GOLIATH: Gradient Inversion of Tabular Diffusion Models

Giulio Segalini[†]
Université de Neuchâtel
Neuchâtel, Switzerland
giulio.segalini@unine.ch

Aditya Shankar[†]
Delft University of Technology
Delft, the Netherlands
a.shankar@tudelft.nl

Jérémie Decouchant
Delft University of Technology
Delft, the Netherlands
J.Decouchant@tudelft.nl

Lydia Y. Chen
Université de Neuchâtel and
Delft University of Technology
lydiaychen@ieee.org

Abstract

Gradient inversion reconstructs training data from shared gradients. However, existing methods target classification tasks and image diffusion, and do not address mixed tabular diffusion, where numerical columns follow Gaussian noising while categorical columns are corrupted by discrete masks. This mixed continuous-discrete training objective changes both the inversion target and the gradient structure: an attacker must recover not only rows, timesteps, and noise, but also the latent mask pattern. We propose GOLIATH, the first gradient inversion attack designed for tabular diffusion models. GOLIATH employs a cyclic inversion loop that jointly recovers rows and diffusion latents across Gaussian and mixed diffusion regimes. We introduce a tabular-aware objective function that balances numerical and categorical gradient contributions while improving categorical recovery through enforced consistency between noise levels and mask structures. To accommodate large batches, we aggregate multiple reconstructions across epochs using a row-alignment heuristic. Evaluated across ten tabular datasets, GOLIATH achieves up to 89.1% per-cell reconstruction accuracy and consistently outperforms gradient-inversion baselines across Gaussian and mixed tabular diffusion. Code is available[1].

1 Introduction

Synthetic tabular-like data generation is widely used for augmentation, imputation, and simulations [35, 51, 44, 42]. Among generative paradigms, *diffusion* models [46, 22] are state-of-the-art [28, 44, 54], due to their stepwise *denoising* process that transforms noise into realistic samples. This stepwise generative approach lends generation stability and improved quality [11] over generative adversarial networks (GANs) [20] and variational autoencoders (VAEs) [27].

Recent works [32, 9, 47] train diffusion models through *federated learning* (FL) [34], where a server pools clients' gradients. For tabular data, this enables institutions to learn a shared generator without openly sharing their data, as needed for collaborative analysis of financial and medical data. FedTabDiff [41] studies this setting on mixed-type financial and medical datasets, while DP-FedTabDiff [40] additionally incorporates differential privacy. These applications motivate studying what the shared updates reveal. Although raw data is never shared, local gradients allow an adversarial

[†]Equal contribution
[1]https://github.com/gsegalini/goliath

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

server to reconstruct the original data through *gradient inversion* [14, 8, 58]. But existing attacks face three main challenges when applied to tabular diffusion.

First, most gradient inversion attacks are built for *classification* models, where gradients correspond to labeled samples [50, 58, 8, 14]. Differently, diffusion models are *generative* and gradients arise from denoising objectives over noisy inputs, timesteps, and stochastic perturbations; *there is no label*. As a result, existing inversion techniques do not directly apply.

The second challenge arises from the *heterogeneous* structure of tabular data. Existing inversion methods for diffusion are limited to *images* [24], assuming a homogeneous and fully continuous feature space. Tabular data combines *continuous* and *categorical* features, yielding two distinct families of tabular diffusion models: (i) *continuous-only* models that apply *Gaussian* noise to all features [43, 55, 54]; and (ii) *mixed* models that combine Gaussian noise for numerical features with *discrete* corruptions, e.g., *masking* for categorical ones [44]. These two settings induce different inversion targets: recovering noise and timestep in the first case, and additionally recovering discrete masks in the second. Beyond the targets, the gradient itself is heterogeneous: numerical and categorical channels deposit signal into disjoint output rows at different scales, so a single image-style gradient distance lets the larger channel dominate and leaves the smaller one under-recovered.

Third, tabular diffusion is trained on *batches* of rows, so that gradients correspond to an unordered set of records rather than a single instance. This contrasts with prior work on image diffusion inversion, which is limited to single-sample reconstruction [24]. Prior classification-based gradient inversion methods improve reconstruction accuracy by aggregating multiple solutions [48, 50, 31]. However, such aggregation requires aligning rows across reconstructions. Existing approaches align samples using class labels, either directly or through inferred labels, which does not work for tabular diffusion since generative training has no class labels. Alignment must therefore rely on data-based matching, typically implemented via the *Hungarian algorithm* [29]. However, its *cubic* complexity makes it impractical for large batches [48], making scalable row alignment challenging.

To address the aforementioned challenges, we introduce GOLIATH [2], a gradient inversion attack for tabular diffusion. GOLIATH runs a cyclic inversion loop that jointly recovers the data, the diffusion timestep, the noise, and, for mixed diffusion, the categorical mask. We stabilize optimization with an exponential moving average (EMA) normalization to balance contributions from continuous and discrete channels. This loop is paired with a cross-epoch row-alignment step that aggregates reconstructions across training epochs without class labels.

To summarize, we make the following three contributions:

Cyclic tabular inversion: GOLIATH extends classification-based inversion to generative diffusion through a cyclic inversion procedure that replaces label recovery with the recovery of the latent variables governing the diffusion process. We jointly infer the diffusion timestep, noise, and (in mixed models) categorical masks, to reconstruct the data from the resulting generative process.

Tabular-aware objective for heterogeneous features: We design a tabular-aware gradient matching objective that balances the gradient contributions of numerical and discrete features through an EMA scheme. In the mixed setting, we enforce consistency between the recovered timestep distribution and mask-density distribution, enabling joint recovery of continuous values and discrete masks.

Cross-epoch row alignment: We introduce a data-based row-alignment heuristic that enables cross-epoch ensembling in the tabular diffusion setting without class labels. This technique allows the inversion of large batches, countering gradient dilution effects.

We evaluate GOLIATH on ten benchmark datasets with per-cell reconstruction accuracy and median Distance-to-Closest-Record (DCR), comparing against TABLEAK [48]- and GIDM [24]-based baselines, and assess robustness to known defenses [49, 56, 19].

2 Background & Related Studies

Tabular diffusion models. Diffusion models [46, 22] generate data by learning to reverse a Markovian noising process. A *forward* diffusion process progressively corrupts the data with noise, and a neural network learns to predict and remove this noise via a *reverse* (denoising) process. New

[2] GOLIATH stands for Gradient Optimization Leaking Individual Attributes from Tables

samples are generated by starting from pure Gaussian noise and iteratively applying the learned denoising procedure. For a tabular sample (row) $x_0 \in \mathbb{R}^d$ composed of discrete and continuous features, the forward process produces noisy samples $\{x_t\}_{t=1}^T$ over T steps:

$$q(x_t \mid x_{t-1}) \sim \mathcal{N}(\sqrt{\alpha_t}x_{t-1}, (1-\alpha_t)I),$$

where α_t controls the noise variance at step t. Using a convenient reparameterization [22], we can sample $q(x_t|x_0)$ directly from the clean sample x_0:

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\,\epsilon, \quad \epsilon \sim \mathcal{N}(0, I_d), \tag{1}$$

with $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$. The reverse process is parameterized by a neural network $\epsilon_\theta(x_t, t)$ that predicts the noise ϵ at each step, iteratively reconstructing clean samples starting from $x_T \sim \mathcal{N}(0, I)$. For a batch $X = \{x_0^{(b)}\}_{b=1}^B$ of B table rows, the training loss for the model is given by:

$$\mathcal{L} = \frac{\sum_{b=1}^B ||\epsilon_\theta(x_t^{(b)}, t^{(b)}) - \epsilon^{(b)}||^2}{B}, \tag{2}$$

where $t^{(b)}, \epsilon^{(b)}$ represent a randomly sampled diffusion step and noise corresponding to the b^{th} sample.

Tabular diffusion models come in two families. *Gaussian* models [55, 43] apply Eq. 1 to every feature after a continuous encoding and train with the mean-squared-error (MSE) loss of Eq. 2. *Mixed* models [44] use a Gaussian forward process for continuous features and masked diffusion for categorical ones; earlier variants use multinomial diffusion [28] or encode categoricals into continuous latents before Gaussian diffusion [54]. Masked diffusion noises a discrete feature d by probabilistically replacing it with a special mask token [M], set by the diffusion schedule $\bar{\alpha}_t$:

$$q_m(x_t^d \mid x_0^d) \;=\; \bar{\alpha}_t\, \delta_{x_t^d,\, x_0^d} \;+\; (1-\bar{\alpha}_t)\, \delta_{x_t^d,\, [\mathrm{M}]}, \tag{3}$$

where $\delta_{a,b}$ is the Kronecker delta. The token keeps its original value with probability $\bar{\alpha}_t$ and becomes [M] with probability $1-\bar{\alpha}_t$. Continuous features retain the Gaussian forward process of Eq. 1. The mixed training loss adds a cross-entropy term on the masked categorical logits to the MSE term on the continuous noise-prediction target, with per-timestep weights w_c, w_d [44] as follows:

$$\mathcal{L}_{\text{mixed}} = w_c(t)\,\mathcal{L}_{\text{MSE}} + w_d(t)\,\mathcal{L}_{\text{CE}}, \tag{4}$$

Gradient inversion on classification. Gradient inversion exploits the fact that shared gradients encode information about the training data. For a loss ℓ and parameters θ,

$$\min_{x',y'} \; \mathcal{D}(\nabla_\theta \ell(x', y', \theta), \nabla_\theta \ell(x, y, \theta)),$$

where $\mathcal{D}$ is a gradient-matching distance and x', y' are dummy inputs and labels [58]. Most attacks target images [17, 53, 18] or text [10, 5, 38], where the feature space is homogeneous.

Spear's [13] rank-1 factorization $\nabla W = \delta \cdot h^\top$ recovers the input activation h to a linear layer; for the first layer this activation is the raw data x. In a diffusion setting, it would recover the data at x_t, not x_0, leaving the diffusion forward map to invert under unknown (t, ϵ). The Cocktail Party Attack [25] recasts batch gradient inversion as blind source separation via independent component analysis (ICA), which assumes a linear aggregation of statistically independent sources. Diffusion's noise-prediction loss couples each sample's contribution through (t_b, ϵ_b) in a nonlinear way that breaks this assumption. AGIC [50] and TGIA-RO [31] aggregate reconstructions across training epochs and match samples between epochs via inferred class labels. TABLEAK [48] iteratively optimizes dummy samples under a cosine-similarity gradient distance, adds softmax relaxation for categorical features, and pools multiple reconstruction candidates for the same observed gradient; row ordering across pooled candidates is resolved by the Hungarian algorithm [29], which runs in $O(N^3)$. All of these attacks invert a supervised loss and therefore recover y' as part of the solution, usually through exact methods [57], while generative diffusion provides no such label.

Gradient inversion on image diffusion. GIDM [24] targets image diffusion. With no class labels, the loss depends on the clean sample x_0, the injected noise ϵ, and the diffusion step t. GIDM runs a two-phase procedure: a *generative phase* that searches latent space with a pretrained diffusion sampler for an *dummy* reconstruction, followed by a *fine-tuning phase* that further optimizes the dummy. However, the process is Gaussian-only, so there is no mask latent; the data domain is image-only, so feature types are homogeneous; and the generative phase requires a pretrained generative model, which couples the attack to the availability of such a model or to the knowledge of the data distribution. None of these assumptions transfer to mixed tabular diffusion.

3 GOLIATH

GOLIATH attacks tabular diffusion by running a cyclic inversion loop that jointly recovers the training data and the diffusion latents, paired with a row-alignment step that aggregates reconstructions across epochs. We state the threat model (Sec. 3.1), give an overview (Sec. 3.2), detail the cyclic inversion for Gaussian (Sec. 3.3) and masked (Sec. 3.4) diffusion, and close with cross-epoch aggregation (Sec. 3.5) and the row-alignment heuristic (Sec. 3.5).

3.1 Preliminaries

We assume a standard federated learning setup where a *semi-honest* server orchestrates the training of a tabular diffusion model. To reconstruct individual client rows, the server uses the shared gradients and the global model parameters. We further assume that the server knows the model architecture and the data schema, in line with prior gradient inversion threat models [14, 8]: the server distributes the shared model, and clients use a common feature space. Diffusion parameters (noise levels, timesteps and mask) are assumed private. The server observes individual client updates; secure aggregation that hides these updates is outside our threat model. App. H.5 extends single-gradient inversion to updates after multiple local training steps.

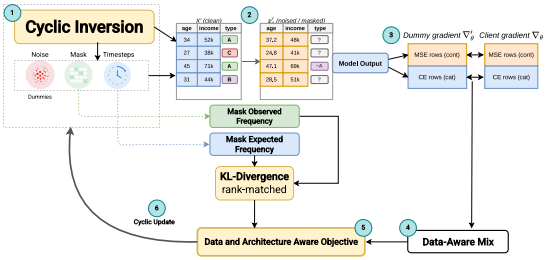


Figure 1: Overview of GOLIATH's mixed-diffusion inversion loop.

To achieve this reconstruction, the server performs gradient matching. Unlike classification, diffusion gradients ∇_θ depend on the training data x_0, the diffusion step t, and the noise ϵ. Given a client's update on a batch of size B, the server initializes dummy variables for the data (X'), timesteps (t'), and noise (ϵ'). For masked diffusion, dummy masking logits (m') are also added as free variables (detailed in Sec. 3.4). The server jointly optimizes these variables to minimize the distance between the observed and dummy gradients:

$$\min_{X',t',\epsilon'} \mathcal{D}(\nabla_\theta, \nabla'_\theta).$$

The dummy gradient $\nabla'_\theta \triangleq \nabla_\theta \mathcal{L}'$ is computed by replacing ground-truth inputs with the dummy variables $\{x_0'^{(b)}, t'^{(b)}, \epsilon'^{(b)}\}_{b=1}^{B}$:

$$\mathcal{L}' = \frac{\sum_{b=1}^{B} ||\epsilon_\theta(x_t'^{(b)}, t'^{(b)}) - \epsilon'^{(b)}||^2}{B} \tag{5}$$

Gaussian and mixed diffusion gradients differ in shape and scale, and require dedicated gradient-matching objectives, detailed respectively in Sec. 3.3 and Sec. 3.4.

3.2 Overview

At each training epoch, the server runs a **cyclic inversion** loop that reconstructs the client's private rows by gradient matching, recovering data, noise and timesteps for all tabular diffusion architectures and, for mixed-diffusion models, discrete categorical mask states.

To successfully target both fully continuous and mixed-data tabular architectures, this loop is driven by **regime-specific** gradient-matching objectives. For Gaussian diffusion, we use a combined distance metric (Sec. 3.3), while for mixed diffusion, we use a specialized loss that explicitly splits the gradient into its distinct MSE and cross-entropy channels (Sec. 3.4). To handle the specific mechanics of mixed diffusion, the inversion loop must additionally recover the mask latent m'. Because discrete masks disrupt continuous optimization, we optimize this latent through a continuous relaxation. We further enhance the reconstructions through **timestep-mask coupling**, which guides the optimization to match the observed dummy mask distribution with the expected one inferred from timesteps.

Finally, to effectively aggregate these reconstructions across multiple epochs, GOLIATH applies a **data-based row-alignment heuristic** (Sec. 3.5). Because tabular rows are permutation-invariant, this step aligns the current epoch's output to the previous epoch's state, enabling cross-epoch ensembling (Sec. 3.5) without relying on class labels.

3.3 Cyclic Inversion

In classification tasks, gradient inversion methods often first reconstruct the discrete labels before optimizing the input features, since accurate label recovery improves reconstructions [57, 18]. But this strategy does not directly extend to diffusion models, where the variables t and ϵ jointly influence the gradient. Hence, discrete label-recovery techniques are not applicable. Instead, we start from dummies X', t' and ϵ', and iteratively *cycle* through the following updates using a step size η:

$$\begin{aligned} X' &\leftarrow X' - \eta \nabla_{X'} \mathcal{D}(\nabla'_\theta, \nabla_\theta)\,, \\ t' &\leftarrow t' - \eta \nabla_{t'} \mathcal{D}(\nabla_\theta, \nabla'_\theta)\,, \\ \epsilon' &\leftarrow \epsilon' - \eta \nabla_{\epsilon'} \mathcal{D}(\nabla_\theta, \nabla'_\theta)\,, \end{aligned}$$

Under Gaussian diffusion the training gradient comes from a single MSE noise-prediction term, and both the magnitude and the direction of the gradient carry useful reconstruction signal. A pure cosine distance discards the magnitude; a pure MSE distance is dominated by whichever layers have the largest scale and drifts with it over the run. We combine both, with each term rescaled by its own running magnitude so that neither dominates the other.

At iteration k, let $m_k = \mathrm{MSE}(\nabla_\theta, \nabla'_{\theta,k})$ and $c_k = 1 - \mathrm{CosSim}(\nabla_\theta, \nabla'_{\theta,k})$. We maintain exponential moving averages of the two raw values with smoothing parameter β,

$$\begin{aligned} \hat{s}_k^{\mathrm{MSE}} &= \beta\, \hat{s}_{k-1}^{\mathrm{MSE}} + (1-\beta)\, \mathrm{sg}(m_k), \\ \hat{s}_k^{\cos} &= \beta\, \hat{s}_{k-1}^{\cos} + (1-\beta)\, \mathrm{sg}(c_k), \end{aligned}$$

where $\mathrm{sg}(\cdot)$ denotes a stop-gradient and the EMAs are initialized to the first values observed. The gradient-matching objective is

$$\mathcal{D}(\nabla_\theta, \nabla'_\theta) \;:=\; r\, \frac{m_k}{\hat{s}_k^{\mathrm{MSE}}} \;+\; (1-r)\, \frac{c_k}{\hat{s}_k^{\cos}}, \tag{6}$$

where r is a parameter that indicates the importance of each individual term. We set $r = n_{\mathrm{num}}/(n_{\mathrm{num}} + n_{\mathrm{cat_dims}})$, the ratio of continuous cells to all cells (continuous plus one-hot categorical, and mask coordinates under mixed diffusion) in the schema. The EMA normalizations keep both terms at $O(1)$ across the whole inversion run, so **neither signal is suppressed** by the other as magnitudes drift. The weight r routes more of the objective toward the MSE term when the data is numeric and toward the cosine term when the data is dominated by categorical or mask channels. App. G.1 reports the loss ablation study, comparing this joint loss function with its pure MSE and pure cosine endpoints and showing its importance.

Dummy initialization, optimizer settings, and the contrast with GIDM [24] are deferred to App. C.

3.4 Tabular-aware Inversion of Masked Diffusion

Unlike Gaussian diffusion, Masked diffusion [44] adds binary masks of categorical features as an inversion target, together with the data and the timestep. We extend the cyclic inversion with **dummy masking logits** $m' := \{m'^{(b,d)}\}$, one per sample b and discrete feature d, as a fourth free variable optimized alongside X', t', and ϵ'; continuous features keep ϵ' as their free variable. Fig. 1 summarizes the optimization flow.

Split gradient-matching objective. Mixed diffusion training produces gradients from two heterogeneous channels (Eq. 4): an MSE term on continuous features and a cross-entropy term on masked categoricals. These two channels deposit signal into disjoint rows of the model's output head, and they operate at different scales, so a single cosine computed over the concatenated parameter gradient is dominated by whichever channel has larger norm at a given iteration. To keep both channels balanced, we **split the gradient** at the output head into its MSE part $\nabla_\theta^{\mathrm{mse}}$ (rows feeding the continuous outputs) and its CE part $\nabla_\theta^{\mathrm{ce}}$ (rows feeding the categorical outputs), and match them with a double cosine:

$$\mathcal{D}_{\mathrm{mix}}(\nabla_\theta, \nabla'_\theta) \;:=\; \alpha\big(1 - \mathrm{CosSim}(\nabla_\theta^{\mathrm{mse}}, \nabla_\theta'^{\mathrm{mse}})\big) + (1-\alpha)\big(1 - \mathrm{CosSim}(\nabla_\theta^{\mathrm{ce}}, \nabla_\theta'^{\mathrm{ce}})\big), \tag{7}$$

with $\alpha = n_{\text{num}}/(n_{\text{num}} + n_{\text{cat_cols}})$, the fraction of columns that are continuous. The split replaces one angular match between two mixed vectors with two separate angular matches, one per signal type, so the categorical-heavy and numerical-heavy regimes **both contribute informative updates** instead of whichever channel happens to carry more weight. App. G.1 reports the ablation over different gradient loss objectives, including Eq. 6.

Continuous relaxation. Since binary mask variables are not differentiable, we apply a sigmoid relaxation at temperature τ:

$$\tilde{x}_t'^{(b,d)} = \left(1 - \sigma_\tau(m'^{(b,d)})\right) x_0'^{(b,d)} + \sigma_\tau(m'^{(b,d)})\,[\mathrm{M}],$$

where $\sigma_\tau(z) = (1 + e^{-z/\tau})^{-1}$ is the sigmoid at temperature τ. High temperature yields soft, near-uniform masks ($\sigma \approx 0.5$), while low temperature forces near-binary decisions. The cyclic update extends to four variable blocks, adding m' alongside X', t', and ϵ':

$$\begin{aligned} X' &\leftarrow X' - \eta \nabla_{X'} \mathcal{D}(\nabla_\theta, \nabla'_\theta), \\ t' &\leftarrow t' - \eta \nabla_{t'} \mathcal{D}(\nabla_\theta, \nabla'_\theta), \\ \epsilon' &\leftarrow \epsilon' - \eta \nabla_{\epsilon'} \mathcal{D}(\nabla_\theta, \nabla'_\theta), \\ m' &\leftarrow m' - \eta \nabla_{m'} \mathcal{D}(\nabla_\theta, \nabla'_\theta). \end{aligned}$$

The dummy gradient ∇'_θ is computed from a loss that mirrors the training objective: MSE on continuous features (as in Eq. 5) and cross-entropy on discrete features, each weighted by the corresponding soft mask probability $\sigma_\tau(m'^{(b,d)})$.

Temperature annealing. To encourage convergence to hard binary masks, we apply exponential decay after each optimization step: $\tau_{k+1} = \kappa\,\tau_k$, with initial temperature τ_0 and decay factor $\kappa < 1$. As $\tau \to 0$, the sigmoid approaches a hard step function, yielding a near-binary mask at convergence. The masking logits $m'^{(b,d)}$ are initialized to 0, corresponding to $\sigma(0) = 0.5$, i.e., a uniform prior over masked and unmasked states before optimization.

Timestep-mask coupling. Inverting mixed continuous-discrete diffusion introduces a **structural constraint** absent from Gaussian-only settings: under the masked forward process (Eq. 3), the masking probability for sample b is $\gamma_b = 1 - \bar{\alpha}_{t'_b}$, so each mask decision is a Bernoulli draw parameterized entirely by the timestep. Optimizing t' and m' independently leaves a *nullspace* in which timestep and mask density trade off while keeping ∇'_θ unchanged.

Let $p_b = D_{\text{cat}}^{-1} \sum_d \sigma_\tau(m'^{(b,d)})$ be the relaxed mask density for row b. We rank-sort $\{p_b\}$ and $\{\gamma_b\}$ before computing the KL, making the term a batch-level distribution-matching prior rather than an exact row-wise coupling.

For the divergence we use the symmetric Bernoulli KL,

$$\mathrm{KL}_{\text{sym}}(p \,\|\, q) := (p - q)\left(\log \tfrac{p}{1-p} - \log \tfrac{q}{1-q}\right),$$

the Bregman divergence on the Bernoulli manifold in logit space. It is nonnegative and zero when $p = q$. The coupling regularizer is

$$\mathcal{R}(m', t') \;=\; \frac{1}{B} \sum_{b=1}^{B} \mathrm{KL}_{\text{sym}}\left(\tilde{p}_b \,\middle\|\, \tilde{\gamma}_b\right), \tag{8}$$

where $\tilde{p}_1 \le \cdots \le \tilde{p}_B$ and $\tilde{\gamma}_1 \le \cdots \le \tilde{\gamma}_B$ are the rank-sorted sequences. The total attack objective under masked diffusion is

$$\min_{X',\, t',\, \epsilon',\, m'} \; \mathcal{D}(\nabla_\theta, \nabla'_\theta) + \lambda_k \cdot \mathcal{R}(m', t'),$$

where λ_k follows a warmup-flat $\to$ linear ramp $\to$ plateau $\to$ cooldown schedule over the optimization iterations. App. G.2 reports the ablation of the KL timestep-mask prior on all nine datasets, isolating its contribution to reconstruction accuracy and timestep recovery.

3.5 Cross-Epoch Ensembling

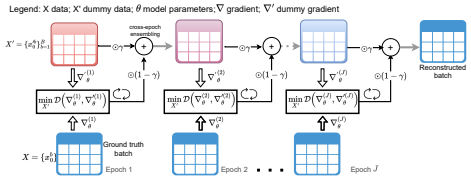


Figure 2: Cross-epoch ensembling. Reconstructions from successive training epochs are row-aligned and averaged with decaying weight.

The cyclic optimization in Sec. 3.3 reconstructs the dummy data X', diffusion steps t', and noise ϵ' from a single observed gradient. In classification inversion, AGIC [50] shows that combining reconstructions obtained from gradients of several consecutive training epochs tightens the feasible set because each new gradient is an additional constraint on the same underlying batch. We carry this aggregation idea into the diffusion setting. We assume batches repeat across epochs: if a client trains on M batches in a fixed order, then the gradient observed at step $M{+}1$ corresponds to the same private rows as step 1, but under model parameters already updated by the previous M batches. The ingredient that makes the aggregation usable on tabular data is a **data-based cross-epoch row-alignment step** (Sec. 3.5). App. G.7 adapts the AGIC feasible-set argument as an identifiability intuition, rather than a recovery guarantee for our optimizer. The single-gradient attack does not require repeated batches; App. H.6 evaluates the optional ensembling step under reshuffling.

Let $\tilde{X}_{J+1}$ be the reconstruction obtained by inverting the gradient from epoch $J{+}1$ alone. We maintain a running estimate X' and update it after each epoch via a moving average,

$$X' \leftarrow \rho X' + (1-\rho)\tilde{X}_{J+1}, \tag{9}$$

with $\rho \in (0, 1)$, balancing updates from earlier constraints while adapting to the latest gradient.

Eq. 9 averages reconstructions in the order in which their rows happen to come out of the optimizer. We summarize the multiple epoch loop in Fig. 2. Tabular gradients are row-permutation-invariant, so two consecutive reconstructions of the same batch may emerge under different row permutations, and naively averaging them degrades the estimate. The next section describes how we resolve this.

Cross-epoch row alignment. The minibatch training loss averages over rows (Eq. 5), so gradient inversion recovers a batch whose row order is unconstrained by the observed gradient. To combine two reconstructions of the same batch, they must first be aligned. This can be solved with the Hungarian algorithm (bipartite matching) [29]. We propose a greedy alternative that avoids solving the full assignment problem and favors already good reconstructions to leak more information.

Given a batch of N rows and k columns, we build an $N \times N$ similarity matrix S where S_{r_1,r_2} is the similarity between row r_1 of the first reconstruction and row r_2 of the second. We then repeatedly select the highest-scoring still-valid entry in S, record the match, and remove the selected row and column. Pseudocode is in Alg. 1 (App. E).

4 Evaluation

We evaluate GOLIATH across diffusion families on ten benchmark datasets, covering main reconstruction accuracy (Tabs. 1 and 2), robustness to defenses, and cross-epoch ensembling. DCR results, component ablations, diffusion-latent sensitivity studies, and additional pooled-ensembling experiments are in Apps. F and G; line plots show 95% confidence intervals.

4.1 Experimental Setup

Implementation. We implement GOLIATH in Python 3.12 with PyTorch 2.8.0 [37] and run experiments on an NVIDIA H100 GPU. Training and inversion use Adam [26]; unless stated otherwise, cyclic inversion runs for $2,000$ iterations with learning rate 0.06, and diffusion backbones use 500 diffusion steps. On average, inverting one gradient takes 8.5 seconds and uses approximately 700 MB of GPU memory. The gradient distance is chosen from the schema (Dynamic, App. G.1); further details are in App. C.

Metrics. We report cell accuracy [48] and median Distance-to-Closest-Record (DCR) [30]. For the main reconstruction tables, we compute both metrics once per inverted batch and report the mean and standard deviation (as a subscript) over these batch-level values. Metric details are in App. D.

Table 1: Reconstruction accuracy (↑, mean and standard deviation) across datasets and batch sizes under Gaussian diffusion. Subscripts are standard deviations across attacked batches.

Batch Size	Method	Dataset									
		Abalone	**Adult**	**Bean**	**Default**	**Gesture**	**Letter**	**Magic**	**Shoppers**	**Credit**	**Census**
1	TabLeak	$9.8\%_{8.6}$	$20.0\%_{15.6}$	$5.6\%_{7.0}$	$8.1\%_{6.5}$	$3.7\%_{4.4}$	$3.5\%_{4.4}$	$8.5\%_{8.5}$	$11.2\%_{7.1}$	$26.1\%_{17.3}$	$21.2\%_{12.1}$
	Adaptive TabLeak	$27.2\%_{30.6}$	$71.1\%_{21.6}$	$42.0\%_{27.3}$	$61.5\%_{28.6}$	$44.8\%_{29.6}$	$54.5\%_{28.5}$	$38.5\%_{32.1}$	$50.8\%_{31.6}$	$66.0\%_{22.3}$	$65.7\%_{26.1}$
	GIDM	$30.9\%_{25.9}$	$12.9\%_{6.7}$	$10.1\%_{9.3}$	$25.2\%_{11.8}$	$24.7\%_{11.2}$	$21.5\%_{11.9}$	$33.8\%_{25.3}$	$29.1\%_{12.7}$	$28.3\%_{9.2}$	$22.3\%_{4.5}$
	GOLIATH 1-epoch	$44.8\%_{30.1}$	$75.5\%_{20.7}$	$51.6\%_{29.0}$	$52.6\%_{22.1}$	$54.3\%_{27.7}$	$48.2\%_{25.0}$	$48.7\%_{27.1}$	$\mathbf{61.4\%}_{25.3}$	$77.0\%_{19.2}$	$73.5\%_{22.5}$
	GOLIATH 5-epochs	$\mathbf{56.0\%}_{26.8}$	$\mathbf{88.9\%}_{13.2}$	$\mathbf{54.0\%}_{30.4}$	$\mathbf{65.2\%}_{25.2}$	$\mathbf{66.9\%}_{23.9}$	$\mathbf{56.3\%}_{28.3}$	$\mathbf{55.1\%}_{27.1}$	$55.2\%_{26.1}$	$\mathbf{88.4\%}_{15.3}$	$\mathbf{89.1\%}_{10.1}$
8	TabLeak	$20.3\%_{3.7}$	$21.8\%_{3.4}$	$12.3\%_{2.5}$	$13.6\%_{2.3}$	$9.4\%_{2.8}$	$10.7\%_{3.2}$	$19.9\%_{3.8}$	$18.7\%_{2.6}$	$31.9\%_{4.3}$	$28.8\%_{6.9}$
	Adaptive TabLeak	$38.5\%_{5.5}$	$45.4\%_{7.6}$	$37.0\%_{5.3}$	$42.6\%_{6.1}$	$37.1\%_{6.5}$	$36.5\%_{5.0}$	$38.1\%_{5.4}$	$46.3\%_{5.8}$	$49.6\%_{5.0}$	$43.5\%_{8.5}$
	GIDM	$24.0\%_{3.6}$	$22.3\%_{2.6}$	$9.7\%_{3.3}$	$17.1\%_{4.2}$	$15.1\%_{2.4}$	$21.4\%_{2.4}$	$27.3\%_{4.5}$	$23.7\%_{4.1}$	$35.1\%_{2.1}$	$24.2\%_{1.4}$
	GOLIATH 1-epoch	$39.5\%_{5.1}$	$63.8\%_{7.4}$	$45.8\%_{10.2}$	$53.1\%_{9.1}$	$50.4\%_{8.9}$	$45.2\%_{9.6}$	$38.8\%_{4.0}$	$55.6\%_{7.2}$	$60.8\%_{7.5}$	$66.1\%_{9.2}$
	GOLIATH 5-epochs	$\mathbf{50.0\%}_{8.1}$	$\mathbf{63.9\%}_{4.8}$	$\mathbf{53.2\%}_{8.2}$	$\mathbf{61.3\%}_{7.1}$	$\mathbf{61.4\%}_{8.4}$	$\mathbf{47.8\%}_{7.0}$	$\mathbf{46.8\%}_{5.4}$	$\mathbf{61.0\%}_{5.2}$	$\mathbf{60.8\%}_{5.1}$	$\mathbf{72.9\%}_{4.3}$
16	TabLeak	$24.7\%_{3.2}$	$25.8\%_{3.4}$	$17.5\%_{2.6}$	$16.1\%_{1.8}$	$12.1\%_{1.7}$	$12.9\%_{2.1}$	$23.5\%_{2.3}$	$20.8\%_{2.2}$	$33.7\%_{2.7}$	$34.1\%_{9.1}$
	Adaptive TabLeak	$40.1\%_{3.4}$	$43.6\%_{5.7}$	$35.4\%_{3.3}$	$38.2\%_{2.8}$	$32.8\%_{2.9}$	$34.6\%_{2.7}$	$37.6\%_{3.3}$	$41.2\%_{3.8}$	$48.5\%_{3.1}$	$41.4\%_{6.7}$
	GIDM	$28.7\%_{4.0}$	$22.1\%_{1.4}$	$8.7\%_{3.2}$	$16.6\%_{2.3}$	$13.4\%_{1.6}$	$20.7\%_{2.9}$	$29.1\%_{3.1}$	$24.8\%_{2.7}$	$36.1\%_{1.7}$	$25.1\%_{1.2}$
	GOLIATH 1-epoch	$44.3\%_{3.9}$	$61.6\%_{4.2}$	$46.5\%_{7.0}$	$46.9\%_{5.2}$	$47.5\%_{6.1}$	$45.2\%_{7.1}$	$43.6\%_{3.5}$	$50.4\%_{5.0}$	$\mathbf{60.2\%}_{3.9}$	$63.9\%_{6.3}$
	GOLIATH 5-epochs	$\mathbf{51.6\%}_{6.0}$	$\mathbf{62.3\%}_{3.5}$	$\mathbf{53.8\%}_{4.3}$	$\mathbf{56.7\%}_{3.8}$	$\mathbf{58.3\%}_{5.9}$	$\mathbf{46.6\%}_{5.2}$	$\mathbf{50.4\%}_{4.2}$	$\mathbf{58.6\%}_{3.6}$	$59.7\%_{2.9}$	$\mathbf{69.3\%}_{3.4}$
32	TabLeak	$27.7\%_{2.3}$	$28.5\%_{3.5}$	$19.0\%_{1.6}$	$18.7\%_{1.3}$	$14.3\%_{1.4}$	$15.5\%_{1.7}$	$26.6\%_{2.4}$	$22.9\%_{1.8}$	$35.5\%_{1.4}$	$39.0\%_{11.0}$
	Adaptive TabLeak	$42.0\%_{2.2}$	$43.2\%_{3.9}$	$35.9\%_{1.4}$	$36.7\%_{2.0}$	$31.0\%_{2.1}$	$34.4\%_{1.8}$	$41.2\%_{1.4}$	$40.4\%_{2.2}$	$48.4\%_{3.0}$	$39.9\%_{7.9}$
	GIDM	$32.2\%_{4.5}$	$22.7\%_{1.6}$	$9.0\%_{2.8}$	$17.1\%_{2.2}$	$13.3\%_{2.2}$	$20.0\%_{5.8}$	$32.6\%_{4.4}$	$26.4\%_{4.4}$	$36.8\%_{1.2}$	$25.9\%_{0.8}$
	GOLIATH 1-epoch	$48.7\%_{3.2}$	$58.1\%_{3.4}$	$44.6\%_{3.4}$	$45.9\%_{2.8}$	$43.9\%_{4.3}$	$40.3\%_{3.5}$	$48.5\%_{2.7}$	$48.7\%_{2.4}$	$56.0\%_{2.2}$	$60.9\%_{3.9}$
	GOLIATH 5-epochs	$\mathbf{51.3\%}_{4.4}$	$\mathbf{62.3\%}_{2.3}$	$\mathbf{52.8\%}_{3.0}$	$\mathbf{53.6\%}_{2.2}$	$\mathbf{49.7\%}_{3.7}$	$\mathbf{45.0\%}_{2.2}$	$\mathbf{53.0\%}_{3.0}$	$\mathbf{55.3\%}_{2.4}$	$\mathbf{60.3\%}_{1.9}$	$\mathbf{67.7\%}_{2.0}$

Table 2: Reconstruction accuracy (↑, mean and standard deviation) across datasets and batch sizes under mixed diffusion. Subscripts are standard deviations across attacked batches.

Batch Size	Method	Dataset									
		Abalone	**Adult**	**Bean**	**Default**	**Gesture**	**Letter**	**Magic**	**Shoppers**	**Credit**	**Census**
1	Random Mask	$22.7\%_{23.3}$	$40.5\%_{16.3}$	$20.7\%_{21.7}$	$43.2\%_{23.1}$	$30.8\%_{24.6}$	$30.2\%_{27.2}$	$31.3\%_{29.5}$	$44.7\%_{16.6}$	$45.5\%_{15.4}$	$36.1\%_{11.7}$
	TabLeak	$8.0\%_{9.8}$	$21.2\%_{14.0}$	$6.9\%_{10.7}$	$8.5\%_{6.1}$	$4.1\%_{7.8}$	$4.5\%_{6.4}$	$10.2\%_{8.3}$	$16.2\%_{14.2}$	$25.4\%_{12.0}$	$30.5\%_{11.3}$
	Adaptive TabLeak	$32.6\%_{28.1}$	$50.8\%_{19.7}$	$39.9\%_{23.3}$	$49.0\%_{22.3}$	$47.6\%_{25.3}$	$36.8\%_{21.7}$	$37.7\%_{24.3}$	$49.4\%_{19.7}$	$52.3\%_{16.5}$	$53.9\%_{23.3}$
	GOLIATH 1-epoch	$\mathbf{52.6\%}_{33.2}$	$59.5\%_{29.4}$	$52.0\%_{28.5}$	$59.2\%_{24.7}$	$56.0\%_{27.8}$	$51.4\%_{29.3}$	$44.6\%_{26.8}$	$59.1\%_{21.8}$	$65.1\%_{23.3}$	$45.3\%_{20.5}$
	GOLIATH 5-epochs	$49.7\%_{28.8}$	$\mathbf{68.7\%}_{14.0}$	$\mathbf{66.7\%}_{21.9}$	$\mathbf{74.0\%}_{16.4}$	$\mathbf{72.4\%}_{18.8}$	$\mathbf{65.5\%}_{20.2}$	$\mathbf{59.2\%}_{26.5}$	$\mathbf{77.0\%}_{14.3}$	$\mathbf{76.8\%}_{12.2}$	$\mathbf{58.0\%}_{12.4}$
8	Random Mask	$35.2\%_{7.7}$	$38.0\%_{4.2}$	$30.0\%_{5.9}$	$32.9\%_{5.5}$	$27.3\%_{6.8}$	$23.9\%_{5.8}$	$32.9\%_{7.3}$	$37.0\%_{4.1}$	$44.0\%_{3.4}$	$38.0\%_{4.9}$
	TabLeak	$21.6\%_{4.1}$	$26.6\%_{3.9}$	$13.7\%_{2.9}$	$14.4\%_{3.6}$	$9.2\%_{2.6}$	$11.0\%_{3.4}$	$18.7\%_{4.5}$	$19.1\%_{3.7}$	$33.5\%_{3.4}$	$33.5\%_{5.3}$
	Adaptive TabLeak	$41.7\%_{6.1}$	$44.5\%_{4.4}$	$39.9\%_{7.7}$	$41.4\%_{6.8}$	$45.6\%_{9.1}$	$32.0\%_{3.9}$	$40.6\%_{5.8}$	$41.3\%_{4.8}$	$48.4\%_{4.8}$	$46.0\%_{5.9}$
	GOLIATH 1-epoch	$43.3\%_{5.8}$	$44.8\%_{6.0}$	$44.4\%_{8.5}$	$48.4\%_{6.2}$	$48.7\%_{8.1}$	$36.4\%_{6.4}$	$44.3\%_{5.5}$	$48.2\%_{6.3}$	$49.4\%_{5.2}$	$42.6\%_{5.6}$
	GOLIATH 5-epochs	$\mathbf{47.1\%}_{7.0}$	$\mathbf{49.5\%}_{4.4}$	$\mathbf{53.6\%}_{7.2}$	$\mathbf{58.0\%}_{5.5}$	$\mathbf{62.0\%}_{7.4}$	$\mathbf{44.1\%}_{5.4}$	$\mathbf{48.5\%}_{5.4}$	$\mathbf{55.9\%}_{4.8}$	$\mathbf{53.1\%}_{4.3}$	$\mathbf{48.3\%}_{5.3}$
16	Random Mask	$40.1\%_{4.5}$	$40.4\%_{3.9}$	$33.0\%_{4.0}$	$34.4\%_{3.6}$	$29.8\%_{4.7}$	$26.4\%_{3.9}$	$38.4\%_{3.5}$	$39.5\%_{2.5}$	$44.7\%_{2.9}$	$40.2\%_{4.5}$
	TabLeak	$23.7\%_{3.6}$	$28.0\%_{2.6}$	$16.2\%_{2.9}$	$17.1\%_{1.9}$	$12.1\%_{1.9}$	$13.9\%_{2.7}$	$23.1\%_{3.9}$	$21.8\%_{2.8}$	$34.8\%_{2.8}$	$31.4\%_{4.3}$
	Adaptive TabLeak	$42.7\%_{3.8}$	$44.6\%_{2.6}$	$36.8\%_{3.9}$	$39.0\%_{3.8}$	$41.4\%_{5.6}$	$31.8\%_{2.8}$	$41.0\%_{4.5}$	$39.6\%_{2.7}$	$49.1\%_{3.0}$	$47.2\%_{3.0}$
	GOLIATH 1-epoch	$46.5\%_{4.2}$	$46.0\%_{3.3}$	$41.2\%_{3.6}$	$43.7\%_{4.2}$	$49.8\%_{6.5}$	$35.3\%_{3.5}$	$46.6\%_{3.5}$	$44.0\%_{3.9}$	$50.0\%_{3.2}$	$40.3\%_{7.6}$
	GOLIATH 5-epochs	$\mathbf{49.3\%}_{5.3}$	$\mathbf{50.6\%}_{2.8}$	$\mathbf{51.3\%}_{4.4}$	$\mathbf{54.2\%}_{3.7}$	$\mathbf{59.3\%}_{5.8}$	$\mathbf{41.5\%}_{3.4}$	$\mathbf{49.2\%}_{4.1}$	$\mathbf{54.3\%}_{3.5}$	$\mathbf{53.4\%}_{2.8}$	$\mathbf{52.0\%}_{4.2}$
32	Random Mask	$47.0\%_{2.6}$	$43.1\%_{2.4}$	$35.1\%_{2.9}$	$38.4\%_{2.2}$	$33.3\%_{4.4}$	$28.4\%_{2.0}$	$43.2\%_{3.1}$	$43.3\%_{2.0}$	$47.5\%_{1.4}$	$43.5\%_{3.8}$
	TabLeak	$25.7\%_{2.4}$	$29.1\%_{2.2}$	$18.4\%_{2.3}$	$18.5\%_{1.8}$	$14.4\%_{2.0}$	$15.8\%_{2.2}$	$25.3\%_{2.7}$	$22.5\%_{2.3}$	$36.2\%_{1.7}$	$32.4\%_{3.8}$
	Adaptive TabLeak	$45.6\%_{1.7}$	$45.4\%_{2.5}$	$37.0\%_{2.2}$	$37.6\%_{2.0}$	$39.2\%_{2.4}$	$32.6\%_{1.3}$	$43.1\%_{2.5}$	$39.9\%_{2.3}$	$49.4\%_{1.5}$	$46.7\%_{2.6}$
	GOLIATH 1-epoch	$49.5\%_{3.3}$	$45.5\%_{2.7}$	$39.7\%_{2.8}$	$38.6\%_{2.5}$	$43.1\%_{3.4}$	$33.5\%_{1.7}$	$49.9\%_{2.1}$	$40.2\%_{1.8}$	$50.1\%_{1.8}$	$43.1\%_{5.4}$
	GOLIATH 5-epochs	$\mathbf{52.4\%}_{3.6}$	$\mathbf{50.4\%}_{2.3}$	$\mathbf{52.0\%}_{2.7}$	$\mathbf{51.5\%}_{2.4}$	$\mathbf{55.4\%}_{3.0}$	$\mathbf{41.6\%}_{1.5}$	$\mathbf{51.4\%}_{2.2}$	$\mathbf{52.3\%}_{1.9}$	$\mathbf{55.5\%}_{1.3}$	$\mathbf{54.8\%}_{3.5}$

Baselines. We compare GOLIATH against the following baselines across the two diffusion families. *GIDM (Gaussian)* [24] is a two-phase image-diffusion attack that optimizes t and ϵ with an MSE gradient distance. We run both phases, using a model pretrained on the attacked dataset for the generative phase and optimizing the dummy data directly in the fine-tuning phase. *TabLeak (Gaussian)* [48] is the closest tabular gradient-inversion baseline; we use its $1-\mathrm{CosSim}$ distance, remove label recovery because diffusion training has no labels, and hold t and ϵ fixed at random initialization. *TabLeak (masked)* applies the same method to mixed diffusion without optimizing t, ϵ, or the mask m. *Adaptive TabLeak* extends both variants to optimize these diffusion latents with the same cyclic schedule, while retaining TabLeak's cosine distance and omitting our timestep-mask prior and cross-epoch ensembling. *Random-mask (masked)* keeps our cyclic inversion and mixed-gradient objective but draws m uniformly at random, isolating the contribution of mask recovery.

Datasets. We use ten public benchmarks, including the category-heavy Census (KDD) dataset; Tab. 4 in App. B summarizes the nine benchmark schemas and sizes, while App. H.3 describes Census.

4.2 Main Reconstruction Results

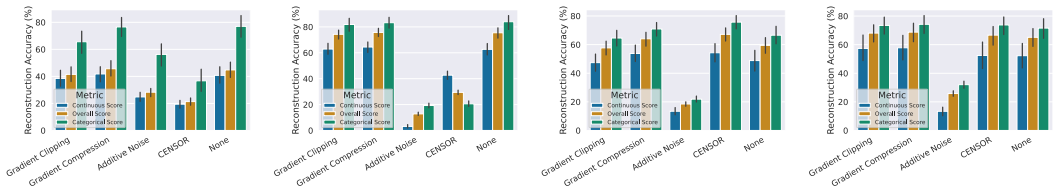
Tabs. 1 and 2 report cell accuracy under Gaussian and mixed diffusion; DCR results are in App. F. The 1-epoch and 5-epoch variants separate per-epoch inversion quality from cross-epoch aggregation.

Under Gaussian diffusion, GOLIATH recovers **substantially more cells** than TabLeak and GIDM across the ten tabular datasets. The gap reflects optimization of the diffusion latents t and ϵ, plus the EMA-scaled combined distance of Eq. 6, which preserves signal that GIDM's pure MSE term discards on mixed tabular schemas.

Under mixed diffusion, GOLIATH **remains well ahead** of TabLeak, which cannot recover the mask m', and of Random-mask, which cannot jointly align m' with t', although the five-epoch margin

Table 3: Per-column reconstruction accuracy on Adult (Gaussian diffusion, batch size 32, one epoch, 256-row subsets). Mean and standard deviation in percentage points over three seeds.

Column	Race	Sex	Marital	Country	Relation	Age	Overall
Accuracy	$83.2_{2.2}$	$82.4_{2.1}$	$69.7_{1.8}$	$67.4_{1.8}$	$67.3_{1.0}$	$38.5_{2.4}$	$59.8_{0.8}$



(a) Abaloné (Gaussian). (b) Adult (Gaussian). (c) Adult (mixed). (d) Credit (mixed).

Figure 3: Reconstruction accuracy with defenses at batch size 1.

over TabLeak shrinks (35.6 versus 40.0 points on average). **Timestep-mask coupling** contributes to this margin (App. G.2); the best distance depends on the schema (App. G.1). Adaptive TabLeak substantially improves over fixed-latent TabLeak, but GOLIATH with five epochs remains ahead in all 80 configurations, by 14.6 percentage points on average. With one epoch, GOLIATH wins 74 configurations; Adaptive TabLeak leads on Default and Letter at batch size 1 under Gaussian diffusion and on Census at all four batch sizes under mixed diffusion. Thus, the advantage is not explained solely by optimizing diffusion latents, although mixed-diffusion Census requires ensembling (App. H.1).

4.3 Leakage Granularity and Backbone Coverage

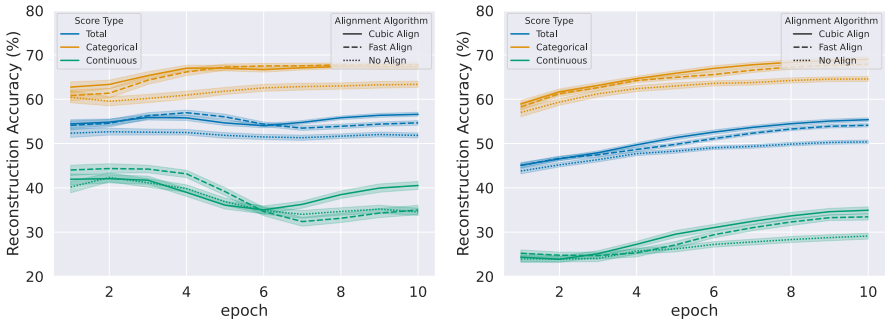
Leakage granularity. We measure whole-row recovery and per-column accuracy to determine what the cell averages conceal. Across Adult, Abalone, Gesture, Letter, and Magic, only 0.2% of rows have every cell correct at batch size 32 under either diffusion family, using exact categorical matches and the numerical tolerance of App. D. However, sensitive attributes remain exposed: on Adult under Gaussian diffusion, race and sex are recovered at 83.2% and 82.4%, compared with 59.8% across all columns (Tab. 3). App. H.2 reports whole-row thresholds, within-batch nearest-neighbour matching, and cardinality breakdowns. High-cardinality categorical accuracy falls from 71.2% at batch size 1 to 46.1% at 32, while low-cardinality columns remain easier to recover. Census-Income (KDD), included in the main accuracy tables, has 34 categorical columns (App. H.3).

FT-Transformer backbone. We replace the MLP with a timestep-conditioned FT-Transformer [21] and apply GOLIATH across all nine datasets at batch size 32 with five-epoch ensembling. Mean cell accuracy is 46.3% under Gaussian diffusion and 42.4% under mixed diffusion, compared with 53.7% and 51.4% for the MLP. The attack therefore **transfers across backbones**, although its accuracy depends on the architecture and dataset. On Abalone, the transformer exceeds the MLP (56.0% versus 51.3% under Gaussian diffusion), while on Adult it trails (46.3% versus 62.3%). App. H.4 reports results on all datasets, feature-type subscores, and the transformer-specific implementation and optimization budget.

4.4 Robustness Against Defenses

We evaluate GOLIATH at batch size 1 against four common gradient-perturbation defenses: gradient clipping [49], top-k compression [3], DP-style noise [19], and CENSOR [56]; (also see App. C). Fig. 3 reports reconstruction accuracy on two datasets per diffusion family: a continuous-heavy and a categorical-heavy schema under Gaussian diffusion and two categorical-heavy schemas under mixed diffusion. For the latter, we focus on categorical-heavy schemas, since the discrete noise mechanism on these features is what distinguishes masked diffusion from Gaussian diffusion. App. G.4 gives all accuracy and DCR results.

Clipping and compression **barely affect** GOLIATH, indicating that the attack does not rely only on unusually large or sparse gradient coordinates. Noising reduces both continuous and categorical



(a) Gaussian diffusion.

(b) Mixed diffusion.

Figure 4: Adult cross-epoch reconstruction subscores at batch size 32. Cubic Align: Hungarian.

recovery, as does CENSOR under Gaussian diffusion. **Categorical recovery remains substantial** under CENSOR with mixed diffusion on schemas with several categorical columns, where continuous recovery is also largely intact. This warrants future research into tabular specific defenses that take into account the usage of mixed loss functions. App. H.7 adds per-example clipping with calibrated Gaussian noise and single-release privacy accounting. At $\delta = 10^{-5}$ and $\varepsilon \in \{0.5, 1, 2, 4, 10\}$, cell accuracy drops to 11.5–14.4% on Adult and 20.2–21.9% on Credit. Separate defended-training runs show a substantial training-loss cost at batch size 1; these single-release budgets do not describe the privacy of the full training trajectory.

4.5 Cross-Epoch Ensembling and Alignment

We evaluate multi-epoch reconstruction at batch size 32, comparing our alignment heuristic (called Fast Align) with the Hungarian baseline of Sec. 3.5 and a no-alignment baseline. Fig. 4 shows that the heuristic **nearly matches Hungarian matching** and outperforms no-alignment, confirming that row matching is necessary across epochs; full alignment and DCR results appear in Apps. G.6 and G.5. On Adult under Gaussian diffusion, later epochs reduce the continuous subscore, consistent with weaker continuous-feature gradients after convergence; categorical recovery is less affected because it is already high before aggregation. App. H.6 evaluates randomized batch order: gains remain positive in the tested settings, although Gaussian diffusion benefits less on Adult. Apps. H.8 and H.9 further examine matching bias and accuracy versus observed epochs.

5 Conclusion

GOLIATH attacks Gaussian-only and mixed Gaussian-masked tabular diffusion by **jointly recovering data, timesteps, noise, and categorical masks**. Across the ten tabular datasets, its tabular-aware objective, timestep-mask coupling, and global greedy row alignment **outperform gradient-inversion baselines** under both diffusion families, while the effect of standard defenses depends on the defense, diffusion family, and schema.

Limitations. GOLIATH assumes a semi-honest server with access to individual client updates before secure aggregation. Cross-epoch gains rely on repeated batches: reshuffling roughly halves them on Adult under Gaussian diffusion. At batch size 32, only 0.2% of rows are fully correct, and the FT-Transformer trails the MLP on average. Calibrated noise suppresses recovery, but our budgets cover single releases, not full training. App. A discusses further limitations.

Acknowledgments

This research is partly funded by Priv-GSyn project, 200021E_229204, and Tracer project, 2000-1-242997, of SNSF, the DEPMAT project, P20-22 / N21022, of the Dutch Research Council, NVIDIA Academic Grant Program, and ASM international NV.

References

[1] Census-Income (KDD). UCI Machine Learning Repository, 2000. DOI: https://doi.org/10.24432/C5N30T.

[2] Dry Bean. UCI Machine Learning Repository, 2020. DOI: https://doi.org/10.24432/C50S4B.

[3] A. F. Aji and K. Heafield. Sparse communication for distributed gradient descent. *arXiv preprint arXiv:1704.05021*, 2017.

[4] B. Balle and Y.-X. Wang. Improving the Gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 394–403. PMLR, 2018. URL `https://proceedings.mlr.press/v80/balle18a.html`.

[5] M. Balunovic, D. Dimitrov, N. Jovanović, and M. Vechev. LAMP: Extracting text from gradients with language model priors. *Advances in Neural Information Processing Systems*, 35: 7641–7654, 2022.

[6] B. Becker and R. Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

[7] R. Bock. MAGIC Gamma Telescope. UCI Machine Learning Repository, 2004. DOI: https://doi.org/10.24432/C52C8B.

[8] V. Carletti, P. Foggia, C. Mazzocca, G. Parrella, and M. Vento. SoK: Gradient inversion attacks in federated learning. In *34th USENIX Security Symposium (USENIX Security 25)*, pages 6439–6459, 2025.

[9] M. de Goede, B. Cox, and J. Decouchant. Training diffusion models with federated learning. *arXiv preprint arXiv:2406.12575*, 2024.

[10] J. Deng, Y. Wang, J. Li, C. Wang, C. Shang, H. Liu, S. Rajasekaran, and C. Ding. TAG: Gradient attack on transformer-based language models. In *Findings of the association for computational linguistics: EMNLP 2021*, pages 3600–3610, 2021.

[11] P. Dhariwal and A. Q. Nichol. Diffusion models beat GANs on image synthesis. In M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 8780–8794, 2021. URL `https://proceedings.neurips.cc/paper/2021/hash/49ad23d1ec9fa4bd8d77d02681df5cfa-Abstract.html`.

[12] D. I. Dimitrov, M. Balunovic, N. Konstantinov, and M. Vechev. Data leakage in federated averaging. *Transactions on Machine Learning Research*, 2022.

[13] D. I. Dimitrov, M. Baader, M. Müller, and M. Vechev. SPEAR: Exact gradient inversion of batches in federated learning. *Advances in Neural Information Processing Systems*, 37: 106768–106799, 2024.

[14] J. Du, J. Hu, Z. Wang, P. Sun, N. Z. Gong, K. Ren, and C. Chen. SoK: On gradient leakage in federated learning. *arXiv preprint arXiv:2404.05403*, 2024.

[15] S. Elfwing, E. Uchibe, and K. Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11, 2018. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2017.12.012. URL `https://www.sciencedirect.com/science/article/pii/S0893608017302976`. Special issue on deep reinforcement learning.

[16] J. Fan, T. Liu, G. Li, J. Chen, Y. Shen, and X. Du. Relational data synthesis using generative adversarial networks: A design space exploration. *arXiv preprint arXiv:2008.12763*, 2020.

[17] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller. Inverting gradients-how easy is it to break privacy in federated learning? *Advances in neural information processing systems*, 33: 16937–16947, 2020.

[18] J. Geng, Y. Mou, F. Li, Q. Li, O. Beyan, S. Decker, and C. Rong. Towards general deep leakage in federated learning. *arXiv preprint arXiv:2110.09074*, 2021.

[19] R. C. Geyer, T. Klein, and M. Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

[20] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680, 2014. URL https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html.

[21] Y. Gorishniy, I. Rubachev, V. Khrulkov, and A. Babenko. Revisiting deep learning models for tabular data. In *Advances in Neural Information Processing Systems*, volume 34, pages 18932–18943. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/9d86d83f925f2149e9edb0ac3b49229c-Paper.pdf.

[22] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[23] H. Hofmann. Statlog (German Credit Data). UCI Machine Learning Repository, 1994. DOI: https://doi.org/10.24432/C5NC77. Accessed via OpenML credit-g, dataset ID 31: https://www.openml.org/data/v1/download/31/credit-g.arff.

[24] J. Huang, C. Hong, S. Roos, and L. Y. Chen. GIDM: Gradient inversion of federated diffusion models. In *International Conference on Availability, Reliability and Security*, pages 380–401. Springer, 2025.

[25] S. Kariyappa, C. Guo, K. Maeng, W. Xiong, G. E. Suh, M. K. Qureshi, and H.-H. S. Lee. Cocktail party attack: Breaking aggregation-based privacy in federated learning using independent component analysis. In *International Conference on Machine Learning*, pages 15884–15899. PMLR, 2023.

[26] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*, 2015.

[27] D. P. Kingma and M. Welling. Auto-encoding variational Bayes. In Y. Bengio and Y. LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114.

[28] A. Kotelnikov, D. Baranchuk, I. Rubachev, and A. Babenko. TabDDPM: Modelling tabular data with diffusion models. In *International conference on machine learning*, pages 17564–17579. PMLR, 2023.

[29] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[30] A. D. Lautrup, T. Hyrup, A. Zimek, and P. Schneider-Kamp. SynthEval: a framework for detailed utility and privacy evaluation of tabular synthetic data. *Data Mining and Knowledge Discovery*, 39(1):6, 2025.

[31] B. Li, H. Gu, R. Chen, J. Li, C. Wu, N. Ruan, X. Si, and L. Fan. Temporal gradient inversion attacks with robust optimization. *IEEE Transactions on Dependable and Secure Computing*, 22 (4):3383–3397, 2025.

[32] D. Li, W. Xie, Z. Wang, Y. Lu, Y. Li, and L. Fang. FedDiff: Diffusion model driven federated learning for multi-modal and multi-clients. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(10):10353–10367, 2024.

[33] R. Madeo, P. Wagner, and S. Peres. Gesture Phase Segmentation. UCI Machine Learning Repository, 2013. DOI: https://doi.org/10.24432/C5Z32C.

[34] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[35] S. S. Narasimhan, S. Agarwal, O. Akcin, S. Sanghavi, and S. P. Chinchali. Time weaver: A conditional time series generation model. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL `https://openreview.net/forum?id=WpKDeixmFr`.

[36] W. Nash, T. Sellers, S. Talbot, A. Cawthorn, and W. Ford. Abalone. UCI Machine Learning Repository, 1994. DOI: https://doi.org/10.24432/C55C7W.

[37] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[38] I. Petrov, D. I. Dimitrov, M. Baader, M. N. Müller, and M. Vechev. DAGER: Exact gradient inversion for large language models. *Advances in Neural Information Processing Systems*, 37: 87801–87830, 2024.

[39] C. Sakar and Y. Kastro. Online Shoppers Purchasing Intention Dataset. UCI Machine Learning Repository, 2018. DOI: https://doi.org/10.24432/C5F88Q.

[40] T. Sattarov, M. Schreyer, and D. Borth. Federated diffusion modeling with differential privacy for tabular data synthesis, 2024. URL `https://arxiv.org/abs/2412.16083`.

[41] T. Sattarov, M. Schreyer, and D. Borth. FedTabDiff: Federated learning of diffusion probabilistic models for synthetic mixed-type tabular data generation, 2024. URL `https://arxiv.org/abs/2401.06263`.

[42] A. Shankar, L. Chen, A. van Deursen, and R. Hai. WaveStitch: Flexible and fast conditional time series generation with diffusion models. *Proceedings of the ACM on Management of Data*, 3(6):1–25, 2025.

[43] A. Shankar, Y. Wang, R. Hai, and L. Y. Chen. Harpoon: Generalised manifold guidance for conditional tabular diffusion. In *The Fourteenth International Conference on Learning Representations*, 2026.

[44] J. Shi, M. Xu, H. Hua, H. Zhang, S. Ermon, and J. Leskovec. TabDiff: a mixed-type diffusion model for tabular data generation. In *The Thirteenth International Conference on Learning Representations*, 2025.

[45] D. Slate. Letter Recognition. UCI Machine Learning Repository, 1991. DOI: https://doi.org/10.24432/C5ZP40.

[46] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

[47] F. V. Stanley Jothiraj and A. Mashhadi. Phoenix: A federated generative diffusion model. In *Companion Proceedings of the ACM Web Conference 2024*, pages 1568–1577, 2024.

[48] M. Vero, M. Balunović, D. I. Dimitrov, and M. Vechev. TabLeak: tabular data leakage in federated learning. In *Proceedings of the 40th International Conference on Machine Learning*, pages 35051–35083, 2023.

[49] W. Wei, L. Liu, Y. Wu, G. Su, and A. Iyengar. Gradient-leakage resilient federated learning. In *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, pages 797–807. IEEE, 2021.

[50] J. Xu, C. Hong, J. Huang, L. Y. Chen, and J. Decouchant. AGIC: Approximate gradient inversion attack on federated learning. In *2022 41st International Symposium on Reliable Distributed Systems (SRDS)*, pages 12–22. IEEE, 2022.

[51] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni. Modeling tabular data using conditional GAN. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 7333–7343, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/254ed7d2de3b23ab10936522dd547b78-Abstract.html.

[52] I.-C. Yeh. Default of Credit Card Clients. UCI Machine Learning Repository, 2009. DOI: https://doi.org/10.24432/C55S3H.

[53] H. Yin, A. Mallya, A. Vahdat, J. M. Alvarez, J. Kautz, and P. Molchanov. See through gradients: Image batch recovery via GradInversion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16337–16346, 2021.

[54] H. Zhang, J. Zhang, Z. Shen, B. Srinivasan, X. Qin, C. Faloutsos, H. Rangwala, and G. Karypis. Mixed-type tabular data synthesis with score-based diffusion in latent space. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=4Ay23yeuz0.

[55] H. Zhang, L. Fang, Q. Wu, and P. S. Yu. DiffPuter: Empowering diffusion models for missing data imputation. In *The Thirteenth International Conference on Learning Representations*, 2025.

[56] K. Zhang, S. Cheng, G. Shen, B. Ribeiro, S. An, P.-Y. Chen, X. Zhang, and N. Li. CENSOR: Defense against gradient inversion via orthogonal subspace Bayesian sampling. In *32nd Annual Network and Distributed System Security Symposium (NDSS)*, 2025.

[57] B. Zhao, K. R. Mopuri, and H. Bilen. iDLG: Improved deep leakage from gradients. *arXiv preprint arXiv:2001.02610*, 2020.

[58] L. Zhu, Z. Liu, and S. Han. Deep leakage from gradients. *Advances in neural information processing systems*, 32, 2019.

A Impact & Limitations

Impact. This work studies an offensive privacy attack to expose risks in federated or collaborative tabular diffusion training. Its main positive impact is to show that sharing gradients can leak sensitive tabular records even when raw data is never transmitted, which can help in evaluating whether additional protections are needed before deploying such systems. As research of attacks and defenses goes hand in hand, we also hope that the robustness experiments can inform the design of future defenses and the understanding of their limitations. The same attack could also be misused by a malicious or honest-but-curious server that has access to client gradients. To reduce this risk, we evaluate only public datasets, release no private data or pretrained victim models, and include robustness experiments against common gradient defenses, so that the possible implications can be better understood.

Federated tabular diffusion in practice. Institutions with a shared schema can train a generator collaboratively while retaining their original records locally. For example, medical or financial datasets can support synthetic-data generation for downstream model development without first centralizing the source tables. FedTabDiff [41] evaluates this approach on financial and medical data, and DP-FedTabDiff [40] studies its combination with differential privacy. These works motivate the setting; keeping records local alone does not establish privacy of the updates exchanged during training. Our semi-honest server follows the training protocol and attempts reconstruction from the updates it receives, without modifying the model to induce leakage.

Backbone scope. Our main comparison uses a shared MLP backbone with timestep conditioning for both Gaussian and mixed diffusion. The FT-Transformer experiments establish transfer to a second architecture, with lower average reconstruction accuracy and substantial dataset dependence (App. H.4). They do not establish the same accuracy for all tabular diffusion architectures or conditioning mechanisms.

Gradient access and repeated batches. The attack assumes that the server observes client gradients before secure aggregation or other cryptographic protections are applied. The default cross-epoch ensemble assumes fixed repeated batches; reshuffling reduces its gain in some settings (App. H.6). The single-gradient attack is independent of batch order, and App. H.5 evaluates updates over several local steps on a fixed client subset. These experiments do not cover secure aggregation across clients or establish recovery under arbitrary local training schedules.

Defense coverage. The defense experiments are intended as robustness tests against common gradient perturbations, not as a complete privacy evaluation. In particular, the DP-noise setting adds Gaussian noise directly to gradients without a clipping-and-accounting pipeline, so it should not be interpreted as a certified (ε, δ)-DP guarantee. App. H.7 separately evaluates clipped, calibrated Gaussian releases and defended training. The reported privacy budgets apply to one release with fixed clipping parameters, not the complete training run or the data-dependent calibration procedure. Training loss is only a proxy for utility; we do not evaluate downstream synthetic-data quality under this defense.

Leakage metrics. Cell accuracy and DCR summarize leakage at the table level. Our per-column, cardinality, and row-level measurements expose differences hidden by the aggregate scores (Sec. 4.3, App. H.2). The nearest-neighbour diagnostic uses only the attacked batch and the evaluation alignment; it is not population-level identity recovery. High categorical accuracy also depends on category frequencies, and should not be interpreted as an attribute-inference advantage over a frequency-based predictor.

Table 4: Overview of datasets showing feature types (categorical vs. continuous), post-encoding dimensions, dataset scale, schema type (categorical-heavy vs continuous-heavy), and license.

Dataset	# cat.	# con.	no. encod.	# samples	Type	License
abalone [36]	1	8	11	4177	continuous-heavy	CC BY 4.0
adult [6]	9	6	110	32561	categorical-heavy	CC BY 4.0
bean [2]	1	16	23	13611	continuous-heavy	CC BY 4.0
default [52]	3	21	34	30000	continuous-heavy	CC BY 4.0
gesture [33]	1	32	37	9522	continuous-heavy	CC BY 4.0
letter [45]	1	16	42	20000	continuous-heavy	CC BY 4.0
magic [7]	1	10	12	19020	continuous-heavy	CC BY 4.0
shoppers [39]	4	14	31	12330	continuous-heavy	CC BY 4.0
credit [23]	14	7	63	1000	categorical-heavy	CC BY 4.0
census [1]	34	7	490	141544	categorical-heavy	CC BY 4.0

B Datasets

Tab. 4 details the ten public datasets used in the evaluation; App. H.3 describes Census. The type column labels each dataset as *categorical-heavy* (more categorical than continuous columns) or *continuous-heavy* (the reverse). The UCI dataset pages list all dataset sources under the Creative Commons Attribution 4.0 International (CC BY 4.0) license. The Credit dataset is accessed as OpenML's `credit-g` dataset, whose metadata asks users to cite the original UCI Statlog German Credit Data source. Adult, Credit and Census are the only categorical-heavy schemas in Tab. 4; the rest are continuous-heavy. Mixed-diffusion experiments that depend specifically on the categorical channel (e.g., the knowledge-of-t-and-mask study in App. G.3) are run on Adult and Credit.

C Implementation Details

Shared model architecture. Both diffusion families use the same backbone: a fully connected network with timestep conditioning. The input $x \in \mathbb{R}^{d_{\text{in}}}$ is projected to a latent dimension of 1024, and the core is four linear layers with SiLU activations [15]. Diffusion timesteps are embedded with fixed sinusoidal positional embeddings using logarithmically spaced frequencies, passed through a two-layer MLP with SiLU activations to produce a 1024-dimensional embedding that is added to the projected input before the main MLP.

Gaussian training recipe. For Gaussian diffusion we follow DiffPuter [55] and treat all features (continuous and one-hot categorical) as continuous. The training loss is the MSE noise-prediction objective in Eq. 2. We train with Adam at learning rate 10^{-4} for 1000 epochs, using a linear noise schedule with $T = 500$ timesteps and $\alpha_1 = 0.9999 \to \alpha_T = 0.98$.

Mixed training recipe. For mixed diffusion we follow TabDiff [44]: continuous features use Gaussian noising (as above) and categorical features use discrete masking with a [M] absorbing state, whose marginal follows Eq. 3. The training objective is the split loss in Eq. 4: MSE on the continuous part and cross-entropy on the masked categorical part, summed. All other hyperparameters (Adam, learning rate 10^{-4}, 1000 epochs, $T = 500$) are shared with the Gaussian recipe. The backbone's output head has one continuous head and one categorical head per categorical column, so the two training channels deposit gradient on disjoint output rows; this is the partition that GOLIATH's double-cosine objective (Eq. 7) exploits.

Cross-epoch moving average. For the update in Eq. 9 we use a linearly increasing schedule for ρ, going from 0.5 at epoch 1 to 0.9 at the final epoch. Early on, current and historical reconstructions are equally weighted; later, historical reconstructions dominate, which stabilizes the running estimate as per-epoch constraints accumulate.

Attack initialization. We initialize the dummy data following the approach in TabLeak [48]. Continuous features in the dummy data X' are initialized from a uniform distribution $[-1, 1]$, while one-hot encoded categories are sampled from a standard normal distribution followed by a softmax to ensure that categories sum to 1. The noise ϵ' is initialized from a standard normal distribution. Timesteps are all initialized from the median value of the allowed range.

Comparison with GIDM. GOLIATH differs from GIDM [24] in two ways. It is single-phase and requires no pretrained generative model, which is rarely available for sensitive tabular domains such as medical records. It also optimizes t' as a continuous scalar rather than a one-hot vector, making the inversion agnostic to the diffusion schedule length. For non-integer t', timestep-indexed schedules are evaluated by linear interpolation between the neighboring integer entries after clamping t' to $[0, T-1]$; this is used for $\bar{\alpha}_{t'}$ and, in mixed diffusion, for the timestep-dependent categorical loss weights.

Defense implementation. Tab. 5 summarizes the defense settings used in Sec. 4.4. For gradient clipping [49], we bound the observed gradient norm as $g \leftarrow g \cdot \min\left(1, \frac{b}{\|g\|}\right)$. For compression [3], we zero all coordinates whose absolute value falls below the selected per-tensor quantile. For DP noise [19], we add independent Gaussian noise to every gradient coordinate, $g_i \leftarrow g_i + z_i$ with

$z_i \sim \mathcal{N}(0, \sigma^2)$. This experiment should be interpreted as a Gaussian-noise robustness test rather than a certified DP mechanism: we do not combine it with clipping in the same defense branch, and we do not report an (ε, δ) privacy accountant. For CENSOR [56], we project gradients onto an orthogonal subspace and select the normalized orthogonal gradient that best preserves training utility. The perturbation experiments in Fig. 3 use batch size 1 and modify the attacker's observed gradients; App. H.7 separately evaluates calibrated releases and training on defended gradients.

Table 5: Defense settings used in the robustness experiments.

Defense	Setting	Implementation detail
Clipping	$b = 4$	Clip the full gradient list by global ℓ_2 norm.
Compression	10%	Zero entries below the per-tensor 10th absolute-value percentile.
DP noise	$\sigma = 0.1$	Add i.i.d. $\mathcal{N}(0, 0.1^2)$ noise to every gradient coordinate.
CENSOR	20 trials	Keep the orthogonal gradient candidate with the lowest training loss.

D Metrics

During our evaluation, we record two metrics: reconstruction accuracy and median Distance-to-Closest-Record (DCR).

D.1 Accuracy Metric

The accuracy metric is defined as the percentage of correctly reconstructed cells in the table, first proposed by Vero et al. [48]. Formally, given the original batch X and the reconstructed batch X' (already aligned), with K categorical discrete features x^D and X'^D and L continuous features x^C and X'^C, the accuracy is computed as:

$$\text{accuracy}(X, X') = \frac{1}{K+L}\left(\sum_{i=1}^{K}(X_i^D \equiv X_i'^D) + \sum_{i=1}^{L}(X_i^C \approx X_i'^C)\right)$$

Here, $\equiv$ is the binary equivalence function, and $\approx$ is a binary function that returns 1 or 0 depending on whether the values are considered close enough. We follow the same definition as TabLeak [48] and consider the reconstructed value $\hat{x}_i^C$ to be close enough if it falls in the range $[X_i^C - 0.319\sigma_i^C, X_i^C + 0.319\sigma_i^C]$. Here σ_i^C is the standard deviation of continuous feature i computed on the training split. The coefficients are chosen so that a Gaussian random variable n falls in the range with probability 25%. Reported means and standard deviations are computed over the inverted batches; throughout the paper, table entries of the form μ_σ denote the mean μ with standard deviation σ as a subscript. Unless stated otherwise, we evaluate all available batches when the dataset has at most 100 batches for a configuration, and otherwise evaluate a subset of 100 batches.

D.2 Distance-to-Closest-Record

We compute median Distance-to-Closest-Record (DCR) using the implementation from SynthEval [30]. This metric has been used extensively in the literature [16, 54] and is computed as follows:

$$DCR(X', X_{\text{ref}}) = \frac{\text{median}_{x_i' \in X'} \min_{x_j \in X_{\text{ref}}} d(x_i', x_j)}{\text{median}_{x_i \in X_{\text{ref}}} \min_{x_j \in X_{\text{ref}},\, j \neq i} d(x_i, x_j)},$$

where X_{ref} is the real batch and d is the Gower distance. This yields a non-negative value, where lower values indicate reconstructed rows closer to real records and therefore a higher level of data leakage. This indicates that the recovered batch captures information leaked from the gradient. Reported means and standard deviations are computed over the inverted batches.

E Algorithm(s)

The heuristic algorithm for row alignment is provided in Alg. 1.

Algorithm 1 Greedy Alignment

Require: Existing batch E, New batch O.
Ensure: Alignment indices $\mathbf{m}$, Similarity scores $\mathbf{s}$.
$C \leftarrow \mathbf{CostMatrix}(E, O)$ {Compute cost matrix}
$S \leftarrow \mathbf{1} - C$ {Convert cost to similarity}
$N \leftarrow \text{rows}(S)$
Initialize $\mathbf{m}$ and $\mathbf{s}$ as arrays of size N
for $\ell \leftarrow 1, \ldots, N$ **do**
  $(i, j) \leftarrow \text{argmax}_{r,c}\, S_{r,c}$ {Best still-valid pair}
  $\mathbf{m}[i] \leftarrow j$
  $\mathbf{s}[i] \leftarrow S_{i,j}$
  $S_{i,:} \leftarrow -\infty$ {Mask row i to prevent re-selection}
  $S_{:,j} \leftarrow -\infty$ {Mask column j to prevent re-selection}
end for
Return $\mathbf{m}, \mathbf{s}$

Gaussian training (Alg. 2). For each row x_0, sample a diffusion step $t \sim U[1, T]$ and Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$, and update the model by the MSE between predicted and true noise [22].

Algorithm 2 Gaussian Diffusion Model Training

repeat
  Sample data point $x_0 \sim p(x_0)$
  Sample timestep $t \sim \text{Uniform}(\{1, \ldots, T\})$
  Sample noise $\epsilon \sim \mathcal{N}(0, I)$
  Noise prediction $\epsilon_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t}\, \epsilon, t)$
  Update model parameters: $\nabla_\theta \left\| \epsilon - \epsilon_\theta(x_t, t) \right\|_2^2$
until convergence

Mixed training (Alg. 3). Continuous features are noised as in the Gaussian recipe; each categorical cell is replaced by [M] with probability $1 - \bar{\alpha}_t$ and otherwise kept. The training loss has two channels: MSE on the continuous noise target and cross-entropy on the masked categorical logits (Eq. 4).

Algorithm 3 Mixed Diffusion Model Training

repeat
  Sample data point $x_0 = (x_0^{\text{cont}}, x_0^{\text{cat}}) \sim p(x_0)$
  Sample timestep $t \sim \text{Uniform}(\{1, \ldots, T\})$
  Sample continuous noise $\epsilon \sim \mathcal{N}(0, I)$
  For each categorical cell d, draw $b_d \sim \text{Bernoulli}(1 - \bar{\alpha}_t)$; set $\tilde{x}_0^{\text{cat},d} = \text{[M]}$ if $b_d = 1$ else $x_0^{\text{cat},d}$
  Form noised input $x_t = (\sqrt{\bar{\alpha}_t} x_0^{\text{cont}} + \sqrt{1 - \bar{\alpha}_t}\, \epsilon, \tilde{x}_0^{\text{cat}})$
  Predict $(\hat{\epsilon}, \hat{\ell}^{\text{cat}}) \leftarrow \epsilon_\theta(x_t, t)$
  Update model parameters: $\nabla_\theta \left[\left\| \epsilon - \hat{\epsilon} \right\|_2^2 + w_d(t)\, \text{CE}\Big(\hat{\ell}^{\text{cat}}, x_0^{\text{cat}} \Big)_{\{d : b_d = 1\}} \right]$
until convergence

In our implementation $w_c(t) \equiv 1$ and $w_d(t)$ is the ELBO weight at step t tied to the noise schedule (Eq. 4).

F DCR Results

Tabs. 6 and 7 report median Distance-to-Closest-Record (DCR) under Gaussian and mixed diffusion on nine datasets, broadly following the accuracy ordering of Tabs. 1 and 2 for the methods shown; batch-size-1 rows use a different scale (App. D). Lower DCR indicates that reconstructed rows are closer to true records, signaling stronger leakage.

Table 6: DCR (↓, mean and standard deviation) across different batch sizes and datasets. (Gaussian Diffusion)

Batch Size	Method	Abalone	Adult	Bean	Default	Gesture	Letter	Magic	Shoppers	Credit
1	TabLeak	$0.94_{0.1}$	$0.81_{0.1}$	$0.98_{0.0}$	$0.95_{0.0}$	$0.99_{0.0}$	$0.99_{0.0}$	$0.96_{0.0}$	$0.91_{0.1}$	$0.75_{0.2}$
	GIDM	$0.95_{0.1}$	$0.91_{0.1}$	$0.99_{0.0}$	$0.95_{0.0}$	$0.99_{0.0}$	$1.00_{0.0}$	$0.94_{0.0}$	$0.89_{0.1}$	$0.77_{0.1}$
	GOLIATH 1-epoch	$0.91_{0.0}$	$0.50_{0.2}$	$0.96_{0.0}$	$0.90_{0.0}$	$0.98_{0.0}$	$0.95_{0.0}$	**$0.92_{0.0}$**	**$0.80_{0.0}$**	$0.40_{0.1}$
	GOLIATH 5-epochs	**$0.91_{0.0}$**	**$0.43_{0.1}$**	**$0.95_{0.0}$**	**$0.88_{0.0}$**	**$0.97_{0.0}$**	**$0.94_{0.0}$**	$0.93_{0.0}$	$0.81_{0.1}$	**$0.37_{0.1}$**
8	TabLeak	$5.86_{2.7}$	$2.38_{0.7}$	$7.14_{3.4}$	$5.41_{2.3}$	$8.21_{4.0}$	$3.00_{0.6}$	$5.00_{1.6}$	$4.65_{2.1}$	$1.74_{0.4}$
	GIDM	$4.00_{1.6}$	$2.26_{0.6}$	$17.03_{15.1}$	$3.84_{1.6}$	$4.75_{2.1}$	$1.57_{0.3}$	$2.61_{0.9}$	$3.12_{0.9}$	$1.58_{0.3}$
	GOLIATH 1-epoch	$1.54_{0.5}$	$0.79_{0.4}$	$1.22_{0.5}$	$1.09_{0.4}$	$1.24_{0.6}$	$0.65_{0.2}$	$1.39_{0.4}$	$1.12_{0.4}$	**$0.63_{0.2}$**
	GOLIATH 5-epochs	**$0.92_{0.3}$**	**$0.75_{0.2}$**	**$0.94_{0.3}$**	**$0.85_{0.2}$**	**$0.95_{0.5}$**	**$0.60_{0.1}$**	**$0.93_{0.2}$**	**$0.92_{0.2}$**	$0.65_{0.1}$
16	TabLeak	$7.07_{2.3}$	$2.54_{0.5}$	$7.72_{2.5}$	$5.25_{1.5}$	$8.32_{4.2}$	$3.23_{1.0}$	$5.45_{1.3}$	$5.00_{1.9}$	$1.88_{0.3}$
	GIDM	$4.90_{2.0}$	$2.89_{0.5}$	$40.97_{25.4}$	$5.25_{1.8}$	$7.43_{3.6}$	$1.99_{0.4}$	$3.48_{1.0}$	$3.75_{1.2}$	$1.79_{0.3}$
	GOLIATH 1-epoch	$1.71_{0.4}$	$1.05_{0.3}$	$1.79_{0.5}$	$1.37_{0.3}$	$1.39_{0.5}$	$0.75_{0.1}$	$1.43_{0.2}$	$1.24_{0.3}$	**$0.72_{0.1}$**
	GOLIATH 5-epochs	**$1.05_{0.2}$**	**$0.87_{0.2}$**	**$1.32_{0.3}$**	**$1.04_{0.2}$**	**$1.03_{0.3}$**	**$0.73_{0.1}$**	**$0.98_{0.1}$**	**$0.98_{0.1}$**	$0.73_{0.1}$
32	TabLeak	$8.69_{2.3}$	$2.71_{0.4}$	$10.45_{2.2}$	$6.47_{1.2}$	$7.85_{2.2}$	$4.44_{1.2}$	$5.95_{1.4}$	$6.25_{2.2}$	$2.03_{0.2}$
	GIDM	$6.26_{3.0}$	$3.25_{0.2}$	$60.02_{38.1}$	$8.37_{2.2}$	$9.74_{3.4}$	$3.55_{1.9}$	$4.00_{1.5}$	$5.83_{2.7}$	$1.95_{0.2}$
	GOLIATH 1-epoch	$1.82_{0.3}$	$1.26_{0.2}$	$2.74_{0.5}$	$1.63_{0.3}$	$1.68_{0.4}$	$0.96_{0.2}$	$1.43_{0.2}$	$1.58_{0.3}$	$0.99_{0.1}$
	GOLIATH 5-epochs	**$1.18_{0.2}$**	**$0.90_{0.1}$**	**$1.87_{0.3}$**	**$1.16_{0.2}$**	**$1.34_{0.3}$**	**$0.86_{0.1}$**	**$1.02_{0.1}$**	**$1.10_{0.1}$**	**$0.82_{0.1}$**

Table 7: DCR (↓, mean and standard deviation) across different batch sizes and datasets. (Masked Diffusion)

Batch Size	Method	Abalone	Adult	Bean	Default	Gesture	Letter	Magic	Shoppers	Credit
1	Random Mask	$0.94_{0.1}$	$0.74_{0.1}$	$0.97_{0.0}$	$0.92_{0.0}$	$0.98_{0.0}$	$0.96_{0.0}$	$0.93_{0.0}$	$0.86_{0.0}$	$0.65_{0.1}$
	TabLeak	$0.96_{0.1}$	$0.82_{0.1}$	$0.99_{0.0}$	$0.95_{0.0}$	$0.99_{0.0}$	$0.99_{0.0}$	$0.95_{0.0}$	$0.90_{0.1}$	$0.76_{0.1}$
	GOLIATH 1-epoch	$0.93_{0.1}$	$0.60_{0.2}$	$0.95_{0.0}$	$0.90_{0.0}$	$0.98_{0.0}$	$0.95_{0.0}$	$0.94_{0.0}$	$0.83_{0.0}$	$0.52_{0.2}$
	GOLIATH 5-epochs	**$0.91_{0.0}$**	**$0.45_{0.1}$**	**$0.94_{0.0}$**	**$0.88_{0.0}$**	**$0.97_{0.0}$**	**$0.94_{0.0}$**	**$0.92_{0.0}$**	**$0.79_{0.0}$**	**$0.36_{0.1}$**
8	Random Mask	$3.25_{2.8}$	$1.51_{0.4}$	$3.69_{2.2}$	$1.73_{0.6}$	$4.48_{2.7}$	$1.91_{1.0}$	$3.00_{1.7}$	$1.80_{0.7}$	$1.21_{0.2}$
	TabLeak	$6.39_{3.5}$	$2.03_{0.6}$	$7.28_{4.3}$	$5.73_{3.8}$	$10.54_{8.3}$	$2.65_{0.9}$	$7.07_{4.3}$	$5.28_{5.4}$	$1.67_{0.4}$
	GOLIATH 1-epoch	$1.36_{0.5}$	$1.18_{0.4}$	$1.34_{0.6}$	$1.17_{0.3}$	$1.38_{0.7}$	$0.83_{0.2}$	$1.14_{0.3}$	$1.33_{0.4}$	$1.01_{0.3}$
	GOLIATH 5-epochs	**$1.02_{0.4}$**	**$0.91_{0.2}$**	**$0.90_{0.3}$**	**$0.88_{0.2}$**	**$0.91_{0.4}$**	**$0.63_{0.1}$**	**$0.90_{0.2}$**	**$0.95_{0.2}$**	**$0.83_{0.2}$**
16	Random Mask	$2.99_{1.6}$	$1.77_{0.4}$	$4.59_{1.9}$	$1.87_{0.7}$	$4.63_{3.1}$	$1.77_{0.7}$	$2.62_{1.0}$	$1.76_{0.4}$	$1.33_{0.2}$
	TabLeak	$9.22_{4.4}$	$2.49_{0.5}$	$10.23_{3.6}$	$6.52_{2.6}$	$9.71_{6.1}$	$2.75_{1.9}$	$6.73_{3.1}$	$5.56_{4.2}$	$1.78_{0.3}$
	GOLIATH 1-epoch	$1.46_{0.4}$	$1.42_{0.3}$	$2.13_{0.6}$	$1.47_{0.2}$	$1.29_{0.5}$	$0.96_{0.2}$	$1.27_{0.2}$	$1.43_{0.3}$	$1.10_{0.2}$
	GOLIATH 5-epochs	**$1.10_{0.3}$**	**$1.08_{0.2}$**	**$1.37_{0.3}$**	**$1.07_{0.2}$**	**$0.98_{0.3}$**	**$0.79_{0.1}$**	**$1.06_{0.2}$**	**$1.03_{0.2}$**	**$0.90_{0.1}$**
32	Random Mask	$2.72_{0.6}$	$1.87_{0.2}$	$4.90_{1.4}$	$1.93_{0.4}$	$3.64_{2.1}$	$1.78_{0.5}$	$2.33_{0.7}$	$1.73_{0.4}$	$1.40_{0.1}$
	TabLeak	$12.22_{3.4}$	$2.75_{0.2}$	$12.65_{5.2}$	$9.53_{3.3}$	$8.66_{3.4}$	$5.17_{2.3}$	$7.82_{2.5}$	$10.52_{5.1}$	$1.93_{0.3}$
	GOLIATH 1-epoch	$1.65_{0.3}$	$1.71_{0.2}$	$3.15_{0.6}$	$1.89_{0.3}$	$1.72_{0.5}$	$1.16_{0.2}$	$1.34_{0.2}$	$1.89_{0.3}$	$1.25_{0.1}$
	GOLIATH 5-epochs	**$1.36_{0.3}$**	**$1.30_{0.2}$**	**$1.77_{0.3}$**	**$1.21_{0.1}$**	**$1.13_{0.3}$**	**$0.94_{0.1}$**	**$1.17_{0.2}$**	**$1.14_{0.1}$**	**$0.97_{0.1}$**

G Additional Experiments

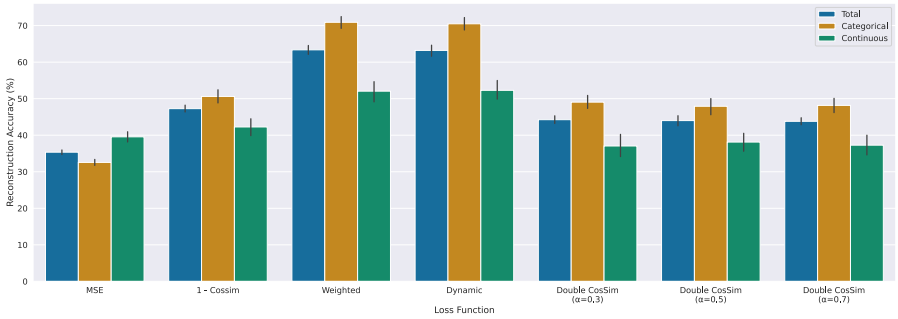
This appendix collects the ablations of the individual components and additional reconstruction plots that were cut from the main body for space. Unless stated otherwise, plots report mean reconstruction accuracy over all cells (Total) and separately for categorical and continuous cells, with 95% confidence intervals.

G.1 Ablation: Gradient Distance Functions

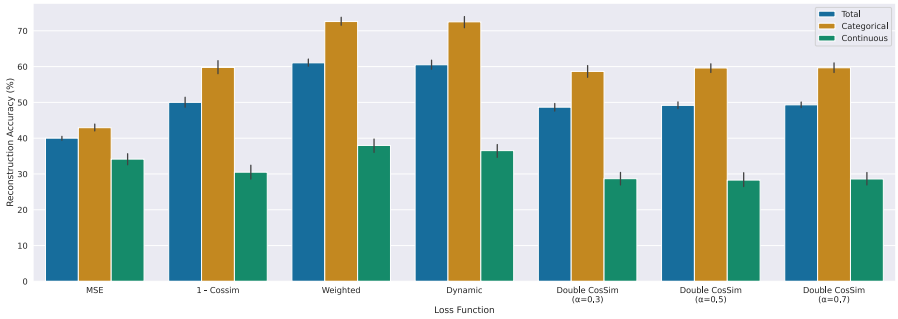
We compare gradient distances at batch size 16 over two attacked epochs, and each bar pools both epochs. These are: pure MSE, pure $1 - \mathrm{CosSim}$, the EMA-scaled combined distance of Eq. 6 (Weighted), and the MSE/CE split double cosine of Eq. 7 (Double-CosSim) with varying α. Fig. 5 reports Gaussian-diffusion results on all nine datasets, and Fig. 6 reports the corresponding mixed-diffusion results. Dynamic chooses the distance from the diffusion family and schema: it uses MSE when the categorical encoding has at most three one-hot dimensions (Abalone and Magic), Double-CosSim with the α of Sec. 3.4 under mixed diffusion when categorical columns outnumber continuous ones (Adult and Credit), and Weighted otherwise. Under Gaussian diffusion, Double-CosSim still splits the output gradient into the rows corresponding to continuous and categorical features, although both parts are produced by the MSE training objective.

Under Gaussian diffusion, Weighted and Dynamic perform best, averaging 50.4% and 50.3% across the nine datasets. In comparison, $1 - \mathrm{CosSim}$ reaches 41.9%, MSE 39.9%, and the best Double-CosSim setting 41.4%. The gap is largest on the categorical-heavy schemas: Weighted reaches 63.4% on Adult and 61.0% on Credit, while MSE reaches 35.4% and 40.0%. Abalone and Magic are the exception, as MSE matches Weighted on both (45.6% versus 45.1% and 45.9% versus 45.5%). These schemas have few categorical dimensions, so most of the signal is continuous and the cosine term likely adds little. Double-CosSim brings no benefit under Gaussian diffusion, likely because both of its parts come from the same MSE objective.

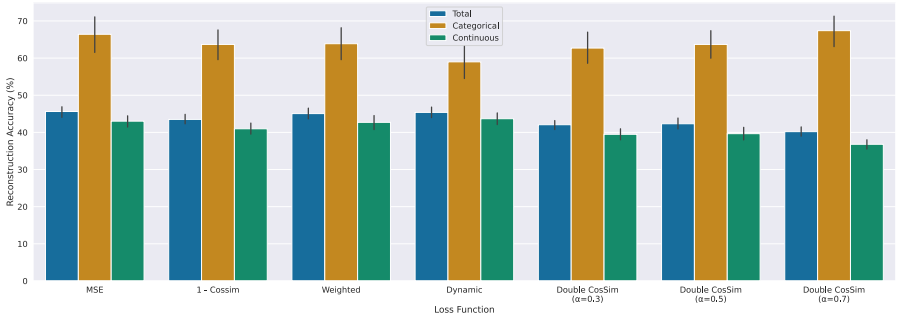
Under mixed diffusion, Dynamic and Weighted again lead, averaging 45.2% and 45.0%, compared with 40.4% for MSE, 39.6% for the best Double-CosSim setting, and 38.9% for $1 - \mathrm{CosSim}$. The benefit of the split objective depends on the schema. Double-CosSim is best on Adult, reaching 48.0% at $\alpha = 0.3$ versus 44.7% for Weighted, and is comparable to Weighted on Credit (51.6% versus 51.2%). On the other seven datasets, it trails Weighted by 0.5 to 12.2 percentage points. No single distance suffices because each favors different feature types: on Adult, MSE has the highest continuous score (40.8%, versus at most 25.7% for the others) but the lowest categorical score (32.6%, versus at least 58.8%). This motivates Dynamic, which stays within two percentage points of the best distance on every dataset under both diffusion families. Differences of a few points should be read with care: Dynamic and Weighted share the same configuration on seven Gaussian-diffusion datasets, yet their means differ by up to 3.1 percentage points on Shoppers.



(a) Adult (categorical-heavy).

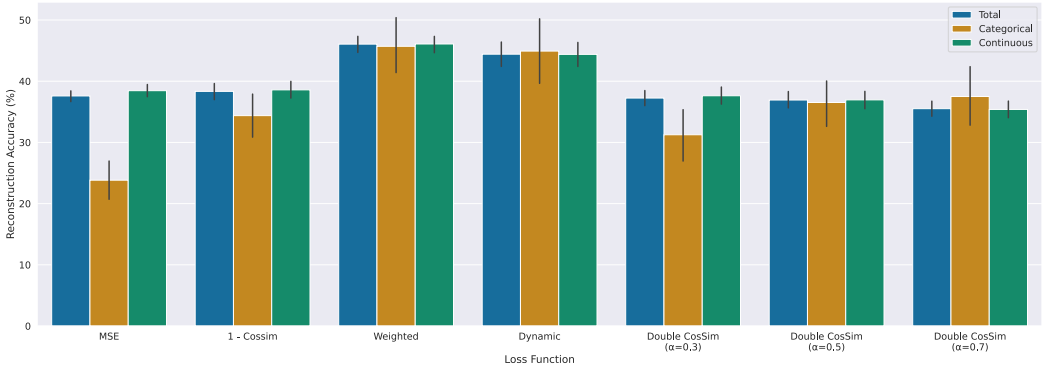


(b) Credit (categorical-heavy).

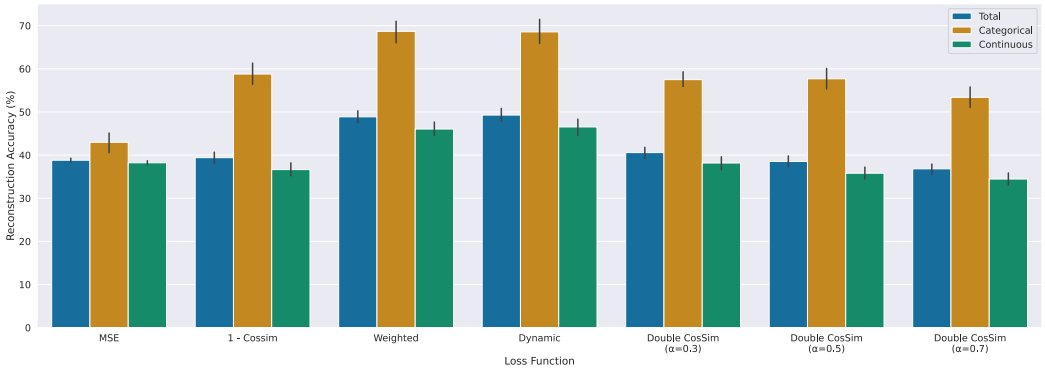


(c) Abalone.

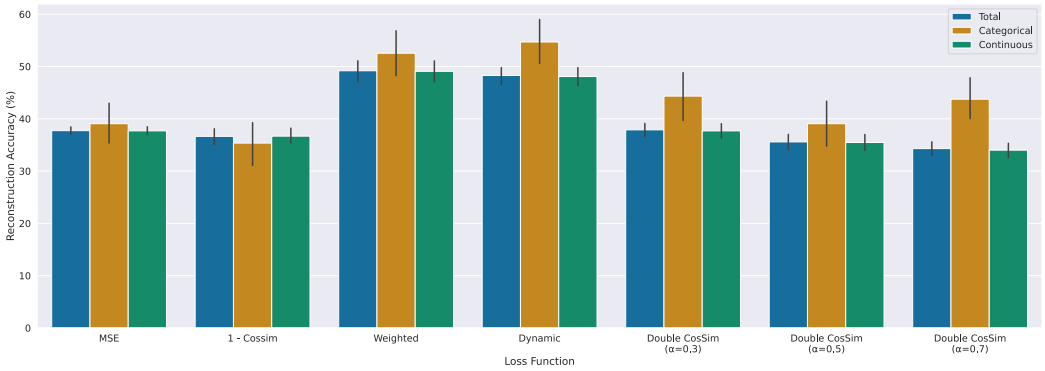
Figure 5: Gradient distance ablation under Gaussian diffusion at batch size 16, comparing MSE, $1 - \mathrm{CosSim}$, the EMA-scaled combined distance (Weighted), Dynamic, and Double-CosSim (varying α) on all nine datasets. For Gaussian diffusion, Double-CosSim splits continuous-feature and categorical-feature output rows even though both signals come from the MSE objective. Categorical-heavy datasets first, then continuous-heavy.



(d) Bean.

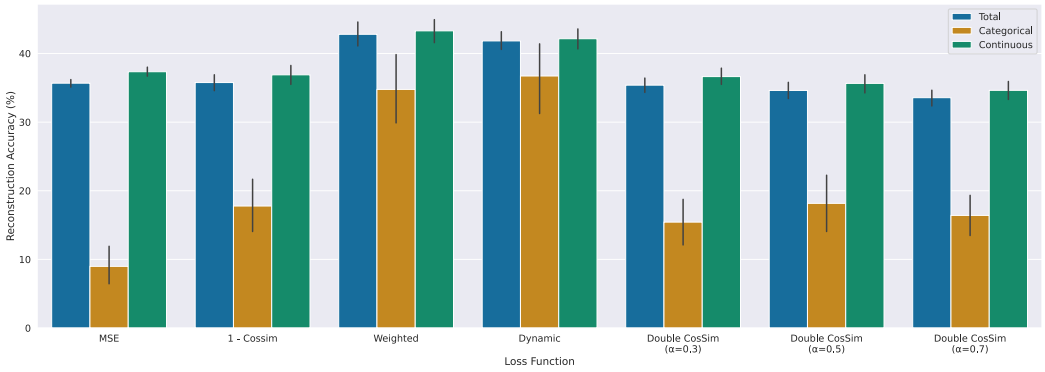


(e) Default.

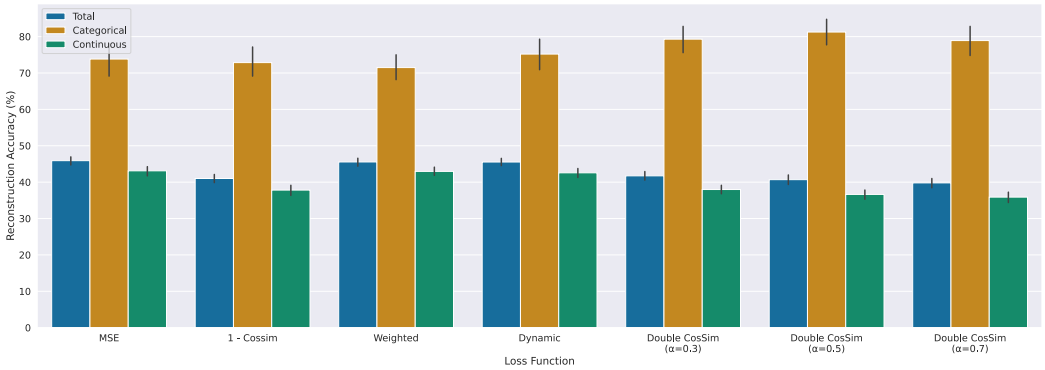


(f) Gesture.

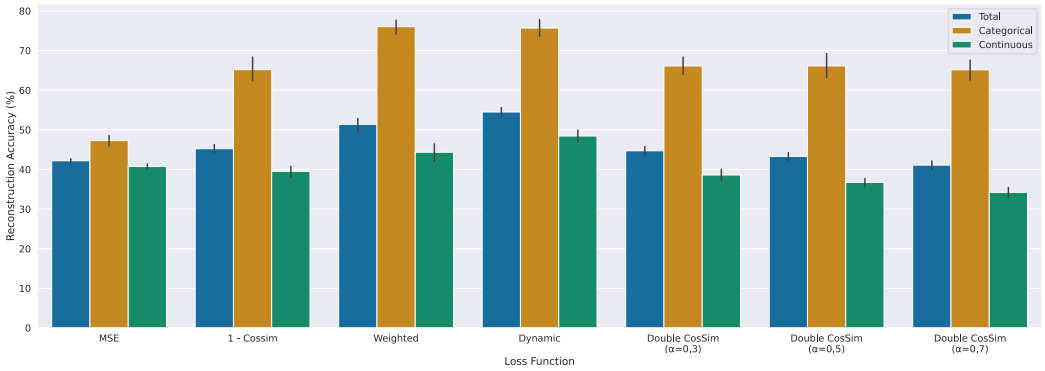
Figure 5: Gradient distance ablation under Gaussian diffusion (continued).



(g) Letter.

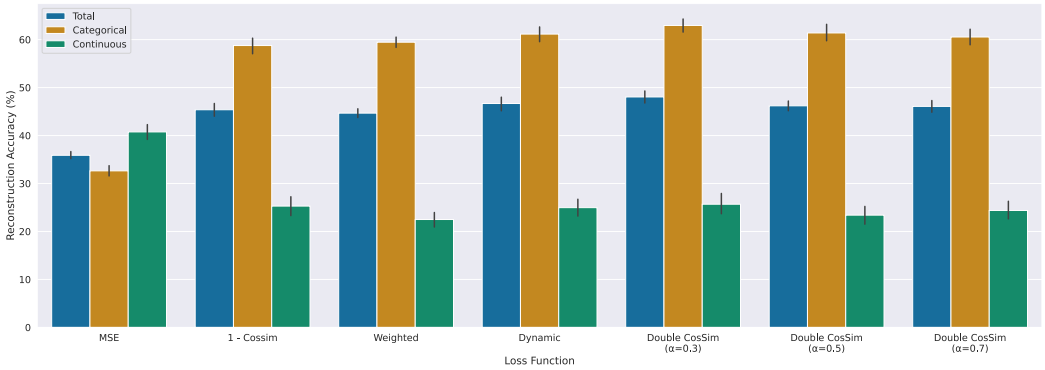


(h) Magic.

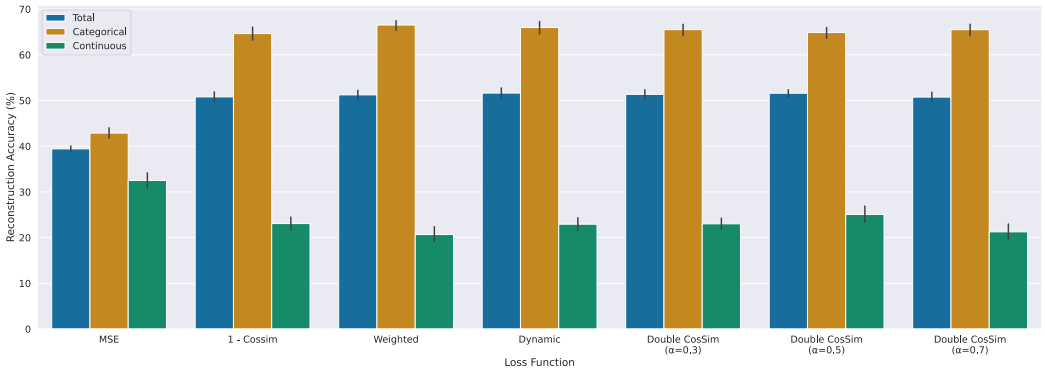


(i) Shoppers.

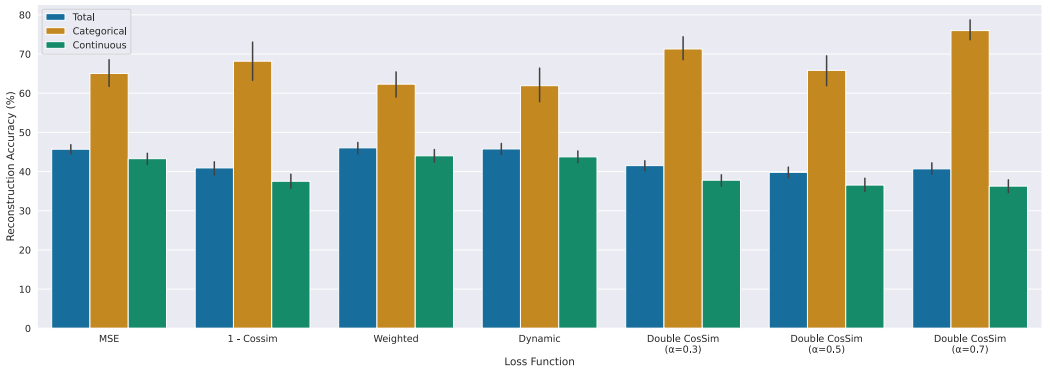
Figure 5: Gradient distance ablation under Gaussian diffusion (continued).



(a) Adult (categorical-heavy).

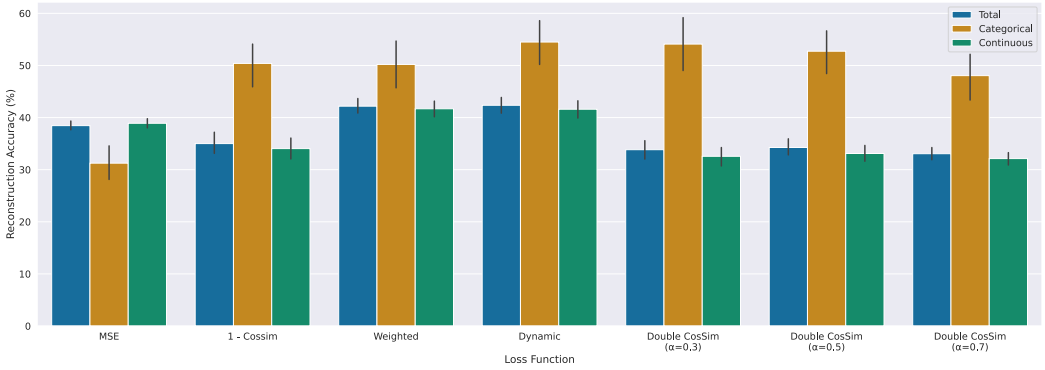


(b) Credit (categorical-heavy).

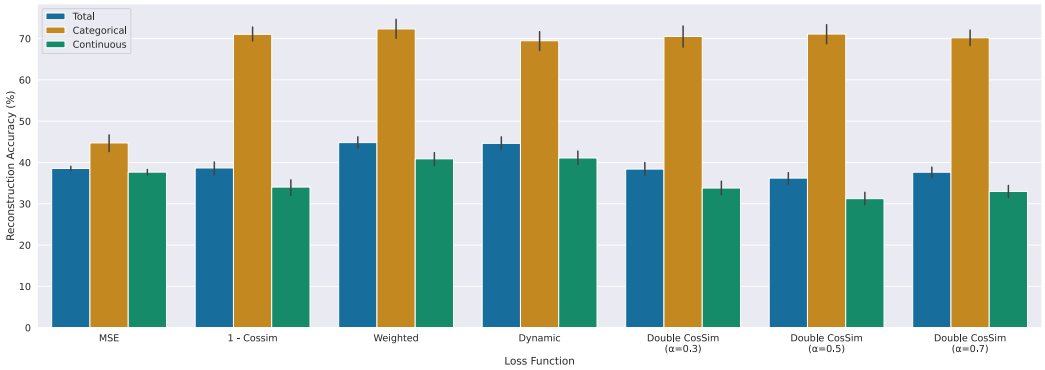


(c) Abalone.

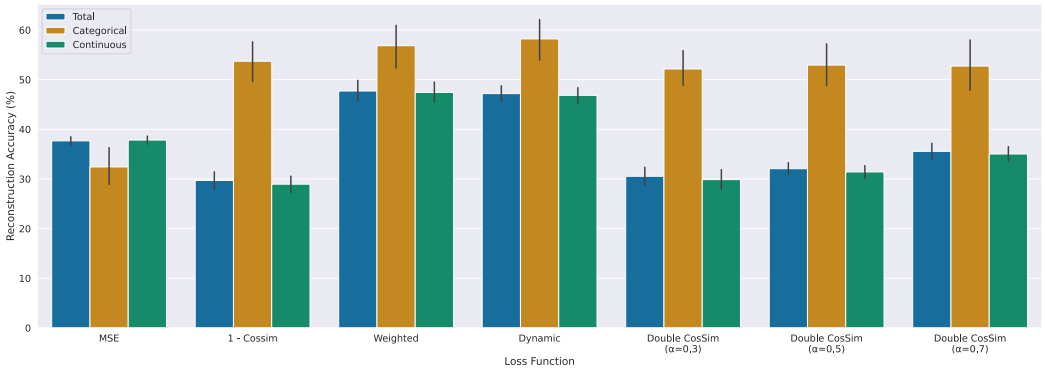
Figure 6: Gradient distance ablation under mixed diffusion at batch size 16, comparing MSE, the single cosine over the full gradient, the EMA-scaled combined distance (Weighted), Dynamic, and the MSE/CE split double cosine (varying α) on all nine datasets. Categorical-heavy datasets first, then continuous-heavy.



(d) Bean.

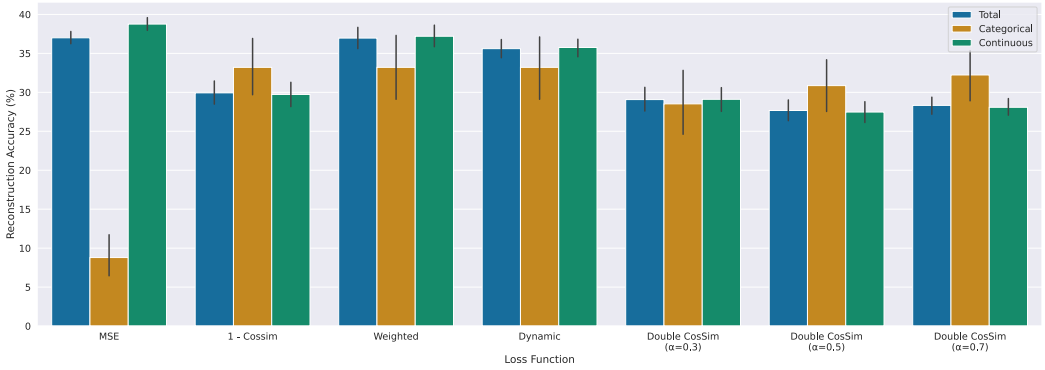


(e) Default.

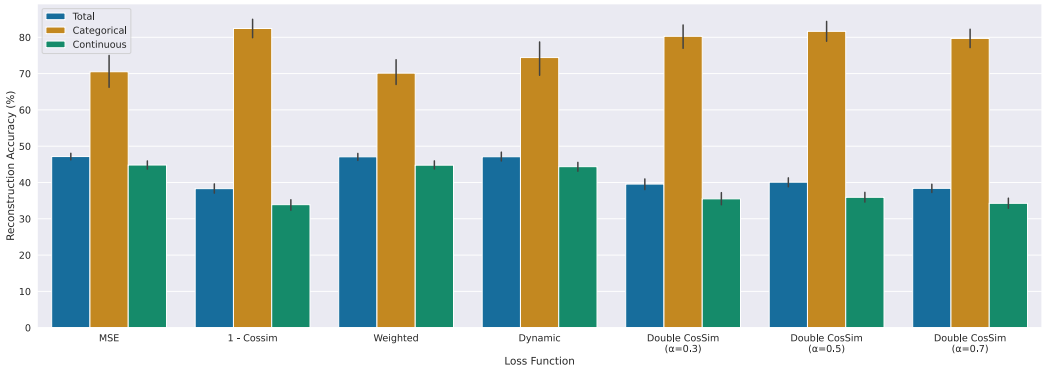


(f) Gesture.

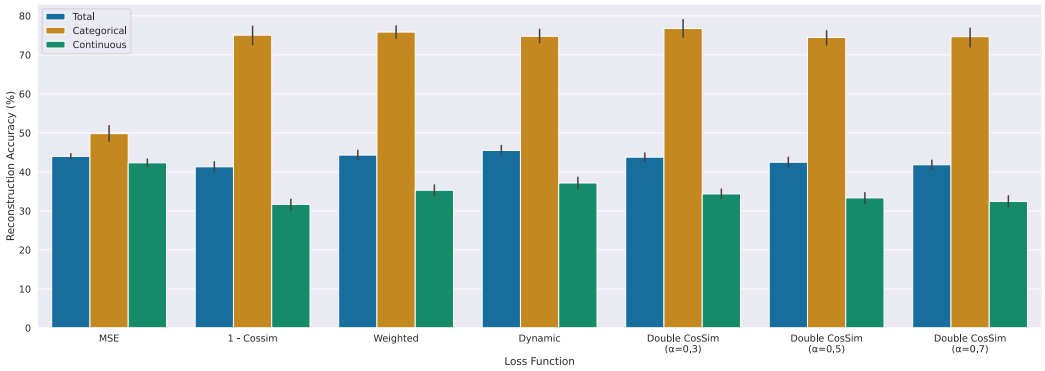
Figure 6: Gradient distance ablation under mixed diffusion (continued).



(g) Letter.



(h) Magic.



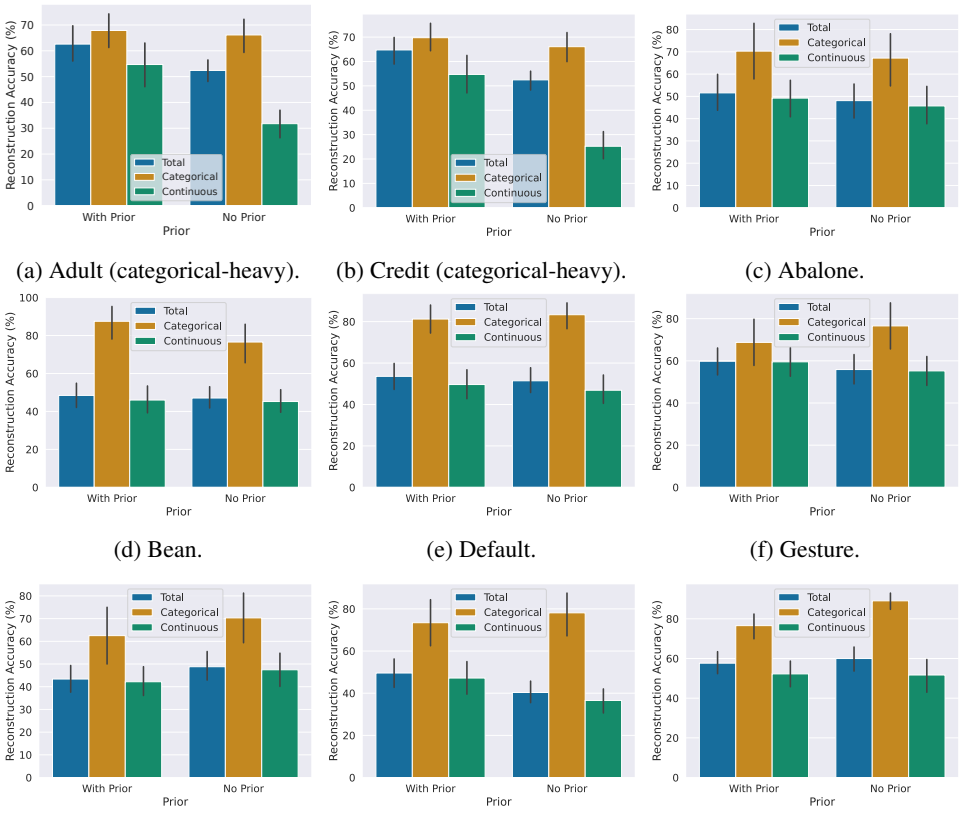
(i) Shoppers.

Figure 6: Gradient distance ablation under mixed diffusion (continued).

G.2 Ablation: Timestep-Mask KL Prior

We isolate the effect of the KL timestep-mask prior (Eq. 8) at batch size 1 under mixed diffusion. The prior is added to the gradient-matching objective with the schedule described in Sec. 3.4. Fig. 7 compares GOLIATH with and without the prior on the nine datasets.

The prior improves total accuracy on seven of the nine datasets, with an average change of $+3.9$ percentage points. The gains are clearest on Credit ($+12.3$), Adult ($+10.2$), and Magic ($+9.2$), where they exceed two standard errors. The remaining gains are within about one standard error, and accuracy drops on Letter (-5.4) and Shoppers (-2.3). Most of the gain comes from continuous features, which improve by 29.5, 22.9, and 10.6 percentage points on Credit, Adult, and Magic. Categorical accuracy changes by less than four percentage points on Adult and Credit. On the other datasets, categorical changes are larger but noisier, from -12.5 on Shoppers to $+10.9$ on Bean, since these schemas have few categorical cells at batch size 1. Only the Shoppers change exceeds two standard errors. This matches the intended mechanism: when the recovered mask density is high (mostly-masked rows), the prior pulls t' toward a larger value, which repositions the continuous noise-prediction target and gives the MSE channel a chance to recover the underlying numerical values. Timestep recovery supports this reading: the prior reduces timestep MAE from 138.3 to 40.5 on Adult and from 126.9 to 44.7 on Credit, with much smaller changes elsewhere (e.g., 127.4 to 104.6 on Magic and 117.8 to 124.4 on Letter). The effect is therefore largest on schemas with many categorical columns, where mask density likely says more about t, although Magic also benefits.



(a) Adult (categorical-heavy). (b) Credit (categorical-heavy). (c) Abalone.

(d) Bean. (e) Default. (f) Gesture.

(g) Letter. (h) Magic. (i) Shoppers.

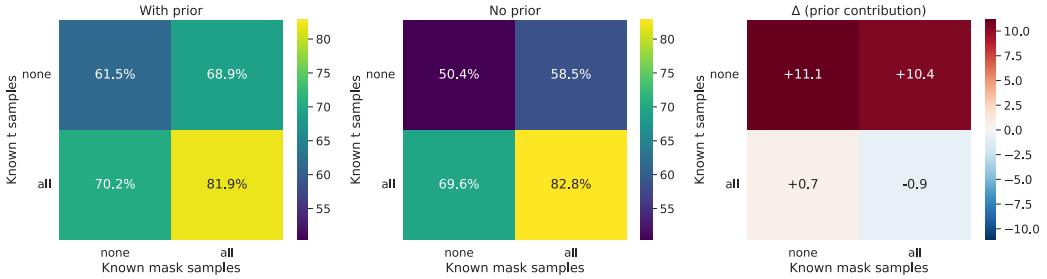
Figure 7: Effect of the KL timestep-mask prior under mixed diffusion at batch size 1, on all nine datasets. Categorical-heavy datasets first, then continuous-heavy.

G.3 Knowledge of Timesteps and Mask with and without the Prior

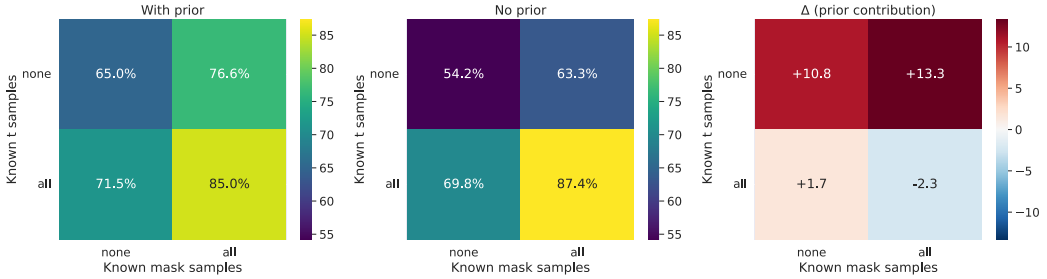
We study how much of the improvement from the KL prior survives when the attacker is given partial or full knowledge of the diffusion timestep t and the mask m. The experiment runs under mixed

diffusion on two categorical-heavy datasets, Adult and Credit, at batch size 1. The 2×2 heatmaps compare the only distinct settings, where each latent is either unknown (0%) or known (100%). Each row of Figs. 8 and 9 shows runs with the prior, runs without it, and their difference.

When t is unknown, the prior helps substantially because it is the only signal tying mask density to timestep; the recovered t' and m' drift when this coupling is removed. With both latents unknown, the prior raises accuracy from 50.4% to 61.5% on Adult and from 54.2% to 65.0% on Credit, while timestep MAE falls from 127.2 to 46.7 and from 126.9 to 39.6. The gain persists when only the mask is known, at 10.4 percentage points on Adult and 13.3 on Credit. Once t is fully revealed, the gap between prior and no-prior shrinks into noise (-2.3 to $+1.7$ percentage points): the prior has nothing left to add once the timestep is not a free variable. Remaining differences at high knowledge levels are within seed-to-seed variation.

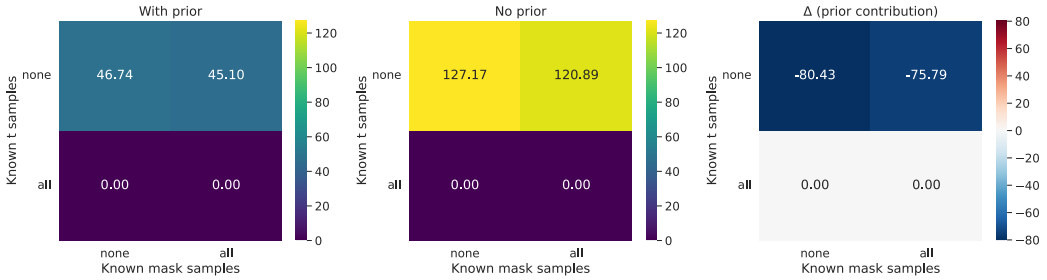


(a) Adult, cell accuracy.

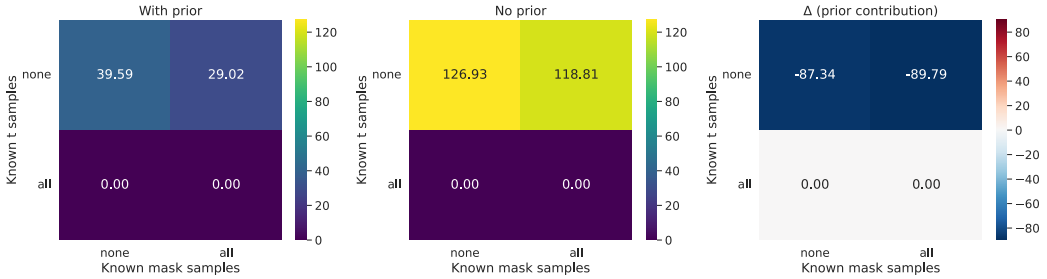


(b) Credit, cell accuracy.

Figure 8: Reconstruction accuracy (↑) under unknown or known t (rows) and mask m (columns) at batch size 1, mixed diffusion, on two categorical-heavy datasets. Each pair of heatmaps compares runs with the KL prior to runs without it.



(a) Adult, timestep MAE.



(b) Credit, timestep MAE.

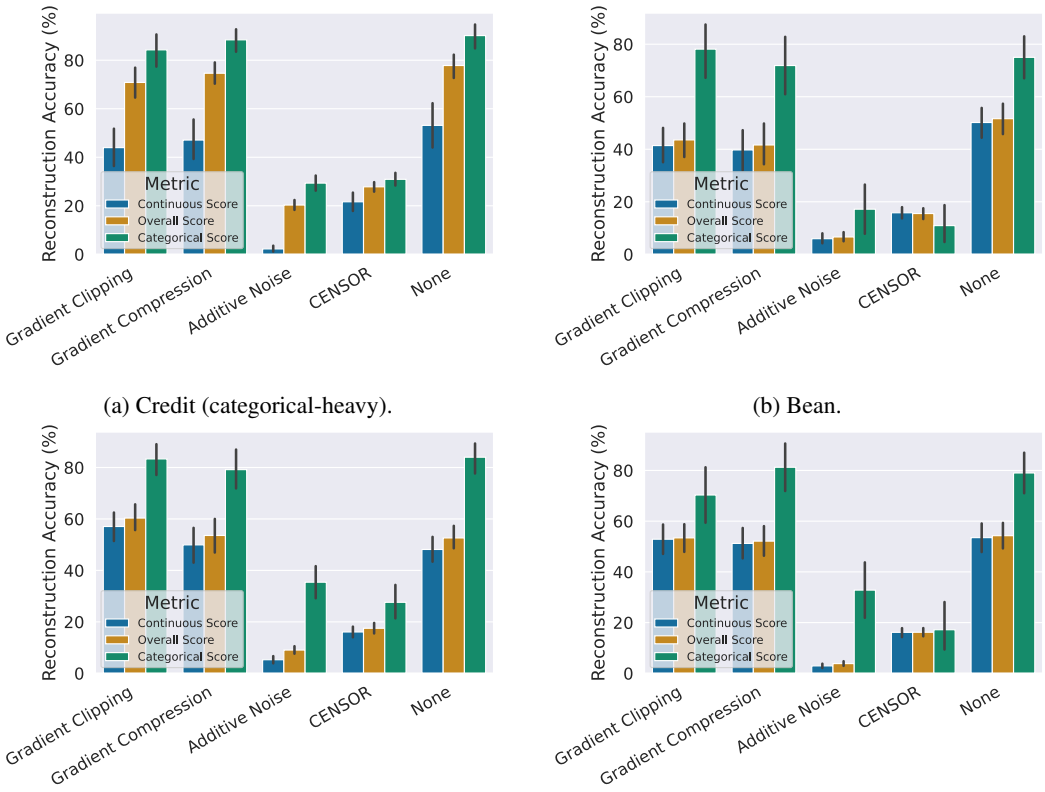
Figure 9: Timestep MAE (↓) under unknown or known t (rows) and mask m (columns) at batch size 1, mixed diffusion, on two categorical-heavy datasets. Each pair of heatmaps compares runs with the KL prior to runs without it.

G.4 Defense Robustness: Full Results

This section reports the defense robustness plots omitted from the main body, completing coverage of all nine datasets and DCR metrics. Figs. 10 and 11 cover the remaining reconstruction accuracy and DCR results under Gaussian diffusion. Figs. 12 and 13 cover the corresponding results under mixed diffusion. Each panel compares the four defenses with the undefended attack, and the averages below are taken across the nine datasets.

Clipping and compression barely affect the attack. Under Gaussian diffusion, they reach 55.3% and 55.5% total accuracy, compared with 57.1% without a defense; under mixed diffusion, they reach 54.1% and 54.3%, compared with 55.5%. Mean DCR changes by at most 0.01. Additive noise is the strongest defense, reducing total accuracy to 13.6% under Gaussian diffusion and 18.4% under mixed diffusion. It affects both feature types: under Gaussian diffusion, categorical accuracy falls from 82.5% to 33.4% and continuous accuracy from 50.2% to 8.2%. Continuous recovery therefore drops more in relative terms, although categorical accuracy also falls by more than half. Categorical accuracy stays higher on Magic and Abalone (73% and 56%), whose single categorical column has few categories.

CENSOR reduces accuracy to 20.9% under Gaussian diffusion, but its effect under mixed diffusion depends strongly on the schema. On the four datasets with at least three categorical columns (Adult, Credit, Default, and Shoppers), total accuracy stays between 59% and 67%, with no consistent drop in either subscore compared with the undefended attack. On the five datasets with a single categorical column, accuracy falls to between 7% and 23%. DCR shows the same pattern. Under Gaussian diffusion, noise and CENSOR raise DCR on Adult from 0.50 to 0.88 and on Credit from 0.40 to about 0.80, whereas CENSOR under mixed diffusion leaves it close to the undefended values (0.55 versus 0.60 on Adult and 0.51 versus 0.52 on Credit). On continuous-heavy datasets, DCR is high even without a defense, so it separates the defenses less clearly than cell accuracy.



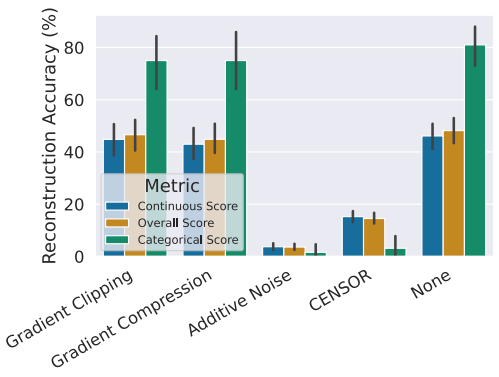
(a) Credit (categorical-heavy).

(b) Bean.

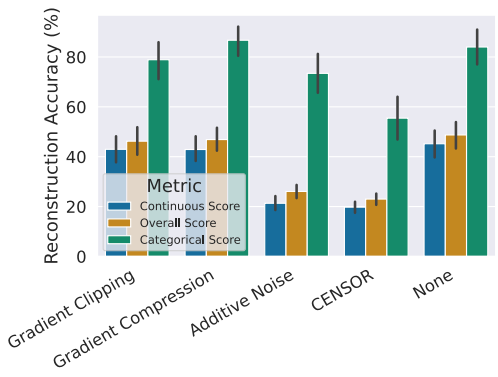
(c) Default.

(d) Gesture.

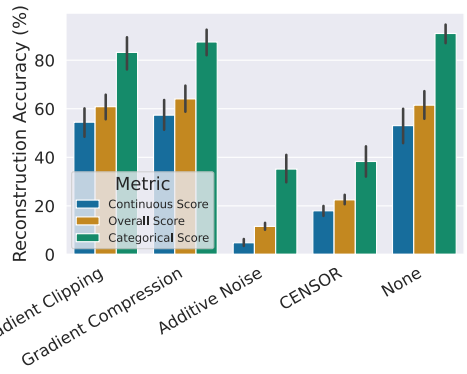
Figure 10: Reconstruction accuracy of GOLIATH against four defenses at batch size 1 under Gaussian diffusion, remaining datasets. The remaining categorical-heavy dataset is shown first, then continuous-heavy datasets.



(e) Letter.

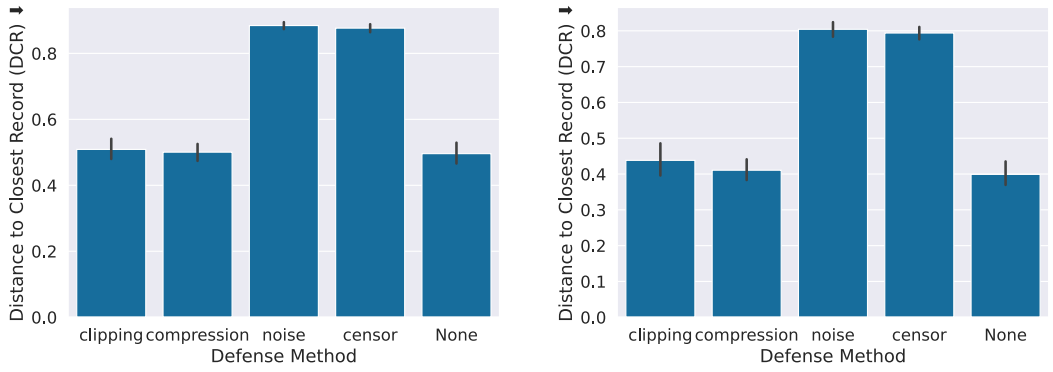


(f) Magic.



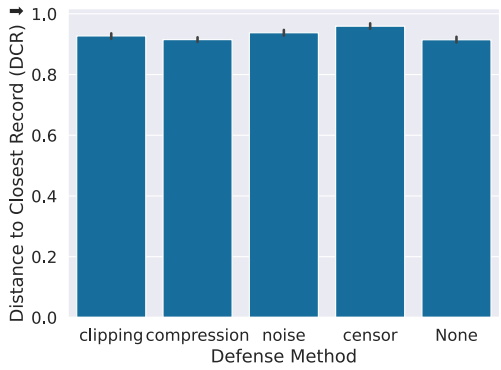
(g) Shoppers.

Figure 10: Reconstruction accuracy under Gaussian diffusion (continued).

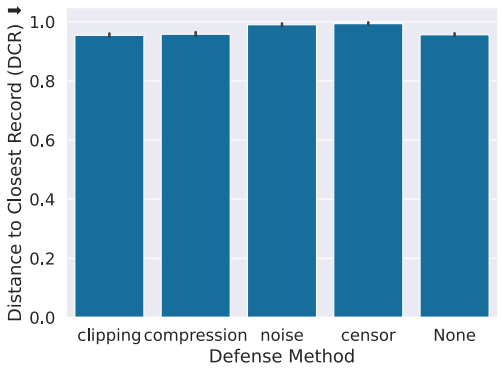


(a) Adult (categorical-heavy).

(b) Credit (categorical-heavy).

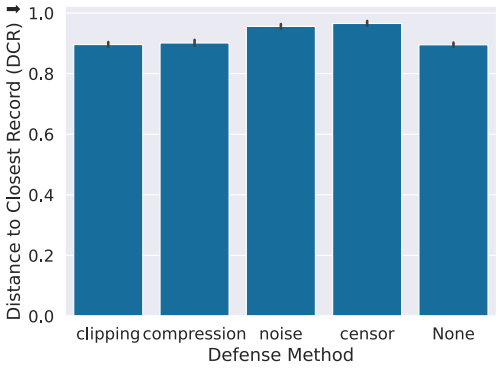


(c) Abalone.

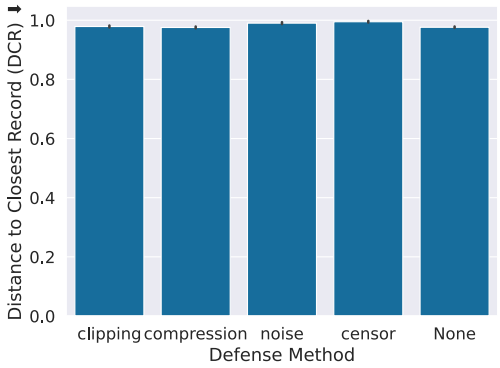


(d) Bean.

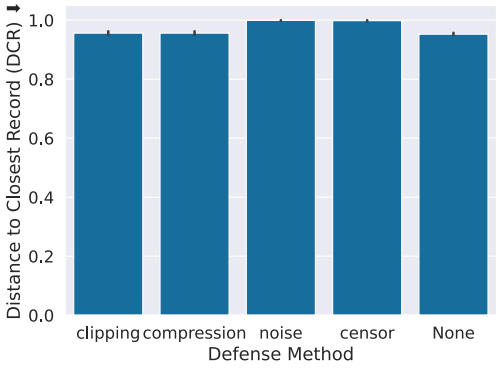
Figure 11: DCR of GOLIATH against four defenses at batch size 1 under Gaussian diffusion, all nine datasets. Categorical-heavy datasets first, then continuous-heavy.



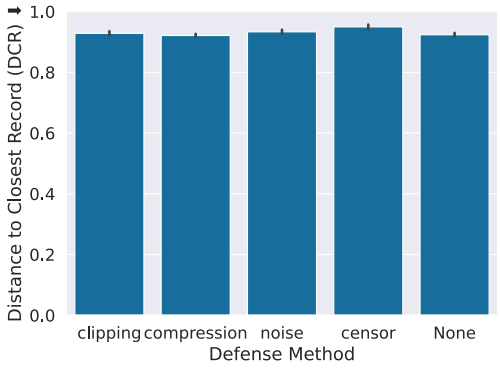
(e) Default.



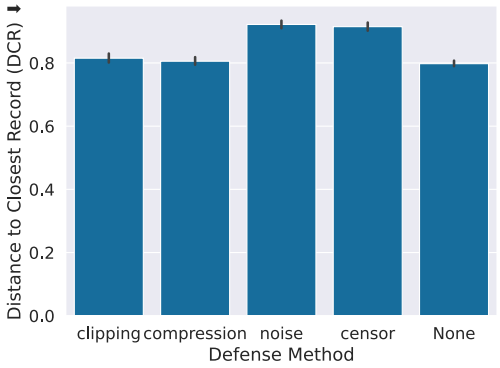
(f) Gesture.



(g) Letter.

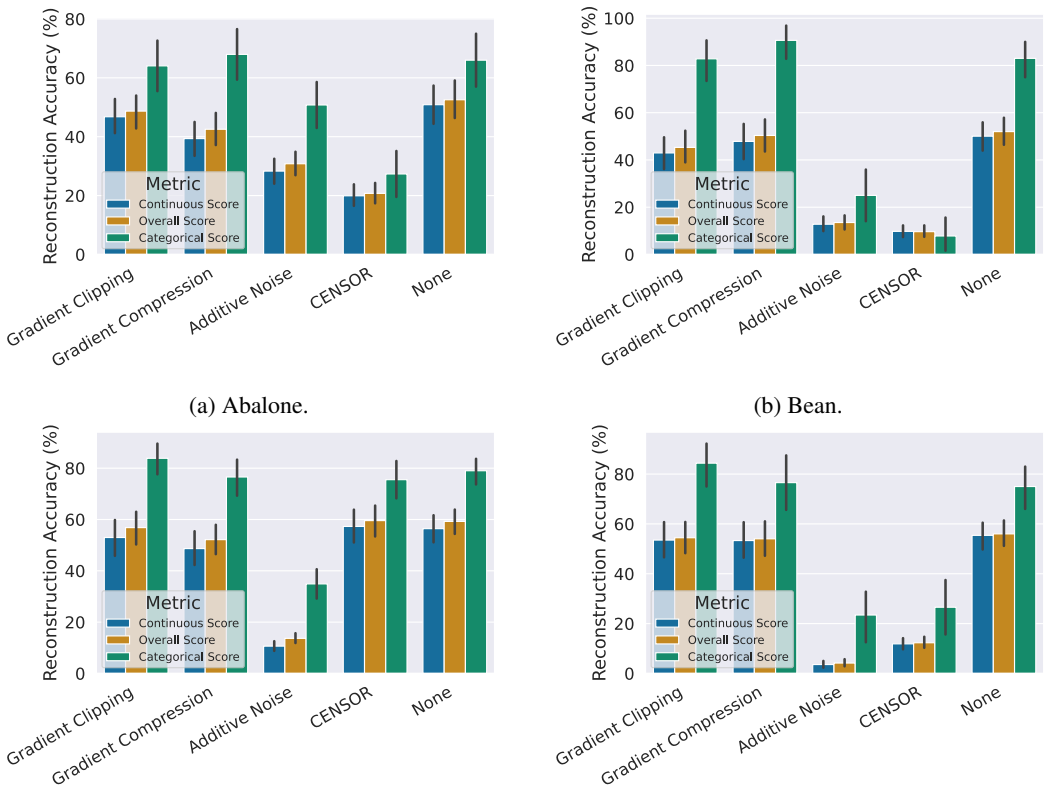


(h) Magic.



(i) Shoppers.

Figure 11: DCR under Gaussian diffusion (continued).



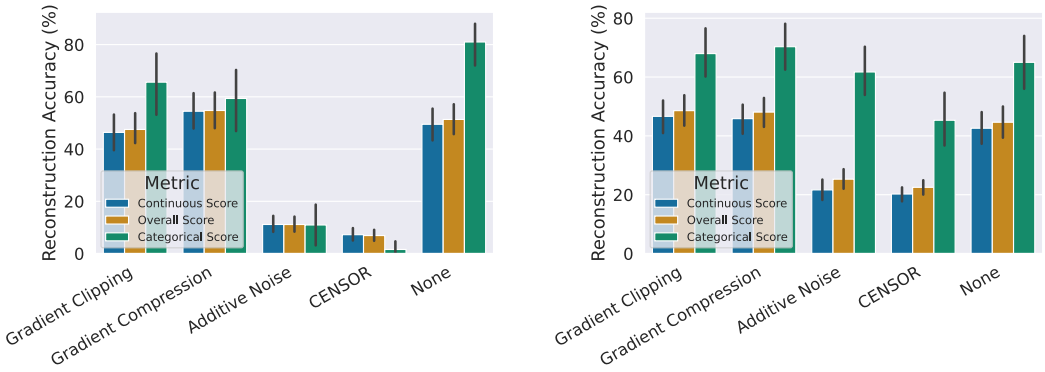
(a) Abalone.

(b) Bean.

(c) Default.

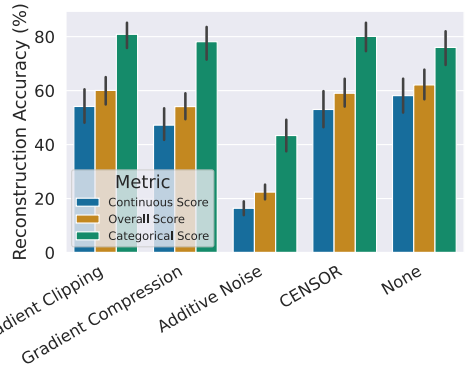
(d) Gesture.

Figure 12: Reconstruction accuracy of GOLIATH against four defenses at batch size 1 under mixed diffusion, remaining datasets.



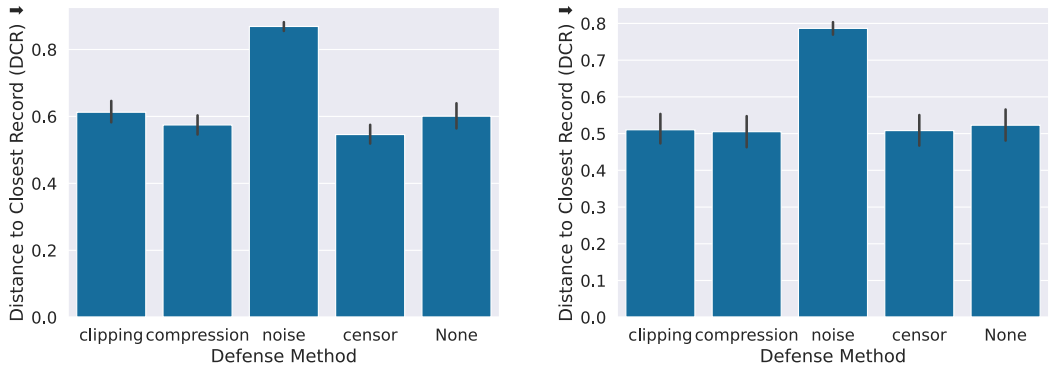
(e) Letter.

(f) Magic.



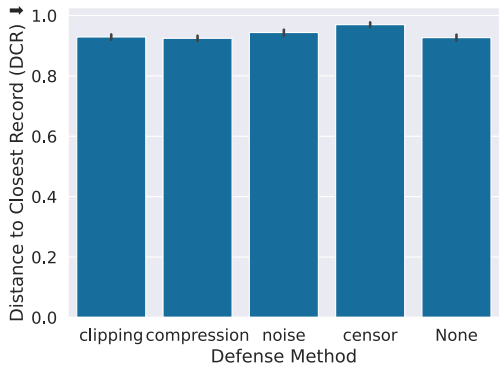
(g) Shoppers.

Figure 12: Reconstruction accuracy under mixed diffusion (continued).

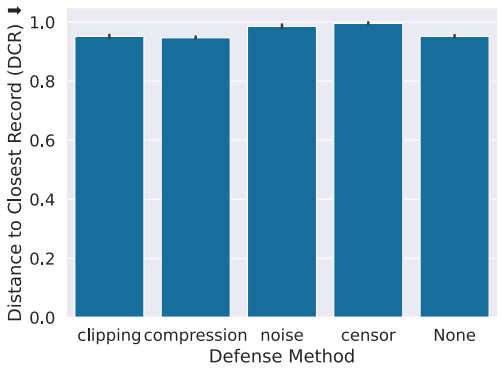


(a) Adult (categorical-heavy).

(b) Credit (categorical-heavy).

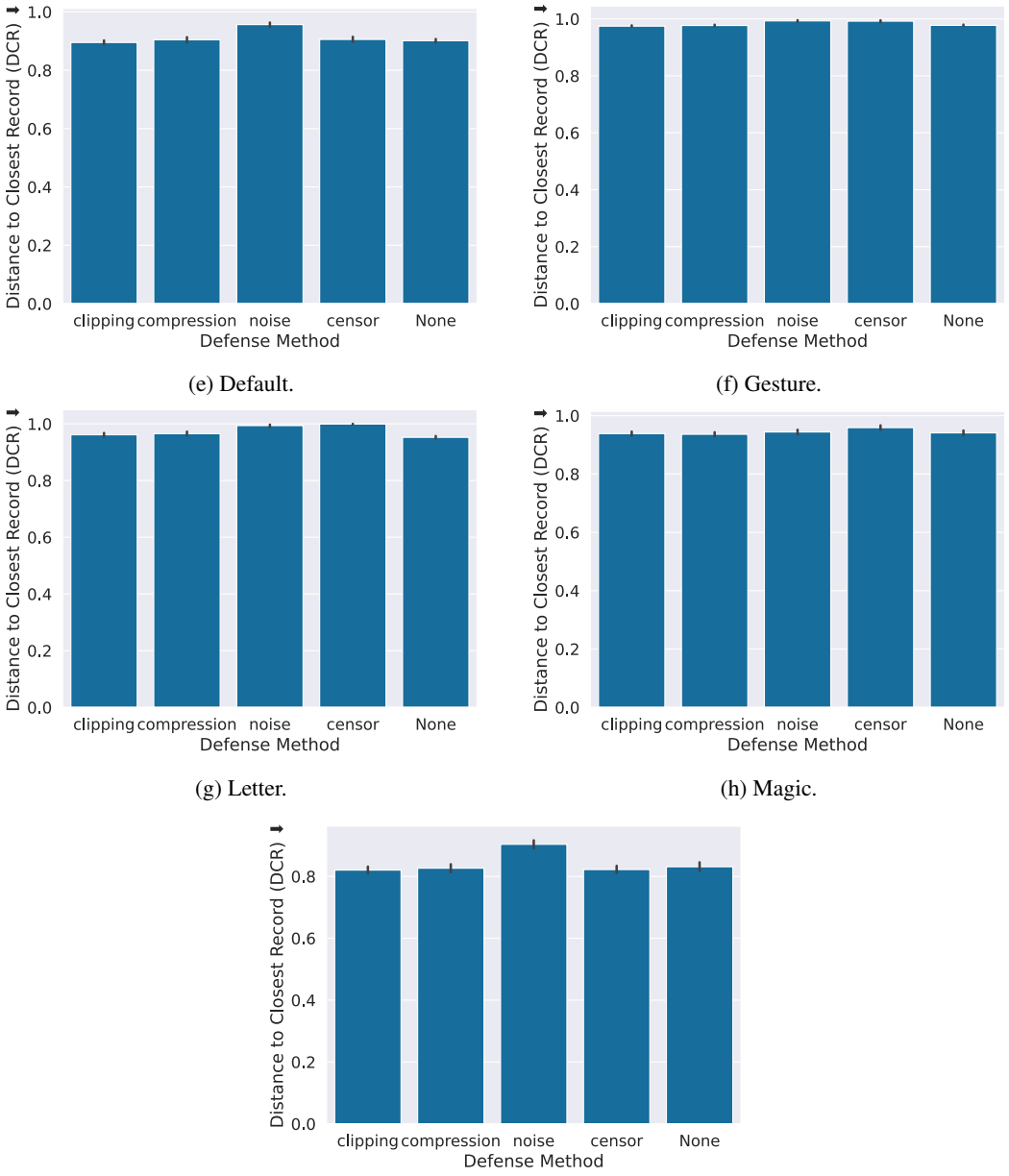


(c) Abalone.



(d) Bean.

Figure 13: DCR of GOLIATH against four defenses at batch size 1 under mixed diffusion, all nine datasets. Categorical-heavy datasets first, then continuous-heavy.



(e) Default.

(f) Gesture.

(g) Letter.

(h) Magic.

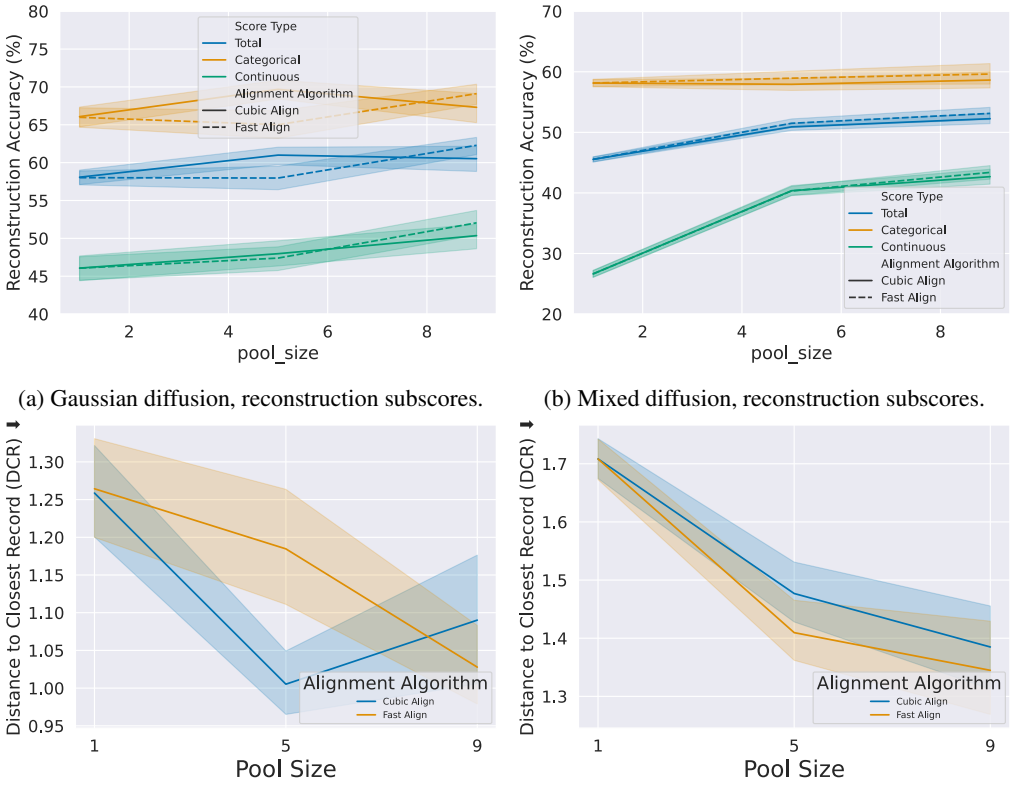
(i) Shoppers.

Figure 13: DCR under mixed diffusion (continued).

G.5 Pool-Size Ablation: Adult and Default Datasets

TabLeak [48] introduces pooled median ensembling to escape local minima of the gradient-matching objective. We run the pool-size ablation as an additional experiment to test whether this strategy remains viable in diffusion settings, where each reconstruction must also recover diffusion latents. Figs. 14 and 15 evaluate pool sizes 1, 5, and 9 on Adult and Default under Gaussian and mixed diffusion. In both diffusion types, categorical features are already well-recovered at pool size one, but their subscore is less stable as pool size changes. The continuous subscore, starting lower, is the main beneficiary of larger pool sizes and accounts for most of the visible gain in the total score. For categorical features, each pool member can settle on a different valid one-hot mode when several categories induce similar gradient matches. Median pooling is therefore less natural for categorical columns than for continuous columns, and larger pools can introduce small mode conflicts after row alignment. This may explain why gains plateau and the categorical subscore can fluctuate as pool size increases.

The masked variant does not show this degradation: the KL prior couples the mask to the recovered timestep, and the discrete mask latent gives pool members more structurally distinct reconstruction paths. The aggregate is therefore more robust and consistently improves over the no-pooling baseline, showing that pooled ensembling remains useful for diffusion inversion when the pool members are sufficiently diverse. In both cases, the gap between heuristic alignment and the cubic baseline is within the run-to-run variance (score delta ≤ 0.03, within one standard deviation), confirming that our global greedy heuristic reduces alignment cost without sacrificing reconstruction quality.



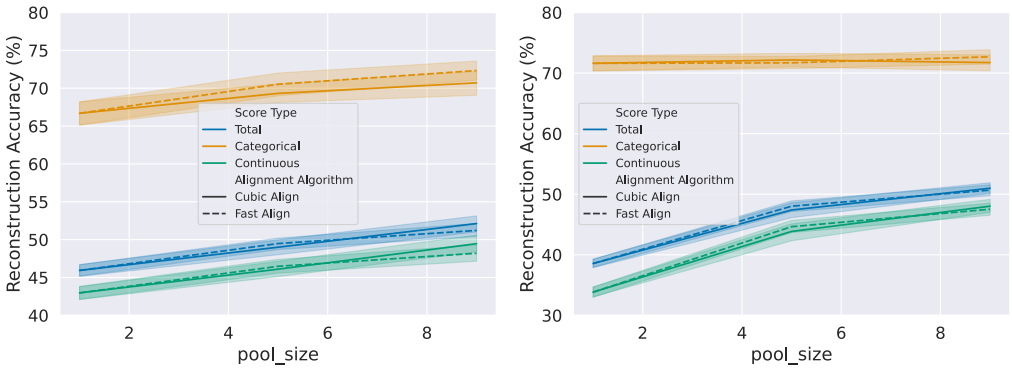
(a) Gaussian diffusion, reconstruction subscores.

(b) Mixed diffusion, reconstruction subscores.

(c) Gaussian diffusion, DCR.

(d) Mixed diffusion, DCR.

Figure 14: Impact of pool size on reconstruction subscores (total, categorical, continuous) and DCR, Adult dataset.



(a) Gaussian diffusion, reconstruction subscores.

(b) Mixed diffusion, reconstruction subscores.



(c) Gaussian diffusion, DCR.

(d) Mixed diffusion, DCR.

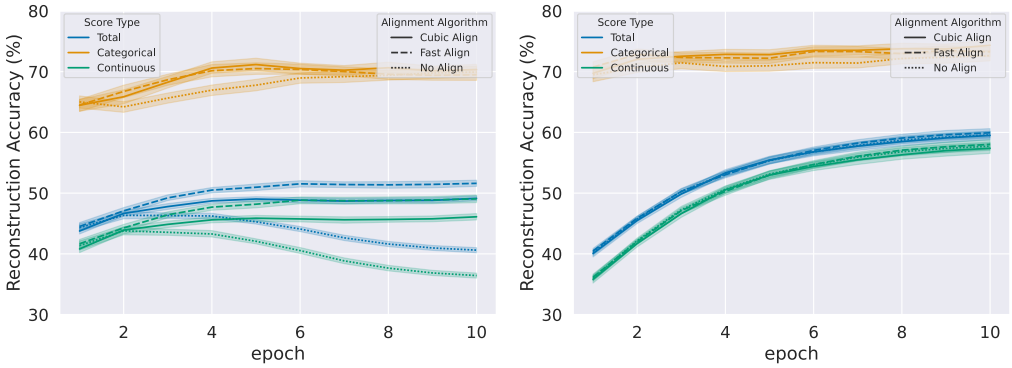
Figure 15: Impact of pool size on reconstruction subscores (total, categorical, continuous) and DCR, Default dataset.

G.6 Cross-Epoch Ensembling: Full Results

These figures extend Fig. 4 to the remaining datasets and to DCR, comparing Fast Align, Hungarian matching, and no alignment at batch size 32 over ten epochs. Figs. 16 and 17 report accuracy subscores, and Figs. 18 and 19 report DCR.

With either alignment method, total accuracy improves from the first to the tenth epoch in all eight dataset-diffusion settings. Fast Align gains as little as 0.6 percentage points on Adult under Gaussian diffusion and more than 19 on Default under mixed diffusion, where accuracy rises from 40.5% to 59.9%. The two methods end within 2.5 percentage points of each other in every setting, and neither is consistently ahead.

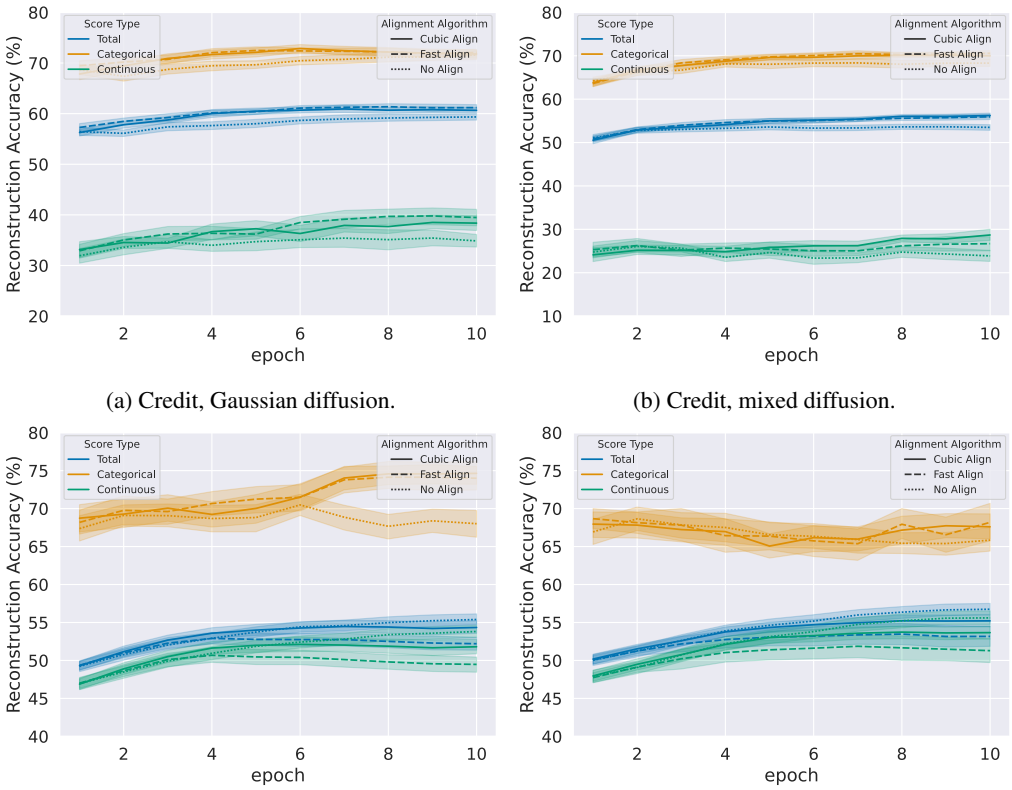
Whether alignment helps depends on the dataset and the metric. No alignment gives the lowest final accuracy on Adult and Credit under both diffusion families and on Default under Gaussian diffusion. On the latter, accuracy even declines from 44.3% to 40.6%, while Fast Align reaches 51.6%. However, no alignment nearly ties Fast Align on Default under mixed diffusion (59.6% versus 59.9%) and is the most accurate on Abalone, reaching 55.4% and 56.7% under Gaussian and mixed diffusion, compared with 52.2% and 53.2% for Fast Align. DCR tells a different story on Abalone, where no alignment ends at 1.38 and 1.25, compared with 0.91 and 1.11 for Fast Align. More generally, Fast Align reduces DCR in all eight settings, and no alignment always ends with a higher DCR, by as little as 0.01 on Default under mixed diffusion and as much as 0.52 on Default under Gaussian diffusion. Higher cell accuracy without alignment therefore does not mean that reconstructed records are closer to the true rows.



(a) Default, Gaussian diffusion.

(b) Default, mixed diffusion.

Figure 16: Reconstruction subscores (total, categorical, continuous) over training epochs for heuristic alignment, Hungarian matching, and no-alignment baseline on Default, at batch size 32.



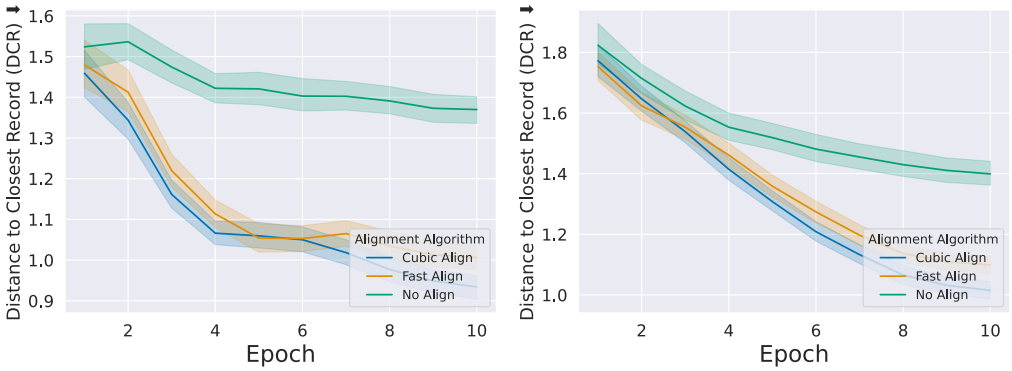
(a) Credit, Gaussian diffusion.

(b) Credit, mixed diffusion.

(c) Abalone, Gaussian diffusion.

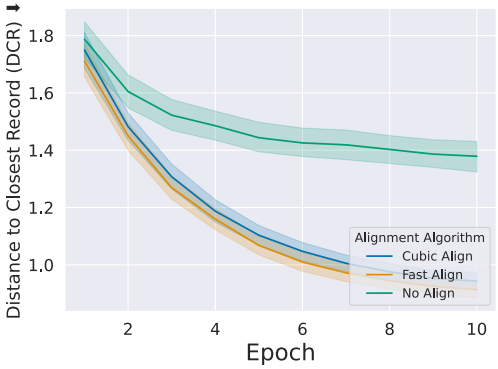
(d) Abalone, mixed diffusion.

Figure 17: Reconstruction subscores over training epochs for heuristic alignment, Hungarian matching, and no-alignment baseline (Credit and Abalone), at batch size 32.

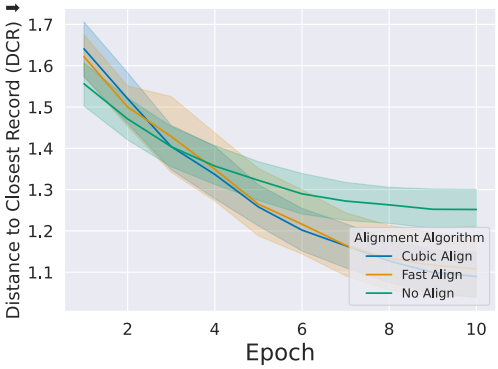


(a) Adult, Gaussian diffusion.

(b) Adult, mixed diffusion.

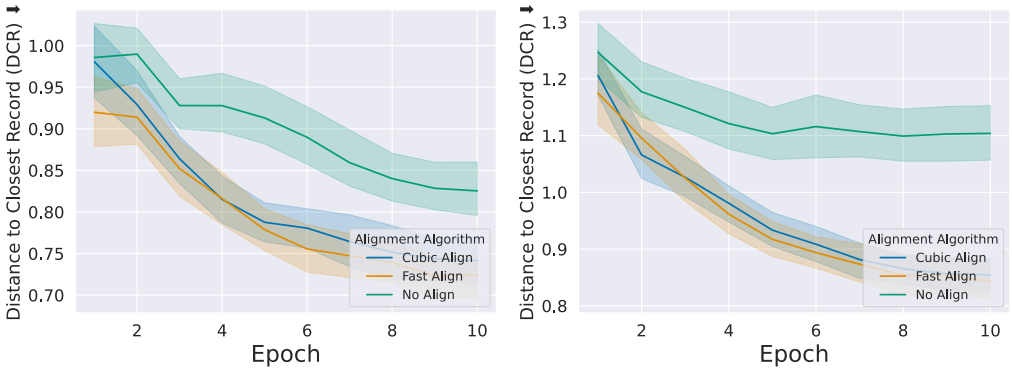


(c) Abalone, Gaussian diffusion.



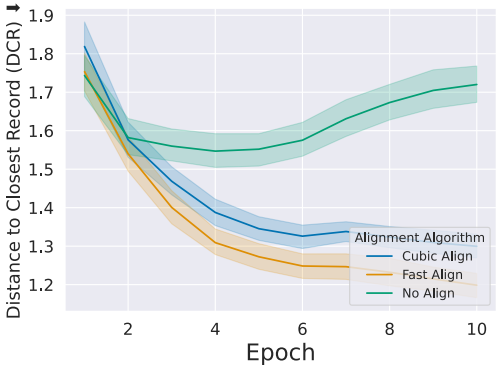
(d) Abalone, mixed diffusion.

Figure 18: DCR over training epochs for heuristic alignment, Hungarian matching, and no-alignment baseline (Adult and Abalone), at batch size 32.

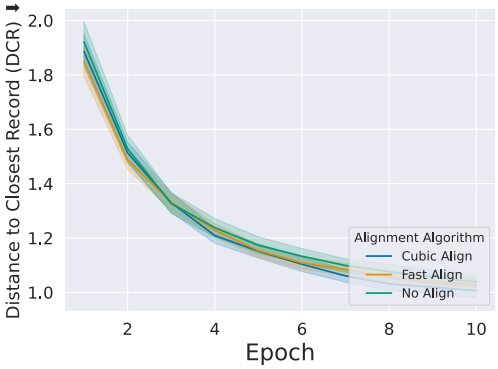


(a) Credit, Gaussian diffusion.

(b) Credit, mixed diffusion.



(c) Default, Gaussian diffusion.



(d) Default, mixed diffusion.

Figure 19: DCR over training epochs for heuristic alignment, Hungarian matching, and no-alignment baseline (Credit and Default), at batch size 32.

G.7 Cross-Epoch Identifiability Proofs

We restate here the asymptotic identifiability assumption and the feasible-set contraction lemma that formalize the AGIC-style cross-epoch aggregation used in Sec. 3.5. The argument is a straightforward adaptation of the AGIC [50] analysis to the permutation-equivalence class of a fixed batch.

Define the cumulative gradient-matching loss over J observed gradients as

$$\mathcal{L}_J\Big(X', \{z'^{(j)}\}_{j=1}^J\Big) := \sum_{j=1}^{J} \mathcal{D}\Big(\nabla_\theta^{(j)}, \nabla_\theta'^{(j)}(X', z'^{(j)})\Big), \tag{10}$$

where $z'^{(j)}$ denotes the epoch-specific dummy diffusion latents, i.e., $(t'^{(j)}, \epsilon'^{(j)})$ for Gaussian diffusion and $(t'^{(j)}, \epsilon'^{(j)}, m'^{(j)})$ for mixed diffusion. We define the δ-tolerant feasible set as its projection onto the data batch,

$$\mathcal{C}_J(\delta) := \Big\{X' : \exists \{z'^{(j)}\}_{j=1}^J \text{ such that } \mathcal{L}_J\Big(X', \{z'^{(j)}\}_{j=1}^J\Big) \le \delta\Big\}. \tag{11}$$

Assumption 1 (Asymptotic identifiability). *Let $\mathcal{P}$ denote the set of all permutations of batch elements. Assume that the observed gradients $\{\nabla_\theta^{(j)}\}_{j\ge1}$ are generated by repeated training on the same private batch X, with epoch-specific diffusion latents $\{z^{(j)}\}_{j\ge1}$. Define the asymptotic feasible set*

$$\mathcal{C}_\infty(0) := \Big\{X' : \exists \{z'^{(j)}\}_{j\ge1} \textit{ such that } \mathcal{L}_\infty\Big(X', \{z'^{(j)}\}_{j\ge1}\Big) = 0\Big\}. \tag{12}$$

Then in the limit of infinite epochs, the feasible set coincides with the permutation-equivalence class of the true batch,

$$\mathcal{C}_\infty(0) = \{\pi(X) : \pi \in \mathcal{P}\}. \tag{13}$$

Lemma 1 (Feasible-set contraction). *Let $\mathcal{C}_\infty(\delta)$ be defined as $\mathcal{C}_\infty(0)$ with $\mathcal{L}_\infty \le \delta$ in place of $\mathcal{L}_\infty = 0$. Under Assumption 1, the feasible sets defined in Eq. 11 satisfy, for any $\delta \ge 0$ and $J \ge 1$,*

$$\mathcal{C}_\infty(0) \subseteq \mathcal{C}_\infty(\delta) \subseteq \mathcal{C}_J(\delta) \subseteq \mathcal{C}_{J-1}(\delta) \subseteq \cdots \subseteq \mathcal{C}_1(\delta). \tag{14}$$

Each additional epoch can only restrict the data-space feasible set; the inclusion need not be strict.

Proof. By construction of the cumulative loss and Assumption 1, the true batch X together with its realized epoch-specific latents $\{z^{(j)}\}_{j=1}^J$ satisfies $\mathcal{L}_J = 0$ for all J, so $X \in \mathcal{C}_j(\delta)$ for every j and every $\delta \ge 0$. Now consider a candidate $X' \in \mathcal{C}_J(\delta)$. By definition, there exists a latent sequence $\{z'^{(j)}\}_{j=1}^J$ such that $\mathcal{L}_J(X', \{z'^{(j)}\}_{j=1}^J) \le \delta$. Since each term $\mathcal{D}(\cdot,\cdot) \ge 0$, for any $j \le J$ the prefix sequence $\{z'^{(\ell)}\}_{\ell=1}^j$ satisfies $\mathcal{L}_j(X', \{z'^{(\ell)}\}_{\ell=1}^j) \le \mathcal{L}_J(X', \{z'^{(\ell)}\}_{\ell=1}^J) \le \delta$. Hence any solution in $\mathcal{C}_J(\delta)$ is in $\mathcal{C}_j(\delta)$ for all $j \le J$, proving the nested inclusion in Eq. 14. □

The nested inclusion follows from nonnegativity of the loss; Assumption 1 additionally posits uniqueness of an exact infinite-observation solution up to permutation. Neither statement guarantees convergence of finite-epoch minimizers or of the EMA update in Eq. 9. We use this argument as an identifiability intuition and evaluate finite-epoch recovery empirically in Sec. 4.5 and App. H.9. Large batches can weaken identifiability because different records contribute similar terms to a summed gradient; no comparable effect of feature correlation is established by our experiments.

H Additional Evaluation

This appendix extends the main evaluation with stronger baseline comparisons, finer leakage metrics, an additional dataset and backbone, and alternative update and defense settings. We report seed-level variability where repeated runs are available and distinguish it from the batch-level variability in Tabs. 1 and 2.

H.1 Adaptive TabLeak

Adaptive TabLeak uses the same cyclic optimization of dummy data, timesteps, noise, and, under mixed diffusion, masks as GOLIATH. It retains the cosine gradient distance of TabLeak, disables the

timestep-mask prior, and uses one epoch without cross-epoch ensembling. This separates the benefit of recovering the diffusion latents from the tabular-aware objective and ensembling. We evaluate all ten datasets, batch sizes 1, 8, 16, and 32, and both diffusion families, for 80 configurations. The main tables report the mean and sample standard deviation over attacked batches within one run: 60 batches at size 1 and 30 at larger sizes, except Credit at size 32, where the available run contains 22 batches. These subscripts quantify batch variation, not uncertainty across independent seeds.

With five-epoch ensembling, GOLIATH exceeds Adaptive TabLeak in all configurations, by 16.2 percentage points on average under Gaussian diffusion and 13.0 under mixed diffusion (14.6 overall). Without ensembling, the average gains are 9.7 and 4.3 points, respectively, with wins in $38/40$ Gaussian and $36/40$ mixed configurations. The six exceptions are Default and Letter at batch size 1 under Gaussian diffusion and Census at all four batch sizes under mixed diffusion; ensembling restores the advantage in all six cases, although the Census margins remain small ($+2.3$ to $+8.1$ points). Tab. 8 summarizes the gains by dataset. The five-epoch comparison measures the complete attack, including its access to additional gradients; the one-epoch comparison controls for this difference.

Table 8: Five-epoch GOLIATH versus Adaptive TabLeak, averaging four batch sizes and two diffusion families per dataset. Gaps are in percentage points; the smallest gap is over the eight configurations.

Dataset	GOLIATH	Adaptive TabLeak	Mean gap	Smallest gap
gesture	0.607	0.400	+20.7	+16.2
bean	0.547	0.380	+16.7	+12.0
default	0.593	0.432	+16.1	+3.7
census	0.640	0.480	+16.0	+2.3
shoppers	0.587	0.436	+15.1	+4.3
adult	0.621	0.486	+13.5	+4.9
abalone	0.509	0.388	+12.1	+5.4
credit	0.635	0.515	+12.0	+4.3
magic	0.517	0.397	+12.0	+7.9
letter	0.486	0.366	+11.9	+1.8

H.2 Row-Level and Per-Feature Leakage

We evaluate Adult, Abalone, Gesture, Letter, and Magic with batch sizes 1, 8, and 32, using client subsets of 32, 128, and 256 rows, respectively. Each setting uses single-epoch inversion and three independent runs. We first average attacked batches within each run, then datasets equally where a pooled score is reported, and finally report the mean and population standard deviation across the three runs. Only complete runs enter this aggregation.

Row-level recovery. A cell is correct when its categorical value matches exactly or its numerical value lies within the tolerance used by the cell-accuracy metric (App. D). Whole-row recovery requires every cell to be correct; the 80% and 90% thresholds require the corresponding fraction of correct cells within a row. These are tolerance-based reconstruction measures, not literal equality of numerical records. Tab. 9 shows that full-row recovery falls sharply with batch size, reaching 0.2% at size 32 under both diffusion families, even though many individual cells remain recoverable.

Within-batch nearest-neighbour agreement. For each reconstructed row, we find the most similar original row within the attacked batch using the same cell-similarity score. We count agreement with the row assigned by the evaluation alignment. At batch size 32, this agreement is 50.0% under Gaussian and 47.4% under mixed diffusion. The diagnostic describes how distinctly reconstructions match rows within that batch; it uses ground-truth evaluation alignment and does not measure identification against an external population. At batch size 1, it is 100% by construction.

Sensitive attributes. Tab. 3 reports Adult under Gaussian diffusion at batch size 32. Marital, Country, and Relation denote marital-status, native-country, and relationship. Their accuracies are 69.7%, 67.4%, and 67.3%, respectively; race and sex exceed 82%, while age reaches 38.5%. Thus, the overall 59.8% cell accuracy conceals substantial variation across attributes. The comparison is

Table 9: Row-level leakage across five datasets. Percentages, mean ± population standard deviation over three seeds after averaging datasets equally. All cells uses exact categorical matches and numerical tolerance; NN agreement is restricted to the attacked batch and is trivial at batch size 1.

Batch	Diffusion	All cells	$\geq 90\%$	$\geq 80\%$	NN agreement
1	Gaussian	8.3 ± 1.1	14.6 ± 1.6	22.5 ± 2.2	100.0 ± 0.0
8	Gaussian	1.7 ± 0.6	3.9 ± 1.0	9.8 ± 0.8	64.0 ± 0.7
32	Gaussian	0.2 ± 0.1	0.4 ± 0.1	3.2 ± 0.4	50.0 ± 0.5
1	Mixed	12.9 ± 0.3	18.8 ± 2.2	27.9 ± 0.8	100.0 ± 0.0
8	Mixed	1.1 ± 0.4	2.0 ± 0.6	3.8 ± 0.8	57.2 ± 0.6
32	Mixed	0.2 ± 0.1	0.5 ± 0.1	1.5 ± 0.2	47.4 ± 0.7

descriptive: category imbalance can contribute to high accuracy, and we do not estimate improvement over a majority-category predictor here.

Categorical cardinality. Tab. 10 separates columns with at most four, five to nine, and at least ten categories. On Adult at batch size 1, accuracy decreases from 93.8% in the lowest bin to 70.5% in the highest, while numerical accuracy is 61.5%. The same ordering holds on Census-Income (KDD). Tab. 11 shows that high-cardinality recovery decreases from 71.2% to 46.1% as the batch grows from 1 to 32. The low-cardinality bin is non-monotonic and includes only datasets with populated bins, so the bin averages should not be interpreted as a controlled effect of cardinality alone.

Table 10: Per-cell accuracy (%) by categorical cardinality at batch size 1 under Gaussian diffusion. Mean ± population standard deviation over three runs. Empty bins are marked by a dash.

Dataset	≤ 4 categories	5–9 categories	≥ 10 categories	Numerical
adult	93.8 ± 4.4	77.6 ± 5.8	70.5 ± 5.7	61.5 ± 6.0
Census	84.7 ± 6.3	70.1 ± 5.2	62.9 ± 2.2	50.5 ± 1.2
magic	80.2 ± 1.5	–	–	44.2 ± 5.1
gesture	–	76.0 ± 10.3	–	49.7 ± 3.5
letter	–	–	71.9 ± 4.4	45.7 ± 2.3
abalone	72.9 ± 1.5	–	–	38.2 ± 3.4

Table 11: Categorical cell accuracy (%) versus batch size under Gaussian diffusion, averaging the five leakage datasets with nonempty bins and then three seeds.

Batch	≤ 4 categories	5–9 categories	≥ 10 categories
1	82.3	76.8	71.2
8	71.3	67.9	61.7
32	76.0	61.9	46.1

H.3 Many-Category Tables

The main accuracy tables also include Census-Income (KDD) [1] (CC BY 4.0), which contains demographic and employment records from the US Census Bureau. Our processed schema has 34 categorical and seven numerical columns, including ten categorical columns with at least ten categories and up to 52 levels, compared with nine categorical columns in Adult. We concatenate the publisher's data and test partitions before applying our standard train/test split, drop the instance-weight column and rows with missing entries, and include the income category as a modeled column. Tabs. 1 and 2 report its one- and five-epoch accuracy at batch sizes 1, 8, 16, and 32. Thus, many categorical columns do not by themselves prevent inversion, although recovery varies with the diffusion family and category cardinality. The reconstruction comparison uses one run per configuration, with 50 attacked batches per epoch at sizes 1 and 8 and 100 at sizes 16 and 32. The Census baselines use one run per configuration, with 50 attacked batches for TabLeak, GIDM, and random-latent baselines and 30–60 for Adaptive TabLeak. The Census cardinality row in Tab. 10

combines the first epoch of the 50-row run with two additional 32-row runs, averaging their per-run means equally.

H.4 FT-Transformer Backbone

We adapt the FT-Transformer [21] to the diffusion denoising objective. Each scalar input feature, including one-hot categorical coordinates, receives a learned affine token embedding of width 256. A sinusoidal timestep embedding followed by a two-layer MLP is added to every feature token. The encoder has four layers, eight attention heads, and feed-forward width 512; a shared linear head predicts one scalar per output token. The mixed-diffusion input additionally contains the mask coordinates. Feature-tokenizer weights and biases are initialized independently so that zero-valued features have distinct tokens. For the split gradient distance, numerical and categorical contributions are identified through the corresponding tokenizer parameters.

We report batch size 32, five epochs, and 128 rows per client subset, averaged over two completed runs per dataset and diffusion family. Inversion uses $1,000$ iterations per gradient at learning rate 0.06, selected after screening on Adult; the MLP comparison retains its $2,000$-iteration budget. Consequently, the comparison establishes transfer under these settings, rather than isolating architecture under identical optimization budgets. Tabs. 12 and 13 report feature-type and per-dataset results. The average MLP advantage is 7.4 percentage points under Gaussian diffusion and 9.0 under mixed diffusion; gaps in both tables are in percentage points. The transformer leads on Abalone under both families, but trails on the other eight datasets.

Table 12: Backbone comparison averaged equally over nine datasets at batch size 32 after five epochs. FT denotes FT-Transformer; cat. and num. give categorical and numerical cell accuracies.

Diffusion	FT	MLP	Gap	FT cat.	MLP cat.	FT num.	MLP num.
Gaussian	0.463	0.537	7.4	0.530	0.621	0.430	0.485
Mixed	0.424	0.514	9.0	0.461	0.618	0.409	0.445

Table 13: Cell accuracy by backbone and dataset at batch size 32, after five epochs. FT averages two runs; MLP uses the main-results run. G and M denote Gaussian and mixed diffusion.

Dataset	FT (G)	MLP (G)	Gap (G)	FT (M)	MLP (M)
abalone	0.560	0.513	−4.8	0.548	0.524
magic	0.512	0.530	1.8	0.493	0.514
letter	0.385	0.450	6.5	0.332	0.416
shoppers	0.487	0.553	6.5	0.434	0.523
credit	0.520	0.603	8.3	0.427	0.555
gesture	0.402	0.497	9.5	0.406	0.554
bean	0.414	0.528	11.4	0.410	0.520
default	0.421	0.536	11.5	0.391	0.515
adult	0.463	0.623	16.0	0.377	0.504

H.5 Multiple Local Updates

We extend inversion to the model update released by one client after $K \in \{1, 2, 4, 8\}$ local SGD steps, following the differentiable local-training approach of Dimitrov et al. [12]. Each step uses the same full client subset of 32 or 64 rows with fresh diffusion latents. The attacker knows the initial model, the local learning rate (10^{-4}), and K, and observes only the final parameter difference. We optimize the dummy rows and a separate set of dummy diffusion latents for each local step, differentiating through the unrolled SGD updates to match that difference. This is a per-client full-batch local-update experiment, not aggregation across clients or a sweep over unknown minibatch schedules.

Tab. 14 reports three-seed means and sample standard deviations on Adult and Abalone. Gaussian reconstruction is broadly stable across K, with no monotonic decline. Under mixed diffusion, Adult declines from 43.3% to 39.9% for 32 rows and from 47.6% to 40.0% for 64 rows, whereas Abalone

remains stable. Several local steps therefore do not remove leakage in these settings, although the results do not establish robustness to arbitrary FedAvg configurations.

Table 14: Cell accuracy for a single client update after K full-batch local SGD steps. Mean $\pm$ sample standard deviation over three seeds.

Dataset	Diffusion	Rows	$K = 1$	$K = 2$	$K = 4$	$K = 8$
abalone	Gaussian	32	0.480 ± 0.019	0.473 ± 0.030	0.520 ± 0.024	0.485 ± 0.046
abalone	Gaussian	64	0.506 ± 0.011	0.512 ± 0.023	0.513 ± 0.019	0.502 ± 0.035
adult	Gaussian	32	0.493 ± 0.021	0.494 ± 0.038	0.517 ± 0.014	0.535 ± 0.006
adult	Gaussian	64	0.485 ± 0.015	0.503 ± 0.003	0.515 ± 0.022	0.512 ± 0.003
abalone	Mixed	32	0.491 ± 0.011	0.483 ± 0.027	0.470 ± 0.032	0.502 ± 0.028
abalone	Mixed	64	0.506 ± 0.023	0.507 ± 0.023	0.503 ± 0.011	0.523 ± 0.009
adult	Mixed	32	0.433 ± 0.004	0.426 ± 0.024	0.411 ± 0.018	0.399 ± 0.022
adult	Mixed	64	0.476 ± 0.030	0.443 ± 0.010	0.412 ± 0.012	0.400 ± 0.019

H.6 Minibatch Reshuffling

The single-gradient attack does not use batch order. Cross-epoch ensembling, however, can average unrelated rows when a batch position contains different records in successive epochs. We evaluate this effect on Adult and Abalone, with 320 rows, batch size 32, and five epochs under both diffusion families. We compare fixed batches, reshuffling with per-slot ensembling, and two global-pool variants that match each new batch against the reconstructions of the full client subset, using either greedy or Hungarian assignment.

Tab. 15 reports fifth-epoch accuracy minus first-epoch accuracy for each setting. On Adult under Gaussian diffusion, the gain falls from 6.2 to 3.0 percentage points with reshuffling; under mixed diffusion it changes from 7.5 to 7.8 points. All tested gains remain positive, but global-pool matching does not consistently restore the fixed-order gain. In particular, neither global-pool variant reaches the fixed-order result on Adult under Gaussian diffusion. These trajectories measure the net change across training and ensembling, not an isolated causal effect of ensembling.

Table 15: Change in cell accuracy from the first to the fifth observed epoch, in percentage points, at batch size 32. G and M denote Gaussian and mixed diffusion. One run per setting.

Ensembling setting	Adult (G)	Adult (M)	Abalone (G)	Abalone (M)
Fixed order	+6.2	+7.5	+1.8	+1.9
Reshuffled (per-slot)	+3.0	+7.8	+0.9	+1.2
Reshuffled (global pool)	+1.8	+7.3	+1.1	+2.9
Reshuffled (global, Hungarian)	+2.9	+7.1	+2.0	+2.9

H.7 Clipped Gaussian Releases and Defended Training

We complement the uncalibrated perturbations in Sec. 4.4 with per-example clipping and Gaussian noise under Gaussian diffusion. For gradients g_i, the released average is

$$\widetilde{g} = \frac{1}{B}\left(\sum_{i=1}^{B} g_i \min\left\{1, \frac{C}{\|g_i\|_2}\right\} + \xi\right), \qquad \xi \sim \mathcal{N}(0, z^2 C^2 I). \tag{15}$$

For a fixed clipping bound C, fixed normalization, and add/remove adjacency, we compute the privacy budget of one release with the analytic Gaussian mechanism [4], at $\delta = 10^{-5}$ without subsampling amplification. The bounds are $C = 0.3$ on Adult and $C = 0.4$ on Credit, chosen by rounding diagnostic median per-example norms (0.278 and 0.370). The accounting treats these bounds as fixed: it does not account for private-data calibration, previous clean training, or the full training trajectory.

Attack-side sweep. The attacker observes a defended gradient at batch size 1, while the client in this sweep continues to train on clean gradients. Tab. 16 therefore measures reconstruction from an individual perturbed release. At the evaluated budgets $\varepsilon \in \{0.5, 1, 2, 4, 10\}$, cell accuracy falls to 11.5–14.4% on Adult and 20.2–21.9% on Credit. The drop begins at much smaller noise multipliers, whose single-release privacy budgets remain large.

Table 16: Single-release privacy accounting and cell reconstruction accuracy at batch size 1 under Gaussian diffusion. Each configuration is one run. The zero-noise reference uses the same clipping pipeline.

ε at $\delta = 10^{-5}$	Adult	Credit
∞ ($z = 0$)	0.728	0.692
≈ 5426 ($z = 0.01$)	0.614	0.627
≈ 697 ($z = 0.03$)	0.321	0.525
$0.5, 1, 2, 4, 10$	0.115–0.144	0.202–0.219

Utility-side sweep. Separate runs train the client on its own clipped, noised gradients, without inversion. At batch size 1, training uses 64 rows for 20 epochs (1,280 updates); at size 256, it uses 1,024 rows for 300 epochs (1,200 updates). Tab. 17 gives final-epoch denoising loss at the actual noise multipliers tested, rather than pairing a pooled loss range with every privacy budget. At batch size 1, even $z = 0.003$ increases loss close to 1, the expected noise-prediction loss of a zero-output model. At size 256, losses remain much lower for $z \leq 0.07$, showing that the cost depends strongly on the training regime. These losses do not measure synthetic-data quality, and their training-level privacy budgets require composition over all updates; they are not the single-release budgets in Tab. 16. The utility sweep does not include every noise multiplier in the target-budget attack sweep. The results support a small-batch utility cost in our setup, not a general incompatibility between differential privacy and tabular diffusion.

Table 17: Final-epoch training loss with defended gradients. One run per configuration; all values use the calibrated clipping bounds.

	Batch 1		Batch 256	
z	Adult	Credit	Adult	Credit
0	0.503	0.344	0.169	0.209
0.003	0.956	0.937	0.147	0.203
0.01	1.014	0.980	0.162	0.204
0.03	1.007	0.986	0.184	0.226
0.07	0.997	1.000	0.239	0.217
0.69	1.010	0.994	0.958	0.930

H.8 Evaluation Alignment Diagnostic

To isolate the effect of row assignment on the reported score, we apply greedy and Hungarian matching to the same reconstructions, without rerunning inversion. We evaluate Adult and Abalone under Gaussian diffusion at batch sizes 8, 32, 128, and 256. Hungarian matching improves cell accuracy by 1.4–1.9 percentage points, although 40.6–61.0% of row assignments differ (Tab. 18). Thus, different assignments often have similar cell scores, and assignment disagreement alone overstates the effect on accuracy. This diagnostic concerns evaluation matching; the cross-epoch alignment comparison in Sec. 4.5 evaluates the effect on the attack itself.

Table 18: Evaluation alignment on the same reconstructions under Gaussian diffusion, averaged over Adult and Abalone. The gap is in percentage points; each configuration is one run.

Batch	Hungarian	Greedy	Gap	Rows reassigned
8	0.544	0.529	$+1.5$	40.6%
32	0.542	0.528	$+1.4$	46.9%
128	0.544	0.525	$+1.9$	56.6%
256	0.544	0.526	$+1.8$	61.0%

H.9 Accuracy Versus Observed Epochs

Tab. 19 reports Adult at batch size 32, averaging the evaluated alignment variants. Pooled accuracy increases from 49.1% after one observed epoch to 53.8% after ten. The improvement comes mainly from mixed diffusion (44.7% to 53.3%); Gaussian recovery fluctuates around 54%. These separate runs average alignment variants, so their absolute values and epoch gains differ from those in Tabs. 1 and 15. This supports the usefulness of additional observations in some settings, but neither demonstrates strict feasible-set contraction nor verifies the asymptotic identifiability assumption in App. G.7.

Table 19: Adult cell accuracy versus observed epochs, averaging the evaluated alignment variants at batch size 32. Each variant is one run; the pooled column averages the two diffusion families.

Observed epochs	Gaussian	Mixed	Pooled
1	0.536	0.447	0.491
2	0.540	0.461	0.501
4	0.551	0.487	0.519
6	0.533	0.509	0.521
10	0.544	0.533	0.538

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The introduction lists three contributions: cyclic tabular inversion, a tabular-aware objective for heterogeneous features, and fast row alignment. These claims are supported by Secs. 3.3, 3.4, and 3.5; the abstract's 89.1% per-cell accuracy is the best five-epoch Gaussian result reported in Tab. 1.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The Conclusion states the pre-aggregation gradient access assumption, the repeated-batch assumption, the effects of batch size and backbone, and the single-release scope of our privacy accounting. App. A expands on these points and further discusses the limited defense sweep and metric coverage.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The cross-epoch argument is stated in App. G.7, including the asymptotic identifiability assumption, the cumulative feasible-set definition, and the feasible-set contraction lemma.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The optimizer, attack iterations, learning rate, diffusion steps, model architecture, training recipes, noise schedule, and defense settings are reported in Sec. 4.1 and App. C. Pseudocode for Gaussian training, mixed training, and fast row alignment is provided in App. E, and the code link appears in the abstract.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The abstract links the public code repository (`https://github.com/gsegalini/goliath`). The datasets are public tabular datasets listed in App. B, and the implementation details needed to run the main experiments are described in Sec. 4.1 and App. C.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The optimizer, attack hyperparameters, diffusion schedule, model architecture, training recipes, defense settings, and dataset schemas are provided in Sec. 4.1 and Apps. B, C, and G.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: The main accuracy and DCR tables report means with batch-level standard deviations, and the line plots report 95% confidence intervals as stated in Sec. 4.1. The additional experiments distinguish seed-level standard deviations from single-run point estimates (App. H).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Sec. 4.1 reports the GPU type, software stack, average inversion time per gradient, and approximate GPU memory use.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This work studies a privacy attack on federated tabular diffusion training using public datasets. The paper frames the attack as a way to expose privacy risks in systems that otherwise avoid sharing raw data, and it evaluates several common defenses in Sec. 4.4.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: GOLIATH is an offensive tool that exposes privacy weaknesses in federated tabular diffusion training. App. A discusses the impact of exposing these strategies before deployment.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The work uses public tabular datasets and does not release pretrained models, scraped datasets, or generated data assets that require additional release safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper cites the public datasets in App. B, where Tab. 4 reports the license for each dataset (App. H.3 for Census). The main software libraries are cited in Sec. 4.1.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The paper releases the attack code under the MIT License, and the abstract links the public repository (`https://github.com/gsegalini/goliath`). The experimental setup and implementation details are documented in Sec. 4.1 and App. C.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: No human subjects were involved.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: No human subjects were involved.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs were not used in the core method development of this research, but only for writing and editing purposes.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.