# DIFFERENTIALLY PRIVATE ADAPTATIONS FOR TABULAR FOUNDATION MODELS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Tabular foundation models (TFMs) achieve excellent predictive performance on novel tasks through in-context learning (ICL). Tabular ICL provides labeled records as context for pretrained models, avoiding costly per-task training or fine-tuning. However, the sensitivity of records for tabular ICL raises privacy concerns. Our work first shows that this is a real risk by instantiating highly effective membership inference attacks to infer whether specific target records were part of the private context in a TFM. To provably protect against such attacks, we propose two novel methods for differentially private adaptations of TFMs. First, we present TabDPSGD which adapts a TFM via gradient-based training on a learnable context and a subset of the model's parameters. TabDPSGD releases the trained context and the adapted weights, which can be used together for predictions informed by private data. However, gradients are not available for API-based TFMs. To remove this constraint, we also propose TabPATE, which only requires access to TFM outputs. TabPATE partitions private data across an ensemble of TFM teachers, privately aggregates their predictions on synthetic candidate queries, and releases the resulting labeled candidates as a student context for ICL. Across seven datasets and two TFMs, TabDPSGD beats the strongest evaluated private baseline on six datasets, while TabPATE outperforms private classification baselines.

## 1 INTRODUCTION

Tabular foundation models (TFMs) are competitive predictors for tabular tasks (Hollmann et al., 2023; Ma et al., 2025; Qu et al., 2026; Kong & Das, 2026; Grinsztajn et al., 2026b). They typically are pretrained across millions of tables (Hollmann et al., 2025) and apply in-context learning (ICL) (Brown et al., 2020) to make predictions on novel tasks, where a labeled dataset is provided as context and used to predict labels for new queries. TFMs are often applied without task-specific training, but fine-tuning can further improve performance on specific domains (Breejen et al., 2024). Although pretraining commonly uses synthetic data (Hollmann et al., 2025), downstream adaptation via fine-tuning or ICL uses real records from the target domain that may contain sensitive information, such as medical data or financial transactions. This use of real data raises concerns that private information may leak from the adapted TFM.

We first empirically verify this concern. To this end, we rely on the standard tool for evaluating privacy risk in machine learning, namely membership inference attacks (MIAs) (Shokri et al., 2017; Carlini et al., 2022). A MIA tests whether a specific record was part of a model's training data. Prior work has successfully adapted this approach to infer membership of in-context data for large language models (LLMs) (Duan et al., 2023a;b; Wen et al., 2024) and TFMs (Carvalho & Cordy, 2026). In a similar vein, we instantiate multiple different MIAs against tabular ICL to investigate the severity of leakage under different attacker assumptions and capabilities. We show that leakage even occurs with a passive adversary who observes only prediction probabilities, where our attacks reach a 9.9% true positive rate (TPR) at a 1% false positive rate (FPR). A stronger adversary that can also inject records into the context reaches up to 75.4% TPR at 1% FPR. These results motivate the need for privacy-protecting methods for TFMs to mitigate leakage risks.

Therefore, we ask: *Can a TFM be adapted to a private table while protecting the privacy of individual records?* In this work, we answer this question affirmatively with two methods for private adaptations of TFMs, both with formal guarantees under differential privacy (DP) (Dwork et al., 2006). To the best of our knowledge, we are the first to bring DP training to TFMs. Figure 1 shows an overview of



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

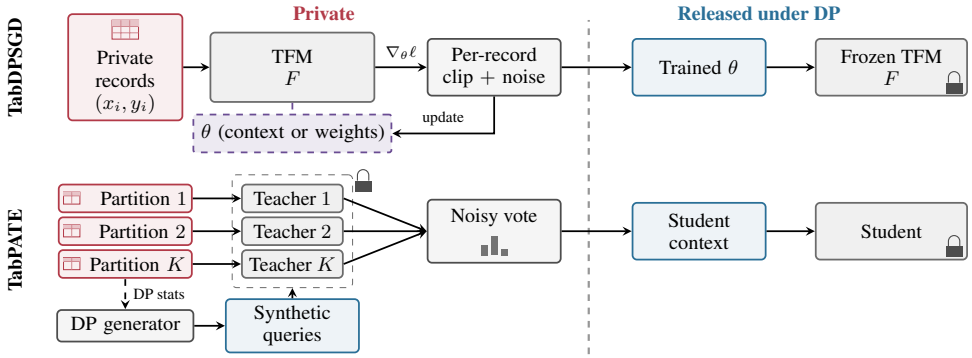


Figure 1: **Our two methods for private TFM adaptations.** TabDPSGD (top) trains a learnable context, TFM weights, or both from clipped and noised per-record gradients, then releases the trained parameters $\theta$. TabPATE (bottom) splits the private table into $K$ disjoint partitions, uses one frozen TFM to create $K$ teachers with different contexts, privately aggregates their predictions on synthetic candidate records, and releases the labeled records as a student context for inference.

both methods. First, we introduce **TabDPSGD**, a method that trains with gradients and replaces the sensitive context records with learnable parameters. Specifically, it leverages private data to *train a context* that is then used during inference by the TFM. Additionally, it can also train (parts of) the TFM weights. It clips and noises each per-record gradient before updating the trained parameters, as in the original DPSGD algorithm (Abadi et al., 2016). We evaluate five variants that train: (1) the learnable context alone or (2) together with an affine output transformation; (3) LoRA weights (Hu et al., 2022); (4) the TFM's output head; or (5) all TFM weights.

However, TabDPSGD requires gradients with respect to the learnable context or the TFM weights. These are not possible to obtain for some TFM APIs that return predictions alone (*e.g.,* Prior Labs' hosted TabPFN service (Prior Labs, 2026)). Therefore, we propose our second method, **TabPATE**, which only requires access to the model's output labels and only modifies the model's context. TabPATE adapts Private Aggregation of Teacher Ensembles (PATE) (Papernot et al., 2018; Duan et al., 2023a) to tabular ICL. Specifically, TabPATE partitions the private table across teacher TFMs, queries them on synthetic records, and aggregates their votes using the differentially private GNMax algorithm (Papernot et al., 2018). The resulting labeled records are released as a student context that is used for private prediction. Unlike earlier methods built on PATE, such as PromptPATE (Duan et al., 2023a), TabPATE does not need public in-distribution data. Instead, it uses the structure of tabular data, whose features are typically bounded, typed, and low-dimensional compared to images and text, hence teacher queries can be synthetically generated. Moreover, we provide a tighter data-dependent privacy analysis of GNMax for binary classification.

We assess TabDPSGD on seven tabular datasets and two TFMs against a DPSGD-trained MLP, DPBoost (Li et al., 2020), and four DP synthetic data generation methods. Similarly, we evaluate TabPATE on five classification datasets. Our results show that TabDPSGD and TabPATE achieve state-of-the-art performance on tabular predictive tasks under privacy constraints.

In summary, we make the following contributions:

1. We empirically verify that the in-context data provided to a TFM is subject to privacy risks. We tailor diverse MIAs to the tabular ICL setting and establish that private in-context data can leak under all attack setups.
2. We propose TabDPSGD, which adapts a TFM to a private table by training a learnable context, TFM weights, or both, and provides a formal DP guarantee for the private table.
3. We propose TabPATE, which adapts a TFM to a private table without access to gradients or public in-distribution data. To enable TabPATE, we derive a tighter data-dependent accounting for GNMax aggregation under privacy constraints in the binary case.
4. We empirically establish that our private adaptations of a pretrained TFM yield better utility than private training from scratch, setting the state-of-the-art for private predictions on tabular data.

## 2 Background and Related Work

**Tabular Foundation Models (TFMs).** TFMs (Hollmann et al., 2023; Ma et al., 2025; Qu et al., 2025) are pretrained transformer-based (Vaswani et al., 2017) prior-data fitted networks (Müller et al., 2022) that perform predictive tabular tasks through ICL. Given labeled records as context, a TFM predicts the label of a novel query in a single forward pass, without any gradient update. Recent TFMs (Qu et al., 2026; Grinsztajn et al., 2026b; Kong & Das, 2026; Hosseinzadeh et al., 2026) outperform strong gradient-based methods such as XGBoost (Chen & Guestrin, 2016), CatBoost (Prokhorenkova et al., 2018), and RealMLP (Holzmüller et al., 2024) on popular tabular benchmarks (Liu et al., 2025; Erickson et al., 2025). However, similar to the prevailing situation with LLMs, the strongest TFMs are accessible only through commercial APIs (Prior Labs, 2026), which raises concerns about the privacy of data provided as context.

**Adaptations of TFMs.** TFMs are typically pretrained on synthetic data to be general-purpose predictors, but it can still be advantageous to adapt TFMs to specific tasks. For instance, gradient-based adaptations of the TFM's weights using real tables can improve domain performance (Breejen et al., 2024). Variants either fine-tune the entire TFM (Thomas et al., 2024; Rubachev et al., 2025) or use parameter-efficient fine-tuning methods (*e.g.,* inserting LoRA blocks (Hu et al., 2022)) (Xu et al., 2025; Liu & Ye, 2025; Tanna et al., 2026). Alternatively, one can adapt the context by optimizing context tokens directly (Ma et al., 2024; Feuer et al., 2024), or by retrieving the most relevant records (Thomas et al., 2024). These methods use real (possibly sensitive) data without evaluating the potential risks of privacy leakage. Our TabDPSGD uses gradient-based adaptations, while our TabPATE uses context adaptations, and both provably limit privacy leakage.

**Membership Inference Attacks (MIAs).** MIAs ask whether a candidate record was part of the data on which a model was trained (Shokri et al., 2017; Carlini et al., 2022). MIAs also serve as canonical tools for auditing empirical privacy protection. Intuitively, the candidate record used for training stands out because the model responds to it with higher confidence than to a record it has not seen. Prior work adapts MIAs to the data placed in an LLM prompt (Duan et al., 2023a;b; Wen et al., 2024). While recent work has shown that TFMs can be vulnerable to MIAs (Carvalho & Cordy, 2026), we systematically verify the effectiveness of MIAs in the tabular ICL setting across a wider range of threat models, including context manipulation attacks.

**Differential Privacy (DP).** DP (Dwork et al., 2006) is a mathematical framework that provides a formal privacy guarantee by bounding the influence of any single record on a released output. It is one of the main tools used to defend against MIAs and other privacy attacks (Aerni et al., 2024; Duan et al., 2023a; Marek et al., 2026), and to prevent unintended memorization of training data (Capano & Böhler, 2026). We use consider neighbouring datasets $D, D' \in \mathcal{X}$ where one is obtained from the other by adding or removing one record. Formally, a randomized algorithm $\mathcal{M} : \mathcal{X} \to \mathcal{Y}$ is $(\varepsilon, \delta)$-DP if for all neighboring datasets $D$, $D' \in \mathcal{X}$ and for all $Y \subseteq \mathcal{Y}$:

$$\Pr[\mathcal{M}(D) \in Y] \leq e^{\varepsilon} \cdot \Pr[\mathcal{M}(D') \in Y] + \delta. \tag{1}$$

Intuitively, DP algorithms ensure that outputs obtained from similar inputs are difficult to distinguish, making it hard to tell if any given record was included in $D$. $\varepsilon$ and $\delta$ quantify the strength of the privacy guarantee, with smaller values implying stronger guarantees. There are two canonical ways to make a machine learning algorithm satisfy Eq. (1). First, **DPSGD** (Abadi et al., 2016) adds privacy during training. It clips the gradient of every record to a fixed norm and adds Gaussian noise to the sum of the clipped gradients at each update. The $\varepsilon$, $\delta$ parameters are tracked by a numerical accountant over the course of a training run (Gopi et al., 2021). Second, **PATE** (Papernot et al., 2017; 2018) ensures privacy during knowledge transfer for classification. It partitions the private data, trains a separate teacher model on each partition, labels public data (from the same distribution as the private data) by computing an argmax over noisy teacher votes, and trains a student model on those privately released labels. DPSGD requires access to the model's gradients, while PATE needs only predictions. In our work, we follow both approaches to provide differentially-private adaptations for TFMs.

**DP Predictors for Tabular Tasks.** For private tabular prediction tasks, prior work follows three directions. The first trains a predictor directly under DP, such as an MLP or a decision tree ensemble, with a differentially private training algorithm (Abadi et al., 2016; Li et al., 2020). The second releases a synthetic table under DP using marginals (McKenna et al., 2021; 2022), a generative model (Fang et al., 2022), or an evolutionary search over synthetic records (Tran et al., 2026), after which any predictor can be trained non-privately. In contrast, our methods focus on private adaptations of pretrained TFMs. We find this approach achieves better utility compared to private training from scratch and private data generation.

**Setup and Notation.** We write $F$ for the TFM and $W$ for its pretrained weights. $F(x; \mathcal{C}, W)$ denotes a prediction, where $x$ is the query record and $\mathcal{C}$ is the TFM's private labeled context. For classification tasks, $F(x; \mathcal{C}, W)$ is a probability vector over the $M$ classes of the downstream task, and for regression it is a scalar. For adapted TFMs we denote the learnable parameters as $\theta = (\theta_{\mathcal{C}}, \theta_W)$, where $\theta_{\mathcal{C}}$ represents the learnable context tokens which replace $\mathcal{C}$, and $\theta_W$ denotes a subset of $W$. We write $F_\theta(x)$ for the resulting adapted predictor. The private table $D_{\text{priv}} = \{(x_i, y_i)\}_{i=1}^N$ is the data to be protected. Table 5 collects the notation we use throughout this paper.

## 3 Membership Inference Attacks on Tabular ICL

We first analyze the vulnerability of the private context to leakage. In our threat model, the adversary holds a target record $(x_t, y_t)$ and must decide whether it is part of the private context $\mathcal{C}$. We consider adversary capabilities along two axes. *Output access* allows accessing either labels only (black-box) or prediction probabilities (gray-box) from the TFM. *Context manipulation* allows either passive or active control of $\mathcal{C}$. The former keeps $\mathcal{C}$ unmodified, whereas the latter enables the adversary to inject additional records into $\mathcal{C}$. Combined, these axes define four threat models, detailed in Appx. B.1. We extend MIA techniques to TFMs to create one attack in each setting, namely label-only perturbation (Choquette-Choo et al., 2021), likelihood-ratio attacks (LiRA) (Carlini et al., 2022), Brainwash context manipulation (Wen et al., 2024), and LiRA with Truth Serum context poisoning (Tramèr et al., 2022).

All attacks use the same likelihood ratio calibration (Carlini et al., 2022). Instead of training shadow models, as is typically done for LiRA, we create shadow contexts $\mathcal{C}_i$ that each define a predictor $F(x; \mathcal{C}_i, W)$. For each target record $x_t$ we create $N$ shadow contexts that include $x_t$, and $N$ that exclude it. We then extract an attack-specific signal $s(x_t)$ per-shadow context, fit Gaussian densities $\pi_{\text{IN}}$ and $\pi_{\text{OUT}}$ on the two sets of signals, and score membership by the log-likelihood ratio

$$\text{score}(x_t) = \log \frac{\pi_{\text{IN}}(s(x_t))}{\pi_{\text{OUT}}(s(x_t))}. \tag{2}$$

The four attacks differ only in the signal $s(x_t)$, and the shadow contexts. The details of each case are given in Appx. B.1. In short, label-only perturbation uses prediction stability under perturbations, LiRA uses logit-scaled confidence, Brainwash uses the number of mislabeled context injections required to flip the prediction, and Truth Serum uses confidence after mislabeled context injection.

**Experimental Setup.** We run all four MIAs against an ICL-adapted TabPFNv2.5 (Grinsztajn et al., 2026a). We use the following five datasets for the evaluation: Blood Transfusion, Diabetes, German Credit, Vehicle, and KC1. We partition each dataset into an $80\%/20\%$ train/test split and use Bernoulli sampling with $p = 0.5$ to sample shadow context members from the train split. We treat each MIA as a binary classification task using the scores from Eq. (2) to rank members and non-members $x_t$, and report AUC and TPR@1%FPR to evaluate the strength of each attack.

**Results.** Tabular ICL leaks membership information from its context in all four threat models. Table 1 shows that even a passive adversary observing prediction probabilities reaches an average AUC of $0.73$ and TPR@1%FPR of $9.9\%$. This is the most realistic scenario given that all of today's leading TFMs output prediction probabilities. With active context injection, these metrics rise to an AUC of $0.96$ and TPR of $75.4\%$. These results establish that records placed in the context of a TFM are exposed to a measurable privacy risk. Simply applying a pretrained TFM to new data via tabular ICL leaves that data vulnerable. Hence, in the rest of the paper we pursue private adaptations of TFMs, which use private data to modify the TFM or tabular ICL mechanism, maintaining predictive utility while providing a formal upper bound on the leakage of the private adaptation data.

Table 1: **Privacy leaks from tabular ICL.** Per-dataset results of our four instantiated MIAs showing AUC and TPR@1%FPR. Each row is a distinct threat model defined by what the adversary observes and whether it can inject records into the context. Mean is the average across rows.

| | Blood Transfusion | | Diabetes | | German Credit | | Vehicle | | KC1 | | Mean | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Attack** | AUC | TPR | AUC | TPR | AUC | TPR | AUC | TPR | AUC | TPR | AUC | TPR |
| Perturbation (black-box, passive) | 0.51 | 1.0% | 0.53 | 1.5% | 0.56 | 1.7% | 0.64 | 1.3% | 0.52 | 1.5% | 0.55 | 1.4% |
| LiRA (gray-box, passive) | 0.60 | 3.6% | 0.68 | 6.7% | 0.76 | 10.9% | 0.91 | 20.0% | 0.68 | 8.5% | 0.73 | 9.9% |
| Brainwash (black-box, active) | 0.66 | 6.7% | 0.81 | 18.9% | 0.88 | 32.4% | 0.96 | 77.7% | 0.73 | 12.7% | 0.81 | 29.7% |
| Truth Serum (gray-box, active) | 0.83 | 15.0% | 1.00 | 98.2% | 1.00 | 99.8% | 1.00 | 96.8% | 0.96 | 67.0% | 0.96 | 75.4% |

## 4 TABDPSGD: DPSGD FOR PRIVATE TFM ADAPTATIONS

We propose TabDPSGD, which, to the best of our knowledge, is the first method that adapts a TFM to a downstream task under DP by leveraging the canonical DPSGD algorithm (Abadi et al., 2016). For TabDPSGD, we replace the real tabular context $\mathcal{C}$ with the learned context $\theta_{\mathcal{C}}$. Then, TabDPSGD adapts parameters $\theta$, consisting of the context parameters $\theta_{\mathcal{C}}$ and also a subset $\theta_W$ of all of the weights $W$ in a pretrained TFM. For each private record $(x_i, y_i)$, TabDPSGD minimizes loss $\ell_i(\theta) = \ell(F_\theta(x_i), y_i)$, using cross-entropy for classification and squared error for regression. We present a detailed Algorithm 1 for our TabDPSGD in Appx. C.

We evaluate five TabDPSGD variants, namely four parameter-efficient fine-tuning (PEFT) variants and full fine-tuning. We detail each variant below.

**TabDPSGD-Context.** This variant trains only the learned context $\theta_{\mathcal{C}}$. The TFM itself is kept frozen, *i.e.,* its parameters are left unchanged. In the context, numeric values are trainable, and each categorical column carries trainable logits decoded through a differentiable softmax. TabDPSGD-Context is the tabular counterpart of private prompt tuning for LLMs (Duan et al., 2023a). We initialize the learned context by setting its values to evenly spaced quantiles of a uniform distribution, ensuring uniform coverage of the initialization range.

**TabDPSGD-Logit.** This variant adds a trainable scale $\theta_{s_j}$ and bias $\theta_{b_j}$ to each model output $j$. For classification, the parameters are added to the query logit of each class. For regression, they are added to the distributional logits before decoding. We initialize $\theta_{s_j} = 1$ and $\theta_{b_j} = 0$, so training starts at the behavior of the frozen model.

**TabDPSGD-LoRA.** This variant extends LoRA (Hu et al., 2022), which represents an update to a frozen linear map as the product of two trainable low-rank factors. LoRA was originally applied to the query and value projections of self-attention in LLMs. We apply the low-rank updates to the feed-forward layers of each transformer block. We initialize the first low-rank factor with Kaiming uniform initialization (He et al., 2015) and the second to zero, so the adapter initially preserves the frozen TFM.

**TabDPSGD-Head.** This variant trains the TFM's output head, *i.e.,* $\theta_W$ is the head alone. The head is a two-layer network that maps the final representation of a query record to class logits or to the regression output. Its role differs from the output head of an LLM. An LLM applies its head at every token position to predict the next token. A TFM applies its head to query records only.

**TabDPSGD-Full.** We train all the parameters of the TFM, *i.e.,* $\theta_W = W$. Full fine-tuning has shown strong performance in the non-private setting (Rubachev et al., 2025), hence we include it to investigate whether it stays competitive under DP. This variant has the largest memory cost out of all of our proposed TabDPSGD variants, since it carries a per-record gradient and optimizer state for every TFM parameter.

While these variants can successfully achieve DP guarantees for the private data when gradients are accessible, API-based tabular prediction services, such as Prior Labs' hosted TabPFN service (Prior Labs, 2026), only provide access to prediction probabilities. This motivates the second method for our differentially private adaptations of TFMs, which modifies only the context and does not require access to the parameters of the underlying TFM.

## 5 TABPATE: PATE FOR PRIVATE TFM ADAPTATIONS

PATE (Papernot et al., 2017; 2018) trains an ensemble of teacher models on disjoint partitions of private data, privately aggregates their votes on public unlabeled data, and trains a student model on the resulting labeled data. Intuitively, PATE performs a private knowledge transfer from the teacher models to the student. The student model is released for public predictions. TabPATE retains these components but replaces the trained teachers with one frozen TFM applied to different contexts, and replaces the trained student with a labeled context. Given the private table $D_{\text{priv}}$, TabPATE releases a student context $\tilde{\mathcal{C}}$ that conditions every subsequent prediction without further access to $D_{\text{priv}}$. Unlike previous methods built on PATE (Papernot et al., 2018; Duan et al., 2023a), **TabPATE requires neither teacher training nor public data from the same distribution as the private data**.

**Teachers.** Following PATE, we split the private table into $K$ disjoint partitions $D_1, \ldots, D_K$ via a deterministic mapping function $\pi$ (*i.e.,* a hashing function), which assigns each labeled record $(x, y)$ to partition $D_{\pi(x,y)}$. Thus, adding or removing one record $(x, y)$ changes only $D_{\pi(x,y)}$; the

assignments of all other records remain unchanged. A well-distributed hash improves partition balance but is not needed for privacy. Each nonempty partition acts as the context for a frozen TFM $F$, while an empty partition abstains. TabPATE does not train separate teacher models. The same frozen TFM generates every teacher output, and partition $D_j$ influences only teacher $j$.

**Candidate queries.** PATE assumes an unlabeled public dataset to transfer the teachers' knowledge to the student. We remove this assumption by leveraging properties of tabular data. Tabular features are typed, bounded, and low-dimensional, which allows us to sample synthetic candidates from the same feature space with limited privacy cost. The candidates are unlabeled and used as queries for the teachers. We draw each feature of a candidate independently within the range fixed by the released scaling rule, which spends no privacy budget. Alternatively, we spend a small part of the budget to estimate the marginal distribution of each feature (*i.e.,* its mean and variance) privately and draw candidates from these estimates, so that they concentrate where the private records lie. Appx. D specifies the query-generation procedure and its privacy cost.

**Private knowledge transfer.** Let $Q = (q_i)_{i=1}^{L}$ be the fixed sequence of sampled candidates. Each teacher votes for one of the two classes on every candidate query $q_i$, and a vote histogram $n(q_i)$ counts these votes. In PATE, the disjoint partitions bound the sensitivity of this histogram because a private record enters only one partition. TabPATE labels every candidate with GNMax (Papernot et al., 2018), which independently adds $\mathcal{N}(0, \sigma^2)$ noise to each class count and returns the noisy argmax. The privacy cost of an answer is data-dependent and decreases as teacher agreement increases. We compose these costs using numerical privacy accounting with Rényi DP (Mironov, 2017).

**Student.** The classical PATE student is a model trained by supervised learning on many noisy labeled queries. TabPATE instead forms the student context $\tilde{\mathcal{C}} = \{(q_i, \hat{y}_i) \mid 1 \le i \le L\}$ from the noisy labels. The same frozen TFM $F$ conditions on this context to predict $F(q_{\text{new}}; \tilde{\mathcal{C}}, W)$ for a new query record $q_{\text{new}}$. This inference is post-processing and spends no further privacy budget.

**Tighter accounting.** The privacy cost of releasing all $L$ labels is the sum of per-query Rényi divergences between the label distributions on neighboring tables. Standard PATE (Papernot et al., 2018) bounds each term from two quantities only: the data-independent Rényi DP of GNMax, and the probability that the noisy vote deviates from the plurality class. This bound is loose in binary classification for two reasons. First, the label released for one query is exactly Bernoulli in the vote gap, so both neighboring output laws are available in closed form and the divergence can be computed exactly rather than bounded through worst-case sensitivity. Second, PATE takes a worst case over neighbors separately for each query, so one query may be charged for a change in one teacher and the next query for a change in another, although one record can affect only one teacher. We propose an accountant (Appx. E.1) that fixes the affected teacher across all queries, sums the exact per-query divergences, and only then maximizes over teachers. Our accountant allows answering more queries under the same budget, which improves the student's utility.

**Certified release.** Because this data-dependent cost is computed from the private votes, it is not itself a privacy guarantee. Publishing it, or using it to decide how many labels to release, would leak information about the private table. PATE handles this by releasing a noisy estimate of the cost as a post-hoc diagnostic (Appx. E.2), while the labels retain only their data-independent guarantee. We instead want the released student context to carry the tighter bound as a formal guarantee. We therefore follow generalized propose-test-release (PTR) (Redberg et al., 2023). Specifically, we propose a cost before observing the votes, privately test whether the data-dependent cost of the labels stays below it using smooth sensitivity (Nissim et al., 2007; Papernot et al., 2018), and release the student context only if the test passes (Appx. E.3).

## 6 Experimental Evaluation

We evaluate our TabDPSGD and TabPATE methods on tabular classification and regression tasks. We also compare our approaches against prior DP methods. Our code for the empirical evaluation is available at: https://anonymous.4open.science/r/dp-tfms-1DFA/.

**Datasets.** We carry out our evaluation on seven tabular datasets used in prior work for tabular ICL, with five classification tasks and two regression tasks. We evaluate TabDPSGD on all seven and TabPATE on the five classification datasets. We provide additional information about the datasets in Appx. F.1. Moreover, in contrast to prior work (Tran et al., 2026; McKenna et al., 2022; 2021), we include data normalization and preprocessing costs in the total privacy budget (see Appx. F.3).

Table 2: **TabDPSGD leads the private comparison.** We report the Elo rating and scores for every dataset. The scale fixes the DPSGD MLP at 1000 Elo. The best score per column is **bolded** and the second best is underlined.

| Method | Elo ↑ | Electricity AUC ↑ | ACS Employ AUC ↑ | Credit AUC ↑ | Churn AUC ↑ | Polish AUC ↑ | California $R^2$ ↑ | Bike Sharing $R^2$ ↑ |
|---|---|---|---|---|---|---|---|---|
| TabDPSGD-Context (ours, TabICLv2) | <u>1225</u> | 0.846 | <u>0.868</u> | 0.726 | 0.798 | 0.815 | <u>0.790</u> | **0.894** |
| TabDPSGD-Context (ours, Real-TabPFN-2.5) | 1141 | <u>0.855</u> | 0.858 | 0.754 | 0.773 | 0.810 | 0.759 | 0.772 |
| TabDPSGD-Logit (ours, TabICLv2) | 1159 | <u>0.855</u> | 0.853 | 0.729 | 0.773 | 0.821 | 0.782 | 0.850 |
| TabDPSGD-Logit (ours, Real-TabPFN-2.5) | 1132 | 0.851 | 0.844 | <u>0.755</u> | 0.795 | 0.815 | 0.751 | 0.643 |
| TabDPSGD-LoRA (ours, TabICLv2) | **1342** | 0.848 | **0.873** | 0.750 | 0.812 | <u>0.822</u> | 0.764 | <u>0.865</u> |
| TabDPSGD-LoRA (ours, Real-TabPFN-2.5) | 1215 | 0.853 | 0.867 | 0.751 | 0.794 | 0.820 | 0.747 | 0.829 |
| TabDPSGD-Head (ours, TabICLv2) | 1187 | **0.867** | 0.865 | 0.749 | 0.790 | 0.812 | **0.798** | 0.470 |
| TabDPSGD-Head (ours, Real-TabPFN-2.5) | 1095 | 0.852 | 0.860 | 0.729 | **0.827** | 0.812 | 0.731 | 0.619 |
| TabDPSGD-Full (ours, TabICLv2) | 1159 | 0.844 | 0.866 | **0.758** | <u>0.821</u> | 0.796 | 0.758 | 0.750 |
| TabDPSGD-Full (ours, Real-TabPFN-2.5) | 937 | 0.828 | 0.839 | 0.706 | 0.785 | 0.809 | 0.755 | 0.860 |
| DPSGD MLP (Abadi et al., 2016) | 1000 | 0.844 | 0.862 | 0.745 | 0.747 | <u>0.822</u> | 0.671 | 0.406 |
| DPBoost (Li et al., 2020) | 81 | 0.700 | 0.789 | 0.684 | 0.683 | 0.502 | −0.096 | −0.096 |
| AIM (McKenna et al., 2022) | 253 | 0.830 | 0.725 | 0.502 | 0.723 | 0.662 | 0.513 | 0.141 |
| MST (McKenna et al., 2021) | 229 | 0.740 | 0.827 | 0.550 | 0.703 | 0.783 | 0.140 | −0.190 |
| DP-CTGAN (Fang et al., 2022) | 157 | 0.771 | 0.766 | 0.524 | 0.714 | 0.786 | −0.027 | −0.412 |
| Tab-PE (Tran et al., 2026) | 902 | 0.838 | 0.821 | 0.731 | 0.753 | **0.827** | 0.607 | 0.205 |

**TFMs and budgets.** We evaluate TabDPSGD with Real-TabPFN-2.5 (Garg et al., 2025) and TabICLv2 (Qu et al., 2026). TabPATE uses TabICLv2 and TabPFN-3 (Grinsztajn et al., 2026b) as teachers and TabICLv2, TabPFN-3, and XGBoost (Chen & Guestrin, 2016) as students. Our main results use $\varepsilon = 1$ with $\delta = 10^{-5}$. Additionally, we evaluate TabDPSGD and the baselines at $\varepsilon \in \{1, 3, 10\}$ on two of the datasets.

**Baselines.** We use two non-TFM private predictors as baselines. First, we train a multilayer perceptron (MLP) with DPSGD (Abadi et al., 2016). Second, we use DPBoost (Li et al., 2020), which adapts LightGBM (Ke et al., 2017) with DP. We also evaluate four DP synthetic data generation methods. AIM (McKenna et al., 2022) and MST (McKenna et al., 2021) fit graphical models to noisy low-order marginals, DP-CTGAN (Fang et al., 2022) fits a generative model, and Tab-PE (Tran et al., 2026) iteratively refines a set of synthetic records through noisy nearest-neighbor votes of the private records. A non-private predictor such as XGBoost can be trained on each released synthetic table without further privacy cost. We describe the private baselines in depth in Appx. F.2.

**Metrics.** We report AUC for classification and $R^2$ for regression tasks. Since the AUC and $R^2$ metrics are not directly comparable, we additionally compute an Elo rating to summarize each method's performance across tasks. To compare a private method with its non-private baseline, we report the share it retains of the normalized score, which rescales AUC and $R^2$ so that an uninformative predictor scores 0 and a perfect predictor scores 1. Further details are provided in Appx. F.4.

### 6.1 TabDPSGD

**Comparison with private baselines.** Our results in Table 2 show that TabDPSGD performs better than the private baselines in most cases. In particular, TabDPSGD-LoRA with TabICLv2 achieves the highest Elo rating of 1342, compared with 1000 for the strongest evaluated baseline, DPSGD MLP. The choice of TFM and trained parameters both matter. TabICLv2 achieves a higher Elo than Real-TabPFN-2.5 for every adaptation variant, and all five TabICLv2 variants rank above the DPSGD MLP. Among the TabDPSGD variants, LoRA achieves the highest Elo, while Context, Head, and Full each attain the best score on at least one dataset. Full fine-tuning does not consistently improve over the PEFT methods and reaches at most 1159 Elo.

**Privacy budgets.** We also compare TabDPSGD with the baselines across different privacy budgets $\varepsilon \in \{1, 3, 10\}$. Table 3 shows the AUC score for the Churn dataset and $R^2$ scores for Bike Sharing. Overall, TabDPSGD outperforms the baselines on all privacy budgets. We also observe that TabDPSGD provides its largest gains at tighter privacy budgets, while the advantage over private training from scratch narrows as $\varepsilon$ increases. On Churn, the improvement decreases from 0.080 AUC at $\varepsilon = 1$ to 0.012 AUC at $\varepsilon = 10$. We observe the same trend on Bike Sharing, where the best TabDPSGD model improves over the DPSGD MLP by 0.488 $R^2$ at $\varepsilon = 1$, and by 0.113 $R^2$ at $\varepsilon = 10$.

Table 3: **TabDPSGD leads across privacy budgets.** Scores at $\varepsilon \in \{1, 3, 10\}$ on Churn and Bike Sharing. The best score per column is **bolded** and the second best is underlined.

| Method | Churn | | | Bike Sharing | | |
|---|---|---|---|---|---|---|
| | $\varepsilon = 1$ AUC ↑ | $\varepsilon = 3$ AUC ↑ | $\varepsilon = 10$ AUC ↑ | $\varepsilon = 1$ $R^2$ ↑ | $\varepsilon = 3$ $R^2$ ↑ | $\varepsilon = 10$ $R^2$ ↑ |
| TabDPSGD-Context (ours, TabICLv2) | 0.798 | 0.825 | <u>0.845</u> | **0.894** | 0.883 | **0.931** |
| TabDPSGD-Context (ours, Real-TabPFN-2.5) | 0.773 | 0.830 | **0.848** | 0.772 | **0.904** | 0.763 |
| TabDPSGD-Logit (ours, TabICLv2) | 0.773 | 0.828 | 0.838 | 0.850 | 0.893 | 0.913 |
| TabDPSGD-Logit (ours, Real-TabPFN-2.5) | 0.795 | 0.808 | 0.828 | 0.643 | 0.807 | 0.886 |
| TabDPSGD-LoRA (ours, TabICLv2) | 0.812 | 0.831 | 0.829 | <u>0.865</u> | 0.888 | <u>0.923</u> |
| TabDPSGD-LoRA (ours, Real-TabPFN-2.5) | 0.794 | 0.814 | 0.832 | 0.829 | 0.762 | 0.920 |
| TabDPSGD-Head (ours, TabICLv2) | 0.790 | 0.830 | 0.837 | 0.470 | 0.593 | 0.691 |
| TabDPSGD-Head (ours, Real-TabPFN-2.5) | **0.827** | 0.823 | 0.748 | 0.619 | 0.810 | 0.842 |
| TabDPSGD-Full (ours, TabICLv2) | <u>0.821</u> | **0.834** | 0.834 | 0.750 | 0.771 | 0.836 |
| TabDPSGD-Full (ours, Real-TabPFN-2.5) | 0.785 | <u>0.832</u> | 0.785 | 0.860 | <u>0.903</u> | 0.864 |
| DPSGD MLP (Abadi et al., 2016) | 0.747 | 0.817 | 0.836 | 0.406 | 0.585 | 0.818 |
| DPBoost (Li et al., 2020) | 0.683 | 0.608 | 0.777 | −0.096 | 0.201 | 0.355 |
| AIM (McKenna et al., 2022) | 0.723 | 0.759 | 0.663 | 0.141 | 0.495 | 0.316 |
| MST (McKenna et al., 2021) | 0.703 | 0.702 | 0.747 | −0.190 | 0.138 | 0.081 |
| DP-CTGAN (Fang et al., 2022) | 0.714 | 0.396 | 0.662 | −0.412 | 0.223 | 0.442 |
| Tab-PE (Tran et al., 2026) | 0.753 | 0.819 | 0.828 | 0.205 | 0.381 | 0.553 |

**Comparison with non-private adaptation.** Figure 4 in Appx. G.3 compares the private utility with the corresponding non-private baselines. The best private TabDPSGD variant retains a median of $91\%$ of the utility of non-private ICL with TabICLv2 and $90\%$ with Real-TabPFN-2.5, compared with $83\%$ for the DPSGD MLP relative to the tuned MLP. The largest difference occurs on Bike Sharing, where the DPSGD MLP loses substantially more utility and TabDPSGD achieves its largest advantage. Moreover, Table 14 shows that without privacy, a learned context alone stays within $0.02$ AUC or $R^2$ of non-private ICL with Real-TabPFN-2.5 on five of six datasets, and within $0.058$ on Polish, while training weight parameters alongside it changes the score by at most $0.006$ on those five datasets and adds $0.024$ on Polish. The utility TabDPSGD-Context loses at $\varepsilon = 1$, from $0.035$ on ACS Employ to $0.175$ on Bike Sharing, therefore comes mostly from the clipping and noise of DPSGD rather than from replacing the real records with a learned context.

Figure 5 in Appx. G.4 compares the best private and non-private configurations for the same Real-TabPFN-2.5 adaptation variant and dataset. Private training consistently prefers a higher learning rate, a larger batch size, and fewer optimization steps than non-private training, each in 27 of 30 configurations. This qualitatively agrees with prior work on private language models, which finds that DP training benefits from larger batches than standard non-private fine-tuning (Li et al., 2022), and that tighter privacy favors larger batches and fewer iterations (McKenna et al., 2025). Moreover, comparing Table 2 with Table 13 in Appx. G.2, we show that private and non-private Elo favor different variants when the non-private variants use a learned context. In particular, LoRA achieves the highest private Elo at $\varepsilon = 1$, while Context achieves the highest non-private Elo with a learned context. With the real context, LoRA also achieves the highest non-private Elo.

### 6.2 TabPATE

**Comparison with private baselines.** We compare TabPATE against the private prediction baselines on the five classification datasets. Table 4 reports the corresponding Elo and AUC scores. Both TabPATE variants rank above the strongest baseline (DPSGD MLP at $1000$ Elo), with TabPFN-3 teachers reaching $1313$ Elo and TabICLv2 teachers reaching $1128$. Against these baselines, TabPATE achieves the best AUC on four of the five datasets, while the DPSGD MLP performs best on Electricity. TabPATE with TabPFN-3 teachers matches the best TabDPSGD variant with TabICLv2 at $1313$ Elo and achieves a higher AUC on Churn and Polish. For the same underlying TFM, when the access to TFM parameters is available, the stronger gradient-based TabDPSGD method is preferred over TabPATE due to its superior performance. In our experiments with TabICLv2 used for both of our methods, TabDPSGD achieved an Elo score of 1313, compared with 1123 for TabPATE.

Table 4: **TabPATE and TabDPSGD lead the private classification ranking.** We report Elo and AUC across five classification datasets, with the DPSGD MLP fixed at 1000 Elo. We **bold** the best value per column, and underline the second best.

| Method | Elo ↑ | Electricity AUC ↑ | ACS Employ AUC ↑ | Credit AUC ↑ | Churn AUC ↑ | Polish AUC ↑ |
|---|---|---|---|---|---|---|
| TabPATE (ours, TabICLv2) | 1128 | 0.828 | 0.866 | 0.757 | 0.825 | 0.832 |
| TabPATE (ours, TabPFN-3) | **1313** | 0.830 | 0.868 | **0.758** | 0.825 | **0.849** |
| TabDPSGD, best variant (ours, TabICLv2) | **1313** | **0.867** | **0.873** | **0.758** | 0.821 | 0.822 |
| TabDPSGD, best variant (ours, Real-TabPFN-2.5) | 1204 | 0.855 | 0.867 | 0.755 | **0.827** | 0.820 |
| DPSGD MLP (Abadi et al., 2016) | 1000 | 0.844 | 0.862 | 0.745 | 0.747 | 0.822 |
| DPBoost (Li et al., 2020) | 355 | 0.700 | 0.789 | 0.684 | 0.683 | 0.502 |
| AIM (McKenna et al., 2022) | 486 | 0.830 | 0.725 | 0.502 | 0.723 | 0.662 |
| MST (McKenna et al., 2021) | 517 | 0.740 | 0.827 | 0.550 | 0.703 | 0.783 |
| DP-CTGAN (Fang et al., 2022) | 486 | 0.771 | 0.766 | 0.524 | 0.714 | 0.786 |
| Tab-PE (Tran et al., 2026) | 973 | 0.838 | 0.821 | 0.731 | 0.753 | 0.827 |

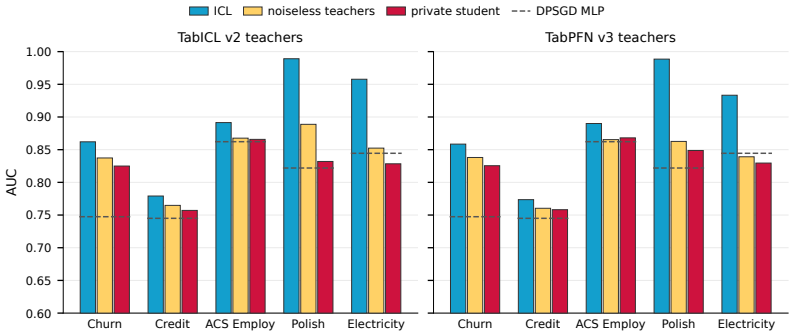


Figure 2: **The data partitioning accounts for most of the utility cost of TabPATE.** We report AUC at each step for every dataset and teacher TFM. The dashed line marks the DPSGD MLP.

**Sources of utility cost.** Figure 2 splits the utility cost of TabPATE into steps: ICL on the full labeled table, noiseless teachers that vote on disjoint partitions of the table, and the private student, which adds privacy noise and the transfer to a student context. Partitioning the table causes most of the loss. The private student is within $0.03$ AUC of the noiseless teachers in nine of ten configurations, and within $0.06$ in the last one (Polish with TabICLv2 teachers). Each teacher context holds only about 50 labeled records, while ICL without privacy keeps improving with larger contexts (Figure 10 in Appx. H.3). This reflects the central trade-off in PATE between using more teachers to improve vote aggregation and retaining enough data for each teacher to make accurate predictions.

**Tighter accounting.** Our tighter data-dependent accounting composes a per-teacher divergence instead of the worst-case per-query bound of PATE. Figure 8 in Appx. H.2 shows the number of queries answered versus teacher agreement. The results demonstrate that our accounting strategy is strictly better than prior ones. Additionally, Figure 9 shows the number of queries answered by each accountant on the real datasets. Recounted for each released configuration at its own teacher count and noise, our accounting allows a median of $2.46\times$ as many queries as standard PATE.

## 7 Conclusions

We showed that records placed in the context of a TFM are vulnerable to membership inference, even when the adversary observes only prediction probabilities. We proposed two methods that adapt a TFM to a private table under DP. TabDPSGD trains a learned context and, optionally, some or all of the TFM weights with DPSGD. TabPATE needs only predicted labels and releases a student context of synthetic records labeled by frozen teachers with GNMax. Our analysis of GNMax in the binary case lets TabPATE answer more queries than standard PATE under the same budget. Both methods outperform every private baseline and set the state of the art for private predictions on tabular data. Private tables can therefore benefit from TFMs, whether the TFM runs locally or behind an API.

## AI USE STATEMENT

In this work, we have used generative AI tools to improve readability, generate figures and tables, and assist in drafting formal proofs. We have not used generative AI tools for any other tasks for which disclosure is required. We take full responsibility for the contents of this manuscript and have verified all AI-generated output, for example, by cross-checking tables against the direct outputs of our execution scripts.

## REFERENCES

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 308–318, 2016.

Michael Aerni, Jie Zhang, and Florian Tramèr. Evaluations of machine learning privacy defenses are misleading. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pp. 1271–1284, 2024.

Borja Balle, Gilles Barthe, Marco Gaboardi, Justin Hsu, and Tetsuya Sato. Hypothesis Testing Interpretations and Renyi Differential Privacy. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108, pp. 2496–2506. PMLR, 2020. URL https://proceedings.mlr.press/v108/balle20a.html.

Ralph Allan Bradley and Milton E. Terry. Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. *Biometrika*, 39:324–345, 1952.

Felix den Breejen, Sangmin Bae, Stephen Cha, and Se-Young Yun. Fine-tuned in-context learning transformers are excellent tabular data classifiers. *arXiv: 2405.13396*, 2024.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901, 2020.

Francesco Capano and Jonas Böhler. Probing Memorization of Tabular In-Context Learning. In *The Second Workshop on the Impact of Memorization on Trustworthy Foundation Models at ICML*, 2026. URL https://openreview.net/forum?id=wet1THUD4q.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy*, pp. 1897–1914, 2022. doi: 10.1109/SP46214.2022.9833649.

Tânia Carvalho and Maxime Cordy. Privacy Vulnerabilities of Attention Layers in Tabular Foundation Models and Protection of High-Risk Queries. *arXiv:2606.26021*, 2026. URL https://arxiv.org/abs/2606.26021.

Karan Chadha, Matthew Jagielski, Nicolas Papernot, Christopher A. Choquette-Choo, and Milad Nasr. Auditing Private Prediction. In *Proceedings of the 41st International Conference on Machine Learning*, 2024. URL https://proceedings.mlr.press/v235/chadha24a.html.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SigKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016.

Christopher A. Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-Only Membership Inference Attacks. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1964–1974, 2021.

Haonan Duan, Adam Dziedzic, Nicolas Papernot, and Franziska Boenisch. Flocks of stochastic parrots: Differentially private prompt learning for large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a. URL https://openreview.net/forum?id=u6Xv3FuF8N.

Haonan Duan, Adam Dziedzic, Mohammad Yaghini, Nicolas Papernot, and Franziska Boenisch. On the privacy risk of in-context learning. In *The 61st Annual Meeting Of The Association For Computational Linguistics*, 2023b.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography*, pp. 265–284, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-32732-5.

Nick Erickson, Lennart Purucker, Andrej Tschalzev, David Holzmüller, Prateek Mutalik Desai, David Salinas, and Frank Hutter. TabArena: A Living Benchmark for Machine Learning on Tabular Data. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-0519.

Mei Ling Fang, Devendra Singh Dhami, and Kristian Kersting. Dp-ctgan: Differentially private medical data generation using ctgans. In *Artificial Intelligence in Medicine*, pp. 178–188, Cham, 2022. Springer International Publishing. ISBN 978-3-031-09342-5.

Vitaly Feldman and Tijana Zrnic. Individual Privacy Accounting via a Rényi Filter. In *Advances in Neural Information Processing Systems*, volume 34, pp. 28080–28091, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/ec7f346604f518906d35ef0492709f78-Paper.pdf.

Benjamin Feuer, Robin T Schirrmeister, Valeriia Cherepanova, Chinmay Hegde, Frank Hutter, Micah Goldblum, Niv Cohen, and Colin White. TuneTables: Context Optimization for Scalable Prior-Data Fitted Networks. *Advances in Neural Information Processing Systems*, 37:83430–83464, 2024.

Anurag Garg, Muhammad Ali, Noah Hollmann, Lennart Purucker, Samuel Müller, and Frank Hutter. Real-TabPFN: Improving Tabular Foundation Models via Continued Pre-training With Real-World Data. In *1st ICML Workshop on Foundation Models for Structured Data*, 2025. URL https://openreview.net/forum?id=BtEiqKsIMw.

Sivakanth Gopi, Yin Tat Lee, and Lukas Wutschitz. Numerical Composition of Differential Privacy. In *Advances in Neural Information Processing Systems*, volume 34, pp. 11631–11642, 2021.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Benjamin Jäger, Dominik Safaric, Simone Alessi, Adrian Hayler, Mihir Manium, Rosen Yu, Felix Jablonski, Shi Bin Hoo, Anurag Garg, Jake Robertson, Magnus Bühler, Vladyslav Moroshan, Lennart Purucker, Clara Cornu, Lilly Charlotte Wehrhahn, Alessandro Bonetto, Bernhard Schölkopf, Sauraj Gambhir, Noah Hollmann, and Frank Hutter. TabPFN-2.5: Advancing the State of the Art in Tabular Foundation Models. *arXiv:2511.08667*, 2026a. URL https://arxiv.org/abs/2511.08667.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Mihir Manium, Shi Bin Hoo, Magnus Bühler, Anurag Garg, Dominik Safaric, Jake Robertson, Benjamin Jäger, Simone Alessi, Adrian Hayler, Vladyslav Moroshan, Lennart Purucker, Philipp Singer, Alan Arazi, Julien Siems, Jan Hendrik Metzen, Georg Grab, Nick Erickson, Siyuan Guo, Eliott Kalfon, Simon Bing, David Salinas, Clara Cornu, Lilly Charlotte Wehrhahn, Diana Kriuchkova, Kursat Kaya, Lydia Sidhoum, Marie Salmon, Jerry Chen, Madelon Hulsebos, Yann LeCun, Samuel Müller, Bernhard Schölkopf, Sauraj Gambhir, Noah Hollmann, and Frank Hutter. TabPFN-3: Technical Report. *arXiv:2605.13986*, 2026b.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=cp5PvcI6w8_.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

David Holzmüller, Léo Grinsztajn, and Ingo Steinwart. Better by default: Strong pre-tuned MLPs and boosted trees on tabular data. In *Advances in Neural Information Processing Systems*, volume 37, pp. 26577–26658, 2024. doi: 10.52202/079017-0837.

Rasa Hosseinzadeh, Alex Labach, Zexin Xue, Shuyi Han, Valentin Thomas, and Anthony L. Caterini. TabDPT-turbo: Efficient in-context learning for tabular prediction. In *2nd ICML Workshop on Foundation Models for Structured Data*, 2026. URL https://openreview.net/forum?id=Y00pwFyrHR.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Weihao Kong and Abhimanyu Das. Introducing TabFM: A zero-shot foundation model for tabular data. Google Research Blog, 2026. URL https://research.google/blog/introducing-tabfm-a-zero-shot-foundation-model-for-tabular-data/.

Qinbin Li, Zhaomin Wu, Zeyi Wen, and Bingsheng He. Privacy-Preserving Gradient Boosting Decision Trees. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 784–791, 2020.

Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=bVuP3ltATMz.

Zinan Lin, Sivakanth Gopi, Janardhan Kulkarni, Harsha Nori, and Sergey Yekhanin. Differentially Private Synthetic Data via Foundation Model APIs 1: Images. In *International Conference on Learning Representations*, volume 2024, pp. 35109–35159, 2024.

Si-Yang Liu, Hao-Run Cai, Qi-Le Zhou, Huai-Hong Yin, Tao Zhou, Jun-Peng Jiang, and Han-Jia Ye. Talent: A tabular analytics and learning toolbox. *Journal of Machine Learning Research*, 26(226): 1–16, 2025.

Siyang Liu and Han-Jia Ye. TabPFN Unleashed: A Scalable and Effective Solution to Tabular Classification Problems. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=5DD3RCcVcT.

Junwei Ma, Valentin Thomas, Guangwei Yu, and Anthony L. Caterini. In-context data distillation with tabPFN. In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2024. URL https://openreview.net/forum?id=EkEFVhe5Gj.

Junwei Ma, Valentin Thomas, Rasa Hosseinzadeh, Alex Labach, Jesse C. Cresswell, Keyvan Golestan, Guangwei Yu, Anthony L. Caterini, and Maks Volkovs. TabDPT: Scaling Tabular Foundation Models on Real Data. In *Advances in Neural Information Processing Systems*, volume 38, pp. 172692–172722, 2025. doi: 10.52202/085713-5748.

Bartłomiej Marek, Lorenzo Rossi, Vincent Hanke, Xun Wang, Michael Backes, Franziska Boenisch, and Adam Dziedzic. Benchmarking Empirical Privacy Protection for Adaptations of Large Language Models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=jY7fAo9rfK.

Ryan McKenna, Gerome Miklau, and Daniel Sheldon. Winning the NIST Contest: A scalable and general approach to differentially private synthetic data. *Journal of Privacy and Confidentiality*, 11 (3), 2021. doi: 10.29012/jpc.778.

Ryan McKenna, Brett Mullins, Daniel Sheldon, and Gerome Miklau. AIM: An Adaptive and Iterative Mechanism for Differentially Private Synthetic Data. *Proceedings of the VLDB Endowment*, 15 (11):2599–2612, 2022. doi: 10.14778/3551793.3551817. URL https://par.nsf.gov/biblio/10359064.

Ryan McKenna, Yangsibo Huang, Amer Sinha, Borja Balle, Zachary Charles, Christopher A. Choquette-Choo, Badih Ghazi, Georgios Kaissis, Ravi Kumar, Ruibo Liu, Da Yu, and Chiyuan Zhang. Scaling Laws for Differentially Private Language Models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=DE6dqmcmQ9.

Ilya Mironov. Rényi Differential Privacy. In *2017 IEEE 30th Computer Security Foundations Symposium*, pp. 263–275, 2017. doi: 10.1109/CSF.2017.11.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=KSugKcbNf9.

Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*, pp. 75–84, 2007.

Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=HkwoSDPgg.

Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Ulfar Erlingsson. Scalable Private Learning with PATE. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rkZB1XbRZ.

Prior Labs. TabPFN Client. https://github.com/PriorLabs/tabpfn-client, 2026. GitHub repository. Accessed 2026-09-20.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. CatBoost: unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICL: A Tabular Foundation Model for In-Context Learning on Large Data. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 50817–50847, 2025.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICLv2: A Better, Faster, Scalable, and Open Tabular Foundation Model. In *Proceedings of the 43rd International Conference on Machine Learning*, Proceedings of Machine Learning Research, 2026.

Rachel Redberg, Yuqing Zhu, and Yu-Xiang Wang. Generalized PTR: User-Friendly Recipes for Data-Adaptive Algorithms with Differential Privacy. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206, pp. 3977–4005. PMLR, 2023. URL https://proceedings.mlr.press/v206/redberg23a.html.

Ivan Rubachev, Akim Kotelnikov, Nikolay Kartashev, and Artem Babenko. On finetuning tabular foundation models. *arXiv:2506.08982*, 2025. URL https://arxiv.org/abs/2506.08982.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership Inference Attacks Against Machine Learning Models . In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18, 2017. doi: 10.1109/SP.2017.41.

Aditya Tanna, Pratinav Seth, Mohamed Bouadi, and Vinay Kumar Sankarapu. Exploring fine-tuning for tabular foundation models. In *Proceedings of the ACM Web Conference 2026*, pp. 8613–8616, 2026.

Valentin Thomas, Junwei Ma, Rasa Hosseinzadeh, Keyvan Golestan, Guangwei Yu, Maksims Volkovs, and Anthony Caterini. Retrieval & fine-tuning for in-context tabular models. *Advances in Neural Information Processing Systems*, 37:108439–108467, 2024.

Florian Tramèr, Reza Shokri, Ayrton San Joaquin, Hoang Le, Matthew Jagielski, Sanghyun Hong, and Nicholas Carlini. Truth Serum: Poisoning Machine Learning Models to Reveal Their Secrets. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2779–2792. Association for Computing Machinery, 2022. ISBN 9781450394505. doi: 10.1145/3548606.3560554.

Toan Tran, Arturs Backurs, Zinan Lin, Victor Reis, Li Xiong, and Sergey Yekhanin. Differentially private synthetic data via APIs 4: Tabular data. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=WB0hLRRlcj.

Tim van Erven and Peter Harremoës. Rényi divergence and Kullback-Leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All You Need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Rui Wen, Zheng Li, Michael Backes, and Yang Zhang. Membership inference attacks against in-context learning. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pp. 3481–3495, 2024. doi: 10.1145/3658644.3690306.

Derek Qiang Xu, F Olcay Cirit, Reza Asadi, Yizhou Sun, and Wei Wang. Mixture of In-Context Prompters for Tabular PFNs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=2fojNANZSv.

# A NOTATION

Table 5 collects the symbols used in the paper.

Table 5: **Notation.** Symbols used throughout the paper, grouped into the shared setup, TabDPSGD, and TabPATE.

| Symbol | Meaning |
|---|---|
| $F$ | Pretrained TFM |
| $W$ | Pretrained TFM weights |
| $M$ | Number of classes of the downstream task |
| $N$ | Number of private records, $N = \lvert D_{\text{priv}} \rvert$ |
| $D_{\text{priv}}$ | Private table $\{(x_i, y_i)\}_{i=1}^{N}$, protected one record at a time |
| $\mathcal{C}$ | Context of real labeled records used by non-private ICL |
| $\theta$ | Trained parameters $(\theta_{\mathcal{C}}, \theta_W)$, what TabDPSGD releases |
| $\theta_{\mathcal{C}}$ | Context parameters, used by TabDPSGD in place of labeled records |
| $\theta_W$ | Weight parameters, TFM weights or an additive update to them |
| $T$, $q$, $\eta$, $B$ | Steps, sampling rate, learning rate, expected batch size |
| $K$ | Number of TabPATE teachers |
| $D_k$ | Partition $k$ of $D_{\text{priv}}$ |
| $F_{D_k}$ | Teacher $k$ reading partition $D_k$ |
| $Q$ | Set of synthetic candidate query records |
| $n(x)$ | Vote histogram over the $M$ classes for candidate query $x$ |
| $\tilde{\mathcal{C}}$ | Released student context |
| $F_{\tilde{\mathcal{C}}}$ | Student reading the student context |
| $\sigma$ | Standard deviation of the Gaussian noise added to each GNMax count |

# B ADDITIONAL DETAILS AND RESULTS ON MEMBERSHIP INFERENCE

## B.1 THREAT MODELS AND MIA SIGNALS

We evaluate four MIAs against tabular ICL, one for each threat model. The two adversary axes are *output access* and *context manipulation*. Output access is the type of access the adversary has to the output of the target TFM. In the *black-box* scenario it observes only the predicted label $\hat{y}_{\mathcal{C}}(x)$, and in the *gray-box* scenario it observes the full output $F(x; \mathcal{C}, W)$. The context manipulation axis determines whether the adversary can inject records into the context of the TFM (active) or not (passive). In the passive setting, we assume that an adversary provides only the query, while the private context is provided by the TFM provider. The provider might additionally allow the client to add its own records to increase prediction accuracy, which motivates the active setting. Combining both axes gives the four threat models of Table 6, and we construct one MIA for each. All four use the same likelihood ratio calibration of Eq. (2) and differ only in the signal $s(x_t)$.

Table 6: MIA methods for each threat model.

| | **Passive** | **Active** |
|---|---|---|
| **Black-box** | Perturbation | Brainwash |
| **Gray-box** | LiRA | Truth Serum |

**Perturbation.** Black-box passive access. For each target $x_t$ we draw $n_{\text{rep}}$ feature-wise Gaussian perturbations $\xi_i$, scaled by each feature's standard deviation, and measure the fraction of perturbed copies whose predicted label is unchanged:

$$s(x_t) = \frac{1}{n_{\text{rep}}} \sum_{i=1}^{n_{\text{rep}}} \mathbf{1}\big[\hat{y}_{\mathcal{C}}(x_t + \xi_i) = \hat{y}_{\mathcal{C}}(x_t)\big].$$

Members tend to be more stable, since their input-output mapping is present in the context. We encode categorical features using integer codes to compute their standard deviation.

**LiRA.** Gray-box passive access. The signal is the logit-scaled confidence on the true label:

$$s(x_t) = \log \frac{F(x_t; \mathcal{C}, W)_{y_t}}{1 - F(x_t; \mathcal{C}, W)_{y_t}}.$$

Context members tend to receive higher confidence.

**Brainwash.** Black-box active access. The adversary observes no probability, so it builds a graded signal out of labels alone. It injects $k$ copies of the target record carrying a wrong label $y' \neq y_t$, giving the context $\mathcal{C}_k = \mathcal{C} \uplus \{(x_t, y')\}^k$, and raises $k$ until the predicted label flips to $y'$, up to a fixed cap $k_{\max}$:

$$s(x_t) = \min\big(\{\, k \in \{1, \ldots, k_{\max}\} : \hat{y}_{\mathcal{C}_k}(x_t) = y' \,\} \cup \{k_{\max}\}\big).$$

A target that already sits in the context resists the wrong label longer, so a member needs more injected copies than a non-member.

**Truth Serum.** Gray-box active access, which combines the capabilities of the two MIAs above. The adversary observes the same logit-scaled confidence as LiRA, but on the poisoned context $\mathcal{C}_k$ for a fixed number of injected copies $k$:

$$s(x_t) = \log \frac{F(x_t; \mathcal{C}_k, W)_{y_t}}{1 - F(x_t; \mathcal{C}_k, W)_{y_t}}.$$

The injected copies pull the prediction of a non-member away from the true label, while a member's own context record holds it up, which separates the IN and OUT distributions of Eq. (2) further than either MIA does alone.

**Comparison to threshold-based MIAs.** LiRA is the strongest of the passive MIAs we tried. We compare it to threshold attacks that score a target with a single statistic of the prediction, namely the loss, the confidence, the prediction gap, the entropy, the correctness, and the modified entropy. LiRA reaches a mean AUC of $0.727$, against $0.554$ for the best threshold attack (Table 7). We therefore use calibrated per-record likelihood ratios for the membership inference results in the main paper.

Table 7: LiRA compared to threshold attacks. Mean over the five datasets.

| Attack | AUC | TPR@1%FPR |
|---|---|---|
| LiRA | 0.727 | 9.9% |
| Loss | 0.554 | 2.6% |
| Prediction gap | 0.554 | 2.5% |
| Confidence | 0.537 | 2.6% |
| Entropy | 0.537 | 2.6% |
| Correctness | 0.534 | 1.1% |
| Modified entropy | 0.474 | 0.4% |

### B.2 Per-Dataset Results

The perturbation attack performs above chance on all datasets, although the signal is modest (Table 8). Vehicle has the highest AUC in this setting.

### B.3 Poisoning Ablation

We observe that poisoning strongly amplifies leakage. Table 9 shows that the average AUC increases with the number of injected records. With $k = 20$ mislabeled copies, the average AUC increases from $0.73$ to $0.96$, and several datasets become almost perfectly separable. This corresponds to a

Table 8: **Label-only perturbation MIA.** Results use LiRA calibration with 128 shadow contexts.

| Dataset | AUC | TPR@1%FPR | TPR@5%FPR |
|---|---|---|---|
| German Credit | 0.559 | 1.7% | 6.0% |
| Blood Transfusion | 0.509 | 1.0% | 4.7% |
| Diabetes | 0.533 | 1.5% | 5.6% |
| Vehicle | 0.635 | 1.3% | 7.2% |
| KC1 | 0.518 | 1.5% | 5.5% |
| Mean | 0.551 | 1.4% | 5.8% |

small fraction of the context in our experiments, so active manipulation reveals membership even where passive MIAs are weaker.

Table 9: **LiRA with context poisoning.** AUC and TPR at $1\%$ FPR for different numbers $k$ of injected records.

| | $k=0$ | | $k=5$ | | $k=10$ | | $k=20$ | |
|---|---|---|---|---|---|---|---|---|
| **Dataset** | AUC | TPR | AUC | TPR | AUC | TPR | AUC | TPR |
| KC1 | 0.68 | 8.5% | 0.83 | 12.7% | 0.92 | 39.2% | 0.96 | 67.0% |
| Vehicle | 0.91 | 20.0% | 1.00 | 99.7% | 1.00 | 100% | 1.00 | 96.8% |
| Diabetes | 0.68 | 6.7% | 0.79 | 9.1% | 0.97 | 55.5% | 1.00 | 98.2% |
| German Credit | 0.76 | 10.9% | 0.96 | 61.4% | 0.99 | 85.5% | 1.00 | 99.8% |
| Blood Transfusion | 0.60 | 3.6% | 0.64 | 0.2% | 0.71 | 0.0% | 0.83 | 15.0% |
| Average | 0.73 | 9.9% | 0.84 | 36.6% | 0.92 | 56.0% | 0.96 | 75.4% |

### B.4 Perturbation Ablations

The label-only perturbation MIA is robust to its two hyperparameters. Changing the number of perturbations $n_{\text{rep}}$ (Table 10) or the perturbation scale $\sigma$ (Table 11) moves the AUC on German Credit only between $0.524$ with 100 perturbations and $0.546$ at scale $0.50$. The ablation therefore rules out a poorly tuned perturbation setting as the cause of its modest success.

Table 10: **Perturbation-count ablation on German Credit.** We use 64 shadow contexts.

| Perturbations $n_{\text{rep}}$ | AUC | TPR@1%FPR |
|---|---|---|
| 10 | 0.526 | 2.1% |
| 25 | 0.531 | 2.3% |
| 50 | 0.532 | 1.1% |
| 100 | 0.524 | 1.1% |

All four attacks use the same number of shadow contexts, so their cost is set by the number of queries per target. LiRA and Truth Serum need one query per target after calibration, the perturbation attack needs $n_{\text{rep}}$ queries, and Brainwash needs up to $k_{\max}$ queries.

## C Additional Details on TabDPSGD

### C.1 TabDPSGD Algorithm

Algorithm 1 states the TabDPSGD training procedure. It applies DPSGD (Abadi et al., 2016) to the selected parameters $\theta$. The context parameters $\theta_C$ replace the labeled records a TFM would otherwise condition on, so each private record enters only as a query. The TFM computes each query separately, so a query conditions on the learned context but not on the other queries in its batch. Each per-record gradient therefore depends only on its own record and $\theta$, and no gradient couples two private records. This is the condition under which Algorithm 1 satisfies DP.

Table 11: **Perturbation-scale ablation on German Credit.** We use $n_{\text{rep}} = 50$ and 64 shadow contexts.

| Scale $\sigma$ | AUC | TPR@1%FPR |
|---|---|---|
| 0.05 | 0.530 | 1.7% |
| 0.10 | 0.532 | 1.1% |
| 0.20 | 0.536 | 1.3% |
| 0.50 | 0.546 | 1.8% |

**Algorithm 1** TabDPSGD: DPSGD training of the parameters $\theta = (\theta_{\mathcal{C}}, \theta_W)$

**Require:** $D_{\text{priv}} = \{(x_i, y_i)\}_{i=1}^N$, pretrained TFM $F$, initial parameters $\theta^{(0)} = (\theta_{\mathcal{C}}^{(0)}, \theta_W^{(0)})$, clip norm $C$, noise multiplier $\sigma_{\text{dp}}$, sampling rate $q$, expected batch size $B$, steps $T$
**Ensure:** $\theta^{(T)}$
1: **for** $t = 1, \dots, T$ **do**
2: sample $\mathcal{B}_t \subseteq D_{\text{priv}}$, each record independently with probability $q$
3: $\bar{g}_i \leftarrow \text{clip}_C\big(\nabla_\theta \, \ell\big(F(x_i; \theta^{(t-1)}), y_i\big)\big)$ for each $i \in \mathcal{B}_t$
4: $\tilde{g}_t \leftarrow \big(\sum_{i \in \mathcal{B}_t} \bar{g}_i + \mathcal{N}(0, \sigma_{\text{dp}}^2 C^2 I)\big)/B$
5: $\theta^{(t)} \leftarrow \text{Optimizer}(\theta^{(t-1)}, \tilde{g}_t)$
6: **end for**
7: **return** $\theta^{(T)}$

# D Additional Details on TabPATE

## D.1 Query Generation

**Data-independent queries.** We draw a scrambled Sobol sequence and map each numerical column to a Gaussian with a fixed mean and standard deviation, so that $95\%$ of the draws fall inside the interval $[-1, 1]$ fixed by the released scaling rule of Eq. (17). The remaining draws are clipped to this interval. Each categorical column is spread evenly over its categories. The scrambling improves coverage compared with independent random draws, while each column remains independent and no private record determines a candidate. Generating these candidates therefore spends no privacy budget.

**DP marginal queries.** This strategy estimates the mean and standard deviation of each numerical column privately, from a histogram released with Gaussian noise of scale $\sigma_c$. Candidates keep the Gaussian form of the data-independent queries, but with these estimates in place of the fixed mean and standard deviation. Categorical columns are drawn uniformly as before and are not estimated. Algorithm 2 gives the full procedure.

Let $d$ be the number of numerical columns, which all lie in $[-1, 1]$ after the released scaling rule of Eq. (17). We split $[-1, 1]$ into $B = 256$ bins of equal width with midpoints $m_1, \dots, m_B$, and let $c_{jb}$ count the records whose value in column $j$ falls in bin $b$. A record adds one count to each column, so adding or removing it changes the $d \times B$ counts by at most $\sqrt{d}$ in $\ell_2$ norm. The Gaussian mechanism releases the noisy counts $\tilde{c}_{jb} = \max(0, c_{jb} + Z_{jb})$ with $Z_{jb} \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_c^2)$.

We then place every record at the midpoint of its bin and compute the mean and standard deviation of each column from the noisy counts,

$$\hat{\mu}_j = \frac{\sum_{b=1}^B \tilde{c}_{jb} m_b}{\sum_{b=1}^B \tilde{c}_{jb}}, \qquad \hat{s}_j = \sqrt{\frac{\sum_{b=1}^B \tilde{c}_{jb} \, (m_b - \hat{\mu}_j)^2}{\sum_{b=1}^B \tilde{c}_{jb}}}. \tag{3}$$

Each numerical value of a candidate is then drawn as $\text{clip}(\hat{\mu}_j + \hat{s}_j \xi, -1, 1)$ with $\xi \sim \mathcal{N}(0, 1)$, independently across columns. Everything after the noisy release is post-processing and spends no further budget.

**Algorithm 2** DP marginal queries, $\mathrm{MarginalQueries}(D_{\mathrm{priv}}, \sigma_c, L, B)$

**Require:** private table $D_{\mathrm{priv}}$ with $d$ numerical columns in $[-1, 1]$, noise scale $\sigma_c$, number of candidates $L$, number of bins $B$
**Ensure:** candidate queries $Q = (q_i)_{i=1}^{L}$
  split $[-1, 1]$ into $B$ bins of equal width with midpoints $m_1, \ldots, m_B$
  **for** each numerical column $j$ **do**
    $c_{jb} \leftarrow \big|\{z \in D_{\mathrm{priv}} : z_j \text{ falls in bin } b\}\big|$ for $b = 1, \ldots, B$
    $\tilde{c}_{jb} \leftarrow \max(0, c_{jb} + Z_{jb})$, where $Z_{jb} \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_c^2)$, for $b = 1, \ldots, B$
    compute $\hat{\mu}_j$ and $\hat{s}_j$ from $(\tilde{c}_{jb})_{b=1}^{B}$ by Eq. (3)
  **end for**
  **for** $i = 1, \ldots, L$ **do**
    **for** each feature column $j$ **do**
      **if** column $j$ is categorical **then**
        $q_{ij} \sim U\{0, \ldots, V_j - 1\}$
      **else**
        $q_{ij} \leftarrow \mathrm{clip}(\hat{\mu}_j + \hat{s}_j \xi_{ij}, -1, 1)$, where $\xi_{ij} \sim \mathcal{N}(0, 1)$
      **end if**
    **end for**
  **end for**
  **return** $Q = (q_i)_{i=1}^{L}$

We select the noise scale $\sigma_c$. The histogram release is then $(\alpha, \alpha\rho_c)$-RDP at every order $\alpha$, with $\rho_c = d/(2\sigma_c^2)$ (Mironov, 2017). The accountant adds $\alpha\rho_c$ to the preprocessing curve of Eq. (18), and GNMax spends the rest of the budget on labels.

### D.2 TabPATE Algorithm

TabPATE (Algorithm 3) labels each candidate query with GNMax (Papernot et al., 2018), which adds independent $\mathcal{N}(0, \sigma^2)$ noise to each class count and returns the noisy argmax. $\mathrm{GenerateQueries}$ draws the data-independent or DP marginal queries, and $\mathrm{TrainStudent}$ turns the released student context into a trained student without further access to $D_{\mathrm{priv}}$, so it spends no further budget.

**Algorithm 3** TabPATE

**Require:** private table $D_{\mathrm{priv}}$, frozen TFM $F$ with weights $W$, number of candidates $L$, number of teachers $K$, noise scale $\sigma$, fixed public partition function $\pi$
**Ensure:** released student context $\tilde{\mathcal{C}}$ and student $S$
1: $Q = (q_i)_{i=1}^{L} \leftarrow \mathrm{GenerateQueries}(D_{\mathrm{priv}}, L)$
2: $D_j \leftarrow \{(x, y) \in D_{\mathrm{priv}} : \pi(x, y) = j\}$ for $j = 1, \ldots, K$
3: set $\tilde{\mathcal{C}} \leftarrow \varnothing$
4: **for** $i = 1, \ldots, L$ **do**
5:  $v_{ij} \leftarrow \mathrm{argmax}_{b \in \{0,1\}} F(q_i; D_j, W)_b$ for each nonempty partition $D_j$
6:  $n_b(q_i) \leftarrow \sum_{\{j | D_j \neq \varnothing\}} \mathbb{1}_{[v_{ij} = b]}$ for $b \in \{0, 1\}$
7:  $\hat{y}_i \leftarrow \mathrm{argmax}_{b \in \{0,1\}} \big(n_b(q_i) + Z_{ib}\big)$, where $Z_{i0}, Z_{i1} \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$
8:  $\tilde{\mathcal{C}} \leftarrow \tilde{\mathcal{C}} \cup \{(q_i, \hat{y}_i)\}$
9: **end for**
10: $S \leftarrow \mathrm{TrainStudent}(\tilde{\mathcal{C}})$
11: **return** $\tilde{\mathcal{C}}$ and $S$

# E PRIVACY ANALYSIS OF TABPATE

## E.1 TIGHTER PRIVACY ANALYSIS OF GNMAX IN THE BINARY CASE

The standard PATE accountant bounds the probability that GNMax does not return the plurality class, converts this bound into a per-query RDP cost, and sums these costs across queries (Papernot et al., 2018). This loses tightness in binary classification because the released label has an exact Bernoulli law and one differing record can affect only one fixed teacher across all queries. We therefore compute the exact directed Bernoulli divergence and compose costs separately for each teacher before maximizing over teachers.

Here, we derive a data-dependent upper bound on the neighboring Rényi divergence of binary GNMax. Following Papernot et al. (2018), we analyze one aggregation, compose across queries, and distinguish the pointwise bound from the mechanism's unconditional RDP guarantee.

Exact noisy-argmax divergence also appears in work on auditing private prediction. Chadha et al. (2024) use it to compare empirical privacy lower bounds with analytical guarantees for private prediction mechanisms. Our goal is tighter data-dependent accounting rather than auditing. We specialize the divergence to binary classification, where the two output probabilities are available in closed form and require no numerical integration.

Throughout this section, $D$ denotes the private table at which a data-dependent quantity is evaluated. We omit the argument $D$. When a neighboring table $D'$ is fixed, primes denote evaluation at $D'$. We retain explicit arguments for any other table.

Let the two vote counts for a fixed query be $n_0$ and $n_1$, where $0 \le n_0 + n_1 \le K$, and define the signed vote gap

$$d = n_1 - n_0.$$

One private record can affect one teacher. A change to the label returned by that teacher changes the vote gap by two, while a teacher that starts or ceases to vote changes the gap by one. Thus the vote gaps of neighboring tables differ by at most two and remain in $\{-K, \dots, K\}$.

We use RDP to compare the output laws induced by neighboring private tables and to compose multiple label releases. For distributions $P$ and $Q$ on a finite output space $\mathcal{Y}$ and an order $\alpha > 1$, the Rényi divergence is

$$D_\alpha(P\|Q) \coloneqq \frac{1}{\alpha - 1} \log \left( \sum_{y \in \mathcal{Y}} P(y)^\alpha Q(y)^{1-\alpha} \right). \tag{4}$$

**Definition 1 (Rényi differential privacy, Mironov, 2017)** *(Mironov, 2017) A randomized mechanism $\mathcal{M}$ satisfies $(\alpha, \rho_\alpha)$-RDP if, for every ordered pair of neighboring private tables $D \sim D'$,*

$$D_\alpha(\mathcal{M}(D)\|\mathcal{M}(D')) \le \rho_\alpha. \tag{5}$$

Because the neighbor relation is symmetric, Eq. (5) must hold in both directions.

Following standard PATE analysis, RDP costs compose additively at a common order (Papernot et al., 2018; Mironov, 2017):

**Theorem 1 (Mironov, 2017, Prop. 1)** *Let $\mathcal{M} = (\mathcal{M}_1, \dots, \mathcal{M}_L)$ be a sequence of adaptive mechanisms. If $\mathcal{M}_i$ is $(\alpha, \rho_i)$-RDP conditional on every output of the preceding mechanisms, then $\mathcal{M}$ is $(\alpha, \sum_{i=1}^{L} \rho_i)$-RDP.*

For independent queries on a fixed candidate sequence, Rényi divergence adds across queries by product distribution additivity (Mironov, 2017). Theorem 1 extends this property to adaptive sequences.

The next theorem states the tighter conversion from RDP to approximate DP used in our experiments (Balle et al., 2020). We write $[x]_+ = \max\{x, 0\}$ for the positive part of $x$.

**Theorem 2 (Balle et al., 2020, Thm. 20)** *If a mechanism is $(\alpha, \rho_\alpha)$-RDP, then for every $\delta \in (0,1)$ it is $(\varepsilon_\alpha^{\mathrm{DP}}(\delta), \delta)$-DP, where*

$$\varepsilon_\alpha^{\mathrm{DP}}(\delta) = \left[\rho_\alpha + \log\left(1 - \frac{1}{\alpha}\right) - \frac{\log\delta + \log\alpha}{\alpha - 1}\right]_+ . \tag{6}$$

One binary GNMax query returns a Bernoulli random variable. The following specialization of Eq. (4) gives its divergence in closed form.

**Definition 2 (Bernoulli Rényi divergence)** *For Bernoulli distributions with success probabilities $p$ and $q$, define the exact order-$\alpha$ Rényi divergence*

$$h_\alpha(p,q) \coloneqq \frac{1}{\alpha - 1}\log\left(p^\alpha q^{1-\alpha} + (1-p)^\alpha (1-q)^{1-\alpha}\right). \tag{7}$$

Definition 2 follows by substituting the two Bernoulli probability masses into Eq. (4). The same expression appears in the RDP analysis of randomized response (Mironov, 2017).

The next lemma gives the output distribution of one binary GNMax query and its exact divergence for a neighboring vote histogram.

**Lemma 1 (Output law of binary GNMax)** *Let $d_i = n_1(q_i) - n_0(q_i)$. Define*

$$p_\sigma(d) \coloneqq \Phi\left(\frac{d}{\sqrt{2}\sigma}\right).$$

*Here, $\Phi$ denotes the standard Gaussian cumulative distribution function. Then the released label is distributed as $\mathrm{Bernoulli}(p_\sigma(d_i))$. If a neighboring private table changes the vote gap by $\Delta$, its output law is $\mathrm{Bernoulli}(p_\sigma(d_i + \Delta))$, and the exact directed order-$\alpha$ divergence is*

$$h_\alpha(p_\sigma(d_i), p_\sigma(d_i + \Delta)). \tag{8}$$

*Proof:* Binary GNMax returns class 1 exactly when $Z_{i0} - Z_{i1} \le d_i$. Since $Z_{i0} - Z_{i1} \sim \mathcal{N}(0, 2\sigma^2)$, its success probability is $\Phi(d_i/(\sqrt{2}\sigma)) = p_\sigma(d_i)$. A neighbor changes $d_i$ by $\Delta$ while the noise is identical, so the output law is $\mathrm{Bernoulli}(p_\sigma(d_i + \Delta))$. Substituting the two Bernoulli probabilities into Definition 2 gives Eq. (8). □

The following lemma supplies a uniform privacy guarantee that does not depend on the realized vote gap. This guarantee also bounds every directed charge used below.

**Lemma 2 (Data-independent binary GNMax guarantee)** *For every order $\alpha > 1$, binary GNMax is $(\alpha, \alpha/\sigma^2)$-RDP under add/remove adjacency. Equivalently, its exact Bernoulli divergence is bounded uniformly over neighboring vote histograms by*

$$\max_{\substack{d, d+\Delta \in \{-K,\dots,K\} \\ \Delta \in \{-2,-1,+1,+2\}}} h_\alpha(p_\sigma(d), p_\sigma(d+\Delta)) \le \frac{\alpha}{\sigma^2}.$$

*Consequently, binary GNMax on a fixed sequence of $L$ candidate queries is $(\alpha, L\alpha/\sigma^2)$-RDP (Papernot et al., 2018).*

*Proof:* By Lemma 1, the label is the sign of $d + (Z_{i1} - Z_{i0})$, where $Z_{i1} - Z_{i0} \sim \mathcal{N}(0, 2\sigma^2)$ and the gap $d$ changes by at most two between neighbors, so the Gaussian mechanism bound $\alpha \cdot 2^2/(2 \cdot 2\sigma^2) = \alpha/\sigma^2$ (Mironov, 2017) and post-processing give the claim, and Theorem 1 gives the cost over $L$ queries. The bound is not tight, because taking the sign discards information, so the finite maximum in the display is strictly smaller. □

We now make the partitioning and the case of an empty partition explicit. A private table is a multiset of labeled records $(x, y)$. Fix before observing the table a deterministic mapping function

$$\pi(x,y) \in \{1,\dots,K\}, \qquad D_j(T) = \{(x,y) \in T : \pi(x,y) = j\}, \tag{9}$$

where $T$ is any private table and $\uplus$ denotes multiset union. Write $D_j = D_j(D)$ and, for a neighboring table $D'$, write $D'_j = D_j(D')$. The privacy proof requires only that $\pi$ is fixed and evaluated pointwise.

The values of $\pi$ need not be random or secret. In particular, if $D' = D \uplus \{(x^\star, y^\star)\}$, then $D'_j = D_j$ for every $j \neq \pi(x^\star, y^\star)$ and only $D'_{\pi(x^\star,y^\star)}$ gains a record.

We assume deterministic teacher inference, including deterministic tie breaking, once $D_j$, $Q$, and $W$ are fixed. Let $v_{ij}$ be teacher $j$'s prediction for query $q_i$, and let $v'_{ij}$ denote its prediction under $D'$. We can equivalently decompose the previously defined gap $d_i = n_1(q_i) - n_0(q_i)$ into signed teacher contributions. Let $a_{ij}$ be $-1$ for a vote for class 0, $+1$ for a vote for class 1, and 0 when the partition is empty. Then

$$a_{ij} \coloneqq \begin{cases} 2v_{ij} - 1, & D_j \neq \varnothing, \\ 0, & D_j = \varnothing, \end{cases} \qquad d_i = \sum_{j=1}^{K} a_{ij}.$$

One record addition or removal changes at most one teacher state, so every gap changes by at most two.

When one record in partition $j$ is added or removed, the vote gap $d_i$ of query $i$ changes by an amount in the following set:

$$\mathcal{A}_{ij} \coloneqq \begin{cases} \{-1, 0, +1\}, & |D_j| = 0, \\ \{0, -a_{ij}, -2a_{ij}\}, & |D_j| = 1, \\ \{0, -2a_{ij}\}, & |D_j| \geq 2. \end{cases} \tag{10}$$

The first line covers an abstaining teacher that begins to vote. In the second line, removing the sole record causes abstention and adding a record may preserve or flip the vote. In the last line the partition remains nonempty, so its vote is preserved or flipped. Including zero in the first line is harmless and keeps every set nonempty.

For $o \in \{\rightarrow, \leftarrow\}$, define the two directed charges

$$c_{\alpha,\rightarrow}(d, t) \coloneqq h_\alpha(p_\sigma(d), p_\sigma(d + t)), \qquad c_{\alpha,\leftarrow}(d, t) \coloneqq h_\alpha(p_\sigma(d + t), p_\sigma(d)).$$

**Definition 3 (Data-dependent binary GNMax bound)** *For a fixed sequence of $L$ candidate queries, let*

$$f_\alpha(D) \coloneqq \max_{1 \leq j \leq K} \max_{o \in \{\rightarrow,\leftarrow\}} \sum_{i=1}^{L} \max_{t \in \mathcal{A}_{ij}(D)} c_{\alpha,o}(d_i(D), t). \tag{11}$$

*This deterministic scalar is defined for every table, including tables whose partitions are empty or hold a single record.*

By Lemma 2, $0 \leq f_\alpha(D) \leq L\alpha/\sigma^2$.

The next theorem shows that $f_\alpha(D)$ bounds the divergence in both directions for every neighboring table, including tables with empty partitions.

**Theorem 3 (Pointwise bound on the neighboring divergence)** *For every neighboring pair $D \sim D'$, binary GNMax on the fixed query sequence satisfies*

$$D_\alpha(\mathcal{M}(D) \| \mathcal{M}(D')) \leq f_\alpha(D), \qquad D_\alpha(\mathcal{M}(D') \| \mathcal{M}(D)) \leq f_\alpha(D).$$

*Proof:* Because the mapping function $\pi$ is fixed and evaluated on each record separately, the two tables differ in at most one partition, say $j^\star$. Hence all other teacher states agree. Put $t_i = d'_i - d_i$. The cases in Eq. (10) give $t_i \in \mathcal{A}_{ij^\star}$ for every query.

The independent Gaussian noises make the output law a product of Bernoulli distributions. Additivity of Rényi divergence therefore gives

$$D_\alpha(\mathcal{M}(D) \| \mathcal{M}(D')) = \sum_{i=1}^{L} c_{\alpha,\rightarrow}(d_i, t_i) \leq f_\alpha(D).$$

Using $c_{\alpha,\leftarrow}$ gives the reverse inequality. The maxima in Eq. (11) cover the affected teacher, both divergence directions, and every possible change in each query's vote gap caused by the differing record. The same teacher $j$ and direction $o$ are used across the entire query sequence, whereas the inner maximum is taken separately for each query. It may therefore combine changes in the vote gap

that no single neighboring record realizes simultaneously. This relaxation preserves a valid upper bound while avoiding a coupled optimization over neighboring records. □

This theorem gives a pointwise, data-dependent upper bound. It is not by itself a global RDP parameter because $f_\alpha(D)$ depends on private votes. The next subsections show how to make this a global RDP bound.

### E.2 Private Release of the Data-Dependent Bound

Following Nissim et al. (2007) and Papernot et al. (2018), we privatize $f_\alpha(D)$ with Gaussian noise calibrated to a smooth upper bound on its local sensitivity.

**Definition 4 (Smooth sensitivity)** *Let* $\mathrm{dist}(D, D')$ *be the minimum number of record additions or removals transforming* $D$ *into* $D'$*. For a scalar function* $f$*, define*

$$\mathrm{LS}_f(D) = \max_{D' \,:\, \mathrm{dist}(D,D')=1} |f(D) - f(D')|$$

*and*

$$\mathrm{SS}_{\xi,f}(D) = \max_{s \in \mathbb{Z}_{\geq 0}} e^{-\xi s} \max_{D' \,:\, \mathrm{dist}(D,D') \leq s} \mathrm{LS}_f(D').$$

The set $\mathcal{A}_{ij}$ depends on $D$ only through the truncated partition size $m_j \coloneqq \min\{|D_j|, 2\}$ and the signed vote $a_{ij}$, so we write $\mathcal{A}_{ij} = \mathcal{A}(m_j, a_{ij})$, where $\mathcal{A}(m, a)$ is given by the three cases of Eq. (10) and $a = 0$ exactly when $m = 0$. One record addition or removal acts on the pair $(m_j, a_{ij})$ of a teacher in one of two ways. An unaffected teacher keeps its pair while the gap moves by some $u \in \{-2, \ldots, 2\}$. The affected teacher moves from $(m, a)$ to $(m', a')$ with $|m - m'| = 1$ or $m = m' = 2$, and the gap moves by $u = a' - a$. We define the per-query variation by comparing the inner maxima of Eq. (11) before and after such a change:

$$r_\alpha(d) \coloneqq \max \left| \max_{t \in \mathcal{A}(m,a)} c_{\alpha,o}(d, t) - \max_{t' \in \mathcal{A}(m',a')} c_{\alpha,o}(d + u, t') \right|, \tag{12}$$

where the outer maximum ranges over both directions $o$ and over every tuple $(m, a, u, m', a')$ that one record change can produce as described above. This is a finite maximization that depends only on the observed gap and public parameters. It does not compare unrelated individual charges from the two change sets. Moreover, $0 \leq r_\alpha(d) \leq \alpha/\sigma^2$ because every charge lies in this interval.

The next lemma lifts this per-query charge variation to the local sensitivity of the full data-dependent bound.

**Lemma 3 (Local sensitivity bound)** *For every private table* $D$*,*

$$\mathrm{LS}_{f_\alpha}(D) \leq \sum_{i=1}^{L} r_\alpha(d_i)\,.$$

*Proof:* For a table $T$, write $f_{j,o}(T)$ for the sum in Eq. (11) at fixed teacher $j$ and direction $o$, so that $f_\alpha(T) = \max_{j,o} f_{j,o}(T)$. Fix $D \sim D'$ and put $u_i = d_i(D') - d_i(D)$. The mapping function $\pi$ is evaluated on each record separately, so at most one teacher, say $j^\star$, is affected. If $j \neq j^\star$, the pair $(m_j, a_{ij})$ is the same at $D$ and $D'$, and $u_i \in \{-2, \ldots, 2\}$. If $j = j^\star$, the pair moves from $(m, a)$ to $(m', a')$ with $|m - m'| = 1$ or $m = m' = 2$, and only this teacher's contribution changes, so $u_i = a' - a$. In both cases the tuple is one of those in the maximum defining $r_\alpha$, so the $i$-th summands of $f_{j,o}(D)$ and $f_{j,o}(D')$ differ by at most $r_\alpha(d_i(D))$. Summing over queries yields $|f_{j,o}(D) - f_{j,o}(D')| \leq \sum_i r_\alpha(d_i(D))$. Applying $|\max_x A_x - \max_x B_x| \leq \max_x |A_x - B_x|$ over $(j, o)$ proves the claim. □

The lemma reduces the table-level local sensitivity to a sum of per-query bounds over the vote gaps.

For integer $s \geq 0$, define the vote-gap window

$$W_s(d) \coloneqq \{\bar{d} \in \{-K, \ldots, K\} : |\bar{d} - d| \leq 2s\}$$

and

$$A_s(D) \coloneqq \sum_{i=1}^{L} \max_{d \in W_s(d_i)} r_\alpha(d).$$

The window is a convenient superset of the gaps reachable in $s$ edits. It intentionally ignores parity and constraints across queries. Define

$$S_{\xi,\alpha}(D) \coloneqq \max_{0\le s\le K} e^{-\xi s} A_s(D).$$

The next theorem shows that $S_{\xi,\alpha}(D)$ upper-bounds the smooth sensitivity of $f_\alpha$ and varies by at most a factor of $e^\xi$ between neighboring tables.

**Theorem 4 (Smooth upper bound for the binary GNMax cost)** *For every table $D$,*

$$\mathrm{SS}_{\xi,f_\alpha}(D) \le S_{\xi,\alpha}(D).$$

*Moreover, for every neighboring pair $D \sim D'$,*

$$S_{\xi,\alpha}(D) \le e^\xi S_{\xi,\alpha}(D').$$

*Proof:* If $\mathrm{dist}(D, D') \le s$, then each edit changes a gap by at most two, so $d'_i \in W_s(d_i)$. The lemma gives

$$\max_{D' \,:\, \mathrm{dist}(D,D')\le s} \mathrm{LS}_{f_\alpha}(D') \le A_s(D).$$

For $s \ge K$, every $W_s(d)$ is the full gap domain, so $A_s(D) = A_K(D)$ while $e^{-\xi s}$ decreases. Taking the discounted maximum proves $\mathrm{SS}_{\xi,f_\alpha}(D) \le S_{\xi,\alpha}(D)$.

Now let $D \sim D'$. Since $|d_i - d'_i| \le 2$, $W_s(d_i) \subseteq W_{s+1}(d'_i)$. Hence, for $s < K$,

$$e^{-\xi s} A_s(D) \le e^\xi e^{-\xi(s+1)} A_{s+1}(D') \le e^\xi S_{\xi,\alpha}(D').$$

At $s = K$, the windows cover the full domain and $A_K(D) = A_K(D')$, so the same conclusion holds. Maximizing over $s$ proves smoothness. □

Consequently, $S_{\xi,\alpha}$, a $\xi$-smooth upper bound rather than the exact smooth sensitivity, can calibrate the smooth sensitivity mechanism used next.

### E.2.1 Gaussian Noise with Smooth Sensitivity

**Definition 5 (Gaussian noise with smooth sensitivity (GNSS))** *Following Papernot et al. (2018, Definition 22), for a real-valued function $f$ and a function $S_\xi$ satisfying $S_\xi(D) \ge \mathrm{LS}_f(D)$ for all $D$ and $S_\xi(D) \le e^\xi S_\xi(D')$ for all $D \sim D'$, the $(\xi, \sigma_{\mathrm{SS}})$-GNSS mechanism releases*

$$\widetilde{f}(D) = f(D) + S_\xi(D) Z, \qquad Z \sim \mathcal{N}(0, \sigma^2_{\mathrm{SS}}).$$

The following theorem gives the RDP cost of releasing a scalar with Gaussian noise calibrated to a smooth upper bound. Papernot et al. (2018, Theorem 23) state this expression for all $1 < \alpha < 1/(2\xi)$. The bound on the first term in the cited proof yields the stated expression only when $1 - 2\alpha\xi \ge 1/2$, so we state the theorem under the stricter condition $\alpha\xi \le 1/4$ and give a direct proof.

**Theorem 5 (RDP of GNSS, after Papernot et al., 2018, Thm. 23)** *Let $1 < \alpha \le 1/(4\xi)$. The $(\xi, \sigma_{\mathrm{SS}})$-GNSS mechanism in Definition 5 satisfies $(\alpha, \rho^{\mathrm{SS}}_\alpha)$-RDP with*

$$\rho^{\mathrm{SS}}_\alpha = \frac{\alpha e^{2\xi}}{\sigma^2_{\mathrm{SS}}} + \frac{\xi\alpha - \frac{1}{2}\log(1 - 2\alpha\xi)}{\alpha - 1}. \tag{13}$$

*Proof:* Fix neighboring tables $D \sim D'$ and write $a = S_\xi(D)$, $b = S_\xi(D')$, and $\Delta = f(D) - f(D')$. If either scale vanishes, smoothness forces both to vanish and the local sensitivity bound gives $\Delta = 0$, so the output laws coincide. Otherwise, $a, b > 0$, $e^{-\xi} \le b/a \le e^\xi$, and $|\Delta| \le a$. Set

$$v = \alpha b^2 - (\alpha - 1)a^2.$$

The restriction $\alpha\xi \le 1/4$ gives

$$\frac{v}{a^2} = 1 + \alpha\big((b/a)^2 - 1\big) \ge 1 + \alpha(e^{-2\xi} - 1) \ge 1 - 2\alpha\xi \ge \frac{1}{2}.$$

Using the closed-form Rényi divergence between Gaussian distributions (van Erven & Harremoës, 2014), the two output laws $P = \mathcal{N}(f(D), \sigma_{\mathrm{SS}}^2 a^2)$ and $Q = \mathcal{N}(f(D'), \sigma_{\mathrm{SS}}^2 b^2)$ satisfy

$$\begin{aligned} D_\alpha(P\|Q) &= \frac{\alpha\Delta^2}{2\sigma_{\mathrm{SS}}^2 v} + \frac{\alpha\log(b/a) - \frac{1}{2}\log(v/a^2)}{\alpha-1} \\ &\le \frac{\alpha}{\sigma_{\mathrm{SS}}^2} + \frac{\alpha\xi - \frac{1}{2}\log(1-2\alpha\xi)}{\alpha-1} \\ &\le \rho_\alpha^{\mathrm{SS}}. \end{aligned}$$

The same argument applies to every ordered neighboring pair, proving the RDP guarantee. □

The order restriction is equivalently $0 < \xi \le 1/(4\alpha)$. For the single order fixed before release, this restriction can make the discount $e^{-\xi s}$ decay slowly. In particular, $A_K(D) = L\max_{d\in\{-K,\dots,K\}} r_\alpha(d)$ is the boundary term over the full domain, so it can materially affect the maximum defining $S_{\xi,\alpha}(D)$. Whether it dominates depends on the intermediate values $A_s(D)$ and must be checked for the chosen parameters.

Instantiating Definition 5 with $f_\alpha(D)$ and $S_{\xi,\alpha}(D)$ produces the private estimate $\widetilde{f}_\alpha(D) = f_\alpha(D) + S_{\xi,\alpha}(D)Z$. The formal PATE construction handles order selection by fixing a single order before observing the votes and releasing one total sanitized by GNSS at that order (Papernot et al., 2018). After this public choice, no minimization over private diagnostics is performed. Because $\mathbb{E}[Z] = 0$, this estimate can fall below $f_\alpha(D)$ and is not a privacy certificate for the labels. Let $z_{1-\beta}$ denote the $(1-\beta)$-quantile of a standard Gaussian. Adding $z_{1-\beta}\sigma_{\mathrm{SS}}S_{\xi,\alpha}(D)$ gives only a probabilistic upper confidence bound. Using it to control a release requires a valid private filter or odometer and must account for the failure probability $\beta$. For any order $1 < \alpha \le 1/(4\xi)$, applying the RDP-to-DP expression to the estimated data-dependent total gives the private diagnostic

$$\widehat{\varepsilon}_\alpha(\delta) = \left[\widetilde{f}_\alpha(D) + \rho_\alpha^{\mathrm{SS}} + \log\left(1-\frac{1}{\alpha}\right) - \frac{\log\delta + \log\alpha}{\alpha-1}\right]_+, \tag{14}$$

where $\rho_\alpha^{\mathrm{SS}}$ is given by Eq. (13). The parameters $\alpha$, $\xi$, $\sigma_{\mathrm{SS}}$, and $L$ are fixed before observing private votes.

Releasing diagnostics at several orders and then taking their minimum is possible, but the GNSS releases must first be composed. In particular, if orders $\alpha_1, \dots, \alpha_m$ are released, their privacy costs must be added at a common accounting order before the minimum is taken as post-processing. We therefore define a family of possible diagnostics at a single order rather than one mechanism over several orders.

While releasing a private diagnostic $\widetilde{f}_\alpha(D) = f_\alpha(D) + S_{\xi,\alpha}(D)Z$ via GNSS can report the realized cost as in standard PATE, the label sequence itself would retain only its unconditional worst-case bound $L\alpha/\sigma^2$, with no formal certification of data-dependent stopping. Rather than releasing an uninstantiated diagnostic, we proceed directly to a mechanism that formally certifies the data-dependent label release.

### E.3 A Generalized PTR Release for Binary GNMax

The pointwise bound above can also certify a release at a predeclared privacy budget. We use generalized PTR (Redberg et al., 2023), which privately tests an upper confidence bound on $f_\alpha(D)$ and releases the noisy labels only when the test passes. The proposed query sequence and its length are fixed before observing the private votes. This is a separate mechanism from a post-hoc diagnostic.

Fix publicly $\xi > 0$, $1 < \alpha \le 1/(4\xi)$, $\sigma_s, \sigma_2 > 0$, a public floor $s_0 > 0$, and $\beta \in (0,1)$. Write

$$s(D) = \max\{S_{\xi,\alpha}(D), s_0\}, \qquad z_\beta = \Phi^{-1}(1-\beta/2).$$

The floor makes $\log s(D)$ defined even if the smooth upper bound vanishes. Since the maximum with a positive constant preserves both the local sensitivity upper bound and $\xi$-smoothness, $s$ remains a valid scale for GNSS. Draw standard normal variables $Z_2, Z_s$ that are independent of each other and of all GNMax label noises, and compute internally

$$\mu(D) = \log s(D) + \xi\sigma_2 Z_2 + \xi\sigma_2 z_\beta, \tag{15}$$

$$U_\alpha(D) = f_\alpha(D) + \sigma_s s(D) Z_s + \sigma_s z_\beta e^{\mu(D)}. \tag{16}$$

The test passes when $U_\alpha(D) \le r$, for a public proposed RDP charge $r \ge 0$. If it passes, release the $L$ binary GNMax labels as a student context. Otherwise, release only $\perp$. Neither the teacher votes nor the internal values $\mu$ and $U_\alpha$ are released.

**Theorem 6 (Binary GNMax release certified by PTR)** *Fix a public sequence of $L$ queries, a public mapping function $\pi$, deterministic teacher inference, and a GNMax noise scale $\sigma > 0$. Fix publicly $\alpha > 1$, $0 < \xi \le 1/(4\alpha)$, $\sigma_s, \sigma_2, s_0 > 0$, $r \ge 0$, and $\beta, \delta_{\text{test}}, \delta_{\text{lab}} \in (0,1)$ before observing the private table. Assume the label noises are independent across queries and independent of the test noises $Z_2, Z_s$. Let $f_\alpha$ and $S_{\xi,\alpha}$ be the pointwise divergence bound and its smooth sensitivity upper bound defined above. Set*

$$\rho_\alpha^{\text{test}} = \frac{\alpha}{2\sigma_2^2} + \frac{\alpha e^{2\xi}}{\sigma_s^2} + \frac{\xi\alpha - \frac{1}{2}\log(1-2\alpha\xi)}{\alpha - 1},$$
$$\varepsilon_{\text{test}} = \left[\rho_\alpha^{\text{test}} + \log\left(1 - \frac{1}{\alpha}\right) - \frac{\log\delta_{\text{test}} + \log\alpha}{\alpha - 1}\right]_+,$$
$$\varepsilon_{\text{lab}} = \left[r + \log\left(1 - \frac{1}{\alpha}\right) - \frac{\log\delta_{\text{lab}} + \log\alpha}{\alpha - 1}\right]_+.$$

*The PTR mechanism defined by Eq.* (15) *and* (16) *satisfies*

$$(\varepsilon_{\text{test}} + \varepsilon_{\text{lab}},\ \delta_{\text{test}} + \delta_{\text{lab}} + \beta)\text{-}\textbf{\textit{DP}}$$

*under add/remove record adjacency. The released student context and every prediction made solely from it inherit the same guarantee by post-processing.*

*Proof:* Write $T(D) = \mathbf{1}\{U_\alpha(D) \le r\}$ for the test and $\mathcal{M}(D) \in \{0,1\}^L$ for the binary GNMax labels. We verify the hypotheses of the generalized PTR theorem (Redberg et al., 2023, Thm. 4.3).

By Theorem 4, $S_{\xi,\alpha}$ bounds the local sensitivity of $f_\alpha$ and is $\xi$-smooth. The floored scale satisfies $s(D) \ge \mathrm{LS}_{f_\alpha}(D)$ and, for every $D \sim D'$,

$$s(D) = \max\{S_{\xi,\alpha}(D), s_0\} \le \max\{e^\xi S_{\xi,\alpha}(D'), e^\xi s_0\} = e^\xi s(D').$$

Since adjacency is symmetric and $s \ge s_0 > 0$, it follows that $|\log s(D) - \log s(D')| \le \xi$. The Gaussian release $\mu$ therefore costs $(\alpha, \alpha/(2\sigma_2^2))$-RDP. Its deterministic shift $\xi\sigma_2 z_\beta$ is public. Independently, $V(D) = f_\alpha(D) + \sigma_s s(D) Z_s$ is a GNSS release with $\sigma_{\text{SS}} = \sigma_s$, so Theorem 5 applies under $\alpha\xi \le 1/4$. Composition gives $(\alpha, \rho_\alpha^{\text{test}})$-RDP for $(\mu, V)$. Both $U_\alpha = V + \sigma_s z_\beta e^\mu$ and $T$ are post-processing of this pair. Hence Theorem 2 makes $T$ $(\varepsilon_{\text{test}}, \delta_{\text{test}})$-DP.

Each event $Z_2 < -z_\beta$ and $Z_s < -z_\beta$ has probability $\beta/2$. Outside their union, $e^{\mu(D)} \ge s(D)$ and, since $z_\beta > 0$,

$$U_\alpha(D) \ge f_\alpha(D) - \sigma_s s(D) z_\beta + \sigma_s z_\beta e^{\mu(D)} \ge f_\alpha(D).$$

Consequently, if $f_\alpha(D) > r$, the test passes with probability at most $\beta$.

Finally, if $f_\alpha(D) \le r$, Theorem 3 bounds both directed order-$\alpha$ divergences between the label laws at $D$ and at every neighbor $D'$ by $r$. Applying Theorem 2 to each direction shows that $\mathcal{M}$ satisfies $(\varepsilon_{\text{lab}}, \delta_{\text{lab}})$ data-dependent DP at $D$ in the sense of Redberg et al. (2023, Def. 4.1). The private test, its probability $\beta$ of a false pass, and this data-dependent guarantee are the hypotheses of Redberg et al. (2023, Thm. 4.3), which yields $(\varepsilon_{\text{test}} + \varepsilon_{\text{lab}}, \delta_{\text{test}} + \delta_{\text{lab}} + \beta)$-DP. Attaching the fixed public queries to the accepted labels and making predictions solely from the resulting student context are post-processing. □

**Ex-ante PTR certification versus post-hoc diagnostic.** The noisy estimate $\widetilde{f}_\alpha(D) = f_\alpha(D) + S_{\xi,\alpha}(D)Z$ provides a private diagnostic of the realized cost, but because the noise $Z$ can be negative, $\widetilde{f}_\alpha(D)$ can underestimate $f_\alpha(D)$. Using an uncertified estimate to stop adaptive labeling without an odometer or private filter leaks privacy and does not carry a formal guarantee (Feldman & Zrnic, 2021). In contrast, Theorem 6 tests an explicit upper confidence bound $U_\alpha(D)$ before any labels are released and aborts with $\perp$ if the test fails. This ensures that the released student context is covered by an end-to-end $(\varepsilon_{\text{test}} + \varepsilon_{\text{lab}}, \delta_{\text{test}} + \delta_{\text{lab}} + \beta)$-DP certificate.

# F Experimental Details

## F.1 Datasets

We evaluate TabDPSGD and the private baselines on seven datasets, five for classification and two for regression. We evaluate TabPATE on all five classification datasets. Every dataset uses one shared 80%/20% split into a train split and a test split. Table 12 reports the task, number of records, and feature composition of every dataset.

Table 12: Datasets we use to evaluate TabDPSGD, TabPATE, and the private baselines. TabPATE uses the five classification datasets.

| Dataset | Task | Records | Features | Categorical | Numerical |
|---|---|---|---|---|---|
| Polish Companies Bankruptcy (Polish) | Classification | 5,910 | 64 | 0 | 64 |
| Bank Customer Churn (Churn) | Classification | 10,000 | 10 | 4 | 6 |
| Credit Card Clients Default (Credit) | Classification | 30,000 | 23 | 3 | 20 |
| ACS Employ | Classification | 37,881 | 16 | 0 | 16 |
| Electricity | Classification | 38,474 | 7 | 0 | 7 |
| California Housing (California) | Regression | 20,640 | 8 | 0 | 8 |
| Bike Sharing | Regression | 17,379 | 12 | 4 | 8 |

## F.2 Private Baselines

Our private baselines represent two common approaches to private tabular prediction. DPSGD MLP and DPBoost train a predictor directly on the private table under DP, whereas AIM, MST, DP-CTGAN, and Tab-PE release a synthetic table on which an ordinary predictor can be trained without additional privacy cost.

**DPSGD MLP.** This baseline trains an MLP with DPSGD (Abadi et al., 2016). DPSGD clips each per-record gradient and adds Gaussian noise to the aggregated gradients before updating the model.

**DPBoost.** DPBoost (Li et al., 2020) is a differentially private GBDT algorithm that adapts the tree-based gradient-boosting framework LightGBM (Ke et al., 2017). It proposes gradient-based data filtering and geometric leaf clipping for training differentially private GBDTs. We note that DPBoost satisfies pure $\varepsilon$-DP (Li et al., 2020).

**AIM.** AIM (McKenna et al., 2022) adaptively selects low-order marginals that improve its current model, measures them with the Gaussian mechanism, and fits a graphical model to the noisy measurements. It then samples a synthetic table from the fitted model.

**MST.** MST (McKenna et al., 2021) privately estimates pairwise dependencies and uses them to construct a maximum spanning tree. It measures the corresponding low-order marginals with the Gaussian mechanism, fits a graphical model to the noisy measurements, and samples a synthetic table from that model.

**DP-CTGAN.** DP-CTGAN (Fang et al., 2022) trains the discriminator of a conditional generative adversarial network with DPSGD. The generator is optimized only through the privatized discriminator, so releasing records sampled from the generator is post-processing.

**Tab-PE.** Tab-PE (Tran et al., 2026) extends the Private Evolution framework (Lin et al., 2024) to tabular data. It uses an evolutionary process to iteratively improve a candidate dataset by leveraging tabular operators to produce variations, privately scoring them, and selecting the best candidates for propagation.

## F.3 Preprocessing and End-to-End Privacy Accounting

Numerical bounds depend on private records and therefore require privacy accounting. Prior work assumes public numerical bounds (Tran et al., 2026; McKenna et al., 2022). Our reported results include the cost of releasing these bounds.

**Private min-max normalization.** We release the scaling bounds of each numerical column in two stages. The locate stage releases a private bracket that contains most of the column's values. The estimate stage then releases, within this bracket, an estimated median $\hat{m}$ and an interquartile range $\widehat{\mathrm{IQR}}$. These quantile statistics are more robust to outliers than the mean and standard deviation, and follow recent work on tabular models such as RealMLP (Holzmüller et al., 2024). We set the released lower and upper scaling bounds to $\hat{L} = \hat{m} - 3\max(\widehat{\mathrm{IQR}}, 10^{-6})$ and $\hat{U} = \hat{m} + 3\max(\widehat{\mathrm{IQR}}, 10^{-6})$. We then apply min-max normalization as

$$x' = \mathrm{clip}\left(2\frac{x - \hat{L}}{\hat{U} - \hat{L}} - 1,\ -1,\ 1\right), \tag{17}$$

Applying Eq. (17) to records outside the fitting subset is post-processing of the released bounds and has no additional privacy cost.

**Budget composition.** All configurations use joint composition in RDP (Mironov, 2017). It splits $(\varepsilon, \delta)$ between preprocessing and training and compose them by basic composition. Let $\rho_{\mathrm{tot}}(\varepsilon, \delta)$ denote the scalar allowance used to construct the preprocessing RDP curve for a run with target $(\varepsilon, \delta)$. If preprocessing receives the share $\omega$ and its locate stage receives the share $\gamma$, then

$$\rho_{\mathrm{pre}} = \omega\rho_{\mathrm{tot}}, \qquad \rho_{\mathrm{loc}} = \gamma\rho_{\mathrm{pre}}, \qquad \rho_{\mathrm{est}} = (1-\gamma)\rho_{\mathrm{pre}}. \tag{18}$$

At every RDP order $\alpha$, the two stages contribute $\alpha\rho_{\mathrm{loc}}$ and $\alpha\rho_{\mathrm{est}}$. These contributions add to $\alpha\rho_{\mathrm{pre}}$. The accountant represents this release as an RDP certificate and combines its curve with the downstream mechanism before reporting the composed privacy value under $(\varepsilon, \delta)$-DP. For TabDPSGD, this combines the certificate with the composed subsampled Gaussian updates and gives the reported end-to-end guarantee. For TabPATE, the experiments combine preprocessing, the cost of DP marginal queries when applicable, and the data-dependent accounting.

### F.4 Evaluation Metrics

**AUC.** We use AUC for both classification utility and attacker performance. For classification it is computed from the predicted probabilities of the positive class on the test split. For membership inference, it is computed from the MIA scores.

**$\mathbf{R}^2$.** For regression, we report the coefficient of determination on the test split. $R^2$ is invariant under an affine map of the target, so it takes the same value whether a method predicts the target rescaled by its released interval or in its original units. The $R^2$ metric is defined as:

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2},$$

where $y_i$ and $\hat{y}_i$ are the true and predicted targets of test record $i$ and $\bar{y}$ is the mean of the true targets in the test split. $R^2 = 1$ indicates perfect predictions, $R^2 = 0$ matches predicting $\bar{y}$ for every record, and $R^2 < 0$ denotes worse performance than predicting $\bar{y}$.

**TPR at 1% FPR.** Following the low-FPR evaluation of LiRA (Carlini et al., 2022), we report the true positive rate at a false positive rate of $1\%$. We threshold the membership score of Eq. (2) so that only $1\%$ of non-member records are flagged. This measures whether an MIA identifies a small set of records with high confidence, which an average-case metric such as AUC can hide.

**Elo Rating.** Scores are not comparable across tables and tasks, so we rank methods with an Elo rating (Bradley & Terry, 1952) fitted jointly over all pairwise comparisons. Each pair of methods is compared once on each table on which both are evaluated and scores a win, a tie, or a loss by AUC or $R^2$. A tie occurs only when the two scores are equal and counts as half a win to each side. We fix the scale so that DPSGD MLP sits at 1000.

**Normalized Score.** To compare a private method with its non-private baseline, we rescale each score $s$ so that an uninformative predictor scores 0 and a perfect predictor scores 1:

$$u = \frac{s - c}{1 - c},$$

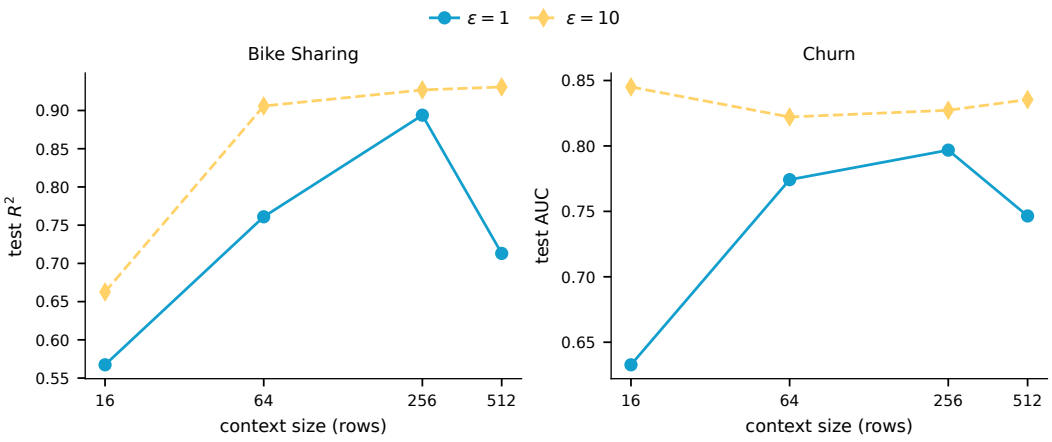


Figure 3: **Utility at different context sizes.** Utility score of TabDPSGD-Context using TabICLv2 against the number of learned context records, for $\varepsilon = 1$ and $\varepsilon = 10$.

where $c = 0.5$ for AUC and $c = 0$ for $R^2$. We report the ratio of a private method's normalized score to that of its non-private baseline, which is the share of the normalized score the private method retains. A ratio of raw AUC values would overstate this share, since an uninformative predictor already reaches 0.5 AUC.

### F.5 Hyperparameter Search

We carefully tune TabDPSGD, TabPATE, and all baselines separately on each dataset using Optuna. We reserve 20% of the data for testing and use 20% of the remaining data for validation. Each trial is trained on the training split and evaluated on the validation split, using AUC for classification and $R^2$ for regression. The configuration with the best validation score is then retrained on the combined training and validation data and evaluated once on the held-out test set. Each private run is calibrated to the reported $(\varepsilon, \delta)$, and thus satisfies the stated privacy guarantee for any fixed hyperparameter configuration. As is standard in empirical machine learning evaluations, we treat hyperparameter tuning as a model-selection step and do not account for the additional privacy loss incurred by selection on the validation set.

## G Additional Results on TabDPSGD

### G.1 Context Size Ablation

TabDPSGD trains the context parameters $\theta_C$ together with (parts of) the model parameters $\theta_W$. While the number of parameters in $\theta_W$ is fixed by the adaptation, one can choose the size (*i.e.,* the number of synthetic records) of the learned context $\theta_C$. We analyze the influence of context size on utility.

**Setup.** We fix the context size to 16, 64, 256, or 512 records and run a separate hyperparameter search at each size, with all other hyperparameters drawn from the same search space. We use TabICLv2 and only train the context, keeping all model parameters frozen. We evaluate on two of the seven datasets in Table 12, one classification, and one regression.

**Results.** Figure 3 shows the test score of the configuration that each search selected. We observe that at the lower privacy budget ($\varepsilon = 1$) increasing the context size to 512 leads to a drop in utility on both tables. In contrast, the score increases when the number of context records is increased to 512 at $\varepsilon = 10$. Our results suggest that a higher privacy budget allows for more context records.

### G.2 Non-Private Adaptation Results

**Setup.** We adapt Real-TabPFN-2.5 without privacy, *i.e.,* without gradient clipping and noising. We train using either a learned context, as in TabDPSGD, or the real training split as in-context data.

Table 13: **Non-private ranking.** Elo rating and the score on each dataset. Every fine-tuned method uses Real TabPFN v2.5. The Elo scale is fixed so the tuned MLP sits at 1000. The best score per column is bolded and the second best is underlined.

| Method | Elo ↑ | Electricity AUC ↑ | ACS Employ AUC ↑ | Churn AUC ↑ | Polish AUC ↑ | California $R^2$ ↑ | Bike Sharing $R^2$ ↑ |
|---|---|---|---|---|---|---|---|
| Fine-tuned LoRA (real context) | **1947** | 0.952 | **0.894** | **0.865** | <u>0.988</u> | <u>0.885</u> | <u>0.965</u> |
| Fine-tuned Full (real context) | 1907 | 0.957 | **0.894** | <u>0.863</u> | 0.985 | **0.887** | **0.966** |
| Non-private ICL (Real-TabPFN-2.5) | 1757 | 0.926 | **0.894** | **0.865** | 0.979 | 0.862 | 0.956 |
| Non-private ICL (TabICLv2) | 1727 | 0.958 | 0.891 | 0.862 | **0.989** | 0.879 | 0.959 |
| Fine-tuned Head (real context) | 1670 | 0.928 | <u>0.893</u> | <u>0.863</u> | 0.985 | 0.862 | 0.954 |
| XGBoost (Chen & Guestrin, 2016) | 1628 | <u>0.964</u> | **0.894** | 0.854 | 0.951 | 0.860 | 0.955 |
| Fine-tuned Logit (real context) | 1586 | 0.926 | **0.894** | <u>0.863</u> | 0.985 | 0.861 | 0.946 |
| Hist GBDT | 1558 | 0.958 | <u>0.893</u> | 0.857 | 0.959 | 0.844 | 0.951 |
| Random forest | 1325 | **0.967** | 0.891 | 0.850 | 0.927 | 0.829 | 0.938 |
| Fine-tuned Context (learned context) | 1201 | 0.907 | 0.892 | 0.847 | 0.921 | 0.845 | 0.947 |
| Fine-tuned Full (learned context) | 1186 | 0.908 | 0.891 | 0.853 | 0.923 | 0.838 | 0.943 |
| Fine-tuned LoRA (learned context) | 1154 | 0.907 | 0.892 | 0.848 | 0.919 | 0.815 | 0.948 |
| MLP | 1000 | 0.907 | 0.886 | 0.846 | 0.910 | 0.812 | 0.950 |
| Fine-tuned Logit (learned context) | 962 | 0.864 | 0.883 | 0.853 | 0.944 | 0.773 | 0.862 |
| Fine-tuned Head (learned context) | 851 | 0.867 | 0.884 | 0.845 | 0.921 | 0.788 | 0.902 |
| Linear | −1540 | 0.821 | 0.838 | 0.763 | 0.876 | 0.686 | 0.401 |

Table 14: **A learned context alone comes close to non-private ICL.** Scores of Real-TabPFN-2.5 with a learned context trained with and without privacy, with the best method fine-tuned over a learned context, and with non-private ICL. Bold marks the best score per dataset.

| Method | Electricity AUC | ACS Employ AUC | Churn AUC | Polish AUC | California $R^2$ | Bike Sharing $R^2$ |
|---|---|---|---|---|---|---|
| TabDPSGD-Context ($\varepsilon = 1$) | 0.855 | 0.858 | 0.773 | 0.810 | 0.759 | 0.772 |
| Learned context without privacy | 0.907 | 0.892 | 0.847 | 0.921 | 0.845 | 0.947 |
| Best fine-tuned method (learned context) | 0.908 | 0.892 | 0.853 | 0.944 | 0.838 | 0.948 |
| Non-private ICL | **0.926** | **0.894** | **0.865** | **0.979** | **0.862** | **0.956** |

Note that TabDPSGD-Context trains only the context, hence its non-private counterpart which uses the training data directly as context is standard non-private ICL. We rank each method against five tuned classical learners, including XGBoost (Chen & Guestrin, 2016) and an MLP. We fit the same Elo rating again on this panel, with the tuned MLP fixed at 1000. We evaluate on all datasets in Table 12 except Credit.

**Results.** Without privacy, fine-tuning Real-TabPFN-2.5 over the real context gives the highest Elo. Table 13 gives the ranking. LoRA fine-tuning using real data as the context leads to 1947 Elo, followed by full fine-tuning at 1907. Training the context costs utility compared to using real data, although the cost is moderate. Table 14 compares the non-private ICL results with the best adaptation that trains the context, the adaptation that only trains the context, and its private counterpart (*i.e.,* TabDPSGD-Context). On five of the six datasets, the best method that trains the context (with or without fine-tuning) trails the non-private ICL score by a maximum of 0.02. In contrast, training the context with DP loses an average AUC of 0.068 and an average $R^2$ of 0.130 compared to training the context without DP.

### G.3 Privacy Cost Relative to Non-Private Adaptation

We compare each private result with the corresponding non-private predictor on the six datasets for which both are available. We report the score decrease and the share of the normalized score retained (Appx. F.4).

Across matched adaptation methods, TabDPSGD-Head retains the highest median utility at 93%, followed by TabDPSGD-Logit and TabDPSGD-LoRA at 87%, and TabDPSGD-Context and TabDPSGD-Full at 84%. The DPSGD MLP retains 81%. Relative to non-private ICL, the best private variant retains median utility of 91% for TabICLv2 and 90% for Real-TabPFN-2.5. Figure 4 shows the per-dataset variation behind these summaries.

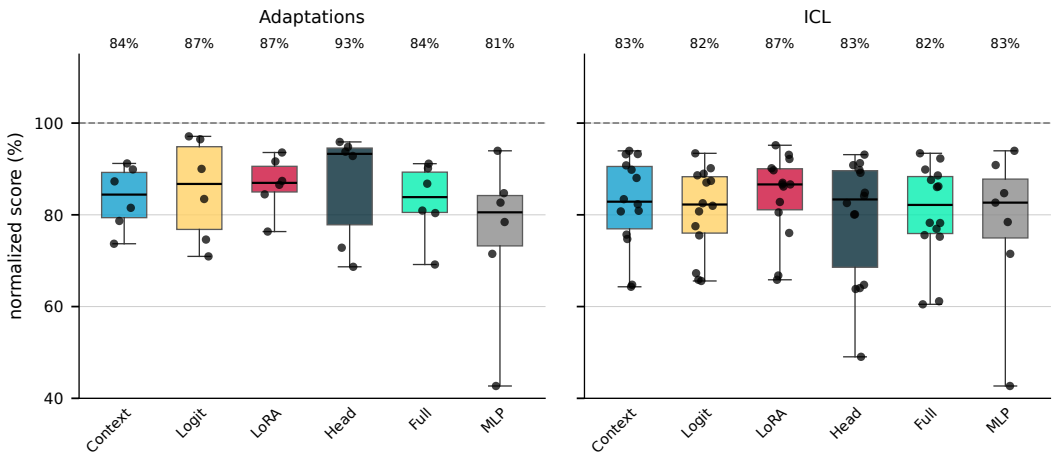


Figure 4: **TabDPSGD retains most of the utility in the private settings.** Share of the non-private normalized score retained by each private method. Left, each TabDPSGD variant against non-private fine-tuning of the same PEFT method; right, against non-private ICL with the same TFM. Each marker is one dataset, its shape naming the TFM, and the tuned MLP and DPSGD MLP pair is shown for comparison. The median is printed above each box. The dashed line marks no privacy cost.

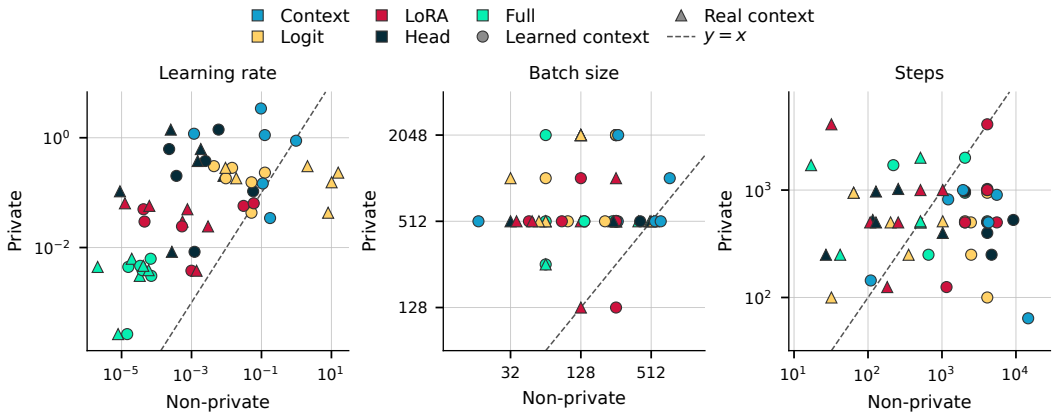


Figure 5: **Private fine-tuning uses higher learning rates and larger batches.** Hyperparameters of the private winner against the non-private winner for the same Real-TabPFN-2.5 variant and dataset. The dashed line marks equal values.

### G.4 Private and Non-Private Hyperparameters

Figure 5 compares the best private and non-private hyperparameter configurations for the same Real-TabPFN-2.5 adaptation variant and dataset. The figure reports the selected learning rate, batch size, and number of optimization steps for each configuration. We observe that the private setting prefers higher batch sizes and learning rates.

### G.5 Trainable Parameters

Figure 6 shows the trained parameters of each private method for the configuration selected at $\varepsilon = 1$.

TabDPSGD-Full trains the complete TFM and is larger than the DPSGD MLP on every dataset. However, the context and logit PEFT methods train orders of magnitude fewer parameters than the MLP.

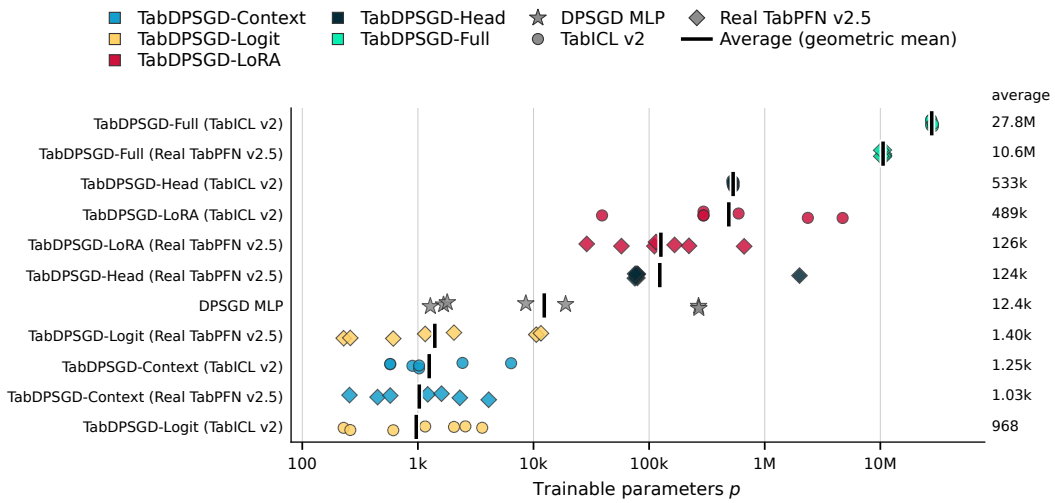


Figure 6: **PEFT greatly reduces the number of trained parameters.** Trained parameters on a log scale. Markers denote datasets, the black bars show geometric means, and marker shapes distinguish the TFMs.

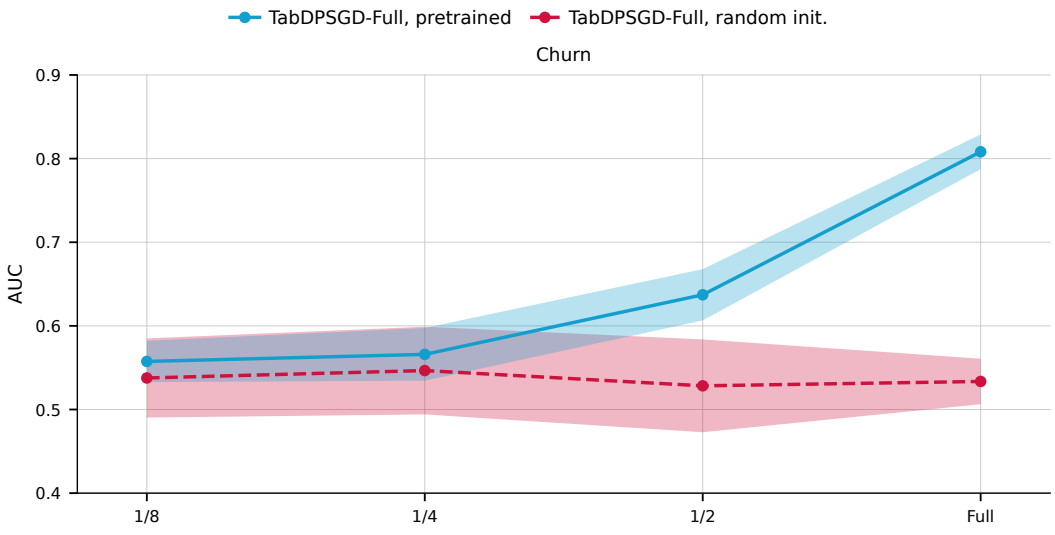


Figure 7: **The pretrained TFM prior helps TabDPSGD-Full when enough records are available.** AUC of TabDPSGD-Full on TabICLv2 at $\varepsilon = 1$ on Churn, starting from the pretrained TFM weights and from a random initialization, on subsets of the training records. Lines show the mean over five seeds and shaded areas one standard deviation.

### G.6 Importance of the Pretrained TFM Prior

To isolate the role of the TFM prior, we train TabDPSGD-Full on TabICLv2 from a random initialization instead of the pretrained TFM weights at $\varepsilon = 1$ on subsets of the Churn dataset and average each point over five seeds. Figure 7 shows that the randomly initialized TFM stays close to chance at every training size, reaching an AUC of $0.534$ on all training records against $0.808$ for the pretrained TFM. The pretrained TFM leads with at least half of the records ($0.637$ against $0.528$), while with a quarter or fewer the two overlap within one standard deviation, which suggests that the prior helps most when enough records are available for private fine-tuning.

# H Additional Results on TabPATE

## H.1 Candidate-Query Ablation

To further understand the importance of the candidate queries, Table 15 compares synthetic and real candidate queries with TabICLv2 teachers across different types of students.

Table 15: **Real candidate queries favor the TFM student, and synthetic ones favor the XGBoost student.** AUC of each type of student with TabICLv2 teachers, for synthetic and real candidate queries. Bold marks the best AUC per column for each type of candidate queries.

| Candidate queries | Student | Churn | Credit | ACS Employ | Polish | Electricity | Mean |
|---|---|---|---|---|---|---|---|
| Synthetic | XGBoost | 0.822 | **0.757** | **0.866** | **0.832** | **0.828** | **0.821** |
| Synthetic | TFM | **0.825** | 0.746 | 0.861 | 0.816 | 0.819 | 0.814 |
| Real | XGBoost | 0.797 | **0.755** | **0.869** | 0.832 | 0.852 | 0.821 |
| Real | TFM | **0.823** | **0.755** | 0.868 | **0.848** | **0.854** | **0.830** |

**Real candidate queries outperform synthetic ones.** They increase the best TFM student's AUC on every dataset except Churn, by $0.016$ on average, and the AUC of the better of the two students by $0.008$ on average. The largest gain is $0.034$ on Electricity, where real queries release 129 more records. Credit gains only $0.009$ despite releasing 1,825 more, so the number of released records does not explain the gain by itself.

**The better student depends on the candidate queries.** With real candidate queries, the TFM student beats the XGBoost student on four of five datasets, by $0.009$ AUC on average. With synthetic ones, the XGBoost student wins on four of five, by $0.007$ on average. Four of the ten margins are below $0.003$, less than a new draw of the vote noise can change the best student's AUC, so each order holds on average rather than on every dataset.

## H.2 Benefit of the Tighter Binary Accounting

**The tighter charge uses every vote margin.** Figure 8 reports how many labels fit within $\varepsilon = 1$ when all queries share a vote margin, with $K = 300$ and $\sigma/K = 0.2$. Our binary accounting allows $1.61$ times as many queries as standard PATE at a tied vote. The factor reaches $4.72$ near a margin of half the teachers and remains $2.79$ at unanimity.

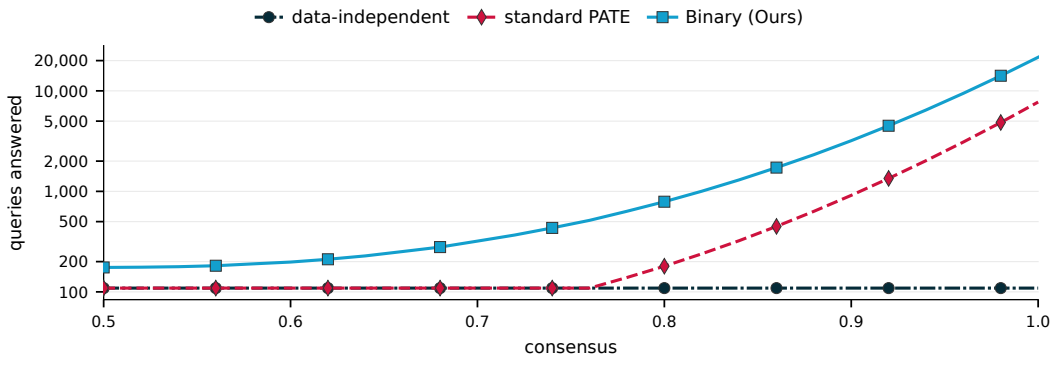


Figure 8: **The tighter binary allows more queries at every vote margin.** Queries answered at a specific teacher consensus rate with a privacy budget of $\varepsilon = 1$ and $\delta = 10^{-5}$, with $K = 300$ and $\sigma/K = 0.2$.

**The gain persists under real settings.** Figure 9 shows the number of queries answered by each configuration across the different accountant. Our tighter binary accountant allows between $1.66$ and $3.99$ times as many queries as standard PATE, with a median factor of $2.46$.

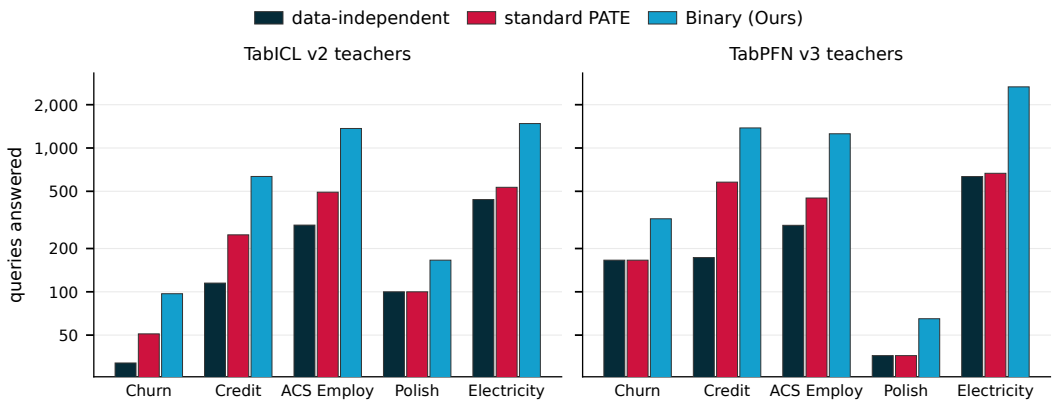


Figure 9: **Our tighter accounting allows to answer more queries across real settings.** Number of queries answered across real settings and tables by each accounting strategy.

### H.3 Teacher Context Size and the Cost of Partitioning

The size of each teacher partition motivates the utility decomposition in Figure 2. Figure 10 shows that non-private ICL continues to improve as its context grows, whereas partitioning leaves each TabPATE teacher with only between 42 and 74 labeled records on average. The release length alone does not recover the same information, since its labels are noised, its records are synthetic, and a student rather than a teacher TFM conditions on them.

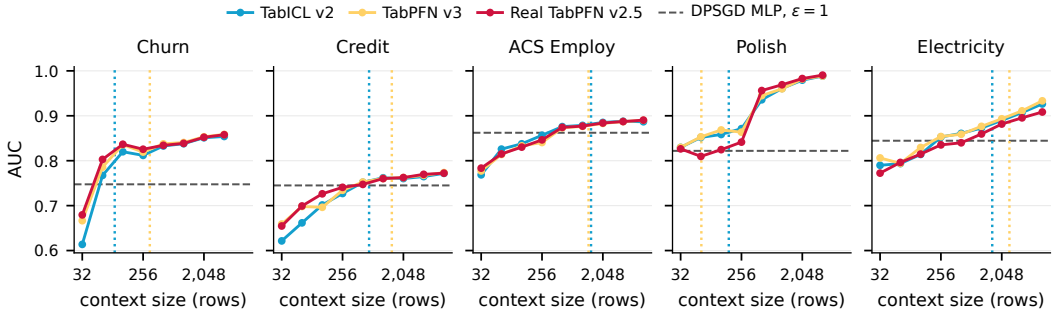


Figure 10: **More labeled records improve non-private ICL beyond a teacher partition.** Outer-test AUC against the number of true-labeled training records for three TFMs. Dashed horizontal lines mark the DPSGD MLP at $\varepsilon = 1$, and dotted vertical lines mark the lengths of the selected TabPATE releases.