TabLeak: Tabular Data Leakage in Federated Learning

Mark Vero [1] **Mislav Balunović** [2] **Dimitar I. Dimitrov** [2] **Martin Vechev** [2]

Abstract

While federated learning (FL) promises to preserve privacy, recent works in the image and text domains have shown that training updates leak private client data. However, most high-stakes applications of FL (*e.g.,* in healthcare and finance) use tabular data, where the risk of data leakage has not yet been explored. A successful attack for tabular data must address two key challenges unique to the domain: (i) obtaining a solution to a high-variance mixed discrete-continuous optimization problem, and (ii) enabling human assessment of the reconstruction as unlike for image and text data, direct human inspection is not possible. In this work we address these challenges and propose TabLeak, the first comprehensive reconstruction attack on tabular data. TabLeak is based on two key contributions: (i) a method which leverages a softmax relaxation and pooled ensembling to solve the optimization problem, and (ii) an entropy-based uncertainty quantification scheme to enable human assessment. We evaluate TabLeak on four tabular datasets for both FedSGD and FedAvg training protocols, and show that it successfully breaks several settings previously deemed safe. For instance, we extract large subsets of private data at $> 90\%$ accuracy even at the large batch size of 128. Our findings demonstrate that current high-stakes tabular FL is excessively vulnerable to leakage attacks.

1. Introduction

Federated Learning (McMahan et al., 2017) (FL) has emerged as the most prominent approach to training machine learning models collaboratively without requiring sensitive data of different parties to be collected in a central database. While prior work has examined privacy leakage from exchanged updates in FL on images (Zhu et al., 2019; Geiping et al., 2020; Yin et al., 2021) and text (Deng et al., 2021; Dimitrov et al., 2022a; Gupta et al., 2022), many applications of FL involve tabular datasets incorporating highly sensitive personal data such as financial information and health status (Borisov et al., 2021; Long et al., 2021; Rieke et al., 2020). However, as no prior work has studied the issue of privacy leakage in tabular data, we are unaware of the true extent of its risks. This is also a cause of concern for US and UK public institutions which have recently launched a \$1.6 mil. prize competition[1] to develop privacy-preserving FL solutions for financial fraud detection and infection risk prediction, both being tabular datasets.

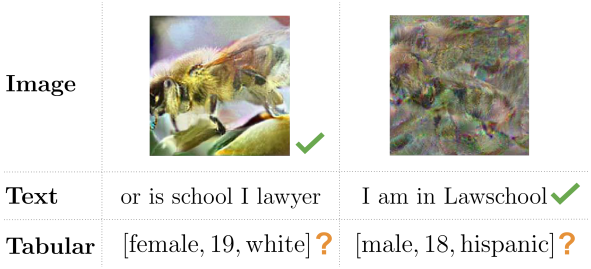


Figure 1: Comparison of image, text, and tabular data reconstruction. While the attack success can be judged by human inspection in images and text, for tabular data it is not possible, as both reconstructions *look* plausible. The image reconstruction example is taken from Yin et al. (2021).

Ingredients of a Data Leakage Attack A successful attack builds on two pillars: (i) ability to reconstruct private data from client updates with high accuracy, and (ii) a mechanism that allows a human to assess the obtained reconstructions *without* knowledge of the true data. Advancing along the first pillar typically requires leveraging the unique aspects of the given domain, *e.g.,* image attacks employ image priors (Geiping et al., 2020; Yin et al., 2021), while attacks on text make use of pre-trained language models (Dimitrov et al., 2022a; Gupta et al., 2022). However, in the image and text domains, the second pillar naturally comes for free, as the credibility of the obtained data can be assessed simply by human inspection, in contrast to tabular data, where this is not possible, as illustrated in Fig. 1.

[1]Department of Information Technology and Electrical Engineering, ETH Zurich, Zurich, Switzerland [2]Department of Computer Science, ETH Zurich, Zurich, Switzerland. Correspondence to: Mark Vero <mveroe@ethz.ch>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

[1]https://petsprizechallenges.com/

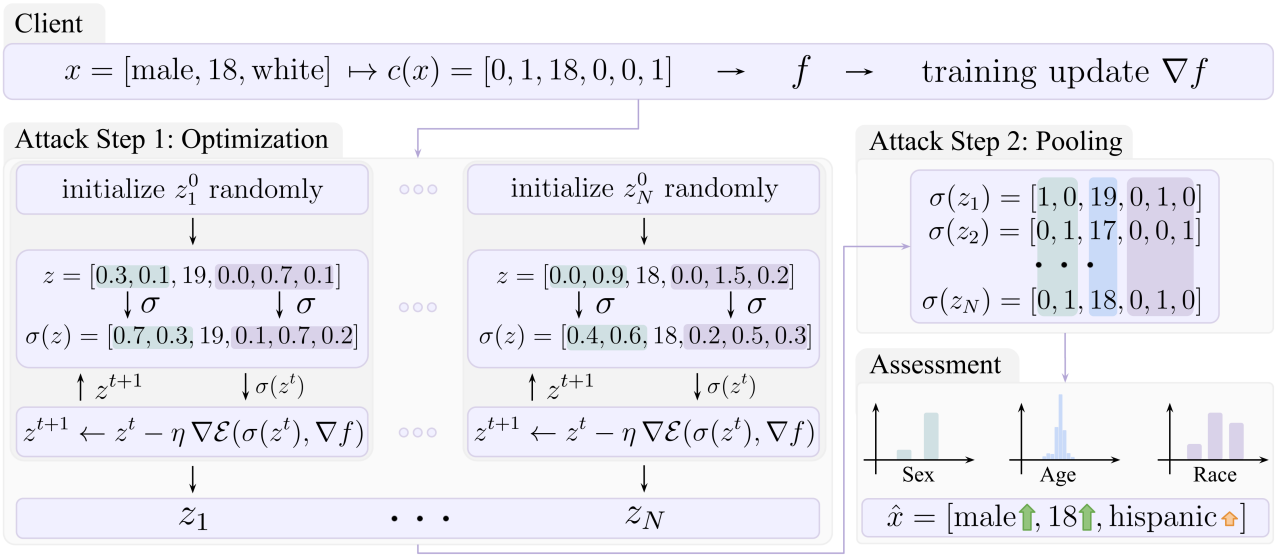


Figure 2: Overview of TabLeak. Our approach transforms the optimization problem into a fully continuous one by optimizing continuous versions of the discrete features, obtained by applying softmax (Attack Step 1, middle boxes), resulting in N candidate solutions (Attack Step 1, bottom). Then, we pool together an ensemble of N different solutions $z_1, z_2, ..., z_N$ obtained from the optimization to reduce the variance of the reconstruction (Attack Step 2). Finally, we assess the quality of the reconstruction by computing the entropy from the feature distributions in the ensemble (Assessment).

Key Challenges A strong attack for tabular data must address two unique challenges, one along each pillar: (i) due to the presence of both discrete and continuous features, the attack needs to solve a mixed discrete-continuous optimization problem of high variance, and (ii) unlike with image and text data, assessing the quality of the reconstruction is no longer possible via human inspection, requiring a mechanism to quantify the uncertainty of the reconstruction.

This Work In this work we propose the first comprehensive attack on tabular data, TabLeak, addressing the above challenges. Using our attack, we conduct the first comprehensive evaluation of the privacy risks posed by data leakage in tabular FL. We provide an overview of our approach in Fig. 2, showing the reconstruction of a client's private data point $x = [\texttt{male, 18, white}]$, from the corresponding update ∇f received by the server. We tackle the first challenge in two steps. In Attack Step 1, we create N separate optimization problems with different initializations. We transform the mixed discrete-continuous optimization problem into a fully continuous one using a softmax relaxation. Once optimization completes, in Attack Step 2, we reduce the variance of the final reconstruction by pooling over the different solutions. To address challenge (ii, Assessment), we rely on the observation that when the N reconstructions agree on a certain feature, it tends to be reconstructed well. We measure the agreement using entropy. In our example, *sex* and *age* exhibit a low entropy reconstruction and are also correct. Meanwhile, the high disagreement over the *race* feature is indicative of its incorrect reconstruction.

Comparing our domain-specific attack with prior works adapted from other domains on both FL protocols, FedSGD and FedAvg in various settings on four popular tabular datasets, we reveal the high vulnerability of such systems on tabular data, even in scenarios previously deemed as safe. We observe that on small batch sizes tabular FL systems are nearly transparent, where most attacks recover $> 90\%$ of the private data. Further, our attack retrieves 70.8% - 84.9% of the client data at the practically relevant batch size of 32 on the examined datasets, improving by 12.7% - 14.5% on prior art. Additionally, even on batch sizes as large as 128, we show how an adversary can recover a quarter of the private data well above 90% accuracy; leading to alarming conclusions about the privacy of FL on tabular data.

Main Contributions Our main contributions are:

- First effective domain-specific data leakage attack on tabular data called TabLeak, enabling novel insights into the unique aspect of tabular data leakage.
- An effective uncertainty quantification scheme, enabling the assessment of obtained samples and allowing an attacker to extract highly accurate subsets of features even from poor reconstructions.
- An extensive experimental evaluation, revealing the excessively high vulnerability of FL with tabular data by successfully conducting attacks even in setups previously deemed safe.

2. Background and Related Work

Federated Learning FL is a framework developed to facilitate the distributed training of a parametric model while preserving the privacy of the data at source (McMahan et al., 2017). Formally, we have a parametric function $f_\theta(x) = y$, where θ are the parameters. Given a dataset as the union of private datasets of clients $\mathcal{S} = \bigcup_{k=1}^{K} \mathcal{S}_k$, we now wish to find a θ^* such that $\frac{1}{N}\sum_{(x_i,y_i)\in\mathcal{S}} \mathcal{L}(f_{\theta^*}(x_i), y_i)$ is minimized, without first collecting the dataset $\mathcal{S}$ in a central database. McMahan et al. (2017) propose two training algorithms: FedSGD (a similar algorithm was also proposed by Shokri & Shmatikov (2015)) and FedAvg, that allow for the distributed training of f_θ, while keeping the data partitions $\mathcal{S}_k$ at client sources. The two protocols differ in how the clients compute their local updates in each step of training. In FedSGD, each client calculates the update gradient with respect to a randomly selected batch of their own data and shares it with the server. During FedAvg, the clients conduct a few epochs of local training on their own data before sharing their resulting parameters with the server. In each case, after the server has received the gradients/parameters from the clients, it aggregates them, updates the model, and broadcasts it to the clients; concluding an FL training step.

Data Leakage Attacks Although the design goal of FL was to preserve the privacy of clients' data, recent work has uncovered substantial vulnerabilities. Melis et al. (2019) first presented how one can infer certain properties of the clients' data. Later, Zhu et al. (2019) demonstrated that an *honest-but-curious* server can use the current state of the model and the received updates to reconstruct the clients' data, breaking the privacy promise of FL. Under this threat model, there has been extensive research on designing tailored attacks for images (Geiping et al., 2020; Zhao et al., 2020; Geng et al., 2021; Huang et al., 2021; Jin et al., 2021; Balunović et al., 2021; Yin et al., 2021; Jeon et al., 2021; Dimitrov et al., 2022b) and natural language (Deng et al., 2021; Dimitrov et al., 2022a; Gupta et al., 2022). However, no prior work has comprehensively dealt with data leakage attacks on tabular data, despite its significance in real-world high-stakes applications (Borisov et al., 2021). While, Wu et al. (2022) describe an attack on tabular data where a malicious client learns some distributional information from other clients, they do not reconstruct any private data points. Some works also consider a threat scenario where a *malicious server* may change the model or the updates sent to the clients (Fowl et al., 2021; Wen et al., 2022); but in this work we focus on the honest-but-curious setting.

In FedSGD, given the gradient $\nabla_\theta \mathcal{L}(f_\theta(x), y)$ of some client (shorthand: $g(x, y)$), we solve the following optimization problem to retrieve the client's private data (x, y):

$$\hat{x},\ \hat{y} = \underset{x',y'}{\arg\min}\, \mathcal{E}(g(x,y), g(x',y')) + \lambda\mathcal{R}(x'). \quad (1)$$

In Eq. 1 we denote the *gradient matching loss* as $\mathcal{E}$ and $\mathcal{R}$ is an optional regularizer for the reconstruction. The work of Zhu et al. (2019) used the mean squared error for $\mathcal{E}$, on which Geiping et al. (2020) improved using the cosine similarity loss. Zhao et al. (2020) first demonstrated that the private labels y can be estimated before solving Eq. 1, reducing the complexity of Eq. 1 and improving the attack results. Their method was later extended to batches by Yin et al. (2021) and refined by Geng et al. (2021). Eq. 1 is typically solved using continuous optimization tools such as L-BFGS (Liu & Nocedal, 1989) and Adam (Kingma & Ba, 2015). Although analytical approaches exist, they do not generalize to batches with more than a single data point (Zhu & Blaschko, 2021).

Domain-Specific Attacks Depending on the data domain, distinct tailored alterations to Eq. 1 have been proposed in the literature, *e.g.,* using the total variation regularizer for images (Geiping et al., 2020) and exploiting pre-trained language models in language tasks (Dimitrov et al., 2022a; Gupta et al., 2022). These mostly non-transferable domain-specific solutions are necessary as each domain poses unique challenges. Our work is first to identify and tackle the key challenges to data leakage in the tabular domain.

Privacy Threat of Tabular FL Regulations and personal interests prevent institutions from sharing privacy-sensitive tabular data, such as STI and drug test results, social security numbers, credit scores, and passwords. To this end, FL was proposed to enable inter-owner usage of such data. However, in a strict sense, if FL on tabular data leaks *any* private information, it does not fulfill its original design purpose, severely undermining trust in institutions employing such solutions. In our work we show that tabular FL, in fact, leaks *large amounts* of private information.

Mixed Type Tabular Data Mixed type tabular data is commonly used in healthcare, finance, and social sciences, which entail high-stakes privacy-critical applications (Borisov et al., 2021). Here, data is collected in a table with mostly human-interpretable columns, *e.g.,* age, and race of an individual. Formally, let $x \in \mathcal{X}$ be one *row* of data and let $\mathcal{X}$ contain K discrete columns and L continuous columns, *i.e.,* $\mathcal{X} = \mathcal{D}_1 \times \cdots \times \mathcal{D}_K \times \mathcal{U}_1 \times \cdots \times \mathcal{U}_L$, where $\mathcal{D}_i \subset \mathbb{N}$ and $\mathcal{U}_i \subset \mathbb{R}$. For processing with neural networks, discrete features are usually one-hot encoded, while continuous features are preserved. The one-hot encoding of the i-th discrete feature x_i^D is a binary vector $c_i^D(x)$ of length $|\mathcal{D}_i|$ that has a single non-zero entry at the position marking the encoded category. We retrieve the represented category by taking the argmax of $c_i^D(x)$ (projection to obtain x). Using the described encoding, one row of data $x \in \mathcal{X}$ is encoded as: $c(x) = \left[c_1^D(x),\, \ldots,\, c_K^D(x),\, x_1^C,\, \ldots,\, x_L^C\right]$, containing $d \coloneqq L + \sum_{i=1}^{K} |\mathcal{D}_i|$ entries.

3. Tabular Leakage

In this section, we briefly summarize the challenges in tabular leakage and present our solution to these, followed by our end-to-end reconstruction attack.

Key Challenges In the tabular domain, a strong attack has to address two unique challenges: (i) the presence of both categorical and continuous features requires the attacker to solve a significantly harder mixed discrete-continuous optimization problem of higher variance (addressed in Sec. 3.1.1 and Sec. 3.1.2), and (ii) as exemplified previously in Fig. 1, in contrast to images and text, it is hard for an unassisted adversary to assess the credibility of the reconstructed data in the tabular domain (addressed in Sec. 3.2).

3.1. Building a Strong Base Attack

We solve challenge (i) by introducing two components to our attack; a softmax relaxation to turn the mixed discrete-continuous problem into a fully continuous one (see Sec. 3.1.1), and pooled ensembling to reduce the variance in the final reconstruction (see Sec. 3.1.2).

3.1.1. The Softmax Relaxation

In accordance with prior literature on data leakage attacks, we aim to conduct the optimization in continuous domain. For this we employ the softmax relaxation, which turns the hard mixed discrete-continuous optimization problem into a fully continuous one. This drastically reduces its complexity, while still facilitating the recovery of correct discrete structures.

The recovery of one-hot vectors requires the integer constraints of all entries taking values in $\{0, 1\}$ and summing to one. Relaxing the integer constraints by allowing the reconstructed entries to take real values in $[0, 1]$, we are still left with a constrained optimization problem not well suited for popular continuous optimization tools, such as Adam (Kingma & Ba, 2015). Therefore, we aim to implicitly enforce the constraints introduced above.

For this, we extend the method of Zhu et al. (2019) used for inverting the discrete labels when jointly optimizing for both the labels and the data. Let $z \in \mathbb{R}^d$ be our approximate intermediate solution for the true one-hot encoded data $c(x)$ during optimization. Then we can implicitly enforce all constraints described above by applying a *softmax* to z_i^D for all i between 1 and K, *i.e.,* define:

$$\sigma(z_i^D)[j] \coloneqq \frac{\exp(z_i^D[j])}{\sum_{k=1}^{|\mathcal{D}_i|} \exp(z_i^D[k])} \qquad \forall j \in \mathcal{D}_i. \tag{2}$$

Therefore, in each round of optimization we will have the following approximation of the true data point: $c(x) \approx \sigma(z) = \left[\sigma(z_1^D), \ldots, \sigma(z_K^D), z_1^C, \ldots, z_L^C\right]$. In order to preserve notational simplicity, we write $\sigma(z)$ to mean the application of softmax to each group of entries representing a given categorical variable separately. Inverting a batch of data, the softmax is applied in parallel to the batch points.

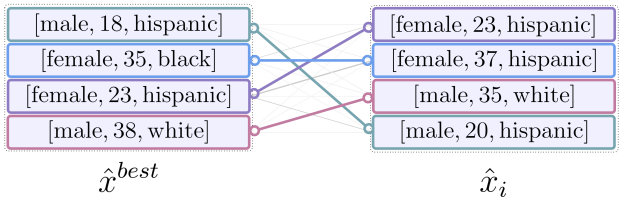


Figure 3: Maximum similarity matching of a sample $\hat{x}_i$ of batch size 4 from the collection of reconstructions to the best-loss sample $\hat{x}^{best}$.

3.1.2. Pooled Ensembling

In general, the data leakage optimization problem possesses multiple local minima (Zhu & Blaschko, 2021) and is sensitive to initialization (Wei et al., 2020). Additionally, we observed and confirmed in a targeted experiment in App. E that in tabular data the mix of discrete and continuous features introduces further variance, in contrast to image and text, where the problem is fully continuous or fully discrete, respectively. We alleviate this problem by running independent optimization processes with different initializations and ensembling their results through feature-wise pooling.

Exploiting the structural regularity of tabular data, we can combine independent reconstructions to obtain an improved and more robust final estimate of the true data by applying feature-wise pooling. Formally, we run N independent rounds of optimization with $i.i.d.$ initializations recovering potentially different reconstructions $\{\sigma(z_j)\}_{j=1}^N$. Then, we obtain a final estimate of the true encoded data, denoted as $\sigma(\hat{z})$, by pooling across these reconstructions in parallel for each batch-point and feature:

$$\sigma_i^D(\hat{z}) = \text{pool}\left(\left\{\sigma_i^D(z_j)\right\}_{j=1}^N\right) \qquad \forall i \in [K] \tag{3}$$

$$\hat{z}_i^C = \text{pool}\left(\left\{(z_i^C)_j\right\}_{j=1}^N\right) \qquad \forall i \in [L]. \tag{4}$$

Where the pool$(\cdot)$ operation can be any permutation invariant mapping. In our attack we use median pooling.

However, the above equations can not be applied in a straight-forward manner as soon as we aim to reconstruct batches containing more than just a single data point. As the batch-gradient is an average of the per-sample gradients, when running the leakage attack we may retrieve the batch-points in a different order at every optimization instance. Hence, it is not immediately clear how we can combine the obtained samples; *i.e.,* we need to reorder each batch such that their rows match to each other, and only then we can pool. We reorder by first selecting the sample that produced the best reconstruction loss at the end of optimization $\hat{z}^{best}$,

with projection $\hat{x}^{best}$. Then, we match the rows of every other sample in the collection with respect to $\hat{x}^{best}$. Concretely, we calculate the similarity (shown in Eq. 6 in Sec. 4) between each pair of rows of $\hat{x}^{best}$ and another sample $\hat{x}_i$ in the collection and find the maximum similarity reordering of the rows with the help of bipartite matching solved by the Hungarian algorithm (Kuhn, 1955). This process is depicted in Fig. 3. Repeating this for each sample, we reorder the entire collection with respect to the best-loss sample, effectively reversing the permutation differences in the independent reconstructions. Finally, we can apply feature-wise pooling for each row over the collection.

3.2. Assessment via Entropy

We now address challenge (ii), assessing reconstructions. To recap, it is close-to-impossible for an uninformed adversary to assess the quality of the obtained private sample when it comes to tabular data, as almost any reconstruction may constitute a credible data point when projected back to mixed discrete-continuous space. This challenge does not arise as prominently in the image (or text) domain, because one can easily judge by looking at a picture, if it is just noise or an actual image, as exemplified in Fig. 1. To address this issue, we propose to estimate the reconstruction uncertainty by looking at the level of agreement over a certain feature for different reconstructions. Concretely, given a collection of leaked samples as in Sec. 3.1.2, we can observe the distribution of each feature over the samples. Intuitively, if this distribution is "peaky", *i.e.,* concentrates the mass heavily on a certain value, then we can assume that the feature has been reconstructed correctly, whereas if there is high disagreement between the reconstructed samples, we can assume that this feature's recovered final value should not be trusted. We can quantify this by measuring the entropy of the feature distributions induced by the recovered samples.

Discrete Features Let $p(\hat{x}_i^D)_m \coloneqq \frac{1}{N}\,\text{Count}_j(\hat{x}_{ij}^D = m)$ be the relative frequency of projected reconstructions of the i-th discrete feature of value m in the ensemble. Then, we can calculate the normalized entropy of the feature as $\bar{H}_i^D = \frac{-1}{\log|\mathcal{D}_i|}\sum_{m=1}^{D_i} p(\hat{x}_i^D)_m \log p(\hat{x}_i^D)_m$. Note that the normalization allows for comparing features with different domain sizes, *i.e.,* it ensures that $\bar{H}_i^D \in [0,1]$, as $H(k) \in [0, \log|\mathcal{K}|]$ for any finite discrete random variable $k \in \mathcal{K}$.

Continuous Features In case of continuous features, we calculate the entropy by first making the standard assumption that the errors of the reconstructed continuous features follow a Gaussian distribution. As such, we first estimate the sample variance $\hat{\sigma}_i^2$ for the i-th continuous feature and then plug it in to calculate the entropy of the corresponding Gaussian: $H_i^C = \frac{1}{2} + \frac{1}{2}\log 2\pi\hat{\sigma}_i^2$. Cross-feature comparability can be achieved by scaling all features, *e.g.,* standardization.

Algorithm 1 TabLeak against training by FedSGD

1: **function SINGLEINVERSION** (Neural Network: f_θ, Client Gradient: $g(c(x), y)$, Reconstructed Labels: $\hat{y}$, Initial Reconstruction: z_j^0, Iterations: T, # Discrete Features: K)
2: **for** t **in** $0, 1, \ldots, T-1$ **do**
3: **for** k **in** $1, 2, \ldots, K$ **do**
4: $\sigma(z_{kj}^D) \leftarrow$ **softmax**(z_{kj}^D)
5: **end for**
6: $z_j^{t+1} \leftarrow z_j^t - \eta\,\nabla_z \mathcal{E}_{CS}(g(c(x), y), g(\sigma(z_j^t), \hat{y}))$
7: **end for**
8: **return** z_i^T
9: **end function**
10:
11: **function TABLEAK** (Neural Network: f_θ, Client Gradient: $g(c(x), y)$, Reconstructed Labels: $\hat{y}$, Ensemble Size: N, Iterations: T, # Discrete Features: K)
12: $\{z_j^0\}_{j=1}^N \sim \mathcal{U}_{[0,1]^d}$
13: **for** j **in** $1, 2, \ldots, N$ **do**
14: $z_j^T \leftarrow$ **SINGLEINVERSION**$(f_\theta, g(c(x), y), \hat{y}, z_j^0, T, K)$
15: **end for**
16: $\hat{z}^{best} \leftarrow \arg\min_{z_j^T} \mathcal{E}_{CS}(g(c(x), y), g(\sigma(z_j^T), \hat{y}))$
17: $\sigma(\hat{z}) \leftarrow$ **MATCHANDPOOL**$(\{\sigma(z_j^T)\}_{j=1}^N, \hat{z}^{best})$
18: $\bar{H}^D, H^C \leftarrow$ **CALCULATEENTROPY**$(\{\sigma(z_j^T)\}_{j=1}^N)$
19: $\hat{x} \leftarrow$ **PROJECT**$(\sigma(\hat{z}))$
20: **return** $\hat{x}$, $\bar{H}^D$, H^C
21: **end function**

3.3. Combined Attack

Following Geiping et al. (2020), we use the cosine similarity loss as our reconstruction objective, defined as:

$$\mathcal{E}_{CS}(z) \coloneqq 1 - \frac{\langle g(c(x), y),\, g(\sigma(z), \hat{y})\rangle}{\|g(c(x), y)\|_2\, \|g(\sigma(z), \hat{y})\|_2}, \qquad (5)$$

where (x, y) are the true data, $\hat{y}$ are the labels reconstructed beforehand, and we optimize for z. Our end-to-end attack, TabLeak is shown in Alg. 1. First, we reconstruct the labels using the label reconstruction method of Geng et al. (2021) and input them into our attack. Then, we initialize N independent dummy samples for an ensemble of size N (Line 12). Starting from each initial sample we optimize independently (Lines 13-15) via the SINGLEINVERSION function. In each optimization step, we apply the softmax relaxation of Sec. 3.1.1, and let the optimizer differentiate through it (Line 4). After the optimization processes have reached the maximum number of allowed iterations T, we identify the sample $\hat{z}^{best}$ producing the best reconstruction loss (Line 16). Using $\hat{z}^{best}$, we match and pool to obtain the final encoded reconstruction $\sigma(\hat{z})$ in Line 17 as described in Sec. 3.1.2. Finally, we return the projected private data reconstruction $\hat{x}$ and the corresponding feature-entropies $\bar{H}^D$ and H^C, quantifying the uncertainty in the leaked sample.

Table 1: The mean inversion accuracy [%] and standard deviation of different methods over varying batch sizes with given true labels (True y) and with reconstructed labels (Rec. $\hat{y}$) on the Adult dataset.

Label	Batch Size	TabLeak	TabLeak (no pooling)	TabLeak (no softmax)	Inverting Gradients Geiping et al. (2020)	Deep Gradient Leakage Zhu et al. (2019)	Random
True y	8	$\mathbf{95.2 \pm 8.8}$	92.5 ± 11.8	91.3 ± 7.1	91.1 ± 7.3	61.2 ± 4.7	53.9 ± 4.4
	16	$\mathbf{89.9 \pm 7.3}$	85.3 ± 9.7	79.0 ± 4.0	75.0 ± 5.2	60.2 ± 3.3	55.1 ± 3.9
	32	$\mathbf{79.3 \pm 4.5}$	74.3 ± 4.5	70.8 ± 3.3	66.6 ± 3.5	60.8 ± 1.9	58.0 ± 2.9
	64	$\mathbf{73.4 \pm 3.0}$	68.9 ± 3.1	67.3 ± 3.2	62.5 ± 3.1	61.3 ± 1.4	59.0 ± 3.2
	128	$\mathbf{71.4 \pm 1.2}$	67.4 ± 1.4	65.2 ± 2.1	59.5 ± 2.1	62.9 ± 1.0	61.2 ± 3.1
Rec. $\hat{y}$	8	$\mathbf{86.7 \pm 12.2}$	83.8 ± 13.6	82.7 ± 10.5	83.3 ± 9.7	56.1 ± 5.4	53.9 ± 4.4
	16	$\mathbf{83.0 \pm 7.7}$	78.6 ± 8.1	76.4 ± 5.4	73.0 ± 3.5	57.2 ± 3.4	55.1 ± 3.9
	32	$\mathbf{76.9 \pm 4.8}$	72.4 ± 4.8	68.9 ± 4.2	66.3 ± 3.4	58.4 ± 2.5	58.0 ± 2.9
	64	$\mathbf{72.8 \pm 3.3}$	68.5 ± 3.5	66.8 ± 2.9	63.1 ± 3.2	60.1 ± 1.7	59.0 ± 3.2
	128	$\mathbf{71.4 \pm 1.3}$	67.5 ± 1.5	65.0 ± 2.2	59.5 ± 2.1	62.3 ± 1.0	61.2 ± 3.1

4. Experimental Evaluation

In this section, we first detail the evaluation metric we used to assess the obtained reconstructions and explain our experimental setup. Then, we evaluate our attack in various settings against prior methods, establishing a new state-of-the-art, while uncovering the significant vulnerability of tabular FL. Next, we demonstrate the effectiveness of our entropy-based uncertainty quantification method. Finally, we test our attack on varying architectures, over federated training, and against a defense mechanism. Our code is available at: `https://github.com/eth-sri/tableak`.

Evaluation Metric As no prior work on tabular data leakage exists, we propose a metric for measuring the reconstruction accuracy, inspired by the 0-1 loss, allowing the joint treatment of categorical and continuous features. For a reconstruction $\hat{x}$, we define the accuracy as:

$$\text{accuracy}(x, \hat{x}) \coloneqq \frac{1}{K+L}\left(\sum_{i=1}^{K} \mathbb{I}\{x_i^D = \hat{x}_i^D\} + \sum_{i=1}^{L} \mathbb{I}\{\hat{x}_i^C \in [x_i^C - \epsilon_i,\, x_i^C + \epsilon_i]\}\right), \tag{6}$$

where x is the ground truth and $\{\epsilon_i\}_{i=1}^{L}$ are constants determining how close the reconstructed continuous features have to be to the original value in order to be considered a privacy breach. We provide more details on our metric in App. A and experiments with additional metrics in App. C.4.

Baselines We consider two established prior attacks; the seminal work on gradient inversion of Zhu et al. (2019), Deep Gradient Leakage, and the more recent strong attack of Geiping et al. (2020), Inverting Gradients. For a fair comparison, we provide the labels to both attacks in the same manner as we do for TabLeak and remove all non-tabular domain-specific elements from the attacks (*i.e.,* image priors). Additionally, we also compare against random guessing. Here, ignoring the gradient updates, we randomly sample reconstructions from the per-feature marginals of the input dataset. To obtain the 1-way marginals, first we uniformly discretize the continuous features in 100 bins, and then estimate the marginals of all features by counting. Although this baseline is usually not realizable in practice (as it assumes knowledge of the marginals), it is imperative to compare against it, as performing below this baseline signals that no private information is extracted from the client updates. Note that as both the selection of a batch and the random baseline represent sampling from the (approximate) data distribution, the random baseline monotonously increases in accuracy with growing batch size.

Experimental Setup For all attacks, we use the Adam optimizer (Kingma & Ba, 2015) with learning rate 0.06 for 1 500 iterations and without a learning rate schedule to perform the optimization in Alg. 1. Unless stated otherwise, we attack a fully connected neural network (NN) at initialization with two hidden layers of 100 neurons each. All experiments were carried out on four popular mixed-type tabular binary classification datasets, the Adult census dataset (Dua & Graff, 2017), the German Credit dataset (Dua & Graff, 2017), the Lawschool Admission dataset (Wightman, 2017), and the Health Heritage dataset from Kaggle (2012). Due to the space constraints, here we report only our results on the Adult dataset, and refer the reader to App. D for full results on all four datasets. Finally, for all reported numbers below, we estimate the mean and standard deviation of each reported metric on 50 randomly sampled batches. For further details on the experimental setup, we refer the reader to App. B. For experiments with varying network sizes and attacks against defenses, please see App. C.

General Results against FedSGD In Tab. 1 we present the results of TabLeak against FedSGD training, together with two ablation experiments, each time removing either the pooling (no pooling) or the softmax component (no softmax). We compare our results to the baselines introduced

Table 2: Mean and standard deviation of the inversion accuracy [%] on FedAvg with local dataset sizes of 32 on the Adult dataset. The accuracy of the random baseline for 32 data points is 57.7 ± 3.6.

	TabLeak			Inverting Gradients (Geiping et al., 2020)		
n. batches	1 epoch	5 epochs	10 epochs	1 epoch	5 epochs	10 epochs
1	$\mathbf{80.7 \pm 3.8}$	$\mathbf{75.8 \pm 3.3}$	$\mathbf{72.8 \pm 3.2}$	65.2 ± 2.7	56.1 ± 4.1	53.1 ± 4.2
2	$\mathbf{79.2 \pm 4.2}$	$\mathbf{75.6 \pm 2.7}$	$\mathbf{73.1 \pm 5.0}$	64.8 ± 3.3	56.4 ± 4.8	56.2 ± 4.8
4	$\mathbf{79.7 \pm 3.6}$	$\mathbf{76.2 \pm 3.0}$	$\mathbf{73.7 \pm 3.6}$	64.8 ± 3.4	58.7 ± 4.6	56.6 ± 5.0

above, on batch sizes $8, 16, 32, 64$, and 128, once assuming knowledge of the true labels (top) and once using labels reconstructed by the method of Geng et al. (2021) (bottom). Notice that the noisy label reconstruction only influences the results for lower batch sizes, and manifests itself mostly in higher variance in the results. Further, we find that for batch size 8 (and lower, see App. D) most attacks reveal close to all the private data, exposing a trivial vulnerability of tabular FL of high concern. In case of larger batch sizes, even up to 128, TabLeak can uncover a significant portion of the client's private data, well above random guessing, while the attacks of Zhu et al. (2019) and Geiping et al. (2020) fail to do so, demonstrating the necessity of a domain tailored attack when evaluating the privacy threat. Further, the results on the ablation attacks demonstrate the effectiveness of each attack component, both providing non-trivial improvements over the baseline attacks that are preserved when combined in TabLeak. Demonstrating generalization beyond Adult, we include our results on the German Credit, Lawschool Admissions, and Health Heritage datasets in App. D.1, where we also improve on the state-of-the-art by at least $12.7\% - 14.5\%$ on batch size 32 on each dataset and up to 21.8% on other batch sizes. As Inverting Gradients (Geiping et al., 2020) dominates Deep Gradient Leakage (Zhu et al., 2019) in almost all settings in Tab. 1, we omit the attack of Zhu et al. (2019) from further comparisons.

Categorical vs. Continuous Features An important consequence of having mixed type features is that the attack success clearly differs by feature type. As we can observe in Fig. 4, the continuous features exhibit an up to 30% lower accuracy than the categorical features for the same batch size. We suggest that this is due to the discrete nature of categorical features and their encoding. While trying to match the gradients by optimizing the reconstruction, having the correct categorical features will have a greater effect on the gradient alignment, as when encoded, they take up the majority of the data vector. Also, when reconstructing a one-hot encoded categorical feature, we only have to be able to retrieve the location of the maximum in a vector of length D_i, whereas for the successful reconstruction of a continuous feature we have to retrieve its value correctly up to a small error. Therefore, especially when the optimization process is aware of the discrete structure (*e.g.,* by using the

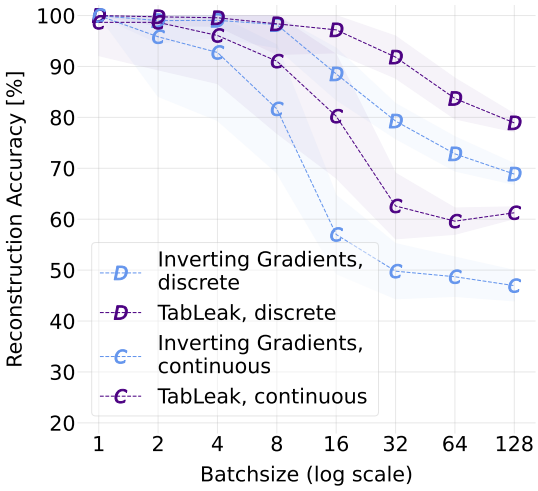


Figure 4: The inversion accuracy on the Adult dataset over varying batch size separated for discrete (D) and continuous (C) features.

softmax relaxation), categorical features are much easier to attack. This finding of ours uncovers a critical privacy risk in tabular federated learning, as sensitive features are often categorical, *e.g.,* gender, race, or STI test results.

Federated Averaging In training with FedAvg (McMahan et al., 2017) participating clients conduct local training of several updates before communicating their new parameters to the server. Note that the more local updates are conducted by the clients, the harder a leakage attack becomes, making FedAvg to be regarded as a more secure protocol. Although this training method is of significantly higher practical importance than FedSGD, most prior work does not evaluate against it. Transferring TabLeak into the framework of Dimitrov et al. (2022b) (for details please see App. B and the work of Dimitrov et al. (2022b)), we evaluate our attack and the strong baseline attack of Geiping et al. (2020) in the setting of Federated Averaging. We present our results of retrieving a client dataset of size 32 over varying number of local batches and epochs on the Adult dataset in Tab. 2, while assuming full knowledge of the true labels. We observe that our combined attack significantly outperforms the random baseline of 57.7% accuracy even up to 40 local

Table 3: The mean and the standard deviation of the attack accuracy [%] over different architectures inverting batches of size 32. The random baseline at this batch size is 58.0 ± 2.9.

Attack	Linear	FC NN	FC NN large	CNN (BN)	ResNet (BN)
Inverting Gradients (Geiping et al., 2020)	55.3 ± 1.7	66.6 ± 3.5	89.2 ± 3.8	43.4 ± 2.0	61.7 ± 4.0
GradInversion (Yin et al., 2021)	$\mathbf{61.3 \pm 2.7}$	67.7 ± 2.5	88.0 ± 3.0	72.8 ± 2.7	67.6 ± 2.6
TabLeak	44.5 ± 1.9	$\mathbf{79.3 \pm 4.5}$	$\mathbf{89.6 \pm 8.3}$	$\mathbf{83.7 \pm 2.7}$	$\mathbf{71.4 \pm 9.2}$

Table 4: The mean and standard deviation of the accuracy [%] of each feature type in the top 25% and the bottom 25% ranked in the batch according to entropy (lowest on top).

Batch	Categorical		Continuous	
Size	Top 25%	Bottom 25%	Top 25%	Bottom 25%
8	99.7 ± 2.3	96.0 ± 10.3	97.2 ± 9.4	84.2 ± 21.6
16	99.9 ± 0.6	89.8 ± 13.0	98.2 ± 3.5	60.8 ± 19.9
32	99.1 ± 2.6	75.5 ± 8.0	94.2 ± 4.7	43.6 ± 8.2
64	97.8 ± 2.6	66.1 ± 5.3	92.9 ± 3.5	41.3 ± 5.8
128	94.3 ± 1.9	62.8 ± 3.8	93.5 ± 2.3	42.2 ± 3.7

updates, and in some cases beating the baseline attack of Geiping et al. (2020) by almost 20%. Meanwhile, the baseline attack fails to consistently outperform random guessing whenever the local training is longer than one epoch. This shows that non-domain-tailored attacks are not sufficient to uncover relevant vulnerabilities, risking an illusion of privacy. As FedAvg with tabular data is of high practical relevance, our results of successful attacks are concerning. Further details of the experimental setup and results on other datasets can be found in App. B and App. D, respectively.

Assessing Reconstructions via Entropy We demonstrate the effectiveness of our assessment mechanism, looking at reconstructions from TabLeak and their corresponding feature entropies. For a reconstructed batch, we rank each per-sample feature (*i.e.,* all features over all rows in a single batch, distinguishing discrete and continuous features) according to decreasing entropy (lowest on top). Then, we take the top 25% of the features of lowest entropy and the bottom 25% from the resulting ranking, and report their accuracy for varying batch sizes in Tab. 4. The results in Tab. 4 confirm our expectation that features with low uncertainty score tend to be reconstructed much better than those exhibiting high entropy, asserting that our assessment mechanism is effective in separating well reconstructed data from poorly reconstructed, exposing an accuracy gap of up to 50% in the retrieved features. In fact, the top quarter of all features is reconstructed at an accuracy well above 90% on all batch sizes. Note that this ranking is done *without* any knowledge of the true data, hence it is accessible to a real adversary. This shows that even reconstructions of lower overall accuracy (*e.g.,* 71.4% on batch size 128, see Tab. 1) only provide a false sense of privacy, as an adversary can still extract correct data with high confidence, resulting in a strong breach of privacy even at large batch sizes, previously deemed as safe. We include results on all four datasets in App. D.4, leading to analogous conclusions.

Attack Performance and Model Architecture To assess how the attack success of our and competing methods depends on the chosen architecture, in this experiment we attack five different models, reconstructing batches of size 32: a logistic regressor (Linear), the two-layer NN used in prior experiments (FC NN), a large-three layer NN with 400 neurons in each layer (FC NN large), a convolutional NN (CNN) with a batch normalization (BN) layer, and a fully connected NN with residual connections and two BN layers (ResNet). For this experiment we also introduce an additional baseline, GradInversion (Yin et al., 2021), which is an attack tailored to image data, and was designed to overcome the limitations of BN layers. Note that this attack operates under strictly stronger assumptions than TabLeak, as it requires knowledge of the BN statistics, which Huang et al. (2021) argue is unrealistic. To adapt GradInversion to tabular data, we change the original reconstruction loss of squared error to cosine similarity, as it leads to better results in this domain. Additionally, as this attack requires the selection of several additive prior parameters, we evaluate the attack on a grid, and report only the best results here, providing an unrealistic advantage to this baseline. Our results are shown in Tab. 3. We make several important insights on the influence of the architecture on the attack success: (i) TabLeak is the strongest overall attack, even when confronted with BN layers, (ii) larger networks are significantly more vulnerable to all attacks, (iii) linear models are seemingly unbreakable under realistic assumptions. Therefore, while BN layers might only provide a false sense of privacy, it is imperative to consider models of limited size to enhance the privacy of FL.

Impact of Network Training Before, we only attacked networks at initialization. To examine how the attack success depends on the training state of the network, we evaluate TabLeak and the baseline attack of Geiping et al. (2020) attacking over the first 15 epochs of federated training with batch size 32. We report our results in Tab. 5. As expected, the attack performance gradually decreases as the network is fitted to the data, a phenomenon already known from

Table 5: The mean and standard deviation of the attack accuracy [%] attacking a network at the 1st, 5th, 10th, and 15th epoch of FedSGD training on batch size 32.

Training Epochs	TabLeak	Inverting Gradients Geiping et al. (2020)
1	$\mathbf{79.1 \pm 4.2}$	67.8 ± 2.1
5	$\mathbf{76.4 \pm 5.7}$	64.5 ± 3.8
10	$\mathbf{74.5 \pm 5.7}$	60.9 ± 3.7
15	$\mathbf{64.5 \pm 7.1}$	57.8 ± 4.0

Table 6: The mean and standard deviation of the attack accuracy [%] for Gaussian noise of varying scale added to the gradients. The right column reports the task accuracy [%] of the NN trained with perturbed gradients.

Noise Scale	TabLeak	Inverting Gradients Geiping et al. (2020)	Network Accuracy
0.0	$\mathbf{79.3 \pm 4.5}$	66.6 ± 3.5	84.6 ± 0.1
10^{-3}	$\mathbf{75.4 \pm 3.8}$	64.1 ± 3.2	84.5 ± 0.2
10^{-2}	$\mathbf{58.0 \pm 2.3}$	46.6 ± 2.8	84.4 ± 0.2
10^{-1}	$\mathbf{41.3 \pm 2.9}$	38.6 ± 2.2	84.1 ± 0.2

prior works (Geiping et al., 2020; Dimitrov et al., 2022b). However, TabLeak maintains a strong performance further into the training, and preserves its advantage over the baseline attack consistently. Further, note that decreasing attack performance over training is of limited practical relevance, as nothing prevents the server from attacking in the early stages of training where the model is vulnerable, breaching the privacy of the participating clients.

Defending with Noise As the undefended systems attacked above are critically vulnerable, we evaluate the effectiveness of a common defense method against gradient leakage attacks (Zhu et al., 2019; Dimitrov et al., 2022a;b). To defend against an honest-but-curious server the clients add Gaussian noise of zero mean and fixed scale to the gradients before communicating them to the server. Although this defense is inspired by differential privacy (DP) (Dwork, 2006), due to no clipping, it does not provide theoretical guarantees. We test this defense method against TabLeak and the baseline attack of Geiping et al. (2020) on varying noise scales at batch size 32, and report our results in Tab. 6. Encouragingly, we can observe that using the right amount of noise poses a viable defense against leakage attacks on the Adult dataset at this batch size, reducing the attack accuracy, while having a -0.5% impact on the task accuracy of the NN trained on the noisy gradients. In App. C.1, we present our results against this defense on all four datasets and over varying batch sizes, where we observe that smaller batch sizes require more noise to reduce vulnerability.

5. Discussion

Using TabLeak, we showed that an honest-but-curious FL server can reconstruct large amounts of private data with little effort, even in setups previously deemed as safe, such as large batch sizes and FedAvg. In particular, using our uncertainty quantification scheme, we can reconstruct a quarter of all features in a large batch of size 128 at $> 93\%$ accuracy. In a practical example, assuming all adults in the US have a bank account (≈ 260 million) at banks cooperating in FL, at least 65 million people would be affected by a potential attack leaking their information with high confidence, deeming it the sixth-largest financial data breach in history (Kost, 2022). Further, our discovery of the disproportionate vulnerability of discrete features argues for targeted mitigation, *e.g.,* by exploring alternative, safer representations.

Our results raise great concerns about current industrial FL systems potentially employed by financial, healthcare, or other institutions managing privacy-critical tabular data. Individuals' trust in such institutions when handing over sensitive information is fundamental in allowing for effective operation, to which end various legal institutions have been established, *e.g., barrister secrecy*, *medical secrecy*, or latest, *GDPR*. Apart from the damage inflicted on individuals whose private information may be abused by adversaries, the potential long-term loss of trust in institutions could lead to a wider impact on the services they are able to provide.

Defenses In addition to our attacks on *undefended* systems uncovering the excessive intrinsic risk in tabular FL, we showed in a promising experiment how a DP-inspired defense of adding noise to the communicated gradients can be leveraged to mitigate this risk. While this defense appears as effective in our experimental setup, in practice, the large associated cost in utility makes for limited applicability (Jayaraman & Evans, 2019) of DP methods. Therefore, it is necessary that further theoretically principled approaches are pursued in defending against data leakage attacks in FL.

6. Conclusion

In this work, we explored data leakage in tabular FL using TabLeak, the first data leakage attack on tabular data, obtaining state-of-the-art results against both popular FL training protocols, and uncovering the excessive vulnerability of tabular FL, breaking several setups previously thought of as safe. As tabular data is ubiquitous in privacy critical applications, our results raise important concerns regarding practical systems currently using FL. Therefore, we advocate for further research on advancing provable defenses.

Acknowledgements This work has received funding from the Swiss State Secretariat for Education, Research and Innovation (SERI) (SERI-funded ERC Consolidator Grant).

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016. ISBN 9781450341394. doi: 10.1145/2976749.2978318. URL https://doi.org/10.1145/2976749.2978318.

Balunović, M., Dimitrov, D. I., Staab, R., and Vechev, M. Bayesian framework for gradient leakage, 2021. URL https://arxiv.org/abs/2111.04706.

Borisov, V., Leemann, T., Seßler, K., Haug, J., Pawelczyk, M., and Kasneci, G. Deep neural networks and tabular data: A survey, 2021. URL https://arxiv.org/abs/2110.01889.

Deng, J., Wang, Y., Li, J., Wang, C., Shang, C., Liu, H., Rajasekaran, S., and Ding, C. TAG: Gradient attack on transformer-based language models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021. doi: 10.18653/v1/2021.findings-emnlp.305. URL https://aclanthology.org/2021.findings-emnlp.305.

Dimitrov, D. I., Balunović, M., Jovanović, N., and Vechev, M. Lamp: Extracting text from gradients with language model priors, 2022a. URL https://arxiv.org/abs/2202.08827.

Dimitrov, D. I., Balunović, M., Konstantinov, N., and Vechev, M. Data leakage in federated averaging, 2022b. URL https://arxiv.org/abs/2206.12395.

Dua, D. and Graff, C. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Dwork, C. Differential privacy. In *Proceedings of the 33rd International Conference on Automata, Languages and Programming - Volume Part II*, 2006. ISBN 3540359079. doi: 10.1007/11787006_1. URL https://doi.org/10.1007/11787006_1.

Fowl, L., Geiping, J., Czaja, W., Goldblum, M., and Goldstein, T. Robbing the fed: Directly obtaining private data in federated learning with modified models, 2021. URL https://arxiv.org/abs/2110.13057.

Geiping, J., Bauermeister, H., Dröge, H., and Moeller, M. Inverting gradients - how easy is it to break privacy in federated learning? In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Geng, J., Mou, Y., Li, F., Li, Q., Beyan, O., Decker, S., and Rong, C. Towards general deep leakage in federated learning, 2021. URL https://arxiv.org/abs/2110.09074.

Gupta, S., Huang, Y., Zhong, Z., Gao, T., Li, K., and Chen, D. Recovering private text in federated learning of language models, 2022. URL https://arxiv.org/abs/2205.08514.

Huang, Y., Gupta, S., Song, Z., Li, K., and Arora, S. Evaluating gradient inversion attacks and defenses in federated learning. *Advances in Neural Information Processing Systems*, 34:7232–7241, 2021.

Jayaraman, B. and Evans, D. E. Evaluating differentially private machine learning in practice. In *USENIX Security Symposium*, 2019.

Jeon, J., Kim, J., Lee, K., Oh, S., and Ok, J. Gradient inversion with generative image prior, 2021. URL https://arxiv.org/abs/2110.14962.

Jin, X., Chen, P.-Y., Hsu, C.-Y., Yu, C.-M., and Chen, T. Cafe: Catastrophic data leakage in vertical federated learning, 2021. URL https://arxiv.org/abs/2110.15122.

Kaggle. Health heritage prize. https://www.kaggle.com/c/hhp, 2012. Accessed: May 17, 2023.

Kendall, M. G. A New Measure of Rank Correlation. *Biometrika*, 30(1-2), 1938. ISSN 0006-3444. doi: 10.1093/biomet/30.1-2.81. URL https://doi.org/10.1093/biomet/30.1-2.81.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *Proc. of ICLR*, 2015. URL http://arxiv.org/abs/1412.6980.

Kost, E. 10 biggest data breaches in finance. https://tinyurl.com/yw34ynpe, 2022. Accessed: May 24, 2023.

Kuhn, H. W. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2), 1955. doi: https://doi.org/10.1002/nav.3800020109. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800020109.

Liu, D. C. and Nocedal, J. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45, 1989.

Long, G., Tan, Y., Jiang, J., and Zhang, C. Federated learning for open banking, 2021. URL https://arxiv.org/abs/2108.10749.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54, 2017. URL http://proceedings.mlr.press/v54/mcmahan17a.html.

Melis, L., Song, C., De Cristofaro, E., and Shmatikov, V. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, 2019. doi: 10.1109/SP.2019.00029.

Rieke, N., Hancox, J., Li, W., Milletarì, F., Roth, H. R., Albarqouni, S., Bakas, S., Galtier, M. N., Landman, B. A., Maier-Hein, K., Ourselin, S., Sheller, M., Summers, R. M., Trask, A., Xu, D., Baust, M., and Cardoso, M. J. The future of digital health with federated learning. *npj Digital Medicine*, 3(1), 2020. doi: 10.1038/s41746-020-00323-1. URL https://doi.org/10.1038%2Fs41746-020-00323-1.

Shokri, R. and Shmatikov, V. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015. ISBN 9781450338325. doi: 10.1145/2810103.2813687. URL https://doi.org/10.1145/2810103.2813687.

Wei, W., Liu, L., Loper, M., Chow, K.-H., Gursoy, M. E., Truex, S., and Wu, Y. A framework for evaluating gradient leakage attacks in federated learning, 2020. URL https://arxiv.org/abs/2004.10397.

Wen, Y., Geiping, J., Fowl, L., Goldblum, M., and Goldstein, T. Fishing for user data in large-batch federated learning via gradient magnification, 2022. URL https://arxiv.org/abs/2202.00580.

Wightman, F. L. LSAC national longitudinal bar passage study, 2017.

Wu, H., Zhao, Z., Chen, L. Y., and Van Moorsel, A. Federated learning for tabular data: Exploring potential risk to privacy. In *2022 IEEE 33rd International Symposium on Software Reliability Engineering (ISSRE)*, pp. 193–204, 2022. doi: 10.1109/ISSRE55969.2022.00028.

Yin, H., Mallya, A., Vahdat, A., Alvarez, J. M., Kautz, J., and Molchanov, P. See through gradients: Image batch recovery via gradinversion, 2021. URL https://arxiv.org/abs/2104.07586.

Zhao, B., Mopuri, K. R., and Bilen, H. idlg: Improved deep leakage from gradients, 2020. URL https://arxiv.org/abs/2001.02610.

Zhu, J. and Blaschko, M. B. R-GAP: recursive gradient attack on privacy. In *Proc. of ICLR*, 2021. URL https://openreview.net/forum?id=RSU17UoKfJF.

Zhu, L., Liu, Z., and Han, S. Deep leakage from gradients. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 2019.

A. Accuracy Metric

To ease the understanding, we start by repeating our accuracy metric here, where we measure the reconstruction accuracy between the retrieved sample $\hat{x}$ and the ground truth x as:

$$\text{accuracy}(x,\, \hat{x}) \coloneqq \frac{1}{K+L}\left(\sum_{i=1}^{K} \mathbb{I}\{x_i^D = \hat{x}_i^D\} + \sum_{i=1}^{L} \mathbb{I}\{\hat{x}_i^C \in [x_i^C - \epsilon_i,\, x_i^C + \epsilon_i]\}\right). \tag{7}$$

Note that the binary treatment of continuous features in our accuracy metric enables the combined measurement of the accuracy on both the discrete and the continuous features. From an intuitive point of view, this measure closely resembles how one would judge the correctness of numerical guesses. For example, guessing the age of a 25 year old, one would deem the guess good if it is within 3 to 4 years of the true value, but the guesses 65 and 87 would be both qualitatively incorrect. In order to facilitate scalability of our experiments, we chose the $\{\epsilon_i\}_{i=1}^{L}$ error-tolerance-bounds based on the global standard deviation if the given continuous feature σ_i^C and multiplied it by a constant, concretely, we used $\epsilon_i = 0.319\,\sigma_i^C$ for all our experiments. Note that $Pr[\mu - 0.319\,\sigma < x < \mu + 0.319\,\sigma] \approx 0.25$ for a Gaussian random variable x with mean μ and variance σ^2. For our metric this means that assuming Gaussian zero-mean error in the reconstruction around the true value, we accept our reconstruction as privacy leakage as long as we fall into the 25% error-probability range around the correct value. In Tab. 7 we list the tolerance bounds ϵ_i for the continuous features of the Adult dataset produced by this method. We would like to remark here, that we fixed our metric parameters before conducting any experiments, and did not adjust them based on any obtained results. Note also that in App. C we provide results where the continuous feature reconstruction accuracy is measured using the commonly used regression metric of root mean squared error (RMSE), where TabLeak also achieves the best results, signaling that the success of our method is independent of our chosen metric.

Table 7: Resulting tolerance bounds on the Adult dataset when using $\epsilon_i = 0.319\,\sigma_i^C$, as used by us for our experiments.

feature	`age`	`fnlwgt`	`education-num`	`capital-gain`	`capital-loss`	`hours-per-week`
tolerance	4.2	33699	0.8	2395	129	3.8

B. Further Experimental Details

Here we give an extended description to our experimental details provided in Sec. 4, additionally we provide the specifications of each used dataset in Tab. 8. For all attacks, we use the Adam optimizer (Kingma & Ba, 2015) with learning rate 0.06 for $1\,500$ iterations and without a learning rate schedule. We chose the learning rate based on our experiments on the baseline attack where it performed best. In line with Geiping et al. (2020), we modify the update step of the optimizer by reducing the update gradient to its element-wise sign. We attack a fully connected neural network with two hidden layers of 100 neurons each at initialization. However, we provide a network-size ablation in Fig. 9, where we evaluate our attack against the baseline method for 5 different network architectures. For each reported metric we conduct 50 independent runs on 50 different batches to estimate their statistics. For all FedSGD experiments we clamp the continuous features to their valid ranges before measuring the reconstruction accuracy, both for our attacks and the baseline methods. Additionally, for TabLeak and its ablation experiments, we encourage the continuous features to stay within bounds by wrapping them in a sigmoid function during optimization. Note that assuming some knowledge of the admissible ranges for the continuous columns is not unrealistic, as in most cases realistic ranges can be assumed based on the name of the feature column, especially if the feature standard deviations are known to the server (as in our case). For the network accuracy shown in the experiment "Defending with Noise", we train the two layer attacked network for 10 epochs at batch size 32 with learning rate 0.01 using mini-batch stochastic gradient descent. We ran each of our experiments on single cores of Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60GHz.

Federated Averaging Experiments For experiments on attacking the FedAvg training algorithm, we fix the clients' local dataset size at 32 and conduct an attack after local training with learning rate 0.01 on the initialized network described above. We use the FedAvg attack-framework of Dimitrov et al. (2022b), where for each local training epoch we initialize an independent mini-dataset matching the size of the client dataset, and simulate the local training of the client. At each reconstruction update, we use the mean squared error between the different epoch data means ($D_{\text{inv}} = \ell_2$ and $g =$ mean in Dimitrov et al. (2022b)) as the permutation invariant epoch prior required by the framework, ensuring the consistency of the

Table 8: Dataset specifications.

	Features	Discrete Features	Continuous Features	Encoded Features	Data Points
Adult	14	8	6	105	45 222
German	20	13	7	63	1 000
Lawschool	7	5	2	39	96 584
Health Heritage	17	6	11	110	218 415

reconstructed dataset. For the full technical details, please refer to the manuscript of Dimitrov et al. (2022b). For choosing the prior parameter λ_{inv}, we conduct line-search on each setup and attack method pair individually on the parameters $[0.0, 0.5, 0.1, 0.05, 0.01, 0.005, 0.001]$, and pick the ones providing the best results. Further, to reduce computational overhead, we reduce the ensemble size of TabLeak from 30 to 15 for these experiments on all datasets.

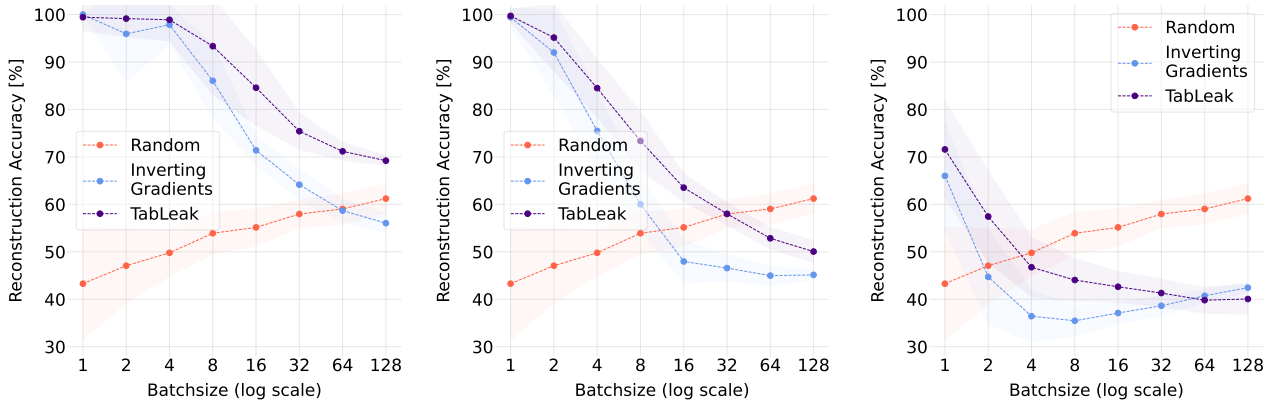
C. Further Experiments

In this subsection, we present several further experiments:

- Results of attacking neural networks defended using differentially private noisy gradients in App. C.1.
- Ablation study on the impacts of the neural network's size on the reconstruction difficulty in App. C.2.
- Ablation study on the impact of the neural network's architecture on the reconstruction difficulty in App. C.3.
- Measuring the Root Mean Squared Error (RMSE) of the reconstruction of continuous features in App. C.4.
- Testing TabLeak and the baseline attack on a high-dimensional synthetic dataset in App. C.5.
- Ablation study on the impact of training on the attack difficulty in App. C.6.
- Ablation study on the impact of number of attack iterations on the attack success in App. C.7.

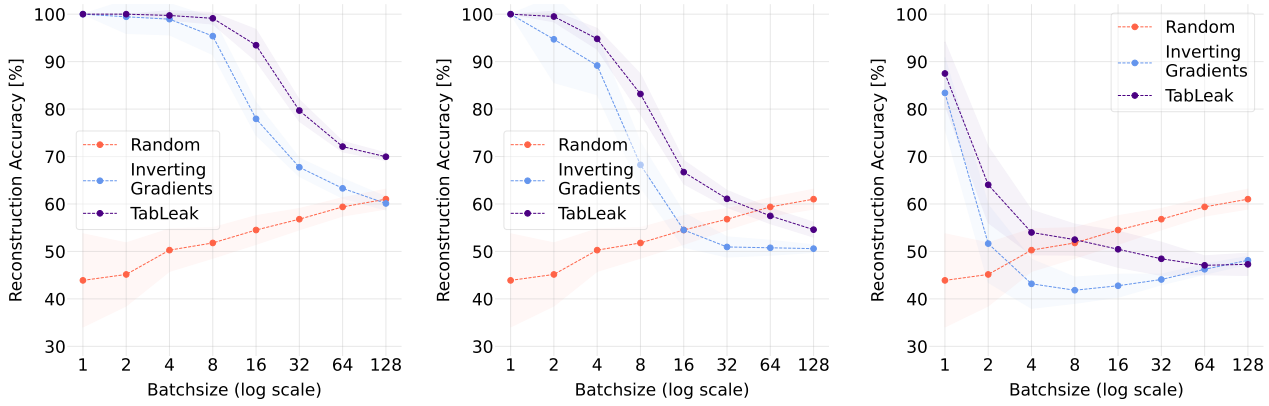
C.1. Attack against Gaussian DP Defense

Differential privacy (DP) has recently gained popularity, as a way to prevent privacy violations in FL (Abadi et al., 2016; Zhu et al., 2019). Unlike, empirical defenses which are often broken by specifically crafted adversaries (Balunović et al., 2021), DP provides guarantees on the amount of data leaked by a FL model, in terms of the magnitude of random noise the clients add to their gradients prior to sharing them with the server (Abadi et al., 2016; Zhu et al., 2019). Naturally, DP methods balance privacy concerns with the accuracy of the produced model, since bigger noise results in worse models that are more private. In this subsection, we evaluate TabLeak, and Inverting Gradients (Geiping et al., 2020) against DP-inspired defended gradient updates, where zero-mean Gaussian noise is added with standard deviations 0.001, 0.01, and 0.1 to the client gradients. Note that this defense does not lead to DP guarantees, as the gradients are not clipped prior to adding noise. Nevertheless, this method is in line with prior works on gradient leakage (Zhu et al., 2019; Dimitrov et al., 2022a;b). We present our results on the Adult, German Credit, Lawschool Admissions, and Health Heritage datasets in Fig. 5, Fig. 6, Fig. 7, and Fig. 8, respectively. Although both attack methods are affected by the defense, our method consistently produces better reconstructions than the baseline method. However, for high noise level (standard deviation $= 0.1$) and larger batch size both attacks break, advocating for the use of DP-inpsired defenses in tabular FL to prevent the high vulnerability exposed by this work.



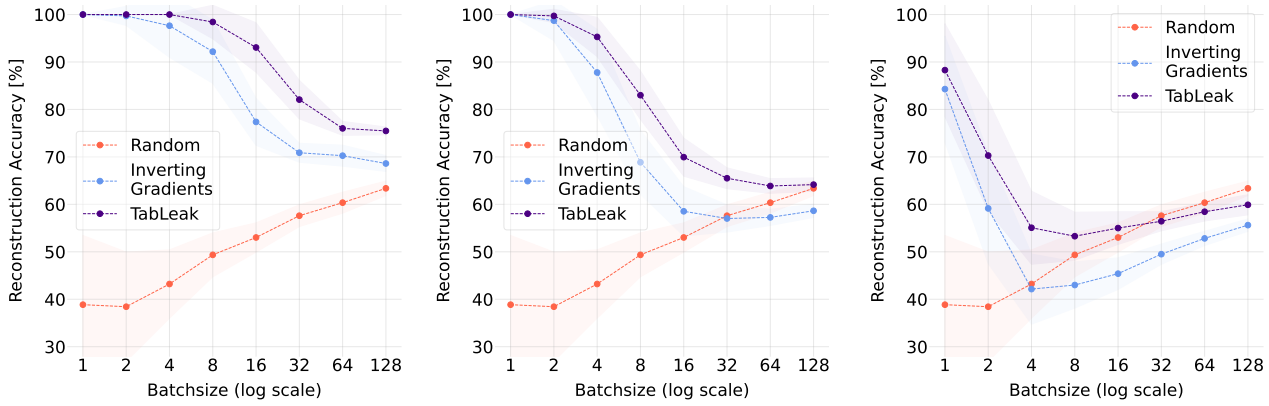
(a) Noise standard deviation = 0.001 (b) Noise standard deviation = 0.01 (c) Noise standard deviation = 0.1

Figure 5: Mean and standard deviation accuracy [%] curves over batch size at varying Gaussian noise level σ added to the client gradients for differential privacy on the **Adult** dataset.



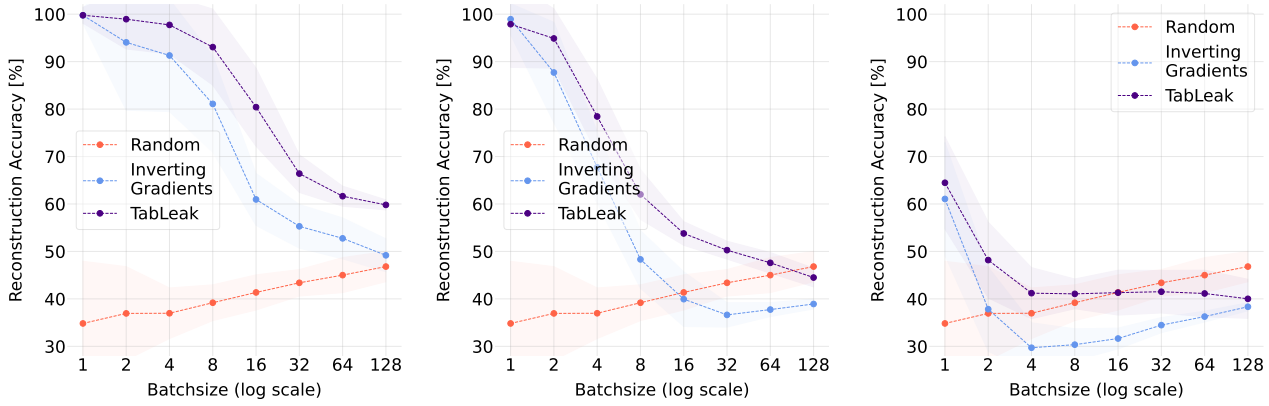
(a) Noise standard deviation = 0.001 (b) Noise standard deviation = 0.01 (c) Noise standard deviation = 0.1

Figure 6: Mean and standard deviation accuracy [%] curves over batch size at varying Gaussian noise level σ added to the client gradients for differential privacy on the **German Credit** dataset.



(a) Noise standard deviation = 0.001 (b) Noise standard deviation = 0.01 (c) Noise standard deviation = 0.1

Figure 7: Mean and standard deviation accuracy [%] curves over batch size at varying Gaussian noise level σ added to the client gradients for differential privacy on the **Lawschool Admissions** dataset.



(a) Noise standard deviation = 0.001 (b) Noise standard deviation = 0.01 (c) Noise standard deviation = 0.1

Figure 8: Mean and standard deviation accuracy [%] curves over batch size at varying Gaussian noise level σ added to the client gradients for differential privacy on the **Health Heritage** dataset.

Table 9: Mean and standard deviation of the peak test accuracy of each of the examined 6 models on the four discussed datasets over training.

	Linear	Layout 1	Layout 2	Layout 3	Layout 4	Layout 5
Adult	84.7 ± 0.1	84.9 ± 0.1	$\mathbf{85.0 \pm 0.1}$	84.8 ± 0.1	84.8 ± 0.1	84.7 ± 0.1
German	73.0 ± 1.4	80.0 ± 1.1	79.5 ± 0.6	$\mathbf{80.9 \pm 0.7}$	78.9 ± 1.0	79.4 ± 1.8
Lawschool	87.4 ± 0.0	89.6 ± 0.1	89.8 ± 0.0	$\mathbf{90.0 \pm 0.1}$	89.8 ± 0.1	89.8 ± 0.1
Health Heritage	80.9 ± 0.0	$\mathbf{81.2 \pm 0.1}$	$\mathbf{81.2 \pm 0.0}$	$\mathbf{81.2 \pm 0.1}$	$\mathbf{81.2 \pm 0.1}$	81.1 ± 0.1

C.2. Varying Network Size

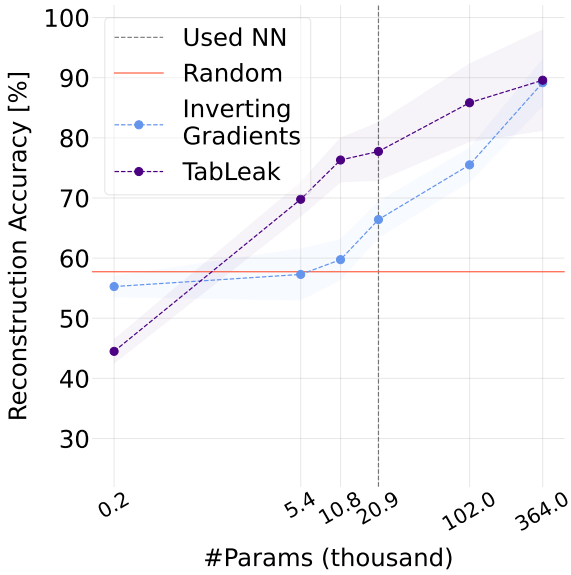
To understand the effect the choice of the network has on the obtained reconstruction results, we defined 4 additional fully connected networks, two smaller, and two bigger ones to evaluate TabLeak on. As a simple linear model is often a good baseline for tabular data, we add it also to the range of attacked models. Concretely, we examined the following six models for our attack:

- Linear: a linear classification network: $f_W(c(x)) = \sigma(Wc(x) + b)$,
- NN 1: a single hidden layer neural network with 50 neurons,
- NN 2: a single hidden layer neural network with 100 neurons,
- NN 3: a neural network with two hidden layers of 100 neurons each (network used in main body),
- NN 4: a neural network with three hidden layers of 200 neurons each,
- NN 5: a three hidden layer neural network with 400 neurons in each layer.

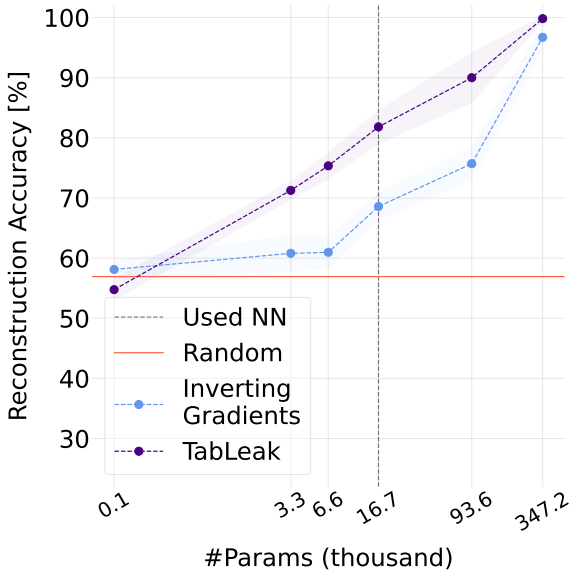
We attack the above networks, aiming to reconstruct a batch of size 32. We plot the accuracy of TabLeak and of Inverting Gradients (Geiping et al., 2020) as a function of the number of parameters in the network in Fig. 9 for all four datasets. We can observe that with increasing number of parameters in the network, the reconstruction accuracy significantly increases on all datasets, and rather surprisingly, allowing for near perfect reconstruction of a batch as large as 32 in some cases. Observe that on both ends of the presented parameter scale the differences between the methods degrade, *i.e.,* they either both converge to near-perfect reconstruction (large networks) or to random guessing (small networks). Therefore, the choice of our network for conducting the experiments was instructive in examining the differences between the methods.

Additionally, to better understand the relevance of the models examined here, we train them on each of the datasets for 50 epochs and observe their behavior through monitoring their performance on a secluded test set of each dataset. We do this for 5 different initializations of each model, and report the mean and the standard deviation of the test accuracy at each training epoch for each model. Note that we do not train the models using any FL protocol, merely, this experiment serves to give a better understanding between the relation of the given dataset and the model used, putting also the attack success data in better perspective. For training, we use the Adam Kingma & Ba (2015) optimizer and batch size 256 for each of the datasets, except for the German Credit dataset, where we train with batch size 64 due to its small size. We provide all test accuracy curves over training in Fig. 10. From the accuracy curves we can observe that most large models that are easy to attack tend to overfit quickly to the data, indicating a heavily overparameterized regime. Additionally, in Tab. 9 we provide the peak mean test accuracies per dataset and model, effectively corresponding to a 'perfect' early-stopping. The linear model could appear to be an overall good choice, as it is very hard to attack and shows good stability during training, however, it does not achieve competitive performance on most datasets. In Tab. 9 the non-linear models always outperform the linear model, and achieve comparable performance across themselves in this ideal setting, where overfitting can be prevented by monitoring on the test data[2]. Conclusively, simpler non-linear models shall be pursued for FL on tabular data, as they are less prone to overfitting and provide better protection from data leakage attacks.

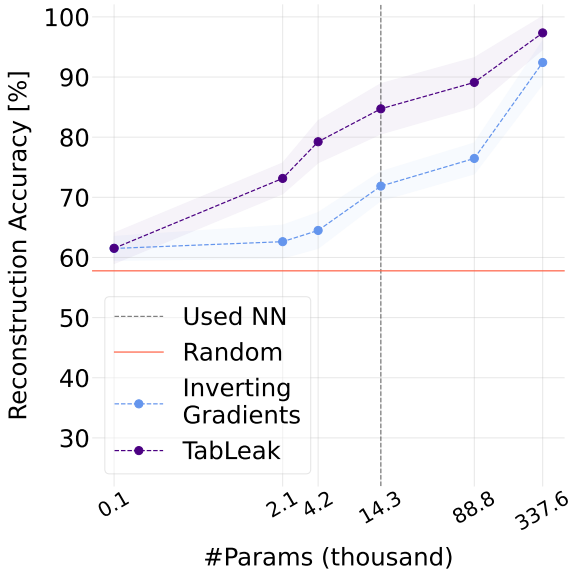
[2]In practice a proxy metric would be necessary to achieve early-stopping, such as monitoring the performance on a separate validation set split from the training data



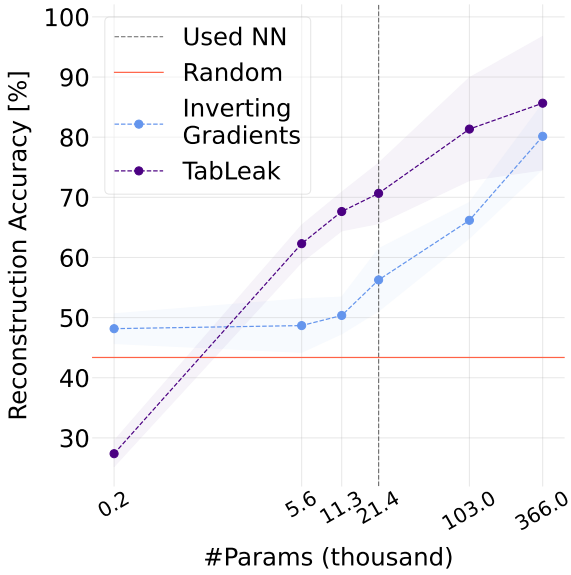
(a) Adult: $d = 105$



(b) German Credit: $d = 63$

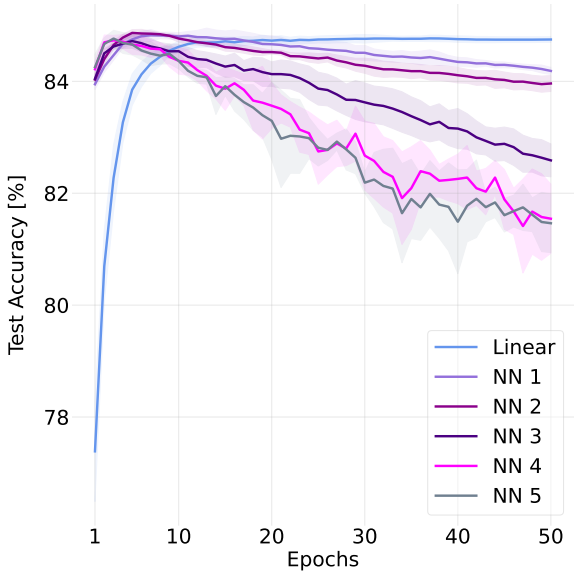


(c) Lawschool Admissions: $d = 39$

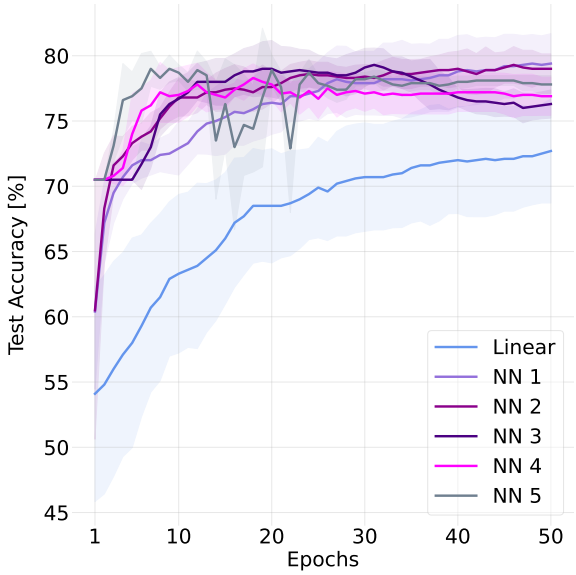


(d) Health Heritage: $d = 110$

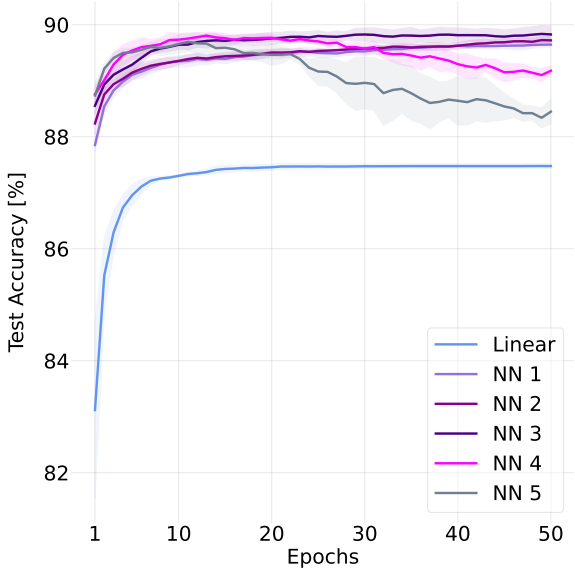
Figure 9: Mean attack accuracy curves with standard deviation for batch size 32 over varying network size (measured in number of parameters, #Params, log scale) on all four datasets with d number of features after encoding. We mark the network we used for our other experiments with a dashed vertical line. From left to right we have the following models: Linear, NN 1, NN 2, NN 3, NN 4, and NN 5.



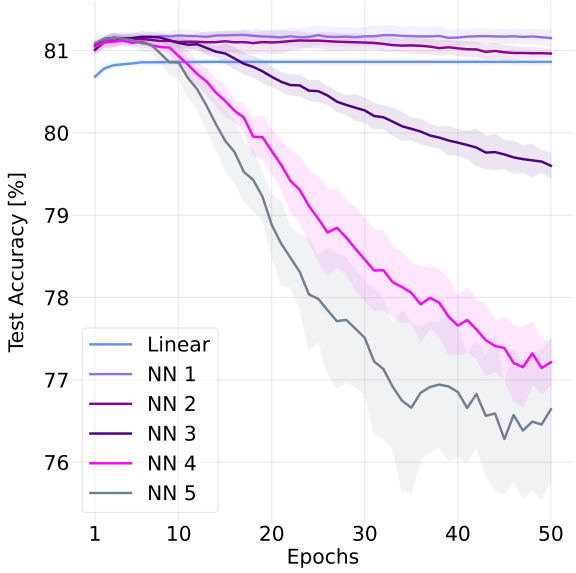
(a) Adult



(b) German Credit



(c) Lawschool Admissions



(d) Health Heritage

Figure 10: Mean test and standard deviation of the test accuracy over epochs during five independent runs of training for each examined model on all four datasets. For our experiments elsewhere we used the network corresponding to Layout 3, marked in dark violet here.

C.3. Varying Network Architecture

To investigate the impact of the network architecture on the attack success, we test TabLeak and two baseline methods, Inverting Gradients (Geiping et al., 2020) and GradInversion (Yin et al., 2021) (introduced in Sec. 4 in the main body) on various network architectures. The examined architectures are:

- Linear: a linear classification network: $f_W(c(x)) = \sigma(Wc(x) + b)$;
- FC NN: a neural network with two hidden layers of 100 neurons each (network used in the main body);
- FC NN large: a three hidden layer neural network with 400 neurons in each layer;
- CNN (BN): a convolutional neural network with a single initial convolutional layer of kernel size 3 and 16 output channels, followed by a batch normalization layer (BN) and two fully connected hidden layers of 100 neurons;
- ResNet (BN): a fully connected neural network with two residual blocks, each containing a batch normalization layer.

Our results on all four datasets are included in Tabs. 10 to 13. Note that on the last two architectures we raised the number of iterations for all the attacks to $7\,000$. We can confirm the three observations we have already made in the main body of the paper: (i) TabLeak is the strongest overall attack across various architectures, with BN layers not impacting its position either, and in line with App. C.2, we again observe that large networks are excessively vulnerable to all attacks (ii) and that the linear model is hard to break for any attack (iii). Therefore, we again argue for a conservative architecture choice, with as little parameters as possible that are still fit to solve the underlying task.

Table 10: The mean and the standard deviation of the attack accuracy [%] over different architectures inverting batches of size 32 on the **Adult** dataset. The random baseline at this batch size is 58.0 ± 2.9.

Attack	Linear	FC NN	FC NN large	CNN (BN)	ResNet (BN)
Inverting Gradients (Geiping et al., 2020)	55.3 ± 1.7	66.6 ± 3.5	89.2 ± 3.8	43.4 ± 2.0	61.7 ± 4.0
GradInversion (Yin et al., 2021)	$\mathbf{61.3 \pm 2.7}$	67.7 ± 2.5	88.0 ± 3.0	72.8 ± 2.7	67.6 ± 2.6
TabLeak	44.5 ± 1.9	$\mathbf{79.3 \pm 4.5}$	$\mathbf{89.6 \pm 8.3}$	$\mathbf{83.7 \pm 2.7}$	$\mathbf{71.4 \pm 9.2}$

Table 11: The mean and the standard deviation of the attack accuracy [%] over different architectures inverting batches of size 32 on the **German Credit** dataset. The random baseline at this batch size is 56.8 ± 2.2.

Attack	Linear	FC NN	FC NN large	CNN (BN)	ResNet (BN)
Inverting Gradients (Geiping et al., 2020)	$\mathbf{58.1 \pm 1.5}$	69.7 ± 2.2	96.7 ± 2.1	60.7 ± 2.3	66.4 ± 3.0
GradInversion (Yin et al., 2021)	57.8 ± 1.4	68.8 ± 1.8	97.0 ± 2.1	71.7 ± 1.9	70.9 ± 2.3
TabLeak	54.8 ± 1.8	$\mathbf{84.2 \pm 2.8}$	$\mathbf{99.8 \pm 0.4}$	$\mathbf{79.2 \pm 3.6}$	$\mathbf{74.5 \pm 4.2}$

Table 12: The mean and the standard deviation of the attack accuracy [%] over different architectures inverting batches of size 32 on the **Lawschool Admissions** dataset. The random baseline at this batch size is 57.6 ± 2.3.

Attack	Linear	FC NN	FC NN large	CNN (BN)	ResNet (BN)
Inverting Gradients (Geiping et al., 2020)	$\mathbf{61.5 \pm 2.0}$	71.0 ± 2.8	92.4 ± 3.6	61.1 ± 2.1	69.4 ± 3.2
GradInversion (Yin et al., 2021)	61.4 ± 2.1	71.8 ± 2.4	91.5 ± 4.2	74.7 ± 3.0	74.5 ± 3.0
TabLeak	$\mathbf{61.5 \pm 2.5}$	$\mathbf{84.9 \pm 4.0}$	$\mathbf{97.3 \pm 2.8}$	$\mathbf{79.1 \pm 3.1}$	$\mathbf{82.6 \pm 6.7}$

Table 13: The mean and the standard deviation of the attack accuracy [%] over different architectures inverting batches of size 32 on the **Health Heritage** dataset. The random baseline at this batch size is 43.4 ± 2.8.

Attack	Linear	FC NN	FC NN large	CNN (BN)	ResNet (BN)
Inverting Gradients (Geiping et al., 2020)	48.2 ± 2.5	57.7 ± 4.1	80.1 ± 5.4	29.2 ± 4.9	54.1 ± 6.1
GradInversion (Yin et al., 2021)	$\mathbf{48.6 \pm 3.1}$	58.2 ± 2.6	84.3 ± 5.6	64.9 ± 3.1	$\mathbf{60.1 \pm 2.8}$
TabLeak	27.4 ± 2.1	$\mathbf{70.8 \pm 4.5}$	$\mathbf{85.7 \pm 11.1}$	$\mathbf{72.9 \pm 4.6}$	46.7 ± 12.8

C.4. Continuous Feature Reconstruction Measured by RMSE

In order to examine the potential influence of our choice of reconstruction metric on the obtained results, we further measured the reconstruction quality of continuous features on the widely used Root Mean Squared Error (RMSE) metric as well. Concretely, we calculate the RMSE between the L continuous features of our reconstruction $\hat{x}^C$ and the ground truth x in a batch of size n as:

$$\text{RMSE}(x^C, \hat{x}^C) = \frac{1}{L} \sum_{i=1}^{L} \sqrt{\frac{1}{n} \sum_{j=1}^{n} (x_{ij}^C - \hat{x}_{ij}^C)^2}. \tag{8}$$

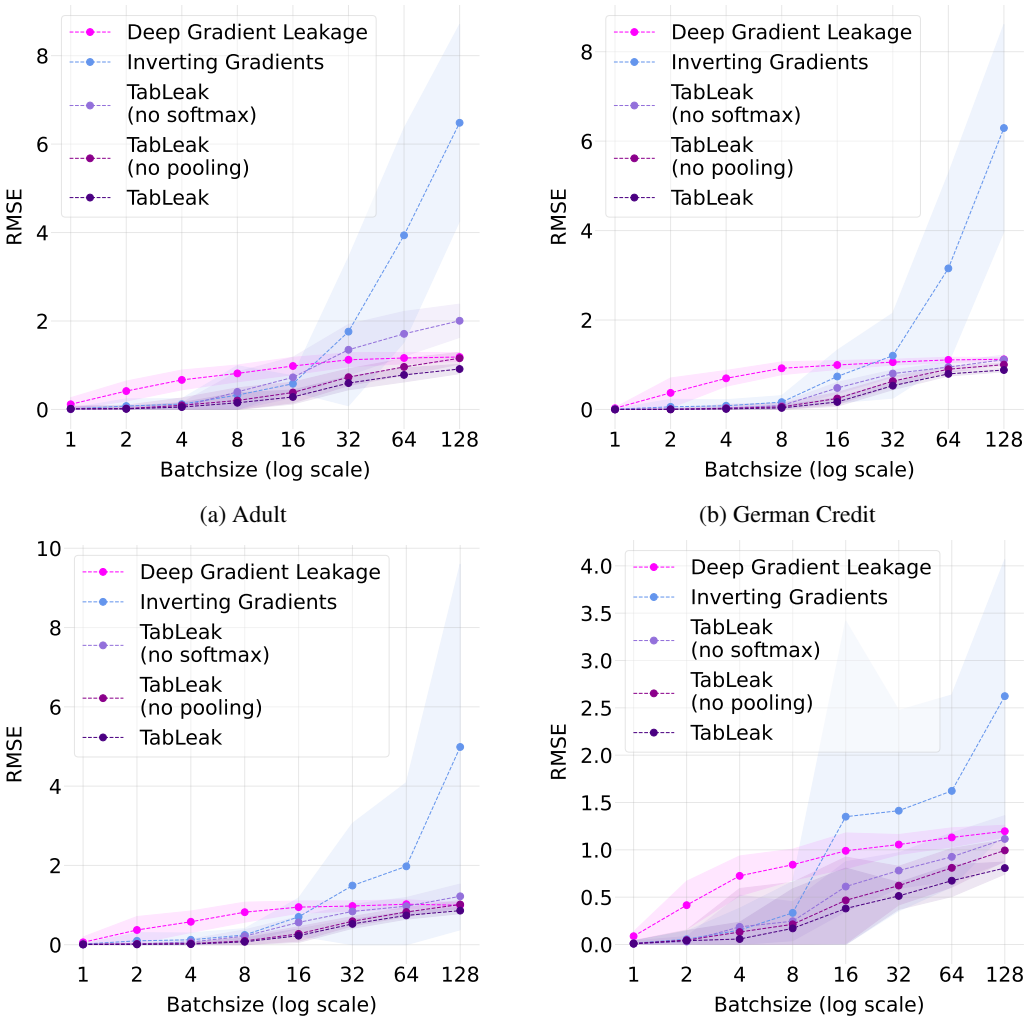
As our results in Fig. 11 demonstrate, TabLeak achieves significantly lower RMSE than Inverting Gradients (Geiping et al., 2020) on large batch sizes, for all four datasets examined. This indicates that the strong results obtained by TabLeak in the rest of the paper are not a consequence of our evaluation metric.

C.5. Attacking High-Dimensional Datasets

To understand how gradient inversion attacks scale with the number of features and the encoded dimension of the dataset, we attack a synthetic dataset with 125 discrete and 125 continuous features (generated with the procedure explained in App. E.1), resulting in 1231 dimensions when encoded, *i.e.,* around $12\times$ the dimension of Adult. We attack this setup both with TabLeak and the baseline attack of Geiping et al. (2020), reporting our results in Tab. 14, for batch sizes 16, 32, and 64. We can observe that while it is significantly harder to obtain high accuracy from such a high-dimensional dataset, TabLeak still strongly outperforms both the baseline and random guessing, achieving at least $1.68\times$ higher accuracy.

Table 14: The mean and standard deviation of the attack accuracy [%] on a synthetic dataset with 125 discrete and 125 continuous features (1231 dimensions one-hot encoded).

Batch Size	TabLeak	Inverting Gradients Geiping et al. (2020)	Random
16	$\mathbf{70.0 \pm 4.1}$	39.5 ± 3.2	19.6 ± 0.6
32	$\mathbf{55.0 \pm 2.3}$	30.6 ± 2.1	20.1 ± 0.5
64	$\mathbf{40.0 \pm 1.1}$	23.8 ± 1.2	20.7 ± 0.4



(a) Adult

(b) German Credit

(c) Lawschool Admissions

(d) Health Heritage

Figure 11: The mean and standard deviation of the Root Mean Squared Error (RMSE) of the reconstructions of the continuous features on all four datasets over batch sizes.

C.6. Attacking During Training

We evaluate how TabLeak and the baseline attack of Geiping et al. (2020) perform when attacking a network not only at initialization, but after some epochs of federated training have already been conducted. We expect to confirm the findings of prior works (Geiping et al., 2020; Dimitrov et al., 2022b), *i.e.,* that training degrades the performance of the attacks. Indeed, looking at our results collected in Tab. 15, we can observe that on all four datasets training negatively impacts the attack success. Moreover, we confirm the partial observation made already on Adult in the main body of this paper, namely that TabLeak preserves a high performance further into the training than the baseline attack, reinforcing the significance of the improvement TabLeak brings over prior methods.

Training Epochs	TabLeak	Inverting Gradients Geiping et al. (2020)
1	$\mathbf{79.1 \pm 4.2}$	67.8 ± 2.1
5	$\mathbf{76.4 \pm 5.7}$	64.5 ± 3.8
10	$\mathbf{74.5 \pm 5.7}$	60.9 ± 3.7
15	$\mathbf{64.5 \pm 7.1}$	57.8 ± 4.0

(a) Adult

Training Epochs	TabLeak	Inverting Gradients Geiping et al. (2020)
1	$\mathbf{94.2 \pm 4.4}$	78.9 ± 4.0
5	$\mathbf{92.6 \pm 5.0}$	77.5 ± 5.0
10	$\mathbf{92.2 \pm 3.7}$	76.0 ± 4.6
15	$\mathbf{89.7 \pm 4.0}$	72.4 ± 4.1

(b) German Credit

Training Epochs	TabLeak	Inverting Gradients Geiping et al. (2020)
1	$\mathbf{85.7 \pm 4.2}$	71.4 ± 2.2
5	$\mathbf{78.0 \pm 4.6}$	70.1 ± 3.4
10	$\mathbf{74.9 \pm 2.7}$	68.3 ± 3.5
15	$\mathbf{74.9 \pm 3.1}$	66.7 ± 3.9

(c) Lawschool Admissions

Training Epochs	TabLeak	Inverting Gradients Geiping et al. (2020)
1	$\mathbf{69.5 \pm 4.5}$	58.0 ± 4.2
5	$\mathbf{69.3 \pm 5.3}$	54.8 ± 4.5
10	$\mathbf{64.8 \pm 6.0}$	52.1 ± 3.1
15	$\mathbf{62.3 \pm 6.9}$	50.7 ± 3.3

(d) Health Heritage

Table 15: The mean and standard deviation of the attack accuracy [%] attacking a network at the 1st, 5th, 10th, and 15th epoch of FedSGD training on batch size 32.

C.7. Impact of Attack Iterations

We conduct an ablation study over the attack iterations to understand its influence on the attack success. In all the presented experiments we chose to run the attacks for 1 500 iterations before reporting our results; our goal here is to understand how this choice influenced our results. Tab. 16 shows that although the baseline attack is faster to converge, it tops out at a significantly lower accuracy level than TabLeak on all datasets, while TabLeak manages to improve on the accuracy presented in the paper on most cases, by allowing for more iterations. This further underlines the significant performance difference between TabLeak and prior gradient inversion attacks.

Attack Iterations	TabLeak	Inverting Gradients Geiping et al. (2020)
10	46.5 ± 1.44	60.5 ± 1.37
500	65.8 ± 3.86	66.8 ± 2.35
1 500	78.8 ± 4.25	66.4 ± 2.57
10 000	$\mathbf{83.7 \pm 2.91}$	67.6 ± 2.81

(a) Adult

Attack Iterations	TabLeak	Inverting Gradients Geiping et al. (2020)
10	55.4 ± 1.62	62.0 ± 1.14
500	79.2 ± 2.51	68.4 ± 2.28
1 500	82.1 ± 2.56	68.7 ± 2.75
10 000	$\mathbf{84.7 \pm 2.92}$	68.8 ± 3.18

(b) German Credit

Attack Iterations	TabLeak	Inverting Gradients Geiping et al. (2020)
10	61.0 ± 1.80	65.7 ± 1.81
500	78.1 ± 3.33	71.8 ± 2.42
1 500	$\mathbf{86.9 \pm 3.49}$	71.7 ± 1.99
10 000	86.4 ± 4.29	71.8 ± 2.53

(c) Lawschool Admissions

Attack Iterations	TabLeak	Inverting Gradients Geiping et al. (2020)
10	29.8 ± 1.31	51.9 ± 1.79
500	59.6 ± 3.57	57.5 ± 3.76
1 500	$\mathbf{71.3 \pm 5.25}$	57.6 ± 4.76
10 000	$\mathbf{71.3 \pm 3.65}$	54.0 ± 6.12

(d) Health Heritage

Table 16: The mean and standard deviation of the attack accuracy [%] on batch size 32 over attack iterations.

Table 17: The mean inversion accuracy [%] and standard deviation of different methods over varying batch sizes with given true labels (top) and with reconstructed labels (bottom) on the **Adult** dataset.

Label	Batch Size	TabLeak	TabLeak (no pooling)	TabLeak (no softmax)	Inverting Gradients Geiping et al. (2020)	Deep Gradient Leakage Zhu et al. (2019)	Random
True y	1	99.4 ± 2.8	99.1 ± 4.4	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$	97.0 ± 9.3	43.3 ± 11.8
	2	99.3 ± 5.0	99.2 ± 5.5	$\mathbf{99.6 \pm 1.3}$	97.6 ± 6.9	77.5 ± 12.8	47.1 ± 7.9
	4	98.1 ± 4.7	96.6 ± 7.8	$\mathbf{98.7 \pm 3.4}$	96.4 ± 7.2	65.3 ± 8.2	49.8 ± 4.9
	8	$\mathbf{95.2 \pm 8.8}$	92.5 ± 11.8	91.3 ± 7.1	91.1 ± 7.3	61.2 ± 4.7	53.9 ± 4.4
	16	$\mathbf{89.9 \pm 7.3}$	85.3 ± 9.7	79.0 ± 4.0	75.0 ± 5.2	60.2 ± 3.3	55.1 ± 3.9
	32	$\mathbf{79.3 \pm 4.5}$	74.3 ± 4.5	70.8 ± 3.3	66.6 ± 3.5	60.8 ± 1.9	58.0 ± 2.9
	64	$\mathbf{73.4 \pm 3.0}$	68.9 ± 3.1	67.3 ± 3.2	62.5 ± 3.1	61.3 ± 1.4	59.0 ± 3.2
	128	$\mathbf{71.4 \pm 1.2}$	67.4 ± 1.4	65.2 ± 2.1	59.5 ± 2.1	62.9 ± 1.0	61.2 ± 3.1
Rec. $\hat{y}$	1	99.4 ± 2.8	99.3 ± 3.6	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$	98.9 ± 2.6	43.3 ± 11.8
	2	98.1 ± 9.6	98.1 ± 9.6	$\mathbf{98.7 \pm 7.1}$	95.9 ± 11.5	77.9 ± 14.1	47.1 ± 7.9
	4	89.6 ± 13.5	87.8 ± 15.3	$\mathbf{89.8 \pm 13.0}$	87.9 ± 13.7	58.1 ± 12.4	49.8 ± 4.9
	8	$\mathbf{86.7 \pm 12.2}$	83.8 ± 13.6	82.7 ± 10.5	83.3 ± 9.7	56.1 ± 5.4	53.9 ± 4.4
	16	$\mathbf{83.0 \pm 7.7}$	78.6 ± 8.1	76.4 ± 5.4	73.0 ± 3.5	57.2 ± 3.4	55.1 ± 3.9
	32	$\mathbf{76.9 \pm 4.8}$	72.4 ± 4.8	68.9 ± 4.2	66.3 ± 3.4	58.4 ± 2.5	58.0 ± 2.9
	64	$\mathbf{72.8 \pm 3.3}$	68.5 ± 3.5	66.8 ± 2.9	63.1 ± 3.2	60.1 ± 1.7	59.0 ± 3.2
	128	$\mathbf{71.4 \pm 1.3}$	67.5 ± 1.5	65.0 ± 2.2	59.5 ± 2.1	62.3 ± 1.0	61.2 ± 3.1

D. All Main Results

In this subsection, we include all the results presented in the main part of this paper for the Adult dataset alongside with the corresponding additional results on the German Credit, Lawschool Admissions, and the Health Heritage datasets.

D.1. Full FedSGD Results on all Datasets

In Tab. 17, Tab. 18, Tab. 19, and Tab. 20 we provide the full attack results of our method compared to Inverting Gradients (Geiping et al., 2020) and the random baseline on the Adult, German Credit, Lawschool Admissions, and Health Heritage datasets, respectively. Looking at the results for all datasets, we can confirm the observations made in Sec. 4, *i.e.,* (i) the lower batch sizes are vulnerable to any non-trivial attack, (ii) not knowing the ground truth labels does not significantly disadvantage the attacker for larger batch sizes, and (iii) TabLeak provides a strong improvement over the baselines for practically relevant batch sizes over all datasets examined.

D.2. Categorical vs. Continuous Features on all Datasets

In Fig. 12, we compare the reconstruction accuracy of the continuous and the discrete features on all four datasets. We confirm our observations, shown in Fig. 4 in the main text, that a strong dichotomy between continuous and discrete feature reconstruction accuracy exists on all 4 datasets.

D.3. Federated Averaging Results on all Datasets

In Tab. 21, Tab. 22, Tab. 23, and Tab. 24 we present our results on attacking the clients in FedAvg training on the Adult, German Credit, Lawschool Submissions, and Health Heritage datasets, respectively. We described the details of the experiment in App. B above. Confirming our conclusions drawn in the main part of this manuscript, we observe that TabLeak achieves non-trivial reconstruction accuracy over all settings and even for large numbers of updates, while the baseline attack often fails to outperform random guessing, when the number of local updates is increased.

Table 18: The mean inversion accuracy [%] and standard deviation of different methods over varying batch sizes with given true labels (top) and with reconstructed labels (bottom) on the **German Credit** dataset.

Label	Batch Size	TabLeak	TabLeak (no pooling)	TabLeak (no softmax)	Inverting Gradients Geiping et al. (2020)	Deep Gradient Leakage Zhu et al. (2019)	Random
True y	1	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$	100.0 ± 0.0	43.9 ± 9.8
	2	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$	99.9 ± 0.4	98.0 ± 7.1	84.2 ± 14.9	45.1 ± 6.6
	4	$\mathbf{99.9 \pm 0.4}$	99.5 ± 2.8	99.6 ± 1.2	97.8 ± 6.0	71.0 ± 6.9	50.3 ± 4.5
	8	$\mathbf{99.6 \pm 1.2}$	99.1 ± 2.4	98.4 ± 2.2	96.1 ± 5.2	64.1 ± 2.7	51.8 ± 3.2
	16	$\mathbf{96.3 \pm 3.4}$	93.7 ± 4.5	85.1 ± 3.6	79.3 ± 4.4	63.1 ± 2.1	54.5 ± 3.0
	32	$\mathbf{84.2 \pm 2.8}$	80.1 ± 3.2	72.8 ± 1.9	69.7 ± 2.2	63.4 ± 1.4	56.8 ± 2.2
	64	$\mathbf{74.4 \pm 1.4}$	71.8 ± 1.6	69.7 ± 1.3	66.6 ± 1.8	64.0 ± 1.0	59.4 ± 1.9
	128	$\mathbf{72.3 \pm 0.9}$	69.8 ± 0.7	68.4 ± 1.5	64.5 ± 1.5	65.3 ± 0.7	61.0 ± 2.1
Rec. $\hat{y}$	1	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$	99.1 ± 6.3	43.9 ± 9.8
	2	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$	99.9 ± 0.4	98.8 ± 5.2	86.2 ± 14.3	45.1 ± 6.6
	4	$\mathbf{99.5 \pm 3.2}$	98.7 ± 4.7	99.3 ± 2.9	97.4 ± 6.4	73.0 ± 7.4	50.3 ± 4.5
	8	$\mathbf{97.2 \pm 6.3}$	96.1 ± 7.6	96.2 ± 6.4	94.8 ± 6.5	63.5 ± 4.8	51.8 ± 3.2
	16	$\mathbf{92.0 \pm 6.5}$	90.0 ± 6.6	83.5 ± 4.0	77.9 ± 4.6	61.4 ± 2.9	54.5 ± 3.0
	32	$\mathbf{81.9 \pm 3.4}$	78.4 ± 3.4	71.8 ± 1.9	69.1 ± 2.1	62.1 ± 1.4	56.8 ± 2.2
	64	$\mathbf{73.8 \pm 1.5}$	71.4 ± 1.3	69.5 ± 1.2	66.5 ± 1.7	63.5 ± 1.0	59.4 ± 1.9
	128	$\mathbf{72.3 \pm 0.9}$	69.8 ± 0.7	68.2 ± 1.6	64.4 ± 1.6	65.0 ± 0.6	61.0 ± 2.1

Table 19: The mean inversion accuracy [%] and standard deviation of different methods over varying batch sizes with given true labels (top) and with reconstructed labels (bottom) on the **Lawschool Admissions** dataset.

Label	Batch Size	TabLeak	TabLeak (no pooling)	TabLeak (no softmax)	Inverting Gradients Geiping et al. (2020)	Deep Gradient Leakage Zhu et al. (2019)	Random
True y	1	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$	97.7 ± 9.6	38.9 ± 14.6
	2	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$	99.9 ± 1.0	96.3 ± 10.4	84.6 ± 16.7	38.4 ± 11.5
	4	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$	99.6 ± 2.1	97.6 ± 6.9	76.6 ± 12.5	43.2 ± 7.2
	8	$\mathbf{98.9 \pm 3.8}$	98.5 ± 4.5	95.6 ± 5.1	94.5 ± 5.8	68.5 ± 5.4	49.4 ± 4.6
	16	$\mathbf{95.0 \pm 5.8}$	93.2 ± 6.4	81.3 ± 4.4	77.3 ± 5.5	65.8 ± 3.2	53.0 ± 3.1
	32	$\mathbf{84.9 \pm 4.0}$	82.0 ± 3.7	73.1 ± 2.5	71.0 ± 2.8	67.9 ± 2.3	57.6 ± 2.3
	64	$\mathbf{78.1 \pm 2.17}$	76.6 ± 2.2	73.0 ± 2.1	71.7 ± 2.2	70.4 ± 1.4	60.4 ± 2.2
	128	$\mathbf{77.2 \pm 1.1}$	75.9 ± 1.2	73.5 ± 2.8	71.8 ± 2.7	73.4 ± 0.9	63.4 ± 1.5
Rec. $\hat{y}$	1	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$	100.0 ± 0.0	38.9 ± 14.6
	2	$\mathbf{99.1 \pm 6.0}$	$\mathbf{99.1 \pm 6.0}$	98.7 ± 8.0	95.9 ± 12.0	84.3 ± 17.2	38.4 ± 11.5
	4	$\mathbf{99.5 \pm 3.5}$	99.1 ± 4.3	98.7 ± 5.9	96.8 ± 8.5	79.9 ± 12.6	43.2 ± 7.2
	8	$\mathbf{95.9 \pm 7.8}$	$\mathbf{95.9 \pm 8.0}$	93.4 ± 7.8	91.9 ± 7.9	66.9 ± 7.2	49.4 ± 4.6
	16	$\mathbf{91.2 \pm 7.4}$	88.5 ± 8.7	80.4 ± 5.0	77.4 ± 5.4	64.9 ± 3.9	53.0 ± 3.1
	32	$\mathbf{83.1 \pm 4.2}$	81.2 ± 4.5	72.9 ± 2.5	71.0 ± 2.0	66.1 ± 2.0	57.6 ± 2.3
	64	$\mathbf{77.3 \pm 2.2}$	75.9 ± 2.0	72.5 ± 2.1	71.5 ± 2.4	69.3 ± 1.2	60.4 ± 2.2
	128	$\mathbf{77.0 \pm 1.1}$	75.8 ± 1.2	73.8 ± 2.5	71.8 ± 2.8	72.8 ± 1.0	63.4 ± 1.5

Table 20: The mean inversion accuracy [%] and standard deviation of different methods over varying batch sizes with given true labels (top) and with reconstructed labels (bottom) on the **Health Heritage** dataset.

Label	Batch Size	TabLeak	TabLeak (no pooling)	TabLeak (no softmax)	Inverting Gradients Geiping et al. (2020)	Deep Gradient Leakage Zhu et al. (2019)	Random
True y	1	$\mathbf{99.8 \pm 1.6}$	$\mathbf{99.8 \pm 1.6}$	$\mathbf{99.8 \pm 1.6}$	$\mathbf{99.8 \pm 1.6}$	97.3 ± 6.4	34.8 ± 13.1
	2	97.6 ± 7.9	97.1 ± 9.3	$\mathbf{98.9 \pm 3.3}$	97.9 ± 5.6	70.7 ± 16.9	36.9 ± 9.8
	4	$\mathbf{97.9 \pm 7.7}$	96.4 ± 10.8	95.4 ± 7.8	95.6 ± 8.1	52.0 ± 7.2	37.0 ± 5.3
	8	$\mathbf{95.5 \pm 9.2}$	93.1 ± 11.5	88.6 ± 10.8	86.2 ± 9.0	50.1 ± 3.9	39.2 ± 3.8
	16	$\mathbf{85.4 \pm 9.9}$	79.8 ± 10.5	68.3 ± 5.3	63.6 ± 5.5	50.8 ± 2.2	41.4 ± 3.7
	32	$\mathbf{70.8 \pm 4.5}$	65.5 ± 4.2	61.8 ± 4.0	57.7 ± 4.1	51.6 ± 1.7	43.4 ± 2.8
	64	$\mathbf{65.5 \pm 2.8}$	61.3 ± 2.7	61.2 ± 4.4	57.4 ± 4.7	54.0 ± 1.5	45.0 ± 3.7
	128	$\mathbf{63.5 \pm 1.7}$	59.3 ± 1.6	58.6 ± 4.4	55.6 ± 4.8	55.4 ± 0.8	46.8 ± 3.2
Rec. $\hat{y}$	1	$\mathbf{99.8 \pm 1.6}$	$\mathbf{99.8 \pm 1.6}$	$\mathbf{99.8 \pm 1.6}$	99.6 ± 2.5	96.1 ± 7.7	34.8 ± 13.1
	2	95.1 ± 14.2	95.3 ± 13.7	$\mathbf{95.4 \pm 14.2}$	92.5 ± 16.9	68.2 ± 19.6	36.9 ± 9.8
	4	$\mathbf{86.3 \pm 21.1}$	85.0 ± 22.2	83.4 ± 19.6	83.5 ± 20.7	48.2 ± 11.7	37.0 ± 5.3
	8	$\mathbf{81.5 \pm 16.0}$	77.6 ± 16.9	76.8 ± 13.4	74.5 ± 13.8	48.0 ± 6.3	39.2 ± 3.8
	16	$\mathbf{75.3 \pm 13.1}$	71.2 ± 12.9	65.2 ± 7.6	60.9 ± 6.3	49.6 ± 4.5	41.4 ± 3.7
	32	$\mathbf{65.5 \pm 5.6}$	62.0 ± 5.2	60.3 ± 4.1	56.9 ± 4.0	51.0 ± 3.2	43.4 ± 2.8
	64	$\mathbf{63.9 \pm 2.9}$	60.9 ± 2.5	61.2 ± 4.4	57.7 ± 4.7	53.7 ± 1.3	45.0 ± 3.7
	128	$\mathbf{63.8 \pm 1.8}$	60.3 ± 1.9	58.8 ± 4.7	55.7 ± 5.0	55.4 ± 0.9	46.8 ± 3.2

Table 21: Mean and standard deviation of the inversion accuracy [%] with local dataset size of 32 in FedAvg training on the **Adult** dataset. The accuracy of the random baseline for 32 datapoints is 58.0 ± 2.9.

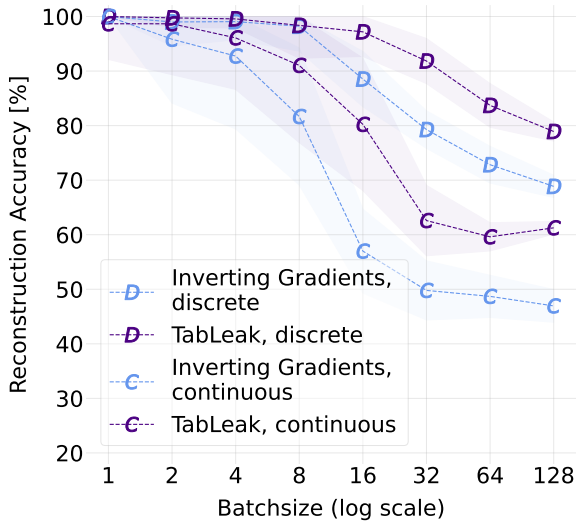
	TabLeak			Inverting Gradients (Geiping et al., 2020)		
n. batches	1 epoch	5 epochs	10 epochs	1 epoch	5 epochs	10 epochs
1	$\mathbf{80.7 \pm 3.8}$	$\mathbf{75.8 \pm 3.3}$	$\mathbf{72.8 \pm 3.2}$	65.2 ± 2.7	56.1 ± 4.1	53.2 ± 4.2
2	$\mathbf{79.2 \pm 4.2}$	$\mathbf{75.6 \pm 2.7}$	$\mathbf{73.1 \pm 5.0}$	64.8 ± 3.3	56.4 ± 4.8	56.2 ± 4.8
4	$\mathbf{79.7 \pm 3.6}$	$\mathbf{76.2 \pm 3.0}$	$\mathbf{73.7 \pm 3.6}$	64.8 ± 3.4	58.7 ± 4.6	56.6 ± 5.0

Table 22: Mean and standard deviation of the inversion accuracy [%] with local dataset size of 32 in FedAvg training on the **German Credit** dataset. The accuracy of the random baseline for 32 datapoints is 56.9 ± 2.1.

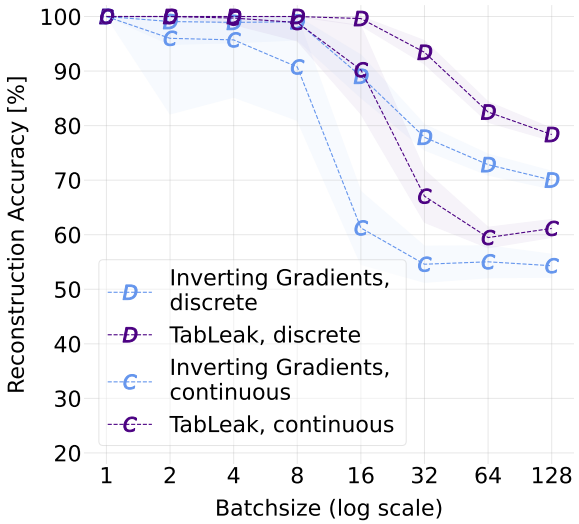
	TabLeak			Inverting Gradients (Geiping et al., 2020)		
n. batches	1 epoch	5 epochs	10 epochs	1 epoch	5 epochs	10 epochs
1	$\mathbf{96.0 \pm 3.4}$	$\mathbf{87.3 \pm 8.2}$	$\mathbf{85.9 \pm 6.2}$	78.2 ± 4.6	65.4 ± 6.2	62.5 ± 6.1
2	$\mathbf{96.2 \pm 3.0}$	$\mathbf{87.2 \pm 5.4}$	$\mathbf{85.4 \pm 9.0}$	78.3 ± 5.8	68.8 ± 6.6	63.4 ± 4.8
4	$\mathbf{96.1 \pm 3.6}$	$\mathbf{85.3 \pm 8.0}$	$\mathbf{83.8 \pm 8.1}$	79.2 ± 4.9	67.4 ± 4.8	62.6 ± 6.5

Table 23: Mean and standard deviation of the inversion accuracy [%] with local dataset size of 32 in FedAvg training on the **Lawschool Admissions** dataset. The accuracy of the random baseline for 32 datapoints is 57.8 ± 2.3.

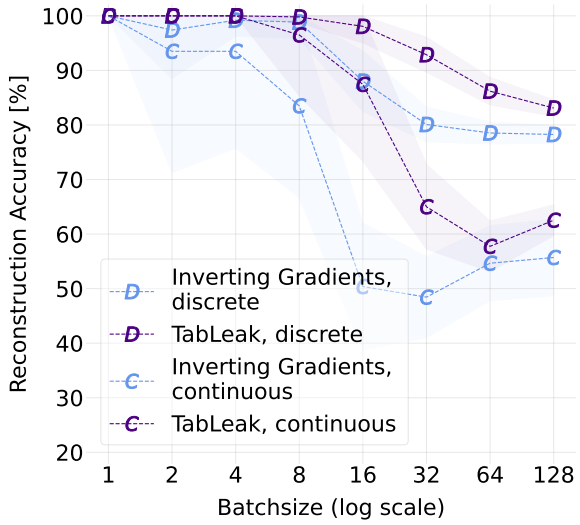
	TabLeak			Inverting Gradients (Geiping et al., 2020)		
n. batches	1 epoch	5 epochs	10 epochs	1 epoch	5 epochs	10 epochs
1	$\mathbf{85.4 \pm 4.2}$	$\mathbf{82.9 \pm 3.1}$	$\mathbf{81.7 \pm 4.0}$	72.2 ± 2.6	68.1 ± 3.1	65.2 ± 2.8
2	$\mathbf{86.2 \pm 4.3}$	$\mathbf{82.8 \pm 3.0}$	$\mathbf{81.4 \pm 3.1}$	72.5 ± 1.9	68.3 ± 4.4	66.2 ± 2.8
4	$\mathbf{85.7 \pm 4.4}$	$\mathbf{81.5 \pm 3.8}$	$\mathbf{80.3 \pm 4.5}$	72.5 ± 2.4	69.4 ± 3.9	67.9 ± 3.8



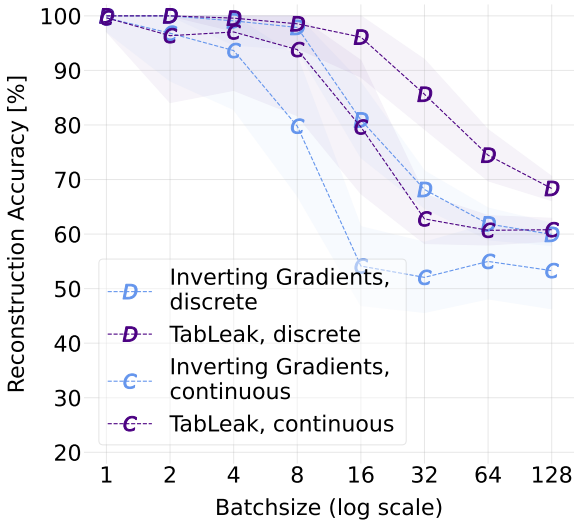
(a) Adult



(b) German Credit



(c) Lawschool Admissions



(d) Health Heritage

Figure 12: Mean reconstruction accuracy curves with corresponding standard deviations over varying batch size, separately for the discrete and the continuous features on all four datasets.

Table 24: Mean and standard deviation of the inversion accuracy [%] with local dataset size of 32 in FedAvg training on the **Health Heritage** dataset. The accuracy of the random baseline for 32 datapoints is 43.4 ± 3.5.

	TabLeak			Inverting Gradients (Geiping et al., 2020)		
n. batches	1 epoch	5 epochs	10 epochs	1 epoch	5 epochs	10 epochs
1	$\mathbf{68.5 \pm 5.0}$	$\mathbf{62.2 \pm 3.5}$	$\mathbf{57.4 \pm 3.0}$	53.8 ± 5.5	41.4 ± 3.6	41.1 ± 3.4
2	$\mathbf{68.1 \pm 4.9}$	$\mathbf{62.4 \pm 4.1}$	$\mathbf{57.0 \pm 2.8}$	52.4 ± 5.7	43.4 ± 4.28	44.4 ± 4.3
4	$\mathbf{67.3 \pm 5.8}$	$\mathbf{62.0 \pm 3.5}$	$\mathbf{57.0 \pm 3.0}$	52.5 ± 6.6	43.4 ± 5.7	44.8 ± 4.4

Table 25: The mean accuracy [%] and entropies with the corresponding standard deviations over batch sizes of the categorical and the continuous features on the **Adult** dataset, together with the rank correlation between mean batch accuracy and mean batch entropy at the given batch size.

	Discrete			Continuous		
	Accuracy	Entropy	Kendall's τ	Accuracy	Entropy	Kendall's τ
1	100.0 ± 0.0	0.02 ± 0.04	NaN	98.7 ± 6.5	-4.00 ± 0.72	-0.28
2	99.8 ± 1.7	0.02 ± 0.05	-0.20	98.7 ± 9.3	-3.75 ± 0.93	-0.20
4	99.6 ± 1.7	0.08 ± 0.11	-0.38	96.2 ± 9.0	-2.61 ± 1.34	-0.56
8	98.3 ± 6.1	0.15 ± 0.14	-0.52	91.0 ± 14.3	-1.69 ± 1.14	-0.66
16	97.2 ± 4.3	0.25 ± 0.11	-0.64	80.0 ± 12.9	-0.63 ± 0.62	-0.65
32	91.5 ± 4.1	0.39 ± 0.06	-0.55	63.1 ± 6.7	0.17 ± 0.31	-0.53
64	83.7 ± 3.7	0.47 ± 0.04	-0.65	59.6 ± 2.5	0.57 ± 0.22	-0.34
128	79.2 ± 1.6	0.51 ± 0.03	-0.42	61.3 ± 1.6	0.80 ± 0.14	-0.34

Table 26: The mean accuracy [%] and entropies with the corresponding standard deviations over batch sizes of the categorical and the continuous features on the **German Credit** dataset, together with the rank correlation between mean batch accuracy and mean batch entropy at the given batch size.

	Discrete			Continuous		
	Accuracy	Entropy	Kendall's τ	Accuracy	Entropy	Kendall's τ
1	100.0 ± 0.0	0.00 ± 0.01	NaN	100.0 ± 0.0	-4.81 ± 0.62	NaN
2	100.0 ± 0.0	0.02 ± 0.03	NaN	100.0 ± 0.0	-4.12 ± 1.36	NaN
4	100.0 ± 0.0	0.06 ± 0.05	NaN	99.7 ± 1.2	-2.75 ± 1.33	-0.19
8	100.0 ± 0.0	0.11 ± 0.07	NaN	98.9 ± 3.4	-1.92 ± 1.01	-0.23
16	99.6 ± 1.3	0.24 ± 0.08	-0.35	90.1 ± 8.0	-0.74 ± 0.33	-0.38
32	93.4 ± 2.2	0.42 ± 0.04	-0.52	67.3 ± 4.8	0.28 ± 0.17	-0.28
64	82.5 ± 1.8	0.55 ± 0.02	-0.64	59.3 ± 2.1	0.80 ± 0.07	-0.29
128	78.5 ± 1.1	0.58 ± 0.02	-0.27	61.2 ± 1.3	1.01 ± 0.04	-0.25

D.4. Full Results on Entropy on all Datasets

In Tab. 25, Tab. 26, Tab. 27, and Tab. 28 we provide the mean and standard deviation of the reconstruction accuracy and the entropy of the continuous and the categorical features over increasing batch size for attacking with TabLeak on the four datasets. Additionally, at each batch size we calculate and report the Kendall's τ rank correlation coefficient (Kendall, 1938) between the mean entropy of the features and the mean accuracy of the features over different batches. Note that if all features are correctly reconstructed, we can not calculate a rank correlation, in these cases we replace the missing value by NaN. We can observe on all datasets a trend of increasing entropy over decreasing reconstruction accuracy as the batch size is increased; and as such providing a signal to the attacker about their overall reconstruction success.

To generalize our results presented in Sec. 4 beyond Adult, we present the corresponding tables on the top and bottom quarters of the data based on the entropy ranking in Tab. 29, Tab. 30, Tab. 31, and Tab. 32 for all four datasets, respectively. We can observe that the entropy is still effective in separating the poorly reconstructed features from the well reconstructed ones. The scheme is especially strong on the categorical features, which is concerning, because, as discussed in Sec. 4, they are already much more vulnerable to leakage attacks.

Table 27: The mean accuracy [%] and entropies with the corresponding standard deviations over batch sizes of the categorical and the continuous features on the **Lawschool Admissions** dataset, together with the rank correlation between mean batch accuracy and mean batch entropy at the given batch size.

	Discrete			Continuous		
	Accuracy	Entropy	Kendall's τ	Accuracy	Entropy	Kendall's τ
1	100.0 ± 0.0	0.01 ± 0.02	NaN	100.0 ± 0.0	-2.94 ± 0.32	NaN
2	100.0 ± 0.0	0.02 ± 0.05	NaN	100.0 ± 0.0	-2.52 ± 0.67	NaN
4	100.0 ± 0.0	0.03 ± 0.04	NaN	100.0 ± 0.0	-2.25 ± 0.53	NaN
8	99.8 ± 1.1	0.10 ± 0.10	-0.28	96.5 ± 11.1	-1.66 ± 0.47	-0.25
16	98.4 ± 2.8	0.23 ± 0.11	-0.37	87.6 ± 14.0	-0.62 ± 0.42	-0.50
32	93.4 ± 2.9	0.42 ± 0.08	-0.50	65.3 ± 8.5	0.21 ± 0.20	-0.45
64	86.9 ± 2.5	0.55 ± 0.05	-0.51	58.4 ± 4.6	0.80 ± 0.11	-0.20
128	83.5 ± 1.6	0.60 ± 0.03	-0.36	62.7 ± 3.1	1.06 ± 0.10	-0.18

Table 28: The mean accuracy [%] and entropies with the corresponding standard deviations over batch sizes of the categorical and the continuous features on the **Health Heritage** dataset, together with the rank correlation between mean batch accuracy and mean batch entropy at the given batch size.

	Discrete			Continuous		
	Accuracy	Entropy	Kendall's τ	Accuracy	Entropy	Kendall's τ
1	100.0 ± 0.0	0.02 ± 0.04	NaN	99.6 ± 2.5	-3.45 ± 0.70	-0.18
2	100.0 ± 0.0	0.05 ± 0.09	NaN	96.4 ± 12.3	-2.88 ± 0.97	-0.47
4	99.6 ± 2.4	0.08 ± 0.10	-0.28	97.0 ± 10.6	-1.92 ± 0.96	-0.34
8	98.4 ± 5.2	0.13 ± 0.11	-0.43	93.9 ± 11.8	-1.19 ± 0.76	-0.56
16	96.0 ± 8.5	0.26 ± 0.10	-0.61	79.5 ± 12.1	-0.25 ± 0.49	-0.55
32	85.8 ± 5.9	0.42 ± 0.06	-0.62	63.2 ± 4.3	0.47 ± 0.24	-0.48
64	73.9 ± 4.5	0.50 ± 0.04	-0.60	60.6 ± 2.9	0.78 ± 0.20	-0.48
128	68.1 ± 2.0	0.55 ± 0.02	-0.26	61.0 ± 2.3	1.03 ± 0.11	-0.42

Table 29: The mean and standard deviation of the accuracy [%] of each feature type in the top 25% and the bottom 25% when ranked in the batch according to the entropy on the **Adult** dataset.

Batch	Categorical		Continuous	
Size	Top 25%	Bottom 25%	Top 25%	Bottom 25%
1	100.0 ± 0.0	100.0 ± 0.0	99.0 ± 7.0	98.0 ± 9.8
2	100.0 ± 0.0	99.3 ± 4.7	98.7 ± 9.3	98.7 ± 9.3
4	100.0 ± 0.0	98.0 ± 7.2	99.7 ± 2.3	92.7 ± 18.6
8	99.7 ± 2.3	96.0 ± 10.3	97.2 ± 9.4	84.2 ± 21.6
16	99.9 ± 0.6	89.8 ± 13.0	98.2 ± 3.5	60.8 ± 19.9
32	99.1 ± 2.6	75.5 ± 8.0	94.2 ± 4.7	43.6 ± 8.2
64	97.8 ± 2.6	66.1 ± 5.3	92.9 ± 3.5	41.3 ± 5.8
128	94.3 ± 1.9	62.8 ± 3.8	93.5 ± 2.3	42.2 ± 3.7

Table 30: The mean and standard deviation of the accuracy [%] of each feature type in the top 25% and the bottom 25% when ranked in the batch according to the entropy **German Credit** dataset.

Batch	Categorical		Continuous	
Size	Top 25%	Bottom 25%	Top 25%	Bottom 25%
1	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
2	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
4	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	98.9 ± 4.8
8	100.0 ± 0.0	100.0 ± 0.0	99.7 ± 2.0	97.7 ± 5.4
16	100.0 ± 0.0	97.8 ± 6.2	98.2 ± 3.5	80.6 ± 13.7
32	100.0 ± 0.0	75.2 ± 7.1	87.3 ± 5.9	51.2 ± 7.1
64	98.9 ± 1.3	66.7 ± 4.0	77.9 ± 5.6	47.2 ± 4.8
128	96.9 ± 1.4	66.0 ± 2.8	78.0 ± 3.0	48.9 ± 2.9

Table 31: The mean and standard deviation of the accuracy [%] of each feature type in the top 25% and the bottom 25% when ranked in the batch according to the entropy **Lawschool Admissions** dataset.

Batch	Categorical		Continuous	
Size	Top 25%	Bottom 25%	Top 25%	Bottom 25%
1	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
2	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
4	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
8	100.0 ± 0.0	98.5 ± 7.8	99.5 ± 3.5	91.5 ± 23.2
16	100.0 ± 0.0	89.2 ± 17.3	96.5 ± 10.9	75.2 ± 24.6
32	99.9 ± 0.9	77.5 ± 9.3	76.6 ± 11.9	53.4 ± 13.4
64	96.6 ± 3.5	76.0 ± 9.4	62.7 ± 9.7	53.0 ± 9.3
128	93.6 ± 3.1	70.5 ± 7.6	68.2 ± 5.9	56.6 ± 6.0

Table 32: The mean and standard deviation of the accuracy [%] of each feature type in the top 25% and the bottom 25% when ranked in the batch according to the entropy **Health Heritage** dataset.

Batch	Categorical		Continuous	
Size	Top 25%	Bottom 25%	Top 25%	Bottom 25%
1	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	98.7 ± 9.3
2	100.0 ± 0.0	100.0 ± 0.0	97.7 ± 10.5	94.3 ± 16.9
4	99.8 ± 1.3	99.3 ± 4.0	98.0 ± 10.8	95.8 ± 13.6
8	99.8 ± 1.3	96.7 ± 10.4	97.7 ± 5.8	88.5 ± 20.6
16	99.0 ± 5.4	92.7 ± 12.4	92.3 ± 9.3	64.8 ± 16.8
32	95.3 ± 3.8	75.9 ± 8.8	77.2 ± 6.7	51.1 ± 8.4
64	88.3 ± 4.5	59.6 ± 6.0	73.3 ± 5.6	52.3 ± 6.1
128	82.4 ± 2.3	54.5 ± 3.1	71.2 ± 4.3	56.3 ± 4.0

E. Studying Pooling

In this subsection, we present three further experiments on justifying and understanding our choices in pooling:

- Experiments on synthetic datasets for understanding the motivation for pooling in App. E.1.
- Ablation study on understanding the impact of the number of samples N in the collection before pooling on the performance of TabLeak in App. E.2.
- Comparison of using mean and median pooling on TabLeak in App. E.3.

Note that in the experiments below we do not make use of the sigmoid restricting the continuous features.

E.1. Variance Study

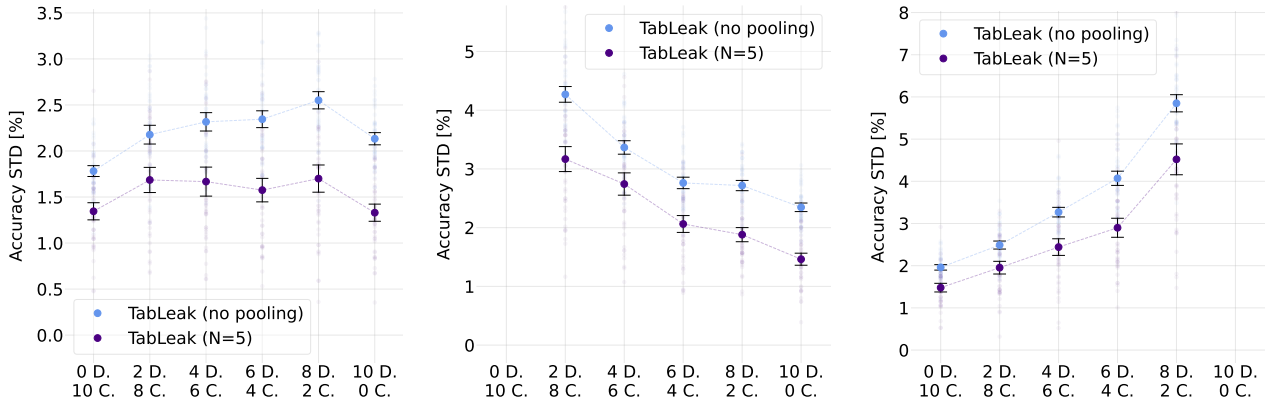
A unique challenge (challenge (i)) of tabular data leakage is that the mix of discrete and continuous features introduces further variance in the final reconstructions. As a solution to this challenge, we propose to produce N independent reconstructions of the same batch, and ensemble them using the pooling scheme described in Sec. 3.1.2. In this subsection, we provide empirical evidence for the subject of challenge (i) and the effectiveness of our proposed solution to it.

Experimental Setup We create 6 synthetic binary classification datasets, each with 10 features, however of varying modality. Concretely; we have the following setups:

- Synthetic dataset with 0 discrete and 10 continuous columns,
- Synthetic dataset with 2 discrete and 8 continuous columns,
- Synthetic dataset with 4 discrete and 6 continuous columns,
- Synthetic dataset with 6 discrete and 4 continuous columns,
- Synthetic dataset with 8 discrete and 2 continuous columns,
- Synthetic dataset with 10 discrete and 0 continuous columns.

The continuous features are Gaussians with means between 0 and 5, and standard deviations between 1 and 3. The discrete features have domain sizes between 2 and 6, and the probabilities are drawn randomly. On each of these datasets we sample 50 batches of size 32 and reconstruct them using TabLeak (no pooling) starting from 30 different initializations in the same experimental setup elaborated in Sec. 4 and in App. B. We then proceed to calculate the standard deviation of the accuracy for each of the 50 batches over their 30 independent reconstructions, providing us 50 statistically independent data points for understanding the variance in the non-pooled reconstruction problem. Further, from the 30 independent reconstructions of each batch, we build 6 independent mini-ensembles of size 5 and conduct median pooling on them (essentially, TabLeak with $N = 5$). We then measure the standard deviation of the error for each of the 50 batches over the 6 obtained pooled reconstructions, obtaining 50 independent data points for analyzing the variance of pooled reconstruction.

Results We present the results of the experiment in Fig. 13; additionally to measuring the same-batch reconstruction accuracy standard deviation for all features together, we also present the resulting measurements when only considering the discrete and the continuous features, respectively. The figures are organized such that the x-axis begins with the synthetic dataset consisting only of continuous features and progresses to the right by decreasing the number of continuous and increasing the number of discrete features at each step by 2. Roughly speaking, the very left column of the figures is similar to data leakage in the image domain, where all features are continuous, and the very right relates to data leakage in the text domain, containing only discrete features. Looking at Fig. 13a, we observe that the mean same-batch STD is indeed higher for datasets consisting of mixed types, providing empirical evidence underlining the first challenge of tabular data leakage. Further, it can be clearly seen that pooling, even with a small ensemble of just 5 samples, decisively decreases the variance of the reconstruction problem, providing strong justification for using pooling in the tabular setting. Finally, from Fig. 13b and Fig. 13c we gain interesting insight in the underlying dynamics of the interplay between discrete and continuous features at reconstruction. Concretely, we observe that as the presence of a given modality is decreasing and its



(a) Combined Accuracy STD (b) Discrete Accuracy STD (c) Continuous Accuracy STD

Figure 13: Mean same-batch reconstruction accuracy standard deviation and 90% confidence interval at batch size 32 estimated from 50 independent batches over synthetic datasets with varying number of discrete (D.) and continuous (C.) features.

Table 33: Reconstruction accuracy [%] and standard deviation of TabLeak on batch size 32 over the size of the ensemble N used for pooling.

$N =$	1	5	10	15	20	25	30
Adult	71.8 ± 4.6	75.1 ± 4.6	76.5 ± 4.6	76.5 ± 4.8	77.1 ± 4.7	77.0 ± 4.7	$\mathbf{77.4 \pm 4.8}$
German	79.0 ± 3.0	81.6 ± 2.9	82.6 ± 2.9	82.9 ± 2.9	83.2 ± 2.8	83.4 ± 2.8	$\mathbf{83.6 \pm 2.7}$
Lawschool	82.3 ± 4.0	84.4 ± 3.7	84.7 ± 4.1	85.2 ± 3.9	85.1 ± 3.9	$\mathbf{85.3 \pm 4.0}$	$\mathbf{85.3 \pm 4.0}$
Health Heritage	64.6 ± 4.2	67.5 ± 4.3	69.1 ± 4.4	69.1 ± 4.5	69.7 ± 4.5	69.5 ± 4.3	$\mathbf{70.1 \pm 4.0}$

place is taken up by the other, the recovery of this modality becomes increasingly noisier. Much in line with the observations on the difference in the recovery success between discrete and continuous features, these results also argue for future work to pursue methods that decrease the disparity between the two different feature types in the mixed setting.

E.2. The Impact of the Number of Samples N

In Tab. 33 we present the results of an ablation study we conducted on TabLeak at batch size 32 to understand the impact of the size of the ensemble N on the performance of the attack. We observed that with increasing N the performance of the attack gets steadily better, albeit, producing diminishing returns, showing signs of saturation on some datasets after $N = 25$. Note that this behavior is expected, and suggests using the largest N that is not yet computationally prohibitive. We chose $N = 30$ for all our experiments with TabLeak (unless explicitly stated otherwise); this allowed us to conduct large-scale experiments while still extracting good performance from TabLeak.

E.3. Choice of the Pooling Function

We compare TabLeak using median pooling to TabLeak with mean pooling in Tab. 34 over the four datasets. As we can observe, in most cases both methods produce similar results, hence the effectiveness of TabLeak is mostly independent of this choice. However, as median pooling demonstrates to provide a slight edge in some cases, we opt for using median pooling in our main experiments with TabLeak.

batch size	TabLeak (median)	TabLeak (mean)
1	$\mathbf{99.4 \pm 2.8}$	$\mathbf{99.4 \pm 2.8}$
2	99.2 ± 5.5	$\mathbf{99.3 \pm 5.0}$
4	$\mathbf{98.0 \pm 4.5}$	97.7 ± 5.3
8	$\mathbf{95.1 \pm 9.2}$	94.8 ± 9.0
16	$\mathbf{89.4 \pm 7.6}$	88.9 ± 7.7
32	$\mathbf{77.6 \pm 4.8}$	77.1 ± 4.7
64	71.2 ± 2.8	$\mathbf{71.7 \pm 2.8}$
128	68.8 ± 1.3	$\mathbf{69.4 \pm 1.4}$

(a) Adult

batch size	TabLeak (median)	TabLeak (mean)
1	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$
2	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$
4	$\mathbf{99.9 \pm 0.4}$	$\mathbf{99.9 \pm 0.4}$
8	$\mathbf{99.7 \pm 1.1}$	99.6 ± 1.1
16	$\mathbf{95.9 \pm 3.4}$	95.6 ± 3.3
32	$\mathbf{83.6 \pm 2.9}$	83.1 ± 3.0
64	$\mathbf{73.0 \pm 1.3}$	72.6 ± 1.3
128	$\mathbf{71.3 \pm 0.8}$	70.8 ± 0.9

(b) German Credit

batch size	TabLeak (median)	TabLeak (mean)
1	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$
2	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$
4	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$
8	98.7 ± 3.8	$\mathbf{98.8 \pm 3.4}$
16	$\mathbf{94.8 \pm 5.6}$	94.6 ± 5.4
32	$\mathbf{84.8 \pm 3.9}$	84.7 ± 3.9
64	$\mathbf{78.2 \pm 2.0}$	$\mathbf{78.2 \pm 2.2}$
128	77.3 ± 1.2	$\mathbf{77.5 \pm 1.2}$

(c) Lawschool Admissions

batch size	TabLeak (median)	TabLeak (mean)
1	$\mathbf{99.8 \pm 1.6}$	$\mathbf{99.8 \pm 1.6}$
2	$\mathbf{97.7 \pm 8.3}$	97.4 ± 9.0
4	$\mathbf{98.2 \pm 6.5}$	98.0 ± 6.7
8	$\mathbf{96.0 \pm 8.2}$	95.6 ± 8.6
16	$\mathbf{86.1 \pm 8.8}$	84.9 ± 9.3
32	$\mathbf{70.0 \pm 4.5}$	69.7 ± 4.4
64	64.7 ± 2.8	$\mathbf{64.8 \pm 2.9}$
128	63.0 ± 1.4	$\mathbf{63.6 \pm 1.5}$

(d) Health Heritage

Table 34: Mean and standard deviation of the reconstruction accuracy [%] using TabLeak with either median or mean pooling, assuming full knowledge of the true labels.