate that maximizes $\overline{\Delta U}(o)/\Delta\mathsf{leak}(o)$. This quantity approximates the ratio-marginal score $\rho(o)$ used by the optimization framework.

Properties. The two-stage estimator replaces repeated retraining of $\mathcal{M}$ with gradient-based ranking and proxy-model certification. We show that LGA preserves the candidate ordering needed for top-k screening, while the proxy yields a tight estimate of utility recovery. Together, these properties preserve the quality of ratio-marginal optimization while substantially reducing computation.

We first analyze LGA ranking. As a first-order approximation of utility impact, LGA preserves the ordering of sufficiently separated candidates, and thus justifies top-k screening.

Theorem 6: *Assume that the validation loss is smooth, the training gradients are bounded, and the learning rate η is sufficiently small. Then for any two operations o_1 and o_2, $\mathsf{LGA}_\theta(\mathcal{D}, o_1)$ and $\mathsf{LGA}_\theta(\mathcal{D}, o_2)$ induce the same ordering as their true utility gains whenever the gap between those gains exceeds $O(\eta^2)$.* □

Proof sketch: A first-order Taylor expansion of the validation loss around θ shows that, relative to one gradient step on $\mathcal{D}$, one step on $\mathcal{D} \oplus o$ changes the validation loss by $-\eta\, \mathsf{LGA}_\theta(\mathcal{D}, o)$ plus a remainder of order $O(\eta^2)$. Hence, up to this second-order error, the LGA score captures the marginal utility impact of o. Since η is shared across all candidates, the separation condition ensures that the remainder terms cannot reverse the ordering of any pair whose utility gains differ by more than $O(\eta^2)$ (see full proof in [1]). □

We next analyze the certification stage. The proxy provides an asymptotically tight estimate of utility recovery as the retained dataset and proxy capacity increase, and thus justifies the use of a lightweight proxy in place of retraining $\mathcal{M}$ in practice.

Theorem 7: *Assume the loss is bounded. Then, with high probability, the true utility recovery of any candidate operation o satisfies*

$$\Delta U(o) \ \geq\ \overline{\Delta U}(o) \ -\ O\left(\frac{|o|}{n}\frac{d}{w} + \frac{1}{\sqrt{n}}\right), \tag{7}$$

where $n = |\mathcal{D}_{t-1} \oplus o|$, $|o|$ is the number of single-record changes in o, d is the input dimension, and w is the proxy width. □

Proof sketch: A difference-of-differences decomposition cancels persistent target–proxy bias and separates the recovery error into validation noise and the operation-induced change in this bias. Hoeffding's inequality controls the validation noise. Telescoping the local-response condition over the $|o|$ neighboring record changes controls the bias change. These yield Equation (7) (see [1]). □

The estimate is accurate because utility recovery is measured as a difference between two retrainings, one on $\mathcal{D}_{t-1}$ and one on $\mathcal{D}_{t-1} \oplus o$. As $\bar{\mathcal{M}}$ and $\mathcal{M}$ are in the same model family, much of the proxy error cancels, leaving only the correction term in Equation (7).

Together, Theorems 6 and 7 justify using $\overline{\Delta U}(o)$ as an asymptotically accurate approximation of $\Delta U(o)$ in ratio-marginal optimization. Because privacy leakage is computed exactly, the correction term in Equation (7) governs the deviation from the exact ratio-marginal score. Thus, these results characterize when the estimated score is a faithful substitute for the exact score used in Theorem 5.

Complexity. Ranking is inexpensive: each candidate requires only an incremental gradient computation at the current parameters, for a total $\mathsf{Cost}_{\mathsf{LGA}}$. Certification retrains the lightweight proxy model $\bar{\mathcal{M}}$ for only the top-k candidates, adding $\mathsf{Cost}_{\mathsf{proxy}}$, where typically

Table 3: Summary of used datasets.

Dataset	#Rows	#Features	Type	Prediction Task
Income	183,941	10	real	Personal income (>$50k)
Diabetes	101,766	47	real	Hospital 30-day readmission prediction
PubCov	152,676	19	real	Public health coverage (covered or not)
BM	45,211	16	real	Term-deposit subscription
BNG	1,000,000	21	synthetic	Credit risk classification (good or bad)

$k \ll |\mathcal{V}_t|$. Thus, TRIM replaces $|\mathcal{V}_t|$ retrainings of the target model per round with gradient evaluations and only k proxy retrainings.

7 EXPERIMENTAL STUDY

Using real-world and synthetic datasets, this section experimentally evaluates TRIM for its privacy-utility tradeoff, scalability, utility estimator, key design choices, and DPIA-style audit records.

Experimental settings. We start with the settings.

Datasets. We evaluated TRIM on four real-world datasets and one synthetic dataset, summarized in Table 3. Income and PubCov are U.S. Census-based prediction tasks from [16]. Diabetes [51] contains 130 U.S. hospitals records for 30-day readmission prediction and includes QIs such as age and diagnosis. BM [36] captures Portuguese bank telemarketing campaigns. BNG (credit-g) [40] is a Bayesian-network-generated German credit-risk benchmark.

For each dataset, we created stratified 70/15/15 training, validation, and test splits, and evaluated all methods on the same splits.

Downstream models. We used two common tabular classifiers: a multilayer perceptron (MLP) and XGBoost [14]. XGBoost is used only as a downstream target model, not for LGA scoring, to test whether MLP-based LGA generalizes across model architectures.

Baselines. We implemented TRIM in ~3k LOC of Python.

We compared TRIM with four representative removal-based baselines: (1) *Horizontal-only minimization*: FIDO [45], which suppresses samples under a validation-loss bound. (2) *Vertical-only minimization*: FeatSel [46], which selects QI subsets by exhaustive search, and PAT [50], which generalizes QIs using a learned tree minimizer. (3) *Hybrid minimization*: EntryWise [21], which suppresses data at the sample-feature level under a utility constraint.

We also tested several TRIM variants for ablation (see Exp-6).

Metrics. For each minimized dataset $\mathcal{D}_M$, we report privacy risk $\mathsf{leak}(\mathcal{D}_M)$, tail risk $\mathsf{leak}_T(\mathcal{D}_M) = \mathrm{P99}_{i\in I}\Delta H_i(\mathcal{D}_M)$, and utility degradation $\Delta U(\mathcal{D}_M)$. Here leak_T captures the 99th percentile of individual privacy risk, complementing the worst-case risk leak.

Configurations. By default, we set the round-t risk ceiling to $\tau_t = \mathsf{leak}(\mathcal{D}_{t-1})$ and used XGBoost as the downstream model. The initial $\mathcal{D}_0$ is a fixed-seed 1.2% sample of the training rows with all QIs generalized to their coarsest levels. Both stages of the utility estimator use logistic regression, independent of the downstream model. The proxy stage certifies the top $k = 7$ LGA-ranked candidates.

All experiments ran on a server with two Intel Xeon Gold 6254 3.10 GHz CPUs, 64 GB DDR4 RAM, and an NVIDIA Tesla V100 32 GB GPU, using Ubuntu 22.04.2 LTS, Linux 5.15.0, CUDA 12.2, Python 3.10.19, and PyTorch 2.9.1. All methods used identical environments, data splits, and random seeds within each run. We repeated each experiment five times and report the mean. Representative results are shown; the remaining results are consistent.

Experimental results. We report six experiments that evaluate TRIM's privacy-utility tradeoff (Exp-1), empirical risk mitigation (Exp-2), scalability (Exp-3), audit logs (Exp-4), hyperparameter sensitivity (Exp-5), and ablations of core system components (Exp-6).

Exp-1: Privacy vs. utility. Figure 4 reports the privacy-utility tradeoff for each method over all datasets and downstream models.

Pareto optimality. At the same utility degradation, TRIM consistently attains the lowest worst-case privacy risk leak across all dataset-model settings. It yields the strongest Pareto frontier.

(1) Figures 4a–4f show that TRIM improves the privacy-utility tradeoff over all baselines. Under stringent utility tolerances (*e.g.,* $\Delta U \leq 0.01$), TRIM already reduces worst-case leakage, while PAT, the strongest baseline on most datasets, often leaves it unreduced (*i.e.,* leak = 0). At the same leak level, PAT incurs 1.55–5.66× and 3.38–6.64× as much utility degradation as TRIM for MLP and XGBoost, respectively. It approaches TRIM only under very loose utility tolerances. This is an endpoint effect: once the utility budget permits near-complete QI generalization, both methods collapse most QIs to their coarsest levels, so their frontiers meet near the fully generalized baseline. Thus, PAT's endpoint convergence does not imply parity under practical utility budgets.

(2) Figures 4g–4h show that on BM, FeatSel and PAT achieve Pareto frontiers closer to that of TRIM. The main reason is that the useful signal in BM is less concentrated in fine-grained personal attributes. Many predictive variables are non-QIs (*e.g.,* contact channel and campaign history), while the useful QIs are mostly binary indicators (*e.g.,* housing/personal loans). Thus, vertical baselines can preserve most task utility with only coarse QI exposure, leaving less room for TRIM's additional horizontal suppression to improve the frontier.

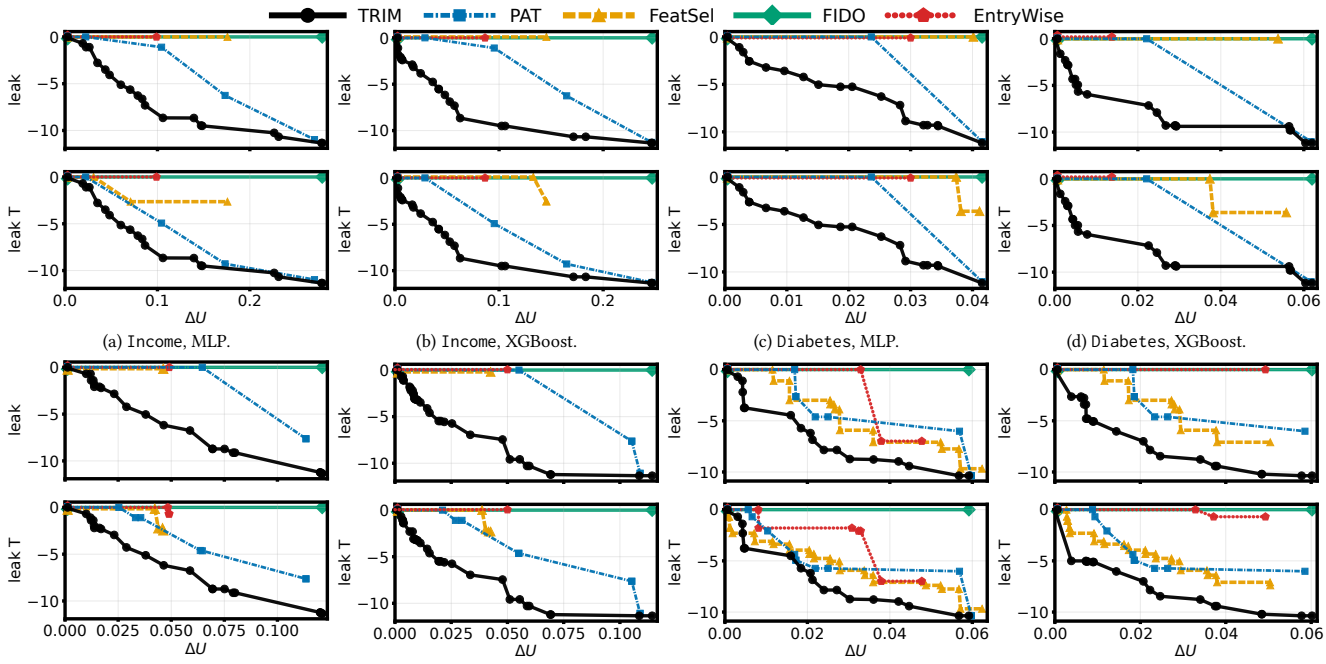
(3) FIDO and EntryWise cannot protect high-risk records, since they optimize indirect targets (*e.g.,* the number of retained records/attributes) rather than mitigating worst-case privacy risk.

(4) As shown in Figures 4a–4h, TRIM achieves similar privacy-utility frontiers on MLP and XGBoost: leak varies by at most 10.9% across the two downstream models. Thus, performance is driven mainly by the released representation and utility budget, not by overfitting to a particular model architecture.

Tail privacy. The 99th-percentile risk leak_T follows the same overall trend as the leak-utility frontier. The gap between TRIM and PAT/FeatSel is smaller, because leak_T discounts the most exposed 1% of records, where vertical-only generalization is weakest. Even under this less sensitive metric, TRIM consistently achieves lower tail risk at small to moderate utility tolerances with XGBoost.

Flexibility. TRIM traces a smooth privacy-utility curve across the achieved utility degradation (Figures 4a–4h), letting an operator choose a DPIA-approved operating point anywhere along this range, flexibly adapting to different levels of regulatory requirements. FeatSel yields only one operating point, and PAT's parameter α specifies neither a privacy nor a utility budget.

Interplay with DP. We evaluated TRIM as a preprocessing step for DPSGD [2], which limits membership-inference by adding noise during training. Table 4 compares TRIM against training on the original dataset and preprocessing with KAnon [44]. We configure KAnon and TRIM to have the same worst-case privacy risk. Across



(a) Income, MLP. (b) Income, XGBoost. (c) Diabetes, MLP. (d) Diabetes, XGBoost.

(e) PubCov, MLP. (f) PubCov, XGBoost. (g) BM, MLP. (h) BM, XGBoost.

Figure 4: Tradeoffs between privacy risk and utility degradation.

Table 4: Model utility with DPSGD training (Income, MLP).

DPSGD e^{ϵ}	No Noise	5	2	1.5
Original Dataset	$\Delta U_0 = 0$	$\Delta U_1 = 0.055$	$\Delta U_2 = 0.060$	$\Delta U_3 = 0.068$
TRIM (leak = −5.46)	$\Delta U_0 + 0.079$	$\Delta U_1 + 0.039$	$\Delta U_2 + 0.041$	$\Delta U_3 + 0.045$
KAnon ($k = 235$)	$\Delta U_0 + 0.104$	$\Delta U_1 + 0.070$	$\Delta U_2 + 0.084$	$\Delta U_3 + 0.102$

the training settings, TRIM incurs 2.5–5.7 percentage points less utility degradation than KAnon. When DP noise is enabled, TRIM adds only 3.9–4.5 points of utility degradation over the corresponding model trained on the original dataset, compared with 7.0–10.2 points for KAnon. Moreover, TRIM's advantage over KAnon widens from 3.1 to 5.7 points as e^{ϵ} decreases from 5 to 1.5, showing that DPIA-oriented data minimization complements DP training.

Exp-2: Empirical risk mitigation. We next study how TRIM protects against empirical privacy attacks on Income under $\epsilon = 0.05$.

Linkability and attribute inference attacks. Figure 5a compares empirical attack errors on the minimized datasets produced by TRIM and all baselines. For linkability, we report how often the adversary incorrectly decides whether two partial records belong to the same individual. For attribute inference, we evaluate how often the adversary incorrectly reconstructs a sensitive value. TRIM increases reconstruction error by $\geq 20.7\%$ and linkage error by $\geq 41.8\%$ relative to all baselines, showing that TRIM mitigates empirical attacks.

Operational relevance of leak. We assess whether leak tracks the success of realizable attacks. We conduct re-identification and reconstruction attacks against TRIM outputs spanning a range of measured leak values and report the corresponding error rates. Figure 5b shows that lower leak yields a less concentrated adversarial posterior and monotonically higher error rates for both attacks. Thus, leak serves as a conservative indicator of operational privacy risk.

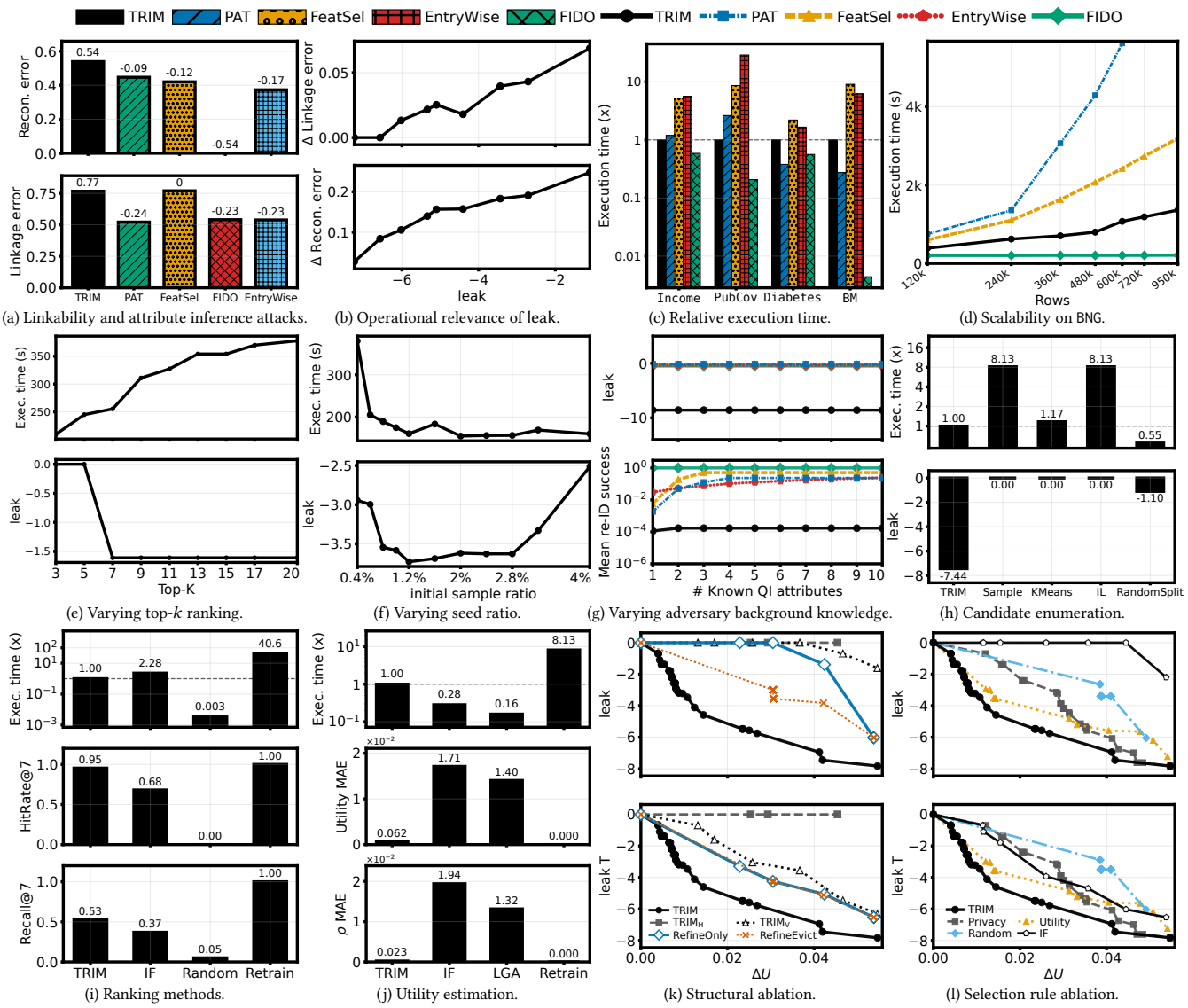
Exp-3: Efficiency. We compare the execution time of all methods.

Execution time. Figure 5c reports the execution time on the real datasets. For TRIM, we measure the time required to first satisfy $\epsilon = 0.05$ on Income and PubCov, 0.02 on Diabetes, and 0.005 on BM. TRIM is 1.02–2.62× faster than PAT on Income and PubCov; PAT is 27.4% and 38.0% faster on BM and Diabetes, respectively, as they have fewer numeric attributes, easing generalization learning. Nevertheless, TRIM achieves a better privacy-utility tradeoff. FeatSel and EntryWise retrain the model per candidate and are much slower, taking 2.20–9.03× and 1.66–28.82× as long as TRIM, respectively. FIDO is faster, but incurs high worst-case risk.

Scalability. To isolate data-size effects, we use synthetic BNG instances with fixed distribution and varying scale factors. Figure 5d shows that TRIM's runtime grows sublinearly with the number of rows: when the dataset size grows from 120k to 600k rows (5×), TRIM takes 2.8× longer, while PAT and FeatSel are 7.5× and 4.1× slower, respectively. This is because each round enumerates at most $|\Pi_t^{\text{filt}}|$ retention classes, rather than arbitrary record subsets, and many rows share the same QI signature. Moreover, TRIM starts from a small seed and admits data gradually, so early rounds operate on a small working set. EntryWise is the least efficient and exhausts the 6-hour budget on datasets larger than 120k rows.

Exp-4: Auditability. We evaluate auditability on Income with $\epsilon = 0.05$. For each run, TRIM logs the declared task, risk and utility policies, estimator configuration, ordered dataset states, and each operation's scope, conservative utility bound, exact leakage change, and decision rule. The excerpt below shows two commit paths: ratio-marginal selection and an off-budget Pareto improvement.

The header fixes the run's purpose, utility tolerance, and risk policy. In log1, RC-7A3 ranks second in preliminary screening but is selected because its $\rho = 0.4266$ exceeds the displayed alternatives' scores. Its positive utility bound supports *necessity*, while its maximum ρ documents *proportionality*. In log2, TRIM refines SCHL, evicts RC-91F/RC-C24, and admits RC-4B8/RC-6D1. The resulting



(a) Linkability and attribute inference attacks. (b) Operational relevance of leak. (c) Relative execution time. (d) Scalability on BNG.

(e) Varying top-k ranking. (f) Varying seed ratio. (g) Varying adversary background knowledge. (h) Candidate enumeration.

(i) Ranking methods. (j) Utility estimation. (k) Structural ablation. (l) Selection rule ablation.

Figure 5: Efficiency, effectiveness, hyperparameter sensitivity, and privacy diagnostics of TRIM.

utility gain and 1.936 leakage reduction justify an off-budget Pareto commit, for which $\rho(o)$ is inapplicable. The terminal record certifies $\Delta U(\mathcal{D}_T) \leq \epsilon$; taken together, the ordered records support replayable DPIA review without establishing legal compliance.

```
[run] data=Income purpose=income>50K target=XGBoost epsilon=5.000e-2 top_k=7
[log1] action=SELECT rank=2 op=retention-class-inclusion target=RC-7A3
       scope={same-QI-signature; all-class-members}
       utility_lb=2.306e-2 leakage_delta=+5.406e-2 rho=4.266e-1 reason=max-rho
       alt={rank=1; op=refine-and-swap; target=AGEP;
            utility_lb=1.279e-3; leakage_delta=+1.800e-2; rho=7.106e-2}
       alt={rank=3; op=retention-class-inclusion; target=RC-2F9;
            utility_lb=8.410e-3; leakage_delta=+1.640e-1; rho=5.128e-2}
[log2] action=PARETO-COMMIT op=refine-and-swap target=SCHL
       step=degree-group->education-code tau=leak(D[8])
       evict={RC-91F,RC-C24; class-risk>tau}
       admit={RC-4B8,RC-6D1; lowest-risk} until=utility>=U(D[8])
       utility_lb=1.095e-1 leakage_delta=-1.936e+0 rho=NA reason=Pareto
[stop] state=D[T] utility_test=DeltaU(D[T])<=epsilon result=PASS
```

Exp-5: Sensitivity to hyperparameters. Beyond the utility tolerance ϵ, we vary the search hyperparameter of TRIM on Income around its default and find that TRIM is robust.

Varying certification budget k. Figure 5e varies k from 3 to 20. Increasing k improves privacy when $k \leq 7$, reducing leak from 0 to -1.6, as the selector can choose from more certified candidates. Beyond $k = 7$, the privacy gain saturates, while runtime increases monotonically by a factor of 1.48 from $k = 7$ to $k = 20$. We thus set $k = 7$ by default, for its best observed privacy-efficiency tradeoff.

Varying seed ratio. Figure 5f examines the effect of the size of the label-covering seed $\mathcal{D}_0$. Larger seeds stabilize utility estimation and accelerate convergence, but incur greater initial exposure because the retention view only admits records. In contrast, seed ratios below 1% yield unstable estimates, weakening candidate ranking and requiring more admissions to meet the utility constraint. A ratio of 1.2% achieves the best observed balance among estimation stability, convergence, and privacy exposure.

Varying adversary background knowledge. As leak depends on the adversary prior π_i, we vary background knowledge from coarse

partial QIs to exact QI values. Figure 5g reports leak and empirical re-identification success at $\epsilon = 0.1$. Both increase as the prior becomes more concentrated, yet TRIM consistently beats the best baseline. Thus, its privacy advantage is robust across adversary priors rather than specific to one background-knowledge assumption.

Exp-6: Ablation studies. We evaluate the effectiveness of TRIM's key design decisions on PubCov.

Quality of retention classes. Figure 5h compares retention classes against alternative horizontal admission units: Sample admits individual records, KMeans forms clusters, IL groups via information loss, and RandomSplit splits groups randomly. Retention classes achieve the lowest risk, reaching leak = −7.44 with low selection time. RandomSplit terminates at leak = −1.10, whereas grouped alternatives leave singleton QI combinations (leak = 0), raising residual risk by 6.34–7.44. This confirms that retention classes provide an effective, auditable enumeration granularity.

LGA-based ranking. Under a ranking budget of $k = 7$, Figure 5i compares LGA with random ranking (Random) and influence-based scoring (IF), using full retraining (Retrain) as ground truth. We report runtime, true-top-1 HitRate@k, and Recall@k, measuring efficiency, success rate of top-1 recovery, and top-k recovery, respectively. LGA achieves the highest HitRate@k (0.95) and Recall@k (0.53), outperforming IF (0.37 and 0.68) and Random (0.05 and 0.00), while running >56.1% faster. Although Retrain is exact by construction, it is 40.6× slower than LGA. Thus, LGA preserves the most useful candidates near the top of the ranking at much lower cost.

End-to-end estimation fidelity. Figure 5j evaluates marginal-utility estimates for all candidates generated throughout TRIM, comparing its two-stage estimator with influence functions (IF) [56] and LGA against full retraining (Retrain) as ground truth. The two-stage estimator achieves the lowest mean absolute error while running 8.13× faster than Retrain, and all estimates satisfy the bound in Theorem 7. Although IF and LGA are faster, their errors are 27.6× and 22.6× higher, respectively. For the selected operation, *i.e.,* the candidate maximizing the ratio-marginal score $\rho(o)$, the mean error of $\rho(o)$ is 98.8% and 98.3% lower than those of IF and LGA. These show that the two-stage estimator provides a practical tradeoff between estimation accuracy and computational efficiency.

Structural ablations. Figure 5k compares TRIM with two single-axis variants on PubCov: TRIM_V uses only vertical attribute refinement, and TRIM_H only horizontal retention-class inclusion. TRIM attains lower leak and leak_T than both, showing that the two operations are complementary. Vertical refinement restores useful detail within retained classes, while horizontal inclusion selects which QI groups enter training. A single-axis approach either keeps risky groups (TRIM_V) or misses fine-grained utility in low-risk groups (TRIM_H), confirming the value of joint horizontal-vertical optimization.

Figure 5k isolates the effect of refine-and-swap operations by comparing full TRIM with bare refinement (RefineOnly) and refine-and-evict without compensating admission (RefineEvict). RefineOnly creates persistent privacy-risk spikes and thus achieves no risk reduction (leak = 0) under most utility tolerances. RefineEvict causes 1.79–3.69× utility loss under the same leak. By evicting risky fragments and admitting low-risk substitutes, refine-and-swap avoids privacy spikes, preserves nondecreasing utility, and consistently yields the best privacy-utility tradeoff.

Selection-rule ablation. Figure 5l replaces the ratio-marginal rule with a utility-only rule (Utility) that maximizes utility gain, a privacy-only rule (Privacy) that minimizes leakage increase, a random rule (Random), and an influence function-induced rule (IF). The ratio-marginal rule gives the best privacy-utility tradeoff because it normalizes each utility gain by its incremental leakage cost. At $\epsilon = 0.02$, it reduces leak by 1.03, 3.20 and 4.59 points relative to Utility, Privacy and IF, respectively. Utility spends privacy budget aggressively, Privacy favors low-risk but low-utility operations, and IF cannot fully capture the effect of composed retention decisions. These support ratio-marginal score as the most effective criterion.

Summary. The experiments show that TRIM effectively supports compliance-aware training-data minimization. *(1) Privacy-utility tradeoff.* Across both MLP and XGBoost, TRIM consistently yields the strongest Pareto frontier and supports flexible privacy-utility operating points. At the same leak level, it reduces model utility degradation by up to 84.9% relative to the strongest baselines; it also reduces tail risk leak_T. *(2) Empirical risk mitigation.* TRIM raises adversarial reconstruction and linkage errors by at least 20.7% and 41.8% over all baselines, respectively. *(3) Efficiency and scalability.* TRIM's end-to-end runtime is comparable to PAT, whereas FeatSel and EntryWise, which require retraining per candidate, take 1.66–28.82× as much time; it also scales sublinearly in data size. *(4) Auditability and robustness.* TRIM's operation log records support DPIA-style review of necessity and proportionality. Sensitivity analyses show that TRIM is robust to the certification budget k, seed ratio, and adversary prior; the defaults $k = 7$ and 1.2% lie near the knees of their respective tradeoffs. *(5) Ablation findings.* The ablations validate retention classes, LGA ranking, two-stage estimation, refine-and-swap, joint horizontal-vertical optimization, and ratio-marginal selection by demonstrating their advantages over their respective alternatives in privacy, utility, or efficiency.

8 CONCLUSION

The novelty of the work consists of (1) a compliance-aware formulation of training-data minimization that minimizes residual re-identification risk under an explicit utility constraint. (2) A retention view that builds a minimized dataset from a fully suppressed one, unifying horizontal retention-class inclusion and vertical attribute refinement. (3) A ratio-marginal optimization strategy that admits the operation with the largest utility gain per unit privacy exposure and yields an approximation guarantee. (4) A two-stage utility estimator that combines gradient-based LGA ranking with proxy certification, avoiding repeated target-model retraining. (5) An auditable minimization process that logs each retained operation with its utility gain, leakage change, and selection rationale. Our experimental study verifies that TRIM is promising in practice.

Future research directions for TRIM include (a) extending it from supervised ML training to broader data-driven workflows, such as analytics, data sharing, and model evaluation, where the unit of necessary data may vary across tasks; and (b) generalizing the framework beyond relational data to multimodal settings, including text, images, time series, and genomic data, where privacy risk arises from richer semantics and cross-modal linkage.

REFERENCES

[1] 2026. Full version with appendices. https://shuhaoliu.github.io/assets/papers/shuhao-vldb27-trim-full.pdf.

[2] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep Learning with Differential Privacy. In *ACM Conference on Computer and Communications Security (CCS)*. 308–318. https://doi.org/10.1145/2976749.2978318

[3] Constantin F. Aliferis, Alexander R. Statnikov, Ioannis Tsamardinos, Subramani Mani, and Xenofon D. Koutsoukos. 2010. Local Causal and Markov Blanket Induction for Causal Discovery and Feature Selection for Classification Part I: Algorithms and Empirical Evaluation. *J. Mach. Learn. Res.* 11 (2010), 171–234.

[4] Galen Andrew, Om Thakkar, Brendan McMahan, and Swaroop Ramaswamy. 2021. Differentially Private Learning with Adaptive Clipping. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 34. 17455–17466.

[5] Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. 2019. Fine-Grained Analysis of Optimization and Generalization for Overparameterized Two-Layer Neural Networks. In *International Conference on Machine Learning (ICML)*, Vol. 97. PMLR, 322–332. https://proceedings.mlr.press/v97/arora19a.html

[6] ARTICLE 29 DATA PROTECTION WORKING PARTY. 2014. Opinion 05/2014 on "Anonymisation Techniques". https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2014/wp216_en.pdf

[7] Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim. 2019. The Privacy Blanket of the Shuffle Model. In *International Cryptology Conference (Crypto)*. 638–667. https://doi.org/10.1007/978-3-030-26951-7_22

[8] Samyadeep Basu, Xuchen You, and Soheil Feizi. 2020. On second-order group influence functions for black-box predictions. In *International Conference on Machine Learning (ICML)*. PMLR, 715–724.

[9] Asia J. Biega, Peter Potash, Hal Daumé, Fernando Diaz, and Michèle Finck. 2020. Operationalizing the Legal Principle of Data Minimization for Personalization. In *ACM Conference on Research and Development in Information Retrieval (SIGIR)*. 399–408. https://doi.org/10.1145/3397271.3401034

[10] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, Brendan McMahan, et al. 2019. Towards Federated Learning at Scale: System Design. In *Proceedings of Machine Learning and Systems (MLSys)*, Vol. 1. 374–388.

[11] Olivier Bousquet and André Elisseeff. 2002. Stability and Generalization. *Journal of Machine Learning Research* 2, Mar (2002), 499–526. https://www.jmlr.org/papers/v2/bousquet02a.html

[12] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks. In *USENIX Security Symposium (Security)*. 267–284.

[13] Venkatesan T. Chakaravarthy, Himanshu Gupta, Prasan Roy, and Mukesh K. Mohania. 2008. Efficient Techniques for Document Sanitization. In *ACM Conference on Information and Knowledge Management (CIKM)*. 843–852. https://doi.org/10.1145/1458082.1458194

[14] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *ACM International Conference on Knowledge Discovery and Data Mining (KDD)*. 785–794. https://doi.org/10.1145/2939672.2939785

[15] Aloni Cohen and Kobbi Nissim. 2020. Towards Formalizing the GDPR's Notion of Singling out. *Proceedings of the National Academy of Sciences (PNAS)* 117, 15 (April 2020), 8344–8352. https://doi.org/10.1073/pnas.1914598117

[16] Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. 2021. Retiring Adult: New Datasets for Fair Machine Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 34. 6478–6490.

[17] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography (TCC)*. 265–284. https://doi.org/10.1007/11681878_14

[18] Cynthia Dwork and Aaron Roth. 2014. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends in Theoretical Computer Science* 9, 3–4 (Aug. 2014), 211–407. https://doi.org/10.1561/0400000042

[19] Pablo A. Estévez, Michel Tesmer, Claudio A. Perez, and Jacek M. Zurada. 2009. Normalized Mutual Information Feature Selection. *IEEE Trans. Neural Networks* 20, 2 (2009), 189–201. https://doi.org/10.1109/TNN.2008.2005601

[20] European Parliament and Council of the European Union. 2016. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the Protection of Natural Persons with Regard to the Processing of Personal Data and on the Free Movement of Such Data, and Repealing Directive 95/46/EC (General Data Protection Regulation). https://eur-lex.europa.eu/eli/reg/2016/679/oj

[21] Prakhar Ganesh, Cuong Tran, Reza Shokri, and Ferdinando Fioretto. 2025. The Data Minimization Principle in Machine Learning. In *ACM Conference on Fairness, Accountability, and Transparency (FAccT)*. 3075–3093. https://doi.org/10.1145/3715275.3732195

[22] Michael Garey and David Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company.

[23] Simson Garfinkel, John M Abowd, and Christian Martindale. 2019. Understanding Database Reconstruction Attacks on Public Data. *Communications of the ACM (CACM)* 62, 3 (2019), 46–53. https://doi.org/10.1145/3287287

[24] Amirata Ghorbani and James Zou. 2019. Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning (ICML)*. PMLR, 2242–2251.

[25] Matteo Giomi, Franziska Boenisch, Christoph Wehmeyer, and Borbála Tasnádi. 2023. A Unified Framework for Quantifying Privacy Risk in Synthetic Data. *Proceedings on Privacy Enhancing Technologies (PETS)* 2023, 2 (April 2023), 312–328. https://doi.org/10.56553/popets-2023-0055

[26] Abigail Goldsteen, Gilad Ezov, Ron Shmelkin, Micha Moffie, and Ariel Farkash. 2022. Data Minimization for GDPR Compliance in Machine Learning Models. *AI and Ethics* 2, 3 (2022), 477–491. https://doi.org/10.1007/s43681-021-00095-8

[27] Arthur Jacot, Franck Gabriel, and Clément Hongler. 2018. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 31. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2018/hash/5a4be1fa34e62bb8a6ec6b91d2462f5a-Abstract.html

[28] Siwon Kim, Sangdoo Yun, Hwaran Lee, Martin Gubri, Sungroh Yoon, and Seong Joon Oh. 2023. ProPILE: Probing Privacy Leakage in Large Language Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 36. 20750–20762. https://doi.org/10.52202/075280-0911

[29] Pang Wei Koh and Percy Liang. 2017. Understanding Black-box Predictions via Influence Functions. In *International Conference on Machine Learning (ICML)*, Vol. 70. PMLR, 1885–1894. https://proceedings.mlr.press/v70/koh17a.html

[30] Qinbin Li, Junyuan Hong, Chulin Xie, Jeffrey Tan, Rachel Xin, Junyi Hou, Xavier Yin, Zhun Wang, Dan Hendrycks, Zhangyang Wang, Bo Li, Bingsheng He, and Dawn Song. 2024. LLM-PBE: Assessing Data Privacy in Large Language Models. *PVLDB* 17, 11 (July 2024), 3201–3214. https://doi.org/10.14778/3681954.3681994

[31] Pierre Lison, Ildikó Pilán, David Sanchez, Montserrat Batet, and Lilja Øvrelid. 2021. Anonymisation Models for Text Data: State of the Art, Challenges and Future Directions. In *Annual Meeting of the Association for Computational Linguistics (ACL)*. 4188–4203. https://doi.org/10.18653/v1/2021.acl-long.323

[32] Shuhao Liu, Wenfei Fan, and Yijia Xu. 2026. Shielding PII to Prevent Re-identification and Preserve Utility. *Proc. ACM Manag. Data* 4, 3, Article 232 (May 2026). https://doi.org/10.1145/3802109

[33] Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems (NIPS)*, Vol. 30. 4768–4777.

[34] Microsoft. 2025. Presidio. https://microsoft.github.io/presidio/

[35] Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. 2020. Coresets for Data-Efficient Training of Machine Learning Models. In *International Conference on Machine Learning (ICML)*, Vol. 119. 6950–6960.

[36] Sérgio Moro, Paulo Cortez, and Paulo Rita. 2014. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems* 62 (2014), 22–31. https://doi.org/10.1016/j.dss.2014.03.001

[37] Arvind Narayanan and Vitaly Shmatikov. 2008. Robust De-Anonymization of Large Sparse Datasets. In *IEEE Symposium on Security and Privacy (SP)*. 111–125. https://doi.org/10.1109/SP.2008.33

[38] Xi Niu, Ruhani Rahman, Xiangcheng Wu, Zhe Fu, Depeng Xu, and Riyi Qiu. 2023. Leveraging Uncertainty Quantification for Reducing Data for Recommender Systems. In *IEEE International Conference on Big Data (BigData)*. 352–359. https://doi.org/10.1109/BigData59044.2023.10386790

[39] Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. 2021. Deep Learning on a Data Diet: Finding Important Examples Early in Training. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 34. 20596–20607. https://openreview.net/forum?id=Uj7pF-D-YvT

[40] OpenML Platform. 2026. BNG (credit-g) dataset. *https://www.openml.org/search?type=data&id=260&sort=runs&status=active.*

[41] Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. 2020. Estimating Training Data Influence by Tracing Gradient Descent. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 33. 19920–19930.

[42] Arij Riabi, Menel Mahamdi, Virginie Mouilleron, and Djamé Seddah. 2024. Cloaked Classifiers: Pseudonymization Strategies on Sensitive Classification Tasks. In *Workshop on Privacy in Natural Language Processing (PrivateNLP)*. 123–136. https://doi.org/10.18653/v1/2024.privatenlp-1.13

[43] Maytal Saar-Tsechansky, Prem Melville, and Foster Provost. 2009. Active Feature-Value Acquisition. *Management Science* 55, 4 (April 2009), 664–684. *https://pubsonline.informs.org/doi/10.1287/mnsc.1080.0952.*

[44] P. Samarati. 2001. Protecting Respondents Identities in Microdata Release. *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 13, 6 (2001), 1010–1027. https://doi.org/10.1109/69.971193

[45] Divya Shanmugam, Fernando Diaz, Samira Shabanian, Michele Finck, and Asia Biega. 2022. Learning to Limit Data Collection via Scaling Laws: A Computational Interpretation for the Legal Principle of Data Minimization. In *ACM Conference on Fairness, Accountability, and Transparency (FAccT)*. 839–849. https://doi.org/10.1145/3531146.3533148

[46] Ted Shaowang, Shinan Liu, Jonatas Marques, Nick Feamster, and Sanjay Krishnan. 2025. Algorithmic Data Minimization for Machine Learning over Internet-of-Things Data Streams. *PVLDB* 18, 13 (2025), 5652–5661. https://doi.org/10.14778/3773731.3773740

[47] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Mem-

bership Inference Attacks against Machine Learning Models. In *IEEE Symposium on Security and Privacy (SP)*. 3–18. https://doi.org/10.1109/SP.2017.41

[48] Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari S. Morcos. 2022. Beyond Neural Scaling Laws: Beating Power Law Scaling via Data Pruning. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 35. 19523–19536. https://doi.org/10.52202/068431-1419

[49] spaCy Developers. 2025. spaCy · Industrial-strength Natural Language Processing in Python. https://spacy.io/

[50] Robin Staab, Nikola Jovanović, Mislav Balunović, and Martin Vechev. 2024. From Principle to Practice: Vertical Data Minimization for Machine Learning. In *IEEE Symposium on Security and Privacy (SP)*. 4733–4752. https://doi.org/10.1109/SP54263.2024.00089

[51] Beata Strack, Jonathan P. DeShazo, Chris Gennings, Juan L. Olmo, Sebastian Ventura, Krzysztof J. Cios, and John N. Clore. 2014. Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records. *BioMed Research International* 2014, Article 781670 (2014), 11 pages. https://doi.org/10.1155/2014/781670

[52] Nishant Subramani, Sasha Luccioni, Jesse Dodge, and Margaret Mitchell. 2023. Detecting Personal Information in Training Corpora: An Analysis. In *Workshop on Trustworthy Natural Language Processing (TrustNLP)*. 208–220. https://doi.org/10.18653/v1/2023.trustnlp-1.18

[53] Maria Irena Szawerna, Simon Dobnik, Therese Lindström Tiedemann, Ricardo Muñoz Sánchez, Xuan-Son Vu, and Elena Volodina. 2024. Pseudonymization Categories across Domain Boundaries. In *Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING)*. 13303–13314. https://doi.org/10.63317/2mduzb3omzkm

[54] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A. Efros. 2018. Dataset Distillation. https://doi.org/10.48550/arXiv.1811.10959 arXiv:1811.10959 [cs.LG]

[55] Alexander Warnecke, Lukas Pirch, Christian Wressnegger, and Konrad Rieck. 2023. Machine Unlearning of Features and Labels. In *Network and Distributed System Security Symposium (NDSS)*. https://doi.org/10.14722/ndss.2023.23087

[56] Shuo Yang, Zeke Xie, Hanyu Peng, Min Xu, Mingming Sun, and Ping Li. 2023. Dataset Pruning: Reducing Training Data by Examining Generalization Influence. In *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=4wZiAXD29TQ

[57] Yu Yang, Hao Kang, and Baharan Mirzasoleiman. 2023. Towards Sustainable Learning: Coresets for Data-Efficient Deep Learning. In *International Conference on Machine Learning (ICML)*, Vol. 202. 39314–39330.

[58] Kui Yu, Lin Liu, Jiuyong Li, Wei Ding, and Thuc Duy Le. 2020. Multi-Source Causal Feature Selection. *IEEE Trans. Pattern Anal. Mach. Intell.* 42, 9 (2020), 2240–2256. https://doi.org/10.1109/TPAMI.2019.2908373

[59] Bo Zhao and Hakan Bilen. 2023. Dataset Condensation with Distribution Matching. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 6503–6512. https://doi.org/10.1109/WACV56688.2023.00645

[60] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. 2021. Dataset Condensation with Gradient Matching. In *International Conference on Learning Representations (ICLR)*. OpenReview.net. https://openreview.net/forum?id=mSAKhLYLSsl

A PROOF OF PROPOSITION 1

Proposition 1. In the exact-QI setting, the risk score $\mathsf{leak}(\mathcal{D}_M)$ is monotonically equivalent to k-anonymity-style singling-out risk.

Proof: The proof consists of the following four steps.

(1) Exact-QI adversary. Consider an adversary $\mathcal{A}$ that knows the exact quasi-identifier (QI) values $a_i \in \mathsf{dom}(A)$ of each target $i \in \mathcal{I}$. Its prior π_i over $\mathsf{dom}(A)$ is the point mass $\pi_i(a) = \mathbf{1}[a = a_i]$, and hence $H_i = 0$. Let $C_i(\mathcal{D}_M) := \{\, t \in \mathcal{D}_M : A(t) = a_i \,\}$ be the matching equivalence class and $k_i := |C_i(\mathcal{D}_M)| \geq 1$.

(2) Induced posterior and its entropy. Under the point-mass prior, every record with $A(t) \neq a_i$ has $\pi_i(A(t)) = 0$, leaving only the k_i records in $C_i(\mathcal{D}_M)$. As these records are indistinguishable under the released QIs, Bayes' rule assigns each of them posterior mass

$$q_i(t \mid \mathcal{D}_M) = \frac{\Pr(\mathcal{D}_M \mid t)\, \pi_i(a_i)}{\sum_{s \in C_i(\mathcal{D}_M)} \Pr(\mathcal{D}_M \mid s)\, \pi_i(a_i)} = \frac{1}{k_i}.$$

The posterior is therefore uniform over $C_i(\mathcal{D}_M)$, with entropy

$$H_i'(\mathcal{D}_M) = -\sum_{t \in C_i(\mathcal{D}_M)} \frac{1}{k_i} \log \frac{1}{k_i} = \log k_i.$$

(3) Equivalence for each target. The individual risk score is

$$\Delta H_i(\mathcal{D}_M) = -\log k_i = \log \frac{1}{k_i}.$$

The adversary's optimal singling-out probability is $p_i^* := \max_t q_i(t \mid \mathcal{D}_M) = 1/k_i$. Thus, $\Delta H_i(\mathcal{D}_M) = \log p_i^*$, and the strict monotonicity of the logarithm establishes equivalence between the score and singling-out probability for each target.

(4) Equivalence. Maximizing the individual scores over targets gives

$$\mathsf{leak}(\mathcal{D}_M) = \log \max_{i \in \mathcal{I}} p_i^* = -\log \min_{i \in \mathcal{I}} k_i.$$

Hence, $\mathsf{leak}(\mathcal{D}_M)$ is a monotone transformation of the worst-case singling-out probability. Moreover, for any integer $k \geq 1$, $\mathcal{D}_M$ satisfies the k-anonymity class-size condition over $\mathcal{I}$ [44] iff

$$\min_{i \in \mathcal{I}} k_i \geq k \quad \Leftrightarrow \quad \max_{i \in \mathcal{I}} p_i^* \leq \frac{1}{k} \quad \Leftrightarrow \quad \mathsf{leak}(\mathcal{D}_M) \leq -\log k.$$

This proves the monotone equivalence in the exact-QI setting. □

B PROOF OF THEOREM 2

Theorem 2. The decision version of DMP is NP-complete.

We consider the decision version of DMP: given a dataset $\mathcal{D}$, a model $\mathcal{M}$, thresholds B and ϵ, decide whether there exists a minimized dataset $\mathcal{D}_M$ such that $\mathsf{leak}(\mathcal{D}_M) \leq B$ and $\Delta U(\mathcal{D}_M) \leq \epsilon$.

Proof: We show that DMP is NP-complete by showing membership in NP and NP-hardness via a polynomial-time (PTIME) reduction from the 0–1 knapsack problem, which is NP-complete (cf. [22]).

Upper bound. We give an NP verifier for the decision version of DMP. It guesses a minimized dataset $\mathcal{D}_M = t_n \circ \cdots \circ t_1(\mathcal{D})$ represented by the sequence of minimization operations applied to $\mathcal{D}$. Since every operation only coarsens or removes information from a finite collection of cells, no operation need be applied twice and the number of distinct operations is polynomial in $|\mathcal{D}|$ and the schema size, so $\mathcal{D}_M$ admits a description of polynomial size.

Given $\mathcal{D}_M$, the verifier computes the residual privacy risk $\mathsf{leak}(\mathcal{D}_M)$ and the utility degradation $\Delta U(\mathcal{D}_M)$, and accepts iff $\mathsf{leak}(\mathcal{D}_M) \leq B$ and $\Delta U(\mathcal{D}_M) \leq \epsilon$. Both quantities are evaluable in PTIME, so the verification runs in PTIME and DMP is in NP.

Lower bound. We prove NP-hardness by reduction from the 0-1 knapsack problem, which is NP-complete (cf. [22]). A knapsack instance Ψ consists of items $1, \ldots, n$ with values $v_1, \ldots, v_n \geq 0$, weights $w_1, \ldots, w_n \geq 0$, a capacity W, and a target value V. It asks whether for some subset S of items, $\sum_{i \in S} w_i \leq W$ and $\sum_{i \in S} v_i \geq V$.

Given such an instance, we construct a DMP instance whose feasible minimized datasets are in one-to-one correspondence with the knapsack subsets, so that the privacy and utility constraints of DMP encode the capacity and the value requirement, respectively.

Construction. The construction uses a single target individual and a feature-additive utility. These instances form a sub-class of DMP, so hardness of this sub-class implies hardness of the general problem.

Let the adversary's target set be the singleton $\mathcal{I} = \{i^*\}$, so that $\mathsf{leak}(\mathcal{D}_M) = \Delta H_{i^*}(\mathcal{D}_M)$. The schema has n QI attributes $A_1, \ldots, A_n$, and the adversary's prior on i^* is the product distribution $\pi_{i^*} = \prod_{i=1}^n \pi^{(i)}$ whose i-th independent marginal carries w_i bits of uncertainty, $H(\pi^{(i)}) = w_i$ (*e.g.*, $\pi^{(i)}$ uniform over 2^{w_i} values). The prior entropy is thus $H_{i^*} = \sum_{i=1}^n w_i$, and the specification is polynomial in the input, as it lists only the n values w_i.

For each attribute A_i we provide two minimization operations: *retain* A_i, which publishes the target's i-th component and eliminates its w_i bits of uncertainty, and *suppress* A_i, which masks the component and leaves those bits intact. A minimized dataset $\mathcal{D}_M$ is therefore determined by the set $S \subseteq \{1, \ldots, n\}$ of retained attributes, whose posterior entropy is $H'_{i^*}(\mathcal{D}_M) = \sum_{i \notin S} w_i$. Consequently,

$$\mathsf{leak}(\mathcal{D}_M) = \Delta H_{i^*}(\mathcal{D}_M) = H_{i^*} - H'_{i^*}(\mathcal{D}_M) = \sum_{i \in S} w_i.$$

We make the downstream utility feature-additive. This is practical and realized by widely used model families whose accuracy decomposes across features such as linear and logistic regression, naive Bayes, and generalized additive models. In this case, attribute A_i contributes utility v_i when retained and nothing when suppressed, so the utility lost relative to full retention is

$$\Delta U(\mathcal{D}_M) = \sum_{i \notin S} v_i = \sum_{i=1}^n v_i - \sum_{i \in S} v_i.$$

Finally, we set the privacy threshold $B = W$ and the utility tolerance $\epsilon = \sum_{i=1}^n v_i - V$ (if $V > \sum_i v_i$ the knapsack instance is trivially infeasible, and we output a fixed infeasible DMP instance).

In summary, we have constructed a DMP instance whose

- feasible minimized datasets correspond exactly to knapsack subsets: each item i is realized as a QI attribute A_i with operations *retain* and *suppress*, and a dataset $\mathcal{D}_M$ encodes the subset $S = \{\, i : A_i \text{ retained} \,\}$; and
- privacy and utility constraints replicate the two knapsack constraints, mapping each weight w_i to the leakage of retaining A_i and each value v_i to its utility, so that $\mathsf{leak}(\mathcal{D}_M) = \sum_{i \in S} w_i \leq B = W$ encodes the capacity and $\Delta U(\mathcal{D}_M) = \sum_{i \notin S} v_i \leq \epsilon = \sum_{i=1}^n v_i - V$ encodes the value requirement $\sum_{i \in S} v_i \geq V$.

The construction emits n attributes, two operations each, and two thresholds, hence is computable in PTIME.

Correctness. We next verify that the reduction is correct.

($\Rightarrow$) Suppose the knapsack instance Ψ admits a subset S with $\sum_{i \in S} w_i \leq W$ and $\sum_{i \in S} v_i \geq V$. Let $\mathcal{D}_M$ retain exactly the at-

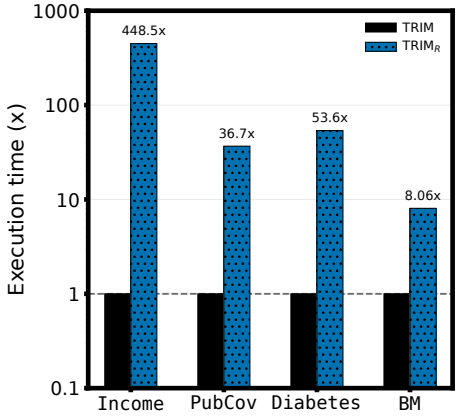


Figure 6: Execution times of TRIM$_R$ relative to that of TRIM.

tributes in S. Then $\text{leak}(\mathcal{D}_M) = \sum_{i \in S} w_i \leq W = B$, and $\Delta U(\mathcal{D}_M) = \sum_{i=1}^{n} v_i - \sum_{i \in S} v_i \leq \sum_{i=1}^{n} v_i - V = \epsilon$. Hence, $\mathcal{D}_M$ is a feasible solution of the DMP instance.

($\Leftarrow$) Conversely, let $\mathcal{D}_M$ be a feasible solution and set $S = \{ i : A_i \text{ is retained in } \mathcal{D}_M \}$. Feasibility gives $\sum_{i \in S} w_i = \text{leak}(\mathcal{D}_M) \leq B = W$ and $\sum_{i=1}^{n} v_i - \sum_{i \in S} v_i = \Delta U(\mathcal{D}_M) \leq \epsilon = \sum_{i=1}^{n} v_i - V$, *i.e.,* $\sum_{i \in S} v_i \geq V$. Thus S is a feasible knapsack subset for Ψ. □

C RETENTION VIEW VS. REMOVAL VIEW

We empirically evaluate the efficiency advantage of the retention view over the removal view. Although the two views admit the same feasible minimized datasets, their opposite search directions can incur substantially different computational costs.

As discussed in Section 4, the retention view begins with the minimally exposed dataset $\mathcal{D}_0$, so its early iterations evaluate candidates on small retained datasets. The following experiment isolates the resulting efficiency gain.

Experiment setup. We compare TRIM with its removal-view counterpart, denoted by TRIM$_R$. Starting from the full training set at the finest generalization levels, TRIM$_R$ reduces the retained information by either coarsening one attribute or removing one retention class at each iteration. It greedily selects the operation with the greatest reduction in leak per unit of utility loss, subject to the prescribed utility tolerance. Thus, TRIM$_R$ searches the same minimization space as TRIM but proceeds in the opposite direction.

To isolate the effect of search direction, we keep all other experimental configurations identical. Both methods use an MLP as the downstream target model and share the same stratified 70/15/15 data splits, random seeds, utility estimator, and candidate-evaluation procedure. We set $\epsilon = 0.05$ for `Income` and `PubCov`, $\epsilon = 0.02$ for `Diabetes`, and $\epsilon = 0.005$ for `BM`.

Results. Figure 6 reports the execution times normalized to that of TRIM. On `Income`, `PubCov`, `Diabetes`, and `BM`, TRIM$_R$ requires 448.5×, 36.7×, 53.6×, and 8.06× the execution time required by TRIM, respectively. Thus, the retention view reduces execution time by more than one order of magnitude on three of the four datasets and by a factor of 8.06 on `BM`.

This difference arises for two reasons:

(1) TRIM begins with suppressed $\mathcal{D}_0$, so its early iterations enumerate and evaluate candidates on small retained datasets. By contrast, TRIM$_R$ begins with the full training set, where the number of retention classes is close to the number of records, and must therefore evaluate a large candidate set from the outset.

(2) Under a tight utility tolerance, few attribute-coarsening operations remain feasible. TRIM$_R$ must instead remove small retention classes individually, and each removal only marginally shrinks the candidate set. It consequently repeats similarly expensive evaluations over many iterations. TRIM avoids this cost by admitting only the information needed to recover the target utility, which empirically realizes the efficiency advantage identified in Section 4.

D PROOF OF PROPOSITION 3

Proposition 3. The retention view of DMP is equivalent to the removal view: both admit the same feasible minimized datasets and attain the same optimal objective value.

Proof: Picture the reachable datasets as a directed acyclic graph (DAG) G. Its nodes are the datasets obtainable from $\mathcal{D}$ by minimization operations (record removal, attribute suppression, or generalization to a coarser QI level), with an edge $\mathcal{D}'' \to \mathcal{D}'$ whenever one such operation turns $\mathcal{D}''$ into $\mathcal{D}'$ by deleting a single atomic unit (a record or a one-level QI generalization).

Let $R(\mathcal{D}_M)$ denote the retained information units in $\mathcal{D}_M$. Each edge removes at least one unit from $R(\cdot)$, so G is acyclic. Its unique source is the original dataset $\mathcal{D}$, where $R(\mathcal{D})$ is complete, and its unique sink is the fully suppressed dataset $\mathcal{D}_0$, where $R(\mathcal{D}_0) = \emptyset$; in $\mathcal{D}_0$, all QIs are masked and only the label-covering seed remains. Thus, every feasible minimized dataset is a node on some path from $\mathcal{D}$ to $\mathcal{D}_0$, *i.e.,*, an element of the interval $\mathcal{L} = [\mathcal{D}_0, \mathcal{D}]$.

The two views are the two directions of travel along G. The removal view follows edges forward from the source $\mathcal{D}$, while the retention view follows them backward from the sink $\mathcal{D}_0$: each reversed edge being a retention operation (attribute refinement or retention-class inclusion) that re-admits the unit its forward edge deleted. Since forward and reverse steps are mutual inverses, every node $\mathcal{D}_M \in \mathcal{L}$ is reached both ways: deleting the units in $R(\mathcal{D}) \setminus R(\mathcal{D}_M)$ walks down from $\mathcal{D}$ to $\mathcal{D}_M$, and admitting the units in $R(\mathcal{D}_M)$ walks up from $\mathcal{D}_0$ to $\mathcal{D}_M$. Both views thus visit exactly the nodes of $\mathcal{L}$ and generate the same minimized datasets.

Finally, both $\Delta U(\mathcal{D}_M)$ and $\text{leak}(\mathcal{D}_M)$ depend only on the resulting dataset, not on the sequence of operations that produces it. Thus the feasible region $\{\mathcal{D}_M \in \mathcal{L} \mid \Delta U(\mathcal{D}_M) \leq \epsilon\}$ and the objective $\text{leak}(\cdot)$ are identical under the removal and retention views. That is, the two views have the same feasible datasets, the same minimizers, and the same optimal value; equivalently, for any thresholds B and ϵ, the decision version has a witness under one view iff it has one under the other. The retention view thus changes only the search direction, not the underlying optimization problem. □

E PROOF OF PROPOSITION 4

Proposition 4. Let $\mathcal{V}_t = \mathcal{V}_t^{\text{attr}} \cup \mathcal{V}_t^{\text{cls}}$ be the operations enumerated by TRIM at round t. Then every candidate $o \in \mathcal{V}_t$ satisfies $\Delta U(o) \geq 0$ and $\Delta\text{leak}(o) \geq 0$. Moreover, over sets of operations, the marginal utility recovery $\Delta U(\cdot)$ is a monotone submodular function and the

marginal privacy cost $\Delta\mathsf{leak}(\cdot)$ is monotone supermodular.

Proof: We work at round t, where the current dataset is $\mathcal{D}_{t-1}$ and TRIM enumerates $\mathcal{V}_t = \mathcal{V}_t^{\mathrm{attr}} \cup \mathcal{V}_t^{\mathrm{cls}}$.

We first recall the relevant notions, then verify them.

Preliminaries. For a dataset functional f, an operation o, and a retained dataset $\mathcal{D}'$, write the marginal gain

$$\Delta f(o \mid \mathcal{D}') := f(\mathcal{D}' \oplus o) - f(\mathcal{D}').$$

Here $\Delta U(o)$ and $\Delta\mathsf{leak}(o)$ are the marginals at the current dataset, *i.e.,* $\Delta U(o) = \Delta U(o \mid \mathcal{D}_{t-1})$ and $\Delta\mathsf{leak}(o) = \Delta\mathsf{leak}(o \mid \mathcal{D}_{t-1})$. Write $\mathcal{D}' \sqsubseteq \mathcal{D}''$ when $\mathcal{D}'$ retains a subset of the information in $\mathcal{D}''$, the order along which the retention view admits data ($\mathcal{D}_0 \sqsubseteq \mathcal{D}_1 \sqsubseteq \cdots$).

We say that f is *monotone* (nondecreasing) if $f(\mathcal{D}') \le f(\mathcal{D}'')$ whenever $\mathcal{D}' \sqsubseteq \mathcal{D}''$; *submodular* if $\Delta f(o \mid \mathcal{D}') \ge \Delta f(o \mid \mathcal{D}'')$ for all $\mathcal{D}' \sqsubseteq \mathcal{D}''$ and o (diminishing marginal returns); and *supermodular* if the reverse inequality holds (increasing differences).

We next show the statements one by one.

(1) Nonnegative marginals. Consider a retention-class inclusion $o \in \mathcal{V}_t^{\mathrm{cls}}$, which admits a not-yet-retained retention class C *i.e.,* the maximal set of records sharing a QI signature. Admitting C enlarges $\mathcal{D}_{t-1}$ with correctly supported, task-relevant coverage, so utility does not decrease and $\Delta U(o) \ge 0$.

Moreover, because retention classes partition records by *exact* signature equality, admitting the fresh signature of C leaves every previously retained class unchanged and merely adds C. Each existing target i thus keeps its posterior term $\log k_i$, while C contributes the terms $\{H_i - \log|C|\}_{i\in C}$ to the maximum defining leak (Equation 2), so $\mathsf{leak}(\mathcal{D}_{t-1} \oplus o) = \max\{\mathsf{leak}(\mathcal{D}_{t-1}),\ \max_{i\in C}(H_i - \log|C|)\} \ge \mathsf{leak}(\mathcal{D}_{t-1})$, *i.e.,* $\Delta\mathsf{leak}(o) \ge 0$.

For a refine-and-swap $o \in \mathcal{V}_t^{\mathrm{attr}}$, both bounds are enforced by Algorithm 1: its admission loop (line 4) iterates while $U(\mathcal{D}') < U(\mathcal{D}_{t-1})$ and halts only once $U(\mathcal{D}') \ge U(\mathcal{D}_{t-1})$, and o is emitted (line 9) only then. From these it follows that $\Delta U(o) \ge 0$. Any candidate with $\Delta\mathsf{leak}(o) < 0$ is intercepted by the triage rule (line 8) and committed as a Pareto improvement outside $\mathcal{V}_t$, so every operation surviving into $\mathcal{V}_t^{\mathrm{attr}}$ has $\Delta\mathsf{leak}(o) \ge 0$. Thus every $o \in \mathcal{V}_t$ satisfies $\Delta U(o) \ge 0$ and $\Delta\mathsf{leak}(o) \ge 0$. Since these guarantees hold at every retained dataset, U and leak are monotone.

(2) Submodularity of utility. Let $\mathcal{E}$ be a fixed set of mutually compatible retention operations such that, for every $S \subseteq \mathcal{E}$, applying the operations in S in any order yields the same retained view $\mathcal{D}(S)$. We assume that there exist fixed coverage sets $R(o)$ such that, for every $S \subseteq \mathcal{E}$, $U(\mathcal{D}(S)) - U(\mathcal{D}(\varnothing)) = \left|\bigcup_{a\in S} R(a)\right|$, $\mathrm{cov}(\mathcal{D}(S)) = \bigcup_{a\in S} R(a)$. Inclusion is coverage-driven: filtering selects under-supported but correctly classified records, and each admitted class restores the task-relevant region it represents. Associating with each operation o the set $R(o)$ of validation points it covers, the utility gain is the points it newly covers, $\Delta U(o \mid \mathcal{D}') = |R(o) \setminus \mathrm{cov}(\mathcal{D}')|$, where $\mathrm{cov}(\mathcal{D}')$ is the region already covered by $\mathcal{D}'$. Since $\mathrm{cov}(\mathcal{D}') \subseteq \mathrm{cov}(\mathcal{D}'')$ whenever $\mathcal{D}' \sqsubseteq \mathcal{D}''$,
$\Delta U(o \mid \mathcal{D}') = |R(o)\setminus\mathrm{cov}(\mathcal{D}')| \ge |R(o)\setminus\mathrm{cov}(\mathcal{D}'')| = \Delta U(o \mid \mathcal{D}'')$, since removing a larger region from $R(o)$ leaves no more points. This is the diminishing-returns inequality, so U is submodular.

(3) Supermodularity of privacy loss. In the exact-QI setting, Proposition 1 gives $\mathsf{leak}(\mathcal{D}') = -\log\min_{i\in I} k_i(\mathcal{D}')$. Consider retained datasets $\mathcal{D}' \sqsubseteq \mathcal{D}''$ and an operation o applicable to both. We assume that $\Delta\mathsf{leak}(o \mid \mathcal{D}') > 0$ and QI attributes are complementary for re-identification. By Proposition 1, the marginal privacy loss is $\Delta\mathsf{leak}(o \mid \mathcal{D}') = \log \frac{\min_{i\in I} k_i(\mathcal{D}')}{\min_{i\in I} k_i(\mathcal{D}'\oplus o)}$. Hence, by the complementarity condition, we can get $1 \le \frac{\min_{i\in I} k_i(\mathcal{D}')}{\min_{i\in I} k_i(\mathcal{D}'\oplus o)} \le \frac{\min_{i\in I} k_i(\mathcal{D}'')}{\min_{i\in I} k_i(\mathcal{D}''\oplus o)}$, so that $\Delta\mathsf{leak}(o \mid \mathcal{D}') \le \Delta\mathsf{leak}(o \mid \mathcal{D}'')$. Thus, these operations satisfy the increasing-differences condition. Since the retained datasets and applicable operations considered here are finite, strict positivity implies that the minimum curvature ratio is positive. Hence, the curvature over these operations satisfies $\gamma \in [0, 1)$.

The proof of Theorem 5 uses this conclusion for the still-unselected positive-marginal operations representing $\mathcal{D}_M$. Pareto-improving refine-and-swap operations are handled separately and do not enter this curvature bound. □

F PROOF OF THEOREM 5

Theorem 5. Let $\mathcal{D}_M$ be an optimal solution to DMP and let $\mathcal{D}_T$ be the dataset returned by TRIM. Then $\mathcal{D}_T$ achieves a constant-factor approximation: $\Delta\mathsf{leak}(\mathcal{D}_T) \le \frac{\log(1/\delta_U)+1/\delta_U}{1-\gamma}\Delta\mathsf{leak}(\mathcal{D}_M)$, where $\gamma \in [0, 1)$ is the curvature of the privacy-risk function and $\delta_U \in (0, 1]$ is the resolution of the normalized utility metric.

Proof: The proof proceeds in five steps. We first formulate the retention view and state the assumptions quantitatively. We then compare the ratio selected by TRIM with the remaining operations of an optimal solution. We next charge all preterminal privacy costs through a logarithmic potential and bound the final operation using the utility resolution. We finally verify feasibility and show that off-budget Pareto improvements preserve the bound.

(1) Setup and assumptions. If $\mathcal{D}_0$ satisfies the utility constraint, the claim is immediate. Consider a nontrivial execution with final operation o_m^*. For any retained dataset $\mathcal{D}'$, define its privacy cost as

$$\Delta\mathsf{leak}(\mathcal{D}') := \mathsf{leak}(\mathcal{D}') - \mathsf{leak}(\mathcal{D}_0).$$

For an operation o, write

$$\Delta\mathsf{leak}(o \mid \mathcal{D}') := \mathsf{leak}(\mathcal{D}' \oplus o) - \mathsf{leak}(\mathcal{D}')$$

for its marginal privacy cost at $\mathcal{D}'$. Before the final round, let $r_t := U(\mathcal{D}) - \epsilon - U(\mathcal{D}_t) > 0$ be the remaining utility gap.

We use the following assumptions.

(A1) *Structural properties*: By Proposition 4, every charged operation has nonnegative utility and privacy marginals, U is monotone submodular, and leak is monotone supermodular. Every off-budget Pareto improvement has nonnegative utility marginal and nonpositive privacy marginal.

(A2) *Bounded privacy curvature*: The supermodular curvature

$$\gamma := 1 - \min_{o,\ \mathcal{D}'\sqsubseteq\mathcal{D}''} \frac{\Delta\mathsf{leak}(o \mid \mathcal{D}')}{\Delta\mathsf{leak}(o \mid \mathcal{D}'')}$$

is bounded away from 1 by an input-independent constant, where the minimum ranges over positive denominators.

(A3) *Fixed utility resolution*: Utility is normalized to $[0, 1]$, and all utility values and ϵ are integer multiples of δ_U. The resolution δ_U is bounded below by an input-independent constant.

(A4) *Candidate completeness*: At every nonterminal round, the remaining operations of $\mathcal{D}_M$ are available for the greedy comparison.

Assumptions (A2)–(A4) match the discretized design of TRIM. Bounded-depth generalization hierarchies and the risk ceiling restrict privacy-marginal inflation, so (A2) can be checked over the reachable retention states. The affine normalization in (A3) scales every utility marginal equally and therefore preserves the ratio-marginal ordering. Moreover, utility differences below validation-sampling and proxy-model uncertainty are not meaningful for termination, which supports rounding utility and ϵ to a fixed operational resolution. Consequently, $r_0 \le 1$, every positive residual satisfies $r_t \ge \delta_U$, and every operation satisfies $\Delta U(o) \le 1$. Finally, (A4) holds under complete enumeration of the applicable atomic operations. Candidate scores may order their evaluation; if pruning is used, its certified upper bound must preserve the comparison with every pruned optimal operation.

(2) Comparator ratio. Let S_M be a set of retention operations that induces $\mathcal{D}_M$, and let S_{t-1} contain the charged operations selected before round t. Define the remaining comparator set as $\mathcal{V}_t^M := S_M \setminus S_{t-1}$. By (A4), $\mathcal{V}_t^M \subseteq \mathcal{V}_t$. Since $\mathcal{D}_M$ satisfies the utility constraint, applying these operations at $\mathcal{D}_{t-1}$ can recover the residual r_{t-1}. Submodularity of U therefore gives

$$\sum_{o \in \mathcal{V}_t^M} \Delta U(o \mid \mathcal{D}_{t-1}) \ge r_{t-1}.$$

By (A2), a privacy marginal at $\mathcal{D}_{t-1}$ is at most its baseline marginal divided by $1-\gamma$. At the baseline, supermodularity makes the sum of the comparator's singleton costs no greater than its total normalized cost, *i.e.,* $\sum_{o \in \mathcal{V}_t^M} \Delta\mathsf{leak}(o \mid \mathcal{D}_{t-1}) \le \frac{\Delta\mathsf{leak}(\mathcal{D}_M)}{1-\gamma}$. These imply that $o \in \mathcal{V}_t^M$ has marginal ratio at least $\frac{(1-\gamma) r_{t-1}}{\Delta\mathsf{leak}(\mathcal{D}_M)}$. Since TRIM selects the largest marginal ratio, its choice o_t^* satisfies

$$\Delta\mathsf{leak}(o_t^* \mid \mathcal{D}_{t-1}) \le \frac{\Delta\mathsf{leak}(\mathcal{D}_M)}{1-\gamma} \frac{\Delta U(o_t^* \mid \mathcal{D}_{t-1})}{r_{t-1}}.$$

This inequality charges the privacy cost of a greedy operation to the fraction of the current utility gap that it recovers.

(3) Preterminal privacy cost. For every round $t < m$, the utility target remains unmet after selecting o_t^*. By (A1), the selected operation and any Pareto improvement have nonnegative utility gain, so $r_t \le r_{t-1} - \Delta U(o_t^* \mid \mathcal{D}_{t-1})$. It follows that

$$\frac{\Delta U(o_t^* \mid \mathcal{D}_{t-1})}{r_{t-1}} \le 1 - \frac{r_t}{r_{t-1}} \le \log\left(\frac{r_{t-1}}{r_t}\right),$$

where the second inequality uses $1 - x \le -\log x$.

Substituting this potential drop into the comparator charge and summing over $t = 1, \dots, m-1$ yields

$$\begin{aligned}\sum_{t=1}^{m-1} \Delta\mathsf{leak}(o_t^* \mid \mathcal{D}_{t-1}) &\le \frac{\Delta\mathsf{leak}(\mathcal{D}_M)}{1-\gamma} \log\left(\frac{r_0}{r_{m-1}}\right) \\ &\le \frac{\log(1/\delta_U)}{1-\gamma} \Delta\mathsf{leak}(\mathcal{D}_M),\end{aligned}$$

where the last inequality follows from (A3): $r_0 \le 1$ and $r_{m-1} \ge \delta_U$.

(4) Final approximation factor. The final operation may overshoot the remaining utility gap. The comparator charge from Step (2), however, still applies at round m. By (A3), we have $\Delta U(o_m^* \mid \mathcal{D}_{m-1}) \le 1$ and $r_{m-1} \ge \delta_U$, and $\Delta\mathsf{leak}(o_m^* \mid \mathcal{D}_{m-1}) \le \frac{\Delta\mathsf{leak}(\mathcal{D}_M)}{(1-\gamma)\delta_U}$.

Combining this bound with Step (3), the total privacy cost of the charged operations is at most

$$\frac{\log(1/\delta_U) + 1/\delta_U}{1-\gamma} \Delta\mathsf{leak}(\mathcal{D}_M).$$

Since $1-\gamma$ and δ_U are bounded below by input-independent constants, this is a constant-factor bound.

(5) Feasibility and Pareto improvements. TRIM terminates only when $\Delta U(\mathcal{D}_T) \le \epsilon$, so $\mathcal{D}_T$ satisfies the utility constraint of DMP. Refine-and-swap commits any leakage-reducing operation outside the charged candidate set as a Pareto improvement. Such an operation decreases the remaining utility gap and has nonpositive privacy cost. It therefore cannot increase either the logarithmic charge or the final privacy cost. Because the comparator argument in Step (2) restarts at every reachable state, inserting these commits also preserves all subsequent charges. Consequently,

$$\Delta\mathsf{leak}(\mathcal{D}_T) \le \frac{\log(1/\delta_U) + 1/\delta_U}{1-\gamma} \Delta\mathsf{leak}(\mathcal{D}_M),$$

which proves the theorem. □

G PROOF OF THEOREM 6

Theorem 6. Assume that the validation loss is smooth, the training gradients are bounded, and the learning rate η is sufficiently small. Then for any two operations o_1 and o_2, $\mathsf{LGA}_\theta(\mathcal{D}, o_1)$ and $\mathsf{LGA}_\theta(\mathcal{D}, o_2)$ induce the same ordering as their true utility gains whenever the gap between those gains exceeds $O(\eta^2)$.

Proof: The proof proceeds in three steps. We first state the two hypotheses quantitatively. We then show, by a second-order expansion, that the utility gain $\Delta U(o)$ equals $\eta\,\mathsf{LGA}_\theta(\mathcal{D}, o)$ up to an $O(\eta^2)$ remainder. We finally use this surrogate to rank any two well-separated operations correctly.

(1) Setup and assumptions. Fix the parameters θ of $\mathcal{M}$ and the current dataset $\mathcal{D}$. Write $v = \nabla_\theta L_{\mathsf{v}}(\theta)$ for the validation gradient and $g = \nabla_\theta L_{\mathsf{train}}(\theta, \mathcal{D})$, $g_o = \nabla_\theta L_{\mathsf{train}}(\theta, \mathcal{D} \oplus o)$ for the training gradients before and after applying a candidate o, respectively, so that the LGA score $\mathsf{LGA}_\theta(\mathcal{D}, o) = \langle v,\, g_o - g\rangle$ aligns the validation gradient with the marginal training gradient $g_o - g$ that o induces.

We use the two assumptions in the following form.

(A1) *Smoothness*: L_{v} is L-smooth, so $|L_{\mathsf{v}}(\theta+\delta) - L_{\mathsf{v}}(\theta) - \langle v, \delta\rangle| \le \frac{L}{2}\|\delta\|^2$ for every δ.

(A2) *Bounded gradients*: $\|g\|, \|g_o\| \le G$ for every candidate o.

(2) LGA is a first-order surrogate of utility. The utility gain of o is the reduction in validation loss from the step that includes o, relative to the baseline step on $\mathcal{D}$:

$$\Delta U(o) = L_{\mathsf{v}}(\theta - \eta g) - L_{\mathsf{v}}(\theta - \eta g_o).$$

The two updates displace θ by $-\eta g$ and $-\eta g_o$. Applying the smoothness bound (A1) to each displacement expands the two losses as

$$\begin{aligned}L_{\mathsf{v}}(\theta - \eta g) &= L_{\mathsf{v}}(\theta) - \eta\,\langle v, g\rangle + r, \\ L_{\mathsf{v}}(\theta - \eta g_o) &= L_{\mathsf{v}}(\theta) - \eta\,\langle v, g_o\rangle + r_o,\end{aligned}$$

with second-order remainders $|r| \le \frac{L}{2}\eta^2\|g\|^2$ and $|r_o| \le \frac{L}{2}\eta^2\|g_o\|^2$. Subtracting the second line from the first, the base losses $L_{\mathsf{v}}(\theta)$ cancel, the linear terms combine into $\eta\langle v, g_o - g\rangle = \eta\,\mathsf{LGA}_\theta(\mathcal{D}, o)$, and the remainders form $\rho(o) := r - r_o$:

$$\Delta U(o) = \eta\,\mathsf{LGA}_\theta(\mathcal{D}, o) + \rho(o).$$

By (A2), $|\rho(o)|$ is bounded uniformly over candidates: $|\rho(o)| \le$

$|r| + |r_o| \le \frac{L}{2}\eta^2(\|g\|^2 + \|g_o\|^2) \le LG^2\eta^2$. So the true gain equals the LGA score scaled by the shared factor η, up to an $O(\eta^2)$ error.

(3) Ranking preservation. Fix two operations o_1 and o_2. By Step (2), $\overline{\Delta U}(o_i) = \eta\,\mathrm{LGA}_\theta(\mathcal{D}, o_i) + \rho(o_i)$. Subtracting gives the gain gap as the scaled LGA-score gap plus $\rho(o_1) - \rho(o_2)$, whose magnitude is at most $|\rho(o_1)| + |\rho(o_2)| \le 2LG^2\eta^2$. Thus, if $|\Delta U(o_1) - \Delta U(o_2)| > 2LG^2\eta^2$, this second-order perturbation cannot reverse the scaled score gap. Since $\eta > 0$, $\Delta U(o_1) - \Delta U(o_2)$ and $\mathrm{LGA}_\theta(\mathcal{D}, o_1) - \mathrm{LGA}_\theta(\mathcal{D}, o_2)$ have the same sign. Hence LGA preserves the true ordering whenever the gains differ by more than $2LG^2\eta^2 = O(\eta^2)$, proving the theorem.

Role of a small learning rate. The assumption (A3) of small η controls the remainder. Since the signal $\eta\,\mathrm{LGA}_\theta(\mathcal{D}, o)$ is of order η but the error $\rho(o)$ is of order η^2, a smaller η makes the surrogate more faithful and shrinks the misorder band $2LG^2\eta^2$; therefore, LGA ranks correctly every pair whose scores differ by more than $2LG^2\eta$. Because η is shared, it never alters the LGA order itself; its smallness only certifies that this order agrees with true utility. □

H PROOF OF THEOREM 7

Theorem 7. Assume the loss is bounded. Then, with high probability, the true utility recovery of any candidate operation o satisfies

$$\Delta U(o) \;\ge\; \overline{\Delta U}(o) \;-\; O\!\left(\frac{|o|}{n}\frac{d}{w} + \frac{1}{\sqrt{n}}\right),$$

where $n = |\mathcal{D}_{t-1} \oplus o|$, $|o|$ is the number of single-record changes in o, d is the input dimension, and w is the proxy width.

Proof: The proof proceeds in four steps. We first decompose the proxy–target recovery error into validation noise and the operation-induced change in proxy bias. We then bound the validation noise by concentration and control the bias change by tracing o through a neighboring-record path. Finally, we combine the two bounds to obtain the claimed correction.

(1) Setup and error decomposition. Fix a candidate operation o before drawing its certification samples, write $\mathcal{D}_t = \mathcal{D}_{t-1} \oplus o$, and let $n = |\mathcal{D}_t|$. Assume that $0 \le \ell \le B$ and that the proxy evaluations use i.i.d. validation samples of size $m = \Theta(n)$ drawn independently.

For a dataset $\mathcal{D}'$, let $\theta_{\mathcal{D}'}$ and $\bar\theta_{\mathcal{D}'}$ be the parameters obtained by training the target and proxy, respectively. Let $R(\theta) := \mathbb{E}_{(x,y)}[\ell(x, y; \theta)]$ denote population risk. For $j \in \{t-1, t\}$, let V_j be the validation sample for the proxy at state $\mathcal{D}_j$, and let $\widehat{R}_j(\bar\theta) := m^{-1}\sum_{(x,y)\in V_j}\ell(x, y; \bar\theta)$ denote its empirical validation risk. The samples V_{t-1} and V_t may coincide, but each is independent of training. Since utility is negative risk, $\Delta U(o) = R(\theta_{\mathcal{D}_{t-1}}) - R(\theta_{\mathcal{D}_t})$, while $\overline{\Delta U}(o) = \widehat{R}_{t-1}(\bar\theta_{\mathcal{D}_{t-1}}) - \widehat{R}_t(\bar\theta_{\mathcal{D}_t})$. Define $\Phi(\mathcal{D}') := R(\theta_{\mathcal{D}'}) - R(\bar\theta_{\mathcal{D}'})$ as the target-proxy population-risk gap after training on $\mathcal{D}'$. Adding and subtracting the two proxy population risks yields

$$\begin{aligned}\overline{\Delta U}(o) - \Delta U(o) = {}& \left[\widehat{R}_{t-1}(\bar\theta_{\mathcal{D}_{t-1}}) - R(\bar\theta_{\mathcal{D}_{t-1}})\right] \\ &+ \left[R(\bar\theta_{\mathcal{D}_t}) - \widehat{R}_t(\bar\theta_{\mathcal{D}_t})\right] \\ &+ \left[\Phi(\mathcal{D}_t) - \Phi(\mathcal{D}_{t-1})\right]. \end{aligned} \tag{8}$$

The first two terms are the validation errors before and after o, and the last captures the proxy-bias change induced by o. Hence, the target and proxy may differ arbitrarily in parameters or absolute risk: any persistent gap cancels when comparing their recoveries.

(2) Validation concentration. The proxy recovery uses empirical validation risks, so the first two terms of Equation (8) are ordinary sampling noise. Conditioned on the trained proxy parameters, the losses in each validation sample are independent and lie in $[0, B]$. Hoeffding's inequality and a union bound over the two evaluations imply that, with probability at least $1 - \delta$,

$$\left|\widehat{R}_j(\bar\theta_{\mathcal{D}_j}) - R(\bar\theta_{\mathcal{D}_j})\right| \le B\sqrt{\frac{\log(4/\delta)}{2m}}, \qquad j \in \{t-1, t\}. \tag{9}$$

The two evaluations therefore contribute at most twice the right-hand side of Equation (9), depending only on the validation sample size, not on the target-proxy parameter gap.

(3) Local response along an operation path. It remains to bound the target–proxy risk-gap change induced by o. Write o as $q = |o|$ neighboring single-record changes, $\mathcal{D}^{(0)} = \mathcal{D}_{t-1}, \ldots, \mathcal{D}^{(q)} = \mathcal{D}_t$, with intermediate dataset sizes $\Theta(n)$. This path is only an accounting device: telescoping remains exact despite interactions among record changes. Neighboring-dataset paths are standard in stability analysis [11] and influence-function analysis [29].

For neighboring datasets $\mathcal{D}' \sim \mathcal{D}''$, define $\Delta_T(\mathcal{D}'', \mathcal{D}') := R(\theta_{\mathcal{D}''}) - R(\theta_{\mathcal{D}'})$, and $\Delta_P(\mathcal{D}'', \mathcal{D}') := R(\bar\theta_{\mathcal{D}''}) - R(\bar\theta_{\mathcal{D}'})$. Assume the target and proxy respond similarly to each single-record change:

$$|\Delta_T(\mathcal{D}'', \mathcal{D}') - \Delta_P(\mathcal{D}'', \mathcal{D}')| \le \frac{C_L}{n} r(w, d), \qquad r(w, d) \le c_0\frac{d}{w}. \tag{10}$$

Here d is the input dimension, w is the proxy width, C_L absorbs fixed architecture, norm, and optimization factors, and c_0 is independent of n, $|o|$, d, and w. Intuitively, Equation (10) requires only local, not global, agreement: the target and proxy responses to the same data change converge as d/w decreases.

For each step, the change in Φ is exactly the difference between the target and proxy responses. Summing these changes along the path telescopes from $\mathcal{D}_{t-1}$ to $\mathcal{D}_t$, and Equation (10) gives

$$\begin{aligned}|\Phi(\mathcal{D}_t) - \Phi(\mathcal{D}_{t-1})| &\le \sum_{i=1}^{q}\left|\Delta_T(\mathcal{D}^{(i)}, \mathcal{D}^{(i-1)}) - \Delta_P(\mathcal{D}^{(i)}, \mathcal{D}^{(i-1)})\right| \\ &\le \frac{qC_L}{n} r(w, d) \le c_0 C_L \frac{qd}{nw}. \end{aligned} \tag{11}$$

This bound separates two effects: $|o|/n$ is the fraction of affected records, while d/w captures the finite-width response mismatch. Wide-network analyses provide context for such capacity-dependent approximation [5, 27]. Note that $r(w, d) \le c_0 d/w$ is an explicit proof assumption, not a rate derived from those results.

(4) Combining the bounds. Equation (8) shows that no other error source remains. Applying the triangle inequality together with Equations (9) and (11), we obtain, with probability at least $1 - \delta$,

$$\left|\overline{\Delta U}(o) - \Delta U(o)\right| \le c_0 C_L \frac{|o|d}{nw} + 2B\sqrt{\frac{\log(4/\delta)}{2m}}$$

The first term is the response mismatch accumulated along the operation path, while the second is finite-sample validation noise. Since $m = \Theta(n)$, for fixed δ and bounded B and C_L the right-hand side is $O\big((|o|/n)(d/w) + 1/\sqrt{n}\big)$. Taking the lower side of this two-sided bound yields Equation (7). Therefore, the corrected quantity $\overline{\Delta U}(o)$ minus this error is a lower confidence bound, while the two-sided result shows that the absolute proxy-recovery error converges to zero as the response and sampling terms vanish. □