Auditing Privacy Leakage in Tabular Foundation Model Embeddings

Xun Wang **Adam Dziedzic** **Michael Backes** **Franziska Boenisch**
CISPA Helmholtz Center for Information Security
{xun.wang, adam.dziedzic, backes, boenisch}@cispa.de

Abstract

Tabular Foundation Models (TabFMs) produce context-aware embeddings that are increasingly stored, shared, and reused for retrieval, clustering, and cross-institutional data exchange. These embeddings are often treated as privacy-preserving substitutes for raw tabular records, yet their attribute-level information content remains poorly understood. We present a systematic privacy audit of TabFM embeddings: given an embedding and varying amounts of side information, how accurately can sensitive attributes be recovered? We introduce *Cascade Probing*, a sequential probing method that measures recoverability while accounting for inter-attribute dependencies, significantly outperforming joint probing and optimization-based inversion baselines. Across four TabFMs (TabPFN, TabDPT, Mitra, TabICL) and ten datasets, we find that even without any side information, 65–95% of sensitive attributes can be recovered from embeddings alone. More critically, recoverability exhibits a *Privacy Cliff*: revealing a single known attribute sharply increases the recoverability of all remaining attributes. This phenomenon generalizes across model architectures and diverse domains, and is not eliminated by standard embedding transformations including noise injection, PCA, and knowledge distillation. These findings suggest that we need more formal privacy techniques for protecting sensitive information inside TabFM embeddings. Our code is available at `https://github.com/sprintml/PrivacyCliff`.

1 Introduction

Tabular Foundation Models (TabFMs) [16, 17, 24, 30] leverage in-context learning to achieve state-of-the-art performance on tabular prediction tasks without task-specific training. Beyond predictions, these models produce context-rich embeddings that are increasingly stored, shared, and reused as substitutes for raw tabular records, e.g., for retrieval, clustering, and cross-institutional data exchange [10, 28, 37, 41]. While these embeddings can be considered as abstractions of raw data, their attribute-level information content remains poorly understood. This raises a natural question: *how much sensitive information about the original record can be recovered from a TabFM embedding?*

We study this through *attribute inference*, which aims to recover the sensitive attribute values (e.g., income, diagnosis) of an individual record from the model's embedding output, given partial public knowledge. For classical tabular classifiers and regressors, prior work has shown that predictions alone can reveal hidden attributes of individual records [8, 9]. Unlike membership inference [32], which only asks whether a given data record was *present* during training, attribute inference directly measures what the representation *reveals about the record* it represents, making it well suited to assess the semantic privacy leakage in the setup of shared TabFM embeddings. In contrast to standard classifiers that initial attribute inference attacks have been designed for [8, 9], TabFMs differ in two ways. First, they expose high-dimensional *embeddings* rather than logits or class labels. Second, they natively support feature-level missing values, i.e., specific attributes can be masked with `NaN` while retaining others, and the resulting embedding is a well-defined model output. This enables

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

differential queries that isolate the embedding signal attributable to masked attributes. Prior attribute inference methods, designed for predictions rather than embeddings with feature-level masking, may therefore underestimate the true leakage. We show this effect empirically in Section 5.2.

To overcome this limitation and to leverage the unique properties of TabFMs for stronger privacy auditing, we develop **Cascade Probing**, a method that recovers sensitive attributes sequentially rather than all at once. By decomposing the problem via the probabilistic chain rule and resolving easier attributes first, Cascade Probing exploits inter-attribute dependencies that simpler methods miss.

We apply Cascade Probing to audit four TabFMs (TabPFN [17], TabDPT [24], Mitra [43], TabICL [30]) in an evaluation spanning ten datasets, including four ACS Census benchmarks and two wider datasets with up to 100 features. Our audit reveals that leakage is severe across all models: Cascade Probing recovers 65–95% of sensitive attributes from embeddings alone, substantially more than existing inversion baselines. Models with higher-dimensional, more structured embeddings (TabDPT, TabICL) leak the most, while TabPFN's more entangled representations are harder to decode at baseline. Most critically, we discover a *Privacy Cliff*: knowing even a single public attribute of a record sharply increases recoverability of all remaining attributes, regardless of model or dataset. For practitioners, this means that sharing embeddings alongside even minimal metadata (e.g., demographic fields) can expose the entire record. Finally, we show that standard embedding-space mitigations (noise injection, PCA, knowledge distillation) fail to eliminate recoverability without destroying utility, indicating that privacy must be addressed at the representation level. Our main contributions are:

- We propose **Cascade Probing**, which leverages TabFMs' native NaN-masking to sequentially recover sensitive attributes via the chain rule. It recovers 65–95% of attributes from embeddings alone, outperforming all existing inversion baselines by large margins.
- We discover the **Privacy Cliff**: a single known attribute collapses the privacy of the entire record. This is intrinsic to how TabFMs encode information, not an artifact of the probing method.
- We provide the first systematic privacy audit across four TabFM architectures and ten datasets, showing that the vulnerability is systemic across domains and resists standard embedding-space mitigations.

2 Background

Tabular Foundation Models (TabFMs) are large-scale neural networks that generalize across diverse tabular datasets in a zero-shot manner via in-context learning (ICL). Given a support set $\mathcal{D}_{\text{sup}}$ of labeled examples at inference time, a TabFM produces a context-aware embedding $z = \mathcal{T}(x \mid \mathcal{D}_{\text{sup}})$ for each query record x before making predictions. Prior-Data Fitted Networks (PFNs) such as TabPFN [11, 16, 17] are trained on millions of synthetic datasets generated from structural causal models, effectively approximating Bayesian inference. Other architectures take different approaches: TabICL [30] uses a two-stage attention mechanism to scale to 500K samples, TabDPT [24] combines ICL with retrieval-based pre-training on real-world data, and Mitra [43] targets scalable deployment through AutoGluon. TabArena [7] demonstrates strong predictive performance of tabular foundation models on smaller datasets. Increasingly, users receive intermediate ICL embeddings z for tasks such as retrieval and clustering, rather than only final predictions. Recent analysis [40] reveals that these embeddings are highly separable and linearly decodable. While separability drives predictive performance, prior work on other modalities has shown that structured latent spaces can inadvertently expose sensitive attributes [23, 35]. Whether this risk applies to TabFM embeddings has not been studied and our work provides the first such audit.

Attribute Inference [8, 20] aims to reconstruct individual private attributes of data records from partial information, in contrast to full data reconstruction attacks typically studied in the image or language domain. The risk of attribute inference has been demonstrated across diverse settings, from social network analysis [21, 22, 45] to deep learning models [19, 36]. Our work studies this risk in a new modality: we measure recoverability from TabFM embeddings and optional side information, probing whether sensitive attribute values can be decoded from z. A comparison with other privacy risks, including membership inference and data extraction, is provided in Appendix D. Attribute inference is particularly relevant here because TabFM embeddings represent individual inference-time records, and the concrete risk of sharing them lies in the exposure of specific sensitive attribute values.

Embedding Inversion has been studied in NLP, where adversaries can recover text from sentence embeddings [34], with methods like Vec2Text [26] achieving high-fidelity reconstruction. This risk remains unexplored for TabFMs. Unlike holistic text reconstruction, the tabular domain requires recovering specific, heterogeneous attribute values. We address this gap by combining attribute inference with embedding inversion: our audit measures how accurately sensitive attribute values (x_{sens}) can be decoded from z and optionally known features (x_{pub}).

Tabular Data Synthesis. To enable auditing under data-scarce conditions, we leverage LLM-based tabular synthesis [3, 12, 33]. Building on few-shot tabular prompting [14] and chain-of-thought reasoning [38], we generate synthetic reference datasets from schema metadata and a small number of real samples to train the probing decoders (Section 4.5).

3 Auditing Attribute Leakage in TabFM Embeddings

As TabFM embeddings are increasingly shared for downstream use [28, 41, 43], auditing their privacy leakage is essential. Our goal is not to model a specific external adversary but to evaluate whether these embeddings are safe to share as data surrogates. Below, we formalize the audit objective and specify the assumptions under which recoverability is measured.

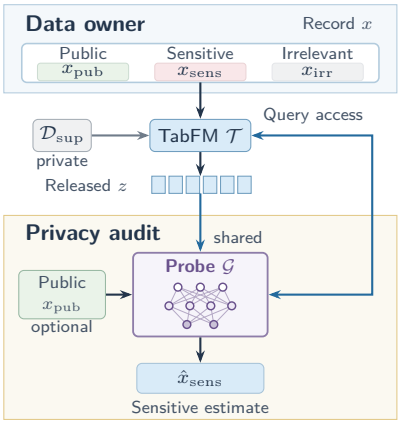


Figure 1: **Audit setup.** The auditor probes sensitive attributes using the released embedding, optional public attributes, and queries to the same TabFM. Support data remain private.

3.1 Audit Objective

We formulate the audit as a **conditional attribute recoverability** problem (Figure 1). A pre-trained TabFM $\mathcal{T}$ processes a query record x conditioned on a private support set $\mathcal{D}_{\text{sup}}$, producing a context-aware embedding $z = \mathcal{T}(x \mid \mathcal{D}_{\text{sup}})$. Each record x is partitioned into three disjoint sets: *public attributes* x_{pub}, which a recipient may plausibly know (e.g., Age, Race) and which serve as optional side information; *sensitive attributes* x_{sens}, whose recoverability we measure (e.g., Income, Diagnosis); and *irrelevant attributes* x_{irr}, the remaining features (e.g., Timestamp) masked with `NaN` during the audit. The set x_{pub} may be empty.

We estimate recoverability by learning a probing function $\mathcal{G}$ that approximates the maximum a posteriori (MAP) estimate:

$$\hat{x}_{\text{sens}} = \mathcal{G}(z, x_{\text{pub}}) \approx \underset{x_{\text{sens}}}{\operatorname{argmax}}\, P(x_{\text{sens}} \mid z, x_{\text{pub}}) \tag{1}$$

When $x_{\text{pub}} \neq \emptyset$, the probe exploits correlations between the side information and the embedding. When $x_{\text{pub}} = \emptyset$, recoverability is measured solely from z. High recoverability indicates that the embedding is unsafe to release without additional privacy protection.

3.2 Audit Assumptions

We parameterize the audit along three dimensions of information that may be available to parties who interact with the embedding downstream.

1. Access & Metadata. We assume **black-box query access to the same embedding endpoint** that produced the released embeddings. Crafted queries use the same TabFM and private support context, while model parameters and the private support set remain inaccessible. This evaluates what an internal auditor or an embedding recipient with such API access can recover. This setting reflects deployments in which users receive record embeddings and can submit additional records to the same embedding API [10, 28]. A static embedding release without access to the originating endpoint is outside our assumed access model. The auditor additionally possesses **metadata knowledge** of the dataset schema (e.g., column names, types). This is a minimal assumption: schema metadata is required to construct valid queries to the API in the first place, and is therefore available to any party with query access.

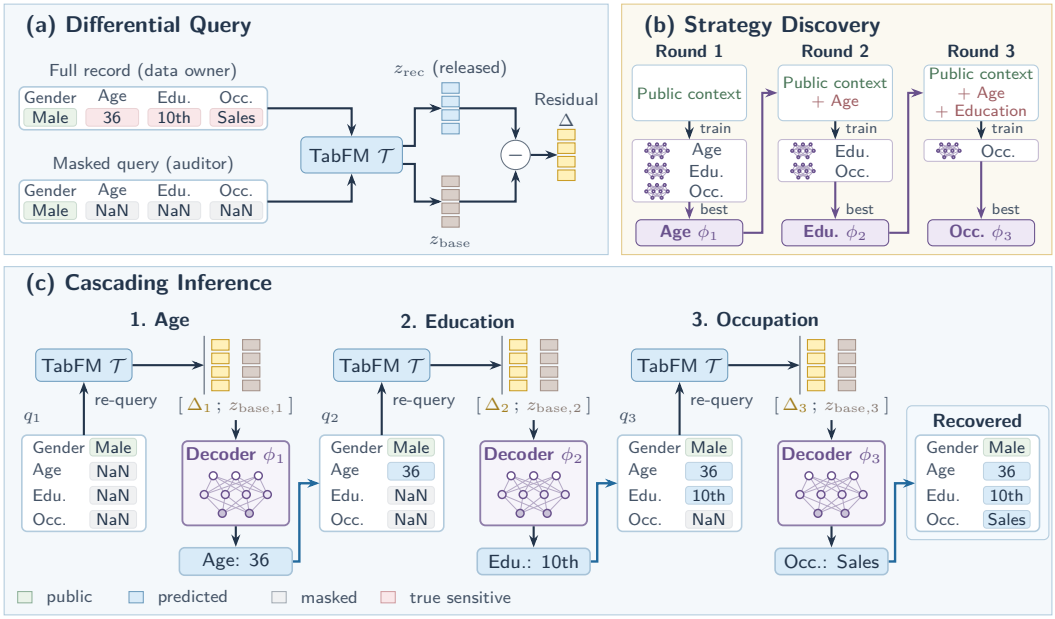


Figure 2: **Overview of Cascade Probing.** (a) Differential Query computes a baseline embedding from a masked query and subtracts it from the released embedding. (b) Strategy Discovery selects the best decoder on held-out reference data at each round, adding the selected attribute's true value to the next round's context. (c) Cascading Inference uses predicted values to update each query. The TabFM is queried again at every step, and the selected decoder receives the updated residual and baseline embedding. The released embedding stays fixed. The figure abbreviates the selected decoder ϕ_{π_t} as ϕ_t. Attribute values and recovery order are illustrative.

2. Side Information Level (K). We define the amount of side information about a specific record by $K = |x_{\text{pub}}|$, distinguishing two settings: **1) No-Context ($K = 0$):** The recipient knows nothing about the record ($x_{\text{pub}} = \emptyset$). Recoverability depends solely on the embedding. **2) Partial-Knowledge ($K \geq 1$):** The recipient knows at least one attribute value. This models realistic scenarios where some demographic or contextual information is available alongside the embedding.

3. Distribution Knowledge. We consider a spectrum from no knowledge to full knowledge of the data distribution, simulating what different recipients could achieve: **1) Shadow (full knowledge):** A recipient with access to a large same-distribution dataset ($\mathcal{D}_{\text{shadow}}$). This represents the strongest recipient and serves as an upper bound on recoverability. **2) Few-Shot (limited knowledge):** A recipient with only a handful of real samples (e.g., $N \leq 30$), used as seeds for LLM-based synthetic data generation. **3) Zero-Shot (no knowledge):** A recipient with no real data, relying on LLM-generated synthetic data from schema metadata alone.

4 Cascade Probing

The goal of Cascade Probing is to measure how accurately sensitive attributes of a record can be recovered from its TabFM embedding. Our key insight is that TabFMs encode strong feature dependencies; instead of predicting all sensitive attributes simultaneously, we probe them sequentially. By resolving more recoverable attributes first, we create a cleaner conditioning signal for probing harder ones. Our method, illustrated in Figure 2, consists of three stages: (1) **Differential Query**, (2) **Strategy Discovery**, and (3) **Cascading Inference**.

4.1 Differential Query

We leverage the native missing-value support of TabFMs to construct a reference embedding that separates public from sensitive information (Figure 2(a)). Let x be the target record. We construct

a masked baseline x_{base} by retaining the known public attributes x_{pub} while masking all sensitive attributes x_{sens} with `NaN`: $x_{\text{base}} = [x_{\text{pub}}, \texttt{NaN}, \ldots, \texttt{NaN}]$.

We query the model to obtain the baseline embedding $z_{\text{base}} = \mathcal{T}(x_{\text{base}})$. The **Embedding Residual** Δ is defined as the difference between the full record embedding z_{rec} and this baseline:

$$\Delta = z_{\text{rec}} - z_{\text{base}}. \tag{2}$$

The raw embedding z_{rec} is a composite representation encoding both public and sensitive features. By computing z_{base} as a reference, we obtain a differential view: the residual Δ captures the embedding shift induced by the sensitive attributes, while z_{base} retains the public context. We provide both as the concatenated input $[\Delta; z_{\text{base}}]$ to our probing models. Note that this is informationally equivalent to $[z_{\text{rec}}; z_{\text{base}}]$; the key enabler is the *differential query* itself, computing a reference embedding via native NaN-masking, which creates a baseline against which the sensitive signal can be measured.

4.2 Chain-Rule Decomposition

Tabular features exhibit strong dependencies. To exploit these dependencies, our **Cascade Probing** predicts sensitive attributes *one at a time*, conditioning each prediction on previously recovered attributes.

Joint Probing. As a baseline, Joint Probing masks all sensitive attributes and trains a single multi-task **probing decoder** to predict all sensitive targets simultaneously from the embedding representation.

Cascade Probing. Let $x_{\text{sens}} = \{a_1, \ldots, a_S\}$ be the set of S sensitive attributes and let $\mathcal{C} = \{z_{\text{rec}}, x_{\text{pub}}\}$ denote the initial information available to the auditor. By the probabilistic chain rule [2], the joint posterior can be factorized as

$$P(x_{\text{sens}} \mid \mathcal{C}) = \prod_{t=1}^{S} P(a_{\pi_t} \mid \mathcal{C}, \underbrace{a_{\pi_1}, \ldots, a_{\pi_{t-1}}}_{\text{Inferred Context}})$$

where π is a recovery order. This formulation motivates an auto-regressive probing procedure: at step t, we apply a per-attribute **probing decoder** ϕ_{π_t} that takes the residual and baseline embedding as input. Previously recovered values enter through the TabFM query used to compute these embeddings.

4.3 Strategy Discovery

The efficacy of the cascade depends critically on the recovery order π. Incorrect early predictions can propagate and amplify. To mitigate error propagation, we determine π offline via a greedy procedure on the auditor's reference dataset, prior to inference (Figure 2(b)).

Finding the globally optimal permutation is computationally intractable ($O(S!)$). We approximate it greedily by selecting the locally best attribute at each step ($O(S^2)$). As detailed in Algorithm 3 (Appendix E), at step t, given the set of previously selected attributes $\mathcal{R}$, we train a temporary probing decoder for every remaining attribute, conditioning on ground-truth values of $\mathcal{R}$ from the reference dataset. We evaluate each probe on a held-out validation set. In particular, the attribute with the best validation performance is selected as π_t and its ground-truth value is added to $\mathcal{R}$ for the next iteration. The selected decoder is retained for inference. This ensures the cascade recovers easier attributes first, building a stable context for harder ones.

4.4 Cascading Inference

At inference time (Figure 2(c)), ground-truth labels are unavailable, so the cascade uses its own predictions. At each step t, the auditor constructs a query q_t from x_{pub} and previously predicted values, leaving the remaining attributes masked. Querying the same TabFM endpoint gives $z_{\text{base},t} = \mathcal{T}(q_t)$ and the residual $\Delta_t = z_{\text{rec}} - z_{\text{base},t}$. The decoder ϕ_{π_t} retained during Strategy Discovery receives $[\Delta_t; z_{\text{base},t}]$ to predict the next attribute. The released embedding z_{rec} remains fixed throughout. Each prediction is locked into the context for the following step. Pseudocode for both Strategy Discovery and Cascading Inference is provided in Appendix E.

4.5 Implementation Details

Mixed feature types. Each probing decoder uses a classification head (Softmax + Cross-Entropy) for categorical attributes and a regression head for continuous ones. Continuous targets are min-max normalized to $[-1, 1]$ using bounds estimated from the reference dataset to prevent scale-dominant features from dominating training. During cascading inference, continuous predictions are inverse-transformed to their original scales before being fed back to the TabFM as context for subsequent steps. Details and ablations are in Appendix J.7.

LLM-based synthetic data. When a large reference dataset is unavailable, we synthesize $\mathcal{D}_{\text{syn}}$ via an LLM from schema metadata and a few real samples, using few-shot in-context learning, stratified conditional generation, and chain-of-thought prompting. $\mathcal{D}_{\text{syn}}$ is generated offline and used only to train the probing strategy (π, Φ).

5 Empirical Evaluation

We break our systematic privacy audit of TabFM embeddings down into five research questions: **RQ1:** How much attribute information is recoverable from TabFM embeddings? **RQ2:** Does partial side information sharply increase recoverability (the Privacy Cliff)? **RQ3:** Is the Privacy Cliff model-specific? **RQ4:** Is recoverability merely tabular correlation? **RQ5:** Can embedding transformations mitigate recoverability? We first describe the experimental setup, then address each RQ in turn.

5.1 Experimental Setup

Datasets. We evaluate on ten tabular datasets spanning diverse domains and scales. Four standard benchmarks serve as our primary evaluation suite: **Adult Census Income** [1], **German Credit** [15], **Heart Disease** [18], and **Diabetes Pima** [27]. We additionally evaluate on four ACS Census datasets from the folktables benchmark [6] to validate generalization across domains and feature compositions: **ACS Employment**, **ACS Public Coverage**, **ACS Income-to-Poverty**, and **ACS Health Insurance**. Each ACS dataset is subsampled to 50K records; TabPFN's default inference configuration further limits the support set $\mathcal{D}_{\text{sup}}$ to 10K samples. We further evaluate recovery on wider tables using COIL2000 [29] and Communities & Crime [31], with 85 and 100 features, respectively. Full dataset details are provided in Appendix B.

Evaluation Scenarios & Data Regimes. We evaluate attribute recoverability under varying side information and data access. In **No-Context** ($K = 0$), no side information is available ($x_{\text{pub}} = \emptyset$) and recoverability is measured solely from the embedding; in **Partial-Knowledge** ($K \geq 1$), we incrementally reveal public features (e.g., Race/Sex) to identify where recoverability saturates. Following Section 3, we consider three distribution knowledge regimes: **Shadow** (access to a same-distribution reference dataset comprising 50% of samples), **Few-Shot** (30 real samples to seed LLM synthesis), and **Zero-Shot** (metadata-only, LLM priors). For Few-Shot and Zero-Shot, we generate 2000 synthetic samples to train the probing decoders.

Splits and evaluation. We partition real records into disjoint support (40%), reference (50%), and test (10%) sets, subject to the support-size cap above. The reference pool in Shadow, or the synthetic pool in Few-Shot/Zero-Shot, is split 80/20 for decoder training/validation. All three regimes use the same held-out test records. We average results over three random seeds for each (K, S).

Attribute Partitioning. For each dataset, we designate a fixed set of *sensitive attributes* (x_{sens}) as recovery targets and a set of *public attributes* (x_{pub}) as potential side information, based on domain-informed priority lists. For example, on Adult, sensitive attributes include Occupation, Age, and Education, while public attributes include Race and Sex. Full priority lists and partitioning details for all datasets are in Appendix C.

Metrics. We report mean categorical accuracy (↑) for categorical features and Normalized Mean Absolute Error (NMAE, ↓, normalized to $[0, 1]$) for continuous features. We additionally report balanced accuracy and macro-F1 (mean class-wise F1 score), giving each class equal weight.

Models. We use TabPFN v2.5 [11] (192-dim embeddings) as the primary model under audit and validate cross-model generalizability on three additional TabFMs: TabDPT [24] (768-dim), Mitra [43] (512-dim), and TabICL [30] (512-dim). For synthetic data generation, we utilize Qwen2.5-7B-Instruct

Table 1: Mean categorical accuracy (↑) at $K=0$ (Shadow setting), averaged over all S. Best per row in bold. NMAE results in Appendix H.1.

Dataset	Joint	DirectOpt	WGAN	TabSyn	UIFV	Cascade
Adult Census Income	0.220	0.460	0.590	0.550	0.629	**0.650**
German Credit	0.440	0.540	0.620	0.610	0.499	**0.740**
Heart Disease	0.550	0.690	0.750	0.750	0.671	**0.790**
ACS Employment	0.517	0.620	0.620	0.680	0.882	**0.900**
ACS Public Coverage	0.460	0.450	0.440	0.420	0.673	**0.750**
ACS Income-to-Poverty	0.332	0.530	0.530	0.510	0.839	**0.845**
ACS Health Insurance	0.419	0.450	0.430	0.430	0.621	**0.686**

as the generator. Implementation details including decoder architecture and training hyperparameters are provided in Appendix F.

Decoder architecture. Probes use a three-layer MLP with hidden widths $[512, 256, 128]$, ReLU activations, and dropout 0.1. Joint Probing shares a backbone across target heads; Cascade Probing uses an independent decoder per step. We train with AdamW (learning rate 10^{-3}, batch size 64) for up to 100 epochs with early-stopping patience 10. Full hyperparameters are in Appendix F.

Baselines. We compare Cascade Probing against five baselines. (1) **Joint Probing** predicts all sensitive attributes simultaneously using a single multi-task decoder. (2) **Direct Optimization** [8] reconstructs attributes by gradient descent on the embedding distance. (3) **GMI+WGAN-GP** [13, 44] combines generative model inversion (GMI) with a tabular WGAN-GP prior. (4) **GMI+TabSyn VAE** [42, 44] replaces the GAN prior with the VAE component of TabSyn. (5) **UIFV Query Attack** [39] learns an inverse network to reconstruct attributes from intermediate representations using query access. We evaluate it on TabFM embeddings in the Shadow setting. Baselines without target embeddings are described in Section 5.5. Iterative refinement is evaluated in Appendix J.4.

5.2 RQ1: How Much Attribute Information Is Recoverable from TabFM Embeddings?

Motivation. We first establish the baseline privacy risk: how much can be recovered from a TabFM embedding when the auditor has no side information about the record ($K=0$)? This represents the minimum leakage inherent to the embedding itself.

Summary of Findings. On TabPFN v2.5, Cascade Probing achieves 65–90% categorical accuracy across the datasets with categorical features, achieving higher accuracy than all evaluated baselines. The gap is largest on complex, high-dimensional datasets (e.g., +51 percentage points over Joint Probing on ACS Income-to-Poverty), confirming that sequential conditional probing exploits inter-attribute dependencies that joint methods miss for risk reporting.

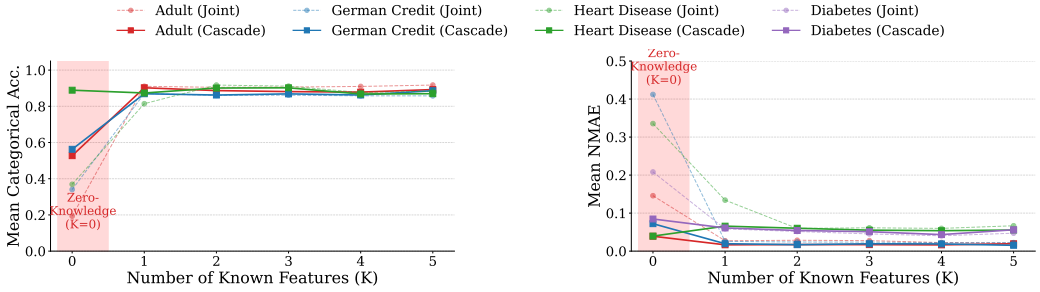
Detailed Results. Table 1 reports No-Context ($K=0$) recoverability in the Shadow setting. Cascade Probing outperforms all baselines on every dataset. Compared with UIFV, the gains range from 0.6 percentage points on ACS Income-to-Poverty to 24.1 points on German Credit. Optimization-based methods (DirectOpt, WGAN, TabSyn) improve over Joint Probing but remain substantially below Cascade, as they attempt to invert the entire embedding jointly without modeling inter-attribute dependencies. Diabetes Pima (all continuous) is reported via NMAE in Appendix H.1. Notably, even Joint Probing recovers 22–55% of attributes, indicating that TabFM embeddings encode substantial attribute information even under the simplest probing strategy. We further investigate whether this recoverability stems from the embedding itself rather than tabular correlations in Section 5.5.

Per-attribute recovery. Additional metrics show that recovery is not solely explained by class imbalance (Table 31). At $K=0$, Adult education achieves 0.65 balanced accuracy and 0.57 macro-F1 across 16 classes, compared with a constant predictor's balanced accuracy of $1/16 \approx 0.06$. German checking-account status achieves 0.98 balanced accuracy and 0.97 macro-F1 across four classes. Gains from $K=0$ to $K=1$ also remain evident in balanced accuracy and macro-F1.

Table 2 further shows that Cascade Probing yields effective audits under limited data access. Few-Shot (30 real samples + LLM synthesis) retains most Shadow-level recoverability, and on Heart Disease even exceeds Shadow (89% vs. 79%). Zero-Shot (no real data) achieves 49–88% accuracy,

Table 2: Cascade Probing at $K=0$ across data regimes, averaged over all S. Acc (↑) for categorical, NMAE (↓) for continuous features.

Dataset	Acc (↑)			NMAE (↓)		
	Shadow	Few-Shot	Zero-Shot	Shadow	Few-Shot	Zero-Shot
Adult	0.650	0.550	0.530	0.030	0.040	0.040
German	0.740	0.590	0.490	0.050	0.070	0.070
Heart	0.790	0.890	0.880	0.110	0.040	0.040
Diabetes	N/A	N/A	N/A	0.080	0.080	0.090



(a) **Categorical Features** (Higher is better). (b) **Continuous Features** (Lower is better).

Figure 3: **The Privacy Cliff.** Attribute recoverability as a function of side information (K known features) in the Few-Shot setting. The red shaded region highlights the No-Context ($K=0$) regime. Both Joint (dashed) and Cascade (solid) drastically jump at $K=1$; recoverability saturates for $K>1$.

comparable to Few-Shot, confirming that schema metadata alone enables practical auditing. Together, these results show that the practical privacy risk does not require access to a large reference dataset.

5.3 RQ2: Does Partial Side Information Sharply Increase Recoverability?

Motivation. In practice, a recipient of shared embeddings may also possess some public attributes of the individuals (e.g., demographic information). We investigate whether even a small amount of side information fundamentally changes recoverability.

Summary of Findings. A single known attribute triggers a drastic jump in recoverability across all datasets, for both Joint and Cascade Probing. This Privacy Cliff is intrinsic to the embeddings, and shows a sharp rise in recoverability at $K=1$ followed by saturation as further attributes are revealed.

Detailed Results. Figure 3 shows recoverability as side information increases from $K=0$ to $K=5$ (Few-Shot setting). At $K=1$, Joint surges from 21% to 91% on Adult and from 33% to 87% on German Credit; Cascade shows a similar jump on these datasets (55%→90% and 59%→87%, respectively). That even the simpler Joint method recovers nearly all attributes with one known feature confirms the cliff is a feature of the embedding, not an artifact of the probing method. Recoverability saturates for $K>1$, and the cliff appears regardless of which single attribute is revealed (Appendix J.3). This suggests that TabFM embeddings are highly **entangled**: knowing a single feature sharply reduces uncertainty over the remaining attributes, creating a binary privacy regime that shifts from substantial leakage at $K=0$ to broad exposure of remaining attributes at $K\geq 1$.

5.4 RQ3: Is the Privacy Cliff Model-Specific?

Motivation. The results above are obtained primarily on TabPFN. To determine whether the Privacy Cliff is a systemic property of TabFM representations rather than an artifact of a specific architecture, we evaluate Cascade Probing on three additional TabFMs with different embedding dimensions and training procedures.

Summary of Findings. All four TabFMs exhibit high recoverability and a clear $K=0 \rightarrow K=1$ jump, with the most pronounced cliff on TabPFN. The Privacy Cliff is not specific to TabPFN, but a shared property of TabFM representations.

Table 3: Cross-model generalization (Few-Shot, Cascade Probing). Accuracy (↑) for categorical features; NMAE (↓) for Diabetes Pima. The $K{=}0 \rightarrow K{=}1$ jump confirms the Privacy Cliff across all TabFMs.

	TabPFN		**TabDPT**		**Mitra**		**TabICL**	
Dataset	$K{=}0$	$K{=}1$	$K{=}0$	$K{=}1$	$K{=}0$	$K{=}1$	$K{=}0$	$K{=}1$
Adult (Acc)	0.550	0.900	0.930	0.960	0.860	0.900	0.950	0.970
German (Acc)	0.590	0.870	0.890	0.970	0.780	0.920	0.940	0.980
Heart (Acc)	0.890	0.870	0.880	0.930	0.870	0.920	0.890	0.930
Diabetes (NMAE)	0.080	0.070	0.043	0.005	0.063	0.020	0.070	0.032

Detailed Results. Table 3 shows that all four models exhibit high recoverability across all datasets. TabICL achieves 95% accuracy on Adult at $K{=}0$, meaning the embedding reveals nearly all sensitive attributes without any side information. The Privacy Cliff generalizes across architectures: every model shows a $K{=}0 \rightarrow K{=}1$ jump (e.g., Mitra on German: 78% → 92%; TabDPT on Diabetes NMAE: 0.043 → 0.005). Interestingly, TabDPT, Mitra, and TabICL show higher $K{=}0$ recoverability than TabPFN, suggesting their embeddings are more structured and thus more immediately decodable, while TabPFN's more entangled representations produce a steeper cliff.

5.5 RQ4: Is Recoverability Merely Tabular Correlation?

Motivation. High recoverability could stem from correlations among tabular features rather than information encoded in the embedding. We compare Cascade with four baselines that use public features and reference data without the target embedding.

Table 4: Attribute recovery from public features or embeddings. Baselines use K public features without the target embedding. Cascade uses the embedding at $K{=}0$. Accuracy (↑) for Adult, German, and Heart, and NMAE (↓) for Diabetes. Best per row in bold.

	KNN		**GBDT**		**MICE**		**TabPFN-imputer**		**Cascade**
Dataset	$K{=}1$	$K{=}4$	$K{=}1$	$K{=}3$	$K{=}1$	$K{=}3$	$K{=}1$	$K{=}3$	$K{=}0$
Adult	0.200	0.370	0.448	0.495	0.253	0.250	0.448	0.502	**0.650**
German	0.380	0.440	0.417	0.387	0.185	0.189	0.419	0.434	**0.740**
Heart	0.610	0.590	0.525	0.565	0.384	0.448	0.544	0.609	**0.790**
Diabetes	0.160	0.100	0.121	0.098	0.106	0.091	0.105	0.085	**0.080**

Summary of Findings. Cascade at $K{=}0$ outperforms all four baselines across the reported settings (Table 4). This supports the value of embedding information beyond recovery from revealed features alone under the evaluated baselines.

Detailed Results. KNN uses five neighbors in the Shadow setting. Gradient-boosted decision trees (GBDT) predict sensitive attributes from revealed features, multiple imputation by chained equations (MICE) performs chained conditional imputation, and TabPFN-imputer uses TabPFN for attribute imputation. All four baselines use public features and reference data without the target embedding, whereas Cascade receives the embedding with no public features. Across the reported public-feature settings, Cascade exceeds the best non-embedding accuracy by 14.8 percentage points on Adult, 30.0 on German, and 18.0 on Heart. The gap is smaller on Diabetes, where Cascade achieves 0.080 NMAE compared with 0.085 for TabPFN-imputer at $K{=}3$.

5.6 RQ5: Can Embedding Transformations Mitigate Recoverability?

Motivation. A natural follow-up question is whether simple post-processing of embeddings before release can reduce recoverability while preserving their downstream utility. We evaluate three representative embedding-space transformations.

Summary of Findings. No transformation eliminates recoverability while preserving utility. Gaussian noise at moderate levels ($\sigma{=}0.5$) offers the best tradeoff, but still leaves substantial leakage. Knowledge distillation is counterproductive, actually increasing recoverability.

Table 5: Mitigation evaluation at K=0, S=all (Shadow, Cascade Probing). Rec is categorical recoverability (↓), and Util is downstream k-NN accuracy (↑). For Diabetes, Rec is NMAE (↑). Arrows indicate better mitigation or preserved utility.

	Adult		**German**		**Heart**		**Diabetes**	
Mitigation	Rec	Util	Rec	Util	Rec	Util	Rec	Util
None	0.932	0.870	0.758	0.703	0.812	0.800	0.056	0.675
Noise σ=0.1	0.839	0.855	0.693	0.723	0.753	0.800	0.067	0.675
Noise σ=0.5	0.679	0.860	0.589	0.733	0.629	0.833	0.101	0.675
Noise σ=1.0	0.607	0.855	0.547	0.723	0.567	0.800	0.118	0.675
Noise σ=5.0	0.449	0.780	0.409	0.663	0.436	0.733	0.151	0.636
PCA k=4	0.734	0.860	0.637	0.713	0.666	0.800	0.083	0.740
PCA k=8	0.830	0.865	0.668	0.723	0.710	0.800	0.068	0.701
PCA k=32	0.921	0.870	0.741	0.703	0.798	0.800	0.057	0.675
KD (64-dim)	0.953	0.870	0.807	0.713	0.858	0.767	0.023	0.727

Detailed Results. We evaluate three embedding-space mitigations (Table 5): Gaussian noise injection, PCA dimensionality reduction, and knowledge distillation (KD). Gaussian noise at σ=0.5 offers the best privacy/utility tradeoff, reducing recoverability by 17–25 percentage points with k-NN utility unchanged or slightly improved (Adult: −1pp; German and Heart: +3pp; Diabetes: 0). Aggressive noise (σ=5.0) reduces recoverability further but degrades utility by 4–9 percentage points. PCA k=4 provides moderate protection; $k\geq$32 has negligible effect. KD is counterproductive: the compressed student representation *increases* recoverability by 2–5 percentage points on categorical datasets (and reduces Diabetes NMAE from 0.056 to 0.023), as the distilled embeddings concentrate task-relevant information. No mitigation eliminates recoverability while preserving embedding utility.

5.7 Additional Analyses

We conduct extensive ablations and additional experiments in the Appendix. **Recovery order** (Appendix J.1): we compare random, reverse, greedy, and exhaustive orderings, confirming that our greedy heuristic is near-optimal. **Decoder architecture** (Appendix J.2): we test 7 architectures spanning a 292× parameter range (49K–14.3M); even a linear probe achieves comparable recoverability, confirming the leakage is intrinsic to the embeddings. **Per-feature cliff** (Appendix J.3): we rank which anchor features most increase recoverability, finding that *any* single feature triggers the cliff. **Iterative refinement** (Appendix J.4): a Joint-then-refine baseline improves over Joint but remains substantially below Cascade, confirming that sequential ordering, not just iteration, is what helps. **L2 normalization** (Appendix J.5): standard post-processing does not reduce recoverability. **Target normalization** (Appendix J.7): min-max normalization is critical for Joint but Cascade is robust without it. **High-dimensional datasets** (Appendix J.8). Cascade outperforms all evaluated baselines on COIL2000 and Communities & Crime when recovering 50 sensitive attributes. **Computational overhead** (Appendix J.9). Discovery becomes costlier with more sensitive targets, but its one-time cost is amortized across records and inference remains fast.

6 Conclusion

We presented a systematic privacy audit of TabFM embeddings as data-sharing artifacts. Our method, **Cascade Probing**, leverages the chain rule to sequentially recover sensitive attributes from embeddings. It reveals substantially higher leakage than joint probing and optimization-based inversion baselines, providing a more faithful assessment of the true privacy risk. Our central finding is the **Privacy Cliff**: a single known attribute sharply increases the recoverability of all remaining attributes. This phenomenon generalizes across four TabFM architectures (TabPFN, TabDPT, Mitra, TabICL) and multiple datasets. Our evaluation of standard embedding-space mitigations (noise injection, PCA, knowledge distillation) shows that none eliminates recoverability without significantly harming utility, indicating that post-hoc defenses are insufficient. These results show that TabFM embeddings encode highly entangled attribute information and should not be assumed safe for a privacy-preserving release without explicit auditing.

Acknowledgments and Disclosure of Funding

FB has received funding from the European Research Council (ERC) under the European Union's Horizon Europe research and innovation programme (grant agreement No 101220235).

References

[1] Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

[2] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, Berlin, 2006. ISBN 9780387310732.

[3] Vadim Borisov, Kathrin Sessler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. Language models are realistic tabular data generators. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=cEygmQNOeI.

[4] Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 2021)*, pages 2633–2650. USENIX Association, 2021.

[5] Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *32nd USENIX Security Symposium (USENIX Security 2023)*, pages 5253–5270. USENIX Association, 2023.

[6] Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. Retiring adult: New datasets for fair machine learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[7] Nick Erickson, Lennart Purucker, Andrej Tschalzev, David Holzmüller, Prateek Desai, David Salinas, and Frank Hutter. Tabarena: A living benchmark for machine learning on tabular data. *Advances in Neural Information Processing Systems*, 38, 2026.

[8] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, page 1322–1333, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450338325. doi: 10.1145/2810103.2813677. URL https://doi.org/10.1145/2810103.2813677.

[9] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. Privacy in pharmacogenetics: an end-to-end case study of personalized warfarin dosing. In *Proceedings of the 23rd USENIX Conference on Security Symposium*, SEC'14, page 17–32, USA, 2014. USENIX Association. ISBN 9781931971157.

[10] Google Cloud. Bigquery ml: The ml.generate_embedding function. https://cloud.google.com/bigquery/docs/reference/standard-sql/bigqueryml-syntax-generate-embedding, 2025.

[11] Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Benjamin Jäger, Dominik Safaric, Simone Alessi, Adrian Hayler, Mihir Manium, Rosen Yu, Felix Jablonski, Shi Bin Hoo, Anurag Garg, Jake Robertson, Magnus Bühler, Vladyslav Moroshan, Lennart Purucker, Clara Cornu, Lilly Charlotte Wehrhahn, Alessandro Bonetto, Bernhard Schölkopf, Sauraj Gambhir, Noah Hollmann, and Frank Hutter. Tabpfn-2.5: Advancing the state of the art in tabular foundation models, 2025. URL https://arxiv.org/abs/2511.08667.

[12] Manbir S Gulati and Paul F Roysdon. TabMT: Generating tabular data with masked transformers. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=qs4swxtIAQ.

[13] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of Wasserstein GANs. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[14] Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. Tabllm: Few-shot classification of tabular data with large language models. In *International Conference on Artificial Intelligence and Statistics*, pages 5549–5581. PMLR, 2023.

[15] Hans Hofmann. Statlog (German Credit Data). UCI Machine Learning Repository, 1994. DOI: https://doi.org/10.24432/C5NC77.

[16] Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=cp5PvcI6w8_.

[17] Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637:319–326, 01 2025. doi: 10.1038/s41586-024-08328-6. URL https://www.nature.com/articles/s41586-024-08328-6.

[18] Andras Janosi, William Steinbrunn, Matthias Pfisterer, and Robert Detrano. Heart Disease. UCI Machine Learning Repository, 1989. DOI: https://doi.org/10.24432/C52P4X.

[19] Bargav Jayaraman and David Evans. Are attribute inference attacks just imputation? In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, page 1569–1582, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450394505. doi: 10.1145/3548606.3560663. URL https://doi.org/10.1145/3548606.3560663.

[20] Jinyuan Jia and Neil Zhenqiang Gong. Attriguard: a practical defense against attribute inference attacks via adversarial machine learning. In *Proceedings of the 27th USENIX Conference on Security Symposium*, SEC'18, page 513–529, USA, 2018. USENIX Association. ISBN 9781931971461.

[21] Jinyuan Jia, Binghui Wang, Le Zhang, and Neil Zhenqiang Gong. Attriinfer: Inferring user attributes in online social networks using markov random fields. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, page 1561–1569, Republic and Canton of Geneva, CHE, 2017. International World Wide Web Conferences Steering Committee. ISBN 9781450349130. doi: 10.1145/3038912.3052695. URL https://doi.org/10.1145/3038912.3052695.

[22] Michal Kosinski, David Stillwell, and Thore Graepel. Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences*, 110 (15):5802–5805, 2013. doi: 10.1073/pnas.1218772110. URL https://www.pnas.org/doi/abs/10.1073/pnas.1218772110.

[23] Francesco Locatello, Gabriele Abbati, Tom Rainforth, Stefan Bauer, Bernhard Schölkopf, and Olivier Bachem. *On the fairness of disentangled representations*. Curran Associates Inc., Red Hook, NY, USA, 2019.

[24] Junwei Ma, Valentin Thomas, Rasa Hosseinzadeh, Alex Labach, Jesse C. Cresswell, Keyvan Golestan, Guangwei Yu, Anthony L. Caterini, and Maksims Volkovs. TabDPT: Scaling tabular foundation models on real data. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=pIZxEOZCId.

[25] Shagufta Mehnaz, Sayanton V. Dibbo, Ehsanul Kabir, Ninghui Li, and Elisa Bertino. Are your sensitive attributes private? novel model inversion attribute inference attacks on classification models. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 4579–4596, Boston, MA, August 2022. USENIX Association. ISBN 978-1-939133-31-1. URL https://www.usenix.org/conference/usenixsecurity22/presentation/mehnaz.

[26] John Morris, Volodymyr Kuleshov, Vitaly Shmatikov, and Alexander Rush. Text embeddings reveal (almost) as much as text. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*,

pages 12448–12460, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.765. URL https://aclanthology.org/2023.emnlp-main.765/.

[27] National Institute of Diabetes, Digestive, and Kidney Diseases. Pima Indians Diabetes Database. UCI Machine Learning Repository, 1990.

[28] PriorLabs. Tabpfn capabilities: Embeddings. https://docs.priorlabs.ai/capabilities/embeddings, 2025.

[29] Peter Putten. Insurance company benchmark (COIL 2000). UCI Machine Learning Repository, 2000. URL https://doi.org/10.24432/C5630S. Dataset.

[30] Jingang QU, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICL: A tabular foundation model for in-context learning on large data. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=0VvD1PmNzM.

[31] Michael Redmond. Communities and crime. UCI Machine Learning Repository, 2002. URL https://doi.org/10.24432/C53W3X. Dataset.

[32] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership Inference Attacks Against Machine Learning Models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE Computer Society, 2017. doi: 10.1109/SP.2017.41. URL https://doi.ieeecomputersociety.org/10.1109/SP.2017.41.

[33] Aivin V. Solatorio and Olivier Dupriez. Realtabformer: Generating realistic relational and tabular data using transformers. *arXiv preprint arXiv:2302.02041*, 2023.

[34] Congzheng Song and Ananth Raghunathan. Information leakage in embedding models. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, CCS '20, page 377–390, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450370899. doi: 10.1145/3372297.3417270. URL https://doi.org/10.1145/3372297.3417270.

[35] Congzheng Song and Ananth Raghunathan. Information leakage in embedding models. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, CCS '20, page 377–390, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450370899. doi: 10.1145/3372297.3417270. URL https://doi.org/10.1145/3372297.3417270.

[36] Robin Staab, Mark Vero, Mislav Balunovic, and Martin Vechev. Beyond memorization: Violating privacy via inference with large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=kmn0BhQk7p.

[37] Valentin Thomas, Junwei Ma, Rasa Hosseinzadeh, Keyvan Golestan, Guangwei Yu, Maksims Volkovs, and Anthony L. Caterini. Retrieval & fine-tuning for in-context tabular models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=337dHOexCM.

[38] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

[39] Jirui Yang, Peng Chen, Zhihui Lu, Qiang Duan, and Yubing Bao. UIFV: Data reconstruction attack in vertical federated learning. In *2025 IEEE International Conference on Web Services (ICWS)*, pages 130–140. IEEE, 2025. doi: 10.1109/ICWS67624.2025.00026.

[40] Han-Jia Ye, Si-Yang Liu, and Wei-Lun Chao. A closer look at tabpfn v2: Understanding its strengths and extending its capabilities. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2502.17361.

[41] Chin-Chia Michael Yeh, Uday Singh Saini, Xin Dai, Xiran Fan, Shubham Jain, Yujie Fan, Jiarui Sun, Junpeng Wang, Menghai Pan, Yingtong Dou, Yuzhong Chen, Vineeth Rakesh, Liang Wang, Yan Zheng, and Mahashweta Das. Treasure: The visa payment foundation model for high-volume transaction understanding, 2025. URL https://arxiv.org/abs/2511.19693.

[42] Hengrui Zhang, Jiani Zhang, Zhengyuan Shen, Balasubramaniam Srinivasan, Xiao Qin, Christos Faloutsos, Huzefa Rangwala, and George Karypis. Mixed-type tabular data synthesis with score-based diffusion in latent space. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=4Ay23yeuz0.

[43] Xiyuan Zhang, Danielle C. Maddix, Junming Yin, Nick Erickson, Abdul Fatir Ansari, Boran Han, Shuai Zhang, Leman Akoglu, Christos Faloutsos, Michael W. Mahoney, Cuixiong Hu, Huzefa Rangwala, George Karypis, and Bernie Wang. Mitra: Mixed synthetic priors for enhancing tabular foundation models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=t8YRsWY6HM.

[44] Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. The secret revealer: Generative model-inversion attacks against deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 253–261, 2020.

[45] Elena Zheleva and Lise Getoor. To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, page 531–540, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605584874. doi: 10.1145/1526709.1526781. URL https://doi.org/10.1145/1526709.1526781.

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 abstract and introduction (Section 1) clearly state the three main contributions: Cascade Probing, the Privacy Cliff phenomenon, and the systematic cross-model audit. All claims are supported by experimental results in Section 5 across four TabFMs and ten datasets.

   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 Impact Statement and Limitations section (Appendix A) includes a dedicated **Limitations** paragraph discussing the scope and open problems of our work.

   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: [N/A]

Justification: The paper is primarily empirical and does not include formal theorems or proofs. The chain-rule decomposition in Section 4 is a standard probabilistic identity used to motivate the method design, not a novel theoretical result.

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: Section 5 describes the experimental setup including datasets, models, metrics, and baselines. Appendix F provides full decoder architecture, training hyperparameters (optimizer, learning rate, batch size, epochs), data splits, and evaluation protocol. Appendix G includes the LLM prompt templates used for synthetic data generation.

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: Code is included as supplementary material. All datasets used (Adult, German Credit, Heart Disease, Diabetes Pima, ACS Census via folktables) are publicly available. The TabFM models (TabPFN, TabDPT, Mitra, TabICL) are publicly released by their respective authors.

   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: Section 5 describes the evaluation scenarios, metrics, and baselines. Appendix F provides complete details: decoder architecture (3-layer MLP with specific dimensions), optimizer (AdamW), learning rate (10^{-3}), batch size (64), epochs (100 with early stopping), data splits (40/50/10 for Shadow; 80/20 for decoder training/validation), and LLM generation hyperparameters.

   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 results (Tables 1, 2) are averaged over all S (number of sensitive attributes). The recovery order ablation (Table 34, Appendix J.1) reports standard deviations ($\pm$ std) for random orderings over 5 permutations.

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: Appendix J.9 reports wall-clock times for Strategy Discovery and per-record inference on an NVIDIA A100 GPU, broken down by the number of sensitive attributes S.

   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: We have reviewed the NeurIPS Code of Ethics. Our work is a privacy audit designed to expose vulnerabilities in TabFM embeddings and motivate stronger protections. We discuss dual-use risks in the Impact Statement and Limitations section (Appendix A). All datasets used are publicly available benchmarks.

   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: The Impact Statement and Limitations section (Appendix A) discusses both positive impacts (enabling privacy auditing, motivating stronger protections) and negative impacts (potential misuse of the audit methodology to recover sensitive attributes). It also discusses dual-use risks of LLM-based synthetic data for auditing.

    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 paper does not release pre-trained models or scraped datasets. The released audit framework is a diagnostic tool designed to help data owners evaluate embedding privacy before deployment, analogous to standard security auditing tools.

    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: All datasets and models are properly cited. The datasets (Adult, German Credit, Heart Disease, Diabetes Pima) are from the UCI ML Repository, and ACS Census datasets are from the folktables benchmark [6]. All TabFM models are cited with their original papers.

    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: We release the Cascade Probing audit framework as supplementary material. The code is documented with instructions for reproducing the main experiments.

    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: The paper does not involve crowdsourcing or research with human subjects.

    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: The paper does not involve human subjects research. All experiments use publicly available benchmark datasets.

    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: [Yes]

    Justification: LLMs (Qwen2.5-7B-Instruct) are used as a component of our method for synthetic reference data generation in Few-Shot and Zero-Shot settings (Section 4.5). The LLM usage, prompts, and hyperparameters are fully described in Section 5 and Appendix G.

    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.

A Impact Statement and Limitations

This work reveals that TabFM embeddings, increasingly shared as substitutes for raw tabular records, retain highly recoverable attribute-level information. By introducing **Cascade Probing**, we show that inter-attribute correlations within embeddings can be systematically exploited to recover sensitive data, demonstrating that current embeddings are far more transparent than previously assumed. Our discovery of the "Privacy Cliff", where minimal side information triggers a sharp increase in recoverability, challenges the assumption that embeddings can be safely released or shared across institutional boundaries. Furthermore, our validation using LLM-synthesized reference data shows that effective auditing (and, by extension, potential misuse) requires minimal real data, lowering the practical barrier. These results underscore an urgent need for privacy-preserving mechanisms, such as differentially private embedding generation, to safeguard structured data in high-stakes domains. We release our audit framework to help data owners evaluate embedding privacy before deployment.

Limitations. This work focuses on diagnosing privacy leakage in TabFM embeddings but does not propose a defense that eliminates recoverability while preserving utility. Our evaluation of

post-hoc mitigations (noise, PCA, knowledge distillation) shows that none is sufficient, but designing principled privacy-preserving embedding mechanisms remains an open problem that requires further research. Additionally, the quality of LLM-synthesized reference data depends on the generator model and the informativeness of the schema metadata; domains with highly specialized or proprietary feature semantics may yield weaker synthetic data, limiting the applicability of the Few-Shot and Zero-Shot audit settings.

B Dataset Details

Table 6 summarizes the ten datasets used in our evaluation.

Table 6: Summary of evaluation datasets. Feature counts and types refer to the representations used in our experiments.

Dataset	Samples	Features	Cat.	Cont.	Domain
Adult Census	48,842	13	8	5	Demographics
German Credit	1,000	20	13	7	Finance
Heart Disease	303	13	8	5	Healthcare
Diabetes Pima	768	8	0	8	Healthcare
ACS Employment	50,000	16	16	0	Census
ACS Pub. Coverage	50,000	19	19	0	Census
ACS Inc.-to-Poverty	50,000	19	19	0	Census
ACS Health Ins.	50,000	25	25	0	Census
COIL2000	9,822	85	85	0	Insurance
Communities & Crime	1,994	100	0	100	Socioeconomic

Adult Census Income [1]: Predicts whether income exceeds \$50K/year based on census data. Contains demographic features (age, sex, race), employment features (workclass, occupation), and financial features (capital-gain/loss).

German Credit [15]: Credit risk assessment dataset with 20 features covering personal information (age, housing, job), account status (checking, savings), and loan details (amount, duration, purpose).

Heart Disease (Cleveland) [18]: Cardiac disease diagnosis with 13 clinical features including demographics (age, sex), symptoms (chest pain type), and medical measurements (blood pressure, cholesterol, heart rate).

Diabetes Pima [27]: Diabetes prediction for Pima Indian women with 8 continuous features: pregnancies, glucose, blood pressure, skin thickness, insulin, BMI, diabetes pedigree function, and age.

ACS Employment [6]: Employment prediction from the US Census ACS PUMS 2018 (all 50 states + PR) via the folktables benchmark. Contains 16 categorical features covering demographics (age, sex, race, marital status, citizenship, nativity), disability indicators (hearing, vision, cognitive, ambulatory), military service, and household relationship. Subsampled to 50K.

ACS Public Coverage [6]: Public health insurance coverage prediction from the same ACS source. Contains 19 categorical features including the employment features above plus state (51 categories), income (binned into 7 brackets), employment status, and fertility. Subsampled to 50K.

ACS Income-to-Poverty Ratio [6]: Poverty prediction (income-to-poverty ratio $< 250\%$) from ACS. Contains 19 categorical features including demographics, disability indicators, employment status, weekly work hours, and grandchild care. Subsampled to 50K.

ACS Health Insurance [6]: Private health insurance prediction from ACS. Contains 25 categorical features, the largest feature set among the ACS datasets, covering demographics, disability, employment, household, and income variables. Subsampled to 50K.

COIL2000 [29] contains customer demographics and insurance-product information. Our evaluation uses 9,822 records and treats the 85 input features as categorical.

Communities & Crime [31] combines socioeconomic, law-enforcement, and crime statistics for US communities. Our evaluation uses 1,994 records and 100 continuous input features. Both wider

datasets are evaluated in the Shadow setting at $K{=}0$ with $S{=}50$ sensitive targets, with additional timing measurements at $S \in \{10, 30, 50\}$.

C Feature Priority Lists

To systematically evaluate recoverability across features of varying sensitivity and observability, we define feature priority lists for each dataset (see Tables 8, 10, 12 and 14). These priority lists establish the order in which features are considered during evaluation:

- **Public Priority**: Features ordered from most public (easily observable) to least public. These represent information a recipient could plausibly know.
- **Sensitive Priority**: Features ordered from most sensitive (highest privacy concern) to least sensitive. These represent the primary recovery targets.

The priority lists serve two purposes:

1. **Feature Selection Order**: When evaluating Cascade Probing, features are processed according to these priority orderings.
2. **Realistic Scenarios**: The orderings reflect real-world settings where recipients typically possess easily observable demographic information while sensitive financial or health data is held out.

Note that a feature may appear in both priority lists. For a given configuration (K, S), we first select the top-K features from the public list as x_{pub}, then select the top-S features from the sensitive list that are not already in x_{pub} as x_{sens} (skipping any overlap). This ensures the partition is always disjoint.

C.1 Adult Census Income

The Adult Census Income dataset (Table 7) contains 13 features for predicting whether income exceeds \$50K/year.

Table 7: Features in the Adult dataset.

Idx	Feature	Type
0	age	Integer
1	workclass	Categorical
2	education	Categorical
3	education-num	Integer
4	marital-status	Categorical
5	occupation	Categorical
6	relationship	Categorical
7	race	Categorical
8	sex	Binary
9	capital-gain	Integer
10	capital-loss	Integer
11	hours-per-week	Integer
12	native-country	Categorical

C.2 German Credit

The German Credit dataset (Table 9) contains 20 features for credit risk assessment.

C.3 Heart Disease

The Heart Disease (Cleveland) dataset (Table 11) contains 13 features for cardiac disease diagnosis.

Table 8: Priority lists for Adult dataset.

Public Priority		Sensitive Priority	
Rank	**Feature**	**Rank**	**Feature**
1	race	1	occupation
2	sex	2	age
3	native-country	3	education
4	education	4	marital-status
5	education-num	5	relationship
6	workclass	6	capital-gain
7	hours-per-week	7	capital-loss
8	capital-gain	8	hours-per-week
9	capital-loss	9	workclass
10	marital-status	10	education-num
11	occupation	11	native-country
12	relationship	12	race
13	age	13	sex

Table 9: Features in the German Credit dataset.

Idx	**Feature**	**Type**	**Idx**	**Feature**	**Type**
0	checking_account	Categorical	10	present_residence	Integer
1	duration	Integer	11	property	Categorical
2	credit_history	Categorical	12	age	Integer
3	purpose	Categorical	13	other_installments	Categorical
4	credit_amount	Integer	14	housing	Categorical
5	savings_account	Categorical	15	num_credits	Integer
6	employment_duration	Categorical	16	job	Categorical
7	installment_rate	Integer	17	num_dependents	Integer
8	personal_status_sex	Categorical	18	telephone	Binary
9	other_debtors	Categorical	19	foreign_worker	Binary

Table 10: Priority lists for German Credit dataset.

Public Priority		Sensitive Priority	
Rank	**Feature**	**Rank**	**Feature**
1	foreign_worker	1	credit_amount
2	age	2	checking_account
3	personal_status_sex	3	savings_account
4	housing	4	credit_history
5	job	5	num_credits
6	telephone	6	purpose
7	num_dependents	7	duration
8	present_residence	8	installment_rate
9	property	9	employment_duration
10	other_installments	10	other_debtors
11	other_debtors	11	other_installments
12	employment_duration	12	property
13	num_credits	13	present_residence
14	installment_rate	14	num_dependents
15	checking_account	15	job
16	savings_account	16	housing
17	credit_history	17	personal_status_sex
18	purpose	18	age
19	duration	19	telephone
20	credit_amount	20	foreign_worker

Table 11: Features in the Heart Disease dataset.

Idx	Feature	Type
0	age	Integer
1	sex	Binary
2	chest_pain_type	Categorical
3	resting_bp	Integer
4	cholesterol	Integer
5	fasting_blood_sugar	Binary
6	rest_ecg	Categorical
7	max_heart_rate	Integer
8	exercise_angina	Binary
9	oldpeak	Continuous
10	slope	Categorical
11	num_vessels	Categorical
12	thal	Categorical

Table 12: Priority lists for Heart Disease dataset.

Public Priority		Sensitive Priority	
Rank	Feature	Rank	Feature
1	sex	1	fasting_blood_sugar
2	age	2	exercise_angina
3	chest_pain_type	3	chest_pain_type
4	fasting_blood_sugar	4	slope
5	rest_ecg	5	thal
6	exercise_angina	6	num_vessels
7	slope	7	rest_ecg
8	thal	8	max_heart_rate
9	num_vessels	9	resting_bp
10	oldpeak	10	cholesterol
11	max_heart_rate	11	oldpeak
12	resting_bp	12	age
13	cholesterol	13	sex

C.4 Diabetes Pima

The Diabetes Pima dataset (Table 13) contains 8 features for diabetes prediction in Pima Indian women. All features are continuous.

Table 13: Features in the Diabetes Pima dataset.

Idx	Feature	Type	Range
0	pregnancies	Integer	0–17
1	glucose	Continuous	44–199 mg/dl
2	blood_pressure	Continuous	24–122 mmHg
3	skin_thickness	Continuous	7–99 mm
4	insulin	Continuous	14–846 μU/ml
5	bmi	Continuous	18.2–67.1 kg/m^2
6	diabetes_pedigree	Continuous	0.08–2.42
7	age	Integer	21–81

D Privacy Attacks and Attribute Inference

Several classes of privacy attacks have been studied in the ML literature. We review three relevant attack families and their applicability to the embedding-sharing setting.

Membership Inference tests whether a specific record was part of the model's training set [32]. It is widely used for auditing trained models but targets training-data memorization and yields only a

Table 14: Priority lists for Diabetes Pima dataset.

Public Priority		Sensitive Priority	
Rank	**Feature**	**Rank**	**Feature**
1	glucose	1	pregnancies
2	bmi	2	age
3	age	3	blood_pressure
4	pregnancies	4	glucose
5	diabetes_pedigree	5	skin_thickness
6	skin_thickness	6	insulin
7	blood_pressure	7	diabetes_pedigree
8	insulin	8	bmi

Algorithm 1 Joint Probing (Training)

Input: Training Dataset $\mathcal{D}_{\text{train}}$, Embedding Model $\mathcal{T}$, Sets $\mathcal{X}_{\text{pub}}, \mathcal{X}_{\text{sens}}$
Output: Multi-Task Decoder Φ_{joint}
Initialize: Multi-Task Decoder Φ_{joint} with $|\mathcal{X}_{\text{sens}}|$ output heads
Compute $Z^{(i)}_{\text{full}} \leftarrow \mathcal{T}(x^{(i)})$ for all $x^{(i)} \in \mathcal{D}_{\text{train}}$
for each batch $(x^{(i)}, x^{(i)}_{\text{sens}}) \in \mathcal{D}_{\text{train}}$ **do**
    Mask inputs: $X^{(i)}_{\text{base}} \leftarrow \text{Mask}(x^{(i)}, \text{keep} = \mathcal{X}_{\text{pub}})$
    Compute base embedding: $z^{(i)}_{\text{base}} \leftarrow \mathcal{T}(X^{(i)}_{\text{base}})$
    Compute signals: $\Delta^{(i)} \leftarrow Z^{(i)}_{\text{full}} - z^{(i)}_{\text{base}}$
    Predict: $\hat{y}^{(i)} \leftarrow \Phi_{\text{joint}}([\Delta^{(i)}; z^{(i)}_{\text{base}}])$
    Update Φ_{joint} to minimize $\sum_{f \in \mathcal{X}_{\text{sens}}} \mathcal{L}(\hat{y}^{(i)}_f, x^{(i)}_{\text{sens}}[f])$
end for
Return Φ_{joint}

single bit of information (member or not). TabFM embeddings are generated for query records at inference time; the privacy risk lies in what the embedding reveals about the individual record, not in whether that record was used for training.

Data Extraction recovers verbatim training samples from a model, as demonstrated for large language models [4] and diffusion models [5]. Like membership inference, data extraction targets the training set rather than inference-time query records. Tabular records are also structured and heterogeneous (mixed categorical and continuous features), making verbatim extraction less informative than recovering specific attribute values.

Attribute Inference recovers individual sensitive attribute values of a specific record from the model's output [8, 19, 25]. Given an embedding z and optionally some public attributes, it measures how accurately the remaining sensitive columns can be decoded. The recovered values (e.g., income, diagnosis, occupation) are concrete and actionable. Since TabFM embeddings represent individual inference-time records, and the risk of sharing them lies in the exposure of specific attribute values, attribute inference provides the most direct measure of privacy leakage in this setting.

E Algorithm Details

E.1 Joint Probing

Joint Probing serves as a baseline that predicts all sensitive attributes simultaneously using a multi-task decoder trained on embedding residuals. Unlike Cascade Probing, it does not exploit inter-attribute dependencies. Pseudocode is given in Algorithms 1 and 2.

E.2 Strategy Discovery

Strategy Discovery is the offline procedure that determines the recovery order π and trains the per-attribute decoders Φ. Given a reference dataset, it greedily selects the next attribute to recover

Algorithm 2 Joint Probing (Inference)

Input: Record Embedding z_{rec}, Public Features x_{pub}, Decoder Φ_{joint}, Model $\mathcal{T}$
Output: Recovered Sensitive Attributes $\hat{x}_{\text{sens}}$
Let query input $q \leftarrow x_{\text{pub}}$ (pad missing with NaN)
Compute base embedding: $z_{\text{base}} \leftarrow \mathcal{T}(q)$
Compute residual: $\Delta \leftarrow z_{\text{rec}} - z_{\text{base}}$
Predict all targets: $\hat{x}_{\text{sens}} \leftarrow \Phi_{\text{joint}}([\Delta; z_{\text{base}}])$
Return $\hat{x}_{\text{sens}}$

Algorithm 3 Strategy Discovery

Input: Reference Dataset $\mathcal{D}_{\text{shadow}} = \{\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{val}}\}$, Embedding Model $\mathcal{T}$, Sets $\mathcal{X}_{\text{pub}}, \mathcal{X}_{\text{sens}}$
Output: Inference Order π, Ordered Decoders Φ
Initialize: $\mathcal{U} \leftarrow \mathcal{X}_{\text{sens}}$, $\mathcal{R} \leftarrow \mathcal{X}_{\text{pub}}$, $\pi \leftarrow []$, $\Phi \leftarrow []$
Compute $Z^{(i)}_{\text{full}} \leftarrow \mathcal{T}(x^{(i)})$ for all $x^{(i)} \in \mathcal{D}_{\text{shadow}}$
while $\mathcal{U} \neq \emptyset$ **do**
  $best_score \leftarrow -\infty$, $best_pair \leftarrow \text{null}$
  Mask inputs: $X^{(i)}_{\text{base}} \leftarrow \text{Mask}(x^{(i)}, \text{keep} = \mathcal{R})$
  Compute signals: $\Delta^{(i)} \leftarrow Z^{(i)}_{\text{full}} - \mathcal{T}(X^{(i)}_{\text{base}})$
  for each candidate feature $f \in \mathcal{U}$ **do**
    Optimize ϕ_f on $\mathcal{D}_{\text{train}}$ to minimize $\mathcal{L}(\phi_f([\Delta^{(i)}; \mathcal{T}(X^{(i)}_{\text{base}})]), x^{(i)}_f)$
    $score \leftarrow \text{Perf}(\phi_f, \mathcal{D}_{\text{val}})$
    if $score > best_score$ **then**
      $best_score \leftarrow score$, $best_pair \leftarrow (f, \phi_f)$
    end if
  end for
  $(f^*, \phi^*) \leftarrow best_pair$
  π.Append(f^*), Φ.Append(ϕ^*)
  $\mathcal{U} \leftarrow \mathcal{U} \setminus \{f^*\}$, $\mathcal{R} \leftarrow \mathcal{R} \cup \{f^*\}$
end while
Return π, Φ

by training a probe for every remaining candidate and picking the one with the best validation performance, then conditions subsequent steps on the ground-truth value of the selected attribute. Pseudocode is given in Algorithm 3.

E.3 Cascading Inference

At inference time, Cascading Inference applies the trained decoders sequentially in the order π. At each step, the auditor reconstructs the query from previously recovered attributes, recomputes the residual against the target embedding, and predicts the next sensitive value. Pseudocode is given in Algorithm 4.

F Implementation Details

This section provides the implementation details for reproducing our experiments.

F.1 Decoder Architecture

Both Joint and Cascade decoders use a shared backbone architecture:

- **Input**: Concatenated residual and base embedding $[\Delta; z_{\text{base}}]$ (dimension: $2 \times d_{\text{emb}}$, where $d_{\text{emb}} = 192$ for TabPFN v2.5)
- **Backbone**: 3-layer MLP with hidden dimensions [512, 256, 128], ReLU activations, and dropout ($p = 0.1$) between layers

Algorithm 4 Cascading Inference

Input: Record embedding z_{rec}, public features x_{pub}, strategy (π, Φ), model $\mathcal{T}$
Output: Recovered sensitive attributes $\hat{x}_{\text{sens}}$
Initialize context $\hat{\mathcal{R}} \leftarrow x_{\text{pub}}$
for $t = 1$ **to** $|\pi|$ **do**
  Construct query $q \leftarrow \hat{\mathcal{R}}$ (pad missing with NaN)
  $\Delta \leftarrow z_{\text{rec}} - \mathcal{T}(q)$ ▷ Residual
  $\hat{x}_{\text{sens}}[\pi_t] \leftarrow \Phi_t([\Delta;\, \mathcal{T}(q)])$ ▷ Predict
  $\hat{\mathcal{R}}[\pi_t] \leftarrow \hat{x}_{\text{sens}}[\pi_t]$ ▷ Update context
end for
Return $\hat{x}_{\text{sens}}$

- **Output Heads**:
  - Categorical features: Linear layer $\rightarrow$ Softmax (output dim = number of categories)
  - Continuous features: Linear layer $\rightarrow$ Tanh (output dim = 1, scaled to $[-1, 1]$)

For **Joint Probing**, a single backbone feeds into $|\mathcal{X}_{\text{sens}}|$ parallel output heads. For **Cascade Probing**, each step t uses an independent decoder ϕ_{π_t} with the same architecture.

F.2 Training Hyperparameters

Table 15: Training hyperparameters for decoder optimization.

Parameter	Value
Optimizer	AdamW
Learning rate	1×10^{-3}
Weight decay	1×10^{-4}
Batch size	64
Epochs	100 (with early stopping)
Early stopping patience	10 epochs
Train/Val split	80% / 20%
Loss Functions	
Categorical targets	Cross-Entropy
Continuous targets	MSE (on normalized values)

F.3 Data Splits and Evaluation Protocol

Shadow Setting. We split the original dataset into three disjoint sets: 40% for TabPFN support (not accessible to the auditor), 50% as the reference dataset $\mathcal{D}_{\text{shadow}}$ for training probing decoders, and 10% as held-out test records for evaluation. The reference dataset is further split 80/20 for decoder training and validation.

Few-Shot and Zero-Shot Settings. For these low-resource scenarios, we sample N_{shot} real records (30 for Few-Shot, 0 for Zero-Shot) to prompt the LLM. We generate 2000 synthetic samples using the LLM, split 80/20 for decoder training/validation. Evaluation uses the same held-out test records as the Shadow setting for fair comparison.

Evaluation. For each (K, S) configuration, we report mean accuracy (categorical) and mean NMAE (continuous) averaged over 3 random seeds.

G LLM Augmentation Prompts

This section presents the prompt templates used for LLM-based synthetic data generation. We employ dataset-specific schemas containing feature names, descriptions, value ranges, categories, and domain constraints.

G.1 Few-Shot Prompt Template

For few-shot augmentation ($N > 0$), we use stratified generation with real examples. The prompt combines chain-of-thought reasoning with a constraint on a categorical feature to ensure diverse coverage.

```
[System] You are generating synthetic tabular data with a SPECIFIC CONSTRAINT. Before
generating, THINK about what characteristics are typical for this constraint.

DATASET: {dataset_name} - {description}

FEATURES:
- {feature_1}: {description_1} ({type_1}, range: {min_1}-{max_1})
- {feature_2}: {description_2} (categories: {categories_2})
- ...

CONSTRAINT: All samples must have {stratify_feature} = ‘‘{value}’’

EXAMPLES (study the relationships for this {stratify_feature} value):
{fewshot_examples_json}

GENERATE {n_samples} samples that:
- ALL have {stratify_feature} = ‘‘{value}’’ (REQUIRED)
- Maintain realistic correlations between features
- Show diversity within the group
- Are unique and not copies of examples

Return ONLY a valid JSON array.
```

Figure 4: Few-shot prompt template for LLM-based data augmentation.

G.2 Zero-Shot Prompt Template

For zero-shot generation ($N = 0$), our prompt incorporates three key components to elicit diverse samples without real data:

1. **Synthetic Diverse Examples**: We programmatically generate 3 example records that demonstrate the expected diversity, with values sampled uniformly across valid ranges for each feature.
2. **Value Distribution Requirements**: Explicit per-feature constraints specifying that categorical features must use ALL categories, binary features must include BOTH values, and continuous features must span LOW, MEDIUM, and HIGH ranges.
3. **Domain Constraints**: Dataset-specific correlations (e.g., for healthcare data: “max heart rate decreases with age”, “exercise-induced angina correlates with ST depression”).

```
DATASET: {dataset_name} - {description}

FEATURES:
- {feature_1}: {description_1} ({type_1}, range: {min_1}-{max_1})
- ...

DOMAIN CONSTRAINTS:
- {constraint_1}
- {constraint_2}
- ...

===== CRITICAL: DIVERSITY REQUIREMENT =====
Your output MUST look like these DIVERSE examples:
{synthetic_examples_json}

===== MANDATORY VALUE DISTRIBUTION =====
- {feature_1}: MUST use ALL categories [{cats}] across samples
- {feature_2}: MUST vary from {min} to {max}, use LOW/MED/HIGH
- ...

Generate {n_samples} samples as a JSON array.
```

Figure 5: Zero-shot prompt template with synthetic examples and value distribution requirements.

G.3 Diversity Hints

We rotate through different diversity hints across generation batches:

- "Focus on samples from the lower end of numerical ranges."
- "Focus on samples from the upper end of numerical ranges."
- "Generate samples representing edge cases near boundaries."
- "Generate a diverse mix covering the full valid range."

G.4 LLM Configuration

All experiments use the following settings:

Table 16: LLM generation hyperparameters.

Parameter	Value
LLM Model	Qwen2.5-7B-Instruct
Temperature	$\tau = 0.9$
Nucleus sampling	top-$p = 0.95$
Top-k sampling	$k = 100$
Repetition penalty	1.1
Samples generated per call	10

H Complete Recoverability Results

This section presents complete recoverability results organized by side information level (K). For each K, we report results across the four primary benchmarks (Adult, German Credit, Heart Disease, Diabetes Pima) and three data regimes (Shadow, Few-Shot, Zero-Shot). Each table shows performance for varying numbers of sensitive features (S), with a mean row at the bottom for comparison with main paper figures.

Metrics: We report mean categorical accuracy (higher is better) for datasets with categorical features, and NMAE% (lower is better) for continuous-only features.

H.1 No-Context ($K = 0$)

In the no-context setting, no side information is available, and recoverability is measured solely from the embedding z with $x_{\text{pub}} = \emptyset$.

H.1.1 Accuracy Results at K=0

The following tables (Tables 17 to 19) report the mean categorical accuracy.

H.1.2 Continuous Feature Results (NMAE) at K=0

The following tables (Tables 20 to 23) report NMAE for continuous features in datasets with mixed feature types.

Key observations at K=0:

- **Cascade dramatically outperforms Joint**: On Adult, Cascade achieves 65% mean accuracy in Shadow setting while Joint only reaches 22%. On German Credit, Cascade (74%) significantly outperforms Joint (44%).
- **Consistent across all datasets**: Heart Disease shows similar patterns with Cascade achieving 79–89% vs Joint's 33–55%.
- **Zero-Shot maintains effectiveness**: Even without real training data, Cascade Probing achieves strong performance (88% on Heart Disease, 53% on Adult) using LLM-generated synthetic data.

Table 17: Adult dataset results with K=0 (zero known features).

S	Shadow		Few-Shot		Zero-Shot	
	Joint	Cascade	Joint	Cascade	Joint	Cascade
1	0.10	0.19	0.08	0.14	0.14	0.14
2	0.07	0.19	0.14	0.10	0.14	0.14
3	0.12	0.58	0.11	0.41	0.06	0.42
4	0.27	0.69	0.18	0.62	0.24	0.56
5	0.27	0.67	0.32	0.59	0.15	0.64
6	0.30	0.69	0.12	0.67	0.19	0.63
7	0.17	0.73	0.19	0.58	0.28	0.60
8	0.19	0.71	0.19	0.56	0.25	0.44
9	0.32	0.84	0.25	0.79	0.11	0.76
10	0.36	0.91	0.34	0.79	0.33	0.73
11	0.27	0.92	0.43	0.82	0.21	0.77
Mean	**0.22**	**0.65**	**0.21**	**0.55**	**0.19**	**0.53**

Mean categorical accuracy reported (higher is better).

Table 18: German Credit dataset results with K=0 (zero known features).

S	Shadow		Few-Shot		Zero-Shot	
	Joint	Cascade	Joint	Cascade	Joint	Cascade
2	0.36	0.36	0.36	0.36	0.36	0.26
3	0.46	0.76	0.46	0.44	0.21	0.52
4	0.40	0.80	0.34	0.82	0.36	0.54
5	0.51	0.78	0.33	0.58	0.22	0.50
6	0.42	0.80	0.45	0.70	0.12	0.61
7	0.41	0.78	0.27	0.49	0.26	0.60
8	0.32	0.78	0.32	0.51	0.19	0.58
9	0.39	0.67	0.32	0.64	0.17	0.41
10	0.41	0.75	0.20	0.54	0.15	0.45
11	0.37	0.75	0.33	0.68	0.25	0.43
12	0.47	0.74	0.48	0.65	0.21	0.44
13	0.49	0.73	0.24	0.51	0.21	0.43
14	0.45	0.72	0.22	0.53	0.12	0.42
15	0.41	0.75	0.22	0.64	0.13	0.45
16	0.46	0.74	0.49	0.67	0.24	0.47
17	0.56	0.73	0.21	0.64	0.21	0.49
18	0.53	0.73	0.33	0.60	0.21	0.46
Mean	**0.44**	**0.74**	**0.33**	**0.59**	**0.21**	**0.49**

Mean categorical accuracy reported (higher is better). $S = 1$ omitted (continuous only).

- **Cascade reduces NMAE significantly**: On Diabetes Pima, Cascade reduces NMAE from 0.15–0.22 to 0.08–0.09 across all settings. For continuous features in mixed datasets, Cascade achieves 3–6$\times$ lower NMAE than Joint on Adult (0.04 vs 0.13–0.15) and German Credit (0.07 vs 0.41–0.43).
- **Joint struggles without categorical context**: On German Credit at S=1–2 (pure continuous features), Joint NMAE exceeds 0.80, indicating near-random predictions. Cascade maintains robust performance (NMAE 0.02–0.16).

H.2 Single-Feature Context (K=1)

With a single known feature ($K = 1$), the recipient has minimal side information. This reveals the Privacy Cliff phenomenon: even a single known feature dramatically increases recoverability.

H.2.1 Accuracy Results at K=1

The following tables (Tables 24 to 26) report the mean categorical accuracy.

Table 19: Heart Disease dataset results with K=0 (zero known features).

	Shadow		Few-Shot		Zero-Shot	
S	Joint	Cascade	Joint	Cascade	Joint	Cascade
1	0.83	0.83	0.17	0.83	0.17	0.83
2	0.70	0.70	0.63	0.78	0.30	0.78
3	0.52	0.82	0.48	0.86	0.37	0.84
4	0.50	0.88	0.45	0.89	0.34	0.89
5	0.55	0.80	0.35	0.91	0.46	0.90
6	0.48	0.83	0.37	0.92	0.35	0.93
7	0.44	0.75	0.37	0.92	0.29	0.89
8	0.56	0.77	0.30	0.91	0.29	0.90
9	0.49	0.79	0.34	0.91	0.42	0.89
10	0.54	0.79	0.39	0.91	0.33	0.90
11	0.51	0.80	0.29	0.90	0.36	0.91
12	0.53	0.76	0.30	0.92	0.30	0.90
Mean	**0.55**	**0.79**	**0.37**	**0.89**	**0.33**	**0.88**

Mean categorical accuracy reported (higher is better).

Table 20: Adult dataset NMAE with K=0 (continuous features only).

	Shadow		Few-Shot		Zero-Shot	
S	Joint	Cascade	Joint	Cascade	Joint	Cascade
2	0.16	0.06	0.16	0.06	0.15	0.06
3	0.20	0.02	0.34	0.08	0.17	0.06
4	0.15	0.04	0.16	0.04	0.17	0.04
5	0.15	0.03	0.17	0.04	0.17	0.04
6	0.12	0.02	0.10	0.03	0.17	0.02
7	0.06	0.01	0.08	0.02	0.09	0.02
8	0.07	0.06	0.11	0.04	0.06	0.04
9	0.10	0.02	0.10	0.03	0.07	0.03
10	0.08	0.01	0.10	0.02	0.09	0.03
Mean	**0.12**	**0.03**	**0.15**	**0.04**	**0.13**	**0.04**

NMAE reported (lower is better). S=1 omitted (no continuous sensitive features).

Table 21: German Credit dataset NMAE with K=0 (continuous features only).

	Shadow		Few-Shot		Zero-Shot	
S	Joint	Cascade	Joint	Cascade	Joint	Cascade
1	0.11	0.10	0.84	0.16	0.82	0.14
2	0.11	0.04	0.83	0.02	0.98	0.02
3	0.10	0.02	0.13	0.02	0.29	0.02
4	0.11	0.02	0.51	0.01	0.25	0.02
5	0.18	0.03	0.47	0.07	0.37	0.04
6	0.11	0.03	0.25	0.02	0.40	0.04
7	0.14	0.03	0.37	0.10	0.44	0.03
8	0.21	0.08	0.19	0.12	0.29	0.12
9	0.23	0.09	0.21	0.09	0.21	0.16
10	0.17	0.08	0.32	0.10	0.20	0.14
Mean	**0.15**	**0.05**	**0.41**	**0.07**	**0.43**	**0.07**

NMAE reported (lower is better). Joint struggles at low S in Few-Shot/Zero-Shot without categorical context.

H.2.2 Continuous Feature Results (NMAE) at K=1

The following tables (Tables 27 to 30) report NMAE for continuous features in datasets with mixed feature types at K=1.

Key observations at K=1:

Table 22: Heart Disease dataset NMAE with K=0 (continuous features only).

	Shadow		Few-Shot		Zero-Shot	
S	Joint	Cascade	Joint	Cascade	Joint	Cascade
8	0.18	0.10	0.23	0.03	0.18	0.03
9	0.17	0.13	0.36	0.04	0.31	0.04
10	0.16	0.11	0.37	0.05	0.50	0.04
11	0.15	0.10	0.36	0.04	0.46	0.04
12	0.16	0.11	0.35	0.04	0.36	0.05
Mean	**0.16**	**0.11**	**0.34**	**0.04**	**0.36**	**0.04**

NMAE reported (lower is better).

Table 23: Diabetes Pima dataset NMAE with K=0 (all features are continuous).

	Shadow		Few-Shot		Zero-Shot	
S	Joint	Cascade	Joint	Cascade	Joint	Cascade
1	0.16	0.16	0.28	0.18	0.24	0.17
2	0.17	0.10	0.24	0.13	0.23	0.13
3	0.15	0.08	0.17	0.06	0.20	0.07
4	0.15	0.06	0.17	0.05	0.17	0.05
5	0.15	0.06	0.19	0.05	0.19	0.09
6	0.12	0.06	0.22	0.05	0.24	0.05
7	0.12	0.06	0.19	0.07	0.29	0.05
Mean	**0.15**	**0.08**	**0.21**	**0.08**	**0.22**	**0.09**

NMAE reported (lower is better).

Table 24: Adult dataset results with K=1 (single known feature).

	Shadow		Few-Shot		Zero-Shot	
S	Joint	Cascade	Joint	Cascade	Joint	Cascade
1	0.99	0.96	0.94	0.91	0.81	0.78
2	0.98	0.98	0.93	0.92	0.84	0.78
3	0.96	0.95	0.89	0.89	0.81	0.79
4	0.96	0.96	0.92	0.89	0.84	0.85
5	0.96	0.96	0.90	0.90	0.83	0.83
6	0.96	0.95	0.90	0.91	0.84	0.86
7	0.96	0.95	0.91	0.90	0.84	0.83
Mean	**0.97**	**0.96**	**0.91**	**0.90**	**0.83**	**0.82**

Mean categorical accuracy reported (higher is better).

Table 25: German Credit dataset results with K=1 (single known feature).

	Shadow		Few-Shot		Zero-Shot	
S	Joint	Cascade	Joint	Cascade	Joint	Cascade
2	1.00	0.96	1.00	1.00	0.99	1.00
3	0.84	0.78	0.86	0.83	0.72	0.67
4	0.84	0.77	0.84	0.86	0.67	0.62
5	0.85	0.79	0.83	0.83	0.71	0.64
6	0.81	0.80	0.83	0.86	0.72	0.72
7	0.81	0.79	0.85	0.84	0.71	0.68
Mean	**0.86**	**0.81**	**0.87**	**0.87**	**0.75**	**0.72**

Mean categorical accuracy reported (higher is better).

- **Dramatic improvement from K=0**: Joint Probing jumps from 19–55% to 75–97% accuracy with just one known feature, confirming the Privacy Cliff.
- **Both methods perform well**: With context, the gap between Joint and Cascade narrows—both achieve high accuracy.

Table 26: Heart Disease dataset results with K=1 (single known feature).

	Shadow		Few-Shot		Zero-Shot	
S	Joint	Cascade	Joint	Cascade	Joint	Cascade
1	0.90	0.90	0.97	1.00	0.97	0.97
2	0.85	0.83	0.90	0.97	0.80	0.88
3	0.91	0.83	0.87	0.92	0.87	0.88
4	0.88	0.85	0.73	0.81	0.72	0.79
5	0.85	0.81	0.78	0.82	0.73	0.82
6	0.87	0.87	0.84	0.86	0.82	0.86
7	0.81	0.79	0.79	0.84	0.72	0.84
8	0.83	0.78	0.68	0.83	0.78	0.84
9	0.84	0.76	0.80	0.85	0.79	0.87
10	0.82	0.80	0.80	0.83	0.73	0.86
11	0.80	0.80	0.80	0.83	0.77	0.83
12	0.82	0.78	0.79	0.84	0.68	0.84
Mean	**0.85**	**0.82**	**0.81**	**0.87**	**0.78**	**0.86**

Mean categorical accuracy reported (higher is better).

Table 27: Adult dataset NMAE with K=1 (continuous features only).

	Shadow		Few-Shot		Zero-Shot	
S	Joint	Cascade	Joint	Cascade	Joint	Cascade
2	0.02	0.01	0.03	0.02	0.03	0.03
3	0.02	0.01	0.03	0.02	0.04	0.03
4	0.02	0.01	0.03	0.02	0.04	0.03
5	0.03	0.01	0.03	0.02	0.04	0.03
6	0.01	0.01	0.02	0.01	0.03	0.01
7	0.01	0.00	0.02	0.01	0.02	0.01
Mean	**0.02**	**0.01**	**0.03**	**0.02**	**0.03**	**0.02**

NMAE reported (lower is better). S=1 omitted (no continuous sensitive features).

Table 28: German Credit dataset NMAE with K=1 (continuous features only).

	Shadow		Few-Shot		Zero-Shot	
S	Joint	Cascade	Joint	Cascade	Joint	Cascade
1	0.02	0.02	0.01	0.02	0.02	0.03
2	0.02	0.02	0.01	0.01	0.02	0.02
3	0.03	0.02	0.03	0.01	0.03	0.02
4	0.04	0.02	0.02	0.01	0.03	0.02
5	0.04	0.04	0.03	0.03	0.05	0.04
6	0.05	0.03	0.04	0.03	0.06	0.03
7	0.04	0.03	0.03	0.02	0.05	0.03
Mean	**0.03**	**0.03**	**0.03**	**0.02**	**0.04**	**0.03**

NMAE reported (lower is better).

- **Cascade maintains advantage in low-resource settings**: On Heart Disease Zero-Shot, Cascade (86%) outperforms Joint (78%) by 8 percentage points.

H.3 Per-Attribute Recovery

Tables 31 and 32 report per-attribute recovery in the Shadow setting at $K{=}0$ and $K{=}1$ across Adult, German Credit, Heart Disease, and Diabetes Pima. The Heart and Diabetes results use $S{=}6$ and are averaged over three seeds. Their revealed attributes at $K{=}1$ are sex and glucose, respectively.

Table 29: Heart Disease dataset NMAE with K=1 (continuous features only).

S	Shadow		Few-Shot		Zero-Shot	
	Joint	Cascade	Joint	Cascade	Joint	Cascade
8	0.09	0.12	0.12	0.05	0.13	0.05
9	0.13	0.13	0.16	0.07	0.16	0.07
10	0.12	0.11	0.13	0.07	0.18	0.07
11	0.12	0.11	0.11	0.07	0.12	0.08
12	0.12	0.11	0.13	0.07	0.13	0.08
Mean	**0.12**	**0.12**	**0.13**	**0.07**	**0.14**	**0.07**

NMAE reported (lower is better). S=1–7 omit continuous features in sensitive set.

Table 30: Diabetes Pima dataset NMAE with K=1 (all features are continuous).

S	Shadow		Few-Shot		Zero-Shot	
	Joint	Cascade	Joint	Cascade	Joint	Cascade
1	0.07	0.09	0.09	0.10	0.08	0.09
2	0.05	0.07	0.07	0.07	0.07	0.08
3	0.06	0.06	0.07	0.07	0.07	0.08
4	0.05	0.06	0.06	0.07	0.07	0.07
5	0.05	0.05	0.06	0.06	0.06	0.06
6	0.05	0.06	0.07	0.07	0.08	0.07
7	0.05	0.05	0.06	0.06	0.08	0.06
Mean	**0.05**	**0.06**	**0.07**	**0.07**	**0.07**	**0.07**

NMAE reported (lower is better).

Table 31: Per-attribute categorical recovery for Cascade Probing in the Shadow setting. Acc. is accuracy, Bal. is balanced accuracy, and F1 is macro-F1 (all higher is better). Classes gives the number of categories. Majority is the majority-class rate, providing an accuracy baseline.

			$K = 0$			$K = 1$		
Attribute	**Classes**	**Majority**	**Acc.**	**Bal.**	**F1**	**Acc.**	**Bal.**	**F1**
Adult occupation	15	0.18	0.50	0.43	0.41	0.93	0.86	0.86
Adult education	16	0.34	0.77	0.65	0.57	0.92	0.83	0.76
Adult marital status	7	0.45	0.62	0.38	0.33	0.96	0.85	0.85
Adult relationship	6	0.40	0.68	0.61	0.54	0.94	0.84	0.85
German checking account	4	0.38	0.98	0.98	0.97	0.98	0.98	0.98
German savings account	5	0.61	0.61	0.20	0.15	0.78	0.59	0.59
German credit history	5	0.55	0.80	0.60	0.61	0.82	0.64	0.64
German purpose	11	0.25	0.65	0.39	0.38	0.85	0.57	0.56
Heart fasting blood sugar	2	0.80	0.80	0.50	0.44	0.80	0.53	0.49
Heart thal	3	0.60	0.82	0.64	0.61	0.81	0.71	0.67
Heart chest pain type	4	0.52	0.87	0.77	0.76	0.84	0.72	0.71
Heart number of vessels	4	0.46	0.74	0.64	0.63	0.69	0.58	0.54
Heart exercise angina	2	0.63	0.78	0.74	0.75	0.88	0.85	0.86
Heart slope	3	0.54	0.81	0.66	0.61	0.81	0.60	0.60

I Target Normalization Ablation: Full Results

This section provides complete results for the target normalization ablation study across all four datasets (Table 33), complementing the German Credit results in Table 39.

These findings have important implications for practitioners:

- **Joint Probing requires careful preprocessing**: Target normalization is essential for balanced multi-task learning across heterogeneous feature types.

Table 32: Per-attribute NMAE (↓) for Cascade Probing on Diabetes Pima in the Shadow setting, with S=6 and three seeds. Glucose becomes public at K=1 and diabetes pedigree enters the sensitive set.

Attribute	$K = 0$	$K = 1$
Insulin	0.056	0.044
Glucose	0.023	Public
Age	0.040	0.031
Skin thickness	0.044	0.038
Blood pressure	0.051	0.038
Pregnancies	0.066	0.055
Diabetes pedigree	Not targeted	0.064

Table 33: Effect of target normalization on attribute inference across all datasets (K=3, S=3).

Dataset	**Method**	**Accuracy ↑**		**NMAE ↓**	
		With	Without	With	Without
Adult	Joint	0.883	0.185	0.033	0.016
	Cascade	0.888	0.628	0.021	0.017
German Credit	Joint	0.940	0.230	0.043	0.140
	Cascade	0.935	0.930	0.039	0.151
Heart Disease†	Joint	—	—	0.038	0.117
	Cascade	—	—	0.050	0.070
Diabetes†	Joint	—	—	0.042	0.104
	Cascade	—	—	0.045	0.037

†Only continuous sensitive features at S=3.

- **Cascade Probing avoids this trap**: independent per-feature heads prevent loss-domination, removing the need for normalization.

J Additional Analyses

J.1 Recovery Order Ablation

We compare four recovery orderings (Shadow, K=0, S=4): (1) **Greedy** (ours): select features by validation performance, (2) **Random**: average over 5 random permutations (± std), (3) **Reverse**: hard-to-easy, (4) **Exhaustive**: best of all $S!$ permutations (oracle upper bound).

Table 34: Recovery order ablation at K=0, S=4 (Shadow, Cascade).

Dataset	**Greedy**	**Random**	**Reverse**	**Exhaustive**
Heart (Acc ↑)	**0.880**	0.791 ±.032	0.746	0.896
Diabetes (NMAE ↓)	**0.060**	0.066 ±.003	0.071	0.048

The greedy heuristic is the best non-oracle strategy, achieving performance within 1–2% of the exhaustive oracle. Reverse (hard-to-easy) performs worst.

J.2 Decoder Architecture Ablation

We test 7 decoder architectures spanning a 292× parameter range to assess sensitivity to decoder capacity (Few-Shot, Cascade, K=0, averaged over S).

Recoverability is stable across small-to-medium decoders (49K–3.3M parameters). Even a 49K linear decoder achieves comparable accuracy to the 363K default MLP. Very large decoders (4.8M+) underperform, consistent with overfitting in the high-capacity regime. This confirms the leakage is an intrinsic property of the embeddings.

Table 35: Decoder architecture ablation (K=0, Few-Shot, Cascade).

Architecture	Params	Adult Acc ↑	German Acc ↑	Heart Acc ↑	Diabetes NMAE ↓
Linear	49K	.500	.630	.760	.067
MLP 2L	58K	.560	.610	.870	.074
MLP 3L (default)	363K	.550	.590	.890	.080
MLP 4L	1.1M	.520	.590	.880	.069
Transformer 4L	3.3M	.540	.600	.780	.063
Transformer 6L	4.8M	.420	.510	.740	.057
Transformer 8L	14.3M	.370	.360	.690	.114

J.3 Per-Feature Privacy Cliff Analysis

To test whether specific anchor features drive the Privacy Cliff, we use each feature as the sole known feature (K=1), probe all remaining features (S=N−1), and rank features by the recoverability they enable (Shadow, Joint).

Table 36: Best vs. worst anchor feature across datasets.

Dataset	Best Anchor	Score	Worst Anchor	Gap
Adult (Acc)	workclass	0.960	education	0.036
German (Acc)	property	0.800	foreign_worker	0.061
Heart (Acc)	resting_ecg	0.852	resting_bp	0.077
Diabetes (NMAE)	glucose	0.010	pregnancies	0.033

Even the worst anchor yields high recoverability (lowest categorical accuracy is 0.739 on German Credit), meaning revealing *any* single feature is far more dangerous than revealing none. The gap between best and worst anchor reaches 7.7pp on Heart Disease.

J.4 Iterative Refinement Baseline

We implement an iterative refinement (IR) baseline that predicts all features jointly, then refines over T rounds by feeding predictions back as context (Shadow, K=0).

Table 37: Iterative Refinement vs. Joint and Cascade Probing (K=0, Shadow).

Dataset	Joint	IR (T=3)	IR (T=5)	Cascade
Heart Disease (Acc ↑)	0.550	0.610	0.620	**0.790**
Diabetes Pima (NMAE ↓)	0.150	0.120	0.110	**0.080**

IR improves over Joint, confirming that iterative correction helps. Cascade still outperforms IR by a large margin, showing that sequential dependency modeling is more effective than simultaneous prediction with refinement.

J.5 L2 Normalization Robustness

We test whether L2 normalization of embeddings reduces recoverability (Few-Shot, Cascade Probing, K=0, averaged over S).

Table 38: Effect of L2 normalization on probing performance.

Dataset	Baseline	L2-Norm
Adult (Acc)	.550	.540
German (Acc)	.590	.590
Heart (Acc)	.890	.910
Diabetes (NMAE)	.080	.080

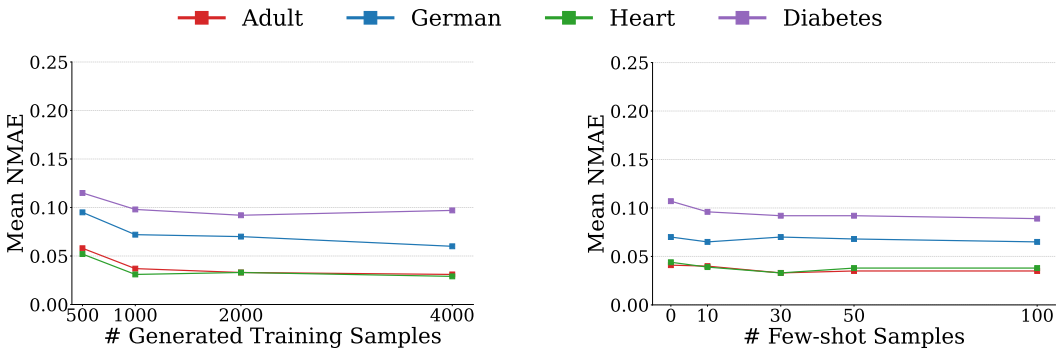
Table 39: Effect of target normalization on German Credit (K=3, S=3).

Method	Accuracy ↑		NMAE ↓	
	With	Without	With	Without
Joint	0.940	0.230	0.043	0.140
Cascade	0.935	0.930	0.039	0.151

L2 normalization does not reduce recoverability; performance is unchanged across all datasets.

J.6 LLM Data Efficiency

We study the efficiency of LLM-based shadow data generation by varying (i) the synthetic training budget N_{syn} and (ii) the few-shot prompting budget N_{shot} (K=0, Cascade, continuous features).



(a) Varying N_{syn}. (b) Varying N_{shot}.

Figure 6: LLM data efficiency (K=0, Cascade, mean NMAE on continuous features).

Increasing N_{syn} improves inversion with diminishing returns, flattening around 1000 samples. For the few-shot ablation (N_{syn}=2000), most improvement comes from the first few real exemplars.

J.7 Target Normalization Ablation

Target normalization is essential for Joint Probing. Table 39 shows that omitting normalization on German Credit collapses Joint categorical accuracy from 94.0% to 23.0%, while Cascade maintains robust accuracy (∼93%) even without normalization. Full results across all datasets are in Appendix I.

J.8 Recovery on High-Dimensional Datasets

We evaluate recovery of S=50 sensitive attributes on COIL2000 and Communities & Crime at K=0 in the Shadow setting. These experiments extend the evaluation to tables with 85 categorical and 100 continuous features, respectively. Unlike Table 1, which averages over sensitive-set sizes, Table 40 reports a fixed S=50.

Table 40: Recovery on high-dimensional datasets with S=50 sensitive targets at K=0 in the Shadow setting. COIL2000 reports categorical accuracy (↑) and Communities & Crime (C&C) reports NMAE (↓), averaged over the sensitive attributes. Best per row in bold.

Dataset (metric)	Joint	DirectOpt	WGAN	TabSyn	UIFV	Cascade
COIL2000 (Acc.)	0.674	0.346	0.411	0.443	0.430	**0.699**
C&C (NMAE)	0.064	0.145	0.108	0.099	0.094	**0.060**

Cascade achieves the strongest recovery on both datasets. Joint Probing is the closest baseline, with 0.674 accuracy on COIL2000 and 0.064 NMAE on Communities & Crime. The smaller gaps show that Cascade's advantage varies across datasets, while substantial recoverability persists with 50 sensitive targets. The cost of discovery and inference for these high-dimensional datasets is reported in Table 42.

J.9 Computational Overhead

Strategy Discovery is performed once for a fixed audit configuration. For S sensitive targets, it trains $S + (S-1) + \cdots + 1 = S(S+1)/2$ candidate probes. The selected probes and recovery order are retained and reused across records without retraining. We report wall-clock times on an NVIDIA A100. Table 41 reports the original German Credit experiment with 20 features and 2K training samples.

Table 41: Computational overhead of Strategy Discovery and per-record inference.

S	**Discovery time**	**Per-record inference**
1	5.5s	0.8ms
4	34.7s	1.8ms
10	3.7 min	3.9ms
20 (all)	12.0 min	7.2ms

Table 42 extends the timing evaluation to COIL2000 and Communities & Crime at $K{=}0$ in the Shadow setting. Discovery cost increases substantially with the number of sensitive targets, reaching 105.6 minutes on COIL2000 at $S{=}50$. This offline cost is amortized across records under the same audit configuration. Per-record inference grows approximately linearly over the measured settings and takes 13.1 ms on Communities & Crime and 36.9 ms on COIL2000 at $S{=}50$.

Table 42: Strategy Discovery and per-record inference times on high-dimensional datasets at $K{=}0$ in the Shadow setting, measured on an NVIDIA A100. Discovery is an offline cost for a fixed audit configuration.

	Communities & Crime		**COIL2000**	
S	Discovery (min)	Inference (ms/record)	Discovery (min)	Inference (ms/record)
10	1.0	2.6	4.5	7.6
30	6.0	7.8	37.7	22.7
50	17.1	13.1	105.6	36.9