A Blessing of Dimensionality in Membership Inference through Regularization

Jasper Tan **Daniel LeJeune** **Blake Mason** **Hamid Javadi** **Richard G. Baraniuk**
Rice University

Abstract

Is overparameterization a privacy liability? In this work, we study the effect that the number of parameters has on a classifier's vulnerability to membership inference (MI) attacks. We first demonstrate how the number of parameters of a model induces a privacy–utility trade-off: increasing the number of parameters generally improves generalization performance at the expense of lower privacy. However, remarkably, we then show that if coupled with proper regularization, increasing the number of parameters of a model can actually simultaneously increase *both* its privacy and performance, thereby eliminating the privacy–utility trade-off. Theoretically, we demonstrate this curious phenomenon for logistic regression with ridge regularization in a bi-level feature ensemble setting. Pursuant to our theoretical exploration, we develop a novel leave-one-out analysis tool to precisely characterize the vulnerability of a linear classifier to the optimal membership inference attack. We empirically exhibit this "blessing of dimensionality" for neural networks on a variety of tasks using early stopping as the regularizer.

1 INTRODUCTION

Recently, the machine learning community has been gravitating towards the trend of increasingly overparameterized models, which have been shown both theoretically [1–3] and empirically [4, 5] to generalize better than their smaller counterparts in diverse settings. These findings encourage machine learning system designers to opt for the largest possible model to maximize performance on unseen data.

However, when training machine learning models on sensitive data [6–8], it is also crucial to understand the attendant privacy issues to prevent data leaks. Alarmingly, multiple attacks have been developed in the literature to perform *membership inference* (MI), which extracts information about specific examples in a model's training dataset, even when given only black-box access [9–11].

Is the trend of increasingly overparameterizing models detrimental to privacy? In this paper, we focus on the effect that the number of parameters of a model has on its vulnerability to MI attacks. We study this problem both theoretically and empirically.

We first demonstrate a parameter-wise privacy–utility trade-off: increasing the number of parameters of a model increases its generalization performance while also increasing its vulnerability to MI attacks. We show this theoretically for logistic regression and empirically for support vector machines and neural networks. This corroborates previous empirical [9, 12] and theoretical [13] findings that larger models are less private.

However, we then show that this is not the end of the story between overparameterization and privacy. Remarkably, we discover that if proper *regularization* is incorporated while increasing the number of parameters, the larger model can actually enjoy greater privacy (stronger protection from MI attacks) for the same generalization performance as its smaller counterpart. That is, there is a "blessing of dimensionality," rather than a curse, and more overparameterized models can in fact be more private when paired with regularization. We show this behavior theoretically for logistic regression with ridge regularization and empirically for neural networks with early stopping.

This behavior is due to the fact that regularization induces its own privacy–utility trade-off: beyond a point, increasing regularization provides greater protection from MI attacks while decreasing generalization performance. However, the trade-off induced by regularization for a larger network traces a trajectory of lower MI vulnerability and better generalization performance than the trade-off for a smaller network. That is, larger networks have better regularization-wise privacy–utility trade-offs.

To demonstrate this effect theoretically, we must be able to precisely characterize the output distribution of a model on a fixed training data point over the randomness of all other training data. We overcome this challenge by developing a novel leave-one-out analysis tool based on the convex Gaussian min-max theorem [14, 15] that we apply to high-dimensional logistic regression in the asymptotic regime. We believe our theoretical tool may be of independent interest to other researchers pursuing theoretical studies of privacy for machine learning models. In this work, we use our tool to provide a precise asymptotic characterization of MI for the optimal black-box MI attack.

For the practitioner, our analysis encourages considering the number of parameters when designing privacy-preserving machine learning models. In particular, larger models may be more beneficial as long as they are carefully coupled with proper regularization. In summary, our paper has three core contributions:

1. We demonstrate how individually increasing either the number of parameters or decreasing the regularization of a classification model **decreases its privacy.**
2. We discover that wider NNs enjoy an improved regularization-induced privacy–utility trade-off compared to narrow ones, and that, controlling for the privacy level by regularization, **increased generalization performance due to overparameterization is not at odds with privacy.**
3. We theoretically analyze high-dimensional logistic regression in the asymptotic regime and replicate our empirical NN observations for a bi-level feature ensemble using a **novel leave-one-out analysis that may be of independent interest.** Using this tool, we also prove the fundamental MI vulnerability for overparameterized logistic regression models.

Related Work. This work contributes to the rapidly growing field of membership inference (MI), a framework being increasingly used to study the privacy implications of machine learning models. Previous works have shown how MI is in principle a task of hypothesis testing with the optimal adversary being the likelihood ratio test (LRT) [16, 17, 13]. We leverage this optimal LRT adversary in our theoretical analysis and in our SVM experiments. Since the distributions for the LRT are typically not known for general models such as neural networks, more practical attack strategies such as binary classification [11, 18] and perturbation-based inference [19, 20] have been proposed. We refer the reader to [21] for a comprehensive survey of MI attacks. For our neural network experiments, we use the loss thresholding attack introduced by [22] and improved by [23] due to its simplicity and effectiveness.

Prior work has also studied how various types of regularization affect MI attacks [13, 24–28]. There are limited studies on the effect of overparameterization on MI. [13] analyze how linear regression models are more susceptible to MI as they become more overparameterized, and [9, 12] empirically observe larger language models being more vulnerable to MI than their smaller counterparts. [22] study the theoretical connection between overfitting and membership advantage but do not connect this to the number of parameters.

In addition to MI, differential privacy (DP) is another popular framework used to study the privacy implications of machine learning algorithms [29–31]. Differentially private training algorithms ensure that models obtained when training on datasets differing in one data point do not differ much. [32, 33] show that larger models achieve better utility for the same DP amount when using fine-tuning, echoing our message that larger models have better privacy–utility trade-offs than smaller ones. By providing rigorous worst-case guarantees, DP also protects models from MI attacks [22], but typically at the cost of having very low utility [34–36]. Indeed, it has been shown that DP techniques provide poorer MI defense vs. utility trade-offs than other MI defense schemes [20, 37]. Furthermore, while they provide powerful information-theoretic guarantees, it is not clear how the DP metrics of (ϵ, δ) translate to vulnerability from real-world MI attacks. As such, we believe both MI and DP analyses complement each other in providing a comprehensive understanding of privacy-preserving machine learning, and we focus on MI in this work.

Our work is strongly related to the "double descent" literature that studies the relationship of overparameterization and generalization error [38]. [4] demonstrate double descent behavior in neural networks as a function of the number of parameters and number of training epochs. To theoretically understand the trade-off between generalization error and an adversary's MI accuracy, we study the popular "bi-level ensemble" model that has been shown to exhibit benign overfitting in classification [39, 40]. To characterize the difference of predictions on training points and test points, we leverage the proportional asymptotics regime, where precise analysis is enabled by tools such as the convex Gaussian min-max theorem [14] and approximate message passing [41–43]. In particular, we build upon [15] to analyze the behavior of logistic regression in the asymptotic regime.

2 THEORETICAL FOUNDATIONS OF MEMBERSHIP INFERENCE

We define our MI problem for classification as follows. Let $\mathcal{S} = ((\mathbf{x}_i, y_i))_{i=1}^n$ be a training dataset of features $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^p$ and labels $y_i \in \mathcal{Y} = \{1, \ldots, k\}$ (i.e., multi-class classification). We assume that each data point and its associated label is an independent sample from a distribution $\mathcal{D}$ over the data such that $\mathcal{S} \sim \mathcal{D}^n$. Furthermore let $\mathcal{F}$ denote a class of machine learning models (e.g., linear models or neural networks) such that for $f \in \mathcal{F}$, $f : \mathcal{X} \to \mathbb{R}^k$, producing a vector of confidence values from which the the final prediction is given as $\hat{y}(\mathbf{x}) = \arg\max_j [f(\mathbf{x})]_j$.

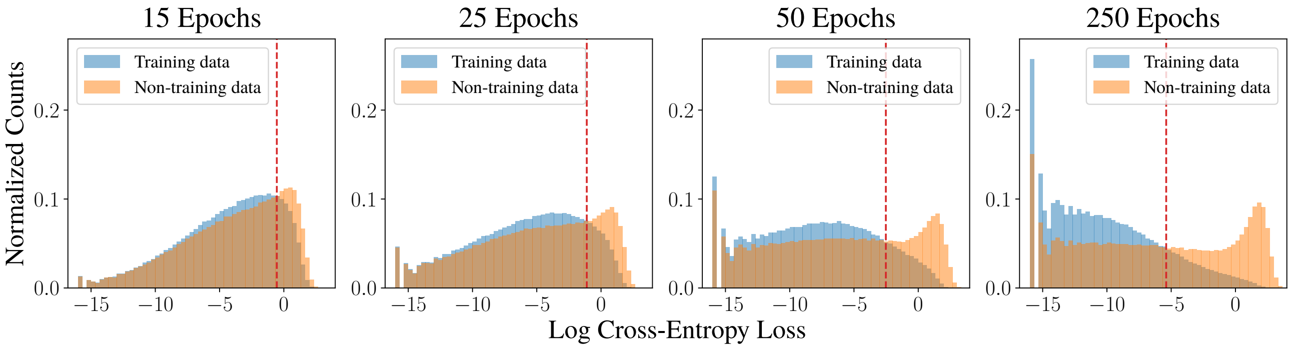


Figure 1: **Loss gap increases with epochs.** Empirical histograms of log cross-entropy losses for training and non-training data points of 20 ResNet18s ($w = 64$) trained on CIFAR10 for different training epochs. While both distributions generally shift towards smaller losses with more epochs, the losses for training points shift at a quicker rate than those for non-training points, enabling loss-threshold attacks. The optimal threshold is depicted with the red dashed line. This illustrates why loss threshold MI accuracy increases with epochs (Figure 4). For visualization purposes, we drop points that achieve 0 (to machine precision) loss.

For each pair $(\mathbf{x}, y)$, we have access to a loss function $\ell : \mathcal{Y} \times \mathbb{R}^k \to \mathbb{R}_{\geq 0}$ that measures the performance of any $f \in \mathcal{F}$ on the data $\mathcal{S}$. The model's *test (misclassification) error* is defined as $\mathcal{E}(f) = \Pr(y \neq \hat{y}(\mathbf{x}))$, where $(\mathbf{x}, y)$ is drawn from $\mathcal{D}$ for our theoretical results or from the test set for our experiments. Finally, let A be a MI adversary. For a fixed model $f \in \mathcal{F}$ trained on $\mathcal{S}$, we assume that $A : \mathcal{F} \times \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$ has access to f and a sample $(\mathbf{x}, y)$ and predicts 1 if it believes $(\mathbf{x}, y) \in \mathcal{S}$ and 0 otherwise. To be rigorous, we define MI as the following experiment [22, 13].

Experiment 1. *Given distribution $\mathcal{D}$, model class $\mathcal{F}$, loss function ℓ, and adversary A, a membership inference experiment is*

1. *Sample $\mathcal{S} \sim \mathcal{D}^n$.*
2. *Learn $\hat{f} \in \arg\min_{f \in \mathcal{F}} \sum_{i=1}^n \ell(y_i, f(\mathbf{x}_i))$.*
3. *Sample $m \in \{0, 1\}$ uniformly at random.*
4. *If $m = 0$, sample a new test data point $(\mathbf{x}, y) \sim \mathcal{D}$. If $m = 1$, sample a training data point $(\mathbf{x}, y) \in \mathcal{S}$ uniformly at random.*
5. *Observe the adversary's prediction $A(\hat{f}, \mathbf{x}, y)$.*

In essence, Experiment 1 reduces the problem of MI to one of hypothesis testing. Accordingly, we quantify the performance of an adversary in terms of its *membership (inference) advantage*, defined as the difference between the true positive rate and the false positive rate.

Definition 1 ([22])**.** *The membership advantage of an adversary A against $\hat{f}$ is*

$$\begin{aligned}\mathrm{Adv}(A) = \Pr(A(\hat{f}, \mathbf{x}, y) = 1 \mid m = 1) \\ - \Pr(A(\hat{f}, \mathbf{x}, y) = 1 \mid m = 0),\end{aligned} \quad (1)$$

where $\Pr(\cdot)$ is taken jointly over all randomness in Experiment 1.

Membership inference can be performed successfully when the model treats points from the training dataset $\mathcal{S}$ "differently" than new test points. More precisely, if the distribution of the model's output on a data point $(\mathbf{x}, y)$ differs significantly when $(\mathbf{x}, y)$ is a training point vs. when it is not, then MI attacks can distinguish between the two distributions to determine if $m = 0$ or $m = 1$. Indeed, we observe in Figure 1 that as a model trains on data, its loss on those data points decreases at a rate faster than its loss on non-training data points. Then, even an attack as simple as thresholding the loss ($A(f, \mathbf{x}, y) = \mathbb{1}\{\ell(y, f(\mathbf{x})) < \tau\}$) [22, 16, 23] can successfully perform MI.

In this work, we consider *black-box adversaries*, which only have access to the data point $(\mathbf{x}, y)$ and the model's output $\hat{f}(\mathbf{x})$ rather than the whole model itself. Our theoretical analysis employs the likelihood ratio test (LRT) attack, which is the optimal black-box MI adversary [16, 17, 13]. Given $(\mathbf{x}, y)$ and $\hat{f}(\mathbf{x})$, the LRT adversary outputs 1 if the likelihood of the model outputting $\hat{f}(\mathbf{x})$ is higher if $(\mathbf{x}, y)$ was a training point than if it was not a training point.

2.1 Analysis Framework and Core Theoretical Result

To theoretically analyze MI, we consider a regularized high-dimensional logistic regression setting. We define the logistic loss $\ell(y, z) = \rho(z) - yz$ in terms of the function $\rho(z) = \log(1+\exp(z))$ whose derivative $\rho'(z) = 1/(1+\exp(-z))$ is the sigmoid function. We let $\mathbf{x}_i \sim \mathcal{N}(\mathbf{0}, \frac{1}{p}\boldsymbol{\Sigma})$ for some positive definite covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{p \times p}$, and for ground truth coefficients $\boldsymbol{\beta}^* \in \mathbb{R}^p$, binary labels $y_i \in \{0, 1\}$ are generated such that $\Pr(y_i = 1|\mathbf{x}_i) = \rho'(\mathbf{x}_i^\top \boldsymbol{\beta}^*)$. Our learned decision function is $\hat{f}(\mathbf{x}) = \mathbf{x}^\top \widehat{\boldsymbol{\beta}}$, yielding predictions $\hat{y}(\mathbf{x}) = \mathbb{1}\{\hat{f}(\mathbf{x}) > 0\}$, where

$$\widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \frac{1}{n} \sum_{i=1}^n \ell(y_i, \mathbf{x}_i^\top \boldsymbol{\beta}) + \frac{\lambda}{2p} \|\boldsymbol{\beta}\|_2^2. \quad (2)$$

We then consider the asymptotic limit as $n, p \to \infty$ with $n/p \to \delta \in (0, \infty)$. This enables us to apply the analysis of [15], who used the convex Gaussian min-max theorem (CGMT) [14] to completely characterize the generalization performance of logistic regression in terms of the solution to a nonlinear system of equations of a few scalar variables; see Appendix B.2 for details.

The challenge in studying MI is that we must additionally characterize the difference between the distributions of model outputs for training points and test points. Existing analyses from frameworks such as the CGMT are insufficient to give us the distributions of the outputs for a *single* training point over the randomness of the remaining training dataset, which we need in order to study the LRT adversary [13]. Therefore, we provide a novel leave-one-out-based characterization of the distribution of the output of a linear model for any specific training point. The following result is an informal statement of a more detailed version in Appendix C.

Theorem 1 (informal version of Theorem 5)**.** *Consider the solution $\widehat{\boldsymbol{\beta}}$ to the optimization problem in* (2)*. There exists $\gamma > 0$ such that in the limit as $n, p \to \infty$ with $n/p \to \delta \in (0, \infty)$, for any training point $\mathbf{x}_i$,*

$$\mathbf{x}_i^\top \widehat{\boldsymbol{\beta}} \xrightarrow{d} \mathrm{Prox}_{\gamma \ell(y_i, \cdot)} \left(\mathbf{x}_i^\top \widehat{\boldsymbol{\beta}}_{-i} \right), \tag{3}$$

where $\widehat{\boldsymbol{\beta}}_{-i}$ is the solution to (2) *with $(\mathbf{x}_i, y_i)$ omitted from the training set, and $\xrightarrow{d}$ denotes convergence in distribution where the randomness is over the other $n-1$ training points.*

That is, the distribution of the model output for a training point is simply the distribution of the proximal operator (defined in Appendix B.1) of the loss function applied to the output of the training point as if it was a new test point. This will drive the loss closer to zero for training points than for new test points, and so as discussed above, an adversary can exploit this difference to perform MI.

We strongly believe this theoretical tool to be of independent interest, opening the door to future theoretical study of privacy in high dimensional linear models, in particular with sharp asymptotics for any given adversary rather than simply worst-case bounds. Our proof strategy is general and applies to general convex losses and regularization penalties, as we describe in Appendix C. A particularly exciting open question for future work is determining what types of losses, regularization, and feature distributions can lead to a small γ such that the resulting model is the most private.

2.2 A Bi-level Feature Ensemble

In order to study the trade-off between accuracy and privacy as a function of overparameterization in machine learning models, we need a setting in which benign overfitting occurs—that is, that as we increase the number of parameters of our model, generalization accuracy increases as well. To that end, we define a bi-level feature ensemble similar to that considered in [39, 40]. In this model, we define $\boldsymbol{\Sigma}$ and $\boldsymbol{\beta}^*$ for some $d < p$ and $\eta > 0$ as

$$[\boldsymbol{\Sigma}]^2_{k,k'} = \begin{cases} \frac{p}{d} & \text{if } 1 \leq k = k' \leq d, \\ \frac{\eta p}{p-d} & \text{if } d < k = k' \leq p, \\ 0 & \text{if } k \neq k', \end{cases} \tag{4}$$
$$\beta^*_k \sim \begin{cases} \mathcal{N}(0, \sigma^2_\beta) & \text{if } 1 \leq k \leq d, \\ 0 & \text{if } d < k \leq p. \end{cases}$$

In this way, there is always a total variance of 1 in the first d features and of η in the tail of $p - d$ features. As $\phi = p/d \to \infty$, this model is known to exhibit benign overfitting [40].

The intuition behind this feature model is that the signal $\boldsymbol{\beta}^*$ is fundamentally low dimensional and is aligned with a small subset of d highly representative features. Meanwhile, there are an abundance of nuisance features of very small magnitude that are uncorrelated with the signal, such that they can absorb label noise [44] without adversely affecting prediction on new examples with uncorrelated nuisance features. In this way, training points can achieve perfect accuracy even under noise while the model still generalizes well. Furthermore, nonlinearities like those used in neural networks are known to add a similar low-magnitude tail of nonzero eigenvalues to the feature covariance in their Gaussian equivalents [2, 45], connecting this feature model with realistic models like neural networks.

2.3 Asymptotic Privacy and Utility

Given the framework of the CGMT [15], we can easily determine the asymptotic generalization error for logistic regression. Thanks to Theorem 1, we can also determine the MI advantage given an adversary A. The following corollary captures these results, specializing the MI advantage to that of the worst-case optimal LRT adversary.

Corollary 2. *Consider the bi-level feature ensemble in* (4) *and the decision function $\hat{f}(\mathbf{x}) = \mathbf{x}^\top \widehat{\boldsymbol{\beta}}$ for $\widehat{\boldsymbol{\beta}}$ solving* (2)*. Then there exist $\alpha, \gamma, \sigma > 0$ such that, in the limit as $p \to \infty$ with $n/p \to \delta \in (0, \infty)$,*

(i) Generalization error. *The misclassification error for a new test pair $(\mathbf{x}, y)$ is given by*

$$\mathcal{E}(\hat{f}) = \mathbb{E}\left[\rho'(Z)\Phi(-\tfrac{\alpha Z}{\sigma})\right], \tag{5}$$

where Φ is the standard normal CDF and $Z \sim \mathcal{N}(0, \sigma^2_\beta)$;

(ii) Membership advantage. *For any training pair $(\mathbf{x}_i, y_i)$, the membership advantage of the optimal adversary is*

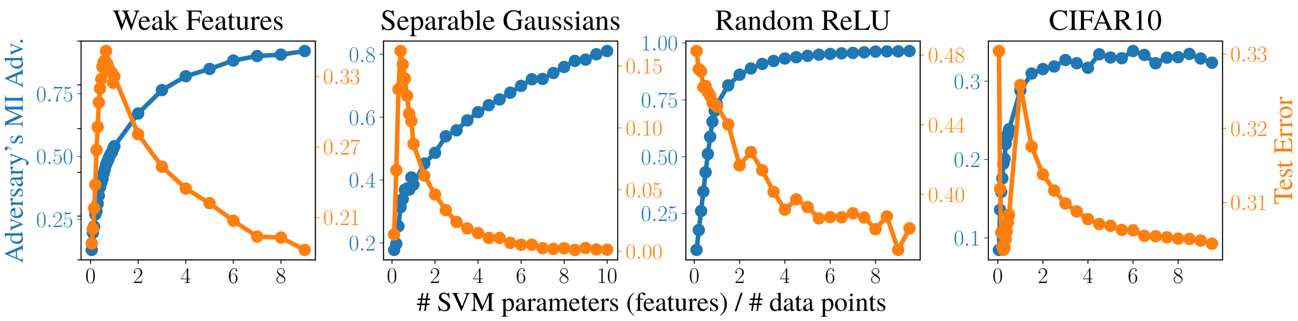


Figure 2: **Privacy vs. parameters (SVMs).** We demonstrate on SVMs for a variety of feature models how increasing overparameterization increases the adversary's MI advantage on the SVM model even as it decreases validation error. Thus, the number of parameters induces a privacy–utility trade-off.

given by

$$\max_A \mathrm{Adv}(A, \hat{f}; \mathbf{x}_i, y_i) = \frac{1}{\sigma} \int_{\mathbb{R}} \max\Big\{\Phi'\Big(\tfrac{z - \alpha \mathbf{x}_i^\top \boldsymbol{\beta}^* + \gamma(\rho'(z) - y_i)}{\sigma}\Big)(1 + \gamma\rho''(z)) - \Phi'\Big(\tfrac{z - \alpha \mathbf{x}_i^\top \boldsymbol{\beta}^*}{\sigma}\Big), 0\Big\} dz, \quad (6)$$

where Φ' is the standard normal PDF.

It is not possible to determine closed-form expressions for (α, γ, σ) in terms of the parameters $(\phi, \eta, \sigma_\beta)$ of the bi-level feature ensemble in general, as the former are the solutions to a system of nonlinear equations (see Theorem 4 in Appendix B.2). This makes direct theoretical analysis of the privacy–utility trade-offs difficult.

However, we can obtain the values (α, γ, σ) by solving the nonlinear system numerically. In the next sections, when we plot theoretical trade-off curves for logistic regression, we solve the nonlinear system and then evaluate the above expressions using numerical integration, reporting the average sample-specific membership advantage $\mathbb{E}_{(\mathbf{x}_i, y_i) \sim \mathcal{D}}[\max_A \mathrm{Adv}(A, \hat{f}; \mathbf{x}_i, y_i)]$. We refer the reader to Appendix E for proof details for Corollary 2 and Appendix F for simulations that verify our computations of the sample-specific training and test densities.

3 INDIVIDUAL PRIVACY–UTILITY TRADE-OFFS

We now demonstrate that there are privacy–utility trade-offs as a function of either the number of model parameters or the amount of regularization, individually. Specifically, when either increasing the number of parameters or decreasing the amount of regularization from an over-regularized state, the resulting machine learning model generally becomes more accurate (improved generalization performance) but becomes less private (higher MI advantage of the adversary). The increase in accuracy with overparameterization has been discussed in detail in the double descent literature, and we refer readers to [46, 4, 38]. The decrease of privacy with overparameterization on MI has been observed for linear regression models in [13], but we show here that the phenomenon is robust, extending to classification models and even highly nonlinear models such as deep neural networks. We show the parameter-wise and regularization-wise tradeoffs experimentally on various machine learning tasks and provide some theoretical insights to their origins. All experimental details not included in the main text can be found in Appendix G. In all neural network plots, shaded areas indicate one standard deviation over repeated trials.

3.1 Parameter-Wise Privacy–Utility Trade-Off

We begin our demonstration with an empirical look at support vector machines (SVMs), ploting both an adversary's membership advantage and the SVM model's validation error as a function of the number of parameters in Figure 2 for a variety of data models. We observe how MI increases (thus damaging privacy) while test error decreases (yielding a more accurate model) as the number of parameters grows. We consider data models based on those that have been shown to exhibit double descent in the overparameterized machine learning literature, including the weak features ensemble from [39], separable Gaussians with irrelevant features (based on synthetic dataset 1 of [47]), random ReLU features [48], and random projections on two classes of CIFAR10. For the MI attack, we estimate the optimal LRT adversary [13] by approximating the model output distributions as discrete histograms using Monte Carlo sampling over a minimum of 20,000 trials.

In Figure 3, we demonstrate that the same phenomenon occurs in NNs. Specifically, we consider NNs that are trained with optimal (with respect to validation error) early stopping: we stop training at the number of training epochs that maximizes validation accuracy. We consider three machine learning tasks: feature vector classification on the Purchase100 dataset [11] using a 2-layer NN, image classification on CIFAR10 [49] using the ResNet18 architecture [50], and language translation on the Multi30K dataset [51] using the

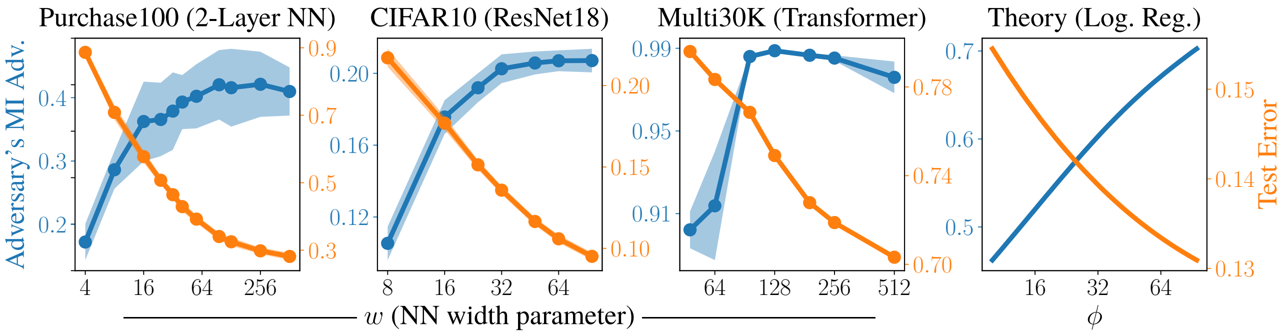


Figure 3: **Privacy vs. parameters (NNs and theory).** For NNs trained to optimal early stopping with respect to validation error, increasing the network's width generally increases MI advantage on the network even as it decreases its test error. We see a similar effect for logistic regression with the bi-level ensemble theoretically when λ is tuned to minimize test error.

Transformer architecture [52]. We control the number of parameters of the networks by scaling the size of the hidden dimensions by a width parameter w. The MI attack we employ is the sample-specific loss threshold attack ("attack R" in [23]): $A(f, \mathbf{x}, y) = \mathbb{1}\left\{\ell(y, f(\mathbf{x})) < \tau(\mathbf{x}, y)\right\}$, where $\tau(\mathbf{x}, y)$ is a sample-specific threshold learned for each data point over reference/shadow models.

Theoretical Insights. Using our theoretical tool from Theorem 1, we can in fact prove that for an extremely broad class of settings, including the bi-level ensemble which exhibits benign overfitting, extreme overparameterization leads to perfect MI by any loss-thresholding adversary. We capture this result in the following theorem. We have omitted some technical conditions related to the convergence of a system of fixed point equations for the statement of part (a); please see Theorem 6 in Appendix D for precise details. In the theorem statements, we assume all scalar variables (such as λ and η) to be fixed unless otherwise specified.

Theorem 3. *If* $\hat{f}(\mathbf{x}) = \mathbf{x}^\top \widehat{\boldsymbol{\beta}}$ *for* $\widehat{\boldsymbol{\beta}}$ *the solution to* (2) *and for some* $\tau > 0$ *we have an adversary* $A(f, \mathbf{x}, y) = \mathbb{1}\left\{\ell(y, f(\mathbf{x})) < \tau\right\}$, *then as* $n, p \to \infty$ *with* $n/p \to \delta \in (0, \infty)$,

- *(a) If* $\lim_{p\to\infty} \left\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{\beta}^*\right\|_2/\sqrt{p}$ *exists and is finite, and* $\liminf_{p\to\infty} \lambda_{\min}(\boldsymbol{\Sigma}) > 0$, *where* $\lambda_{\min}(\boldsymbol{\Sigma})$ *is the smallest eigenvalue of* $\boldsymbol{\Sigma}$, *then as* $\delta \to 0$, $\mathrm{Adv}(A) \to 1$.
- *(b) For the bi-level ensemble in* (4), *if* $p/d \to \phi \in (1, \infty)$ *and* d/n *converges to a fixed value, then as* $\phi \to \infty$, $\mathrm{Adv}(A) \to 1$, *and in the limit as* $\lambda \to \infty$, $\mathcal{E}(\hat{f})$ *is decreasing in* ϕ.

This theorem highlights that as $\delta \to 0$, which means that the model becomes increasingly overparameterized, *any* constant-threshold adversary's MI advantage converges to 1, and the adversary can perform perfect MI attacks on the learned model. We emphasize that the constant-threshold adversary is much weaker than the sample-specific loss threshold adversary we consider in our experiments, and it need not be adapted to the problem in any way, yet overparameterized models are still vulnerable. This is true regardless of any (fixed) value of regularization strength, meaning that ridge regularization is not sufficient to protect against MI attacks, echoing what was observed by [13] in linear regression. This result applies not only to standard isotropic data covariances, but also to highly anisotropic covariances such as the bi-level ensemble.

Part (b) highlights how in the right circumstances, we can still see generalization performance improving with overparameterization—that there is a trade-off between generalization and privacy, just as in our experimental results. We illustrate this alongside neural networks in Figure 3 for the bi-level model with fixed $n/d = 5$, $\sigma_\beta = 10$, and $\eta = 1$, with λ tuned to minimize test error, analogously to the optimal validation error early stopping in the NN experiments. This plot is generated using the expressions in Corollary 2 for test error and the optimal adversary's MI advantage. We see that the generalization error decreases but the adversary's MI advantage increases as the length of the tail of small eigenvalues of $\boldsymbol{\Sigma}$ increases for larger values of ϕ.

3.2 Regularization-Wise Privacy–Utility Trade-Off

Using the same classification tasks and NN architectures as in Section 3.1, we empirically demonstrate an epoch-wise privacy–utility trade-off in Figure 4, where we plot the adversary's MI advantage and the model's generalization error as a function of training epochs. Stopping the training at earlier epochs corresponds to higher regularization, as the model has less opportunity to overfit to training data. We include a variety of NN widths in our plot, demonstrating similar trade-offs across widths.

We also plot the theoretical test error and MI advantage from Corollary 2 for logistic regression with the bi-level feature ensemble as a function of the regularization strength. Specifically, we plot the regularization as a function of $1/\lambda\delta$, where λ is the ℓ_2 regularization parameter, such that smaller

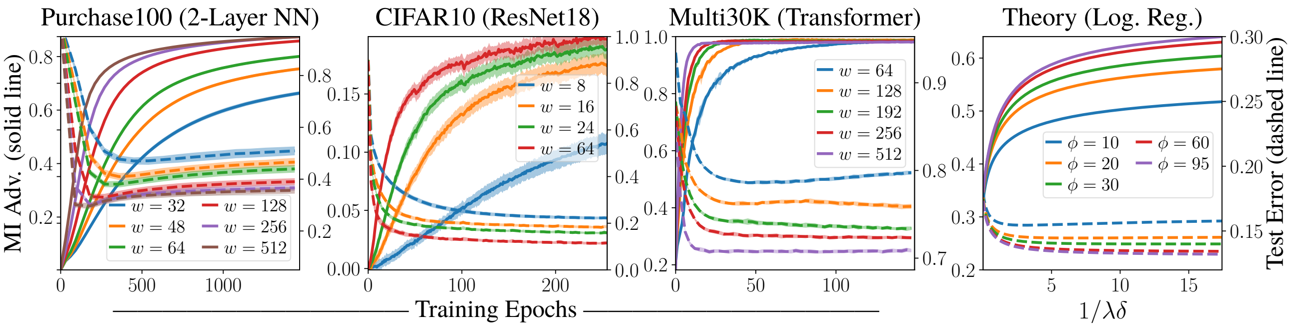


Figure 4: **Privacy vs. regularization.** Regardless of neural network width (parameterized by w), increasing the number of training epochs (decreasing regularization) increases the adversary's MI advantage (solid line) while simultaneously decreasing its test error (dashed line). This induces a regularization-wise privacy–utility trade-off. The same holds theoretically for logistic regression when decreasing ridge regularization under the bi-level feature ensemble setting.

values of $1/\lambda\delta$ correspond to more regularization. Just as we explore a variety of widths for NNs, we consider a variety of values of $\phi = p/d$, measuring the amount of overparameterization for the bi-level feature ensemble.

An interesting observation from Figure 4 is that the adversary's MI advantage can continue to increase even if test error stays the same. Thus, generalization error does not completely characterize MI. Instead, what leads to the increase in MI advantage is the increasing generalization (cross-entropy) loss gap. As the network is trained for more epochs, or the logistic regression model is less regularized, its training loss decreases at a greater rate than its test loss. Thus, it becomes easier to divide the training and test losses with a loss threshold, as illustrated in Figure 1. The losses continue separating even after test error has converged, causing the MI advantage to continue to increase.

4 A BLESSING OF DIMENSIONALITY: ELIMINATING THE PRIVACY–UTILITY TRADE-OFF

We now show that, perhaps counter-intuitively, if we jointly tune both the numbers of parameters and the amount of regularization, we can eliminate the privacy–utility trade-off. The main idea is to *increase* the number of parameters while also *increasing* the regularization appropriately.

Our key observation is that the decrease in the model's generalization error and the increase in an adversary's MI advantage occur at different rates during training for NNs of different widths (recall Figure 4). However, it is difficult to compare these rates across different NN widths when privacy and utility are individually plotted against regularization. Hence, we plot parametric curves for varying widths as a function of regularization (epochs for NNs, and ridge penalty for logistic regression) in a *privacy–utility plane* in Figure 5, which enables us to abstract away the regularization strength and compare trade-off curves across widths directly. In the plot, ideal performance is the lower-left corner, as this represents low MI advantage (high privacy) and low test error. In this representation, the story becomes clear: wider networks can induce better privacy–utility trade-offs. That is, they are both below and to the left of the trade-off curves for narrower networks. The same occurs for theoretical logistic regression with the bi-level ensemble. Thus, increased parameterization is not inherently a privacy liability and can instead actually improve the privacy of a model.

We explicitly show how early stopping (with the appropriate stopping rule) *eliminates* the privacy–utility trade-off for overparameterization in Figure 6. If we tune the number of training epochs for each width such that a fixed MI advantage is reached (which takes fewer epochs for larger widths), then we see from Figure 6a that overparameterization only *decreases* the generalization error. Similarly, tuning the number of epochs to a fixed validation error results in a decrease of the adversary's MI advantage with increasing width, as shown in Figure 6b. In essence, either privacy or improved generalization can be obtained without taking a hit in the other by opting for a larger network with proper regularization. While we do not recommend early stopping alone as a sufficient privacy-preserving mechanism (practitioners should likely also consider the wide collection of existing MI defense schemes), this strongly suggests that practitioners should include wider networks in their model search and then tune their regularization appropriately to achieve a desired level of privacy.

In Appendix H, we include additional experiments that we could not include in the main paper for space reasons, including a version of Figure 6 for Transformers on Multi30K, repeating all of the experiments in Figures 3–6 for global loss thresholding attacks, and the MI vs. test error trade-off for networks trained with DP-SGD [31] on CIFAR10. In all cases, we see the same behavior—when the regularization is tuned for MI, larger models achieve better protection from MI and better classification accuracy than smaller models.

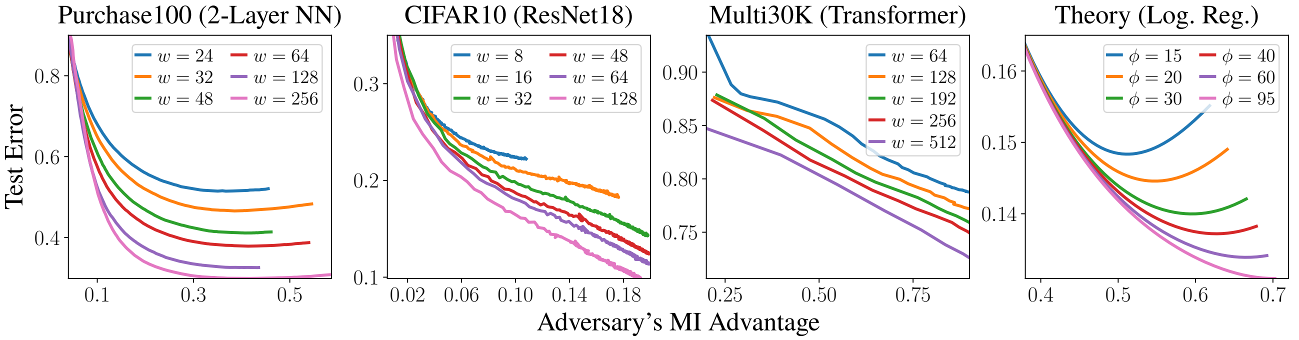
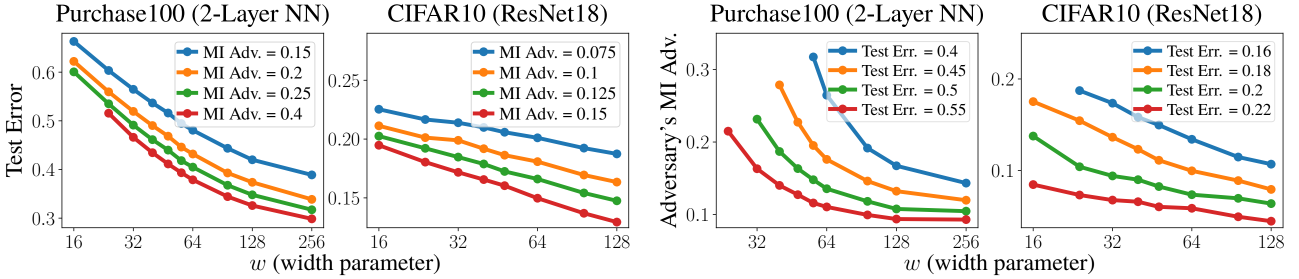


Figure 5: **Trade-offs are better with increased width.** We plot the regularization-wise privacy–utility trade-off of networks of different widths by sweeping through different numbers of training epochs. Observe that, for sufficiently low validation errors, wider networks are closer to the lower-left (high accuracy, high privacy) region compared to narrower networks. The same holds theoretically for logistic regression in the bi-level feature ensemble sweeping through the ridge penalty.



(a) Test error vs. network width for fixed MI adv.

(b) MI adv. vs. network width for fixed test error.

Figure 6: **Overparameterization with early stopping eliminates the privacy–utility trade-off.** (a) For each network width, we train the network until it reaches a given MI advantage value. We then plot the test error of the networks. Observe how test error decreases with parameters at a fixed MI advantage value, showing how proper tuning of parameters and epochs together improves model accuracy without damaging its privacy. Thus, this eliminates the privacy–utility trade-off. (b) Same as (a) but switching the roles of MI advantage and test error.

5 DISCUSSION

We began this exploration with a question: is overparameterization a privacy liability? In our theoretical and empirical investigation, we have demonstrated that overparameterization *can* be a privacy risk, but that it *need not be*, and that, in fact, it often provides even greater privacy when coupled with appropriate regularization. To the best of our knowledge we have provided the first study of this effect in the context of membership inference.

While we showed how ridge regularization for logistic regression and early stopping for neural networks bring out this blessing of dimensionality, many other types of regularization are used in practice. For one example, we include a preliminary experiment using DP-SGD [31] in Figure 13 in Apppendix H, for which we also observe wider networks having better trade-offs. However, not every regularizer may induce the same effect, and an interesting open research direction is to discover which types of regularization or other learning techniques can draw out even more privacy benefits from large models. For example, in the field of differential privacy, by fine-tuning pre-trained language models, [32, 33] achieve better accuracy with larger models than smaller ones for the same privacy budget. A nascent regularization approach strongly worth further study is network pruning, which has been observed to be an effective defense against membership inference attacks [26] as well as a vulnerability in some settings [53].

The phenomenon of better privacy–utility trade-offs for overparameterized models also has important takeaways for our general understanding of the benefits of overparameterization. As we have shown, highly overparameterized models not only have more capacity to memorize than smaller networks (which leads to increased risk of MI), but they also appear to learn the underlying structure of the data *even more quickly* than they memorize data. Identifying the mechanism that provides this benefit in overparameterized models and developing appropriate measures for an "effective" number of parameters that reflects the memorization capacity of the model as a function of both the true number of parameters and forms of regularization are important open questions. We believe our leave-one-out characterization of the training output distribution in Theorem 1 will be helpful in answering these questions with respect to privacy.

Acknowledgements

This work was supported by NSF grants CCF-1911094, IIS-1838177, and IIS-1730574; ONR grants N00014-18-12571, N00014-20-1-2534, and MURI N00014-20-1-2787; AFOSR grant FA9550-22-1-0060; and a Vannevar Bush Faculty Fellowship, ONR grant N00014-18-1-2047.

References

[1] Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J. Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *The Annals of Statistics*, 50(2):949–986, 2022.

[2] Song Mei and Andrea Montanari. The generalization error of random features regression: Precise asymptotics and the double descent curve. *Communications on Pure and Applied Mathematics*, 75(4):667–766, 2022.

[3] Mikhail Belkin, Daniel Hsu, and Ji Xu. Two models of double descent for weak features. *SIAM Journal on Mathematics of Data Science*, 2(4):1167–1180, 2020.

[4] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003, 2021.

[5] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv*, abs/2001.08361, 2020.

[6] Mia Xu Chen, Benjamin N Lee, Gagan Bansal, Yuan Cao, Shuyuan Zhang, Justin Lu, Jackie Tsay, Yinan Wang, Andrew M Dai, Zhifeng Chen, et al. Gmail smart compose: Real-time assisted writing. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2287–2295, 2019.

[7] Google. Learn how google improves speech models. URL `https://support.google.com/assistant/answer/11140942?hl=en`. Retrieved 13 Oct 2022.

[8] Zeynep Batmaz, Ali Yurekli, Alper Bilge, and Cihan Kaleli. A review on deep learning for recommender systems: challenges and remedies. *Artificial Intelligence Review*, 52(1):1–37, 2019.

[9] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650, 2021.

[10] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333, 2015.

[11] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, 2017.

[12] Fatemehsadat Mireshghallah, Kartik Goyal, Archit Uniyal, Taylor Berg-Kirkpatrick, and Reza Shokri. Quantifying privacy risks of masked language models using membership inference attacks. *arXiv*, abs/2203.03929, 2022.

[13] Jasper Tan, Blake Mason, Hamid Javadi, and Richard G. Baraniuk. Parameters or privacy: A provable tradeoff between overparameterization and membership inference. *arXiv preprint arXiv:2202.01243*, 2022.

[14] Christos Thrampoulidis, Ehsan Abbasi, and Babak Hassibi. Precise error analysis of regularized M-estimators in high dimensions. *IEEE Transactions on Information Theory*, 64(8):5592–5628, 2018. doi: 10.1109/TIT.2018.2840720.

[15] Fariborz Salehi, Ehsan Abbasi, and Babak Hassibi. The impact of regularization on high-dimensional logistic regression. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[16] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Herve Jegou. White-box vs black-box: Bayes optimal strategies for membership inference. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 5558–5567, 2019.

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

[18] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. In *26th Annual Network and Distributed System Security Symposium*. The Internet Society, 2019.

[19] Christopher A. Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 1964–1974, 2021.

[20] Yigitcan Kaya, Sanghyun Hong, and Tudor Dumitras. On the effectiveness of regularization against membership inference attacks. *arXiv*, abs/2006.05336, 2020.

[21] Hongsheng Hu, Zoran Salcic, Gillian Dobbie, and Xuyun Zhang. Membership inference attacks on machine learning: A survey. *arXiv*, abs/2103.07853, 2021.

[22] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 268–282, 2018.

[23] Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, and Reza Shokri. Enhanced membership inference attacks against machine learning models. *arXiv*, abs/2111.09679, 2021.

[24] Yigitcan Kaya and Tudor Dumitras. When does data augmentation help with membership inference attacks? In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 5345–5355, 2021.

[25] Erick Galinkin. The influence of dropout on membership inference in differentially private models. *arXiv*, abs/2103.09008, 2021.

[26] Yijue Wang, Chenghong Wang, Zigeng Wang, Shanglin Zhou, Hang Liu, Jinbo Bi, Caiwen Ding, and Sanguthevar Rajasekaran. Against membership inference attack: Pruning is all you need. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pages 3141–3147, 2021.

[27] Liwei Song, Reza Shokri, and Prateek Mittal. Membership inference attacks against adversarially robust deep learning models. In *2019 IEEE Security and Privacy Workshops (SPW)*, pages 50–56, 2019.

[28] Shahbaz Rezaei, Zubair Shafiq, and Xin Liu. Accuracy-privacy trade-off in deep ensemble. *arXiv*, abs/2105.05381, 2021.

[29] Cynthia Dwork. Differential privacy: A survey of results. In *International Conference on Theory and Applications of Models of Computation*, pages 1–19, 2008.

[30] Trung Ha, Tran Khanh Dang, Tran Tri Dang, Tuan Anh Truong, and Manh Tuan Nguyen. Differential privacy in deep learning: an overview. In *2019 International Conference on Advanced Computing and Applications (ACOMP)*, pages 97–102, 2019.

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

[32] Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, Sergey Yekhanin, and Huishuai Zhang. Differentially private fine-tuning of language models. In *The 10th International Conference on Learning Representations*, 2022.

[33] Xuechen Li, Florian Tramèr, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. In *The 10th International Conference on Learning Representations*, 2022.

[34] Bargav Jayaraman and David Evans. Evaluating differentially private machine learning in practice. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 1895–1912, 2019.

[35] Md Atiqur Rahman, Tanzila Rahman, Robert Laganière, Noman Mohammed, and Yang Wang. Membership inference attack against differentially private deep learning model. *Transactions on Data Privacy*, 11(1):61–79, 2018.

[36] T. Tony Cai, Yichen Wang, and Linjun Zhang. The cost of privacy: Optimal rates of convergence for parameter estimation with differential privacy. *The Annals of Statistics*, 49(5):2825 – 2850, 2021.

[37] Jiaxiang Liu, Simon Oya, and Florian Kerschbaum. Generalization techniques empirically outperform differential privacy against membership inference. *arXiv*, abs/2110.05524, 2021.

[38] Yehuda Dar, Vidya Muthukumar, and Richard G. Baraniuk. A farewell to the bias-variance tradeoff? an overview of the theory of overparameterized machine learning. *arXiv*, abs/2109.02355, 2021.

[39] Vidya Muthukumar, Adhyyan Narang, Vignesh Subramanian, Mikhail Belkin, Daniel Hsu, and Anant Sahai. Classification vs regression in overparameterized regimes: Does the loss function matter? *Journal of Machine Learning Research*, 22(222):1–69, 2021.

[40] Ke Wang and Christos Thrampoulidis. Benign overfitting in binary classification of gaussian mixtures. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4030–4034, 2021.

[41] Melikasadat Emami, Mojtaba Sahraee-Ardakan, Parthe Pandit, Sundeep Rangan, and Alyson Fletcher. Generalization error of generalized linear models in high dimensions. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 2892–2901, 2020.

[42] Cedric Gerbelot, Alia Abbara, and Florent Krzakala. Asymptotic errors for teacher-student convex generalized linear models (or: How to prove Kabashima's replica formula). *arXiv*, abs/2006.06581, 2020.

[43] Alyson K Fletcher, Sundeep Rangan, and Philip Schniter. Inference in deep networks in high dimensions. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pages 1884–1888, 2018.

[44] Peter L Bartlett, Philip M Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020.

[45] Jeffrey Pennington and Pratik Worah. Nonlinear random matrix theory for deep learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[46] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32): 15849–15854, 2019.

[47] Mikhail Belkin, Siyuan Ma, and Soumik Mandal. To understand deep learning we need to understand kernel learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 541–549, 2018.

[48] Andrea Montanari, Feng Ruan, Youngtak Sohn, and Jun Yan. The generalization error of max-margin linear classifiers: High-dimensional asymptotics in the overparametrized regime. *arXiv*, abs/1911.01544, 2019.

[49] Alex Krizhevsky. Learning multiple layers of features from tiny images. Master's thesis, University of Tront, 2009.

[50] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[51] Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. Multi30K: Multilingual English-German image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pages 70–74, August 2016.

[52] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[53] Xiaoyong Yuan and Lan Zhang. Membership inference attacks and defenses in neural network pruning. *arXiv*, abs/2202.03335, 2022.

[54] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

[55] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

[56] Ziwei Ji and Matus Telgarsky. The implicit bias of gradient descent on nonseparable data. In *Proceedings of the 32nd Conference on Learning Theory*, volume 99, pages 1772–1798, 2019.

[57] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, volume 20, 2007.

[58] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv*, abs/1412.6980, 2014.

A Blessing of Dimensionality in Membership Inference through Regularization Supplementary Materials

A Limitations and Considerations

A.1 Limitations of this work

A possible limitation of this work is that we focus on a particular class of inference attacks, the loss threshold attack, in most of our experimental results, though we note that as [16] and [13] both argue, the loss threshold attack is essentially optimal for membership inference for many data distributions. More subtly, the procedure we propose for estimating membership inference vulnerability involves computing an empirical estimate. As such, there is uncertainty in this process. In practical settings where the training and validation sets are large, this is likely not a major concern. That said, in settings where the privacy budget is very low and/or privacy is paramount it may additionally be necessary to use high-probability bounds on the adversary's MI advantage rather than the estimate directly for an added layer of security. Furthermore, the theoretical guarantees are in the asymptotic regime. While they show a strong correlation with finite dimension experiments (e.g., Figure 4), developing tight, non-asymptotic results is an open question. [40], for instance, is able to derive non-asymptotic guarantees to connect generalization error to overparameterization, but the same technique does not apply in the case of membership inference: it is important to consider the distribution of the model's output for specific inputs—not just the population on average.

A.2 Ethical Considerations

Ensuring that models protect the data that they are trained on is important for modern machine learning systems. In order to achieve benign overparameterization for membership inference and generalization error jointly, we perform precise tuning and early stopping. When implementing these ideas in practical scenarios, it is recommended that a sensitivity analysis additionally be conducted to ensure that the chosen parameters are sufficiently tight. Without doing so, applying this method may lead to false confidence in a method's robustness to MI attacks. In general, the authors believe that in settings where privacy is of the utmost concern, such as when training with medical data, additional measures beyond those covered in this work should be taken to ensure that the data stays private. Finally, this paper focuses on membership inference in particular and these results are not as general as complete differential privacy. Practitioners should consider additional privacy vulnerabilities beyond membership inference alone.

B Background material

Here we include a few definitions and results borrowed from other works.

B.1 Definitions

We define the proximal operator for a function Ω as follows.

Definition 2 (Proximal operator)**.** *The* proximal operator *of a function* $\Omega\colon \mathbb{R}^p \to \mathbb{R}$ *is defined as*

$$\operatorname{Prox}_{\Omega}(\mathbf{v}) = \operatorname*{arg\,min}_{\mathbf{w}\in\mathbb{R}^p} \Omega(\mathbf{w}) + \frac{1}{2}\|\mathbf{w}-\mathbf{v}\|_2. \tag{7}$$

It will be valuable to consider the first-order optimality condition of the proximal operator; for differentiable penalties, the minimizer $\mathbf{w}^*$ satisfies

$$\nabla\Omega(\mathbf{w}^*) + \mathbf{w} - \mathbf{v} = \mathbf{0}. \tag{8}$$

For our work, we will need the form of the scalar proximal operator for $\Omega(\mathbf{v}) = \frac{1}{2}\|\mathbf{A}\mathbf{v}\|_2^2$ for symmetric $\mathbf{A} \in \mathbb{R}^{p \times p}$, which for $t > 0$ is given by

$$\mathrm{Prox}_{t\Omega}\left(v\right) = \left(\mathbf{I}_p + t\mathbf{A}^2\right)^{-1}\mathbf{v}. \tag{9}$$

We also have the definition of local Lischitzness from Salehi et al. [15].

Definition 3 (Locally Lipschitz). *A function $\Phi\colon \mathbb{R}^d \to \mathbb{R}$ is said to be* locally Lipschitz *if $\forall M > 0$, $\exists L_M \geq 0$, such that $\forall \mathbf{x}, \mathbf{y} \in [-M, +M]^d$, $|\Phi(\mathbf{x}) - \Phi(\mathbf{y})| \leq L_M \|\mathbf{x} - \mathbf{y}\|$.*

B.2 Fixed point equations for logistic regression

We borrow the following theorem (slightly adapted to our notation) from Salehi et al. [15].

Theorem 4 (Theorem 1 of Salehi et al. [15]). *For training data $\mathbf{x}_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \frac{1}{p}\mathbf{I}_p)$ and $y_i \sim \mathrm{Bernoulli}(\mathbf{x}_j^\top \boldsymbol{\beta}^*)$, consider the optimization program*

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min}\, \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, \mathbf{x}_i^\top \boldsymbol{\beta}) + \frac{\lambda}{p}\Omega(\boldsymbol{\beta}), \tag{10}$$

where $\ell(y, z) = \rho(z) - yz$ for $\rho(z) = \log(1 + \exp(-z))$ is the logistic loss, and $\Omega\colon \mathbb{R}^p \to \mathbb{R}$ is a convex regularization function. Consider also the following nonlinear system

$$\begin{cases} \kappa^2\alpha = \dfrac{1}{p}\boldsymbol{\beta}^{*\top}\mathrm{Prox}_{\lambda\sigma\tau\Omega}\left(\sigma\tau(\theta\boldsymbol{\beta}^* + \dfrac{r}{\sqrt{\delta}}\mathbf{g})\right), \\ \gamma = \dfrac{1}{r\sqrt{\delta}p}\mathbf{g}^\top\mathrm{Prox}_{\lambda\sigma\tau\Omega}\left(\sigma\tau(\theta\boldsymbol{\beta}^* + \dfrac{r}{\sqrt{\delta}}\mathbf{g})\right), \\ \kappa^2\alpha^2 + \sigma^2 = \dfrac{1}{p}\left\|\mathrm{Prox}_{\lambda\sigma\tau\Omega}\left(\sigma\tau(\theta\boldsymbol{\beta}^* + \dfrac{r}{\sqrt{\delta}}\mathbf{g})\right)\right\|_2^2, \\ \gamma^2 = \dfrac{2}{r^2}\,\mathbb{E}\left[\rho'(-\kappa Z_1)\left(\kappa\alpha Z_1 + \sigma Z_2 - \mathrm{Prox}_{\gamma\rho}\left(\kappa\alpha Z_1 + \sigma Z_2\right)\right)^2\right], \\ \theta\gamma = -2\,\mathbb{E}\left[\rho''(-\kappa Z_1)\mathrm{Prox}_{\gamma\rho}\left(\kappa\alpha Z_1 + \sigma Z_2\right)\right], \\ 1 - \dfrac{\gamma}{\sigma\tau} = \mathbb{E}\left[\dfrac{2\rho'(-\kappa Z_1)}{1 + \gamma\rho''\left(\mathrm{Prox}_{\gamma\rho}\left(\kappa\alpha Z_1 + \sigma Z_2\right)\right)}\right], \end{cases} \tag{11}$$

where $\mathbf{g} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_p)$ is independent of $\boldsymbol{\beta}^*$ and Ω, and Z_1 and Z_2 are independent standard normal variables. Assume that as $p \to \infty$, $n/p \to \delta$, $\|\boldsymbol{\beta}\|_2/\sqrt{p} \to \kappa$, and that the system in (11) *has a unique solution $(\bar{\alpha}, \bar{\sigma}, \bar{\gamma}, \bar{\theta}, \bar{\tau}, \bar{r})$. Then, as $p \to \infty$, for any locally-Lipschitz function $\Psi\colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, we have*

$$\frac{1}{p}\sum_{j=1}^{p}\Psi(\hat{\beta}_j, \beta_j^*) \xrightarrow{p} \frac{1}{p}\sum_{j=1}^{p}\Psi([\boldsymbol{\Gamma}(\boldsymbol{\beta}^*, \mathbf{g})]_j, \beta_j^*), \tag{12}$$

where $\boldsymbol{\Gamma}(\mathbf{v}, \mathbf{z}) = \mathrm{Prox}_{\lambda\bar{\sigma}\bar{\tau}\Omega}\left(\bar{\sigma}\bar{\tau}(\bar{\theta}\mathbf{v} + \frac{\bar{r}}{\sqrt{\delta}}\mathbf{z})\right)$.

The astute reader may note that Salehi et al. [15] require separable regularizers and drawing $\boldsymbol{\beta}^*$ element-wise i.i.d. from some distribution, but that neither of these are required for their proof technique to go through, so we have stated the more general result here, as we will need both of these assumptions to be relaxed.

C Leave-one-out analysis for membership inference

In order to study MI attacks, we need to understand how the distribution of training points differs from test points. We prove the following result to this end for logistic regression with a ridge penalty; however, the proof strategy is general and applies readily to other losses and penalties for general linear models that admit a result similar to Theorem 4, which includes many common models in machine learning [14, 41, 42].

Theorem 5. *Consider the solution $\widehat{\boldsymbol{\beta}}$ to the optimization problem in* (2)*. Let $\widetilde{\boldsymbol{\beta}}^* = \boldsymbol{\Sigma}^{1/2}\boldsymbol{\beta}$, $\widetilde{\mathbf{x}}_i = \boldsymbol{\Sigma}^{-1/2}\mathbf{x}_i$, and $\widetilde{\Omega}(\widetilde{\boldsymbol{\beta}}) = \frac{1}{2}\|\boldsymbol{\Sigma}^{-1/2}\widetilde{\boldsymbol{\beta}}\|_2^2$. Assume Theorem 4 holds for $\widetilde{\boldsymbol{\beta}}^*$ in place of $\boldsymbol{\beta}^*$ and $\widetilde{\Omega}$ in place of Ω. Then for any training point $\mathbf{x}_i$,*

$$\mathbf{x}_i^\top \widehat{\boldsymbol{\beta}} \xrightarrow{d} \mathrm{Prox}_{\bar{\gamma}\ell(y_i,\cdot)}\left(\mathbf{x}_i^\top \widehat{\boldsymbol{\beta}}_{-i}\right), \tag{13}$$

where $\bar{\gamma}$ is from the result of Theorem 4, and

$$\widehat{\boldsymbol{\beta}}_{-i} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\ \frac{1}{n}\sum_{i'\neq i}^{n} \ell(y_{i'}, \mathbf{x}_{i'}^\top\boldsymbol{\beta}) + \frac{\lambda}{2p}\|\boldsymbol{\beta}\|_2^2. \tag{14}$$

Proof. We first make a leave-one-out modification the optimization problem for a general loss and regularizer:

$$\widehat{\boldsymbol{\beta}} = \boldsymbol{\Sigma}^{-1/2}\cdot \underset{\widetilde{\boldsymbol{\beta}}}{\arg\min}\ \frac{1}{n}\sum_{i'\neq i}^{n} \ell(y_{i'}, \widetilde{\mathbf{x}}_{i'}^\top\widetilde{\boldsymbol{\beta}}) + \frac{\lambda}{p}\overline{\Omega}(\widetilde{\boldsymbol{\beta}}), \tag{15}$$

where

$$\overline{\Omega}_i(\widetilde{\boldsymbol{\beta}}) = \widetilde{\Omega}(\widetilde{\boldsymbol{\beta}}) + \frac{1}{\lambda\delta}\ell(y_i, \widetilde{\mathbf{x}}_i^\top\widetilde{\boldsymbol{\beta}}). \tag{16}$$

Applying Theorem 4 to this problem, the solution is equivalent to one of the form

$$\widehat{\boldsymbol{\beta}}_{\mathrm{equiv}} = \boldsymbol{\Sigma}^{-1/2}\cdot \mathrm{Prox}_{t\overline{\Omega}_i}\left(a\widetilde{\boldsymbol{\beta}}^* + b\mathbf{g}\right), \tag{17}$$

where $t = \lambda\bar{\sigma}\bar{\tau}$, $a = \bar{\sigma}\bar{\tau}\bar{\theta}$, and $b = \bar{\sigma}\bar{\tau}\bar{r}/\sqrt{\delta}$. This proximal operator is the solution $\mathbf{w}^*$ to the equation

$$t\nabla\widetilde{\Omega}(\mathbf{w}^*) + \frac{t}{\lambda\delta}\ell'(y_i, \widetilde{\mathbf{x}}_i^\top\mathbf{w}^*)\widetilde{\mathbf{x}}_i + \mathbf{w}^* - (a\widetilde{\boldsymbol{\beta}}^* + b\mathbf{g}) = 0, \tag{18}$$

where $\ell'(y_i, z) = \partial\ell(y_i, z)/\partial z$. Note that this is equivalent to

$$\mathbf{w}^* = \mathrm{Prox}_{t\widetilde{\Omega}}\left(a\widetilde{\boldsymbol{\beta}}^* + b\mathbf{g} - \frac{t}{\lambda\delta}\ell'(y_i, \widetilde{\mathbf{x}}_i^\top\mathbf{w}^*)\widetilde{\mathbf{x}}_i\right). \tag{19}$$

Here we specialize to the ridge penalty, but this can be extended to separable regularizers with careful application of Stein's lemma. Plugging in the form of the proximal operator for generalized ridge penalties, we have

$$\mathbf{w}^* = \boldsymbol{\Sigma}\left(\boldsymbol{\Sigma} + t\mathbf{I}_p\right)^{-1}\left(a\widetilde{\boldsymbol{\beta}}^* + b\mathbf{g} - \frac{t}{\lambda\delta}\ell'(y_i, \widetilde{\mathbf{x}}_i^\top\mathbf{w}^*)\widetilde{\mathbf{x}}_i\right). \tag{20}$$

We wish to characterize $\mathbf{x}_i^\top\widehat{\boldsymbol{\beta}}$, which is equivalent to characterizing $\mathbf{x}_i^\top\widehat{\boldsymbol{\beta}}_{\mathrm{equiv}} = \widetilde{\mathbf{x}}_i^\top\mathbf{w}^*$. Firstly, we note that for any random vector $\mathbf{u}$ such that $\|\mathbf{u}\|_2^2/\sqrt{p} \to C_{\mathbf{u}} < \infty$ that is independent of $\widetilde{\mathbf{x}}_i$,

$$\frac{1}{p}\mathbf{u}^\top\mathbf{w}^* \xrightarrow{p} \frac{1}{p}\mathbf{u}^\top\boldsymbol{\Sigma}\left(\boldsymbol{\Sigma} + t\mathbf{I}_p\right)^{-1}\left(a\widetilde{\boldsymbol{\beta}}^* + b\mathbf{g}\right). \tag{21}$$

Appealing to Theorem 4 again, this means that the nonlinear system is in fact unaffected by our leave-one-out modification asymptotically, and that both cases have the same solution $(\bar{\alpha}, \bar{\sigma}, \bar{\gamma}, \bar{\theta}, \bar{\tau}, \bar{r})$ to the nonlinear system (11). Therefore,

$$\mathbf{x}_i^\top\widehat{\boldsymbol{\beta}}_{-i} \xrightarrow{d} \widetilde{\mathbf{x}}_i^\top\boldsymbol{\Sigma}\left(\boldsymbol{\Sigma} + t\mathbf{I}_p\right)^{-1}\left(a\widetilde{\boldsymbol{\beta}}^* + b\mathbf{g}\right) \sim \mathcal{N}(0, \kappa^2\bar{\alpha}^2 + \bar{\sigma}^2). \tag{22}$$

Since $\mathbf{g}/\sqrt{p}$ and $\widetilde{\mathbf{x}}_i$ have the same distribution, from the second equation in the nonlinear system (11) we know that

$$\widetilde{\mathbf{x}}_i^\top\boldsymbol{\Sigma}\left(\boldsymbol{\Sigma} + t\mathbf{I}_p\right)^{-1}\widetilde{\mathbf{x}}_i \xrightarrow{\text{a.s.}} \frac{1}{p}\mathbf{g}^\top\boldsymbol{\Sigma}\left(\boldsymbol{\Sigma} + t\mathbf{I}_p\right)^{-1}\mathbf{g} = \frac{\bar{\gamma}\delta}{\bar{\sigma}\bar{\tau}}. \tag{23}$$

All together, this gives us

$$\mathbf{x}_i^\top\widehat{\boldsymbol{\beta}} \xrightarrow{p} \widetilde{\mathbf{x}}_i^\top\boldsymbol{\Sigma}\left(\boldsymbol{\Sigma} + t\mathbf{I}_p\right)^{-1}\left(a\widetilde{\boldsymbol{\beta}}^* + b\mathbf{g}\right) - \bar{\gamma}\ell'(y_i, \mathbf{x}_i^\top\widehat{\boldsymbol{\beta}}) \tag{24}$$

$$\implies \mathbf{x}_i^\top\widehat{\boldsymbol{\beta}} \xrightarrow{d} \mathrm{Prox}_{\bar{\gamma}\ell(y_i, \mathbf{x}_i^\top\widehat{\boldsymbol{\beta}})}\left(\mathbf{x}_i^\top\widehat{\boldsymbol{\beta}}_{-i}\right), \tag{25}$$

which is the stated result. □

D Formal version of Theorem 3 and proof

Theorem 3 is a slightly informal version of the following theorem. The only difference is technical, as we must assume the convergence of the nonlinear system (11) for part (a). The convergence of MI advantage to 1 of part (b) of Theorem 3 is implied by part (a) of the following theorem.

Theorem 6. *Consider the solution $\widehat{\boldsymbol{\beta}}$ to the optimization problem in* (2)*. Then*

(a) If the result of Theorem 5 holds and the minimum eigenvalue of $\boldsymbol{\Sigma}$ is lower bounded by a positive constant for sufficiently small δ, then as $\delta \to 0$, $\mathrm{Adv}(A) \to 1$.

(b) For the bilevel model in (4)*, if $p/d \to \phi \in (1, \infty)$ and d/n converges to a fixed value, then in the limit as $\lambda \to \infty$, $\mathcal{E}(f)$ is decreasing in ϕ.*

This theorem makes claims of two natures: that MI advantage of the adversary goes to 1, and that generalization error is decreasing. For the former, we will show that the output distributions diverge for train and test points such that it becomes trivial to distinguish between the two distributions, and for the latter, we will determine the form of the generalization error and show that it is decreasing in the proposed variable.

D.1 Part (a): membership inference advantage

We will assume the notation and setting from the proof of Theorem 5. When rewriting equations from (11), we will omit the bars (e.g., $\bar{\gamma}$ in the next section) when describing general implications of the equations, and then use bars to describe conclusions about the *unique* fixed point solution that characterizes the limiting estimator, which we assumed to exist in applying Theorem 4.

D.1.1 Growth of $\bar{\gamma}$

First, we show that $\bar{\gamma}$, the scaling factor of the proximal operator in Theorem 5, tends to infinity as $\delta \to 0^+$. This will drive training points to be much different from test points as long as the test point distribution variance doesn't increase. From the sixth equation in the nonlinear system (11), since the right hand side is greater than 0 and the fixed point variables are non-negative, we can conclude that $\sigma\tau > \gamma$. We can combine this with the second equation to yield

$$\begin{aligned}
\gamma &= \frac{1}{p}\mathbf{g}^\top \boldsymbol{\Sigma} \left(\boldsymbol{\Sigma} + \lambda\sigma\tau\mathbf{I}_p\right)^{-1} \mathbf{g}\frac{\sigma\tau}{\delta} && (26)\\
&= \frac{1}{\lambda\delta p}\mathbf{g}^\top \left(\frac{1}{\lambda\sigma\tau}\mathbf{I}_p + \boldsymbol{\Sigma}^{-1}\right)^{-1} \mathbf{g} && (27)\\
&> \frac{1}{\lambda\delta p}\mathbf{g}^\top \left(\frac{1}{\lambda\gamma}\mathbf{I}_p + \boldsymbol{\Sigma}^{-1}\right)^{-1} \mathbf{g} && (28)\\
&\xrightarrow{\text{a.s.}} \frac{1}{\lambda\delta p}\mathrm{tr}\left[\left(\frac{1}{\lambda\gamma}\mathbf{I}_p + \boldsymbol{\Sigma}^{-1}\right)^{-1}\right] && (29)
\end{aligned}$$

Because the smallest eigenvalue $\lambda_{\min}(\boldsymbol{\Sigma}) > 0$, this implies that

$$\lambda\gamma > \frac{1}{\delta}\frac{1}{\frac{1}{\lambda\gamma} + \frac{1}{\lambda_{\min}(\boldsymbol{\Sigma})}} \implies \frac{\lambda\gamma}{\lambda_{\min}(\boldsymbol{\Sigma})} > \frac{1}{\delta} - 1. \quad (30)$$

Therefore, asymptotically, there exists a constant $c_{\bar{\gamma}} > 0$ such that for sufficiently small δ, we have $\lambda\bar{\gamma} \geq c_{\bar{\gamma}}/\delta$, so $\bar{\gamma} \to \infty$ as $\delta \to 0^+$.

D.1.2 Vanishing of output variance.

We next argue that $\kappa^2\bar{\alpha}^2 + \bar{\sigma}^2$ tends to 0 as $\delta \to 0$. We remind the reader that as in the proof of Theorem 5, this is the variance of $\mathbf{x}_i^\top \widehat{\boldsymbol{\beta}}_{-i}$, which is also equal to the variance of the output for an unseen test point.

First, we consider the fourth equation in the nonlinear system (11). Applying the first-order optimality condition of the proximal operator, this is equivalent to

$$r^2 = 2\,\mathbb{E}\left[\rho'(-\kappa Z_1)\rho'\left(\mathrm{Prox}_{\gamma\rho}\left(\kappa\alpha Z_1 + \sigma Z_2\right)\right)^2\right] \leq 2. \tag{31}$$

Similarly, the fifth equation can be written as

$$\begin{aligned}\theta &= \frac{-2}{\gamma}\,\mathbb{E}\left[\rho''(-\kappa Z_1)\left(\kappa\alpha Z_1 + \sigma Z_2 - \gamma\rho'\left(\mathrm{Prox}_{\gamma\rho}\left(\kappa\alpha Z_1 + \sigma Z_2\right)\right)\right)\right] &(32)\\ &= 2\,\mathbb{E}\left[\rho''(-\kappa Z_1)\rho'\left(\mathrm{Prox}_{\gamma\rho}\left(\kappa\alpha Z_1 + \sigma Z_2\right)\right)\right] &(33)\\ &\leq \frac{1}{2}, &(34)\end{aligned}$$

where we have used the fact that the expectation of any odd function of a standard normal variable is zero, and that $\rho''(u) \leq 1/4$ for all $u \in \mathbb{R}$. Thus, both r and θ are upper bounded by constants. Let us now consider the third equation.

$$\begin{aligned}\kappa^2\alpha^2 + \sigma^2 &= \frac{(\sigma\tau)^2}{p}(\theta\widetilde{\boldsymbol{\beta}}^* + \frac{r}{\sqrt{\delta}}\mathbf{g})^\top \boldsymbol{\Sigma}^2\left(\boldsymbol{\Sigma} + \lambda\sigma\tau\mathbf{I}_p\right)^{-2}(\theta\widetilde{\boldsymbol{\beta}}^* + \frac{r}{\sqrt{\delta}}\mathbf{g}) &(35)\\ &\leq \frac{1}{\lambda^2}\left(\kappa^2\theta^2 + \frac{r^2}{\delta}\right) &(36)\\ &\leq \frac{1}{\lambda^2}\left(4\kappa^2 + \frac{1}{4\delta}\right). &(37)\end{aligned}$$

Here the first inequality is obtained by letting $\sigma\tau$ tend to infinity, and the second is obtained by applying our upper bounds for θ and r. Therefore, for sufficiently small δ, there exists c_1 such that $\kappa^2\alpha^2 + \sigma^2 \leq c_1^2/\delta$.

We now wish to return to (31) and (33) to determine tighter upper bounds. To that end, we first prove the following lemma

Lemma 7. *Let Z be a standard normal random variable. For any $a_0, b_0 > 0$, there exist $\delta_0 > 0$ and $c > 0$ such that for all $a \geq a_0$, $b \leq b_0$, and $0 < \delta < \delta_0$,*

$$\Pr\left(\mathrm{Prox}_{a\rho/\delta}\left(\frac{bZ}{\sqrt{\delta}}\right) > \log\left(c\delta\log(1/\delta)\right)\right) \leq \delta^2. \tag{38}$$

Proof. We begin by observing that is sufficient to prove the claim for $a = a_0$ and $b = b_0$, since the probability is monotonically decreasing and increasing, respectively, in each variable for sufficiently small δ. By standard Gaussian tail bounds, for sufficiently small δ,

$$\Pr(Z > 4\log(1/\delta)) \leq \delta^2. \tag{39}$$

The proximal operator is a strictly increasing function of Z, so we can determine the bound on its tail by determining an upper bound on $\mathrm{Prox}_{a\rho/\delta}\left(\frac{4b\log(1/\delta))}{\sqrt{\delta}}\right)$. The first-order optimality condition for the proximal operator is

$$w^* = \frac{4b\log(1/\delta))}{\sqrt{\delta}} - \frac{a}{\delta}\rho'(w^*). \tag{40}$$

It is clear that for sufficiently small δ, $w^* < 0$, since $\rho'(u) \geq 1/2$ for $u \geq 0$. Therefore, since $\rho'(u) = e^u/(1+e^u)$, there exists $c_\delta \in (1/2, 1)$ such that $\rho'(w^*) = c_\delta e^{w^*}$. We can then solve for and bound w^* for some $c > 0$ and sufficiently small δ as

$$\begin{aligned}w^* &= \frac{4b\log(1/\delta))}{\sqrt{\delta}} - W_0\left(\frac{ac_\delta}{\delta}\exp\left(\frac{4b\log(1/\delta))}{\sqrt{\delta}}\right)\right) &(41)\\ &\leq -\log\left(\frac{ac_\delta}{\delta}\right) + \log\left(\frac{4b\log(1/\delta))}{\sqrt{\delta}} + \log\left(\frac{ac_\delta}{\delta}\right)\right) &(42)\\ &\leq \log(c\delta\log(1/\delta)), &(43)\end{aligned}$$

where W_0 is the principal branch of the Lambert W function, and the first inequality follows from the lower bound $W_0(x) \geq \log x - \log\log x$ for $x \geq e$. Let δ_0 be a sufficiently small so that the above arguments hold, and the claim is proved. □

Applying Lemma 7 with $a_0 = c_{\bar{\gamma}}$ and $b_0 = c_1$ to (31), we can use the facts that $\rho'(u) \leq 1$ and that $\rho'(u) \leq e^u$ to obtain for some $c_r > 0$

$$r^2 \leq 2\left(c_r^2 \delta^2 \log^2(1/\delta) + \delta^2\right). \tag{44}$$

Thus for some $c_{\bar{r}} > 0$, $\bar{r} \leq c_{\bar{r}} \delta \log(1/\delta)$ for sufficiently small δ. We then apply Lemma 7 to (33) to similarly obtain for some $c_\theta > 0$

$$\theta \leq \frac{1}{2}\left(c_\theta \delta \log(1/\delta) + \delta^2\right) \tag{45}$$

Thus for some $c_{\bar{\theta}} > 0$, $\bar{\theta} \leq c_{\bar{\theta}} \delta \log(1/\delta)$ for sufficiently small δ. Therefore, returning again to (36), there exists some $c_2 > 0$ such that for sufficiently small δ,

$$\kappa^2 \bar{\alpha}^2 + \bar{\sigma}^2 \leq c_2^2 \delta \log^2(1/\delta). \tag{46}$$

Hence the output variance tends to zero as $\delta \to 0^+$.

D.1.3 Membership inference advantage

We wrap up the proof by proposing two more lemmas for the proximal operator of the logistic loss

Lemma 8. *Fix $C > 0$. For all v such that $|v| < C$ and $y \in \{0, 1\}$,*

$$\lim_{a \to \infty} |\mathrm{Prox}_{a\ell(y,\cdot)}(v)| = \infty \textit{ uniformly}, \tag{47}$$

where $\ell(y, z) = \log(1 + \exp(z)) - yz$ is the logistic loss.

Proof. The proximal operator $\mathrm{Prox}_{a\ell(y,\cdot)}(v)$ is the unique solution $w \in \mathbb{R}$ to the equation

$$w = v + a(y - \rho'(w)). \tag{48}$$

Consider $y = 1$, and suppose the claim was not true. Then there exists $c_1 > 0$ such that for all $a_0 > 0$, there exists $a > a_0$ and $v \in (-C, C)$ such that $|w| < c_1$. Let $c_2 = \rho'(c_1)$. This implies that

$$c_1 + ac_2 > v + a. \tag{49}$$

Since $c_2 < 1$, this inequality does not hold for any $a > a_0$ if a_0 is sufficiently large, leading to a contradiction. The case for $y = 0$ is entirely analogous if we make the substitution $\rho'(w) = 1 - \rho'(-w)$. □

Lemma 9. *Let Z be a standard normal random variable. Then for any $\tau > 0$, if a_n and b_n are sequences such that as $n \to \infty$, $a_n \to \infty$ and $b_n \to 0$, then*

$$\lim_{n \to \infty} \Pr\left(|\mathrm{Prox}_{a_n\ell(y,\cdot)}(b_n Z)| > \tau\right) - \Pr\left(|b_n Z| > \tau\right) = 1, \tag{50}$$

Proof. For sufficiently large n, by a standard tail bound for Gaussian variables, with probability at least $1 - e^{-(\tau/b_n)^2/2}$, we know that $|b_n Z| < \tau$. Again for sufficiently large n, we know that $|\mathrm{Prox}_{a_n\ell(y,\cdot)}(b_n Z)| > \tau$ for all $|b_n Z| < \tau$ by Lemma 8. Thus,

$$\Pr\left(|\mathrm{Prox}_{a_n\ell(y,\cdot)}(b_n Z)| > \tau\right) - \Pr\left(|b_n Z| > \tau\right) \geq 1 - 2e^{-(\tau/b_n)^2/2}, \tag{51}$$

which tends to 1 as $n \to \infty$. □

Applying Lemma 9 to our problem, using the fact that $\bar{\gamma} \to \infty$ and $\kappa^2 \bar{\alpha}^2 + \bar{\sigma}^2 \to 0$, we see that any adversary that applies a threshold $|\hat{f}(\mathbf{x})| > \tau$ for a fixed threshold τ will achieve MI advantage of 1 as $\delta \to 0$. Any loss-based fixed-threshold adversary inherits this behavior, as for the logistic loss, $\ell(y, \hat{f}(\mathbf{x}))$ is a monotonically decreasing function of $|\hat{f}(\mathbf{x})|$, so thresholding the loss is equivalent to thresholding the magnitude of the model output.

D.2 Part (b): test accuracy for the bi-level ensemble

In the bi-level ensemble, when applying Theorem 4 for $\widetilde{\boldsymbol{\beta}}^*$ in place of $\boldsymbol{\beta}^*$, asympotically, the first three equations in the nonlinear system (11) become

$$\begin{cases} \kappa^2\alpha = \dfrac{\sigma\tau\theta\kappa^2}{1+\frac{\lambda\sigma\tau}{\phi}}, \\ \gamma = \dfrac{\sigma\tau}{\delta}\left(\dfrac{1}{\phi+\lambda\sigma\tau}+\dfrac{\phi-1}{\phi+\frac{\lambda\sigma\tau}{\eta}(\phi-1)}\right), \\ \kappa^2\alpha^2+\sigma^2 = \dfrac{(\sigma\tau\theta\phi\kappa)^2+(\sigma\tau r)^2\frac{\phi}{\delta}}{(\phi+\lambda\sigma\tau)^2}+\dfrac{(\sigma\tau r)^2\frac{\phi}{\delta}(\phi-1)}{(\phi+\frac{\lambda\sigma\tau}{\eta}(\phi-1))^2}. \end{cases} \tag{52}$$

As we discussed in the proof of part (a), r and θ are always upper bounded by constants, so as $\lambda\to\infty$, regardless of the behavior of $\sigma\tau$, the left-hand sides of all three equations tend to zero. For this reason, applying our reformulations of the proximal operators and taking appropriate limits, the last three equations in the nonlinear system become

$$\begin{cases} r^2 = \dfrac{1}{4}, \\ \theta = \mathbb{E}\left[\rho''(-\kappa Z_1)\right], \\ \sigma\tau = 4. \end{cases} \tag{53}$$

These simplifications largely result from applying $\rho'(0)=1/2$ and appealing to symmetry arguments. The final equation results from the algebraic manipulation

$$\frac{\gamma}{\sigma\tau} = \mathbb{E}\left[2\rho'(-\kappa Z_1)\left(1-\frac{1}{1+\gamma\rho''\left(\operatorname{Prox}_{\gamma\rho}\left(\kappa\alpha Z_1+\sigma Z_2\right)\right)}\right)\right] \tag{54}$$

$$= \mathbb{E}\left[2\rho'(-\kappa Z_1)\frac{\gamma\rho''\left(\operatorname{Prox}_{\gamma\rho}\left(\kappa\alpha Z_1+\sigma Z_2\right)\right)}{1+\gamma\rho''\left(\operatorname{Prox}_{\gamma\rho}\left(\kappa\alpha Z_1+\sigma Z_2\right)\right)}\right]. \tag{55}$$

Now knowing that $\bar{\sigma}\bar{\tau}=4$, we can consider very large $\lambda\to\infty$ to obtain

$$\begin{cases} \alpha = \dfrac{\theta\phi}{\lambda}+o\left(\frac{1}{\lambda}\right), \\ \gamma = \dfrac{2}{\lambda\delta}+o\left(\frac{1}{\lambda}\right), \\ \kappa^2\alpha^2+\sigma^2 = \dfrac{1}{\lambda^2}\left((\theta\phi\kappa)^2+\dfrac{\phi}{4\delta}\left(1+\dfrac{\eta^2}{\phi-1}\right)\right)+o\left(\frac{1}{\lambda}\right). \end{cases} \tag{56}$$

Generalization error equals $\Pr\left(y\oplus\mathbb{1}\left\{\mathbf{x}^\top\widehat{\boldsymbol{\beta}}>0\right\}=1\right)$, where $\oplus$ is the exclusive or operator, which by symmetry we can compute as

$$\Pr\left(y\oplus\mathbb{1}\left\{\mathbf{x}^\top\widehat{\boldsymbol{\beta}}>0\right\}=1\right) = 2\Pr\left(y=0,\bar{\alpha}\mathbf{x}^\top\boldsymbol{\beta}^*+\bar{\sigma}Z>0\right) \tag{57}$$

$$= 2\mathop{\mathbb{E}}_{\mathbf{x}}\left[\Pr(y=0|\mathbf{x}^\top\boldsymbol{\beta}^*)\Phi\left(\frac{\mathbf{x}^\top\boldsymbol{\beta}^*}{\bar{\sigma}/\bar{\alpha}}\right)\right], \tag{58}$$

where $\Phi\colon\mathbb{R}\to[0,1]$ is the standard normal CDF, and Z is a standard normal random variable. It can be shown that this is decreasing in α/σ, and from the above, in the limit as $\lambda\to\infty$,

$$\frac{\bar{\alpha}^2}{\bar{\sigma}^2} = \frac{4\theta^2\frac{\delta}{\phi}}{1+\frac{\eta^2}{\phi-1}}, \tag{59}$$

which is increasing in ϕ for fixed $d/n=\delta/\phi$.

E Proof of Corollary 2

Proof. The generalization error result immediately follows from (58) the previous section, since

$$\kappa^2 = \left\|\widetilde{\boldsymbol{\beta}}^*\right\|_2^2/p = \boldsymbol{\beta}^{*\top}\boldsymbol{\Sigma}\boldsymbol{\beta}^*/p \to \sigma_\beta^2. \tag{60}$$

For membership advantage, we know from the previous section, Theorem 5, and Theorem 4 that the predictions on training and test points follow

$$\mathbf{x}_i^\top\widehat{\boldsymbol{\beta}} \xrightarrow{d} \mathrm{Prox}_{\bar{\gamma}\ell(y_i,\cdot)}\left(\bar{\alpha}Z_i + \bar{\sigma}W\right), \quad \mathbf{x}^\top\widehat{\boldsymbol{\beta}} \xrightarrow{d} \bar{\alpha}Z + \bar{\sigma}W, \tag{61}$$

where $\mathbf{x}_i^\top\boldsymbol{\beta}^* \xrightarrow{d} Z_i$, $\mathbf{x}^\top\boldsymbol{\beta}^* \xrightarrow{d} Z$, and $W \sim \mathcal{N}(0,1)$ is independent of Z_i or Z. Here randomness is over the training dataset, so for a fixed $\boldsymbol{\beta}^*$ and $\mathbf{x}_i$ (or $\mathbf{x}$), we have a fixed Z_i (or Z). Suppose the adversary is given some $\mathbf{x}'$ and its (noisy) training label y'. If $\mathbf{x}'$ (with corresponding Z') is *not* a training point,

$$\mu_{\text{test}}(\hat{z}|\mathbf{x}=\mathbf{x}', y=y') = \mu_{\text{test}}(\hat{z}|\mathbf{x}=\mathbf{x}') \tag{62}$$

$$= \mu_W\left(\frac{\hat{z}-\bar{\alpha}Z'}{\bar{\sigma}}\right)\frac{1}{\bar{\sigma}}. \tag{63}$$

The first equality is from the independence of the model output and the unused training label, and the second equality comes by the change of variables formula for scalar random variables in terms of μ_W, which is a standard normal Gaussian density.

If $\mathbf{x}'$ is a training point, we have the following probability density:

$$\mu_{\text{train}}(\hat{z}|\mathbf{x}=\mathbf{x}', y=y') = \mu_W\left(\frac{g_y(\hat{z})-\bar{\alpha}Z'}{\bar{\sigma}}\right)\frac{g_y'(\hat{z})}{\bar{\sigma}}. \tag{64}$$

Here $g_y(\cdot)$ is the inverse of $\mathrm{Prox}_{\bar{\gamma}\ell(y,\cdot)}(\cdot)$, which by the first-order optimality condition is

$$g_y(z) = z + \bar{\gamma}(\rho'(z) - y), \quad g_y'(z) = 1 + \bar{\gamma}\rho''(z). \tag{65}$$

We remind the reader that $\rho''(z) = \rho'(z)(1-\rho'(z))$. Therefore, the densities can be easily evaluated by numerical integration.

Since the adversary is given the value of the loss, which is monotonic in $\hat{f}(\mathbf{x}')$, and knows predicted label $\hat{y}(\mathbf{x}')$, the adversary is equivalent to an adversary based on $\hat{f}(\mathbf{x}')$ with the densities described above. The optimal adversary is given by

$$A^*(f, \mathbf{x}', y') = \mathbb{1}\left\{\mu_{\text{train}}(\hat{f}(\mathbf{x}')|\mathbf{x}=\mathbf{x}', y=y') > \mu_{\text{test}}(\hat{f}(\mathbf{x}')|\mathbf{x}=\mathbf{x}', y=y')\right\}, \tag{66}$$

and we can compute its MI advantage specific to $(\mathbf{x}', y')$ as

$$\mathrm{Adv}(A^*, \hat{f}; \mathbf{x}', y') = \int_{\mathbb{R}} \max\left\{\mu_{\text{train}}(z|\mathbf{x}=\mathbf{x}', y=y') - \mu_{\text{test}}(z|\mathbf{x}=\mathbf{x}', y=y'), 0\right\} dz, \tag{67}$$

Additionally, we can numerically evaluate this integral, and then we can compute the average sample-specific membership inference advantage as

$$\mathrm{Adv}(A^*, \hat{f}) = \mathop{\mathbb{E}}_{\mathbf{x}', y'}\left[\mathrm{Adv}(A^*, \hat{f}; \mathbf{x}', y')\right], \tag{68}$$

which we can easily compute by numerical integration over the Gaussian density of Z' and the fact that $\Pr(y'=1|\mathbf{x}') = \rho'(Z')$. □

F Train vs. test distribution simulation

To validate our derivation of the sample-specific output distribution, we compute empirical histograms over many trials for fixed data points and labels. See Figure 7.

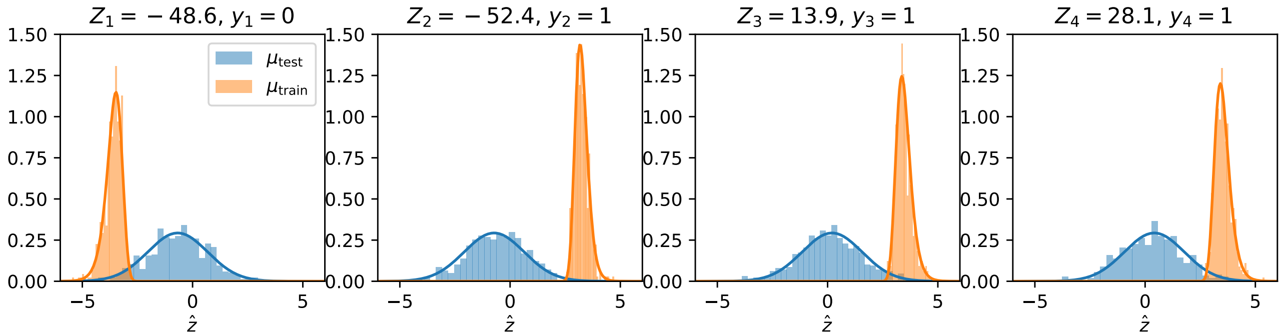


Figure 7: Theoretical densities (solid line) versus empirical histograms of sample-specific output distributions when $\mathbf{x}'$ is a test (blue) or training (orange) training point. For each training sample $(\mathbf{x}_i, y_i)$, $i \in \{1, 2, 3, 4\}$, we plot the corresponding theoretical densities $\mu_{\text{test}}(\hat{z}|\mathbf{x}' = \mathbf{x}_i, y' = y_i)$ and $\mu_{\text{train}}(\hat{z}|\mathbf{x}' = \mathbf{x}_i, y' = y_i)$ as described in Appendix E given $Z_i = \mathbf{x}_i^\top \boldsymbol{\beta}^*$ for a bi-level ensemble with $n/d = 1/2$, $\phi = 3$, $\sigma_\beta = 50$, $\lambda = 0.1$, $\eta = 1$. The histograms are of the outputs over 500 trials where $n = 500$ additional training points are used to train a logistic regression model on bi-level ensemble features either including or not including $(\mathbf{x}_i, y_i)$. Thus even for problems of real data sizes, the theory matches empirical results.

G Experimental setups

This section provides details on the experimental setups for sections 3.1, 3.2, 4 and figures 2, 3, 4, 5, and 6.

G.1 SVM experimental setup

This subsection provides details on the SVM experiments whose results are shown in Figure 2. For all SVM models, we use scikit-learn's SVC class [54]. When the number of SVM parameters is smaller than the number of data points in the training dataset, we add regularization $C = 1$, where C is the corresponding regularization parameter in scikit-learn's SVC class. Else, we use $C = 10^{20}$, essentially applying no regularization to yield the hard-margin SVM. The hard-margin SVM has been studied considerably in the double descent literature [39, 48], especially with regards to its relationship to logistic regression trained with gradient descent [55, 56].

We use the optimal MI adversary [13], which is a likelihood ratio test, as our MI attack. Suppose we are given two discrete distributions over values s_i with probability mass functions $q_{m=0}$ and $q_{m=1}$. The optimal adversary A^* is defined by:

$$A^*(s_i) = \begin{cases} 1 & \text{if } q_{m=1}(s_i) > q_{m=0}(s_i), \\ 0 & \text{otherwise} \end{cases}. \tag{69}$$

For each experiment, the general procedure is as follows. We first generate a D-dimensional data point $\mathbf{x}_0$ with binary label y_0, for some D. This is the data point on which MI will be performed. Then, for an integer p, we perform the following procedure L times. We first generate an $n \times p$ training dataset matrix $\mathbf{X}$, for some n, and a corresponding label vector $\mathbf{y}$. Generally, these are distributed in the same way as $(\mathbf{x}_0, y_0)$. All experiments here are binary classification tasks, so $\mathbf{y}_i \in \{-1, +1\}$ for $i \in \{1, 2, ..., n\}$. We then apply label noise to $\mathbf{y}$: we flip each label $\mathbf{y}_i$ to the other class with probability α. Afterwards, we learn an SVM on $\mathbf{X}$ and $\mathbf{y}$. We then denote by $\hat{y}_0$ the signed distance of $\mathbf{x}_0$ to the decision hyperplane of the learned SVM. We collect the $\hat{y}_0$ of all L learned SVMs into an output vector $\hat{\mathbf{y}}_{m=0}$. We then repeat the same procedure another L times, but this time, before learning the SVM on $\mathbf{X}$ and $\mathbf{y}$, we first replace the first rows $\mathbf{X}_1 = \mathbf{x}_0$ and $\mathbf{y}_1 = y_0$. Label noise is never applied to y_0. We collect the resulting L signed distances to the learned SVM hyperplanes into the output vector $\hat{\mathbf{y}}_{m=1}$. We form discrete histograms for both $\hat{\mathbf{y}}_{m=0}$ and $\hat{\mathbf{y}}_{m=1}$ with bin width b. Finally, we perform the optimal adversary attack on these histograms and measure the corresponding membership advantage. This entire experiment is repeated for multiple values p to generate Figure 2.

The following subsections provide the distributions of $\mathbf{x}_0$, y_0, $\mathbf{X}$, and $\mathbf{y}$, as well as the hyperparameters n (number of data points), D (full data dimensionality), label noise probability α, L (number of samples used to form the histogram), histogram bin width b, and the set of number of features p investigated for each data model.

G.1.1 Weak features

The weak features experiment is based on the weak features ensemble discussed in Definition 9 of [39]. In our experiment, we let $D = 1000$, $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{1}_D, \mathbf{I}_D)$, and $y_0 = 1$. We perform the experiment for $p \in \{5, 10, 15, ..., 95, 100, 200, 300, ..., 900\}$ with number of data points $n = 100$, number of samples $L = 20,000$, histogram bin width $b = 0.05$, and label noise probability $\alpha = 0.2$. The $n \times p$ training dataset matrix $\mathbf{X}$ is generated such that the i'th row $\mathbf{X}_i \sim \mathcal{N}(z_i, \mathbf{I}_p)$, with $z_i \sim \mathcal{N}(0, 1)$. The elements of the label vector $\mathbf{y}$ are defined by $y_i = \text{sign}(z_i)$. In essence, each element of a training data point $\mathbf{X}_i$ is the true signal z_i (on which the label y_i is based) corrupted by Gaussian noise.

G.1.2 Separable Gaussians

The separable Gaussians model is based on synthetic dataset 1 of [47] with some modifications. In our experiment, we set $D = 1000$ and generate $\mathbf{x}_0$ by sampling its individual elements as:

$$\mathbf{x}_{0,j} \sim \begin{cases} \mathcal{N}(1, 1) & \text{if } j \leq 100 \\ \mathcal{N}(0, 1) & \text{otherwise} \end{cases}, \tag{70}$$

with true label $y_0 = 1$. We perform the experiment for $p \in \{10, 20, 30, ..., 90, 100, 150, 200, 250, ..., 1000\}$ with number of data points $n = 100$, number of samples $L = 10,000$, and histogram bin width $b = 0.05$.. Each element of the label vector $\mathbf{y}_i$ is randomly selected from $\{-1, +1\}$ with uniform probability. The individual elements (row i and column j) of the training dataset matrix $\mathbf{X}$ are distributed as:

$$\mathbf{X}_{i,j} \sim \begin{cases} \mathcal{N}(\mathbf{y}_i, 1) & \text{if } j \leq 100 \\ \mathcal{N}(0, 1) & \text{otherwise} \end{cases}. \tag{71}$$

Label noise with probability $\alpha = 1$ is then applied to $\mathbf{y}$ after $\mathbf{X}$ is generated. Essentially, the first $\min(100, p)$ features of each data point depend on its true class, and the remaining features are irrelevant (independent of the class). Thus, in the overparameterized regime, as p increases, we are including more irrelevant features to the model.

G.1.3 Random ReLU features

The random ReLU features model has been studied by multiple papers, such as [48] (section 3) and [57]. In essence, it is a two-layer ReLU neural network with fixed first-layer random weights. Different from the previous SVM data models, here $\mathbf{x}_0$ is defined differently for each trained SVM model because of the random projections. Instead, there is a latent data vector $\mathbf{z}_0$ that is kept fixed for all experiments and on which MI is performed. We set $D = 200$, and generate $\mathbf{z}_0$ by sampling it from $\mathcal{N}(0, \mathbf{I}_D)$. We perform the experiment for $p \in \{10, 20, 30, ..., 90, 100, 150, 200, ..., 950\}$ with number of data points $n = 100$, number of samples $L = 100,000$, histogram bin width $b = 0.001$, and no label noise. To generate the training data, a random $p \times D$ "featurizer" matrix $\mathbf{W}$ is first generated by sampling each row independently from the D-dimensional unit sphere. Then, an $n \times D$ feature data matrix $\mathbf{Z}$ is generated by sampling each element iid standard normal. The training data matrix $\mathbf{X} = \max(0, \mathbf{Z}\mathbf{W}^T)$, where the max operation is applied elementwise. The MI data point $\mathbf{x}_0$ is defined as $\mathbf{x}_0 = \max(0, \mathbf{z}_0^\top \mathbf{W}^\top)$. Note that since $\mathbf{W}$ is sampled for each trained SVM, $\mathbf{x}_0$ changes for each experimental run. To generate the labels of the data points, first, a random vector β is sampled uniformly from the D-dimensional sphere of radius 4 (such that $||\beta||_2 = 4$). Then, y_i is assigned class 1 with probability $\frac{1}{1+e^{-\mathbf{Z}\beta}}$ and class -1 otherwise. The label y_0 of x_0 is defined similarly and is assigned class 1 with probability $\frac{1}{1+e^{-\mathbf{Z}\beta}}$ and class -1 otherwise. Essentially, the class of a data point depends only on $\mathbf{Z}$, and the training set consists of random projections of $\mathbf{Z}$ that are then passed through the ReLU operation.

G.1.4 CIFAR10

To experiment on real data, we train SVMs on random projections of a subset of the CIFAR10 dataset [49]. We first define $\mathbf{z}_0$ to be the first image of the training dataset with class "airplane" converted to grayscale and then vectorized. We perform the experiment for $p \in \{10, 20, 30, ..., 90, 100, 200, 300, ..., 1800, 1900\}$ with number of data points $n = 200$, number of samples $L = 10,000$, histogram bin width $b = 0.05$, and no label noise. To generate the $n \times p$ data matrix, we first randomly sample $\frac{n}{2}$ images uniformly from the "airplane" images of the dataset (excluding $\mathbf{x}_0$) and $\frac{n}{2}$ images from the "automobile" images of the dataset. We convert the images to grayscale, vectorize them, and collect them into a matrix $\mathbf{Z}$ (where each row is a vectorized image). Since each image is of size 32×32, the vectorized image is $D = 1024$ dimensional. We then sample a $p \times 1024$ random projections matrix $\mathbf{W}$, where each row is sampled uniformly from the 1024-dimensional unit sphere. Finally, the data matrix $\mathbf{X} = \mathbf{Z}\mathbf{W}^\top$. The MI point $\mathbf{x}_0 = \mathbf{z}_0^\top \mathbf{W}^\top$. The labels of each data point is -1 if it originated from an "airplane" image and $+1$ if it originated from an "automobile" image.

G.2 Neural network experimental setup

This subsection provides details on the NN experiments whose results are shown in Figures 3, 4, 5, and 6. Unless otherwise specified, we use the default hyperparameters and initalizations of Pytorch implementations. The NN experiments are run on our internal servers with the following GPUs: NVIDIA TITAN X (Pascal), NVIDIA GeForce RTX 2080 Ti, NVIDIA TITAN RTX, and NVIDIA A100. The choice of which particular GPU is used for each experiment is decided only based on availability of the GPUs in our internal servers.

G.2.1 The MI attack

The MI attack employed in these experiments is the loss-threshold attack [22, 16, 23]. Given a trained NN f, the data point of interest $\mathbf{z}_0 = (\mathbf{x}_0, y_0)$, and a loss function ℓ, the prediction $A(f(\mathbf{x}_0), \mathbf{z}_0)$ of this attack is given by:

$$A(f(x_0), y) = \begin{cases} 1 & \text{if } \ell(y_0, f(\mathbf{x}_0)) < \tau_{\mathbf{z}_0} \\ 0 & \text{otherwise} \end{cases}, \tag{72}$$

where $\tau_{\mathbf{z}_0}$ is a calibrated threshold. The threshold is learned with the following procedure. Given a full training dataset $\mathcal{D}$, we train n_{shadow} shadow models on random subsamples of this dataset such that for each $\mathbf{z}_0$ in the full dataset, some models are trained on datasets including $\mathbf{z}_0$ and the rest are trained on datasets that do not include $\mathbf{z}_0$. The shadow models have the same architecture and training procedure as the target models that will be attacked. Let $n_{\text{shadow},\mathbf{z}_0,m=1}$ and $n_{\text{shadow},\mathbf{z}_0,m=0}$ denote the (random) numbers of shadow models trained on $\mathbf{z}_0$ and not trained on $\mathbf{z}_0$, respectively. We then evaluate all these shadow models on $\mathbf{z}_0$ and collect all loss values of the shadow models trained on $\mathbf{z}_0$ into a vector $\mathbf{s}_{\mathbf{z}_0,m=1}$ and the loss values of the shadow models not trained on $\mathbf{z}_0$ into a vector $\mathbf{s}_{\mathbf{z}_0,m=0}$. The membership advantage $\text{Adv}_{\text{shadow}}$ of a threshold $\hat{\tau}_{\mathbf{z}_0}$ is given by:

$$\text{Adv}_{\text{shadow},\mathbf{z}_0} = \frac{\left|\left\{s \in \mathbf{s}_{\mathbf{z}_0,m=1} :\, s < \tau_{\mathbf{z}_0}\right\}\right|}{n_{\text{shadow},\mathbf{z}_0,m=1}} - \frac{\left|\left\{s \in \mathbf{s}_{\mathbf{z}_0,m=0} :\, s < \tau_{\mathbf{z}_0}\right\}\right|}{n_{\text{shadow},\mathbf{z}_0,m=0}}, \tag{73}$$

which is simply the difference of the empirical true positive rate and false positive rate. Note that there are many optimal thresholds that maximize $\text{Adv}_{\text{shadow}}$. Indeed, if $\hat{\tau}_{\mathbf{z}_0}$ is one such optimal threshold, then so is any $\tau \in [s^*_{m=1}, s^*_{m=0}]$, where $s^*_{m=1}$ is the closest element in $\mathbf{s}_{m=1}$ that is less than $\tau_{\mathbf{z}_0}$ and $s^*_{m=0}$ is the closest element in $\mathbf{s}_{m=0}$ that is greater than $\tau_{\mathbf{z}_0}$. Thus, we set the attack's calibrated loss threshold as the midpoint: $\tau_{\mathbf{z}_0} = \frac{1}{2}(s^*_{m=1} + s^*_{m=0})$. This sample-based loss threshold attack, wherein a different threshold is learned for each data point $\mathbf{z}_0$, is the attack we use for the CIFAR10 and Multi30k experiments.

A variation of this attack that we apply for the Purchase100 dataset is the global loss threshold, where $\tau_{\mathbf{z}_0} = \tau$ for every $\mathbf{z}_0$. In words, the same threshold value is applied when attacking the model on any data point. The procedure for threshold calibration is the same, except now $\mathbf{s}_{m=1}$ contains the losses for each of the data points each model was trained on and $\mathbf{s}_{m=0}$ contains the losses for the data points the models were not trained on.

G.2.2 Evaluation procedure

To evaluate the attack, we first randomly subsample a training dataset $\mathcal{S}$ from the full training dataset $\mathcal{D}$ and train a target model on $\mathcal{S}$. Denote by $\bar{\mathcal{S}}$ the data points in $\mathcal{D}$ that are not in $\mathcal{S}$. We collect the losses $t(\mathbf{z}_0)$ of the target model on each data point $\mathbf{z}_0$ in $\mathcal{S}$ into a vector $\mathbf{t}_{m=1}$ and the losses of the target model on each data point in $\bar{\mathcal{S}}$ into a vector $\mathbf{t}_{m=0}$. The membership advantage for the target model is:

$$\text{Adv}_{\text{target}} = \frac{\left|\left\{t(\mathbf{z}_0) \in \mathbf{t}_{m=1} :\, t(\mathbf{z}_0) < \tau_{\mathbf{z}_0}\right\}\right|}{|\mathcal{S}|} - \frac{\left|\left\{t(\mathbf{z}_0) \in \mathbf{t}_{m=0} :\, t(\mathbf{z}_0) < \tau_{\mathbf{z}_0}\right\}\right|}{\bar{\mathcal{S}}}. \tag{74}$$

We repeat this evaluation procedure n_{target} times, each time training a new target model on a newly sampled $\mathcal{S}$. The mean and standard deviation of the membership advantage over all experimental runs is what is reported in the paper figures.

Each shadow and target model is trained for E epochs with checkpoints saved every C epochs, where E and C differ per dataset. For the experiments in Section 3.1 and Figure 3, the checkpoints for each shadow and target model that achieves the highest classification accuracy rate on the dataset's validation set is used for the experiment. The results in Figures 4 and 5 are obtained for each checkpoint. For each curve in Figure 6, for all shadow and target models, we use the same number of epochs: the number of epochs (out of the checkpoints acquired) that achieves a membership advantage (Figure 6a) or test error (6b) closest to the one specified in the figure.

G.2.3 Datasets and architectures

We split each dataset into a "full training dataset" and a validation set. The full training dataset contains all the points on which membership inference will be performed. Each shadow and target model will be trained on a sample of the full training dataset such that the full training dataset would always contain both members (training points) and non-members (test points) for each model. The validation set is only used for calculating classification test error.

Classification on Purchase100. The Purchase100 dataset is based on Kaggle's "acquire valued shoppers" challenge dataset subsequently processed by [11]. It contains 197,324 length-600 binary feature vectors, each belonging to 1 of 100 classes. Each feature vector corresponds to a purchaser, and each entry of the vector corresponds to whether or not a particular product was purchased by the customer. The 100 classes correspond to purchasing styles. We use the first 180,000 data points as the full training dataset and the remaining data points for the validation set. We train two-layer neural networks with hidden dimension w, which we vary. We set $n_{\text{shadow}} = 50$ and $n_{\text{target}} = 50$. Each model is trained on a random sample of 10,000 data points. We use the ADAM optimizer [58] with a learning rate of 0.001 for $E = 3000$ epochs with checkpoints saved every $V = 20$ epochs. For Figure 5, we save checkpoints every $V = 1$ epoch and only display the results for less than 3000 epochs for better visualization (each curve uses a different number of epochs, according to which provides best visualization).

Image classification on CIFAR10. The CIFAR10 dataset [49] contains 60,000 32×32 RGB images, each belonging to 1 of 10 object classes. We use the 50,000 images in the official training dataset as our full training dataset, and the 10,000 images in the official validation dataset as our validation set. We train ResNet18 models [50] to perform image classification on the dataset. To vary the models' widths, we follow [4] and use convolutional layer widths (number of filters) of $[w, 2w, 4w, 8w]$ for different w values. Note that $w = 64$ yields the original ResNet18 architecture. We set $n_{\text{shadow}} = 50$ and $n_{\text{target}} = 50$, where each model is trained on a random sample of 25,000 images. We train for 50,000 gradient steps using the ADAM optimizer [58] with a batch size of 128 (amounting to ≈ 256 epochs through the training dataset), a learning rate of 0.0001 and the cross-entropy loss. Data augmentation is a common technique used in image classification, and so we also employ random translations of up to 4 pixels and random horizontal flipping during training, as was done by [4].

Language translation on Multi30K. The Multi30K dataset [51] consists of 29,001 pairs of English-German sentences. We perform English to German translation on these sentences using the Transformer architecture [52]. To vary the models' widths, we follow [4] and set the encoder/decoder feature sizes to w and the fully connected layers' dimensions to $4w$ for different values of w. We train for 300 epochs using the ADAM optimizer [58] with a learning rate of 0.0001, a batch size of 128, and the cross-entropy loss over each token. We set $n_{\text{shadow}} = 15$ and $n_{\text{target}} = 15$ and train each model on a random sample of $14,500$ sentence pairs. In calculating the loss of a sentence pair for performing membership inference, we sum the cross-entropy loss values over all tokens in the sentence and divide by the sentence length.

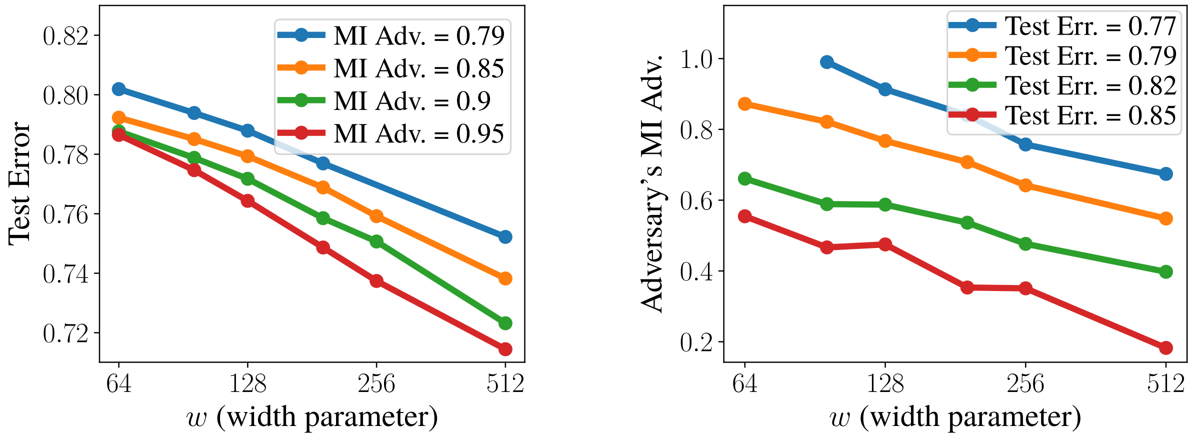
H Additional Experiments

H.1 Blessing of Dimensionality for Multi30K

In Figure 8, we show the equivalent of Figure 6 in the main paper for the transformer architecture on the Multi30k dataset. Similarly to the Purchase100 and CIFAR10 datasets, increasing the width of the neural network here improves either privacy (i.e. decreases membership advantage) or test accuracy when holding the other fixed via proper epoch tuning.

H.2 Global Loss Threshold Attack

In Figures 3, 4, 5, and 6, we used the sample-specific loss threshold attack for CIFAR10 and Multi30K, where a different loss threshold is learned for each data point. Here, we repeat the same experiments using the global loss threshold, where a single threshold value is used for all the data points. Note that in the mentioned figures, we already employed the global loss threshold attack for Purchase100. The trends we observe for the global loss threshold attack are similar to those of the sample-specific loss threshold attack. The results are shown in Figures 9, 10, 11, and 12. We use $n_{\text{shadow}} = n_{\text{target}} = 15$ for both datasets in this experiment.



(a) Test error vs. net width for fixed MI adv.

(b) MI adv. vs. net width for fixed test error.

Figure 8: **Overparameterization with early stopping eliminates the privacy–utility trade-off on Multi30k.** This is similar to Figure 6 in the main body, but performed on the Multi30k dataset with the Transformer architecture. (a) For each network width, we train the network until it reaches a given MI advantage value. We then plot the test error of the networks. Observe how test error decreases with parameters at a fixed MI advantage value. Thus, this eliminates the privacy–utility trade-off. Proper tuning of parameters and epochs together improves model accuracy without damaging its privacy. (b) Same as (a) but switching the roles of MI advantage and test error.

H.3 Privacy-Utility Trade-offs for DP-SGD on CIFAR10

We perform the same experiment of Figure 5 for CIFAR10 with ResNet18 models trained with DP-SGD [31]. In DP-SGD, gradients are clipped to a maximum bound, and noise is added to the gradients before the gradient descent step. The model training procedure is guaranteed to be (ϵ, δ) differentially private for some ϵ and δ according to the amount of noise added and the number of training epochs. The addition of noise also serves as a form of regularization. We thus obtain the regularization-wise privacy-utility trade-off for each network width by varying the amount of noise added. Specifically, we set the gradient clipping bound to 1, the number of epochs to 200, and δ to $\frac{1}{25000}$. For each $\epsilon \in \{1, 2, 3, ..., 14, 15, 16, 20, 50, 100\}$ and each learning rate in $\{0.1, 0.5, 1, 2, 4, 8\}$, we train 5 networks with noise added to the gradients such that the procedure is (ϵ, δ) differentially private. Smaller ϵ parameters yield more noise, which serves as increased regularization. We try different learning rates as it has been observed that learning rate tuning can affect DP-SGD performance. We apply the global loss threshold attack and plot the mean test errors and mean membership advantage across the 5 networks for each ϵ and learning rate for different model widths in Figure 13. For each network width, we only include its Pareto optimal points. That is, we exclude a point if there exists another point that has both lower test error and lower membership advantage. We observe the same phenomenon as in Figure 5. Wider networks enjoy better privacy-utility trade-offs than narrower networks.

H.4 TPR at FPR=1%

In Figure 14, we perform the same experiment as in Figure 5 of the main paper, but we instead use the global loss threshold attack and report the maximum achievable true positive rate (TPR) when the false positive rate (FPR) is constrained to be at most 1%. For the loss threshold attack, the adversary predicts the data point to be a member if the model's loss on the data point is below some τ. When τ is increased, the adversary more frequently predicts the data point as being a member. This increases the adversary's TPR, but it will also increase its FPR. For the attack used in Figure 14, we choose the global thresholds for each individual network that maximizes the TPR under the constraint that the FPR is at most 1%. We refer readers to [17] for additional discussion on using the metric of TPRs for constrained FPRs. In this metric, we still observe the same blessing of dimensionality: wider networks can achieve lower test error and lower MI adversary TPRs than their narrower counterparts.

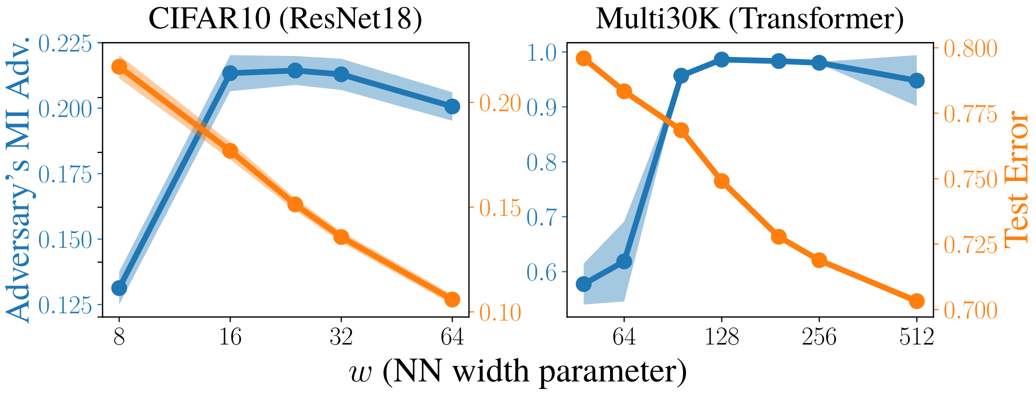


Figure 9: **Privacy vs. parameters (global loss threshold attack).** We repeat the experiment in Figure 3, but now using the global (instead of sample-specific) loss threshold attack. Similarly, wider networks generally suffer from higher vulnerability to MI attacks while achieving lower test error.

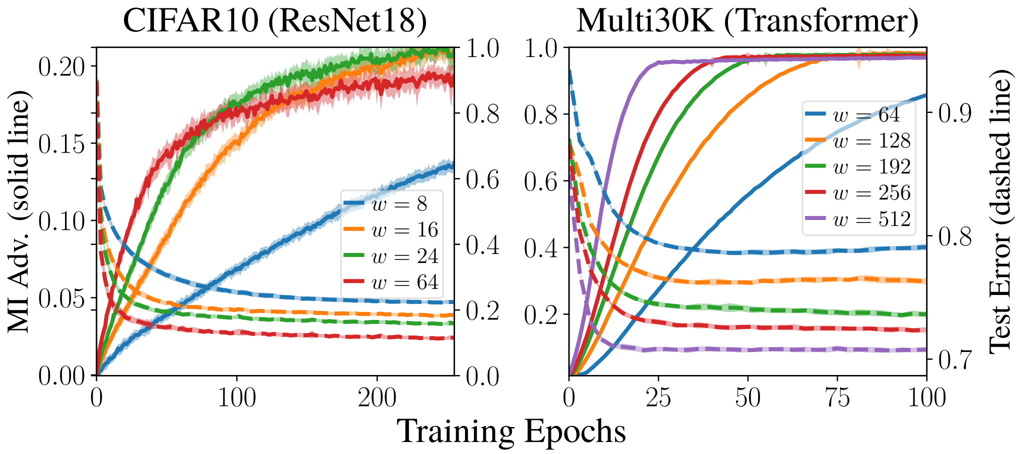


Figure 10: **Privacy vs. epochs (global loss threshold attack).** We repeat the experiment in Figure 4, but now using the global (instead of sample-specific) loss threshold attack. Again, as epochs increase, membership advantage increases while test error decreases.

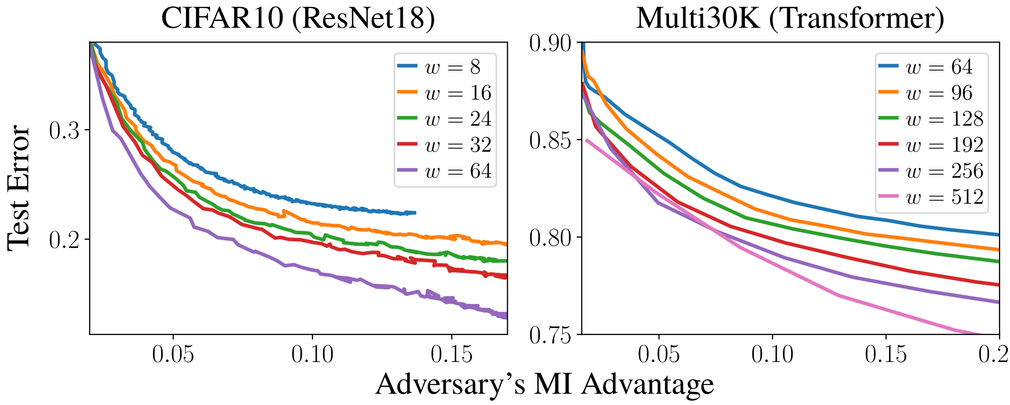
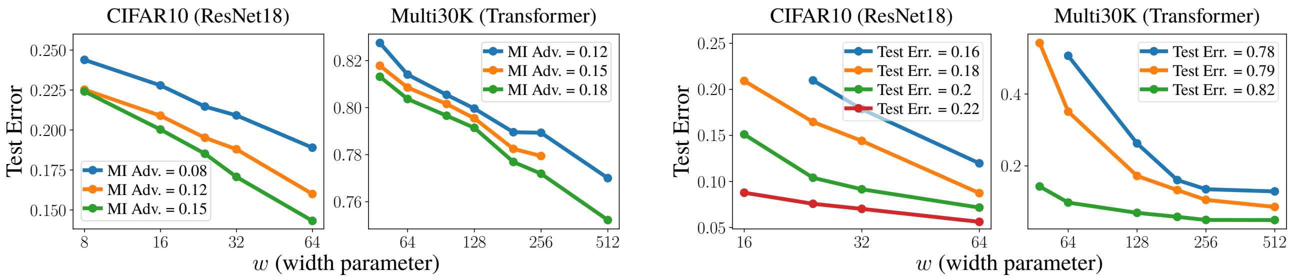


Figure 11: **Trade-offs (global loss threshold attack).** We repeat the experiment in Figure 5, but now using the global (instead of sample-specific) loss threshold attack. We observe again how wider networks are closer to the lower-left portion of the graph, indicating better privacy and better test accuracy compared to their narrower counterparts.



(a) Test error vs. net width for fixed MI adv.

(b) MI adv. vs. net width for fixed test error.

Figure 12: **Overparameterization with early stopping eliminates the privacy–utility trade-off (global loss threshold).** Similar to Figures 6 and 8, but using the global loss threshold. Increasing the parameters can improve either privacy or test accuracy when keeping the other fixed (by epoch tuning).

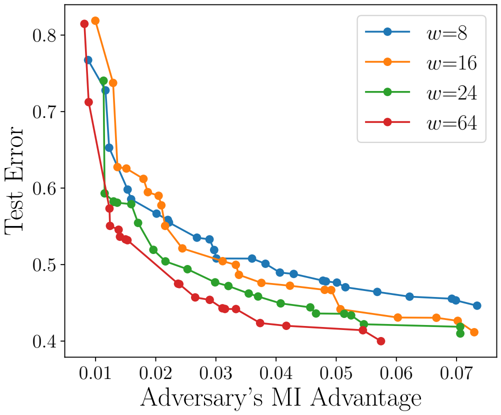


Figure 13: **DP-SGD Trade-off.** We train ResNet18 networks on CIFAR-10 with DP-SGD. We sweep through $\epsilon \in \{1, 2, 3, ..., 15, 16, 20, 50, 100\}$ and learning rates $\{0.1, 0.5, 1, 2, 4, 8\}$. For each ϵ and learning rate, we train 5 networks. Each point on the plot corresponds to the mean test error and mean MI advantage of the global loss threshold attack over the 5 networks for some ϵ and learning rate. We only include points that are Pareto optimal—we exclude a point if there exists another point with both lower test error and lower MI advantage. We fix the clipping bound to 1 and the number of epochs to 200. The plot shows that wider ResNet18 networks achieve better privacy–utility trade-offs than narrower networks when tuning the DP-SGD noise amount added.

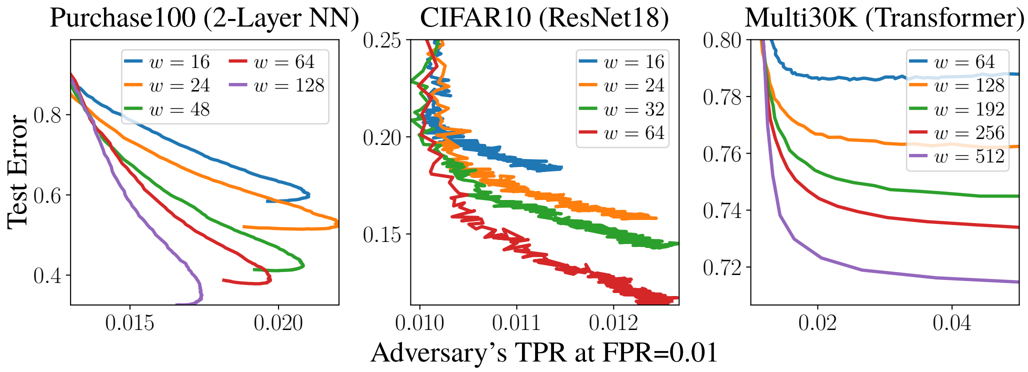


Figure 14: **TPR at FPR=1%.** We show the privacy-utility trade-offs similar to Figure 5 but reporting the global loss threshold's true positive rate (TPR) using the threshold value that maximizes TPR under the constraint that the false positive rate ≤ 0.01. We again observe wider networks enjoying better privacy-utility trade-offs than narrower ones.