Formal Privacy Proof of Data Encoding: The Possibility and Impossibility of Learnable Obfuscation

Hanshen Xiao
Purdue University / NVIDIA Research
West Lafayette, IN, USA
hsxiao.purdue@gmail.com

G. Edward Suh
NVIDIA Research / Cornell University
Westford, MA, USA
suh@ece.cornell.edu

Srinivas Devadas
Massachusetts Institute of Technology
Cambridge, MA, USA
devadas@mit.edu

ABSTRACT

We initiate a formal study on the concept of *learnable obfuscation* and aim to answer the following question: is there a type of data encoding that maintains the "learnability" of encoded samples, thereby enabling *direct* model training on transformed data, while ensuring the privacy of both plaintext and the secret encoding function? This long-standing open problem has prompted many efforts to design such an encryption function, for example, NeuraCrypt and TransNet. Nonetheless, all existing constructions are heuristic without formal privacy guarantees, and many successful reconstruction attacks are known on these constructions assuming an adversary with substantial prior knowledge.

We present both generic possibility and impossibility results pertaining to learnable obfuscation. On one hand, we demonstrate that any non-trivial, property-preserving transformation which enables effectively learning over encoded samples *cannot* offer cryptographic computational security in the worst case. On the other hand, from the lens of information-theoretical security, we devise a series of new tools to produce provable and useful privacy guarantees from a set of heuristic obfuscation methods, including matrix masking, data mixing and permutation, through *noise perturbation*. Under the framework of *PAC Privacy*, we show how to quantify the leakage from the learnable obfuscation built upon obfuscation and perturbation methods against adversarial inference. Significantly sharpened utility-privacy tradeoffs are achieved compared to state-of-the-art accounting methods when measuring privacy against data reconstruction and membership inference attacks.

CCS CONCEPTS

• **Security and privacy** → Privacy-preserving protocols.

KEYWORDS

learnable obfuscation; PAC privacy proof; membership inference; data reconstruction; matrix masking; data mixing; permutation.

1 INTRODUCTION

Over the past few decades, machine learning has experienced tremendous success in a wide range of applications, spanning from image classification and natural language processing to personalized recommendation. This impressive advancement, particularly in the development of deep learning, has been largely facilitated by access to expansive, representative datasets and the computational capabilities to train sophisticated models. As a result, concerns related to data privacy and implementation efficiency have come to the forefront, garnering significant attention from the security and cryptography communities. One critical challenge that has arisen in cloud and collaborative computing is privately releasing data and outsourcing a computationally-intensive training task to some untrusted server. This process is expected to maintain the privacy of both sensitive data and the resulting trained model, with low implementation and communication overhead.

From a broad perspective, there exist two principal applications in the realm of sensitive data publishing and learning. The first involves training a *public* model using private data: a user secretly converts her data into an encoded private version, from which a model is learned, whose utility or accuracy is defined as its ability to recognize the *original data distribution.* This scenario is what (Local) Differential Privacy ((L)DP) [18, 19] and Instahide [32] consider. For example, one may publish a set of distorted cat and dog images that appropriately preserve the privacy of each individual entity; when the noise is not huge, from the altered data, other people can still learn a classifier to recognize the unperturbed cat-dog pictures. It is known that in such a setup, no matter what kind of encoding protocol is applied, that data privacy must be traded off against the utility [10]. Similar arguments that "privacy cannot come free of utility loss" from a lens of (L)DP have also been presented [18, 20]. The obfuscation method introduced in Instahide that claims freedom from utility compromise has been broken [10], and its theoretical vulnerability has also been studied [12]. The intuition is that, if the model trained over the encoded data needs to work well for the original samples, the transform should ensure substantial similarity between the raw and encoded data. Besides such stringent restrictions, it also implicitly suggests that an adversary could have unlimited access to public data from the original sample domain [10], since the meanings of labels and the learning task are openly known to the adversary.

The second important application is to train a *secret* model from private data, which is the central focus of this paper. The second application is a special case of the former application discussed, with the inclusion of an extra secret key. A user first encodes their data with the secret key and sends the encoded data to an untrusted server. The model trained over encoded data is not necessarily known (possibly encrypted) to the server nor does it need to work for the original data; only the user who has the key can apply this model for meaningful prediction. The concept of *learnable obfuscation* that we set out to formalize belongs to this category.

CCS '24, October 14–18, 2024, Salt Lake City, UT, USA

ACM ISBN 979-8-4007-0636-3/24/10.
https://doi.org/10.1145/3658644.3670277

A traditional method to tackle this second application is based on Fully Homomorphic Encryption (FHE) [21]. FHE offers a universal framework for computing over encrypted data, where the operations can subsequently be converted back into plaintext. In the realm of learning, all training computations can be executed over encrypted data, while both intermediate computation and finally-calculated model are encrypted; only the user possessing the key can decrypt the computed encrypted model. Hence, in scenarios aiming to learn a secret model, FHE indicates that computational security does *not* necessarily conflict with model's utility or accuracy, since the user can ultimately acquire the same model as when they train locally. While theoretically this framework allows for any operation and numerous algorithmic advancements have been made to optimize performance, state-of-the-art FHE or even partially homomorphic encryption protocols still carry significant computational and communication overhead [40, 64], which precludes their deployment to train deep neural networks on medium- or large-scale datasets.

To overcome the inherent limitations of FHE, many heuristic approaches are being explored to design alternative solutions that can circumvent the costly oblivious computation: FHE programs, specified in terms of primitive instructions through homomorphism, are orders of magnitude slower than the corresponding native operations [24]. In particular for privately-outsourced machine learning, one ideal scenario is that the computation over encoded data is *not* oblivious to the learner/server: it allows one to train a model *directly* over encoded data using standard optimization methods, such as empirical risk minimization (ERM) via stochastic gradient descent (SGD), thereby approaching the computational overhead as low as that of a non-private machine learning. From the perspective of statistical learning, any dataset that sufficiently captures the key features of the underlying distributions can theoretically facilitate the creation of a useful model, and importantly, this is not confined to a specific dataset.

This leads to an interesting question. Can a user, through secret encoding, transform the original learning dataset into a new one, potentially in a different domain, satisfying the following: this new dataset enables statistical learning directly and of ideal utility; but it is hard for an adversary to relate the transformed dataset to the original one. To develop useful and lightweight learnable obfuscation, one natural idea is to randomly and *uniformly* transform the data domain based on some class of functions where certain topology or locality of datapoints is preserved. Most existing learnable obfuscation proposals follow this line and largely rely on (a combination of) the following three categories of obfuscations.

(a) **Matrix Masking (Random Linear Projection)**: As demonstrated by the well-known Johnson–Lindenstrauss (JL) lemma [37], a properly-selected random linear operator (e.g., a Gaussian matrix) can produce an efficient embedding for a set of points in a higher-dimensional space that nearly preserves their pairwise distance. Random linear projection also plays an important role in compressive sensing and locality-sensitive hashing [33]. Due to this nice statistical property and the simple implementation, random linear projection [41], also named matrix masking in the literature [16, 51, 66], and its variants, such as a multilayer perceptron with random weights, are viewed as ideal constructions for learnable obfuscation, and are adopted by empirical works such as TransNet [30], NeuraCrypt [62] and Syfer [63].

(b) **Data Mixing**: Rooted in *Mixup* [65], a successful data augmentation which considers training a network over mixed virtual samples, Instahide [32] presents the first attempt to obfuscate data by considering their random linear interpolation. To be formal, given a set of k randomly-selected samples $\{(x_1, y_1), (x_2, y_2), \cdots, (x_k, y_k)\}$, where x_i is the sample feature and y_i is the one-hot-vector label, a random weight $\boldsymbol{\lambda} = (\lambda_1, \cdots, \lambda_k)$ is generated where $\lambda_i \geq 0$ and $\sum_{i=1}^{k} \lambda_i = 1$. A virtual sample $(\tilde{x}, \tilde{y})$ to be released is constructed by $(\sum_{i=1}^{k} \lambda_i x_i, \sum_{i=1}^{k} \lambda_i y_i)$. The idea of data mixing also forms the foundation of other follow-up privacy-preserving protocols, such as DP-Instahide [7] and Datamix [42].

(c) **Permutation**:To obfuscate training data, another strategy is permutation. Permutation is usually free of utility compromise given that most learning algorithms are invariant to the ordering of training samples fed to the model. The authors of NeuraCrypt [62] claim a heuristic challenge that, when matrix masking and permutation are combined, an adversary who already knows the plaintexts cannot recover their correspondence to the encoded ciphertext.

We want to mention that data mixing and permutation, as two sample-wise operations, can be applied to both scenarios of learning secret and public models. Regrettably, the assurance of semantic privacy pertaining to *any* of the aforementioned obfuscation approaches remains largely unaddressed. In fact, *none* of these methods can produce input-independent security: under any (hybrid) of the obfuscation strategies described above, an adversary could distinguish the encodings between a dataset composed entirely of zeroes and an adjacent dataset, formed by zeroes except for one non-zero datapoint. To our knowledge, among these heuristic operations, only random linear projection is known to yield (weak) Differential Privacy (DP) guarantees after imposing strong regularization [6].

The absence of formal tools to quantify privacy leakage has emerged as a key impediment to research on learnable obfuscation or even broadly functional (property-preserving) data obfuscation beyond pure noise perturbation. Consequently, due to the lack of systematic investigations, the real privacy-preserving capacity of existing learnable obfuscation constructions can be, not surprisingly, overestimated. For instance, Carlini et al. construct a similarity-measure based attack which successfully addresses the identification challenge proposed in NeuraCrypt [11]. In light of the failures in existing learnable obfuscation proposals and juxtaposed with the only-known provably secure solution via FHE, a fundamental question arises: to achieve cryptographic security, *is it necessary to obscure computation over encoded data for a secure outsourced learning*? In particular, given the known impossibility result that privacy must be traded off against utility to learn a public model [10], in the special scenario where the user is allowed to use an additional secret key for a secret model, can a non-trivial encoding function theoretically achieve perfect privacy and utility, simultaneously? In this paper, we take a first step to formalize the concept of learnable obfuscation and set out to answer the following three key questions: a) what kind of security can one reasonably expect from a learnable obfuscation, b) how to design practical learnable obfuscation, and c) when can such an encoding provide meaningful provable privacy guarantees? Our contributions are summarized below.

(1) We formalize the concept of learnable obfuscation, and present a generic impossibility result to achieve computational security when a non-trivial model of good *encoded accuracy* can be directly trained from encoded data. This suggests that even if a secret key is allowed in learnable obfuscation, privacy still needs to be traded off against utility, similar to the public model scenario.

(2) We then develop a series of new tools to quantify the information leakage from three long-standing heuristic obfuscations: matrix masking, data mixing, and permutation. Using PAC Privacy [58] to determine appropriate *additive noise after heuristic obfuscation*, we show provable hardness against generic adversarial inference with a particular focus on reconstruction and membership attacks. Our results also provide intuitive explanations on how and why these obfuscations save privacy from the information theory perspective. After proper preprocessing, significantly sharpened utility-privacy tradeoffs are achievable compared to state-of-the-art accounting methods against data reconstruction [5] and membership inference attacks [48, 50]. These tradeoffs are further improved by adding *learnable noise.*

(3) We point out the potential applications of learnable obfuscation against reverse engineering attacks. Theoretical studies are presented under mild assumptions on data distribution.

2 PRELIMINARIES AND RELATED WORKS

2.1 PAC Learnability

Roughly speaking, we say two distributions D_0 and D_1 are learnable if given access to the sample pairs (x, y), where the feature x is independently drawn from $D_{0,1}$ with label $y \in \{0, 1\}$ to identify the source, one can find a model to classify *newly-incoming samples* as coming from D_0 or D_1; the sufficient advantage is usually called a small test error. In general, two distributions with a large statistical distance are not necessarily learnable, but learnable distributions must be distinguishable. To be formal, we introduce Probably Approximately Correct (PAC) learning theory [54] as follows.

Definition 1 (PAC Learnable). *Given datapoint universe $\mathcal{X}$, a concept class $C = \{h\colon \mathcal{X} \to \{0,1\}\}$, of target functions (classifiers) is PAC learnable if there exists a polynomial-time algorithm* Alg *which satisfies the following properties. Assume that $\mathcal{S} = \{(x_i, y_i), i = 1, 2, ..., n\}$ is a set of n samples where $x_{[1:n]}$ are i.i.d. generated by an arbitrary distribution* D *on $\mathcal{X}$, and $y_i = h(x_i)$ for some $h \in C$. Taking $\mathcal{S}$ as the input to* Alg, *for arbitrary $\epsilon > 0$ and $\delta \in (0, 1)$, there exists a function $m(\epsilon, \delta) : (0, 1) \times (0, 1) \to \mathbb{Z}^+$ such that once $n \geq m(\epsilon, \delta)$,* Alg(S) *will return a hypothesis $\hat{h} \in C$, with probability at least $1 - \delta$,* $\mathrm{Risk}(\hat{h}, h, \mathrm{D}) \leq \epsilon$. *Here,* $\mathrm{Risk}(\hat{h}, h, \mathrm{D}) = \mathrm{Pr}_{x \sim \mathrm{D}}(\hat{h}(x) \neq h(x))$ *denotes the test error on distribution* D, *for x randomly sampled from* D.

PAC learnability basically states that when the data is generated via some function h belonging to a set C, once we have enough samples/observations, there exists an efficient learning algorithm Alg to return some $\hat{h}$ with high probability such that the test error $\mathrm{Risk}(\hat{h}, h, \mathrm{D})$ of $\hat{h}$ is small. In practice, especially in deep learning, we usually consider applying a neural network architecture to approximate the concept class C, where learning $\hat{h}$ becomes optimizing the parameters/weights of the neural network.

2.2 Security and Privacy Definitions

Definition 2 (Computational Security against Chosen–Plaintext-Attack). *For an encoding function $\mathcal{F}(\cdot, sk) : \mathcal{X}^* \to \mathcal{O}$, where $sk \in \mathcal{K}$ is some secret key, we call $\mathcal{F}$ computationally secure against Indistinguishability under Chosen-Plaintext-Attack (IND-CPA) if the following experiment is impossible.*

A polynomial-time adversary selects two arbitrary plaintexts X_0 and X_1 from $\mathcal{X}^*$ and sends to the user. The user randomly selects $b \in \{0, 1\}$ and a secret key $sk \in \mathcal{K}$, and sends $\mathcal{F}(X_b, sk)$ back to the adversary. After observing $\mathcal{F}(X_b, sk)$, the adversary can return a guess $\hat{b}$ on b with non-negligible advantage, i.e., $\Pr(\hat{b} = b) \geq \frac{1}{2} + \delta$ *for some non-negligible δ in the security parameter.*

Computational security resistant to Indistinguishability under Chosen-Plaintext-Attack (IND-CPA) states that the encoding function satisfies input-independent indistinguishability for a computationally-bounded adversary. Rooted in the same idea of input-independent indistinguishability, DP offers an information-theoretical way to define the individual privacy leakage by measuring the divergence between the likelihoods produced by two adjacent datasets in the worst case.

Definition 3 ((ϵ, δ) Differential Privacy). *Given dataset universe $\mathcal{X}^*$, we say that two datasets $\mathcal{S}, \mathcal{S}' \subseteq \mathcal{X}^*$ are adjacent, denoted as $\mathcal{S} \sim \mathcal{S}'$, if $\mathcal{S} = \mathcal{S}' \cup s$ or $\mathcal{S}' = \mathcal{S} \cup s$ for some additional datapoint s. A randomized processing function $\mathcal{F}$ is said to be (ϵ, δ)-differentially-private (DP) if for any pair of adjacent datasets $\mathcal{S}, \mathcal{S}'$ and any set o in the output space $\mathcal{O}$ of $\mathcal{F}$, it holds that* $\Pr(\mathcal{F}(\mathcal{S}) \in o) \leq e^{\epsilon} \cdot \Pr(\mathcal{F}(\mathcal{S}') \in o) + \delta$.

A semantic interpretation of DP is from a hypothesis testing perspective, where small values of ϵ and δ suggest that either Type I or Type II error should be large [57]. However, the application of DP guarantees to tightly depict more generic inference hardness such as membership inference under arbitrary subsampling rate [50] or reconstruction attacks, especially when the reconstruction objective involves interplay between multiple datapoints [25], is challenging. Moreover, as a worst-case input-independent guarantee, the privacy proof of DP can be, in general, NP-hard [61]. Tight DP analyses for many examples of practical data processing are intractable, such as deep learning [52, 56, 59, 60] (especially over graphs [46]). Given limited tools, the privacy implication from many types of randomness remains open in DP. Indeed, it is not hard to observe that *none* of the previously mentioned heuristic obfuscations *themselves* can produce meaningful DP guarantees and it is also not clear that even combined with additional noise mechanisms, what kind of privacy amplification can be produced.

To produce tighter instance-based privacy analysis on possibly black-box processing functions, a generic framework, termed *PAC Privacy,* was proposed [58]. From a statistical inference standpoint, PAC Privacy describes the information leakage as follows. Given input data from some distribution and for an arbitrary inference task, one can define the optimal *a priori* success rate $(1-\delta_o)$ *before* observing the released output, and the optimal *posterior* success rate $(1-\delta)$ thereafter that an adversary can return a satisfied estimation. PAC Privacy presents an automated analysis framework to quantify the f-divergence [45] between two Bernoulli distributions of parameters δ and δ_o, respectively. Such difference (posterior advantage) can then produce a lower bound on the posterior failure rate δ given δ_o. A formal definition is given below.

Definition 4 ($(\delta, \rho, \mathsf{D})$ PAC Privacy [58]). *For a processing function $\mathcal{F}: \mathcal{X}^* \to \mathcal{O}$, some data distribution D, and an inference criterion function $\rho(\cdot,\cdot)$, we say $\mathcal{F}$ satisfies $(\delta, \rho, \mathsf{D})$-PAC Privacy if the following experiment is impossible:*

A user generates data X from distribution D and sends $\mathcal{F}(X)$ to an adversary. An informed adversary who knows D and $\mathcal{F}$ is asked to return an estimation $\hat{X}$ on X such that with probability at least $(1-\delta)$, $\rho(\hat{X}, X) = 1$.

Equivalently, $\mathcal{F}$ can be defined as $(\Delta_f\delta, \rho, \mathsf{D})$ PAC-advantage private if the posterior advantage measured in f-divergence satisfies

$$\Delta_f\delta = \mathcal{D}_f(\mathbf{1}_\delta \| \mathbf{1}_{\delta_o^\rho}) = \delta_o^\rho f(\frac{\delta}{\delta_o^\rho}) + (1-\delta_o^\rho) f(\frac{1-\delta}{1-\delta_o^\rho}),$$

where $(1-\delta_o^\rho)$ represents the optimal prior success rate, i.e., $1-\delta_o^\rho = \arg\max_{X' \in \mathcal{X}^*} \Pr_{X\sim\mathsf{D}}(\rho(X', X) = 1)$, and $\mathbf{1}_\delta$ and $\mathbf{1}_{\delta_o^\rho}$ represent two Bernoulli distributions of parameters δ and δ_o^ρ, respectively. Here, $\mathcal{D}_f$ is some f-divergence.

The criterion function ρ can be arbitrarily selected depending on which aspect of sensitive information we want to protect or make hard to infer. For example, if our privacy concern is about all bits of a password X, we may define $\rho(\hat{X}, X) = 1$, representing the adversarial success in the inference task, iff the adversary's estimation $\hat{X}$ collides in at least one bit of X; Similarly, for the sensitive input X, if we think a total reconstruction error larger than ψ in an l_p-norm is safe, then $\rho(\hat{X}, X) = 1$ iff $\|\hat{X} - X\|_p \leq \psi$.

Theorem 1 of [58] presents a way to control the posterior advantage $\Delta_f\delta$ in Definition 4, where

$$\Delta_f\delta = \mathcal{D}_f(\mathbf{1}_\delta \| \mathbf{1}_{\delta_o^\rho}) \leq \inf_{\mathsf{P}_W} \mathcal{D}_f(\mathsf{P}_{X,\mathcal{F}(X)} \| \mathsf{P}_X \otimes \mathsf{P}_W). \quad (1)$$

Here, $\mathsf{P}_{X,\mathcal{F}(X)}$ and P_X are the joint distribution and the marginal distribution of $(X, \mathcal{F}(X))$ and X, respectively; P_W represents the distribution of an arbitrary random variable W in the output domain $\mathcal{O}$ of $\mathcal{F}$. In particular, when we select $\mathcal{D}_f$ to be the KL-divergence and $\mathsf{P}_W = \mathsf{P}_{\mathcal{F}(X)}$,

$$\Delta_{KL}\delta = \mathcal{D}_{KL}(\mathbf{1}_\delta \| \mathbf{1}_{\delta_o^\rho}) \leq \mathsf{MI}(X; \mathcal{F}(X)), \quad (2)$$

where $\mathsf{MI}(\cdot,\cdot)$ represents *mutual information* and $\mathcal{D}_{KL}(\mathbf{1}_\delta \| \mathbf{1}_{\delta_o^\rho}) = \delta \log(\frac{\delta}{\delta_o^\rho}) + (1-\delta)\log(\frac{1-\delta}{1-\delta_o^\rho})$. Throughout the paper, *log* stands for natural logarithm. Thus, (2) develops a generic way to connect the hardness of arbitrary inference to the well-known mutual information.

The result in (2) also bridges DP guarantees to arbitrary inference hardness. From [8], if $\mathcal{F}$ satisfies ϵ-DP, then

$$\mathsf{MI}(X; \mathcal{F}(X)) \leq 0.5\epsilon^2 n^2,$$

if X is formed by n datapoints. However, black-box PAC Privacy analysis in many applications can produce much sharpened utility-privacy tradeoffs compared to existing input-independent worst-case analysis and can also present formal proofs of empirical verification on algorithmic robustness against data reconstruction and membership inference attacks [5, 48].

Though [58] proposes a theoretical automatic solution to privatize any processing, the overhead required for analyzing randomized functions could be very expensive, though polynomial in d and $|\Theta|$, $O(poly(d, |\Theta|))$, where d is the output dimension of objective processing $\mathcal{F}$ and $|\Theta|$ is the total number of the random seeds. To this end, we present sharpened, explainable and easily-simulatable bounds in this paper to study random data obfuscation methods.

2.3 Heuristic Obfuscation Operators

In this subsection, we formally define the three main heuristic data obfuscations that are commonly considered in existing learnable obfuscation constructions. For notional clarity, in the following we assume the input dataset X is an $n \times d_o$ real-number matrix, where each row represents an individual d_o-dimensional datapoint.

Matrix Masking: Matrix Masking is an obfuscation function defined as $\mathcal{F}(X) = XW$, where $W \in \mathbb{R}^{d_0 \times d}$ is some random matrix. There are also many variants which take the random linear operator as the building block. For example, NeuraCrypt [62] and TransNet [30] consider the transform function to be an L-layer neural network with random weights, where

$$\mathcal{F}(X) = \sigma_L(\sigma_{L-1}(\cdots\sigma_1(XW_1)\cdots W_{L-1})W_L).$$

Here, σ_l represents the activation function of the l-th layer.

As for its potential security guarantees, we have two remarks. First, essentially we apply an identical transform (same secret key) on each datapoint (each row of X), and this is different from the classic one-time-padding (independent randomness) on each plaintext. Indeed, if we apply independent linear transforms on each datapoint, a much stronger security guarantee is achievable. For example, if we consider an $\mathcal{F}$ where $\mathcal{F}(X) = \mathcal{F}(x_1, \cdots, x_n) = (x_1W_1, \cdots, x_nW_n)$ and $W_{[1:n]}$ are i.i.d. Gaussian matrices, it is not hard to observe that $\mathcal{F}$ only reveals the l_2-norm of each x_i. However, if we independently transform each datapoint, the produced transformed data could be very hard or even impossible to learn (imagine a normalized dataset in the above example).

Second, matrix masking is an operator in the real space. This is different from the classic *learning with errors* (LWE) problem in cryptography, where random linear projection combined with small perturbation could make the inversion computationally hard. The hardness of LWE comes from a restriction to some integer ring. Though recent work has generalized LWE to continuous LWE (mod 1) [26], the modulo operation is necessary in existing hardness proofs. As discussed later in Section 8, the requirement to preserve the learnability of encoded data essentially makes the reduction to standard computational indistinguishability impossible for learnable obfuscation.

Data Mixing: Data mixing can be viewed as another special linear operation (left multiplication) on the rows of X, compared to matrix masking (right multiplication) on the columns. To be formal, data mixing can be defined as an encoding function $\mathcal{F}(X) = MX$, where $M \in \mathbb{R}^{m \times n}$ is a positive random matrix where each row sums to 1.

For example, Instahide [32] considers k-mixing which restricts the Hamming weight of each row of M to equal k, i.e., each produced virtual sample is formed by k datapoints.

Permutation: Permutation is another special linear operation on the rows of X. It can be described as $\mathcal{F}(X) = \Pi X$, where Π is a random permutation unitary matrix that swaps the rows of X. Intuitively, the privacy implication from permutation is weakest; from either the perspective of computational or information-theoretical security, permutation itself cannot hide anything meaningful. However, in the context of data obfuscation for learning, it is generally free of utility compromise and its privacy amplification combined with other randomness largely remains an important open problem.

3 LEARNABLE OBFUSCATION

3.1 Definition of learnable obfuscation

With the PAC-learnable definition in Section 2.1, we develop a formal definition of learnable obfuscation below.

Definition 5 ($(C, \bar{m}_{[1:3]}, \epsilon, \rho_{(1,2)}, \delta)$ learnable obfuscation). *Given a PAC learnable concept class* C *whose input domain is* $\mathcal{X}$*, a learnable obfuscation protocol is expressed by* $\big(T(\cdot,\theta), \mathsf{Alg}\big)$*, where* $T(\cdot,\theta) = \big(T_X(\cdot,\theta_X), T_Y(\cdot,\theta_Y)\big)$*, with random seed* $\theta = (\theta_X, \theta_Y)$*, is a pair of generic randomized transformations, and* T_Y *is restricted to be injective; given* θ*,* $T(\cdot,\theta) : (\mathcal{X},\mathcal{Y}) \rightarrow (\mathcal{X}_T, \mathcal{Y}_T)$ *is deterministic and transforms a sample with feature* $x \in \mathcal{X}$ *and label* $y \in \mathcal{Y}$*, to* $(T_X(x), T_Y(y)) \in (\mathcal{X}_T, \mathcal{Y}_T)$*;* Alg *represents a polynomial-time learning algorithm.* $\big(T(\cdot,\theta), \mathsf{Alg}\big)$ *satisfies two properties:*

(1) ***Learnability Preservation****: For any* $h \in C$ *and any distribution* D*, a user generates* $X = \{x_1, x_2, ..., x_n\}$ *where* x_i *is i.i.d. generated from* D *and constructs a dataset* $\mathcal{S} = \{(x_i, y_i), i = 1, 2, ..., n\}$ *of* n *samples, where* $y_i = h(x_i)$*. The user then generates a random seed* $\theta = (\theta_X, \theta_Y)$ *and transforms the sample set* $\mathcal{S}$ *into a set* $\mathcal{S}_T = \{\big(T_X(x_i,\theta_X), T_Y(y_i,\theta_Y)\big), i = 1, 2, ..., n\}$ *as input to* Alg*. Once* $n \geq \bar{m}_1(\epsilon,\delta)$ *for some function* $m_1(\epsilon,\delta)$*,* $\mathsf{Alg}(\mathcal{S}_T)$ *can return some function* $\hat{h}$ *with probability at least* $1-\delta$ *such that* $\mathsf{Risk}(\hat{h}, T_Y \circ h, T_X(\mathsf{D})) = \Pr_{x\sim\mathsf{D}}[\hat{h}(T_X(x,\theta_X)) \neq T_Y(h(x),\theta_Y)] \leq \epsilon$*.*

(2) For any concept $h \in C$*, any distribution* D*, and the given inference criterion* ρ_1 *and* ρ_2*, there exist functions* $m_2(\rho_1,\delta)$ *and* $m_3(\rho_2,\delta)$ *such that the following two experiments are both impossible: A user generates a dataset* $\mathcal{S} = \{(x_i, y_i), i = 1, 2, ..., n\}$ *of* n *samples where* x_i *is i.i.d. from* D *and* $y_i = h(x_i)$*, and selects a random seed* (θ_X, θ_Y)*, and transforms* $\mathcal{S}$ *into* $\mathcal{S}_T = \{(T_X(x_i,\theta_X), T_Y(y_i,\theta_Y)), i = 1, 2, ..., n\}$*.* $\mathcal{S}_T$ *is sent to an adversary.*

*(a) (****Data Privacy****) The adversary is asked to return an estimation* $\tilde{S}$ *on the input* S*. When* $n \leq \bar{m}_2(\rho_1,\delta)$*, with probability at least* $1-\delta$*,* $\rho_1(\tilde{S}, S) = 1$*.*

*(b) (****Model Security****) The adversary is asked to return an estimation* $\tilde{T}$ *on the transform* $T(\cdot,\theta)$*. When* $n \leq \bar{m}_3(\rho_2,\delta)$*, with probability at least* $1-\delta$*,* $\rho_2(\tilde{T}, T(\cdot,\theta)) = 1$*.*

We want to mention that the label encoding is *not* necessary for a learnable obfuscation, and one can simply generalize Definition 5 to the unsupervised learning setup by only encoding the features/instances x, and all our following results still apply. The requirements of a learnable obfuscation are mainly twofold: utility and privacy. First, after applying some randomized transform function $T(\cdot,\theta)$ on both the features and labels, the transformed data is still PAC-learnable, which allows one to find or approximate a new classifier $\hat{h}$ over the transformed domain by directly looking at encoded data $(T_X(x_i), T_Y(y_i))$ in Alg. The returned model $\hat{h}$ can achieve good *encoded test performance*, captured by $\mathsf{Risk}(\hat{h}, T_Y \circ h, T_X(\mathsf{D}))$. In general, when a decryption is not needed – unlabeled samples can be transformed prior to running inference on the encoded model – the transformation function T_X is not necessarily injective. However, we restrict the label encoding function T_Y to be injective to ensure the transformed learning problem is not trivial. Besides, we always assume the data distribution D of interest satisfies $\mathbb{E}_{x\sim D} h(x) = 1/|\mathcal{Y}|$ for a $|\mathcal{Y}|$-classification problem to avoid trivially distinguishing two data distributions from their labels.

Second, on the privacy side, we introduce two criteria ρ_1 and ρ_2, which capture the adversarial inference with respect to the sensitive input data X and the transform function T, respectively. Once the amount of released data, captured by the threshold functions $\bar{m}_2(\rho_1,\delta)$ and $\bar{m}_3(\rho_2,\delta)$, is limited, the adversary cannot produce a satisfied estimation with high probability $(1-\delta)$. Model security, such that the transform function T itself is also hard to learn, implies provable hardness of *reverse engineering*. In outsourcing with learnable obfuscation, the untrusted server has direct access to the model trained over transformed data. However, if the adversary cannot efficiently approximate the transform T, the model which recognizes transformed data well cannot be applied to the original sample domain.

Finally, we must stress that FHE is *not* a special case of learnable obfuscation. The IND-CPA requirement suggests that FHE cannot satisfy the "(1) Learnability Preservation" requirement in Definition 5: no efficient learning algorithm can recognize a set of computationally-indistinguishable ciphertexts and a polynomial-time algorithm which achieves non-trivial classification accuracy over computationally-indistinguishable ciphertext (encoded samples) without knowing the secret key is a successful attack on FHE. As a concrete example, consider a classification task of labeling images. One set of images is easily classifiable. The other has random labels, and is theoretically impossible to classify. Both sets are FHE-encrypted and sent to the server. Learnability preservation requires that the classifiable set be determined by the server to have (non-negligibly) better encoded accuracy than the random set; this corresponds to a CPA to break the FHE scheme.

3.2 Adversarial Inference

In the following, we formally introduce several adversarial inference problems widely-considered in previous works.

For Input Data Privacy: l_p-norm reconstruction can be defined as $\rho_1(\hat{X}, X) = 1$ only when $\|\hat{X} - X\|_p \leq \psi$ for some constant ψ. The particular l_2-norm case has been studied in [25, 27], where Fisher information is used to produce a reconstruction lower bound for a *bias-specified adversary*, with recent generalization to the Bayesian setup [43]. In the following, we formally define the reconstruction challenge with estimation error measured in l_2-norm.

Definition 6 (Data Reconstruction Challenge). *Given a finite data pool* U *and some processing mechanism* $\mathcal{M}$*, let* $\mathcal{S}$ $=$

$(s_1, s_2, ..., s_N)$ be a set where each datapoint s_i is i.i.d. uniformly selected from U*, and the adversary is asked to return an $\hat{\mathcal{S}}$ on $\mathcal{S}$ after observing $\mathcal{M}(\mathcal{S})$. We say $\mathcal{M}$ satisfies a $(1-\delta, r)$ reconstruction challenge if, for an arbitrary adversary, their success rate to return an estimation with l_2-norm error smaller than r is bounded by $(1-\delta)$, i.e.,* $\Pr_{\mathcal{S}\leftarrow \mathsf{U}, \hat{\mathcal{S}}\leftarrow \mathcal{M}(\mathcal{S})}(\|\mathcal{S}-\hat{\mathcal{S}}\|_2 < r) \leq (1-\delta)$*. In particular, we say $\mathcal{M}$ satisfies a $(1-\delta, r)$ individual reconstruction challenge if we instead ask the adversary to only recover a single s_i datapoint by $\hat{s}$ and* $\Pr_{\mathcal{S}\leftarrow \mathsf{U}, \hat{s}\leftarrow \mathcal{M}(\mathcal{S})}(\|s_i-\hat{s}\|_2 < r) \leq (1-\delta)$.

The individual reconstruction setup in Definition 6 is adopted in many reconstruction robust (ReRo) studies [5, 28]. In particular, [5] shows that if $\mathcal{M}$ satisfies (α, ϵ)-Rényi Differential Privacy (RDP), then the posterior success rate $(1-\delta)$ is upper bounded by

$$1-\delta \leq \left((1-\delta_o)\cdot e^{\epsilon}\right)^{\frac{\alpha-1}{\alpha}}, \tag{3}$$

where $(1-\delta_o)$ is the optimal prior success rate for the adversary to return a satisfied estimation with l_2-norm error smaller than r.

Another important and widely-studied challenge is membership inference [3, 48]. When the data $X = \{x_1, \cdots, x_n\}$ is randomly sampled from a finite set/universe $\mathsf{U} = \{u_1, u_2, \cdots, u_N\}$, a natural question is whether the adversary can identify the participants of X from U. From both an average and a particular individual angle, we formally define membership inference as follows.

Definition 7 (Membership Inference Challenge). *Given a finite data pool $\mathsf{U} = \{u_1, u_2, \cdots, u_N\}$ and some processing mechanism $\mathcal{M}$, X is an n-subset of U randomly selected. The adversary is asked to return an n-subset $\hat{X}$ as the membership estimation of X after observing $\mathcal{M}(X)$. We say $\mathcal{M}(X)$ satisfies a $(1-\bar{\delta})$ average-membership-inference challenge, if for an arbitrary adversary, $\mathbb{E}_X[|X\cap\hat{X}|] \leq n(1-\bar{\delta})$, i.e., on average an adversary cannot identify more than $n(1-\bar{\delta})$ elements of X.*

As for each individual, we say $\mathcal{M}$ is resistant to $(1-\delta_i)$ individual membership inference for the i-th datapoint u_i, if for an arbitrary adversary, $\Pr(\mathbf{1}_{u_i\in X} = \mathbf{1}_{u_i\in\hat{X}}) \leq 1-\delta_i$*. Here, $\mathbf{1}_{u_i\in X}$ ($\mathbf{1}_{u_i\in\hat{X}}$) is an indicator which equals 1 if u_i is in X ($\hat{X}$).*

The individual membership inference challenge in Definition 7 is also widely adopted in many empirical privacy verification or auditing works, such as membership inference attack (MIA) [9, 48].

For Model Security: As previously mentioned, to preserve the intellectual property of the data holder, we also expect learnable obfuscation to resist reverse engineering where the server cannot apply the model trained from transformed data to do meaningful inference on original data. Thus, we consider the following challenge to approximate the learnable obfuscation function itself.

Definition 8 ((ψ, τ) Statistical Encoding Distance). *In the same setup as Definition 5 where the data feature x_i is i.i.d. in distribution* D*, the adversary proposes a function $\tilde{T}_X$ as an estimation of $T_X(\cdot, \theta_X)$ where $\rho_2(\tilde{T}_X, T_X(\cdot, \theta_X)) = 1$ if* $\Pr_{x\sim\mathsf{D}}\left(\|\tilde{T}_X(x) - T_X(x, \theta_X)\|_2 < \psi\right) \geq 1-\tau$.

For sufficiently large values of ψ and τ, a failure of matching Definition 8 implies that the adversary will be unable to effectively transform the original data feature to the target domain using their approximate transform: for a large τ fraction of data to infer, there is an encoded error larger than ψ in the adversarially-constructed (approximately transformed) data when applying the model.

3.3 A Construction

We combine the three main heuristic obfuscations, matrix masking, data mixing and permutations together, along with generic perturbations, and formally describe a construction of learnable obfuscation as Algorithm 1. As a concrete example, Algorithm 1 also depicts the workflow to apply learnable obfuscation. At a high level, the process involves the following steps. First, the user encodes the data using the obfuscations on both the features and labels of the data, then *adds noise to the obfuscated features*, resulting in an encoded version $\mathcal{S}_T$. This encoded data is then sent to an untrusted server, along with the specification of the network architecture to train. Second, the server proceeds to train over the encoded data $\mathcal{S}_T$ and returns a model $\mathcal{N}_T$ to the user. Finally, to predict some newly incoming feature x_q, the user applies $\mathcal{N}_T$ on the transformed version of x_q and decodes the outputted label.

While we adopt PAC learning theory to formally define learnability and learnable obfuscation, in practical scenarios, we do not always have a clear picture of the underlying concept set assumed in a PAC model. Consequently, it is not feasible to find a hypothesis by distinguishing every concept candidate. State-of-the-art machine learning methods often rely on selecting appropriate neural networks depending on the data types and optimizing their parameters to approach the true concept. For instance, the Convolutional Neural Network (CNN) [38] has proven effective in image processing, while the Transformer network [55] excels in Natural Language Processing (NLP). Thus, when designing practical learnable obfuscation, one needs to also take the training method into account and find a suitable network capable of efficiently handling the transformed learning task. In Algorithm 1, we focus on matrix masking on the entire feature domain, which is generally well-suited for training fully-connected networks. However, in more complex tasks involving image or NLP data, it becomes important to preserve certain internal locality of nearby pixels or segments. Besides, more powerful network architectures like Transformer networks may be necessary to achieve better performance. To accommodate advanced data structures and network architectures, Algorithm 1 can be generalized by splitting the entire feature representation into multiple blocks and applying independent matrix masking on each one. Similar ideas have been explored in [62].

We stress that Algorithm 1 is *not equivalent* to previous heuristic proposals, such as Instahide and NeuraCrypt, as we always consider an *additional noise perturbation B*. As detailed in Section 7.1, all our usable provable guarantees require sufficient noise B; without proper preprocessing and perturbation, we cannot provide meaningful privacy guarantees for these existing noise-free proposals [30, 32, 62]. One of our key contributions is to determine how much noise can be *saved* with additional data obfuscation.

4 BARRIER TO LEARNABLE OBFUSCATION

In this section, we show the generic barrier to a learnable obfuscation with computational security if it also enables one to efficiently find a model to classify encoded data.

Algorithm 1 A Framework for Constructions and Applications of Learnable Obfuscation

```
Input: A data pool $\mathsf{U} = \{u_1, \cdots, u_N\}$ with associated labels $\mathsf{V} = \{v_1, \cdots, v_N\}$; a random training dataset of features $X = \{x_1, \cdots, x_n\}$, $x_i \in \mathbb{R}^{d_0}$ of $n$ elements with corresponding labels $Y = \{y_1, y_2, \cdots, y_n\}$, where $y_i \in \{\mathbf{1}_1, \mathbf{1}_2, \cdots, \mathbf{1}_c\}$ for $c$ classes and $\mathbf{1}_l$ represents a $c$-dimensional one-hot vector with non-zero entry $l$; masking matrix distribution $\mathsf{P}_W$; mixing matrix distribution $\mathsf{P}_M$; perturbation distribution $\mathsf{P}_B$; output dimension $d$; number of mixed data $m$, and an neural network model $\mathcal{N}$.
                Phase 1 - Learnable Obfuscation Encoding
Generate a random masking matrix $W \in \mathbb{R}^{d_0 \times d}$ from $\mathsf{P}_W$, $M \in \mathbb{R}^{m \times n}$ from $\mathsf{P}_M$, a random permutation matrix $\Pi_1 \in \mathbb{R}^{m \times m}$, a permutation matrix $\Pi_2 \in \mathbb{R}^{c \times c}$ and noise $B$ from $\mathsf{P}_B$.
$T_X(X) = \Pi_1 M X W + B$, and $T_Y(Y) = \Pi_1 M Y \Pi_2$.
                Phase 2 - Private Outsourcing Training
User sends the transformed version of training sample set $\mathcal{S}_T = \{T_X(X), T_Y(Y)\}$ and the network structure of $\mathcal{N}$ to a server.
The server trains the network $\mathcal{N}$ over $\mathcal{S}_T$, and sends the trained model $\mathcal{N}_T$ back to the user.
                Phase 3 - Application for Inference
User can predict incoming data $x_q$ using the model $\mathcal{N}_T$ by first encoding $x_q$ to $x_q W$ and returning the output $\mathcal{N}_T(x_q W) \cdot \Pi_2^{-1}$.
```

4.1 Intuition

Learnable obfuscation enjoys much freedom in selecting obfuscation to preserve learnability, which can make the meaning of the transformed task totally oblivious to the adversary. However, as an efficiency tradeoff compared to FHE, we allow the adversary to have direct access to the trained model $\mathcal{N}_T$. This theoretically enables the adversary to *approximately estimate the encoded accuracy*, $\mathsf{Risk}(\hat{h}, T_Y \circ h, T_X(\mathsf{D}))$, even if he does not have additional knowledge: given the encoded data, the adversary can always randomly split it into two parts, one for training and one for test. As elaborated later in Section 8, despite the practical success of machine learning, provable hardness of many learning problems also coexists. Even in practice, the learning performances of different datasets vary a lot. For example, the 10-classification problem of training a CNN over CIFAR10, formed by colorful images of animals and vehicles, is much harder than that on MNIST, comprised of samples of handwriting digits. When one trains Resnet18 [29] on the MNIST and CIFAR10 datasets respectively, the accuracy of MNIST can be almost 100%, while it is only about 93% on CIFAR10.

Therefore, given a limited number of samples, it is difficult to overcome such fundamental hardness by simply transforming data before learning from it, meaning that if we request worst-case computational data security, a *necessary* condition is the difference between the encoded accuracy of *any* model the adversary could train from different transformed data should be negligible. Otherwise, the adversary can distinguish the participation of *hardcore* samples, that are hard to classify, in the input set, when the encoded accuracy drops handling this more challenging data.

4.2 Impossibility Result

The following result captures the above ideas and shows that any learnable obfuscation, such that the transformed learning task is non-trivial, *cannot* simultaneously be IND-CPA-secure.

Theorem 1 (Impossibility Result). *Suppose there exists a learnable obfuscation $T(\cdot, \theta) = \big(T_X(\cdot, \theta_X), T_Y(\cdot, \theta_Y)\big)$, where both T_X and T_Y are injective, and satisfy the learnability preservation in Definition 5 with respect to a PAC learnable concept set $C = \{h_0, h_1\}$. Specifically, there exists an efficient algorithm* Alg *such that for some data distribution* D_0, $\mathbb{E}_{x \sim \mathsf{D}_0}[h_0(x)] = 0.5$, *when one inputs the encoded version $\mathcal{S}_T$ of a dataset $\mathcal{S} = \{(x_i, h_0(x_i)), i = 1, 2, \cdots, n\}$, where x_i is i.i.d. from D_0, to* Alg, *once $n \geq m_0$,* Alg *can return some $\hat{h}$, with probability at least* 3/4, *whose encoding accuracy* $\mathsf{Risk}(\hat{h}, T_Y \circ h, T_X(D)) = \Pr_{x \sim \mathsf{D}_0}\big[\hat{h}(T_X(x), \theta_X) \neq T_Y(h(x), \theta_Y)\big] \leq 1/2 - \lambda$, *for some constant $\lambda > 0$. Then, the following claim must be true:*

An adversary can find some PAC learnable concept set $\tilde{C} = \{h_0, \tilde{h}_1, \cdots, \tilde{h}_{2^\nu}\}$, for $\nu \geq 9(16 \log 9/\lambda^2)^2$, and construct a distribution D_1 *such that* $\mathbb{E}_{x \sim \mathsf{D}_1}\mathbb{E}_j[\tilde{h}_j(x)] = 0.5$, *where j is randomly selected from $\{1, 2, \cdots, 2^\nu\}$. The adversary sends the two distributions* D_0 *and* D_1, *and $\tilde{C}$ to the user. The user randomly selects j from $\{1, 2, \cdots, 2^\nu\}$ and generates two datasets S_0 and S_1, each of $n = m_0 + 16 \log 9/\lambda^2$ samples. The features of S_0 and S_1 are i.i.d. generated from* D_0 *and* D_1, *whose labels are determined by h_0 and $\tilde{h}_j$, respectively. The user then generates a random bit $b \in \{0, 1\}$ and seed θ, encodes S_b into $T(S_b, \theta)$, and sends $T(S_b, \theta)$ back to the adversary. The adversary can return $\hat{b} \in \{0, 1\}$ such that* $\Pr(\hat{b} = b) \geq 2/3$.

The proof of Theorem 1 can be found in Appendix A which shows that if there exists some learnable obfuscation such that, for some concept set C, it enables one to train a model with non-trivial, encoded accuracy, captured by $0.5 + \lambda$, then it generally cannot resist a CPA. The adversary can accordingly construct a provably more challenging concept set $\tilde{C}$ compared to C, provided the same number of training samples. Consequently, the adversary can distinguish the encoded dataset labelled via concepts in C or $\tilde{C}$ with constant advantage. In Theorem 1, we restrict T to be injective to avoid the trivial cases, where, for example, one may simply select $T_X = h$ and the encoded data trivially reduces to the label set and there is no need for the server to conduct any additional training.

4.3 Implications

The implications of Theorem 1 are twofold. First, to achieve computational IND-CPA-security, either we need to put stronger restrictions on the plaintext that the adversary can select for the distinguishing challenge, since we cannot afford the worst-case indistinguishability on arbitrary learnable data or, as a necessary requirement, we have to prohibit the adversary's access to the trained model computed over encoded data, which, unfortunately, is beyond the current learnable obfuscation framework (but can be expensively secured by FHE).

Given a new technique, we can check the learnability and privacy requirements of learnable obfuscation: 1) Can the server efficiently classify the encoded samples in the transformed learning task with non-trivial accuracy? 2) Are the data privacy and model security properties satisfied? Suppose that we can find an efficient algorithm to classify transformed data with non-trivial accuracy to satisfy

Property 1. Then, theoretically, we can use the techniques in Section 5 to determine minimal additive noise to satisfy Property 2, and obtain a learnable obfuscation scheme.

The bottom line is that there does not exist a universal and secure learnable obfuscation that works for arbitrary learning tasks. Therefore, if we want to develop efficient obfuscation that does not require adaptively learning from input data, a systematic study on the underlying information-theoretic privacy leakage is demanded. As a final remark, we want to stress that both our proof techniques and the impossibility result in Theorem 1 are different from those in prior works that focus on privately learning a *public* model [10].

5 DATA PAC PRIVACY OF LEARNABLE OBFUSCATION

Given the impossibility results in Section 4, we must consider the utility-privacy tradeoff for learnable obfuscation in general. More importantly, we need to find efficient ways to practically quantify the leakage. In this section, from the angle of PAC Privacy, we present a series of new tools to produce tight and easily-simulatable bounds for Algorithm 1. In particular, to provide a clearer picture of how data mixing and permutation augment the privacy guarantee, we begin with the privacy analysis of matrix masking only and then show the sharpened bound combined with the other obfuscations in Algorithm 1. For simplicity, in the following, we assume U is formed by samples from c categories, each of $N_0 = N/c$ datapoints. A *normalized n-subset* X is formed by n elements, where we select $n_0 = n/c$ many samples from each category of U. Therefore, the corresponding label set is always constant and we only need to focus on the privacy leakage from the encoded feature. We use $\mathcal{N}(\mu, \Sigma)$ to represent a Gaussian distribution of mean μ and (co)variance Σ.

5.1 PAC Privacy of Matrix Masking

From (2), we know the posterior advantage of arbitrary adversarial inference measured in KL-divergence can be bounded by the mutual information. When our release is in a form $XW + B$, obfuscated by matrix masking W and noise perturbation B, the mutual information $\mathsf{MI}(X; XW + B)$ that captures arbitrary inference regarding the entire set X has the following bound.

Theorem 2 (Inference Regarding Entire Set X). *When $W \in \mathbb{R}^{d_0 \times d}$ and $B \in \mathbb{R}^{n \times d}$ are two independent random matrices, where each entry of W is i.i.d. $\mathcal{N}(0, 1)$ and each column of B is i.i.d. in some multivariate Gaussian distribution $\mathcal{N}(\mathbf{0}, \Sigma_B)$ for some non-singular covariance Σ_B, then for X of n data points randomly generated,*

$$\mathcal{MI}(X; XW + B) \le \frac{d}{2} \min \Big\{ \underbrace{\mathbb{E}_{X,X'}\big[Tr(\Sigma^{-1}_{X,B}\Sigma_{X',B}) - n\big]} \mapsto (I),$$
$$\underbrace{\log det(\mathbb{E}_X[\boldsymbol{I}_n + \Sigma_B^{-1}\Sigma_X]) - \mathbb{E}_X\big[\log det(\boldsymbol{I}_n + \Sigma_B^{-1}\Sigma_X)\big]} \mapsto (II)\Big\}, \tag{4}$$

where $\Sigma_X = XX^T$, $\Sigma_{X,B} = XX^T + \Sigma_B$, and X' is distributed independently and identically as X. $det(\cdot)$ represents determinant and $Tr(\cdot)$ represents the trace of a matrix.

Specifically, if our concern is only the membership of a particular u_i from the data pool U, i.e., whether u_i is selected in X, it suffices to consider $\mathsf{MI}(\mathbf{1}_{u_i}; XW + B)$ and we have the following result:

Theorem 3 (Inference Regarding Membership of u_i). *With the same setup as Theorem 2, let X be a normalized n-subset randomly selected from the data pool $\mathsf{U} = \{u_1, u_2, \cdots, u_N\}$, then for $q = n/N$, $\mathcal{MI}(\mathbf{1}_{u_i}; XW + B)$ can be upper bounded by (5). Here, $\mathbf{1}_{u_i}$ is an indicator which represents the participation of u_i in X; X_i is a normalized random n-subset with u_i from U, and X_{-i} is a normalized random n-subset without u_i from $\mathsf{U}\backslash u_i$.*

$$\mathcal{MI}(\mathbf{1}_{u_i}; XW + B) \le \frac{d}{2} \min \Big\{$$
$$q(1-q)\big(\mathbb{E}_{X_i,X_{-i}}\big[Tr(\Sigma^{-1}_{X_i,B}\Sigma_{X_{-i},B}) + Tr(\Sigma^{-1}_{X_{-i},B}\Sigma_{X_i,B}) - 2n\big]\big) \mapsto (I),$$
$$q \log det\big(\mathbb{E}_{X_i}[\boldsymbol{I}_n + \Sigma_B^{-1}\Sigma_{X_i}]\big) + (1-q)\log det\big(\mathbb{E}_{X_{-i}}[\boldsymbol{I}_n + \Sigma_B^{-1}\Sigma_{X_{-i}}]\big)$$
$$- \mathbb{E}_X\big[\log det(\boldsymbol{I}_n + \Sigma_B^{-1}\Sigma_X)\big] \mapsto (II) \qquad \Big\}. \tag{5}$$

The proofs of Theorem 2 and Theorem 3 can be found in Appendix C and D, respectively. We interpret the bounds in (4) and (5) below.

First, given that each entry of masking matrix W is i.i.d. generated from Gaussian $\mathcal{N}(0, 1)$, the distribution of each column of XW is also i.i.d., and *given* the selection of X, it is actually some multivariate Gaussian $\mathcal{N}(\mathbf{0}, XX^T)$. Thus, for a d-dimensional release, where $W \in \mathbb{R}^{d_0 \times d}$, the leakage measured in mutual information in (4) and (5) linearly scales with d, matching the intuition. Second, the distribution of XW is essentially a Gaussian mixture across the random selections (samplings) of X. Thus, if we adopt the KL-divergence expression of mutual information, the objective upper bound is essentially reduced to studying the KL-divergence between two Gaussian mixtures. In (4) and (5), we give two kinds of upper bounds based on two different ideas. The Type (I) bound applies the convexity of KL-divergence and converts the objective to the average pairwise KL-divergence between different Gaussian components. The Type (II) bound relies on the perturbation B and uses the Gaussian KL-divergence decomposition trick in [58].

It is worthwhile noting that both the Type (I) and (II) bounds in (4) and (5) enjoy a simple form, expressed by the expectation over the random sampling of X. This forms the foundation of a simulatable privacy guarantee for any particular dataset and noise selection. As shown later in Section 5.3, all the terms in the Type (I) and (II) bounds can be globally bounded, and thus the objective mutual information can be estimated with confidence by simply empirically averaging the simulated values. Another comment about (5) in Theorem 3 is that $q = n/N$ captures the sampling rate that a particular datapoint u_i is selected in X. Thus, for the Gaussian mixture of $XW + B$, we may consider the two subcases: (i) u_i is not in X captured by $X_{-i}W + B$ for an n-subset X_{-i} randomly selected from $\mathsf{U}\backslash u_i$; and (ii) u_i is in X, denoted by X_i. Analogous to the privacy amplification from sampling in DP [4], the term $q(1-q)$ in (5) can be viewed similarly in the context of individual PAC Privacy.

With the bounds of mutual information, by (2), we are then able to control the posterior advantage measured in KL-divergence for *arbitrary* inference ρ, such as reconstruction hardness. In the following corollary, we focus on the membership challenge in Definition 7, and connect $\mathsf{MI}(X; XW + B)$, associated with the entire input set X, to the average membership inference hardness $\bar{\delta}$.

Corollary 1 (Membership Inference Hardness). *For an arbitrary processing mechanism $\mathcal{M}$ and a normalized n-random-subset X from* *the N-universe $\mathbb{U}$, the adversary's posterior success rate to identify the participation of some $u_i \in \mathbb{U}$ is upper bounded by $(1-\delta_i)$ where, for sampling rate $q = n/N$ and $1-\delta_0 = \max\{q, 1-q\}$,*

$$(1-\delta_i)\log(\frac{1-\delta_i}{1-\delta_0}) + \delta_i \log(\frac{\delta_i}{\delta_0}) \le \mathsf{MI}(\mathbf{1}_{u_i}, \mathcal{M}(X)). \qquad (6)$$

Moreover, the average failure rate $\bar{\delta}$ in Definition 7 satisfies

$$1-\bar{\delta} \le \sum_{j=1}^{n} \frac{\big(\mathsf{MI}(X;\mathcal{M}(X)) + \log 2\big)/n}{-\log \sum_{l=j}^{n} \sum_{\{p_{[1:c]}\}=l} \prod_{z=1}^{c} \left(\binom{n_0}{p_z}\binom{n_0}{n_0-p_z}/\binom{N_0}{n_0}\right)},$$

where $\{p_{[1:c]}\}_{=l}$ represents all sets of non-negative integers $\{p_1, p_2, \cdots, p_c\}$ such that $\sum_{z=1}^{c} p_z = l$.

The proof of Corollary 1 can be found in Appendix B. Combining Corollary 1 with Theorems 2 and 3, to provably prevent adversarial inference, one can theoretically add large enough noise B, until the mutual information bounds in (4) and (5) are small enough to produce satisfied posterior failure rate δ. However, there is still a gap between (4) and (5) and the application of matrix masking being practically usable. For example, consider privately releasing CIFAR10 data, where we randomly sample an $n = 1,000$ subset from the entire training set pool of $N = 50,000$ datapoints. Assume the features of CIFAR10 have been properly embedded and normalized, with details given in Section 7.1. We select the output dimension $d = 500$ and the noise $\Sigma_B = 0.2 \cdot \boldsymbol{I}$. Through (5), on average $\mathsf{MI}(\mathbf{1}_{u_i}, XW + B)$ can only be bounded by 48.5, which cannot produce any useful bound on the membership challenge (δ_i). Or, correspondingly, if one wants $\mathsf{MI}(\mathbf{1}_{u_i}, XW+B) \le 0.3$, where Corollary 1 can then ensure the posterior success rate $(1-\delta_i)$ to identify u_i's membership is upper bounded by 0.2, then one needs to select Σ_B noise that is 17.7 times larger than the expected norm of transformed features!

In fact, this is not (fully) because the bounds (4) and (5) could be loose, but mainly because matrix masking itself leaks a lot when encoding typical data. On one hand, the power of practical data is not uniformly distributed across the entire space, and in most cases XX^T is close to being singular. Therefore, even a small change to X could bring a significant modification to the produced Gaussian distribution $\mathcal{N}(\mathbf{0}, XX^T)$, let alone two randomly generated X and X', which typically share limited common elements for small $q = \frac{n}{N}$. This allows the adversary to easily distinguish them from their encodings. Even worse, the matrix masking encoding of two adjacent datasets that only differ in one datapoint could be very different. Imagine two different selections $X_0 = \{u_1, u_2, \cdots, u_n\}$ and $X_0' = \{u_2, \cdots, u_n, u_{n+1}\}$ which only differ in one datapoint. Intuitively, if the difference between u_i and u_{n+1}, $\|u_1 - u_{n+1}\|$, is close to 0, we should expect that it is impossible to distinguish the encodings of X_0 and X_0'. However, this is not achievable via only matrix masking and the distribution of X_0W and $X_0'W$ could differ a lot: due to the different ordering, the corresponding covariance matrices $\Sigma_{X_0} = X_0X_0^T$ and $\Sigma_{X_0'} = X_0'X_0^T$ are not identical and $\Sigma_{X_0} \cdot \Sigma_{X_0'}^{-1}$ could be far away from being an identity matrix even if $u_1 = u_{n+1}$. To this end, we consider applying data mixing and permutation to address the challenges from ill condition and ordering.

5.2 Privacy Enhancement from Data Mixing and Permutation

We proceed to further incorporate data mixing into obfuscation. As before, to simplify the analysis, we fix the labels of mixed data to be constant in the following artificial way. To generate a total of $m = c^2 m_0$ mixed samples, for any $i, j \in [1:c]$, we consider randomly selecting samples from the i-th and the j-th class to produce m_0 many mixed samples. To be specific, for X which contains $n_0 = n/c$ many samples from each class, each (i,j)*-class-k-mixed sample* is the average of k-randomly-selected samples in X with label i and k-randomly-selected samples in X with label j. We produce m_0 many (i,j)-class-k-mixed samples for each pair (i,j) and for such a mixing matrix M, it is not hard to see that the encoded label set MY is constant, uniformly formed by m up-to-two-hot vectors. Thus, we still only need to consider the privacy leakage from the feature side. With larger k, each (i,j)-class-k-mixed sample approaches the average of the data population from the i-th and j-th class. As shown soon in Theorems 4 and 5, after imposing mixing matrix M on X, the corresponding mutual information bound is similar to (4) and (5), where the only difference is that X becomes MX.

To simultaneously address the unnecessary instability from different orderings, we further impose permutation Π on MXW, which becomes ΠMXW. This allows us to take all the permuted scenarios into account. In particular, for the upper bound on the individual privacy in (8), we only need to consider the divergence of the *closest* pair, i.e., $(u_i, \bar{X})$ and $(u_j, \bar{X})$, differing in one datapoint and identically ordered for some common part $\bar{X}$. We formally present the improved versions of Theorems 2 and 3 below.

Theorem 4 (Improved Theorem 2 with Data Mixing and Permutation). *In the same setup as Theorem 2, with further random data mixing matrix M and permutation matrix Π, $\mathcal{MI}(X; \Pi MXW + B)$ is upper bounded by*

$$\frac{d}{2}\min\Big\{\underbrace{\mathbb{E}_{\Pi,M,\tilde{X},\tilde{X}'}\big[Tr(\Sigma_{\tilde{X},B}^{-1}\Sigma_{\tilde{X}',B}) - m\big]}\mapsto (I), \qquad (7)$$

$$\underbrace{\log det(\mathbb{E}_{\Pi,\tilde{X}}[\boldsymbol{I}_m + \Sigma_B^{-1}\Sigma_{\tilde{X}}]) - \mathbb{E}_{\Pi,\tilde{X}}\big[\log det(\boldsymbol{I}_m + \Sigma_B^{-1}\Sigma_{\tilde{X}})\big]} \mapsto (II)\Big\},$$

where $\tilde{X} = \Pi MX$ and $\tilde{X}' = \Pi MX'$ where X' is independently and identically distributed as X, respectively.

Theorem 5 (Inference Regarding Membership of u_i). *With the same setup as Theorems 3 and 4, under additional data mixing and permutation, $\mathcal{MI}(\mathbf{1}_{u_i}; \Pi MXW + B)$ can be upper bounded by (8). Here, $X_i \overset{c}{\sim} X_{-i}$ denotes a pair of* closest *adjacent matrices in a form $X_i = \{u_i, \bar{X}\}$ and $X_{-i} = \{u_j, \bar{X}\}$ for $i \neq j$: X_i and X_{-i} are adjacent $n \times d_0$ matrices only differing in the first row; $\tilde{X}_i = \Pi MX_i$, $\tilde{X}_{-i} = \Pi MX_{-i}$ and $\tilde{X} = \Pi MX$.*

$$\mathcal{MI}(\mathbf{1}_{u_i}; \Pi MXW + B) \le \frac{d}{2}\min\Big\{q(1-q)\cdot\big(\mathbb{E}_{\Pi,M,X_i \overset{c}{\sim} X_{-i}}\big[$$
$$Tr(\Sigma_{\Pi MX_i,B}^{-1}\Sigma_{\Pi MX_{-i},B}) + Tr(\Sigma_{\Pi MX_{-i},B}^{-1}\Sigma_{\Pi MX_i,B}) - 2m\big]\big) \mapsto (I),$$
$$\mathbb{E}_{\Pi}\big[q \log det(\mathbb{E}_{\tilde{X}_i}[\boldsymbol{I}_m + \Sigma_B^{-1}\Sigma_{\tilde{X}_i}]) + (1-q)\log det(\mathbb{E}_{\tilde{X}_{-i}}[\boldsymbol{I}_m + \Sigma_B^{-1}\Sigma_{\tilde{X}_{-i}}])$$
$$-\mathbb{E}_{\tilde{X}}\big[\log det(\boldsymbol{I}_m + \Sigma_B^{-1}\Sigma_{\tilde{X}})\big]\big] \mapsto (II)\Big\}. \qquad (8)$$

The proofs of Theorems 4 and 5 can be found in Appendix E and F, respectively. Comparing (7) and (8) with (4) and (5), respectively, it is noted that each possible selection of X is now augmented with additional randomness from data mixing and permutation. Specifically for individual privacy captured by the Type (I) upper bound in (8), we only need to compare the divergence of encoded data from the closest adjacent datasets $X_i \overset{\mathsf{c}}{\sim} X_{-i}$ in the *same* ordering under the *same* mixing matrix and the *same* permutation. As shown later in Fig. 1 of Section 7.1, such privacy enhancement is significant when handling practical data. We also observe that Type (II) bounds usually outperform Type (I) in (7); which is contrary to the individual privacy case (8).

5.3 Simulation with Confidence

In this subsection, we show how to produce a high-confidence privacy guarantee via simulations. For simplicity, we consider adding isotropic noise, where each entry of the Gaussian noise B is i.i.d. in a form $\mathcal{N}(0, \sigma^2)$. Based on Corollary 1, the remaining problem is to determine proper σ to produce satisfied mutual information using the upper bounds (7) and (8), described in Theorems 4 and 5. Qualitatively, it is not hard to observe that the bounds always decrease with increasing σ. Therefore, once we can ensure a high-confidence estimation on the privacy guarantee produced by given noise scale σ, via binary search, we can determine the optimal σ for desired security parameters.

In both (7) and (8), once the noise covariance Σ_B is given, the upper bound simply becomes the sum of several expectation terms over the randomness of X. Therefore, on each term, if we can show a high-probability bound of the approximation error within the *empirical averages* from substantially many samplings on X, then by a union bound we can show a high-confidence privacy guarantee for given noise covariance Σ_B. We take the Type (I) bound in (8) as an example and describe this framework as Algorithm 2 in Appendix I. Let $\mathcal{E}_L$ be the empirical average of the simulated values across L independent samplings on X, then the parameters of estimated variance η for confidence bound $(1 - \gamma)$ are selected as follows.

Theorem 6 (Simulation Complexity). *Suppose the data pool* U *is bounded such that* $\|u_i\|_2 \leq 1$ *for any* $u_i \in \mathsf{U}$*. Then, in Algorithm 2, which estimates the Type (I) upper bound in (8) by the empirical average from L independent trials, to ensure with high confidence* $(1 - \gamma)$ *the objective mutual information* $\mathcal{MI}(\mathbf{1}_{u_i}; \Pi M X W + B) \leq \mathcal{E}_L + \eta$*, the estimated variance* η *can be selected as*

$$\eta \geq \frac{dq(1-q)}{\sqrt{2L}} \Big\{ \sqrt{\log(\frac{2}{\gamma})} \big(m(1 + \frac{\sqrt{m}}{\sigma^2}) \big) \Big\}. \tag{9}$$

Similarly, in the same setup for Type (II) upper bounds, it suffices to select η *as,*

$$\eta \geq \frac{d}{2} \Big\{ \log(1 + \frac{m(\sqrt{m} + \alpha)\alpha}{\sigma^{2m}}) + \sqrt{\frac{2\log(\frac{3}{\gamma})}{L}} \big(m \log(1 + \frac{\sqrt{m}}{\sigma^2}) \big) \Big\}, \tag{10}$$

where $\alpha = \sqrt{32 \log(\frac{6m}{\gamma})(1 + \frac{\sqrt{m}}{\sigma^2})^2 / L}$*. Moreover, after replacing* $q(1 - q)$ *by* 1 *in (9), (9) and (10) also work for the Type (I) and (II) upper bounds of (7) in Theorem 4.*

The proof of Theorem 6 can be found in Appendix G. Theoretically, to produce a $(1 - \gamma)$ confidence estimate, we need $L = \tilde{O}(m^4 d^2 \log(1/\gamma))$. However, we need to point out Theorem 6 does *not* put additional assumptions on the dataset U except for a very weak global l_2-norm bound. The analysis in (9) and (10) can be significantly sharpened with mild nonsingularity of mixed samples MX and, in practice, the empirical averages of (7) and (8) converge fast, where hundreds of simulation trials are usually sufficient, as shown later in Fig. 2 of Section 7.1.

5.4 Learnable Noise from Public Data

We now point out that it is *not* necessary to fix the noise B to be isotropic, but instead we can construct *learnable (fake) noise.* Assume some public set (X_{pub}, Y_{pub}) is in the same format as the sensitive set (X, Y). Consider the following construction $\Pi M X W_1 + M_0 X_{pub} W_2$ for independent masking matrices W_1 and W_2 on the features with some fixed mixing matrix M_0. Accordingly, to produce useful labels given that $X_{pub} W_2$ forms an independent transformed set, we consider the concatenation $[\Pi M Y, M_0 Y_{pub}]$. As an example, if both (X, Y) and (X_{pub}, Y_{pub}) are for some 10-classification problems, where both Y and Y_{pub} are formed by one-hot 10-dimensional vectors, the labels of composite $\Pi M X W_1 + M_0 X_{pub} W_2$ are vectors of dimension 20, by simply concatenating $\Pi M Y$ and $M_0 Y_{pub}$. Consequently, the model $\mathcal{N}_T$ trained over such transformed data needs to recognize 10×10 composite classes.

To apply such model $\mathcal{N}_T$ for prediction on some newly-incoming sample x_q, instead of inputting $x_q W_1$ solely, we will generate $x_q W_1 + x_{pub} W_2$ under multiple selections of $x_{pub} \in X_{pub}$, and determine the prediction based on majority voting (ensemble) of $\mathcal{N}_T(x_q W_1 + x_{pub} W_2)$. From a privacy standpoint, the upper bounds in both (7) and (8) still work for such a construction, where the only difference is that the noise covariance Σ_B now becomes $\Sigma_{M_0 X_{pub}, B}$. In Section 7.1, we will show when the distribution of X is concentrated around X_{pub}, i.e., sensitive data is close to public data, then such *learnable noise can efficiently replace random noise* and produce a sharpened utility-privacy tradeoff.

6 HARDNESS OF REVERSE ENGINEERING

We present results on provable hardness of reverse engineering. Specifically, we state the problem as follows. Given some data distribution $x \sim \mathsf{D}$ and the release of $XW + B$, for an independent Gaussian masking matrix W with each entry in $\mathcal{N}(0, 1)$, we wish to characterize the hardness that the adversary can return a function f_{adv} satisfying (ψ, τ) statistical encoding distance (Definition 8):

$$\Pr_{x \sim \mathsf{D}} \big(\|xW - f_{adv}(x)\|_2 < \psi \big) \geq 1 - \tau. \tag{11}$$

Clearly, the hardness must count on D, where in the extreme case if D is degenerate such that $\Pr_{x \sim \mathsf{D}}(x = 0) = 1$, the problem becomes trivial to the adversary. In the following, we set $\mathbb{E}_{x \sim \mathsf{D}}[x] = \mathbf{0}$ with additional two assumptions on D.

Assumption 1. *For any unit* $z \in \mathbb{R}^{d_0}$*,* $\|z\|_2 = 1$*, the variance of* x *after projection satisfies* $\mathbb{E}[\langle x, z \rangle^2] \geq \kappa^2$*, for some parameter* κ*.*

Assumption 1 characterizes the variance of x, which will play the key role in lower bounding the adversarial inference error, captured by ψ. We expect that the projection of x along every direction in

$\mathbb{R}^{d_0}$ is of sufficient energy (variance) κ^2. We want to mention this is a mild assumption for machine learning tasks invariant to data rotation. One may consider uniformly applying a random rotation over the data source x. This can ensure the power of x along any particular direction to be identically the average of that across all directions. After a random rotation, for any fixed x and unit vector z, $\mathbb{E}_x[\langle x, z\rangle^2] = \mathbb{E}_{x,z'}[\langle x, z'\rangle^2]$ for a random unit vector z'.

Assumption 2. *For any $z \in \mathbb{R}^d$ in any fixed distribution D_z over the sphere $\|z\| = 1$, $\langle z, x\rangle$ is of a subGaussian tail such that for some constant K, $\Pr_{z,x}(|\langle z, x\rangle| \geq t) \leq 2e^{-t^2/K^2}$.*

In Assumption 2, we assume concentration of the data source x, which enables us to derive high-probability bounds τ on the estimation error. Roughly speaking, we assume that after applying a Lipschitz-1 linear function on x, it is of a subGaussian tail. Assumption 2 is only used to produce a high probability requirement τ in (11), and is unnecessary if one only cares about the expected error $\mathbb{E}_{x\sim\mathcal{D}}[\|Wx - f_{adv}(x)\|]$. Together, Assumptions 1 and 2 form the foundation of the following inference hardness result.

Theorem 7. *For any data distribution $\mathcal{D}$ satisfying Assumptions 1 and 2, when $d = d_0$, after the observation of $XW + B$, the posterior success rate $(1-\delta)$, that an adversary returns a function f_{adv} matching (ψ, τ)-distance in (11), satisfies*

$$1 - \delta \leq \frac{\mathsf{MI}(X; XW + B) + \frac{d_0}{2}\mathbb{E}_X \log(det(\boldsymbol{I} + \frac{XX^T}{\sigma^2})) + \log(2)}{(d_0^2 - \beta^2)^2/(4d_0^2)}, \tag{12}$$

where $\beta^2 = d^2 - 2d_0\sqrt{t}$, $\psi = \frac{\sqrt{\kappa^2\beta^2 - t}}{2}$, $\tau = \frac{1-e^{-t^2/(CK^4\beta^4)}}{2}$, for a freely selectable parameter t and constant C.

The proof of Theorem 7 can be found in Appendix H. In Theorem 6, we have already shown the simulatable upper bounds of the terms in the numerator of (12). Asymptotically, when $\beta = \Theta(d_0)$, and thus $\psi = \Theta(d_0)$ for some $\tau \in (0, 0.5)$, the denominator of (12), which captures the upper bound of $\log(1/(1-\delta_o))$ for the optimal *a priori* success rate $(1 - \delta_o)$ is $\Theta(d_0)$. When the numerator is constant, it suggests that the adversarial posterior success rate to find an f_{adv} that is (ψ, τ)-close to W according to Eqn. 11, is up to $O(1/d_0)$.

7 EXPERIMENTS

7.1 Reconstruction Robustness

We present experiments to illustrate the practicality of our analysis. We consider applying Algorithm 1 to obfuscate the CIFAR10 dataset, with a total of 50,000 training samples, and the untrusted server trains a 3-layer fully-connected network on transformed data. Prior to applying the obfuscations, we preprocess the data. We embed (uniformly transform) images using a ResNet-50 network pretrained on ImageNet [52] and normalize the l_2-norm of each image to 1. Such preprocessing is necessary to reduce the ill condition of raw data matrices: without it, in the same setup from raw data, mutual information bounds computed using Theorems 2-5 can be 1,000× larger than those from normalized and embedded data[1], which are reported below. We *cannot* show meaningful privacy guarantees for existing learnable obfuscation proposals, e.g., NeuraCrypt [62], that handle raw image data directly, free of additional noise perturbation B, but have questionable security [11].

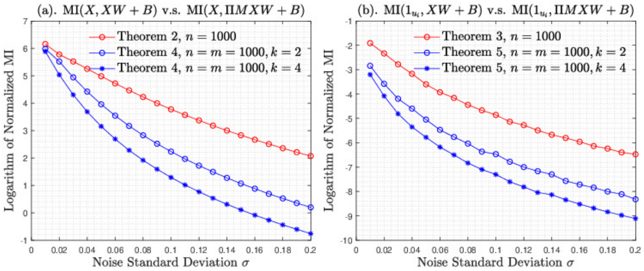

Figure 1: Logarithm of Mutual Information Bounds Computed from Theorems 2-3 (matrix masking only) and 4 -5 (matrix masking+mixing+permutation).

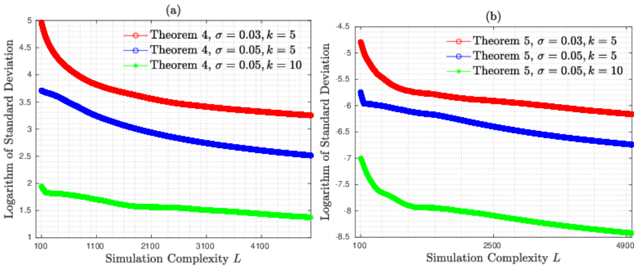

Figure 2: Logarithm of the standard deviation of empirical mean estimator for Theorems 4 and 5.

In the following, we assume that each entry of the (normalized) masking matrix $W \in \mathbb{R}^{d_0 \times d}$ and noise $B \in \mathbb{R}^{m \times d}$ are i.i.d. in $\mathcal{N}(0, 1/d)$ and $\mathcal{N}(0, \sigma^2)$, respectively, such that the l_2 norm of each transformed image is close to 1. From our observation, for matrix-masked data, there usually exists a critical point for the selection of output dimension d, which is around 500 in our CIFAR10 example: when $d \geq 500$, the accuracy improves slowly as d further increases; when $d \leq 500$, the accuracy drops sharply as d decreases. To this end, we fix $d = 500$ in all reported experiments.

In Fig. 1 (a) and (b), we first record the mutual information bounds presented in Theorems 2-5 under various selections of mixing parameter k and noise scale σ for a randomly-selected normalized n-subset X. As explained at the end of Section 5.1, data mixing and permutation mitigates the problems from the ill condition and different orderings of X, and the sharpened bounds from Theorems 4 and 5 can be 10× smaller compared to those obtained from Theorems 2 and 3 when we only apply matrix masking. In Fig. 2, we show the convergence rate of the estimations of the bounds in Theorems 4 and 5 in different scenarios. We record the standard deviation of the empirical mean estimator provided L independent samplings. From Fig. 2, for practical data, one may achieve 10^{-3} variation of individual mutual information estimation for L in a scale of hundreds. In all the following privacy guarantees reported in Tables 3 and 5, we run the simulations until the variation of produced mean estimation for the objective bound is less than 10^{-3}.

We proceed to interpret the privacy guarantee in semantic contexts of an individual reconstruction challenge (Definition 6) and a membership inference challenge (Definition 7), with a clear comparison to existing privacy accounting methods [5, 50]. To match

[1] We also observe that many other embedding methods can help the stability, for example, BERT [15] for language/text data.

$(n, m, k)\backslash\sigma$	0	0.02	0.03	0.04	0.05
(1K,2K,5)	83.4	82.3 (0.40,0)	81.8 (0.56,0)	80.9 (0.63,0)	80.1 (0.67,0)
(1K,4K,5)	84.5	83.7 (0.21,0)	83.0 (0.43,0)	82.6 (0.54,0)	81.8 (0.60, 0)
(1K,4K,10)	82.8	82.4 (0.54,0)	82.1 (0.65,0)	81.6 (0.69, 0.08)	81.4 (0.73,0.56)
(2K,4K,10)	84.3	82.9 (0.39,0)	82.6 (0.56,0)	82.0 (0.63,0.08)	81.8 (0.67, 0.56)
(2K,6K,15)	84.4	83.1 (0.56,0)	82.6 (0.66,0.15)	82.4 (0.71, 0.63)	82.0 (0.74,0.69)
(4K,6K,15)	84.9	84.0 (0.49,0)	83.6 (0.61,0.15)	83.3 (0.67,0.63)	82.6 (0.71,0.69)

Table 1: Test Accuracy (%) (Posterior Reconstruction Failure Probability δ for l_2-norm estimation error smaller than 0.75 ensured by Theorem 4, and ensured by RDP accounting [5]) over encoded preprocessed CIFAR10 data via Algorithm 1.

the i.i.d. generation setup in Definition 6 and [5], we consider a set $\mathcal{S} = \{s_1, s_2, \cdots, s_N\}$, $N = 50,000$, where each s_i is i.i.d. uniformly selected from the embedded CIFAR10 dataset, and X is a normalized subset of size n of $\mathcal{S}$, as our training data, which we obfuscate by applying Algorithm 1. We focus on an adversarial reconstruction task as estimating a single individual s_i with error in l_2-norm ≤ 0.75. Given the random sampling setup, the corresponding optimal prior success rate is $(1 - \delta_o) = 0.16$. In Table 1, we record the performance of the trained model over obfuscated X via Algorithm 1 in various setups, where k captures the mix number as defined in Section 5.2. For each case, we perform 5 independent trials on sampling (X, W, M) and noise B, and report the median test accuracy. The accuracy loss caused by obfuscation (matrix masking and mixing only) varies between 2-3%. As a benchmark, if the *entire* embedded, untransformed CIFAR10 without additional noise is used for non-private training, the same fully-connected network can achieve 94.5% accuracy. The first number in each pair of brackets is the *lower bound* on the failure rate ensured by Theorem 4 for an adversary to return a satisfied reconstruction with error ≤ 0.75.

We also compare the results using [5] which applies RDP accounting (3) to analyze the reconstruction robustness of Algorithm 1 on $\mathcal{S}$. We implement Poisson subsampling to produce X to simulate the (i, j)-class-k-mixing on images from each pair of (i, j) classes, where each individual in each class will be independently selected at a rate $\frac{k}{N/10}$. Thus, we can apply the parallel composition of subsampled Gaussian mechanisms [52] to determine the RDP parameters (3), where the sensitivity of each released mixed data is $1/K$. We want to stress that the RDP accounting can only apply to the noise and subsampling, rather than the additional obfuscations as analyzed in Theorems 2-5. Given the same noise σ, the performance of such differentially-private releasing is very slightly worse than (<0.5%) that of Algorithm 1, mainly because of the variation by Poisson sampling. For brevity, we omit these results.

In comparison, our results produce much tighter privacy analysis against individual reconstruction. From Table 1, in many scenarios, [5] via RDP accounting cannot provide meaningful guarantees, where the adversary's failure rate can only be trivially lower bounded by 0. k, n and the total number of released mixed samples m show the following utility-privacy tradeoff. A larger m and n with a smaller k can ease the transformed learning task; but simultaneously, a smaller n (subsampling rate) produces a larger privacy amplification. Alternately, a larger k, which implies more intensive mixing, and a smaller m can form a smaller and more stable MX training dataset with less leakage. We have similar observations in Table 3 in Section 7.2 when we consider membership inference.

$(n, m, k)\backslash\sigma$	0	0.02	0.03	0.04	0.05
(1K,2K,5)	84.2	83.8 (0.59)	83.6 (0.67)	83.2 (0.71)	82.5 (0.74)
(1K,4K,5)	85.0	84.3 (0.45)	84.0 (0.59)	83.6 (0.66)	83.3 (0.70)
(1K,4K,10)	84.4	83.6 (0.61)	83.2 (0.70)	82.5 (0.74)	81.9 (0.76)
(2K,4K,10)	84.9	84.3 (0.50)	84.0 (0.63)	83.5 (0.68)	83.3 (0.72)
(2K,6K,10)	85.2	84.7 (0.61)	84.3 (0.69)	84.2 (0.73)	83.6 (0.76)
(4K,6K,10)	85.7	85.1 (0.55)	84.5 (0.65)	84.2 (0.70)	83.6 (0.73)

Table 2: Test Accuracy (%) (and Posterior Reconstruction Failure Probability δ ensured by Theorem 4) over encoded preprocessed CIFAR10 data with Learnable Noise.

In Table 2, we further incorporate learnable fake noise (Section 5.4) into Algorithm 1. We randomly select 1,000 samples from CIFAR10 to form a public set X_{pub} and replace the isotropic Gaussian B by $M_0 X_{pub} W_2 + B$ for some fixed k-mixing matrix M_0 and an independent Gaussian matrix W_2. Equivalently, we now add an anisotropic Gaussian noise of covariance $\Sigma_{M_0 X_{pub}, B}$. Besides, we also use the augmented inference method of Section 5.4. For each test data, we apply the model on its mixed versions with 100 randomly selected public samples from X_{pub}. Comparing Table 2 with Table 1, in the same setup (including $\sigma = 0$), the additional learnable noise $M_0 X_{pub} W_2$ significantly amplifies the privacy, and performance improves. This is because we also learn over $X_{pub} W_2$, since the trained classifier handles composite 10×10 classification, and the transformed classes have strong correlation.

7.2 Membership Inference

In this section, we study the provable robustness of Algorithm 1 against membership inference, and conduct a membership inference attack as a means to validate our theory and implementation. Our theory provides upper bounds, and the attack provides a lower bound on adversarial success rate.

We adopt the data generation setup described in Definition 7, where U is still the embedded CIFAR10 of $N = 50,000$ datapoints and X is a random normalized subset of size n of U. This is slightly different from the i.i.d. uniform generation in Section 7.1 to match the data reconstruction setup. But, in both cases. the produced X is roughly a random n normalized subset of U.

To have a fair comparison with [50], we also focus on the positive identification accuracy rate: [50] (and references therein) considers a probabilistic adversary, who, given the observation $\mathcal{M}(X)$, returns $\hat{X}$ in the following distribution,

$$\mathsf{P}(\hat{X} = X_0) = \mathsf{P}(X = X_0 | M(X) = o), \tag{13}$$

and defines the positive accuracy $\Pr(\mathbf{1}_{u_i \in X} = \mathbf{1}_{u_i \in \hat{X}} = 1) = (1 - \delta_i)$. To our knowledge, [50] presents the best-known posterior success rate for a *generic* subsampling rate, which shows that if a processing mechanism $\mathcal{M}$ satisfies ϵ_0-DP, the posterior failure rate δ_i to positively correctly identify the membership of an individual is lower bounded by

$$\frac{(1-q)e^{-\epsilon_0}}{q + (1-q)e^{-\epsilon_0}}. \tag{14}$$

In our case, $q = n/N$ is the rate that an individual gets sampled in X, and the optimal failure rate of above-described positive identification one may expect is $(1 - q)$.

We select the first 100 samples in the CIFAR10 set as our objectives. In Table 3, we record the performance of the trained model over obfuscated data via Algorithm 1 in various setups. For each case, we perform 5 independent trials on sampling (X, W, M) and noise B, and report the median test accuracy. It is not surprising that the performance reported in Table 3 is almost identical to that in Table 1 given the similarity of produced X, as explained earlier.

Similar to Table 1, the first number in each pair of brackets is the *lower bound* on the failure rate ensured by Theorem 5 and Corollary 1 that the adversary can positively identify the membership of the 100 target datapoints. Elaborating, given the mutual information bound from (8), we correspondingly calculate the lower bounds of each δ_i and report the smallest one.

We also compare the results using [50] to apply DP auditing to analyze Algorithm 1 on the same preprocessed data X. The second number in each pair of brackets of Table 3 captures this. It can be seen when ϵ_0 is large, (14) is close to $e^{-\epsilon_0}/q$, which decays exponentially and (14) can only produce usable guarantees when ϵ_0 is some small constant $O(\log(1/q))$. This explains why the DP lower bounds in Table 5 vary heavily and are only meaningful given large enough noise σ, the number of sampled samples n, and mixing parameter k to ensure small enough sensitivity. In comparison, our results produce much tighter instance-based privacy analysis.

$(n, m, k)\backslash\sigma$	0	0.03	0.04	0.05
(1K,2K,5)	83.4	81.7 (0.30,3.10^{-3})	80.9 (0.61,0.01)	80.2 (0.70,0.03)
(1K,4K,5)	84.5	83.2 (0.14,3.10^{-7})	82.6 (0.28,3.10^{-6})	81.6 (0.57,4.10^{-5})
(1K,4K,10)	82.8	82.1 (0.33,5.10^{-12})	81.8 (0.62,2.10^{-6})	81.3 (0.68,6.10^{-3})
(2K,4K,10)	84.3	82.5 (0.18,7.10^{-5})	82.0 (0.45,0.05)	81.7 (0.67,0.11)
(2K,6K,15)	84.4	82.8 (0.37,6.10^{-4})	82.5 (0.59,0.03)	81.9 (0.68,0.14)
(4K,6K,15)	84.9	83.7 (0.09,0.05)	83.3 (0.23,0.08)	82.8 (0.36,0.27)

Table 3: Test Accuracy (%) (and Posterior Membership Positive Identification Failure Probability ensured by Theorem 5, and ensured by Differential Privacy [50]) over encoded preprocessed CIFAR10 data via Algorithm 1.

$(n, m, k)\backslash\sigma$	0.03	0.04	0.05
(1K,2K,5)	(0.44, 0.30)	(0.78, 0.61)	(0.84, 0.70)
(1K,4K,5)	(0.19, 0.14)	(0.50, 0.28)	(0.73, 0.57)
(1K,4K,10)	(0.45, 0.33)	(0.78, 0.62)	(0.79, 0.68)
(2K,4K,10)	(0.34, 0.18)	(0.65, 0.45)	(0.74, 0.67)
(2K,6K,15)	(0.55, 0.37)	(0.81, 0.59)	(0.87, 0.68)
(4K,6K,15)	(0.36, 0.09)	(0.60, 0.23)	(0.71, 0.36)

Table 4: Posterior Membership Positive Identification Failure Probability by Empirical Estimation, and ensured by Theorem 5 over encoded preprocessed CIFAR10 data via Algorithm 1.

We have focused on relatively small q mainly because we want large failure probability ($\leq 1 - q$), and also because our results could be significantly improved in the high q regime: the rate that an individual gets involved in the mixed version MX could be smaller than q, i.e., that it gets selected in X.

In Table 4, we provide empirical estimations of the positive identification rate, as the first number in the brackets, which are determined by a concrete attack, inspired by [9], which we describe below. Given the output (release) $o = \Pi MXW + B$, the noisy obfuscated samples, the adversary randomly selects 100 subsets of U, $X_1^+, \cdots, X_{100}^+$, where each subset is formed by n datapoints and includes the objective datapoint u_0 under membership inference concern. The adversary also randomly and independently selects the obfuscations Π_i^+, M_i^+, W_i^+, for $i = 1, 2, \cdots, 100$. Similarly, the adversary randomly and independently selects $X_i^-, \Pi_i^-, M_i^-, W_i^-$, for $i = 1, 2, \cdots, 100$, where each X_i^- is an n-subset of U, excluding the objective datapoint u_0 under membership inference concern. The adversary then estimates the likelihood that o is produced by some X with or without u_0 by $\frac{1}{100} \cdot \sum_{i=1}^{100} \mathbb{P}\big(\Pi_i^+ M_i^+ X_i^+ W_i^+ + B' = o\big)$ and $\frac{1}{100} \cdot \sum_{i=1}^{100} \mathbb{P}\big(\, \Pi_i^- M_i^- X_i^- W_i^- + B' = o\big)$, respectively, where B' is the Gaussian noise in the same distribution as B. The adversary then, based on the estimated likelihoods, determines the membership guess defined in (13) and we report their average failure rate. As expected, the empirical estimation from a specific attack only provides an upper bound on the adversary's failure rate, while our results from PAC Privacy provide a provable lower bound.

$(n, m, k)\backslash\sigma$	0	0.03	0.04	0.05
(1K,2K,5)	84.2	83.5 (0.53)	83.0 (0.74)	82.6 (0.83)
(1K,4K,5)	85.0	84.1 (0.36)	83.8 (0.58)	83.4 (0.66)
(1K,4K,10)	84.4	83.1 (0.71)	82.7 (0.73)	82.2 (0.84)
(2K,4K,10)	84.9	84.2 (0.43)	83.7 (0.64)	83.4 (0.77)
(2K,6K,10)	85.2	84.5 (0.34)	84.1 (0.54)	83.7 (0.67)
(4K,6K,10)	85.7	84.8 (0.24)	84.3 (0.48)	83.9 (0.61)

Table 5: Test Accuracy (%) (and Posterior Membership Positive Identification Failure Probability ensured by Theorem 5) over encoded preprocessed CIFAR10 data with Learnable Noise.

Similarly, in Table 5, we further incorporate learnable fake noise (Section 5.4) into Algorithm 1. Identically, we randomly select 1,000 samples from CIFAR10 to form a public set X_{pub} and replace the isotropic Gaussian B by $M_0 X_{pub} W_2 + B$ for some fixed k-mixing matrix M_0 and an independent Gaussian matrix W_2. Equivalently, we now add an anisotropic Gaussian noise of covariance $\Sigma_{M_0 X_{pub}, B}$. We also use the augmented inference method of Section 5.4. For each test data, we apply the model on its mixed versions with 100 randomly selected public samples from X_{pub}. Comparing Table 5 with Table 3, in the same setup, the additional learnable noise $M_0 X_{pub} W_2$ significantly amplifies the privacy, and meanwhile the performance is even better.

8 ADDITIONAL RELATED WORK

We elaborate on the fundamental differences between learnable obfuscation and classic cryptographic and DP primitives.

Property-preserving Encryption: To support efficient searching over encrypted data, property-preserving encryption represents a line of work aimed at generating ciphertexts that maintain the necessary information and features of plaintexts while preventing adversaries from distinguishing other features with a significant advantage. There exist various examples in this field, such as order-preserving encryption (OPE) [2] and searchable encryption [35, 49]. However, cryptanalysis and overhead requirements for property-preserving encryption are still under investigation, and certain impossibility results have been discovered. For instance, it has been proven [39, 44] that no efficient construction exists for OPE, or alternatively, the ciphertext needs to be exponentially larger than the plaintext to achieve ideal security while only exposing the plaintext's ordering. As mentioned before, most applications of learnable obfuscation do not necessarily require a decryption paradigm: the user only needs to encode the sample via the learnable obfuscation function and then can apply the model trained on the transformed domain. However, despite this appealing freedom, we have demonstrated that it remains generally impossible for learnable obfuscation to achieve cryptographic security, which advances the understanding of this framework. On the other side, our results for measuring leakage from an information-theoretic perspective point out a new possible way to measure privacy and can be employed to study the leakage of other property-preserving encryption as well.

Hardness Results of Learning: Despite the empirical success of machine learning, there exists a significant body of research focused on studying the impossibility results of learning certain classes of functions, particularly those computed by neural networks. Surprisingly, in the worst-case scenario of data distribution, even learning a two-layer fully-connected network can be computationally challenging. An active area of research in this field revolves around the hardness of learning intersections of halfspaces [36]. For example, Klivans and Sherstov [36], based on the hardness assumption of the shortest vector problem, proved that learning $poly(d)$ halfspaces is hard. This number was further improved to $\Omega(\log(d))$ based on the RSAT assumption [13]. Those results suggest that a two-layer network with $\Omega(\log(d))$ many hidden neurons with Relu activation functions could be hard to learn in general, which is still true even when the network's weights are well-behaved in normal or uniform distributions [14]. Other related hardness results regarding statistical query models or gradient-descent-based methods can be found in [22], [47]. Though with a different motivation, it raises an interesting question as to whether the hardness of learnable obfuscation could count on those impossibility results. Unfortunately, there are two main obstacles to straightforward application in their current forms. In general, to show that learning is impossible, those hardness results mostly reduce to proving that there exists *certain* data distribution $x \sim \mathrm{D}$ or some neural network function $f(\cdot)$ such that $f(x)$ is computationally indistinguishable. On one hand, existing works only support a worst-case hardness, while we need a more generic leakage quantification for private data from practical distributions. On the other hand, prior constructions with a reduction to computational indistinguishability also suggest that no further meaningful learning can be applied to transformed data, and thus we need different techniques for learnable obfuscation.

Private Synthetic Data Release and Differential Privacy: Our primary focus in this paper is on formalizing the privacy leakage from straightforward heuristic obfuscation techniques that maintain topology of input data. However, there is another promising approach for privately releasing synthetic learnable data, which involves the use of generative models. Generative AI techniques, such as Generative Adversarial Network (GAN) [23] and Diffusion models [31], have achieved remarkable success in the many fields. If we can ensure that the generative model trained on sensitive data is already private, then the synthetic data generated afterwards, as much as needed, becomes a postprocessing step that does not introduce additional privacy risks. Nonetheless, this approach differs in a key aspect from the main motivations of learnable obfuscation. Training state-of-the-art generative models is computationally intensive, even without privacy restrictions. An ideal learnable obfuscation scheme is expected to efficiently transform the input data without needing to learn from it. DP-SGD enables private training of a model, that can be subsequently released with privacy guarantees on training data, but besides the heavy computational overhead of training, heavy utility loss is incurred [17]. So far, to efficiently *release data with a provable privacy guarantee*, pure noise perturbation characterized by LDP is still the most general and popular approach. However, as a more challenging problem, LDP [18] also heavily suffers from the curse of dimensionality and strict impossibility results are known for its applications in the high-dimensional ($n \ll d$) scenarios [18].

9 CONCLUSION

In this paper, we formalized the concept of learnable obfuscation and developed a series of new tools to show a provable information-theoretic privacy guarantee of three important heuristic obfuscation methods. Our impossibility results and a successful example (after appropriate data preprocessing) can be used to guide the search of other efficient transformations for learnable obfuscation construction. Our developed tools and results open new possibilities to produce privacy enhancement by exploiting different randomness besides independent isotropic noise, and present an innovative research direction to study general property-preserving encryption from an information-theoretic angle.

REFERENCES

[1] Radoslaw Adamczak. 2015. A note on the Hanson-Wright inequality for random vectors with dependencies. *Electronic Communications in Probability* 20 (2015), 1–13.

[2] Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, and Yirong Xu. 2004. Order preserving encryption for numeric data. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*. 563–574.

[3] Michael Backes, Pascal Berrang, Mathias Humbert, and Praveen Manoharan. 2016. Membership privacy in MicroRNA-based studies. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 319–330.

[4] Borja Balle, Gilles Barthe, and Marco Gaboardi. 2018. Privacy amplification by subsampling: Tight analyses via couplings and divergences. *Advances in neural information processing systems* 31 (2018).

[5] Borja Balle, Giovanni Cherubin, and Jamie Hayes. 2022. Reconstructing training data with informed adversaries. In *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1138–1156.

[6] Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. 2012. The johnson-lindenstrauss transform itself preserves differential privacy. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*. IEEE, 410–419.

[7] Eitan Borgnia, Jonas Geiping, Valeriia Cherepanova, Liam Fowl, Arjun Gupta, Amin Ghiasi, Furong Huang, Micah Goldblum, and Tom Goldstein. 2021. DP-instahide: Provably defusing poisoning and backdoor attacks with differentially private data augmentations. *arXiv preprint arXiv:2103.02079* (2021).

[8] Mark Bun and Thomas Steinke. 2016. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*. Springer, 635–658.

[9] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. 2022. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1897–1914.

[10] N. Carlini, S. Deng, S. Garg, S. Jha, S. Mahloujifar, M. Mahmoody, A. Thakurta, and F. Tramer. 2021. Is Private Learning Possible with Instance Encoding?. In *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 410–427. https://doi.org/10.1109/SP40001.2021.00099

[11] Nicholas Carlini, Sanjam Garg, Somesh Jha, Saeed Mahloujifar, Mohammad Mahmoody, and Florian Tramer. 2021. NeuraCrypt is not private. *arXiv preprint arXiv:2108.07256* (2021).

[12] Sitan Chen, Zhao Song, and Danyang Zhuo. 2020. On InstaHide, Phase Retrieval, and Sparse Matrix Factorization. *arXiv preprint arXiv:2011.11181* (2020).

[13] Amit Daniely. 2016. Complexity theoretic limitations on learning halfspaces. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*. 105–117.

[14] Amit Daniely and Gal Vardi. 2020. Hardness of learning neural networks with natural weights. *Advances in Neural Information Processing Systems* 33 (2020).

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[16] A Adam Ding, Guanhong Miao, and Samuel S Wu. 2020. On the privacy and utility properties of triple matrix-masking. *The Journal of privacy and confidentiality* 10, 2 (2020).

[17] Tim Dockhorn, Tianshi Cao, Arash Vahdat, and Karsten Kreis. 2022. Differentially private diffusion models. *arXiv preprint arXiv:2210.09929* (2022).

[18] John C Duchi, Michael I Jordan, and Martin J Wainwright. 2013. Local privacy and statistical minimax rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*. IEEE, 429–438.

[19] Cynthia Dwork. 2006. Differential privacy. In *International colloquium on automata, languages, and programming*. Springer, 1–12.

[20] Vitaly Feldman, Tomer Koren, and Kunal Talwar. 2020. Private stochastic convex optimization: optimal rates in linear time. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*. 439–449.

[21] Craig Gentry. 2009. Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*. 169–178.

[22] Surbhi Goel, Aravind Gollakota, Zhihan Jin, Sushrut Karmalkar, and Adam Klivans. 2020. Superpolynomial lower bounds for learning one-layer neural networks using gradient descent. In *International Conference on Machine Learning*. PMLR, 3587–3596.

[23] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).

[24] Shruthi Gorantala, Rob Springer, and Bryant Gipson. 2023. Unlocking the potential of fully homomorphic encryption. *Commun. ACM* 66, 5 (2023), 72–81.

[25] Chuan Guo, Brian Karrer, Kamalika Chaudhuri, and Laurens van der Maaten. 2022. Bounding Training Data Reconstruction in Private (Deep) Learning. *arXiv preprint arXiv:2201.12383* (2022).

[26] Aparna Gupte, Neekon Vafa, and Vinod Vaikuntanathan. 2022. Continuous lwe is as hard as lwe & applications to learning gaussian mixtures. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 1162–1173.

[27] Awni Hannun, Chuan Guo, and Laurens van der Maaten. 2021. Measuring data leakage in machine-learning models with Fisher information. In *Uncertainty in Artificial Intelligence*. PMLR, 760–770.

[28] Jamie Hayes, Saeed Mahloujifar, and Borja Balle. 2023. Bounding Training Data Reconstruction in DP-SGD. *arXiv preprint arXiv:2302.07225* (2023).

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[30] Qijian He, Wei Yang, Bingren Chen, Yangyang Geng, and Liusheng Huang. 2020. Transnet: Training privacy-preserving neural network over transformed layer. *Proceedings of the VLDB Endowment* 13, 12 (2020), 1849–1862.

[31] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.

[32] Yangsibo Huang, Zhao Song, Kai Li, and Sanjeev Arora. 2020. Instahide: Instance-hiding schemes for private distributed learning. In *International Conference on Machine Learning*. PMLR, 4507–4518.

[33] Piotr Indyk, Rajeev Motwani, Prabhakar Raghavan, and Santosh Vempala. 1997. Locality-preserving hashing in multidimensional spaces. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*. 618–625.

[34] Ilse CF Ipsen and Rizwana Rehman. 2008. Perturbation bounds for determinants and characteristic polynomials. *SIAM J. Matrix Anal. Appl.* 30, 2 (2008), 762–776.

[35] Seny Kamara and Charalampos Papamanthou. 2013. Parallel and dynamic searchable symmetric encryption. In *Financial Cryptography and Data Security: 17th International Conference, FC 2013, Okinawa, Japan, April 1-5, 2013, Revised Selected Papers 17*. Springer, 258–274.

[36] Adam R Klivans and Alex Sherstov. 2006. Cryptographic hardness results for learning intersections of halfspaces. In *Proc. 47 IEEE Symp. on Foundations of Computer Science*. Citeseer.

[37] Kasper Green Larsen and Jelani Nelson. 2017. Optimality of the Johnson-Lindenstrauss lemma. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 633–638.

[38] Yann LeCun, Yoshua Bengio, et al. 1995. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks* 3361, 10 (1995), 1995.

[39] Kevin Lewi and David J Wu. 2016. Order-revealing encryption: New constructions, applications, and lower bounds. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 1167–1178.

[40] Junyi Li and Heng Huang. 2020. Faster secure data mining via distributed homomorphic encryption. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2706–2714.

[41] Keng-Pei Lin, Yi-Wei Chang, and Ming-Syan Chen. 2015. Secure support vector machines outsourcing with random linear transformation. *Knowledge and Information Systems* 44, 1 (2015), 147–176.

[42] Zhijian Liu, Zhanghao Wu, Chuang Gan, Ligeng Zhu, and Song Han. 2020. DataMix: Efficient Privacy-Preserving Edge-Cloud Inference. In *Computer Vision – ECCV 2020*. Springer International Publishing, 578–595.

[43] Kiwan Maeng, Chuan Guo, Sanjay Kariyappa, and G Edward Suh. 2023. Bounding the Invertibility of Privacy-preserving Instance Encoding using Fisher Information. *arXiv preprint arXiv:2305.04146* (2023).

[44] Raluca Ada Popa, Frank H Li, and Nickolai Zeldovich. 2013. An ideal-security protocol for order-preserving encoding. In *2013 IEEE Symposium on Security and Privacy*. IEEE, 463–477.

[45] Alfréd Rényi. 1961. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, Vol. 4. University of California Press, 547–562.

[46] Sina Sajadmanesh, Ali Shahin Shamsabadi, Aurélien Bellet, and Daniel Gatica-Perez. 2023. Gap: Differentially private graph neural networks with aggregation perturbation. In *USENIX Security 2023-32nd USENIX Security Symposium*.

[47] Ohad Shamir. 2018. Distribution-specific hardness of learning neural networks. *The Journal of Machine Learning Research* 19, 1 (2018), 1135–1163.

[48] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*. IEEE, 3–18.

[49] Emil Stefanov, Charalampos Papamanthou, and Elaine Shi. 2013. Practical dynamic searchable encryption with small leakage. *Cryptology ePrint Archive* (2013).

[50] Anvith Thudi, Ilia Shumailov, Franziska Boenisch, and Nicolas Papernot. 2022. Bounding membership inference. *arXiv preprint arXiv:2202.12232* (2022).

[51] Daniel Ting, Stephen E Fienberg, and Mario Trottini. 2008. Random orthogonal matrix masking methodology for microdata release. *International Journal of Information and Computer Security* 2, 1 (2008), 86–105.

[52] Florian Tramer and Dan Boneh. 2020. Differentially private learning needs better features (or much more data). *arXiv preprint arXiv:2011.11660*.

[53] Joel A Tropp. 2012. User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics* 12 (2012), 389–434.

[54] Leslie G Valiant. 1984. A theory of the learnable. *Commun. ACM* 27, 11 (1984), 1134–1142.

[55] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[56] Di Wang, Hanshen Xiao, Srinivas Devadas, and Jinhui Xu. 2020. On differentially private stochastic convex optimization with heavy-tailed data. In *International Conference on Machine Learning*. PMLR, 10081–10091.

[57] Larry Wasserman and Shuheng Zhou. 2010. A statistical framework for differential privacy. *J. Amer. Statist. Assoc.* 105, 489 (2010), 375–389.

[58] Hanshen Xiao and Srinivas Devadas. 2023. PAC Privacy: Automatic Privacy Measurement and Control of Data Processing. In *Advances in Cryptology–CRYPTO 2023: 43rd Annual International Cryptology Conference*. arxiv:2210.03458.

[59] Hanshen Xiao, Jun Wan, and Srinivas Devadas. 2023. Geometry of Sensitivity: Twice Sampling and Hybrid Clipping in Differential Privacy with Optimal Gaussian Noise and Application to Deep Learning. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*. 2636–2650.

[60] Hanshen Xiao, Zihang Xiang, Di Wang, and Srinivas Devadas. 2023. A theory to instruct differentially-private learning via clipping bias reduction. In *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2170–2189.

[61] Xiaokui Xiao and Yufei Tao. 2008. Output perturbation with query relaxation. *Proceedings of the VLDB Endowment* 1, 1 (2008), 857–869.

[62] Adam Yala, Homa Esfahanizadeh, Rafael GL D' Oliveira, Ken R Duffy, Manya Ghobadi, Tommi S Jaakkola, Vinod Vaikuntanathan, Regina Barzilay, and Muriel Medard. 2021. NeuraCrypt: Hiding Private Health Data via Random Neural Networks for Public Training. *arXiv preprint arXiv:2106.02484* (2021).

[63] Adam Yala, Victor Quach, Homa Esfahanizadeh, Rafael GL D'Oliveira, Ken R Duffy, Muriel Médard, Tommi S Jaakkola, and Regina Barzilay. 2022. Syfer: Neural obfuscation for private data release. *arXiv preprint arXiv:2201.12406* (2022).

[64] Chengliang Zhang, Suyi Li, Junzhe Xia, Wei Wang, Feng Yan, and Yang Liu. 2020. Batchcrypt: Efficient homomorphic encryption for cross-silo federated learning. In *2020 {USENIX} Annual Technical Conference ({USENIX}{ATC} 20)*. 493–506.

[65] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2018. mixup: Beyond Empirical Risk Minimization. In *International Conference on Learning Representations*.

[66] Liang Zhao and Liqun Chen. 2018. Sparse matrix masking-based non-interactive verifiable (outsourced) computation, revisited. *IEEE transactions on dependable and secure computing* 17, 6 (2018), 1188–1206.

A PROOF OF THEOREM 1

We first show the construction of the PAC learnable concept class $\tilde{C}$. Without loss of generality, suppose the feature domain $\mathcal{X} = \mathcal{X}_a \cup \mathcal{X}_b$, where $\mathcal{X}_a$ and $\mathcal{X}_b$ are disjoint and $\mathcal{X}_b = \{x_{b,1}, \cdots, x_{b,v}\}$ of v elements. Now, for the given PAC learnable concept class $C = \{h_0, h_1\}$, we assume $h_{1,2}$ is defined on $\mathcal{X}_a$ to $\{0, 1\}$. Now, we construct $\tilde{C}$ based on C. First, within $\mathcal{X}_a$, for each $j = 1, 2, \cdots, 2^v$, and any $x \in \mathcal{X}_a$, we set $\tilde{h}_j(x) = h_1(x)$. As for the evaluation over $\mathcal{X}_b$, let $\{z_1, z_2, ..., z_{2^v}\}$ be the set which contains all the selections of v-dimensional binary vectors $\{0, 1\}^v$. Then, for each $j \in \{1, 2, \cdots, 2^v\}$, we set $\{\tilde{h}_j(x_{b,1}), \cdots, \tilde{h}_j(x_{b,2^v})\} = z_j$. In addition, given the distribution D_0 defined on $\mathcal{X}_a$, we define D_1 in a mixture form $\mathsf{D}_1 = p \cdot \mathsf{D}_0 + (1-p) \cdot \mathcal{U}_{\mathcal{X}_b}$ for some constant p to determined, and $\mathcal{U}_{\mathcal{X}_b}$ is a uniform distribution over $\mathcal{X}_b$. It is not hard to verify that $\mathbb{E}_{x\sim \mathsf{D}_1}\mathbb{E}_j[h_j(x)] = p\mathbb{E}_{x\sim \mathsf{D}_0}[h_0(x)] + (1-p)\mathbb{E}_{x\sim \mathsf{D}_1}\mathbb{E}_j[\tilde{h}_j(x)] = 0.5$ for randomly-selected $j \in \{1, 2, \cdots, 2^v\}$.

$\mathcal{X}_b$ captures the hardcore samples to the learning problem $\tilde{C}$. Given that the labelling is fully random over $\mathcal{X}_b$, one can only memorize rather than learn it. The encoded version of samples from $\mathcal{X}_b$ also behaves as the hardcore data in the transformed learning task. However, given v is finite and the samples from $\mathcal{X}_a$ are learnable based on the assumption of C, $\tilde{C}$ still satisfies PAC learnability: once the number of n (the size of training set S) is sufficiently large ($\gg v$), where most samples in $\mathcal{X}_b$ have been included in S, we can achieve arbitrarily high accuracy. Through this construction, we can theoretically show $\tilde{C}$ is a more challenging problem compared to C, which forms the foundation of the proof.

Next, we describe the distinguishing attack. For the transformed dataset $T(S_b, \theta)$ returned by the user, the adversary randomly splits it into two parts, denoted by $S^1_{b,T}$ and $S^2_{b,T}$: $S^1_{b,T}$ contains m_0 encoded samples and $S^2_{b,T}$ contains the remaining $(n - m_0)$ ones. The adversary then applies Alg on the first part $S^1_{b,T}$, and tests the returned model $\hat{h}$ on $S^2_{b,T}$. In the case of $b = 0$, based on the assumption, with probability at least 3/4, $\mathsf{Alg}(S^1_{0,T})$ will return some $\hat{h}$ such that $\Pr_{x\sim \mathsf{D}_0}\left[\hat{h}(T_X(x), \theta_X) = T_Y(h(x), \theta_Y)\right] \geq 0.5 + \lambda$. From the Hoeffding inequality, for such an $\hat{h}$, it is of probability at least $1 - e^{-2t^2(n-m_0)}$ that the test accuracy on the $(n - m_0)$ i.i.d. samples of $S^2_{0,T}$ is no less than $0.5 + \lambda - t$ for any $t > 0$. When we set $t = \lambda/2$, conditional on $\Pr_{x\sim \mathsf{D}_0}\left[\hat{h}(T_X(x), \theta_X) = T_Y(h(x), \theta_Y)\right] \geq 0.5+\lambda$, the test accuracy of $\hat{h}$ on $S^2_{0,T}$ is no less than $0.5 + \lambda/2$ with probability at least $1 - e^{-\lambda^2/2\cdot(n-m_0)}$.

When $b = 1$, still by the Hoeffding inequality, we know that with probability $1 - e^{-2t^2(n-m_0)}$, a fraction at least $\eta = (1 - p) - t$ of samples in $S^2_{1,T}$ are from $\mathcal{X}_b$ with random labels. Therefore, for any fixed algorithm Alg, when we apply the $\hat{h}$ returned by $\mathsf{Alg}(S^1_{1,T})$ on $S^2_{1,T}$, it cannot predict those random labels better than random guessing. To be formal, given m' elements with random labelling, with probability at least $(1 - e^{-2t'^2 m'})$, any fixed algorithm cannot predict $(0.5 + t')$ fraction of their labels correctly. Therefore, even if the returned model has perfect accuracy on samples from $\mathcal{X}_a$, with probability at least $1 - e^{-2t^2(n-m_0)} - e^{-2t'^2(1-p-t)(n-m_0)}$, the empirical test accuracy on $S^2_{2,T}$ is at most $1 - (1 - p - t)(0.5 - t')$. If we select $t = p = \lambda/4$ and $t' = \lambda/4$, then with probability at least $1 - e^{-(n-m_0)\lambda^2/8} - e^{-(1-\lambda/2)\lambda^2(n-m_0)/8}$, the empirical test accuracy on $S^2_{2,T}$ is no bigger than $0.5 + \lambda/2$. Finally, it is noted that when one generates n i.i.d. samples from D_1, the probability that the elements selected from $\mathcal{X}_b$ are all distinct is lower bounded by $(1 - n/v)^n$. By a union bound, we have the theorem proposed.

B PROOF OF COROLLARY 1

(6) immediately follows from (2), given that with a random guessing on X, the probability that one can positively identify the participation of an individual is $q = n/N$. As for $\bar{\delta}$, let $\delta^{\geq j}$ represent the posterior chance that the adversary can correctly identify at least j memberships of selected X. Let $\delta_o^{\geq j}$ be the optimal *a priori* success rate before observing $\mathcal{M}(X)$. Given the normalized sampling strategy, where we randomly select n_0 samples from each class, it is not hard to verify that $1 - \delta_o^{\geq j} = \sum_{l=j}^{n} \sum_{\{p_{[1:c]}\}=l} \prod_{z=1}^{c} \left(\binom{n_0}{p_z}\binom{n_0}{n_0-p_z}/\binom{N_0}{n_0}\right)$. Now, by (2), we have

$$\mathcal{D}_{KL}(\mathbf{1}_{\delta^{\geq j}} \| \mathbf{1}_{\delta_o^{\geq j}}) = \delta^{\geq j} \log(\frac{\delta^{\geq j}}{\delta_o^{\geq j}}) + (1 - \delta^{\geq j}) \log(\frac{1 - \delta^{\geq j}}{1 - \delta_o^{\geq j}})$$
$$\geq -(1 - \delta^{\geq j}) \log(1 - \delta_o^{\geq j}) - \log(2) - \delta^{\geq j} \log(\delta_o^{\geq j}),$$

where we use the fact that $-\delta^{\geq j} \log \delta^{\geq j} - (1-\delta^{\geq j}) \log(1-\delta^{\geq j}) \leq \log(2)$. Therefore, given $\delta^{\geq j} \log(\delta_o^{\geq j}) \leq 0$, combining with (2),

$$1 - \delta^{\geq j} \leq \frac{\mathsf{MI}(X; \mathcal{M}(X)) + \log(2)}{-\log(1 - \delta_o^{\geq j})}. \tag{15}$$

Now, by Fubini's theorem on expectation, we know that the expected number of memberships that an adversary can recover in X equals $\sum_{j=1}^{n}(1 - \delta^{\geq j})$, which is upper bounded by (15). Substituting the expression of $\delta_o^{\geq j}$, the claim follows.

C PROOF OF THEOREM 2

We first prove the following fact:

$$\mathcal{MI}(X; XW + B) = \mathbb{E}_X\left[\mathcal{D}_{KL}(\mathsf{P}_{XW+B} \| \mathsf{P}_B | X)\right] - \mathcal{D}_{KL}(\mathsf{P}_{XW+B} \| \mathsf{P}_B). \tag{16}$$

Let $Z = XW + B$ for simplicity, from the definition of mutual information, we have

$$\mathcal{MI}(X; XW + B) = \mathcal{D}_{KL}(\mathsf{P}_{X,XW+B} \| \mathsf{P}_X \mathsf{P}_{XW+B})$$
$$= \int_{X_0,Z_0} \mathbb{P}(X = X_0, Z = Z_0) \log \frac{\mathbb{P}(Z = Z_0 | X = X_0)\mathbb{P}(Z = Z_0 | X = 0)}{\mathbb{P}(Z = Z_0 | X = 0)\mathbb{P}(Z = Z_0)}$$
$$= \Big(\int \mathcal{D}_{KL}(\mathsf{P}_{X_0 W+B} \| \mathsf{P}_B)\mathbb{P}(X = X_0)\, dX_0\Big) - \mathcal{D}_{KL}(\mathsf{P}_{XW+B} \| \mathsf{P}_B). \tag{17}$$

When $W \in \mathbb{R}^{d_0 \times d}$ and $B \in \mathbb{R}^{n \times d}$ are two independent Gaussian random matrices, where each entry of W is i.i.d. $\mathcal{N}(0, 1)$ and each column of B is i.i.d. in some multivariate Gaussian distribution $\mathcal{N}(\mathbf{0}, \Sigma_B)$ for some non-singular covariance Σ_B, then each column of $X_0 W + B$ is i.i.d. in $\mathcal{N}(\mathbf{0}, X_0 X_0^T + \Sigma_B)$. Therefore, if we reshape $X_0 W + B$ into a $1 \times nd$ vector, it is still a multivariate Gaussian whose covariance matrix is in a d-block-diagonal form, $\mathrm{Diag}(X_0 X_0^T + \Sigma_B, \cdots, X_0 X_0^T + \Sigma_B)$, denoted by $\mathrm{Diag}(X_0 X_0^T + \Sigma_B, d)$.

Therefore, by the KL-divergence between multivariate Gaussian,

$$\begin{aligned}
&\mathcal{D}_{KL}\big(\mathsf{P}_{XW+B}\|\mathsf{P}_B|X=X_0)\\
&=\mathcal{D}_{KL}(\mathcal{N}(\mathbf{0},\mathrm{Diag}(X_0X_0^T+\Sigma_B,d))\|\mathcal{N}(\mathbf{0},\mathrm{Diag}(\Sigma_B,d)))\\
&=\frac{d}{2}\big\{\mathrm{Tr}\big(\Sigma_B^{-1}X_0X_0^T)\big)-\log\det\big(\boldsymbol{I}_n+\Sigma_B^{-1}X_0X_0^T)\big)\big\}.
\end{aligned}\tag{18}$$

Now, we turn to handle the other term $\mathcal{D}_{KL}(\mathsf{P}_{XW+B}\|\mathsf{P}_B)$. It is noted that we have the following identity [58],

$$\begin{aligned}
&\mathcal{D}_{KL}(\mathsf{P}_{XW+B}\|\mathsf{P}_B)\\
&=\mathcal{D}_{KL}(\mathsf{P}_{XW+B}\|\mathsf{P}_{\mathrm{Gau}(XW+B)})+\mathcal{D}_{KL}(\mathsf{P}_{\mathrm{Gau}(XW+B)}\|\mathsf{P}_B).
\end{aligned}\tag{19}$$

Here, $\mathrm{Gau}(a)$ represents a Gaussian distribition of the same mean and covariance as that of a. Thus, by (16), we have

$$\begin{aligned}
&\mathcal{MI}(X;XW+B)\\
&=\mathbb{E}_X\big[\mathcal{D}_{KL}(\mathsf{P}_{XW+B}\|\mathsf{P}_B|X)\big]\\
&\qquad-\mathcal{D}_{KL}(\mathsf{P}_{XW+B}\|\mathsf{P}_{\mathrm{Gau}(XW+B)})-\mathcal{D}_{KL}(\mathsf{P}_{\mathrm{Gau}(XW+B)}\|\mathsf{P}_B)\\
&\le\mathbb{E}_X\big[\mathcal{D}_{KL}(\mathsf{P}_{XW+B}\|\mathsf{P}_B|X)\big]-\mathcal{D}_{KL}(\mathsf{P}_{\mathrm{Gau}(XW+B)}\|P_B)\\
&=\frac{d}{2}\big\{\mathbb{E}_X\big[\mathrm{Tr}(\Sigma_B^{-1}XX^T)-\log\det(\boldsymbol{I}_n+\Sigma_B^{-1}XX^T)\big]\\
&\qquad-\big(\mathrm{Tr}(\mathbb{E}_X[\Sigma_B^{-1}XX^T])-\log\det(\mathbb{E}_X[\boldsymbol{I}_n+\Sigma_B^{-1}XX^T])\big)\big\}\\
&=\frac{d}{2}\big\{\log\det(\mathbb{E}_X[\boldsymbol{I}+\Sigma_B^{-1}XX^T])-\mathbb{E}_X\big[\log\det(\boldsymbol{I}+\Sigma_B^{-1}XX^T)\big]\big\}.
\end{aligned}\tag{20}$$

(20) provides the Type (II) upper bound in (4). In the following, we show the Type (I) bound. It is noted that in (17), we can replace P_B with an arbitrary distribution. If we select P_{XW+B} instead, since KL-divergence is non-negative, we have that

$$\mathcal{MI}(X;XW+B)\le\int_{X_0}\mathcal{D}_{KL}(\mathsf{P}_{X_0W+B}\|\mathsf{P}_{XW+B})\mathbb{P}(X=X_0).\tag{21}$$

On the other hand, P_{XW+B} is indeed a Gaussian mixture which can be written as $\sum_X\mathbb{P}(X)\mathcal{N}(\mathbf{0},XX^T+\Sigma_B)$. We can also rewrite P_{X_0W+B} as $\sum_X\mathbb{P}(X)\mathcal{N}(\mathbf{0},X_0X_0^T+\Sigma_B)$. For simplicity, we use $\mathrm{G}_B(X)$ to represent the distribution $\mathcal{N}(\mathbf{0},XX^T+\Sigma_B)$ for any given X. Therefore, due to the convexity of KL-divergence, $\mathcal{D}_{KL}(\lambda\mathsf{P}_1+(1-\lambda)\mathsf{P}_2\|\lambda\mathsf{Q}_1+(1-\lambda)\mathsf{Q}_2)\le\lambda\mathcal{D}_{KL}(\mathsf{P}_1\|\mathsf{Q}_1)+(1-\lambda)\mathcal{D}_{KL}(\mathsf{P}_2\|\mathsf{Q}_2)$, and

$$\mathcal{MI}(X;XW+B)\le\mathbb{E}_X\mathbb{E}_{X'}\mathcal{D}_{KL}\big(\mathrm{G}_B(X)\|\mathrm{G}_B(X')\big),\tag{22}$$

where X' is distributed the same as X. Substituting the expression of KL-divergence between two Gaussians, (22) produces the Type (I) upper bound.

D PROOF OF THEOREM 3

With a similar reasoning as (17), we have

$$\begin{aligned}
\mathcal{MI}(\mathbf{1}_{u_i};XW+B)&=\mathcal{D}_{KL}(\mathsf{P}_{X,XW+B}\|\mathsf{P}_X\otimes\mathsf{P}_{XW+B})\\
&=\mathbb{E}_{\mathbf{1}_{u_i}}\mathcal{D}_{KL}(\mathsf{P}_{XW+B}\|\mathsf{P}_B|\mathbf{1}_{u_i})-\mathcal{D}_{KL}(\mathsf{P}_{XW+B}\|\mathsf{P}_B).
\end{aligned}\tag{23}$$

It is noted that the covariance of XW can still be written as $q\mathbb{E}_{X_i}X_iX_i^T$ $+(1-q)\mathbb{E}_{X_{-i}}X_{-i}X_{-i}^T$, where X_i is a random set containing u_i and X_{-i} is a random set without u_i, for $q=n/N$ equalling the sampling rate. Therefore, when we apply the same trick to introduce $\mathrm{Gau}(XW+B)$, the trace terms in (20) still cancel out, where

$$\begin{aligned}
\mathcal{MI}(\mathbf{1}_{u_i};XW+B)&\le q\mathbb{E}_{X_i}\big[\mathcal{D}_{KL}(\mathsf{P}_{X_iW+B}\|\mathsf{P}_B|X_i)\big]\\
&\qquad+(1-q)\mathbb{E}_{X_{-i}}\big[\mathcal{D}_{KL}(\mathsf{P}_{X_{-i}W+B}\|\mathsf{P}_B|X_{-i})\big]\\
&\qquad-\mathcal{D}_{KL}(\mathsf{P}_{XW+B}\|\mathsf{P}_{\mathrm{Gau}(XW+B)})\\
=\frac{d}{2}\big\{&q\mathbb{E}_{X_i}\big[\mathrm{Tr}(\Sigma_B^{-1}X_iX_i^T)-\log\det(\boldsymbol{I}_n+\Sigma_B^{-1}XX^T)\big]\\
+(1-q)&\mathbb{E}_{X_{-i}}\big[\mathrm{Tr}(\Sigma_B^{-1}X_{-i}X_{-i}^T)-\log\det(\boldsymbol{I}_n+\Sigma_B^{-1}X_{-i}X_{-i}^T)\big]\\
&-\big(\mathrm{Tr}(\mathbb{E}_X[\Sigma_B^{-1}XX^T])-\log\det(\mathbb{E}_X[\boldsymbol{I}_n+\Sigma_B^{-1}XX^T])\big)\big\},
\end{aligned}\tag{24}$$

which can be further written as

$$\begin{aligned}
&\frac{d}{2}\big\{\log\det(\mathbb{E}_X[\boldsymbol{I}+\Sigma_B^{-1}XX^T])-q\mathbb{E}_{X_i}\big[\log\det(\boldsymbol{I}+\Sigma_B^{-1}X_iX_i^T)\big]\\
&\qquad-(1-q)\mathbb{E}_{X_{-i}}\big[\log\det(\boldsymbol{I}+\Sigma_B^{-1}XX^T)\big]\big\},
\end{aligned}\tag{25}$$

which produces the Type (II) bound in (5). As for Type (I), stemming from (21), given that $\mathsf{P}_{XW+B}=q\mathbb{E}_{X_i}\mathrm{G}_B(X_i)+(1-q)\mathbb{E}_{X_{-i}}\mathrm{G}_B(X_{-i})$, by the convexity of KL-divergence,

$$\begin{aligned}
&\mathcal{MI}(\mathbf{1}_{u_i};XW+B)\\
&\le q\big(\mathcal{D}_{KL}\big(\mathbb{E}_{X_i}\mathrm{G}_B(X_i)\|q\mathbb{E}_{X_i}\mathrm{G}_B(X_i)+(1-q)\mathbb{E}_{X_{-i}}\mathrm{G}_B(X_{-i})\big)\\
&+(1-q)\big(\mathcal{D}_{KL}\big(\mathbb{E}_{X_{-i}}\mathrm{G}_B(X_{-i})\|q\mathbb{E}_{X_i}\mathrm{G}_B(X_i)+(1-q)\mathbb{E}_{X_{-i}}\mathrm{G}_B(X_{-i})\big)\\
&\le q(1-q)\mathbb{E}_{X_i}\mathbb{E}_{X_{-i}}\big(\mathcal{D}_{KL}\big(\mathrm{G}_B(X_i)\|\mathrm{G}_B(X_{-i})\big)\big)\\
&\qquad+q(1-q)\mathbb{E}_{X_{-i}}\mathbb{E}_{X_i}\big(\mathcal{D}_{KL}\big(\mathrm{G}_B(X_{-i})\|\mathrm{G}_B(X_i)\big),
\end{aligned}\tag{26}$$

which produces the Type (I) bound in (5).

E PROOF OF THEOREM 4

In the scenario when we further incorporate data mixing and permutation, where the encoded data becomes $\Pi MXW+B$, for any given $X=X_0$, $M=M_0$ and $\Pi=\Pi_0$, $\Pi_0M_0X_0W+B$ is a multivariate Gaussian in a form $\mathsf{P}_{\Pi_0MX_0W}=\mathrm{G}_B(\Pi_0M_0X_0)$. Without loss of generality, we restrict Π to only permute the elements within each category locally. It is noted that for M' and Π' i.i.d. to M and Π, respectively,

$$\begin{aligned}
&\mathcal{MI}(X;\Pi MXW+B)\\
&=\mathbb{E}_{M,\Pi}\big[\mathcal{D}_{KL}(\mathsf{P}_{X,\Pi MXW+B}\|\mathsf{P}_X\mathsf{P}_{\Pi'M'XW+B}|M,\Pi)\big]\\
&=\mathbb{E}_{M,\Pi,X}\mathcal{D}_{KL}(\mathsf{P}_{\Pi MXW+B}\|\mathsf{P}_B|M,\Pi,X)-\mathcal{D}_{KL}(\mathsf{P}_{\Pi MXW+B}\|\mathsf{P}_B)\\
&=\mathbb{E}_{M,\Pi,X}\mathcal{D}_{KL}(\mathrm{G}_B(\Pi MX)\|\mathsf{P}_B|M,\Pi,X)\\
&\qquad\qquad-\mathcal{D}_{KL}(\mathbb{E}_{\Pi,M,X}\mathrm{G}_B(\Pi MX)\|\mathsf{P}_B).
\end{aligned}\tag{27}$$

With a similar reasoning as (20), (27) is upper bounded by

$$\begin{aligned}
&\mathbb{E}_{\Pi,M,X}\big[\mathcal{D}_{KL}(\mathbb{E}_{\Pi,M}\mathrm{G}_B(\Pi MX)\|\mathsf{P}_B|X,M,\Pi)\big]\\
&\qquad-\mathcal{D}_{KL}(\mathsf{P}_{\mathrm{Gau}(\Pi MXW+B)}\|P_B)\\
&=\frac{d}{2}\big\{\mathbb{E}_{\Pi,M,X}\big[\mathrm{Tr}(\Sigma_B^{-1}(\Pi MX)(\Pi MX)^T)\\
&\qquad-\log\det(\boldsymbol{I}+\Sigma_B^{-1}(\Pi MX)(\Pi MX)^T)\big]\\
&\qquad-\big(\mathrm{Tr}(\mathbb{E}_{\Pi,M,X}[\Sigma_B^{-1}(\Pi MX)(\Pi MX)^T])\\
&\qquad-\log\det(\mathbb{E}_{X,\Pi,M}[\boldsymbol{I}_n+\Sigma_B^{-1}(\Pi MX)(\Pi MX)^T])\big)\big\}
\end{aligned}\tag{28}$$

$$\begin{aligned}
&=\frac{d}{2}\big\{\log\det(\mathbb{E}_{\Pi,M,X}[\boldsymbol{I}+\Sigma_B^{-1}(\Pi MX)(\Pi MX)^T])\\
&\qquad-\mathbb{E}_{\Pi,M,X}\big[\log\det(\boldsymbol{I}+\Sigma_B^{-1}(\Pi MX)(\Pi MX)^T)\big]\big\},
\end{aligned}$$

which produces the Type (II) upper bound in (7).

With the same idea, stemming from (21)

$$\begin{aligned}
&\mathcal{MI}(X;\Pi MXW+B)\\
&\leq \int_{X_0} \mathcal{D}_{KL}(\mathrm{P}_{\Pi MX_0W+B}\|\mathrm{P}_{\Pi MXW+B})\mathbb{P}(X=X_0)\\
&\leq \mathbb{E}_{M,\Pi}\Big[\int_{X_0} \mathcal{D}_{KL}(\mathrm{G}_B(\Pi MX_0)\|\mathbb{E}_X \mathrm{G}_B(\Pi MX))\mathbb{P}(X=X_0)\Big]\\
&\leq \mathbb{E}_{\Pi,M,X,X'}\mathcal{D}_{KL}\big(\mathrm{G}_B(\Pi MX)\|\mathrm{G}_B(\Pi MX')\big),
\end{aligned} \tag{29}$$

where we apply the convexity of KL-divergence above. (29) produces the Type (I) bound in (7).

F PROOF OF THEOREM 5

The proof of the Type (II) bound in (8) is straightforward by combining the reasoning in the proof of Theorem 3 and 4. We focus on the Type (I) bound in (8) in the following.

$$\begin{aligned}
&\mathcal{MI}(\mathbf{1}_{u_i};\Pi MXW+B) \leq q\mathcal{D}_{KL}\big(\mathbb{E}_{\Pi,M,X_i}\mathrm{G}_B(\Pi MX_i)\\
&\qquad \|q\mathbb{E}_{\Pi,M,X_i}\mathrm{G}_B(\Pi MX_i)+(1-q)\mathbb{E}_{\Pi,M,X_{-i}}\mathrm{G}_B(\Pi MX_{-i})\big)\\
&\quad +(1-q)\mathcal{D}_{KL}\big(\mathbb{E}_{\Pi,M,X_{-i}}\mathrm{G}_B(\Pi MX_{-i})\\
&\qquad \|q\mathbb{E}_{\Pi,M,X_i}\mathrm{G}_B(\Pi MX_i)+(1-q)\mathbb{E}_{\Pi,M,X_{-i}}\mathrm{G}_B(\Pi MX_{-i})\big)\\
&\quad \leq q(1-q)\mathcal{D}_{KL}\big(\mathbb{E}_{\Pi,M,X_i}\mathrm{G}_B(\Pi MX_i)\|\mathbb{E}_{\Pi,M,X_{-i}}\mathrm{G}_B(\Pi MX_{-i})\big)\\
&\quad +q(1-q)\mathcal{D}_{KL}\big(\mathbb{E}_{\Pi,M,X_{-i}}\mathrm{G}_B(\Pi MX_{-i})\|\mathbb{E}_{\Pi,M,X_i}\mathrm{G}_B(\Pi MX_i)\big).
\end{aligned} \tag{30}$$

Thus, we can consider the following pairing. For any selection of X_i, we consider X_{-i} where X_i and X_{-i} only differ in the one datapoint, i.e., u_i. For given X_i and X_{-i}, given any selection of data mixing M_i and permutation Π_i, there exists a bijection $(M_i,\Pi_i)\leftrightarrow(M_{-i},\Pi_{-i})$ such that $\Pi_i M_i X_i$ and $\Pi_{-i}M_{-i}X_{-i}$ are identical if we replace u_i in X_i by the differing datapoint in X_{-i}. This is equivalent to modelling that the *closest pair* $X_i \overset{c}{\sim} X_{-i} \in \mathbb{R}^{n\times d_0}$ are neighboring, which differ in the first row (X_i's first row is u_i and X_{-i}'s first row is the differing datapoint), while X_i and X_{-i} share the identical second to the n-th rows. We then apply the identical data mixing M and permutation Π and let $\tilde{X}_i = \Pi MX_i \sim \tilde{X}_{-i} = \Pi MX_{-i}$. Moreover, it is noted that such pairing is symmetric where each selection X_i can produce $(N-n)$ many closest pairs. Since the total number $\binom{N-1}{n}$ of different selections of X_{-i} is $(N-n)/n$ times than that $\binom{N-1}{n-1}$ of different selections of X_i, we can virtually duplicate X_{-i} n times and consider the virtual Gaussian mixture distribution.

With this more fine-grained pairing and the convexity of KL-divergence, (30) is further bounded by

$$\begin{aligned}
&\mathcal{MI}(\mathbf{1}_{u_i};\Pi MXW+B)\\
&\leq q(1-q)\mathbb{E}_{\Pi,M,X_i\overset{c}{\sim}X_{-i}}\big\{\mathcal{D}_{KL}\big(\mathrm{G}_B(\Pi MX_i)\|\mathrm{G}_B(\Pi MX_{-i})\big)\\
&\qquad\qquad +\mathcal{D}_{KL}\big(\mathrm{G}_B(\Pi MX_{-i})\|\mathrm{G}_B(\Pi MX_i)\big)\big\}.
\end{aligned} \tag{31}$$

(31) produces the Type (I) bound in (8).

G PROOF OF THEOREM 6

Before we start, we first introduce several important matrix inequalities. First, for any matrix $A\in\mathbb{R}^{m\times d_0}$, $\|AA^T\|_2$, i.e., the largest eigenvalue of AA^T, is upper bounded by

$$\|AA^T\|_2 \leq \sqrt{m}\|AA^T\|_\infty, \tag{32}$$

where $\|AA^T\|_\infty$ is the largest entry of AA^T in absolute value. In the following, we introduce *Von Neumann's trace inequality*. For two $m\times m$ positive semidefinite matrices AA^T and RR^T, whose eigenvalues are $\{e_{A,1}\geq\cdots\geq e_{A,m}\}$ and $\{e_{R,1}\geq\cdots\geq e_{R,m}\}$, in a non-ascending order, respectively,

$$\sum_{j=1}^{m} e_{A,j}e_{B,m-j} \leq \mathrm{Tr}(AA^T\cdot BB^T) \leq \sum_{j=1}^{m} e_{A,j}e_{B,j}. \tag{33}$$

Finally, we will also use the following fact about determinant perturbation [34]. For two $m\times m$ matrices C and E,

$$|\det(C)-\det(C+E)| \leq m\max\{\|C\|_2,\|C+E\|_2\}\|E\|_2. \tag{34}$$

Now, we are ready to prove the theorem. Given that $\|u_i\|_2\leq 1$, we have that after data mixing, the l_2 norm of each row of MX for any subsampled X and mixing matrix M is still bounded by 1, and therefore for any permutation Π, $\|\Pi MX(\Pi MX)^T\|_\infty \leq 1$. On one hand, given that $\Sigma_B = \sigma^2\boldsymbol{I}_m$, we have that the largest eigenvalue of $\Sigma_{\Pi MX_i,B}$ is upper bounded by $\|\Pi MX_i(\Pi MX_i)^T\|_2 + \sigma^2 \leq \sqrt{m}+\sigma^2$ by (32). On the other hand, due to the form of a positive-definite matrix, the smallest eigenvalue of $\Sigma_{\Pi MX_i,B}$ is lower bounded by σ^2. Therefore, we have a global upper bound of $\log(\det(\Sigma_{\Pi MX_i,B}\Sigma^{-1}_{\Pi MX_{-i},B})) \leq m\log(\frac{\sqrt{m}+\sigma^2}{\sigma^2})$. In the following, we consider the upper bound of $\mathrm{Tr}(\Sigma^{-1}_{\Pi MX_i,B}\Sigma_{\Pi MX_{-i},B})$.

With similar reasoning and (33),

$$\mathrm{Tr}(\Sigma^{-1}_{\Pi MX_i,B}\Sigma_{\Pi MX_{-i},B}) \leq \frac{1}{\sigma^2}\mathrm{Tr}(\Sigma_{\Pi MX_{-i},B}) \leq m(\sqrt{m}+\sigma^2)/\sigma^2.$$

Given the global upper bound, the remaining step is to apply a high-probability concentration inequality to describe the confidence interval. With the help of the Hoeffding bound, for i.i.d. numbers $z_1,z_2,\cdots,z_L$ where $z_j\in[a,b]$ and the mean $\mathbb{E}[z_j]=\mu$,

$$\Pr(\mu-\sum_{j=1}^{L} z_j/L \geq \epsilon) \leq e^{-\frac{2\epsilon^2 L}{(b-a)^2}}. \tag{35}$$

By substituting the distribution range of each term to estimate, and applying a union bound on the failure rates, we obtain the high-confidence bound claimed.

Similarly, for the Type (II) upper bound, first given that $\|\boldsymbol{I}_m+\Sigma_B^{-1}\Sigma_{\tilde{X}}\|_2 \leq 1+\sqrt{m}/\sigma^2$, $\log\det(\boldsymbol{I}_m+\Sigma_B^{-1}\Sigma_{\tilde{X}}) \leq m\log(1+\sqrt{m}/\sigma^2)$. The more tricky part is the high-probability bound with respect to the estimation error of $\log\det(\mathbb{E}_{\tilde{X}_i}[\boldsymbol{I}_m+\Sigma_B^{-1}\sum_{\tilde{X}_i}])$. Fortunately, let E be the error from the empirical estimation on $\mathbb{E}_{\tilde{X}_i}[\boldsymbol{I}_m+\Sigma_B^{-1}\sum_{\tilde{X}_i}]$, by (34), we have that $|\det(\mathbb{E}_{\tilde{X}_i}[\boldsymbol{I}_m+\Sigma_B^{-1}\sum_{\tilde{X}_i}])-\det(\mathbb{E}_{\tilde{X}_i}[\boldsymbol{I}_m+\Sigma_B^{-1}\sum_{\tilde{X}_i}]+E)| \leq m(\sqrt{m}+\|E\|_2)\|E\|_2$. On the other hand, we can apply the *Matrix Hoeffding Inequality* [53]: if $A_1A_1^T,\cdots,A_TA_T^T$ are i.i.d. zero-mean $m\times m$ matrices, such that $\|A_jA_j^T\|\leq s^2$ and the mean is μ, then

$$\Pr(\|\sum_{j=1}^{L} A_jA_j^T/T-\mu\| \geq \epsilon) \leq 2m\cdot e^{-\frac{\epsilon^2 L}{8s^2}}. \tag{36}$$

Therefore, in the context of estimation error E for $\mathbb{E}_{\tilde{X}_i}[\boldsymbol{I}_m+\Sigma_B^{-1}\sum_{\tilde{X}_i}]$, by (36) and $\|\boldsymbol{I}_m+\Sigma_B^{-1}\sum_{\tilde{X}_i}\|\leq 1+\sqrt{m}/\sigma^2$, we have that after L trials, $\Pr(\|E\|\geq\epsilon)\leq 2m\cdot e^{-\frac{\epsilon^2 L}{32(1+\sqrt{m}/\sigma^2)^2}}$, and the claim follows.

Algorithm 2 Privacy Guarantee in High-Confidence

```
Input: A labelled data pool $\mathsf{U} = \{u_1, \cdots, u_N\}$with associated labels $\mathsf{V} = \{v_1, \cdots, v_N\}$; sampled dataset size $n$; objective $u_i$; noise covariance $\Sigma_B = \sigma^2 \cdot \boldsymbol{I}_m$; simulation complexity $L$; estimated variance $\eta$.
Independently sample $L$ many normalized $n$ feature sets $X_{i,1}, X_{i,2} \cdots, X_{i,L}$ all containing $u_i$.
For each $X_{i,j}, j = 1, 2, \cdots, L$, randomly sample $u_l$ from $\mathsf{U}$, which has an identical label as $u_i$, but is not included in $X_{i,j}$. Preserving the same ordering, replace $u_i$ in $X_{i,j}$ with $u_l$, which produces $X_{-i,j}$.
Independently generate $L$ masking matrices $\{M_1, \cdots, M_L\}$ and $L$ permutation matrices $\Pi_1, \cdots, \Pi_L$.
Compute the following
    $Q_1 = \sum_{j=1}^{L} \mathrm{Tr}(\Sigma^{-1}_{\Pi_j M_j X_{i,j}, B} \Sigma_{\Pi_j M_j X_{-i,j}, B})$,
    $Q_2 = \sum_{j=1}^{L} \mathrm{Tr}(\Sigma^{-1}_{\Pi_j M_j X_{-i,j}, B} \Sigma_{\Pi_j M_j X_{i,j}, B})$.
Output: $\mathcal{E}_L = \frac{1}{L} \cdot \frac{q(q-1)}{2} (Q_1 + Q_2 - 2m)$.
```

H PROOF OF THEOREM 7

We first introduce the following useful lemma that will help us develop the lower bound of the distance between XW_1 and XW_2 for different matrix maskings.

Lemma 1 (Hanson–Wright inequality [1]). *Let $s \in \mathbb{R}^{d_0}$ be a random vector, which satisfies $\mathbb{E}[x] = \mathbf{0}$ and Assumption 2, then for a matrix $A \in \mathbb{R}^{d_0 \times d_0}$ and any $t \geq 0$,*

$$\Pr(|sAs^T - \mathbb{E}[sAs^T]| > t) \leq 2e^{-\frac{1}{C} \min\{\frac{t^2}{K^4 \|A\|_F^2}, \frac{t}{K^2 \|A\|_F^2}\}}, \tag{37}$$

for some constant C. Here, $\|A\|$ is the operator norm of A and $\|A\|_F$ is the Frobenius norm of A.

It is noted that $\mathbb{E}[xAx^T] = \mathbb{E}[\|xW\|^2] \geq \kappa^2 \|W\|_F^2$, from Assumption 1. Here, $\|W\|_F$ is the Frobenius norm of W, where $\|W\|_F = \sqrt{\sum_{i=1}^{d_0} \sum_{j=1}^{d_0} W^2(i,j)}$. By Lemma 1, if we take $A = WW^T$ into (37), (37) becomes

$$\Pr_x(\|xW\|^2 < \kappa^2 \|W\|_F^2 - t) \leq e^{-\frac{t^2}{CK^4 \|WW^T\|_F^2}}. \tag{38}$$

Let f_{adv} denote the adversary's response. With a similar reasoning as the proof of Corollary 1, we consider the indicator $\rho_2(f_{adv}, W)$ which equals 1 if the adversary successfully approximates the true transformation W such that $\Pr_x(\|xW - f_{adv}(x)\| < \psi) \geq 1 - \tau$. Then, the corresponding posterior success rate $(1 - \delta)$ is upper bounded by

$$1 - \delta \leq \frac{\mathcal{MI}(W; XW + B) + \log(2)}{\log(1/(1 - \delta_o))}, \tag{39}$$

where $(1 - \delta_o)$ is the optimal *a priori* success rate. As for $\mathcal{MI}(W; XW + B)$, it is noted that

$$\begin{aligned} \mathcal{MI}(X, W; XW + B) &= \mathcal{MI}(W; XW + B) + \mathcal{MI}(X; XW + B|W) \\ &= \mathcal{MI}(X; XW + B) + \mathcal{MI}(W; XW + B|X). \end{aligned} \tag{40}$$

Therefore, $\mathcal{MI}(W; XW + B) = \mathcal{MI}(X; XW + B) + \mathcal{H}(XW + B|X) - \mathcal{H}(XW + B|W)$, where $\mathcal{H}$ represents entropy. Since $\mathcal{H}(XW + B|W) \leq \mathcal{H}(XW + B|X, W) = \mathcal{H}(B)$, where conditioning will not increase entropy, we have that $\mathcal{MI}(W; XW + B) \leq \mathcal{MI}(X; XW + B) + \frac{d_0}{2} \mathbb{E}_X \log(\det(\boldsymbol{I} + \frac{XX^T}{\sigma^2}))$. Here, it is noted that $XW + B$ conditional on X is a Gaussian matrix, where each column is i.i.d. multivariate Gaussian $\mathcal{N}(\mathbf{0}, XX^T + \sigma^2 \cdot \boldsymbol{I})$.

In the following, we consider a packing set of W to upper bound the *a priori* success rate $(1 - \delta_o)$.

Lemma 2. *For any two matrices W_0 and W_0', if $\Pr_x \left(\|xW_0 - xW_0'\| < 2\psi_0\right) < 1 - 2\tau_0$, then for an arbitrary function $f(\cdot)$, at most one of the following can hold,*

$$\begin{aligned} &\Pr_{x \sim \mathsf{D}} \left(\|xW_0 - f(x)\| < \psi_0\right) \geq 1 - \tau_0, \\ &\Pr_{x \sim \mathsf{D}} \left(\|xW_0' - f(x)\| < \psi_0\right) \geq 1 - \tau_0. \end{aligned} \tag{41}$$

The proof is straightforward. If both the above inequalities are true, where the function $f(\cdot)$ approximates both W_0 and W_0' well, then we have,

$$\begin{aligned} &\Pr(\|xW_0 - f(x)\| < \psi_0 \wedge \|xW_0' - f(x)\| < \psi_0) \\ &\qquad \leq \Pr(\|xW_0 - xW_0'\| < 2\psi_0). \end{aligned} \tag{42}$$

On the other hand, with a union bound,

$$\begin{aligned} &\Pr\left(\|xW_0 - f(x)\| < \psi_0 \ \wedge \ \|xW_0' - f(x)\| < \psi\right) \\ &\geq \Pr(\|xW_0 - f(x)\| < \psi_0) + \Pr(\|xW_0' - f(x)\| < \psi_0) - 1 \\ &\geq 1 - 2\tau_0. \end{aligned} \tag{43}$$

This contradicts the assumption $\Pr_x(\|xW_0 - xW_0'\| < 2\psi_0) < 1 - 2\tau_0$, and the lemma is proved. Thus, for any adversary-proposing function $f_{adv}(\cdot)$, without loss of generality, we assume that there exists some W_0' such that $\Pr_x \left(\|xW_0' - f_{adv}(x)\| < \psi\right) \geq 1 - \tau$. Then, for any $W_0 \in \mathbb{R}^{d_0 \times d_0}$ such that $\|W_0 - W_0'\|_F^2 = \beta^2$ by Lemma 1, we have

$$\Pr(\|x(W_0 - W_0')\|^2 < \kappa^2 \beta^2 - t) \leq e^{-\frac{t}{CK^4 \beta^4}}. \tag{44}$$

Here, we use the fact that $\|(W_0 - W_0')(W_0 - W_0')^T\|_F^2 \leq \|W_0 - W_0'\|_F^4$. Thus, we may select $\psi = \sqrt{\kappa^2 \beta^2 - t}/2$, $\tau = (1 - e^{-\frac{t^2}{CK^4 \beta^4}})/2$. On the other hand, it also suggests that the optimal rate $(1 - \delta_o)$ is actually upper bounded by $\Pr_w(W \in \mathcal{B}_F(\beta))$, where $\mathcal{B}_F(\beta)$ represents a ball in $\mathbb{R}^{d_0 \times d_0}$ of radius β in Frobenius norm centered at zero, enjoying the maximal probability density for a Gaussian. It is noted that the probability $\Pr_w(W \in \mathcal{B}_F(\beta))$ essentially equals $\Pr(x \leq d^2 - (d^2 - \beta^2))$, for a d^2-degree Chi-square random variable x, which can be further upper bounded by $e^{-((d^2 - \beta^2)/(2d))^2}$.. Thus, $\log(1/\delta_0) \leq (\frac{d^2 - \beta^2}{2d})^2$, and the theorem follows by combining (39) and (40).

I ALGORITHM FOR HIGH-CONFIDENCE SIMULATION

We take the Type (I) bound in (8) as an example. Algorithm 2 is used to estimate the objective mutual information with high confidence.