From Principle to Practice: Vertical Data Minimization for Machine Learning

Robin Staab
ETH Zurich, Switzerland
robin.staab@inf.ethz.ch

Nikola Jovanović
ETH Zurich, Switzerland
nikola.jovanovic@inf.ethz.ch

Mislav Balunović
ETH Zurich, Switzerland
mislav.balunovic@inf.ethz.ch

Martin Vechev
ETH Zurich, Switzerland
martin.vechev@inf.ethz.ch

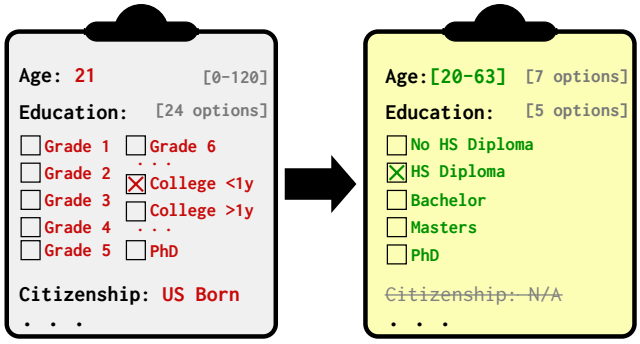
***Abstract*—Aiming to train and deploy predictive models, organizations collect large amounts of detailed client data, risking the exposure of private information in the event of a breach. To mitigate this, policymakers increasingly demand compliance with the *data minimization* (DM) principle, restricting data collection to only that data which is relevant and necessary for the task. Despite regulatory pressure, the problem of deploying machine learning models that obey DM has so far received little attention. In this work, we address this challenge in a comprehensive manner. We propose a novel *vertical DM* (vDM) workflow based on data generalization, which by design ensures that *no full-resolution client data is collected* during training and deployment of models, benefiting client privacy by reducing the attack surface in case of a breach. We formalize and study the corresponding problem of finding generalizations that both maximize data utility and minimize empirical privacy risk, which we quantify by introducing a diverse set of policy-aligned adversarial scenarios. Finally, we propose a range of baseline vDM algorithms, as well as *Privacy-aware Tree* (*PAT*), an especially effective vDM algorithm that outperforms all baselines across several settings. We plan to release our code as a publicly available library, helping advance the standardization of DM for machine learning. Overall, we believe our work can help lay the foundation for further exploration and adoption of DM principles in real-world applications.**

1. Introduction

Advances in machine learning (ML) have enabled organizations to automate tasks such as credit risk scoring [1] or fraud detection [2]. As ML models require large amounts of training samples, organizations increasingly collect different types of detailed client data, hoping to improve the models' performance. The deployment of such models, in turn, necessitates the collection of an even larger amount of highly-detailed client data for inference. These developments have led to growing regulatory concerns regarding the effects of large-scale data collection on individuals' privacy.

1.1. Data Minimization

In an attempt to address this issue, several authorities have developed regulations limiting data collection and processing.



Acc. 81.3% ✓ **Acc.** 80.3% ✓

Figure 1: Vertical data minimization (vDM) can greatly reduce the granularity of the data being collected, while not significantly impacting downstream ML models. We give a full overview of the chosen example in Section 8.4.

Most notably, such concerns are an important part of EU's General Data Protection Regulation (GDPR) [3], California's Privacy Rights Act (CPRA) [4], and the recent Blueprint for a U.S. AI Bill of Rights [5]. The GDPR, for example, defines *data minimization* (DM) in Article 5C as the principle of only collecting and using data that is *"adequate, relevant and limited to what is necessary in relation to the purposes for which it is processed"*. Similarly, the AI Bill of Rights Blueprint dictates *"ensuring that data collection conforms to reasonable expectations and that only data strictly necessary for the specific context is collected."* In the context of ML, this implies that the collection of any personal data (e.g., citizenship) has to be justified by the increase in utility of the resulting model. Furthermore, these regulations also apply for *model deployment*, a setting that many prior approaches cannot feasibly handle (Section 3).

DM in Practice. DM regulations have already had real-world impact—the EU has so far issued at least 166 fines due to DM violations [6]. An example is a fine of 2.75M euros issued in a case of fraud detection for childcare benefits [7], where the tax agency collected applicants' citizenship as a feature for their fraud detection model, even though it has later been shown that a simpler feature (being a resident or not) would have sufficed to achieve the same utility. Similarly,

the EU has fined a delivery company [8] for recording the GPS position of drivers every $12s$, despite less detailed information being sufficient for their purpose. Finally, there are various examples of fines issued when a security breach exposed sensitive data [6], which could have been avoided if such data was not unnecessarily collected to begin with.

On a positive note, there are examples of successful DM applications, e.g., the Norwegian Tax Authority reported using only 30 out of 500 considered variables in their tax error detection system [9]. Our experiments in Section 8 further indicate the feasibility of DM approaches in practice—we illustrate one such example in Figure 1, where collecting only the age group (e.g., 20-63) and the highest obtained diploma (e.g., High School) instead of exact age, educational status, and citizenship, results in minimal utility loss.

1.2. Principled Vertical DM for ML

Reducing the *number of data points* collected is an important and well-studied research problem [10], [11], [12], which can be interpreted as *horizontal data minimization* (hDM). We argue that hDM is not a suitable solution to many of the privacy concerns behind DM, as it offers *no privacy protection* for individual clients whose data was collected. In contrast, we therefore focus on *vertical data minimization* (vDM, formalized in Section 4), as the process of reducing the information collected *within each data point*, which directly protects clients that provide data during model training and especially deployment.

Despite these advantages of vDM, there has so far not been a well-established framework for the principled application and evaluation of vDM for ML. Without standardized evaluation, practitioners lack insight into how well their DM solutions protect client privacy in real-world settings. For example, as we will elaborate on in Section 2.2, one prior attempt to formalize vDM uses metrics that fail to capture the vulnerability to practical privacy attacks. This raises two key questions:

1) *What are the unique privacy challenges posed by DM regulations and why are existing approaches insufficient for addressing them?*

2) *Can we devise an approach for principled application and evaluation of vDM that addresses these concerns for practical machine learning use-cases?*

1.3. This Work

In this work, we tackle these questions, taking several steps to lay the foundations of vDM for ML.

The vDM Setting. First, we differentiate the vDM setting from similar settings highlighting how they are insufficient for addressing the requirements set forth by DM regulations. Afterwards we formalize the vDM setting via the concept of *generalization*, i.e., replacing detailed attributes with less granular ones (e.g., age with an age group). We define a workflow that organizations can employ, which after choosing a generalization based on a small set of full-granularity data points, ensures that during future model training and deployment *no full-granularity data is collected from the clients* (e.g., in a data collection survey, as illustrated in Figure 1), reducing the attack surface in case of a data breach and aligning with regulatory requirements.

To answer the question of which generalizations are suitable, we define two key objectives: *data utility*, which measures how useful the minimized data is for the downstream ML task (by training a classifier), and *empirical privacy risk*, which measures the potential to compromise individual privacy by observing minimized data (e.g., after a data breach). To quantify the latter, we formalize a comprehensive set of diverse and policy-aligned (see Section 5) *adversaries* with different attack objectives and capabilities, e.g., regarding their side information.

The vDM Algorithms. We introduce a range of strong baseline vDM algorithms, as well as *Privacy-aware Tree* (PAT), a vDM algorithm inspired by prior work on tree-based fair encoders [13]. We perform an extensive experimental evaluation of PAT and our baselines in various settings, demonstrating that PAT generally achieves the most favorable utility-privacy tradeoffs compared to all baselines. Our experimental findings highlight the importance of a principled evaluation of empirical privacy risks by illustrating how a naive evaluation would fail to capture key aspects of vDM.

1.4. Key Contributions

Our main contributions are:

- A formalization of the *vertical data minimization (vDM)* setting, and the underlying problem of generalizing the data such that it remains useful for the downstream task (*utility*) and exhibits low *empirical privacy risk* (Section 4).
- A formulation and instantiation of a comprehensive set of adversaries with different attack capabilities as a tool for evaluating the empirical privacy risk of vDM generalizations (Section 5).
- A diverse set of baseline vDM algorithms that can serve as a benchmark for future work (Section 6).
- A novel vDM algorithm, *Privacy-aware Tree* (PAT), achieving state-of-the-art results across multiple datasets (Section 7).
- An extensive experimental evaluation of all introduced vDM algorithms and adversaries on several real-world datasets highlighting the practical applicability of vDM (Section 8).
- A library with all our adversaries, baselines, and PAT , advancing the standardization of vDM available at https://github.com/eth-sri/datamin.

2. Background

In this section, we will introduce the background necessary for the subsequent parts of the paper.

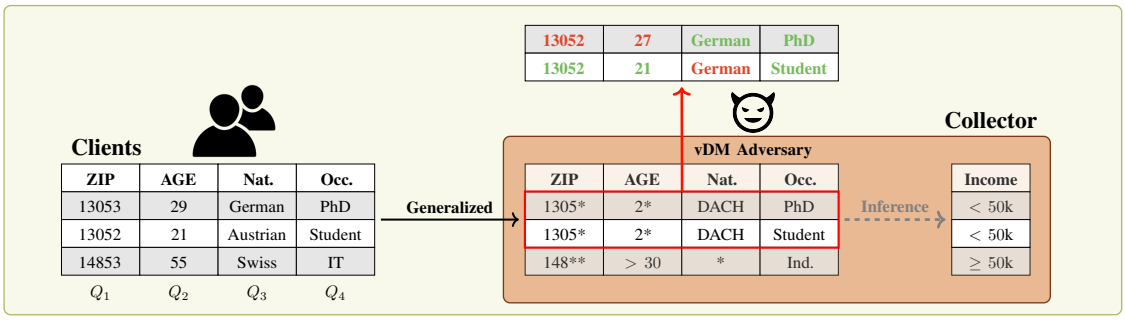


Figure 2: Overview of the vDM deployment setting on an ACSIncome toy dataset (see Section 8). Clients directly provide generalized data, i.e., adversary and collector (which could be the same) only observe generalized records. The vDM adversary tries to reconstruct the original values of the generalized attributes (here all Q_i). The collector runs a downstream ML model on the generalized data for inference.

2.1. Machine Learning

In recent years ML algorithms, specifically deep neural networks, have been applied in classification tasks in a wide variety of settings [14], [15]. Let $\mathcal{D}$ be a data distribution over $\mathcal{X} \times \mathcal{Y}$ and $(x, y) \sim \mathcal{D}$ be a data point consisting of a *record* x and its discrete label $y \in \mathcal{Y} = \{1, \dots, c\}$. The goal of an ML algorithm is to learn a mapping function $f_\theta \colon X \to Y$ (e.g., a neural network) parameterized by θ (e.g., neural network weights), which maps records $x \in X$ to $f_\theta(x) \in Y$ aiming to minimize some objective function $\mathcal{L}$. We will refer to $f = f_\theta$ as a *model*. For classification, we set $\mathcal{L} = \mathbb{E}_{(x,y)\sim\mathcal{D}}[f(x) \neq y]$, the expected rate of misclassification, which in training is approximated using finite samples $S_{train} \sim \mathcal{D}$ (*training set*). We assume that f is trained on S_{train} and evaluated on a distinct *test set* $S_{test} \sim \mathcal{D}$.

2.2. Attribute Generalization and NCP

In ML the term *generalization* commonly refers to *model* generalization, i.e., the model's performance on unseen data. For the scope of this work, we define *attribute* generalization (hereafter referred to as *generalization*) as a function $g \colon X \to Z$ on our data $X \subseteq \mathbb{R}^d$. This is referred to as a *global* generalization as it fixes a single g for all data points. *Local* generalizations, on the other hand, allow different generalization functions for separate data points, enabling two data points with identical values to map to different generalized points. As in [16] we say that g is *single-dimensional* when it can be decomposed into a set of g_i such that $g(x) = (g_1(x_1), \dots, g_d(x_d))$ and *multidimensional* otherwise. Finally, generalizations can be either *strict*, ensuring that the image of each g_i forms a proper partition of the respective attribute domain, or *relaxed*, allowing elements of g_i's image to have non-empty intersections (e.g., `age` both to ranges 20-30 and 25-32).

NCP. We recall the definition of the *normalized certainty penalty (NCP)*, a metric commonly used in data anonymization settings to quantify the information loss of a generalization [17], [18]. For brevity, we focus only on categorical attributes. Assume a generalization g defined on d-dimensional data, and an attribute i with domain D_i. For a generalized record $g(\boldsymbol{x}) = \boldsymbol{z}$ we set $NCP_i(\boldsymbol{z}) = 0$ if $\boldsymbol{z}_i = \boldsymbol{x}_i$, and $NCP_i(\boldsymbol{z}) = |\boldsymbol{z}_i|/|D_i|$ otherwise, where we define $|\boldsymbol{z}_i|$ as $|g_i^{-1}(\boldsymbol{z}_i)|$ (the size of the pre-image of $\boldsymbol{z}_i$). The per-attribute $NCP_i(\boldsymbol{z})$ is combined using pre-selected attribute weights w_i to obtain the $NCP(\boldsymbol{z}) = \sum_{i=0}^{d-1} w_i \cdot NCP_i(\boldsymbol{z})$. Given a dataset with n points this is combined to a (normalized) Global Certainty Penalty $GCP = \frac{1}{n}\sum_{i=1}^{n} NCP(\boldsymbol{z}^{(i)})$ [19].

The only prior work [19] attempting to formalize vDM relied on NCP-based metrics to quantify the privacy risk of a generalization. We argue, however, that NCP is a generic information loss metric incapable of accurately reflecting the adversarial vulnerability of the data when vDM is applied in real-life scenarios. Assume, e.g., an attribute a with possible values $\{a_1, a_2, a_3, a_4\}$ used for medical diagnosis, where $a \in \{a_1, a_2\}$ implies that the patient requires medication, while $a \in \{a_3, a_4\}$ implies otherwise. For an adversary with knowledge of the generalization, generalizing a such that $\{a_1, a_2\} \to g_1$ and $\{a_3, a_4\} \to g_2$ (Gen. 1) reveals whether a patient needs medication. Generalizing $\{a_1, a_4\} \to g_1$ and $\{a_2, a_3\} \to g_2$ (Gen. 2) does not leak this information. Despite this, both generalizations have a GCP score of 0.5, making it impossible to distinguish between them. Motivated by this, we advocate for a more realistic measure of privacy risk that directly quantifies an adversary's success at compromising client privacy in the vDM setting. Based on existing legislature [20], we will formalize a comprehensive set of relevant adversaries in Section 5.

2.3. Personal & Sensitive Data

Legal Definitions. In the EU, the introduction of the GDPR has led to the establishment of a clear definition of *personal* data. According to Article 4 of the GDPR [3], personal data is defined as *"any information relating to an identified or identifiable natural person ('data subject')."* This definition is more rigorous than the Personal Identifiable Information (PII) definitions commonly employed under U.S. jurisdiction. The U.S. Department of Labor defines PII as *"any representation of information that permits the identity of an individual to... be reasonably inferred"* [21]. Both in GDPR and PII,

there is a concept of (especially) *sensitive* data. The GDPR details this in Article 9 as a set of sensitive (special category) personal data (e.g., relating to race, sexual orientation, religion) for which special care needs to be taken.

Technical Definitions. As we elaborate on in Section 3, the well-established research areas of *privacy-preserving data publishing (PPDP)* [22] and, within it, *data anonymization (DA)*, operate with a narrower definition of sensitive data. The standard PPDP setting assumes that a *data collector* wants to release a (fixed) table T of data entries with attributes $(T_1, \ldots, T_d)$ (T_i denoting the i-th attribute), out of which only *one* (instead of all personal attributes) is sensitive.

In particular, each attribute T_i is *at most one* of the following: *Unique Identifier* U_i which directly identifies a single entry, *Quasi-Identifier* $Q = (Q_1, \ldots, Q_{d_q})$, which allows unique identification of at least one record in T by examining the attributes in Q, or *Sensitive* S whose value we want to protect from being connected with any particular individual in T. Some PPDP works [23], [24] attempt to relax the assumption of a single S. However, this usually happens at the cost of data utility [24] and often requires the data to be sliced into multiple tables, each containing only parts of the sensitive attributes. This gap between stricter regulatory requirements (requiring the protection of all personal attributes) and existing work is one of the motivations for our vDM setting in Section 4.

As in PPDP/DA we will focus on tabular data. This both follows a long line of work in related areas [13], [25] and also fits regulatory requirements, which are primarily focused on sensitive attributes in tabular format [3], [26].

3. Motivating a New vDM Setting

In this section, we motivate the need for a new vDM setting in two steps: In Section 3.1, we derive concrete requirements for the vDM setting directly from regulations such as GDPR [3] and the U.S. AI Bill of Rights Blueprint [5]. Using this, we show in Section 3.2 how current privacy-enhancing technologies (PETs) fail to address specific parts of these requirements (summarized in Table 1), motivating our vDM formalization in Section 4.

3.1. A Regulation-Guided vDM Setting

As our vDM setting focuses on data minimization in ML, *the specific purpose* of client data, as required by GDPR Article 5C, is to provide an accurate ML model for *a specific* task (e.g., for medical diagnosis) while protecting client privacy. To properly evaluate this via adversarial risk (as done in Section 5), we first have to clarify assumptions made on the adversary and clients.

Protecting Data Collection. The U.S. AI Bill of Rights dictates that *"...only data strictly necessary for the specific context is collected"*, putting the emphasis of the vDM setting on data collection not only for model training but also for model inference when (new) client data processing shall be "limited to what is necessary" (GDPR Article 5C). As depicted in Figure 2 on a toy example, this implies that the vDM adversary (formalized in Section 5) is positioned between the client and the data collector or even is an honest-but-curious collector. The goal of the vDM adversary hereby is to reconstruct *all* personal attribute values from an observed generalized record. This differs from the adversaries assumed in some other PETs (Section 3.2).

Client Assumptions. Inherent to this focus on data collection is the need to clarify client capabilities. To capture a wide range of use cases, we want to minimize the number of assumptions made about the client. In particular, the vDM setting does not assume any cryptographic capabilities or possibilities to interact with the collector—minimization should happen only on the data with a focus on the amount of data *collected* from clients. This approach is orthogonal to many PETs that aim to protect full-resolution data collection (Section 3.2). To avoid assumptions on client capabilities in vDM, we will require our generalization g to be (1) directly usable on the client side and (2) easily applicable to new data points for inference. Namely, clients should be able to enter values (e.g., `age`) independently of other values (i.e., g is single-dimensional), and data entry should remain consistent across clients (i.e., g is global). Further, we must ensure that each original attribute value has exactly one corresponding generalized value (i.e., g is strict).

3.2. Other Privacy-Enhancing Technologies

We now discuss other commonly applied PETs in ML, highlighting how they are unable to address the vDM setting outlined in Section 3.1. Throughout this section, we split a typical ML workflow into three stages: Initial data *collection*, model *training*, and *deployment*, allowing for a more fine-grained analysis as shown in Table 1. For data collection, we first distinguish whether the PET makes strong assumptions about client capabilities and whether clients assume the collector to be trusted. We then categorize the privacy of client records sent to the collector (*Privacy (Wire)*). The collector uses this data during the training stage to train an ML model. For this, we rank both technical feasibility (e.g., expensive computations, decreased performance) and the privacy of the resulting model against membership inference attacks (*Privacy (Model)*) [27], [28]. With a trained model, we proceed to the deployment (inference) stage. Again we first categorize assumptions on (new) clients and respective trust in the collector. Additionally, we rate how technically feasible the PET is for inference on new records as well as the records' privacy protection (*Privacy (New Record)*).

Federated Learning. Federated Learning aims to protect client privacy by letting them train models locally, only combining the resulting models/gradients on a server [29]. This, however, comes with the strong technical requirement of clients being capable of training a model locally (denoted by ◔ in Table 1) and also was shown to be vulnerable to gradient inversion attacks [30] leaking full resolution data.

Differential Privacy. Differential privacy (DP) [31] has been widely recognized as the privacy standard in various data

TABLE 1: Comparison of PETs to the vDM setting. PETs are rated in each category on a scale {○, ◔, ◑, ◕, ●} with more detail in Section 3.2. We rate assumptions made on client capabilities and trust in the collector during collection/deployment from ○ (*strong*) to ● (*none*). Further, we rate current technical feasibility and client privacy from ○ (*none*) over ◕ (*empirical*) to ● (*guaranteed*). For inference, only vDM yields a feasible solution in an untrusted collector setting.

	Collection			Training		Deployment			
	Client Assumptions	**Trust in Collector**	**Privacy (Wire)**	**Technical Feasibility**	**Privacy (Model)**	**Client Assumptions**	**Trust in Collector**	**Technical Feasibility**	**Privacy (New Record)**
Fed. Learning	◔	◑	◑	◑	◑	◔	-	◔	●
DP (Central)	●	○	○	◕	●	●	○	●	○
DP (Local)	◑	●	●	◔	●	◑	-	-	-
E2E-Crypto	◑	○	●	●	◔	◑	○	●	○
FHE	◑	●	●	◔	●	◑	●	◔	●
SMC	◑	-	●	◔	●	◔	-	◔	●
PPDP/DA	●	○	○	-	-	-	-	-	-
Synthetic Data	●	○	○	●	●	●	○	●	○
vDM	●	◔	◑	●	◑	●	●	●	◕

analytics applications due to its rigorous privacy assurances independent of the adversary's background knowledge. We differentiate between local and central DP [32], both of which try to make it hard to determine whether data of a specific client is included in a training dataset (membership inference) while still allowing meaningful conclusions to be drawn from the aggregate data.

In ML settings, central DP is commonly achieved by adding noise to gradients during training [33]. vDM differs from central DP already in the setup, i.e., by focusing on limiting the amount of personal data put into the system instead of the amount remaining in the model. In particular vDM is concerned with protecting privacy during deployment where centralized DP does not offer any privacy for new clients (○ for *Privacy (New Record)* in Table 1).

Local DP, on the other hand, requires clients to perturb their data locally before sending it to an (untrusted) collector. While closer to the vDM setting, it is infeasible for vDM for three reasons: (1) It requires active participation of clients for perturbation, (2) practical applications of local DP in ML are limited in scalability [32], and (3) it offers no privacy in an inference setting where clients want to receive results on their non-perturbed data (in Table 1 we therefore consider it non-applicable for deployment).

Cryptographic Approaches. For cryptography-based approaches, we differentiate between (1) Simple encryption between client and collector (E2E), (2) Fully-Homomorphic-Encryption (FHE) based schemes [34] which allow ML on top of the encrypted data, and Secure Multi-party Computation (SMC). The first two approaches require clients to have a secure device capable of cryptographic capabilities (◑ for *Client Assumptions* in Table 1). Furthermore, while E2E encryption protects data in transit, it offers no privacy guarantee in case of a curious collector.

FHE, on the contrary, protects data during transit, training, and deployment. However, due to its heavy use of cryptographic primitives, it is limited in *Technical Feasibility* (◔ in Table 1) [35] due to large overheads in memory and runtime [36], [37] or requiring specific architectures [37].

SMC goes beyond this, avoiding a central collector by requiring active participation of (most) clients in the computation [38]. While enabling strict privacy guarantees, this makes it infeasible for larger client sizes and many ML use-cases [39]. A recent line of work has combined SMC with versions of FHE for *secure neural network inference* [40], [41]. While feasible for some scenarios, this approach still suffers from clients having to actively participate in the computation (e.g., to evaluate activation layers [40]).

PPDP/DA. As mentioned in Section 2.3, PPDP aims to protect individuals in a table T from being connected to their sensitive attribute value by releasing only an anonymized data table T' (alongside S). Records in T' consist of generalized, suppressed, and perturbed quasi-identifiers of T, with the PPDP adversary aiming to map specific records in T' to their respective sensitive attribute value. The degree of privacy protection through T' is commonly formalized by ensuring one of the following: each record in T' is indistinguishable from at least $k-1$ other records (k-anonymity [42]), the values for S are well spread (l-diversity [43]), or that the distribution of the sensitive attribute is close to the distribution over the entire dataset (t-closeness [44]). Note that these constraints, as they focus solely on re-identification risk, can often be significantly misaligned with our privacy notion of individual attribute protection.

Unlike vDM, PPDP does not consider inference, i.e., data is released once, and the used generalizations are not required to be applicable to new data points as in vDM. For this reason, most PPDP work focuses on non-strict, multidimensional [16] generalizations, or relies on *perturbing* and *permuting* data in T [45]. While this broadens the space of solutions in the PPDP setting, it makes most algorithms directly non-applicable to the *Deployment* setting in Table 1.

Synthetic Data Generation. Recently, Synthetic Data Generation (SDG) [46], [47] has gained popularity as another

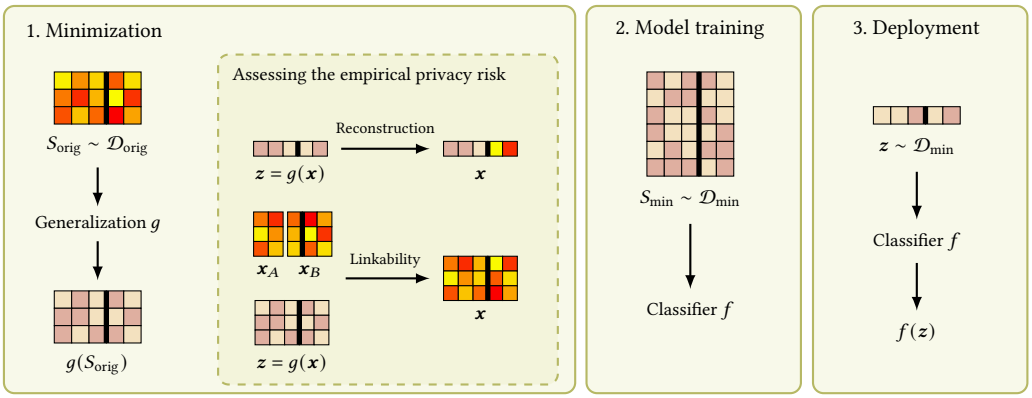


Figure 3: Overview of our vertical data minimization (vDM) workflow. In the *minimization* phase a minimizer proposes a generalization g using a limited sample of full-granularity original data S_{orig}, where each record (here, each row) consists of *non-personal* (here, first 3) and *personal* (here, last 2) attributes. The empirical privacy risk of g is then assessed using a wide range of adversaries (here we show only two examples). In the *model training* phase, we collect a large sample S_{min} of generalized data to train the classifier f, and use it for inference in the *deployment* phase. We use saturated colors, ▮▮▮▮▮, to indicate full-granularity attributes, and desaturated colors, e.g., ▮▮, to indicate generalized attributes.

approach to address similar privacy concerns as PPDP. Unlike PPDP, where we generalize original data, SDG trains a model on the original dataset and uses it to generate entirely new data points. These synthetic data points resemble the original data but do not directly correspond to any individuals. In a deployment setting, SDG cannot protect the privacy of new clients sending their real data (○ in Table 1).

4. Formalizing vDM for ML

Having established the key vDM requirements and highlighted shortcomings of existing PETs, we now formalize the vDM setting, describe the corresponding workflow, and discuss instantiations of utility and empirical privacy risk tailored to the context of ML.

Overview. Assume an organization aims to solve a prediction task by learning and deploying a classifier f to predict labels $y \in \mathcal{Y}$ from records $x \in \mathcal{X} \subseteq \mathbb{R}^d$ with joint distribution $(\boldsymbol{x}, y) \sim \mathcal{D}_{\text{orig}}$. To train f well, the set of training pairs $(\boldsymbol{x}, y)$ should be large and records $\boldsymbol{x}$ as detailed as possible. At the same time, we want to protect the privacy of individuals.

To treat this tradeoff in a principled manner, we introduce the *vDM workflow* depicted in Figure 3 and explained in detail below. Importantly, after having chosen the generalization g in the *minimization* phase (transforming $\mathcal{D}_{\text{orig}}$ into the generalized $\mathcal{D}_{\text{min}}$), *no* full-granularity data from $\mathcal{D}_{\text{orig}}$ is collected during the *model training* or *deployment* phases.

To accurately represent the empirical privacy risk of a generalization, we introduce a wide range of policy-aligned adversaries in Section 5. In particular, we directly reference the EU Working Party [20], which outlines the threats of *inference* (i.e., *reconstruction*), *linkability*, and *singling out* for anonymized data. We formalize these concepts and define several adversaries that aim to *reconstruct* the personal attributes of a client $\boldsymbol{x}$ based on the leaked generalized record $\boldsymbol{z}$, utilizing different degrees of side information, as well as adversaries that aim to use the generalized dataset to link two partial datasets $\boldsymbol{x}_A$ and $\boldsymbol{x}_B$ (*linkability*) or isolate a single individual from the dataset (*singling out*).

Minimization via Generalization. Formally, we propose to train f on low-granularity *generalized* records $\boldsymbol{z} \in \mathcal{Z}$ instead of full-granularity records $\boldsymbol{x} \in \mathcal{X}$. To this end, we define a *generalization* function $g\colon \mathcal{X} \to \mathcal{Z}$ (*global*), which reduces data granularity, produced by a *minimizer* (an algorithm or a human). We set $\mathcal{Z} = \mathcal{Z}_1 \times \mathcal{Z}_2 \times \ldots \times \mathcal{Z}_d$ and $\mathcal{Z}_i = \{1, 2, \ldots, k_i\}$ such that $g(\boldsymbol{x}) := (g_0(x_0), g_1(x_1), \ldots, g_d(x_d))$ generalizes each attribute independently (*single-dimensional*). Further, we require the image of g to be a proper partition of each dimension (attribute) of $\mathcal{X}$ (*strict*). We distinguish continuous attributes (e.g., `salary`) and discrete ones (e.g., `occupation`). Continuous x_i are scaled to $[0, 1]$ and transformed using a non-decreasing function $g_i\colon [0, 1] \to \mathcal{Z}_i$. Discrete x_i with values $\{1, 2, \ldots, c_i\}$, where $c_i \geq k_i$, are transformed using $g_i\colon \{1, 2, \ldots, c_i\} \to \mathcal{Z}_i$. We use $\mathcal{D}_{\text{min}}$ to denote the induced distribution of $(\boldsymbol{z} = g(\boldsymbol{x}), y)$. Finally, we note that all our generalizations are both independent of respective downstream classifiers (results in Appendix G indicate that they transfer well between different downstream architectures) and as depicted in Section 8.4 easy to apply for clients, justifying the ● on *Client Assumptions* in Table 1.

Proposed Workflow. With this setup, the *vDM workflow*, depicted in Figure 3, consists of three distinct phases aiming to minimize the amount of full-resolution data that ever enters the system (i.e., has to be collected from clients). In the *minimization phase*, we collect a small (see Section 8.3), well-protected set S_{orig} of full-granularity pairs $(\boldsymbol{x}, y) \sim \mathcal{D}_{\text{orig}}$ (◔ for *Trust in Collector*), which a minimizer uses to propose a set of generalizations $g^{(i)}$. We evaluate those in terms of utility and empirical privacy risk and select the most suitable one (g). Afterward, there is no more need to collect full-resolution data from clients. In the *model training phase*, we collect a large set S_{min} of *generalized* pairs $(\boldsymbol{z}, y) \sim \mathcal{D}_{\text{min}}$

TABLE 2: Summary of all adversaries introduced in Section 5 to evaluate the empirical privacy risk. All adversaries, as prior knowledge, have access to the generalization g and a small set of full-granularity records S'_{orig}. As part of the breach, they additionally have observed a set of generalized records S'_{min}. We evaluate all adversaries in Section 8.

Attack vector		Adversary	Side information	Goal
Reconstruction	A1	Reconstruction	/	Reconstruct personal attributes $\boldsymbol{x}_P$
	A2	High-certainty Reconstruction	/	Reconstruct $\boldsymbol{x}_P$ with high confidence
	A3	Non-personal Knowledge	$\boldsymbol{x}_N$	Reconstruct personal attributes $\boldsymbol{x}_P$
	A4	Leave-one-out	$\boldsymbol{x}_N$ and $\boldsymbol{x}_{P\setminus\{p\}}$	Reconstruct attribute x_p
	A5	Partial Personal Knowledge	$\boldsymbol{x}_N$ and $\boldsymbol{x}_{\{1,\dots,k-1\}}$	Reconstruct attribute x_k
	A6	Multi-breach Reconstruction	Another set of g and S'_{min}	Reconstruct attributes $\boldsymbol{x}_P$
Linkability	A7	Linkability	$\boldsymbol{x}_A$ and $\boldsymbol{x}_B$ for k individuals	Link each $\boldsymbol{x}_A$ to a corresponding $\boldsymbol{x}_B$
Singling Out	A8	Singling Out	/	Isolate a single individual

and train a classifier f on it (◑ for *Privacy (Wire)* as we already collected S_{orig}). vDM imposes no restrictions on used architectures/algorithms and we show in Section 8 that the resulting loss in utility is small (● for *Technical Feasibility*). Finally, in the *deployment phase*, we deploy f for inference and answer queries from clients, who only need to input their *generalized* records $\boldsymbol{z}$ (● for *Trust in Collector*, ◕ for *Privacy (New Record)*) and receive predictions $f(\boldsymbol{z})$.

As shown in Figure 3, during the training and deployment phases, *no full-granularity data enters the system*, e.g., individuals are never asked about their exact age. This simplifies the security analysis, as regardless of the breach target (e.g., collection, processing, or storage), the only data that can be leaked in the latter two phases are generalized records. This also entails that membership inference attacks can only recover generalized records (◑ for *Privacy (Model)*). While the minimization phase still requires some full-granularity data, we argue that this phase can be protected against breaches more easily as opposed to a live deployment—data policies can be stricter, e.g., via external auditing, ensured data deletion, or the use of commercial data clean rooms [48], [49], which can be impractical for a live deployment.

Evaluating Generalizations. As previously discussed, g should produce generalized data with both *high utility* and *low empirical privacy risk*, two goals generally at odds. The former means that generalized data should contain enough information to solve the original task—formally, to keep the utility risk $\mathcal{UR}(g)$ low, which is defined as the error rate of the best possible classifier f predicting y from $\boldsymbol{z} = g(\boldsymbol{x})$:

$$\mathcal{UR}(g) := \min_{f} \mathop{\mathbb{E}}_{(\boldsymbol{z},y)\sim\mathcal{D}_{\text{min}}} \mathbb{1}\left\{f(\boldsymbol{z}) \neq y\right\}. \tag{1}$$

Evaluating the *empirical privacy risk* of g is more involved. Namely, despite the advantages of the proposed workflow noted above which reduces the attack surface in case of a data breach, it remains unclear how much the generalized records leaked in the training or deployment phases reveal about individuals, to which extent their privacy is protected, and how this should be quantified. We thoroughly study this question next.

5. Assessing the Empirical Privacy Risk

In this section, we formulate a comprehensive set of *adversaries* with different attack capabilities, all aiming to use the data from a breach to compromise client privacy in various ways, often by training adversarial models. Our adversaries can serve as a tool for evaluating vDM or provide insights to organizations in the minimization phase. For this, we define a subset of attributes $P \subseteq \{1, 2, \dots, d\}$ as *personal* (with the rest $N = \{1, 2, \dots, d\} \setminus P$ being non-personal).

Threat Model. We start by defining the assumed prior knowledge of our adversaries. First, we assume all adversaries know the generalization g being attacked, as g needs to be open to every party providing data in the training and deployment phases. Second, all adversaries have a small set S'_{orig} of full-granularity records $\boldsymbol{x}$ (we ignore y here for simplicity)—this can be obtained either by a breach in the minimization step ($S'_{\text{orig}} \equiv S_{\text{orig}}$, the setting considered in Section 8) or by obtaining other samples from $\mathcal{D}_{\text{orig}}$. Notably, this implies that the adversary knows $g(S'_{\text{orig}})$, i.e., the generalized records corresponding to S'_{orig}.

With this prior knowledge of g and S'_{orig}, we define a *breach* as an event where the adversary observes a set of generalized records S'_{min}, obtained by compromising the data collection or storage pipelines in the training or deployment phase, or even via model inversion [50] on f. *All* adversaries we will now introduce share this base threat model, reflecting different ways of utilizing S'_{min} to compromise client privacy, often with additional side information. Note that our base threat model is quite generous to the adversary (e.g., in terms of knowledge of $\mathcal{D}_{\text{orig}}$), and thus even our weakest adversary models a relatively strong attacker.

Overview of Adversaries. Our set of adversaries is diverse, taking into account various levels of side information, but more importantly, directly aligned with policy, with each adversary corresponding to attacks described in [20] (recently studied for synthetic data in [51]). While practitioners can decide that some threats are more relevant, this broad set of adversaries enables a thorough evaluation of the empirical privacy risk. We now describe the adversaries (labeled A1-A8 and summarized in Table 2). We develop practical implementations of all adversaries and apply them in Section 8 evaluating our minimizers from Sections 6-7.

(A1-A2) Reconstruction. The goal of the reconstruction adversary is to use its prior knowledge of S'_{orig} to learn a function h, which can be used to recover personal attributes $h(\boldsymbol{z})_P \approx \boldsymbol{x}_P$ of breached records $\boldsymbol{z} \in S'_{\text{min}}$. Formally, the *Reconstruction (A1)* adversary aims to find h minimizing

the error rate of predicting $\boldsymbol{x}_P$ from $\boldsymbol{z}$. We can use this to define a corresponding empirical privacy risk

$$\mathcal{PR}_{A1}(g) := \min_{h} \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}_{\text{orig}}} \left[\frac{1}{|P|} \sum_{p\in P} \mathbb{1}\left\{h(g(\boldsymbol{x}))_p \neq x_p\right\} \right],$$

which a practical implementation of A1 approximates by sampling from S'_{orig}. When combining different personal attributes in S, we focus on mean aggregation as a reasonable choice for the generic case. We explore this choice for adversaries in Section 8.2 and minimizers in Appendix B.

However, leaking one personal attribute of a single individual with high certainty often has more direct privacy implications than an aggregate metric over many attributes and individuals. Thus, we introduce the *High-certainty Reconstruction (A2)* adversary, which has the same goal as A1 and is trained the same way but calculates the *confidence* for each predicted attribute (i.e., how certain it is that the prediction is correct) and outputs only predictions with the highest confidence. While we refer to Section 8 for details, intuitively, A2 uses logit magnitudes as a proxy for *confidence*, identifying data points that have particularly high empirical privacy risk. We note that empirical privacy risk under A2 can be formalized similarly to $\mathcal{PR}_{A1}(g)$. While we focus on A2 as a variant of A1, the idea of confidence can, in principle, be applied to any adversary.

(A3-A6) Reconstruction with Side Information. A typical scenario is that an adversary has *side information* about an individual (potentially from another breach) and is aiming to utilize this to boost the leakage of unknown personal attributes [52]. To investigate this for vDM, we instantiate several reconstruction adversaries strictly stronger than A1.

The *Non-personal Knowledge (A3)* adversary has knowledge of all non-personal attributes and thus aims to use $(\boldsymbol{z}, \boldsymbol{x}_N)$ to reconstruct $\boldsymbol{x}_P$. The significantly stronger *Leave-one-out (A4)* adversary models the worst case, knowing $(\boldsymbol{z}, \boldsymbol{x}_N, \boldsymbol{x}_{P\setminus\{p\}})$, i.e., all other attributes when predicting x_p. The *Partial Personal Knowledge (A5)* adversary models the intermediate cases between A3 and A4, giving us granular insight into how gracefully a generalization degrades under side information. Assuming $P = \{1, \ldots, p\}$, A5 predicts the personal attribute $\boldsymbol{x}_k$ having knowledge of $(\boldsymbol{z}, \boldsymbol{x}_N, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{k-1})$. The cases $k = 1$ and $k = p$ recover A3 and A4, respectively. In Section 8, we evaluate A5 by averaging its error over all choices of k.

Finally, the *Multi-Breach Reconstruction (A6)* adversary focuses on the case where side information comes from the same source due to several breaches at different points in time. Assume the case where an organization switches from a generalization $g^{(1)}$ to a different generalization $g^{(2)}$, and the adversary observes two breaches S'_{min} and S''_{min} corresponding to the same individuals. Intuitively, observing two (or k) sufficiently different generalizations of the same individual can boost reconstruction. The goal of A6 is to evaluate the resilience of minimizers to repeated breaches.

(A7-A8) Linkability and Singling Out. Next, we consider two non-reconstruction adversaries motivated by [20]. The increasing availability of data makes these attacks common in practice and often an essential first step towards mounting more powerful attacks [51], [53], [54], [55].

The *Linkability (A7)* adversary observes as side information, e.g., from another organization using similar attributes, full-granularity *partial records* (i.e., records with a subset of attributes) for a set of individuals, and aims to use the set of generalized records S'_{min} to connect partial records belonging to the same individual. More formally, for disjoint subsets A and B of attributes $\{1, \ldots, d\}$, A7 has k partial records of the form $\boldsymbol{x}_A$ and $\boldsymbol{x}_B$, and aims to predict for each $\boldsymbol{x}_A$ which $\boldsymbol{x}_B$ corresponds to the same individual.

Finally, the goal of the *Singling Out (A8)* adversary is to isolate a single individual from the dataset, a concept similar to having a small anonymity set which is a known issue in privacy-related areas [56]. Formally, it observes the breached generalized records S'_{min} and outputs a predicate Π that, when applied to the full-granularity records that produced S'_{min}, return exactly one individual.

6. Baseline Minimizers for vDM

Having introduced both the vDM setting and our adversaries, we now present several vDM algorithms that will establish baselines for our PAT minimizer (Section 7).

Uniform Minimizer. As an initial simple baseline, we consider a uniform minimizer. Let k_i be a hyperparameter that denotes how many elements (buckets) $\mathcal{Z}_i$ should at most have. Given a hyperparameter k, the uniform minimizer uses $k_i = k$ for all attributes and generalizes discrete attributes x_i uniformly at random, and continuous attributes x_i to $\lceil kx_i \rceil$.

Feature Selection Minimizer. As attribute suppression (i.e., removal of an attribute) is a special case of generalization, we consider a feature selection minimizer which keeps k of d attributes based on ANOVA F-values, as this method supports both continuous and categorical attributes. In Appendix E we further explore more methods and variants.

Apt Minimizer. We adapt the recently proposed *Apt* method [19] (discussed in Section 10.3) to our setting. *Apt* uses information loss metrics and a decision tree to model the decisions of a classifier trained on original data. As the method was infeasible to run for dataset sizes we consider ($> 24h$ where our minimizer PAT, requires $\approx 1s$), we limit the tree-depth in *Apt* to 10 and each run to $2h$.

Iterative Minimizer. Given k, the *Iterative* minimizer starts from a heuristic generalization g with $k_i = k$, splitting continuous attributes based on k-quantiles and discrete ones based on their weight in a logistic regression model, minimizing the average variance of weights in each group. It then iteratively improves g by reducing k_i using dynamic programming while keeping the resulting classifier error below some threshold. To determine the order in which attributes are generalized, it sorts all attributes based on their estimated impact on classification and adversarial error. We refer to Appendix C for a more detailed description.

Neural Minimizers. Finally, we propose two vDM minimizers that model g using neural networks. We present a brief overview of how such modeling is done for continuous attributes and provide a corresponding description for discrete attributes and more details in Appendix C.

Both minimizers model g as a set of d independent neural networks $g^{(i)}$, with each network responsible for generalizing a single attribute. Let i denote the index of a continuous attribute of record $\boldsymbol{x}$ and let $\boldsymbol{x}_i$ be normalized to $[0,1]$. $g^{(i)}$ learns a monotonic and differentiable generalization by first learning a monotonic transformation $M\colon [0,1] \to [0,1]$. Based on work on monotonic neural networks [57], [58], g ensures M's monotonicity by constraining all linear layer weights to $W \odot W \geq 0$, using tanh activations and batch normalization. The output interval is then split uniformly into k_i buckets (identified by center points c_j), and for a record $\boldsymbol{x}$, the probability of generalizing attribute i to bucket j is taken as the softmax over the bucket-distances $(g^{(i)}(\boldsymbol{x}_i) - c_j)^2$.

AdvTrain Minimizer. The *AdvTrain* minimizer, inspired by [59], utilizes adversarial learning [60] to jointly optimize the generalization $g_\psi : \mathcal{X} \to \mathcal{Z}$, classifier $f_\theta : \mathcal{Z} \to \mathcal{Y}$ and adversary $h_\phi : \mathcal{Z} \to \mathcal{X}_s$, all of which we model as neural networks. *AdvTrain* then optimizes the following objective:

$$\min_{\theta,\psi} \max_{\phi} \mathbb{E}_{\boldsymbol{x},y} \left[(1-\lambda) L_{\text{clf}}(f_\theta(g_\psi(\boldsymbol{x})), y) - \lambda L_{\text{adv}}(h_\phi(g_\psi(\boldsymbol{x})), \boldsymbol{x}) \right],$$

where λ is a factor determining the tradeoff between the classification and adversarial error. We instantiate L_{clf} as the BCE loss between the predicted and true label, while L_{adv} denotes the CE loss between the predicted and true personal attribute (averaged over all such attributes). During training, we optimize g_ψ and f_θ to reduce L_{clf} and increase L_{adv} and optimizing h_ϕ to reduce L_{adv}. For each step of optimizing g_ψ and f_θ, we take N_{inner} steps optimizing h_ϕ.

MutualInf Minimizer. The final baseline, *MutualInf*, uses the same g_ψ, f_θ, and training procedure as *AdvTrain* but replaces the adversarial objective with one minimizing the mutual information between $\boldsymbol{x}$ and its generalization $\boldsymbol{z} = g_\psi(\boldsymbol{x})$:

$$\min_{\theta,\psi} \mathbb{E}_{\boldsymbol{x},y} \left[(1-\lambda) L_{\text{clf}}(f_\theta(g_\psi(\boldsymbol{x})), y) + \lambda L_{\text{inf}}(g_\psi(\boldsymbol{x}), \boldsymbol{x}) \right].$$

To derive L_{inf}, we start from the definition of mutual information $I(\boldsymbol{z}, \boldsymbol{x}) = H(\boldsymbol{z}) - H(\boldsymbol{z}|\boldsymbol{x})$, and apply Jensen's inequality to independently derive upper bounds on $H(\boldsymbol{z})$ and $H(\boldsymbol{z}|\boldsymbol{x})$, which can be approximated via sampling. In Appendix C we describe this process in more detail.

7. Privacy-aware Tree Minimizer

We now propose our minimizer, *Privacy-aware Tree* (PAT), that builds a generalization g using a decision tree.

Classification Trees. We first recall key concepts related to classification trees. Let $\mathcal{S}_{\text{root}} = (\boldsymbol{x}, y) \in \mathbb{R}^d \times \{0,1\}$ be a dataset for binary classification. A decision tree T repeatedly splits a leaf node L with assigned dataset $\mathcal{S}_L$ into children nodes $L^{\leq}$ and $L^{>}$, by picking an attribute $j \in [1,d]$ and choosing a threshold value v such that $\mathcal{S}_{L^{\leq}} = \{(\boldsymbol{x}, y) \in \mathcal{S}_L \mid x_j \leq v\}$ and $\mathcal{S}_{L^{>}} = \mathcal{S}_L \setminus \mathcal{S}_{L^{\leq}}$. The goal is to select v such that it splits samples with different y. A common criterion for selecting j and v is to minimize the *Gini impurity* $Gini_y(\mathcal{S}) = 2 \cdot p_y(1-p_y) \in [0, \frac{1}{2}]$ where $p_y = \sum_{(\boldsymbol{x},y') \in \mathcal{S}} \mathbb{1}_{y'=y} / |\mathcal{S}|$ denotes the relative frequency of class y in $\mathcal{S}$. We build T by repeating this procedure until we reach a predefined maximum number of leaf nodes k^*. At inference, we propagate a point $\boldsymbol{x}'$ through T, reaching a leaf node L', and returning the majority class of $\mathcal{S}_{L'}$.

Categorical Splits. As splits in a decision tree generally are of the form $x_j \leq v$, this limits the tree's ability to partition one-hot-encoded categorical attributes effectively. Typical implementations of decision trees can only single out one category per split ($x_j \leq 0.5$). We avoid this issue in PAT by using the *fairness-aware categorical splits* introduced in [13]. Namely, we represent categorical attributes not via one-hot encoded vectors but instead each category for an attribute x_j with $|C|$ classes with a unique index in $\{1, \ldots, |C|\}$. We explore multiple ways to sort the indices, and for each sorting, consider all possible prefix-postfix splits.

PGini *Criterion.* The next modification compared to standard classification trees is in the criterion, usually focused solely on utility. We aim to include a privacy-aware component, where ideally one can explicitly control the utility-privacy tradeoff. To this end, we extend prior work in fair trees [13], [61] to propose *PGini*, a privacy-aware criterion that accounts for the distribution of multiple personal attributes.

Let P denote our set of personal attributes, and $\mathcal{S}_D$ be a concrete dataset. We modify the multi-class Gini impurity, defined on an attribute a with V possible values as

$$Gini_a(\mathcal{S}_D) = \sum_{v=0}^{V-1} p_{a,v} \cdot (1 - p_{a,v}) \in [0, 1 - \frac{1}{V}]$$

where $p_{a,v} = \sum_{(\boldsymbol{x},y) \in \mathcal{S}_D} \mathbb{1}\{\boldsymbol{x}_a = v\} / |\mathcal{S}_D|$, and define:

$$\begin{aligned} PGini(\mathcal{S}_D) =& (1-\alpha) \cdot 2 \cdot Gini_y(\mathcal{S}_D) + \\ & \alpha \left(1 - \frac{1}{|P|} \sum_{p \in P} \sigma_p \cdot Gini_p(\mathcal{S}_D) \right) \end{aligned} \tag{2}$$

where $\sigma_p = \frac{|p|}{|p|-1}$ and 2 normalize the Gini values to $[0,1]$.

The utility term in Equation (2), $Gini_y(\mathcal{S}_D)$, is minimized as in the usual usage of Gini impurity. Intuitively, this promotes partitions of the input space where each region is still predictive of the target label. In contrast, the privacy term $Gini_p(\mathcal{S}_D)$ is *maximized*, promoting partitions where each region is *non-predictive* of personal attributes. The parameter $\alpha \in [0,1]$ allows for a smooth tradeoff between these two goals, where larger α implies more focus on privacy. This separates our approach from prior tree-based vDM approaches, such as [19], which learn a decision tree without accounting for personal attributes and achieve the targeted utility-privacy tradeoff via pruning. Further, $PGini$ can be easily adapted to weigh personal attributes individually in case some are deemed more sensitive than others.

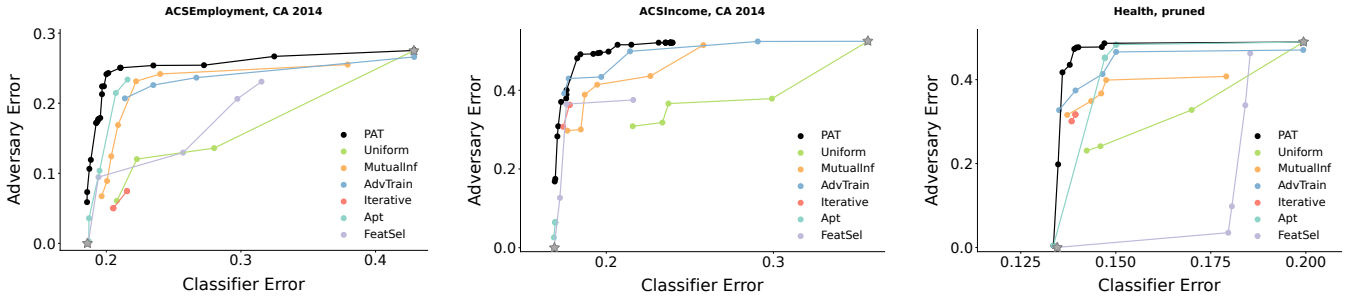


Figure 4: Utility-privacy tradeoffs of candidate generalizations produced by minimizers on ACSEmployment, ACSIncome, and Health (pruned). Classifier and adversarial errors are reported on a held-out test set, while the candidate generalizations are selected on the validation set. Across all datasets, PAT generally achieves the most favorable utility-privacy tradeoff.

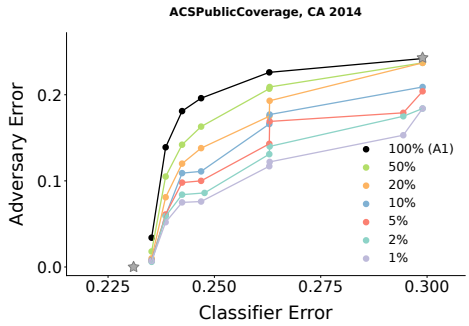


Figure 5: A2 reconstruction error for several generalizations with *AdvTrain* on the ACSPublicCoverage test set.

Obtaining a Generalization. While the leaves of T define a partition of the input space $\mathcal{X}$, the splits in each node V of T depend on prior splits in V's ancestors $\mathbb{A}(V)$. As the splits in $\mathbb{A}(V)$ might be on different attributes than the split in V, the partition defined by T is not by itself a strict generalization as defined in Section 4. Hence, to construct a generalization g, PAT post-processes T, partitioning each attribute into ranges by taking the union of all split thresholds for that attribute encountered in T. Notably, while g partitions $\mathcal{X}$ into strictly more granular subsets than T, it holds for all $\boldsymbol{x} \in \mathcal{X}$ that $T(\boldsymbol{x}) = T(g(\boldsymbol{x}))$, i.e., g fully preserves the utility of T.

8. Experimental Evaluation

In this section, we present an extensive experimental evaluation of our vDM setting utilizing all adversaries from Section 5 and minimizers introduced in Sections 6-7.

In Section 8.1 we present our main results, followed by a study of individual attribute reconstruction in Section 8.2, minimizer training sizes in Section 8.3 and a qualitative study in Section 8.4. In our supplementary material, we provide additional details on adversaries (Appendix A and Appendix G) and experimental parameters (Appendix F).

8.1. Main Results

Generalizations g are learned on a fixed training set S_{orig}, which we here assume to be observed by the adversary ($S'_{\text{orig}} \equiv S_{\text{orig}}$). We set $S_{\text{min}} = g(S_{\text{orig}})$, and split the data into three disjoint parts: training, validation and test. We train a classifier f using training and validation splits, reporting the accuracy on the test split. Finally, we set S'_{min} to the generalized test split, assuming it is breached.

(A1) Reconstruction. The A1 adversary tries to reconstruct the personal attributes of the generalized records from S'_{min}. We run each minimizer with different parameters to produce a diverse set of generalizations and report the utility-privacy tradeoff on two datasets from the ACS suite [62], derived from US census, predicting individuals' employment and income, respectively. Additionally, we evaluate on a preprocessed Health [63] dataset, predicting the Charlson Comorbidity Index. We provide more details on our datasets and their preprocessing in Appendix D.

Figure 4 shows the classifier error $\mathcal{UR}(g)$ and mean adversary error $\mathcal{PR}_{A1}(g)$ on test data for each type of minimizer. We plot all points, marking those that remain on the Pareto front when using the test set. The ★ markers represent the two *limits of generalization*: (i) fully-generalized data (all $k_i = 1$), where both classifier and the adversary predict solely based on the class frequency of each personal attribute, and (ii) non-generalized data (all $k_i = c_i$), giving a lower bound on classification error with trivial adversary error of 0. We observe that PAT consistently achieves the most favorable utility-privacy tradeoffs across all datasets. Additionally, our new baselines provide a wide range of generalizations and can serve as benchmarks for future work. In Appendix E we show similar results on several additional datasets, and in Appendix H further demonstrate the robustness of PAT to temporal distribution shift.

Across all experiments, most minimizers show clear inflection points in the utility-privacy tradeoff. Taking PAT on ACSEmployment as an example, we find that there is almost no decrease in adversarial error until the classifier error reaches around 0.2. Any decrease in classifier error after this comes with a significantly larger decrease in adversarial error. In addition to having pre-defined utility targets (for example, from requirements), these inflection points can assist practitioners in identifying generalizations that yield favorable utility-privacy tradeoffs.

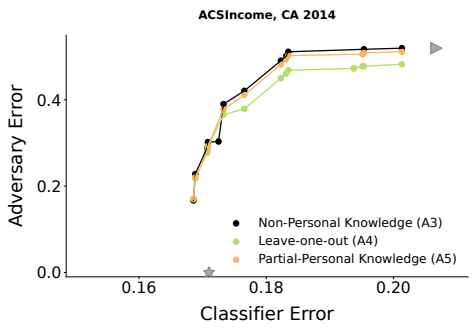


Figure 6: Pareto fronts of A3-A5 adversaries using PAT. In the low error regime, adversaries perform equally, as reconstruction is easier. In higher error regimes, A4 outperforms A3 and A5.

(A2) High-Certainty Reconstruction. A2 operates in the same setting as A1, i.e., tries to reconstruct samples from S'_{min}. Let $h(z)_a$ denote the logits that an A1 adversary predicts for attribute a. Further, let $\mathbb{h}(z)_a = \max(h(z)_a)$ denote the maximum of the logits. We say that an A2 adversary is $k\%$-confident in its prediction for a if $\mathbb{h}(z)_a$ is in the k-th highest percentile of $\mathbb{h}(\mathcal{Z}, a) = \{\mathbb{h}(z)_a \mid z \in \mathcal{Z}\}$, measuring confidence relative to other predictions for a.

Figure 5 shows the results of A2 on the ACSPublicCoverage dataset predicting individuals' insurance coverage. We fix the minimizer to *AdvTrain* (with similar results for other minimizers), and for each point in the Pareto front of the A1 adversary ($k = 100\%$), we report the mean reconstruction error for several A2 adversaries with different k. We observe a significant decrease in the error rate for more confident adversaries, showing that A2 adversaries can recover specific individuals more accurately and relying solely on A1 underestimates the privacy risk.

(A3-A5) Reconstruction with Side Information. Using the PAT minimizer, we evaluate A3-A5 via their Pareto fronts in the utility-privacy tradeoff on the ACSIncome dataset (Figure 6). As predicted in Section 5, the *Partial Personal Knowledge* (A5) adversary lies between the *Non-Personal Knowledge* (A3) and *Leave-one-out* (A4) adversary. Looking closer at A4, we find that even though all but a single attribute have been leaked in full granularity, having this attribute generalized still hinders its reconstruction noticeably. This indicates that vDM offers valuable privacy protection even under heavy side information. Further, we find that the difference between the adversaries decreases in lower error regions. This is not unexpected, as the generalizations achieving lower classifier error are increasingly fine-grained.

(A6) Multi-breach Reconstruction. Table 3 reports the mean reconstruction error for several (C1-C4) multi-breach scenarios, each consisting of two breaches under different generalizations. To this end, we implement an A6 adversary by training two A1 adversaries and averaging their predictions. We find that the A6 adversary has a lower error than any A1 adversary on their respective individual breaches. This shows that minimizing the same data multiple times can increase the privacy risk warranting extra care.

TABLE 3: Mean reconstruction error of A6 adversary in four MBR scenarios, using the *Uniform* minimizer and PAT.

	C1		C2		C3		C4	
g	PAT α=0.1	PAT α=0.7	PAT α=0.3	PAT α=0.7	Unif. k=3	PAT α=0.7	Unif. k=4	PAT α=0.7
$\mathcal{PR}_{A1}$	0.43	0.48	0.47	0.48	0.37	0.48	0.32	0.48
$\mathcal{PR}_{A6}$	**0.42**		**0.45**		**0.33**		**0.27**	

TABLE 4: Percentage of entries correctly linked by A7 on the OCCP and POBP attributes (for different generalizations produced by PAT, *Uniform*, and *AdvTrain* minimizers). *Rand.* denotes a baseline method that does not utilize S'_{min}.

g	Rand.	PAT k^*=8 α=0.3	PAT k^*=20 α=0.3	PAT k^*=50 α=0.3	PAT k^*=50 α=0.9	Unif. k=3	Adv. α=0
#Buckets	–	21	43	348	200	30	33
% Linked	0.24	0.25	0.30	**0.46**	0.27	0.36	0.33

(A7) Linkability. We evaluate our A7 adversary on the ACSIncome dataset, picking two attributes, `OCCP` (`Occupation`) and `POBP` (`Place-of-Birth`), that we want to link. The A7 adversary can now use the minimized data S'_{min} to approximate the distribution $p(\boldsymbol{x}_{OCCP} \mid \boldsymbol{x}_{POBP} = j)$ (see Appendix A) and predict the value for $\boldsymbol{x}_{OCCP}$. Table 4 shows how the probability of correctly linking entries significantly increases with access to higher granularity records.

(A8) Singling Out. Unlike the synthetic data setting [51], [64], in vDM, each $\boldsymbol{x} \sim D_{orig}$ is directly mapped to exactly one $\boldsymbol{z} = g(\boldsymbol{x})$. This makes it considerably easier for an A8 adversary to find a predicate Π, which singles out an individual from S'_{min}. Intuitively, an adversary with access to S'_{min} (and multiplicities of entries) can target $\boldsymbol{z}$ which only rarely occur. In particular, if there exists a $\boldsymbol{z} = g(\boldsymbol{x})$ which is only observed once, the adversary can single out the corresponding $\boldsymbol{x}$ out by defining $\Pi_{\boldsymbol{x}}$ based on the attribute ranges implied by $\boldsymbol{z}$ (more detail in Appendix A).

In Table 5, we report how many $\boldsymbol{x}$ get generalized to the *rarest* $\boldsymbol{z}$ in ACSEmployment (reported as *Utilization*). When multiple $\boldsymbol{z}$ are the rarest, we report their number in parentheses. For example, for PAT with $k^* = 20$ (max. # of leaves) and $\alpha = 0.3$, we find 44 different $\boldsymbol{z}$, with each having only one $\boldsymbol{x}$ getting generalized to it. As expected, more granular generalizations significantly increase the adversary's chance of finding low utilization $\boldsymbol{z}$ to single out.

Summary. Our experiments have shown that the adversaries defined in Section 5 capture a diverse set of attacks on different aspects of vDM, establishing them as a comprehensive way to evaluate generalizations. We have further demonstrated that PAT consistently outperforms new and prior baseline minimizers across different settings.

8.2. Individual Attribute Reconstruction

The results for reconstruction (A1-A6) so far only report the mean over all personal attributes. We now investigate

TABLE 5: Utilization of the least common z in multiple generalizations, for 223,531 ACSEmployment records.

g	PAT k^*=2 α=0.3	PAT k^*=4 α=0.3	PAT k^*=10 α=0.3	PAT k^*=20 α=0.3	PAT k^*=20 α=0.0	PAT k^*=20 α=0.8
#Buckets	17	19	25	33	39	33
Utilization (#)	52k (1)	1.9k (1)	1 (1)	1 (44)	1 (120)	1 (10)

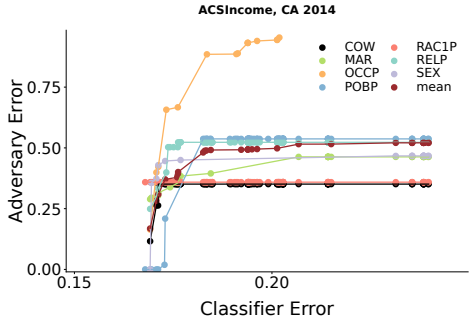


Figure 7: Individual attribute reconstruction of the A1 adversary on ACSIncome, for points from the A1 PAT Pareto front. For many attributes the adversary can only improve over the naive baseline in very low error regimes.

how well an adversary can reconstruct *individual* attributes.

Results. In particular, for A1, we report in Figure 7 the individual attribute reconstruction error for all points on the PAT Pareto front for ACSIncome (we show another example in Appendix G). This more detailed view provides a better insight into A1's capabilities, allowing us to read out the graphs corresponding to max (outer, left Pareto front) or min (inner, right Pareto front) aggregators.

Overall, we observe how the adversary can only accurately reconstruct many personal attributes in low classification error regimes. For higher classification errors, the reported mean adversarial accuracy is dominated by a few attributes, which have many classes (`OCCP` has 477 classes).

For practitioners, this can be especially interesting when their goal is to protect only specific attributes. For example, a practitioner who focuses on the place of birth (`POBP`) can choose the generalization with a classifier error of ≈ 0.18, before the `POBP` adversarial error decreases rapidly.

8.3. Minimizer Training Sizes

As mentioned in Section 4, we further observe that a small S_{orig} of full-granularity data is sufficient for training a minimizer. Taking, e.g., PAT on ACSEmployment in Figure 8, we find that using only 3% of the total data (5% of the training data) yields almost identical minimizer results, with performance only deteriorating for very small S_{orig}.

From a practitioner's point of view, this gives the advantage of only requiring small (well-protected) amounts of data to train g before being able to collect minimized samples. In order to evaluate a larger set of generalizers, one might, however, increase S_{orig} to sizes commonly used for other PETs. Our framework then provides sensible ranges for all possible minimizer parameters, automatically tuning all adversarial hyperparameters.

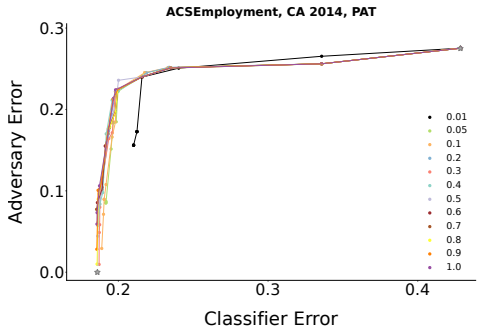


Figure 8: Privacy-Utility Tradeoff for PAT on ACSEmployment with varying size of S_{orig} (as fractions of the training set). We observe worse tradeoffs only for very small S_{orig}.

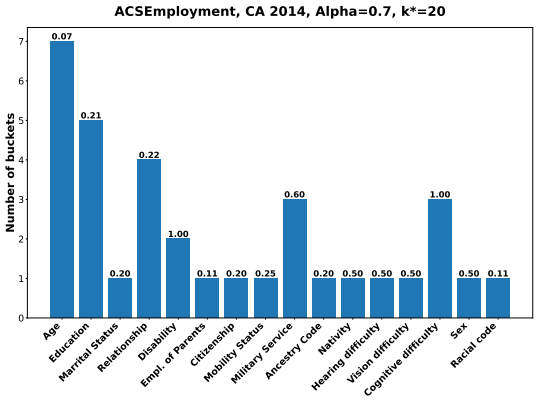


Figure 9: Number of buckets for each attribute in a PAT generalization of ACSEmployment with $\alpha = 0.7, k^* = 20$. Above each bar, we denote the relative size of the attribute w.r.t. its original size, e.g., `age` has been reduced by 93%.

8.4. Qualitative Study

We end our experimental evaluation with a qualitative study on one generalization. Namely, we run PAT on ACSEmployment with $k^* = 20$ and $\alpha = 0.7$, the same parameters as we used for our example in Figure 1.

ACSEmployment has 16 attributes with a total of 196 attribute values (assuming `age` has integer range $[0 - 99]$), which are minimized to just 34 buckets (across all attributes). This increases the classifier error by only 0.01 while increasing the adversarial error from 0 to 0.23. A naive adversary which would only predict the majority value (over $\mathcal{S}'_{orig}$) for each attribute, constituting an upper bound, would have an adversarial error of 0.275. Looking at individual attributes, e.g., `SEX`, we find that we increase the adversarial error from 0 to a nearly random 0.48.

Per-attribute Generalizations. In Figure 9, we explore the values of k_i for each attribute. We find that (1) the attributes that one would generally relate with the employment status (e.g., `Age`, `Educational Status`) are less generalized than

other attributes (e.g., `Sex`). 10 out of 16 attributes (including highly sensitive ones as `Ancestry` and `Race`) can be fully generalized and hence would not even need to be collected. This suggests that client privacy could be greatly improved by deploying vDM in real-world scenarios.

Looking closer at some attributes we find that PAT groups all persons with `age` 20-63 into a single bucket, while the generalization of the `Educational Status` attribute can be roughly summarized as [No high school diploma, Highschool diploma, Assoc. Degree, Masters, Ph.D.] instead of 24 individual categories.

9. Future Work

We believe that due to an increase in regulation, data minimization will be a highly relevant topic for future research. We see four key areas for future work on vDM.

Combining vDM and Other ML Privacy Topics. This includes combining vDM with topics such as differential privacy [65] and secure computation [66], [67], [68]. As vDM stands orthogonal to many of these techniques, we believe their combination can be particularly interesting. Some of our early investigations already indicate that minimized data might be well-suited for common DP training procedures. An additional area of interest is the interplay between vertical and horizontal data minimization.

Other Domains. With most personal attributes (e.g., religion, age, political affiliation) in tabular form, the application of generalizations for privacy protection has so far been focused on tabular data. Adapting such methods to, e.g., images or text is a promising avenue for future work.

Privacy Guarantees. Further enhancing our privacy risk assessment with formal guarantees on the utility-fairness tradeoffs is another important direction that would benefit the practitioners. This is a challenging problem, and it is unclear what kind of approach would be able to provide such guarantees. One potentially promising direction could be to enforce a distribution-wide lower bound on the utilization of each generalization bucket, and use this to bound the adversarial risk following the approach of [13] that offer similar guarantees in the setting of fair representation learning.

New vDM Algorithms. While PAT outperforms all baselines across a variety of settings, it does not come with an optimality guarantee. This leaves the field of vDM open for future algorithms with better utility-privacy tradeoffs.

10. Related Work

10.1. Regulations and Policy

The requirement of data minimization, which limits the collection and processing of personal data to the minimum necessary for a specific purpose, was introduced in 2016 in the E.U.s GDPR [3]. Similar principles have been adapted and integrated into other regulations, such as CPRA [4] and the recent Blueprint for a U.S. AI Bill of Rights [5], emphasizing its relevance. While data protection authorities in certain countries (e.g., the U.K.'s ICO [69] and the Norwegian Data Protection Authority [9]) have proposed some guidelines for complying with DM in ML contexts, and there exists policy literature analyzing similar issues [70], to date, no concrete evaluation tools have been proposed.

10.2. Generalizations in ML

Besides the already mentioned PPDP use-case, the idea of generalizing attributes to coarser representations has been applied in several ML settings under different names: *Discretization* is used to combine similar attribute values in the context of recommender systems [71], [72]. *Binning* is used in, e.g., [73], [74] to derive concrete attribute values out of histogram data. It is worth noting that suppression and feature selection are special cases of generalization.

10.3. Operationalization of DM for ML

We now discuss prior attempts to operationalize DM for ML. [75] discusses a *need-to-know* principle which is similar to DM, but focuses primarily on fairness and does not touch on the generalization of attributes beyond attribute selection. [76] explicitly considers a formalization of DM tailored to recommender systems with techniques that are not applicable to our vDM for ML setting. [77] proposes algorithms for black-box auditing of DM compliance based on model instability. This setting differs from ours as it focuses on auditing an already trained model. [10] focuses on minimizing training data size (*hDM*), an important concern even outside DM [11], [12] that stands orthogonal to *vDM*.

The most closely related work to ours is [19], which applies concepts from data anonymization (discussed in Section 3) to propose a vDM minimizer. However, as shown in Section 2.2, the metrics from data anonymization [18] do not translate well into the vDM setting. Additionally, PAT outperforms the [19] approach both in terms of speed (100x) and in the utility-privacy tradeoff. Crucially, no prior work defines comprehensive evaluation procedures, baselines, and ways to evaluate empirical privacy risks under different adversaries, which we argue is a key issue for vDM.

10.4. Fair Representation Learning

Fair representation learning (FRL) [25], [59], [78], [79] transforms data into a new representation useful for downstream tasks while ensuring that one cannot recover the sensitive attribute. The most common approaches are typically based on adversarial training [59], VAE [80], mutual information [79], [81] and normalizing flows [25]. While FRL partly inspired our approach for PAT in Section 7, FRL and DM consider different scenarios and have two key technical differences: (i) In FRL, it is always necessary to collect full-granularity data to produce the representation, as the transformation is non-interpretable (i.e., cannot be easily applied by a client), and (ii) similar to work in data anonymization (see Section 2.3) FRL usually considers a single sensitive attribute, while we relax this constraint.

10.5. Causal Feature Selection

Another line of work studies causal feature selection [82], [83], [84], [85]. Some works in particular investigate directions related to vDM such as connections to privacy [84] or the design of FRL-inspired methods that offer interpretability [85]. While suitable for certain scenarios, feature selection methods only have a binary choice for each feature. This severely limits their available transformations in comparison to tailor-made minimizers in the vDM setting, as we demonstrate in Section 8 and Appendix E.

11. Conclusion

We have addressed the challenge of formalizing and achieving vertical data minimization, a highly relevant privacy requirement, in the context of ML. We formalized the vDM setting and workflow via generalizations, defined two key requirements of utility and empirical privacy risk, and proposed a set of diverse adversaries as tools for empirical privacy risk evaluation. We introduced several baseline vDM minimizers, as well as the *Privacy-aware Tree* (PAT) minimizer, whose effectiveness we demonstrated on several real-world datasets. Our hope is that our work and public release of the vDM toolbox will enable organizations to more effectively enforce and evaluate data minimization, ultimately helping reduce the privacy risks for individuals.

Acknowledgments

We thank Matthew Jagielski for helpful discussions in early stages of the project. We are grateful to anonymous reviewers for their constructive feedback, and the S&P organizing committee and our shepherd for facilitating a quality review process. This work has received funding from the Swiss State Secretariat for Education, Research and Innovation (SERI) (SERI-funded ERC Consolidator Grant).

References

[1] A. E. Khandani, A. J. Kim, and A. W. Lo, "Consumer credit-risk models via machine-learning algorithms," *JBF*, 2010.

[2] J. Perols, "Financial statement fraud detection: An analysis of statistical and machine learning algorithms," *Auditing: A Journal of Practice & Theory*, 2011.

[3] E. U. EU, "General data protection regulation," 2016. [Online]. Available: https://gdpr-info.eu/

[4] C. S. o. S. CAGOV, "California privacy rights act," 2020. [Online]. Available: https://vig.cdn.sos.ca.gov/2020/general/pdf/topl-prop24.pdf

[5] USGOV, "Blueprint for an AI Bill of Rights | OSTP," 2022. [Online]. Available: https://www.whitehouse.gov/ostp/ai-bill-of-rights/

[6] CMS, "Gdpr enforcement tracker," 2018. [Online]. Available: https://www.enforcementtracker.com/

[7] GDPRhub, "Ap (the netherlands) - tax administration fined for discriminatory and unlawful data processing," 2023, https://gdprhub.eu/index.php?title=AP_(The_Netherlands)_-_Tax_Administration_fined_for_discriminatory_and_unlawful_data_processing.

[8] CMS, "Gdpr enforcement tracker: Etid-790," 2021. [Online]. Available: https://www.enforcementtracker.com/ETid-790

[9] N. D. P. A. DPA, "Artificial intelligence and privacy," 2018. [Online]. Available: https://www.datatilsynet.no/globalassets/global/english/ai-and-privacy.pdf

[10] D. Shanmugam, S. Shabanian, F. Diaz, M. Finck, and A. Biega, "Learning to limit data collection via scaling laws: Data minimization compliance in practice," *ACM FAccT*, 2022.

[11] B. Mirzasoleiman, J. A. Bilmes, and J. Leskovec, "Coresets for data-efficient training of machine learning models," in *ICML*, 2020.

[12] M. Paul, S. Ganguli, and G. K. Dziugaite, "Deep learning on a data diet: Finding important examples early in training," 2021.

[13] N. Jovanović, M. Balunović, D. I. Dimitrov, and M. Vechev, "Fare: Provably fair representation learning with practical certificates," *ICML*, 2023.

[14] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," *JFR*, 2020.

[15] H. Chen, O. Engkvist, Y. Wang, M. Olivecrona, and T. Blaschke, "The rise of deep learning in drug discovery," *Drug discovery today*, 2018.

[16] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Mondrian multidimensional k-anonymity," in *IEEE ICDE*, 2006.

[17] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A. W.-C. Fu, "Utility-based anonymization using local recoding," in *ACM SIGKDD*, 2006.

[18] G. Ghinita, P. Karras, P. Kalnis, and N. Mamoulis, "Fast data anonymization with low information loss," in *VLDB*, 2007.

[19] A. Goldsteen, G. Ezov, R. Shmelkin, M. Moffie, and A. Farkash, "Data minimization for GDPR compliance in machine learning models," *AI and Ethics*, 2022.

[20] A. . D. P. W. Party, "Opinion 05/2014 on anonymisation techniques," 2014. [Online]. Available: https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2014/wp216_en.pdf

[21] U. D. of Labor, "Guidance on the protection of personal identifiable information." [Online]. Available: https://www.dol.gov/general/ppii

[22] B. C. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing: A survey of recent developments," *ACM Computing Surveys*, 2010.

[23] J. Han, F. Luo, J. Lu, and H. Peng, "Sloms: A privacy preserving data publishing method for multiple sensitive attributes microdata." *Journal of Software*, 2013.

[24] A. Anjum, N. Ahmad, S. U. Malik, S. Zubair, and B. Shahzad, "An efficient approach for publishing microdata for multiple sensitive attributes," *The Journal of Supercomputing*, 2018.

[25] M. Balunovic, A. Ruoss, and M. Vechev, "Fair normalizing flows," in *ICLR*, 2022.

[26] "Prohibited Employment Policies/Practices." [Online]. Available: https://www.eeoc.gov/prohibited-employment-policiespractices

[27] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *IEEE S&P*, 2017.

[28] A. Salem, G. Cherubin, D. Evans, B. Kopf, A. Paverd, A. Suri, S. Tople, and S. Zanella-Beguelin, "Sok: Let the privacy games begin! a unified treatment of data inference privacy in machine learning," in *IEEE S&P*, 2023.

[29] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, 2020.

[30] M. Balunović, D. I. Dimitrov, R. Staab, and M. Vechev, "Bayesian framework for gradient leakage," *ICLR*, 2022.

[31] C. Dwork, "Differential privacy," in *ICALP*, 2006.

[32] Q. Ye and H. Hu, "Local Differential Privacy: Tools, Challenges, and Opportunities," in *Web Information Systems Engineering*, 2020.

[33] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *ACM CCS*, 2016.

[34] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, 2009.

[35] I. Chillotti, M. Joye, and P. Paillier, "New Challenges for Fully Homomorphic Encryption."

[36] N. J. Hernandez Marcano, M. Moller, S. Hansen, and R. H. Jacobsen, "On Fully Homomorphic Encryption for Privacy-Preserving Deep Learning," in *IEEE Globecom Workshops*, 2019.

[37] J.-W. Lee, H. Kang, Y. Lee, W. Choi, J. Eom, M. Deryabin, E. Lee, J. Lee, D. Yoo, Y.-S. Kim, and J.-S. No, "Privacy-Preserving Machine Learning With Fully Homomorphic Encryption for Deep Neural Network," 2022.

[38] R. Cramer, I. B. Damgård *et al.*, *Secure multiparty computation*. Cambridge University Press, 2015.

[39] B. Knott, S. Venkataraman, A. Hannun, S. Sengupta, M. Ibrahim, and L. van der Maaten, "Crypten: Secure multi-party computation meets machine learning," *NeurIPS*, 2021.

[40] C. Juvekar, V. Vaikuntanathan, and A. Chandrakasan, "Gazelle: A low latency framework for secure neural network inference," in *USENIX Security*, 2018.

[41] P. Mishra, R. Lehmkuhl, A. Srinivasan, W. Zheng, and R. A. Popa, "Delphi: A cryptographic inference system for neural networks," in *Workshop on Privacy-Preserving Machine Learning in Practice*, 2020.

[42] L. Sweeney, "k-anonymity: A model for protecting privacy," 2002.

[43] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity," in *IEEE ICDE*, 2006.

[44] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity," in *IEEE ICDE*, 2006.

[45] X. He, Y. Xiao, Y. Li, Q. Wang, W. Wang, and B. Shi, "Permutation anonymization: Improving anatomy for privacy preservation in data publication," in *New Frontiers in Applied Data Mining: PAKDD International Workshops*, 2012.

[46] L. Xu and K. Veeramachaneni, "Synthesizing tabular data using generative adversarial networks," *arXiv preprint arXiv:1811.11264*, 2018.

[47] T. Stadler, B. Oprisanu, and C. Troncoso, "Synthetic data–anonymisation groundhog day," in *USENIX Security*, 2022.

[48] Y. Chen, F. Luo, T. Li, T. Xiang, Z. Liu, and J. Li, "A training-integrity privacy-preserving federated learning scheme with trusted execution environment," *Information Sciences*, vol. 522, 2020.

[49] Decentriq, "Future-proof Data Clean Rooms | Decentriq," 2023. [Online]. Available: https://decentriq.com/

[50] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *ACM CCS*, 2015.

[51] M. Giomi, F. Boenisch, C. Wehmeyer, and B. Tasnádi, "A unified framework for quantifying privacy risk in synthetic data," 2022. [Online]. Available: https://arxiv.org/abs/2211.10459

[52] I. E. Olatunji, J. Rauch, M. Katzensteiner, and M. Khosla, "A review of anonymization for healthcare data," *CoRR*, 2021.

[53] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *IEEE S&P*, 2008.

[54] F. Boenisch, R. Munz, M. Tiepelt, S. Hanisch, C. Kuhn, and P. Francis, "Side-channel attacks on query-based data anonymization," in *ACM CCS*, 2021.

[55] A. Cohen and K. Nissim, "Towards formalizing the gdpr's notion of singling out," *Proc. Natl. Acad. Sci. USA*, 2020.

[56] R. Dingledine and N. Mathewson, "Anonymity loves company: Usability and the network effect." in *WEIS*, 2006.

[57] J. Sill, "Monotonic networks," in *NIPS*, 1997.

[58] X. Liu, X. Han, N. Zhang, and Q. Liu, "Certified monotonic neural networks," in *NeurIPS*, 2020.

[59] D. Madras, E. Creager, T. Pitassi, and R. S. Zemel, "Learning adversarially fair and transferable representations," in *ICML*, 2018.

[60] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *NeurIPS*, 2014.

[61] W. Zhang and E. Ntoutsi, "FAHT: An Adaptive Fairness-aware Decision Tree Classifier," in *IJCAI*, 2019.

[62] F. Ding, M. Hardt, J. Miller, and L. Schmidt, "Retiring adult: New datasets for fair machine learning," in *NeurIPS*, 2021.

[63] Kaggle, "Heritage Health Prize," 2012. [Online]. Available: https://kaggle.com/competitions/hhp

[64] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling Tabular data using Conditional GAN," Oct. 2019, arXiv:1907.00503 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1907.00503

[65] M. Abadi, A. Chu, I. J. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *ACM CCS*, 2016.

[66] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy," in *ICML*, 2016.

[67] J. Liu, M. Juuti, Y. Lu, and N. Asokan, "Oblivious neural network predictions via minionn transformations," in *ACM CCS*, 2017.

[68] R. Dathathri, O. Saarikivi, H. Chen, K. Laine, K. Lauter, S. Maleki, M. Musuvathi, and T. Mytkowicz, "CHET: An Optimizing Compiler for Fully-Homomorphic Neural-Network Inferencing," in *ACM PLDI*, 2019.

[69] U. I. C. O. UKICO, "Guide to the general data protection regulations (gdpr)," 2018. [Online]. Available: https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection\-regulation-gdpr/

[70] M. Finck and A. J. Biega, "Reviving purpose limitation and data minimisation in data-driven systems," *Technology and Regulation*, 2021.

[71] H. Guo, B. Chen, R. Tang, W. Zhang, Z. Li, and X. He, "An embedding learning framework for numerical features in CTR prediction," in *KDD*, 2021.

[72] Y. Qu, B. Fang, W. Zhang, R. Tang, M. Niu, H. Guo, Y. Yu, and X. He, "Product-based neural networks for user response prediction over multi-field categorical data," *ACM Trans. Inf. Syst.*, 2019.

[73] U. M. Fayyad and K. B. Irani, "Multi-interval discretization of continuous-valued attributes for classification learning," in *IJCAI*, 1993.

[74] G. Navas-Palencia, "Optimal binning: mathematical programming formulation," 2020.

[75] B. Rastegarpanah, M. Crovella, and K. P. Gummadi, "Fair inputs and fair outputs: The incompatibility of fairness in privacy and accuracy," in *UMAP*, 2020.

[76] A. J. Biega, P. Potash, H. D. III, F. Diaz, and M. Finck, "Operationalizing the legal principle of data minimization for personalization," in *ACM SIGIR*, 2020.

[77] B. Rastegarpanah, K. P. Gummadi, and M. Crovella, "Auditing black-box prediction models for data minimization compliance," in *NeurIPS*, 2021.

[78] R. S. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork, "Learning fair representations," in *ICML*, 2013.

[79] U. Gupta, A. Ferber, B. Dilkina, and G. V. Steeg, "Controllable guarantees for fair outcomes via contrastive information estimation," *AAAI*, 2021.

[80] C. Louizos, K. Swersky, Y. Li, M. Welling, and R. S. Zemel, "The variational fair autoencoder," in *ICLR*, 2016.

[81] J. Song, P. Kalluri, A. Grover, S. Zhao, and S. Ermon, "Learning controllable fair representations," in *AISTATS*, 2019.

[82] C. F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, and X. D. Koutsoukos, "Local causal and markov blanket induction for causal discovery and feature selection for classification part i: Algorithms and empirical evaluation," *JMLR*, 2010.

[83] K. Yu, L. Liu, J. Li, W. Ding, and T. D. Le, "Multi-source causal feature selection," *IEEE TPAMI*, 2020.

[84] S. Tople, A. Sharma, and A. Nori, "Alleviating privacy attacks via causal learning," in *ICML*, 2020.

[85] R. Hasan and M. Fritz, "Understanding utility and privacy of demographic data in education technology by causal analysis and adversarial-censoring," *PoPETs*, vol. 2022.

[86] "Lending club loan data." [Online]. Available: https://www.kaggle.com/datasets/wordsforthewise/lending-club

[87] M. Kelly, R. Longjohn, and K. Nottingham, "The UCI Machine Learning Repository." [Online]. Available: https://archive.ics.uci.edu

Appendix A. More Details on Adversaries

Masking in A1-A6 (Reconstruction). For any reconstruction adversary, let $h(\boldsymbol{z})_p$ denote the logits produced when reconstructing attribute p from $\boldsymbol{z}$. As h is implemented as a neural network, the co-domain (range) of h is the cardinality (number of classes) of p. However, as we assume that the adversary has access to the generalization function g, it can use this information to mask out all logits in $h(\boldsymbol{z})_p$ that correspond to classes in $\boldsymbol{x}_p$ that g could not have mapped to $\boldsymbol{z}_p$. This can significantly limit the co-domain for specific reconstructions, resulting in a stronger adversary. We apply the same masking procedure for all reconstruction adversaries. In particular, A6 intersects the masks corresponding to the different generalizations.

Details on A7 (Linkability). Here we provide further detail on how we approximate $p(\boldsymbol{x}_A \mid \boldsymbol{x}_B = \boldsymbol{b})$ when knowing S'_{min}. For this, let $A, B \subseteq [1, ..., d]$ be any two (not necessarily disjoint) subsets of our attributes. Let $g_A(\boldsymbol{x}) = g(\boldsymbol{x})_A$ denote the generalized attributes of $\boldsymbol{x}$ that are also in A and let $g_A(\boldsymbol{a})$ denote the evaluation of g on a vector $\boldsymbol{a}$ which only contains attributes in A (this is well-defined as g maps attributes independently of each other). We can now get an approximation of $p(\boldsymbol{x}_A \mid \boldsymbol{x}_B = \boldsymbol{b})$ by simply observing the relative sampling frequencies of $\boldsymbol{x}_A$ conditioned on $\boldsymbol{x}_B = \boldsymbol{b}$ (over S'_{min}). In particular, we find that $p(x_A = \boldsymbol{a} \mid x_B = \boldsymbol{b}) \approx \frac{|\{z \in S'_{min} | z_{A \cup B} = g_{A \cup B}(\boldsymbol{a}, \boldsymbol{b})\}|}{|\{z \in S'_{min} | z_B = g_B(\boldsymbol{b})\}|}$.

We note that this is only well-defined if $|\{z \in S'_{min} \mid z_B = g_B(\boldsymbol{b})\}| \geq 1$ which we assume to hold for our cases.

Details on A8 (Singling-out). To more formally describe the A8 adversary, we must model S'_{min} as a multiset. In particular, let X_{min} denote the set of full granularity records used to create S'_{min} ($g(X_{\text{min}}) = S'_{\text{min}}$). We now define the cardinality for any $z \in S'_{\text{min}}$ as $|z| = |\{x \in X_{\text{min}} \mid g(x) = z\}|$ and additionally $z^{\text{min}} = \arg\min_{z \in S'_{\text{min}}} |z|$. In case $z^{\text{min}} = 1$ we can single out the unique $\boldsymbol{x} \in X_{\text{min}}$ for which $g(\boldsymbol{x}) = z^{\text{min}}$ via $\Pi_{\boldsymbol{x}}$, which describes the generalized attributes of $\boldsymbol{z}^{\text{min}}$. In case $z^{\text{min}} > 1$, an adversary still can construct $\Pi_{z^{\text{min}}}$ to target z^{min}. Conditioned on $\Pi_{z^{\text{min}}}$, records $\boldsymbol{x}$ which map to z^{min} have a significantly smaller anonymity set making it easier for an adversary to single an individual.

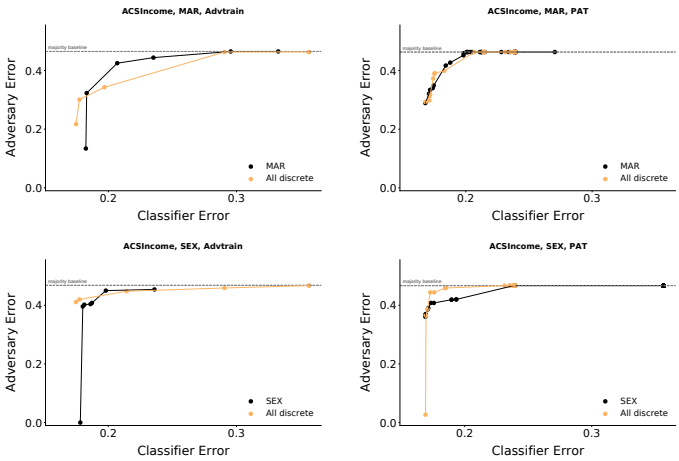


Figure 10: Reconstruction error of A1 adversary on a personal attribute (Married/SEX), when attacking generalizations from a minimizer (*AdvTrain* or *PAT*) trained to protect all discrete attributes (orange) or just the chosen attribute (black).

Appendix B. Protecting Several Personal Attributes

One question that arises when designing minimizers is if protecting a large set of personal attributes is a good proxy for protecting a single personal attribute that may be of particular interest. We investigate this with the following experiment, whose results are shown in Figure 10. In each plot, we train a minimizer (*AdvTrain* or *PAT*) twice. One minimization is, as usual, trained to protect a larger set of personal attributes (orange line), while the other is trained to only protect a single attribute (`MAR` or `SEX` respectively). Both minimizations are then attacked by the A1 adversary, aiming to reconstruct only the single attribute. As we can see in Figure 10, the respective adversarial error curves are very close for both minimizations and target attributes. This gives a strong indication that using mean aggregation (over all personal attributes) when learning minimizers is a reasonable proxy for protecting specific personal attributes.

Appendix C. Details Omitted from Experimental Evaluation

Here we supply all details omitted from Section 8.

Detailed Descriptions of Minimizers. We further formalize the baseline minimizers *AdvTrain*, *MutualInf*, and *Iterative* (Section 6). For this we write the generalization $g_\psi : \mathcal{X} \to \mathcal{Z}$, the classifier $f_\theta : \mathcal{Z} \to \mathcal{Y}$ and the adversary $h_\phi : \mathcal{Z} \to \mathcal{X}_\mathcal{S}$.

Generalization g_ψ as a Neural Network. As shown in Section 6 the *AdvTrain* and *MutualInf* minimizers model the generalization g_ψ as a set of d independent neural networks $g_\psi^{(i)}$. We will now give more detail on how g_ψ is implemented both for discrete and continuous attributes. For this, let $\boldsymbol{x} \in \mathcal{X}$ be an input g_ψ.

For discrete attributes i, let $g_\psi^{(i)} : \mathbb{R}^{c_i} \to \mathbb{R}^{k_i}$ be a network that receives a 1-hot encoding of the attribute value $x_i \in \{1, 2, ..., c_i\}$. The output is the probability p_j of generalizing x_i to value $j \in \{1, 2, ..., k_i\}$ computed by applying the softmax with temperature τ to the unnormalized probabilities produced by $g_\psi^{(i)}$: $p_j(x_i) = \frac{\exp(g_\psi^{(i)}(x_i)_j/\tau)}{\sum_{j'=1}^{k_i} \exp(g_\psi^{(i)}(x_i)_{j'}/\tau)}$.

For continuous attributes i, we assume them to be normalized to $[0, 1]$. The network $g_\psi^{(i)} : [0, 1] \to [0, 1]$ uses $W \odot W \geq 0$ as linear layer weights, together with tanh activations and batch normalization, to ensure a monotonic mapping. Then, we divide $[0, 1]$ into k_i equally-sized intervals and let $c_j = \frac{2j-1}{2k_i}$ be the center of the j-th interval. We set the probability of generalizing x_i to attribute value j as: $p_j(x_i) = \frac{\exp(-(g_\psi^{(i)}(x_i)_j - c_j)^2/\tau)}{\sum_{j'=1}^{k_i} \exp(-(g_\psi^{(i)}(x_i)_{j'} - c_{j'})^2/\tau)}$.

During training, we progressively decrease the temperature τ. Once the training is finished, we generalize each attribute to the value with the highest probability $p_j(x_i)$, essentially corresponding to the limit when $\tau \to 0$. For *AdvTrain* and *MutualInf*, we set $k_i = k$ for all attributes while allowing the network to learn not to use some values.

***Minimization with* MutualInf.** The goal of the *MutualInf* minimizer is to reduce the mutual information between the generalized $\boldsymbol{z}$ and the original attributes $\boldsymbol{x}$, written as $I(\boldsymbol{z}, \boldsymbol{x}) = H(\boldsymbol{z}) - H(\boldsymbol{z}|\boldsymbol{x})$. Using the fact that each generalized attribute z_i is computed independently using x_i and applying Jensen's inequality, we can bound $H(z)$ via $\log P(\boldsymbol{z}) = \log \mathbb{E}_x [P(\boldsymbol{z} \mid \boldsymbol{x})] = \log \mathbb{E}_x \left[\prod_{i=1}^d P(z_i \mid x_i)\right] \geq \mathbb{E}_x \left[\sum_{i=1}^d \log P(z_i \mid x_i)\right]$ and $H(\boldsymbol{z}) = -\mathbb{E}_{\boldsymbol{z}} [\log P(\boldsymbol{z})] \leq -\mathbb{E}_{\boldsymbol{z},\boldsymbol{x}} \left[\sum_{i=1}^d \log P(z_i \mid x_i)\right]$.

Similarly, for the conditional entropy, we can write:

$$H(\boldsymbol{z}|\boldsymbol{x}) = -\mathbb{E}_{\boldsymbol{x}} \mathbb{E}_{\boldsymbol{z}|\boldsymbol{x}} \left[\sum_{i=1}^d \log P(z_i \mid x_i)\right].$$

The upper bound for $H(\boldsymbol{z})$ can be approximated by independently sampling $\boldsymbol{z}$ and $\boldsymbol{x}$, while the conditional entropy can be approximated by first sampling $\boldsymbol{x}$ and then $\boldsymbol{z}$ conditionally on $\boldsymbol{x}$. Since we are using neural networks to model the generalization function $P(z_i = j \mid x_i) = p_j(x_i)$, where $p_j(x_i)$ depends on the parameters θ of the neural networks. We can use this to jointly minimize the mutual information and the classification loss during training:

$$\min_{\theta,\psi} \mathbb{E}_{\boldsymbol{x},y} \left[(1-\lambda) L_{\text{clf}}(f_\theta(g_\psi(\boldsymbol{x})), y) + \lambda L_{\text{inf}}(g_\psi(\boldsymbol{x}), \boldsymbol{x})\right],$$

where λ is (as for *AdvTrain*) a tradeoff factor between two optimization objectives, L_{clf} denotes the classifier loss, and L_{inf} denotes the mutual information objective.

***Minimization with* Iterative.** We now give more detail on the *Iterative* minimizer which uses a heuristic procedure to generalize each attribute to a fixed number of buckets (equivalence classes), trying to improve the generalization while keeping the classification error below a threshold T.

Assume that the number of buckets k is known for each attribute. For discrete attributes, we fit a logistic regression $\langle w, x^{(oh)} \rangle + b$ predicting the target label, where $x^{(oh)}$ denotes the one-hot encoding of the training data x, and sort the array of possible values for x_i w.r.t. the matching element of w, as a proxy for the impact of a value on classification (i.e., *score*). Intuitively, we want to map attribute values with similar scores to the same bucket.

We compute this mapping using dynamic programming with the state (a, k'). We aim to decide how to assign the first a possible values for attribute x_i into k' groups such that the average of the variances of scores inside each group is as small as possible. This is done by trying all possible values $b \leq a$ as starting positions of the last group and then taking b to minimize the average variance of the solution that uses the group $[b, a]$ together with the solution used at $(b-1, k'-1)$. For continuous attributes, we directly split the range $[0, 1]$ based on k-quantiles of the training set.

We now explain how the *Iterative* minimizer determines the number of buckets k for each attribute. We first sort all attributes by an estimate of the attribute's impact on the difference between the classification error and the adversarial error. Let $\Delta_{\text{clf}}^{(i)}$ and $\Delta_{\text{adv}}^{(i)}$ respectively denote the increase in classification and adversarial error from a model which predicts using original attributes to the model which predicts without attribute i (i.e., it is fully generalized). We then sort all attributes by $\Delta_{\text{clf}}^{(i)} - \Delta_{\text{adv}}^{(i)}$ in increasing order and generalize the attributes sequentially.

We first set the number of buckets for each attribute to k (a hyperparameter). Then, we reduce the number of buckets as long as the classification error is below the threshold T and then proceed to the next attribute.

Appendix D. Dataset Details

ACS. In this work, we use several datasets and prediction tasks from the ACS suite, recently proposed by [62]. The suite is derived from the American Community Survey (ACS) data, released by the US Census Bureau. All tasks used in Section 8 are described below. Each dataset offers slices by US state and year, which we fix to California (CA) and 2014, respectively. If not specified otherwise, we set all discrete attributes (see Appendix B in [62] for a detailed list of all attributes) as personal.

ACSEmployment: The task is to predict if an adult is employed. There are 372,553 datapoints for CA in 2014 having 16 attributes (14 personal) each.

ACSIncome: The task is to predict if the yearly income of a person is above $50,000. There are 183,941 datapoints for CA in 2014 having 10 attributes (7 personal) each.

ACSPublicCoverage: The task is to predict if a low-income individual not eligible for Medicare has public health

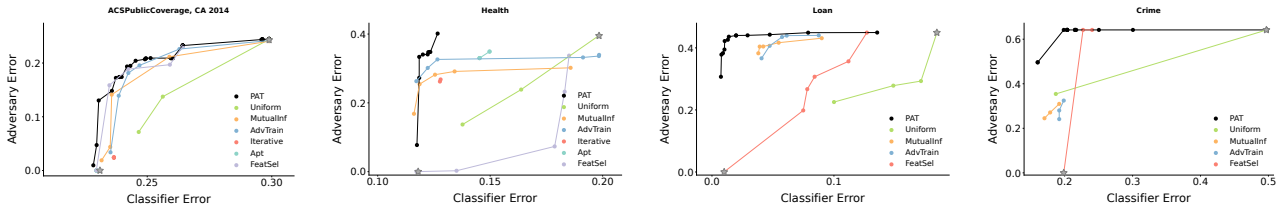


Figure 11: Utility-privacy tradeoffs of candidate generalizations produced by minimizers on ACSPublicCoverage, Health, Loan and UCI Crime. Classifier and adversarial errors are reported on a held-out test set and selected on the validation set.

insurance coverage. There are 152,676 datapoints for CA in 2014 having 19 attributes (16 personal) each.

Health. We further evaluate on the Heritage Health Dataset first proposed in [63], trying to predict the Charlson Comorbidity Index of hospital patients. The health dataset contains 218,415 datapoints. We preprocess the health dataset as described in [25], selecting two versions:

Health: Contains all 101 attributes out of which we select the following as personal: PCG=CANCRM, PCG=COPD, PCG=METAB3, PCG=PRGNCY, Specialty=Internal, PG=EM, PG=SCS, PlaceSvc=Office.

Health, pruned: We subsample the set of attributes to DrugCount_total, DrugCount_months, no_Claims, no_Providers, PayDelay_total, PCG=COPD, PCG=METAB3, Specialty=Internal, PG=EM, PG=SCS, PlaceSvc=Office, AGE>60. Out of these we consider the last 6 as personal.

Loan. We additionally use the Loan dataset [86], an excerpt from the Lending Club loan data from 2015, as previously used by [19]. Here we use 42 attributes of persons, with the goal of predicting loan status. We use all categorical attributes (term, grade, sub_grade, emp_length, home_ownership, verification_status, pymnt_plan, purpose, initial_list_status, application_type, hardship_flag, disbursement_method, issue_d, addr_state) as the personal attributes.

UCI Crime. Finally, we use the Communities and Crime dataset from the UCI repository [87], which combines US socio-economic, law enforcement and crime data from several sources. It contains 128 attributes for each community, and the goal is to predict if the number of violent crimes per capita is above or below the median. We follow the preprocessing from [25], resulting in 99 attributes of which we consider (pre-processed) race and state as personal.

Appendix E. Additional Results

In Figure 11 we show additional results of our main experiment, extending the ones given in Figure 4. We explore several new datasets (described in Appendix D): ACSPublicCoverage from the ACS suite, full Health with 101 attributes, Loan and UCI Crime. In all cases we note the same results—PAT is the best minimizer overall, preserving utility while reducing the privacy risk.

In Figure 12 we explore additional feature selection methods based on mutual information and χ^2 values, and observe that they perform worse on ACSIncome compared

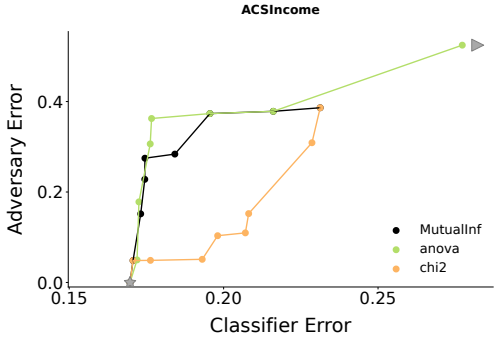


Figure 12: Comparison of feature selection methods on ACSIncome.

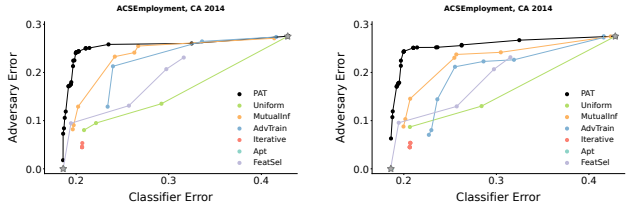


Figure 13: Main runs with two different seeds on ACSEmployment. We observe similar behavior across all seeds.

to the ANOVA method we used in our main experiment as a feature selection baseline. We additionally experimented with a variant of feature selection where we iteratively remove 1 feature at a time (until k remain), but observed that in all cases this resulted in the same final feature set.

Appendix F. Experimental Parameters

In all experiments, we use 60% of the data as the training set (used to fit the minimizers), 10% as held-out validation (used to select the Pareto front of generalizations and internal validation), and 30% as the test set (used for final results). We repeat runs with 5 seeds observing the same behavior across all main experiments, as we show in Figure 13.

As PAT requires each class to be in the training set at least once, we always (i.e., for all minimizers) first select a set of samples that ensures this, filling the training set with randomly drawn from the remaining dataset. The batch size is set to 256. We model the classifier and the adversary as neural networks with one hidden layer of width 50 and ReLU

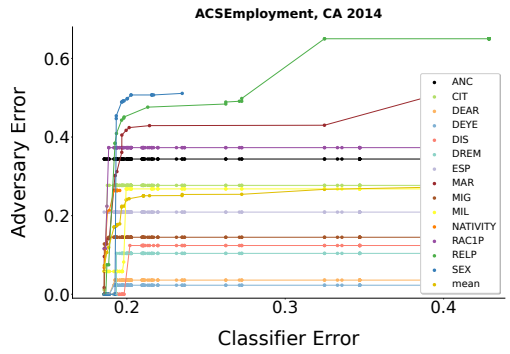


Figure 14: Utility-privacy tradeoff on ACSEmployment. For points on the Pareto front in Section 8 we report the adversarial error per attribute.

activations. For evaluation, classifier and adversary are trained for 20 epochs with learning rate 0.01 multiplied by 0.1 after 10 epochs. The value of the $L2$ regularization parameter is automatically tuned on the validation. Our framework performs several runs with various parameters before using the validation set to choose the Pareto front.

For the *uniform baseline*, the only varying parameter is k. We perform runs for $k \in \{1, 2, 3, 4, 5\}$. For *univariate feature selection*, we try all values $k \in \{2, 4, 8, 10, 20, all\}$.

We run *Apt* and the *Iterative* minimizer with 7-11 values of the target classifier error T, chosen for each dataset to be between the two limit points (no and full generalization).

For the *Iterative* minimizer, we set the initial number of buckets $k = 4$, as we did not observe further improvement with more buckets. For both *Iterative* and *Apt* we set a time-limit of $2h$ per run, $5\times$ the next slowest minimizer.

For the neural minimizers (*AdvTrain* and *MutualInf*), we perform runs with 9 values of $\lambda \in [0, 1]$, fixing the number of buckets to $k = 5$. We use the same classifier/adversary architectures as used in the evaluation and perform 20 epochs of training with starting learning rate of 0.01, multiplied by 0.1 every 5 epochs. We use no L2 regularization and schedule the softmax temperature from 2.0 to 0.5 over the course of training. For *AdvTrain* we set $N_{\text{inner}} = 1$.

For PAT, we ran experiments with the maximum number of leaves $k^* \in \{2, 4, 6, 8, 10, 20, 50, 100, 200\}$, requiring that at least 100 samples end up in every leaf. For each run, we selected the same 12 possible values for α in the range $[0, 1]$. PAT required roughly 3 seconds per run, significantly outperforming all other minimizers in terms of speed.

Appendix G. Architectures & Individual Attributes

We also perform experiments on the architecture of our A1 adversary and downstream classifier. In particular, for the ACSIncome dataset, we investigate how the adversarial architecture used in Section 8 (referred to as MLP-2) is sufficient for the reconstruction of personal attributes. For this, we additionally introduce MLP-3, a 3 layer neural network with an intermediate layer width of 100 neurons, as well as MLP-5, a 5 layer neural network with an intermediate width of 256 neurons. Table 6 shows the adversarial reconstruction accuracy using the aforementioned architectures for the possible minimizer choices AdvTrain, Uniform, and PAT. We find that for AdvTrain and Uniform, using a slightly larger architecture can sometimes improve the reconstruction accuracy, while for PAT, MLP-2 consistently performs the best. This indicates that the results for PAT in Section 8 are reasonably close to the best achievable adversarial error rate.

TABLE 6: Classifier and adversary error for different choices of minimizers and adversarial network architectures. We highlighted cases where the error is at least 2% smaller than on the baseline MLP-2 network.

	Arch.	Unif. k=3	Unif. k=5	Adv. α=0	Adv. α=0.5	PAT k^*=4 α=0.7	PAT k^*=10 α=0.7	PAT k^*=50 α=0.7
Classifier Error	MLP-2	0.36	0.22	0.17	0.22	0.2	0.19	0.17
	MLP-3	0.36	0.21	0.17	0.22	0.2	0.19	0.18
	MLP-5	0.36	0.21	0.18	0.22	0.2	0.19	0.18
Adversary Error	MLP-2	0.52	0.31	0.39	0.48	0.47	0.46	0.36
	MLP-3	0.52	**0.24**	**0.3**	0.5	0.47	0.46	0.36
	MLP-5	0.52	**0.24**	**0.31**	**0.43**	0.5	0.47	0.37

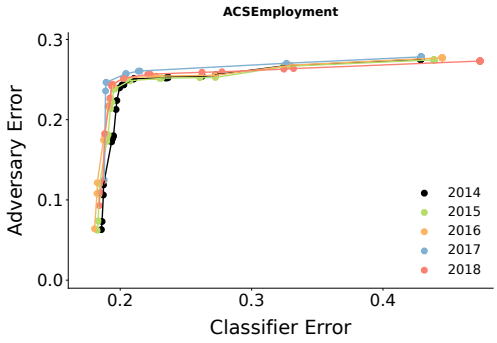


Figure 15: Robustness of PAT to distribution shift.

Extending the experiment from Section 8.2, we present another individual attribute plot in Figure 14 for the ACSEmployment dataset. As ACSEmployment has more personal attributes (14) compared to ACSIncome, Figure 14 becomes harder to interpret, however when focussing on individual attributes, we can observe the same trends as in Section 8.2. In particular, we again find for all attributes strong inflection points, after which the adversarial error decreases drastically.

Appendix H. Robustness to Distribution Shift

We test the robustness of PAT to temporal distribution shifts in the data. We fix the generalizations obtained on ACSEmployment data from 2014 (as used in our main experiments) and evaluate them under data from 2015, 2016, 2017 and 2018. The results are shown in Figure 15—we notice no significant degradation. We further remark that such an evaluation can be used in practice to monitor model drift, and when significant drift is detected a new small dataset of updated data can be collected to retrain the minimizer.

Appendix I.
Meta-Review

The following meta-review was prepared by the program committee for the 2024 IEEE Symposium on Security and Privacy (S&P) as part of the review process as detailed in the call for papers.

I.1. Summary

This paper proposes a data minimization workflow for machine learning tasks, focusing on collecting only the essential features and reducing the resolution of feature samples (e.g., buckets) to minimize privacy loss during potential data breaches. The authors assess its performance in eight adversarial contexts across five datasets.

I.2. Scientific Contributions

- Provides a Valuable Step Forward in an Established Field.

I.3. Reasons for Acceptance

1) This paper provides a valuable step forward in an established field. Developing machine learning models that adhere to data minimization principles has historically been a challenge. This research underscores the practical viability of data minimization methods.

I.4. Noteworthy Concerns

1) The authors didn't provide a theoretical analysis detailing the precise level of privacy assurance the proposed method can achieve within a given utility error boundary.

Appendix J.
Response to the Meta-Review

The meta-review notes that the work does not provide a theoretical analysis of the privacy-utility tradeoffs. As we point out in our future work section, obtaining a non-trivial and fully general theoretical result for vDM is a hard problem, with no clear solution within the scope of this paper. We agree that this can be a valuable extension of our work and encourage future efforts in this direction. We hope that the extensive vDM foundation set up by our work in terms of setting formalization, empirical risk estimation, baselines and PAT, can aid researchers in such follow-ups.