# *Explaining the Model, Protecting Your Data*: Revealing and Mitigating the Data Privacy Risks of Post-Hoc Model Explanations via Membership Inference

**Catherine Huang** [1] **Martin Pawelczyk** [1] **Himabindu Lakkaraju** [1]

## Abstract

Predictive machine learning models are becoming increasingly deployed in high-stakes contexts involving sensitive personal data; in these contexts, there is a trade-off between model *explainability* and data *privacy*. In this work, we push the boundaries of this trade-off: with a focus on foundation models for image classification fine-tuning, we reveal unforeseen privacy risks of post-hoc model explanations and subsequently offer mitigation strategies for such risks. First, we construct VAR-LRT and L1/L2-LRT, two new membership inference attacks based on feature attribution explanations that are significantly more successful than existing explanation-leveraging attacks, particularly in the low false-positive rate regime that allows an adversary to identify specific training set members with confidence. Second, we find empirically that optimized differentially private fine-tuning substantially diminishes the success of the aforementioned attacks, while maintaining high model accuracy. We carry out a systematic empirical investigation of our 2 new attacks with 5 vision transformer architectures, 5 benchmark datasets, 4 state-of-the-art post-hoc explanation methods, and 4 privacy strength settings.

## 1. Introduction

Predictive machine learning (ML) models are becoming increasingly deployed in high-stakes contexts such as medical diagnoses and loan approvals. Since these models rely on sensitive personal data, regulatory principles that enforce safe and trustworthy model training and usage have become increasingly important. One key regulatory principle is the *Right to Privacy*, which aims to protect against training data leakage (Weller, 2019). The right to privacy is a limit to model *explainability*, which is itself another important pillar of trustworthy ML. Given the inherent complexity of high-stakes models, explanations are increasingly necessary in offering users information about how models make decisions with respect to data. It is common to use *post-hoc* explanations that explain the behavior of a trained model on a *specific* data example; *feature attributions* are a broad and commonly studied sub-class of post-hoc explanations (Lundberg & Lee, 2017; Ribeiro et al., 2016; Shrikumar et al., 2017).

One widely used standard to empirically verify whether a model obeys privacy is membership inference attacks (MIAs) (Shokri et al., 2017), which predict if a data example was used to train a target model. Successful MIAs are a violation of privacy—if an adversary knows that a patient's medical record was used to train a model that predicts the optimal treatment for a particular disease, the adversary can correctly conclude that this patient has the disease.

There is limited work on the susceptibility of model explanations to membership inference, let alone work on the privacy risks of model explanations altogether. Our work reveals unforeseen data privacy violations of post-hoc feature attribution explanations through addressing the following question: Can we devise new membership inference attacks on feature attribution explanations that 1) have higher success than existing explanation-leveraging attacks, and 2) allow an adversary to confidently identify specific members of the training set in the "low false-positive rate regime"?

Furthermore, despite the literature on the privacy risks of model explanations, there has been little work on actually *mitigating* these risks through privacy-preserving methods. We hypothesize that the mathematical framework *differential privacy (DP)* (Dwork & Roth, 2014) can be used during training to mitigate privacy violations of explanations, and we address this under-explored problem through the following question: Does differential privacy defend against an adversary's ability to leverage model explanations to infer sensitive training data membership information, while preserving model accuracy?

[*] Equal contribution [1] Harvard University. Correspondence to: Catherine Huang <catherinehuang@college.harvard.edu>.

*Proceedings of the $41^{st}$ International Conference on Machine Learning*, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

## 2. Related Work

Prior work shows that explanations do risk leaking sensitive training data information via membership inference: Shokri et al. (2021) and Pawelczyk et al. (2022) show that backpropagation-based explanations and algorithmic recourse, respectively, can leak training set membership information. However, this existing work is limited: Shokri et al.'s attacks are evaluated using average-case metrics that do not characterize whether the attack can confidently identify any specific members of the training set. Pawelczyk et al.'s work highlights that an adversary can accurately identify specific training set members with high confidence but focuses only on counterfactual explanations of binary classification models, thus not addressing a broader class of feature attribution explanations on more complex deep classification models. Moreover, both works involve training low-dimensional real-world datasets; neither addresses the privacy risks of explanations coming from models trained or fine-tuned on complex datasets.

Liu et al. (2024) develop a attack that trains a shadow model to infer membership based on how the target model behaves differently when the input is perturbed according to its feature attribution explanation. This recent work addresses the limitations of the aforementioned works by systematically demonstrating strong attack success, particularly at identifying specific training data members, in a deep learning setting.

In response to work on attacks against deep learning models, privacy-preserving deep learning has risen in importance as a research area. Albeit limited, there exists emergent work on differentially private (DP) computation of post-hoc explanations. Huang et al. (2023) propose and evaluate two methods to generate DP counterfactual explanations in logistic regression models. Even though Huang et al. find that DP reduces what an adversary can infer about training set membership, this work is limited specifically to counterfactual explanations of logistic regression models. They do not address privacy-preserving explanations in deep learning settings (in which privacy is a greater concern) or the broader class of post-hoc feature attribution explanations.

**Where Our Work Stands** Our work extends that of Shokri et al. and Pawelczyk et al. by developing successful MIAs, leveraging feature attribution explanations on deep image classification models, that can confidently identify specific training set members at low FPR. Our attacks differ from Liu et al.'s in terms of adversarial information access: Liu et al.'s attack requires access to target model losses and entire attribution vectors, but our attack assumes no access to losses and utilizes only one-dimensional summaries of attributions. We extend Huang et al.'s work by offering a unified evaluation, which current literature lacks, of the impact of DP model training on feature attribution-based MIA performance in deep learning settings. Furthermore, our attacks are on foundation models and require no training from scratch, not even of the *shadow models* used in the attacks—we conduct membership inference from *fine-tuning* alone, a less computationally expensive process.

## 3. Preliminaries

Let $D_{train} = \{\mathcal{X}, \mathcal{Y}\} = \{(\mathbf{x_i}, y_i)\}_{i=1}^{N}$ be a training dataset drawn from some underlying distribution $\mathbb{D}$. Let $f_\theta$ be the model parameterized by $\theta$, $\mathbf{x} \in \mathbb{R}^d$ be an input feature vector, and $y \in [k]$ be an output label. $\mathcal{X} \in \mathbb{R}^{N \times d}$ denotes the feature set, and $\mathcal{Y} \in [k]^N$ denotes the labels over $\mathcal{X}$.

**The Case for Foundation Models** In this work, we evaluate pre-trained vision transformers on image classification fine-tuning tasks—membership inference, in our case, seeks to infer whether an example was used to *fine-tune* the model. Appendix B discusses the choice of studying foundation models and specifically the vision transformer architecture.

**Post-Hoc Feature Attribution Explanations** A **post-hoc explanation** function $\varphi$ takes as input a trained model $f_\theta$ and a point of interest $\mathbf{x} \in \mathbb{R}^d$. A **feature attribution** post-hoc explanation $\varphi(\mathbf{x})$ is a $k$-dimensional vector whose $i$-th coordinate, $\varphi_i(\mathbf{x})$, reflects the extent to which the $i$-th feature influences the prediction the model outputs for $\mathbf{x}$. We study the following feature attribution methods: Input $*$ Gradient (IXG), Saliency Maps (SL), Integrated Gradients (IG), and (a gradient-based approximation to) SHAP (GS). We describe each of these methods in Appendix C.

**Membership Inference Attacks** Suppose an adversary possesses a set of data examples. The goal of a **membership inference attack (MIA)** is for an adversary to create a function that predicts whether each data example belongs to the training set of $f_\theta$. MIAs are predominantly loss-based, testing if the loss of the model for each example is below some threshold; we note that loss-based attacks require adversarial access to true labels. Traditionally, MIAs are evaluated using average-case metrics such as the receiver operating characteristic (ROC) curve—which plots attack true positive rate (TPR) against false positive rate (FPR)—and the area under that curve (AUC).

**Likelihood Ratio Attacks and the Low-FPR Regime** Carlini et al. (2021) propose a re-formulation of the MIA problem to focus not on average-case performance but rather on the "low FPR regime." If an MIA has high TPR at low FPR, that means it can *confidently* identify the training set membership of a few observations in a sensitive dataset. Attack success at low FPR is a greater privacy violation than an attack that only *unreliably* achieves high aggregate success rate. Carlini et al. also initiated the practice of reporting *log-scaled* ROC curves, rather than linearly scaled curves, to make visible TPRs at very low FPRs.

Carlini et al. additionally propose the **Likelihood Ratio Attack (LiRA)** that is significantly more successful, in particular at low FPRs, than prior MIAs (Shokri et al., 2017; Yeom et al., 2018; Jayaraman et al., 2021; Song & Mittal, 2021; Ye et al., 2022; Watson et al., 2022; Sablayrolles et al., 2019; Long et al., 2020). In LiRA, the adversary trains shadow models on datasets with and without target example $(\mathbf{x}, y)$. Let $\mathbb{Q}_{in}(\mathbf{x}, y) = \{f \leftarrow \mathcal{T}(D_{attack} \cup \{(\mathbf{x}, y)\} \mid D_{attack} \leftarrow \mathbb{D}\}\}$ represent the distribution of models trained on datasets containing $(\mathbf{x}, y)$. Likewise, we have $\mathbb{Q}_{out}(\mathbf{x}, y) = \{f \leftarrow \mathcal{T}(D_{attack} \setminus \{(\mathbf{x}, y)\} \mid D_{attack} \leftarrow \mathbb{D}\}\}$. The adversary estimates the likelihood ratio $\hat{\Lambda}(f_\theta; (\mathbf{x}, y)) \approx \frac{p(f_\theta|\mathbb{Q}_{in}(\mathbf{x},y))}{p(f_\theta|\mathbb{Q}_{out}(\mathbf{x},y))}$ and then thresholds on $\hat{\Lambda}$: $\text{Membership}_{\text{LiRA},\tau}(\mathbf{x}, y) = \text{True if } \hat{\Lambda} \geq \tau, \text{ False otherwise}$, where $\tau$ is a threshold that maximizes TPR at a given FPR.

**Explanation-Based MIAs** Shokri et al. (2021) propose an explanation-based attack that directly thresholds on the explanation variance. Example $\mathbf{x}$ is predicted to be a member of the training set iff $\text{Var}(\varphi(\mathbf{x})) \leq \tau$, where $\tau$ is an optimal threshold we assume that the adversary has access to. We elaborate on the intuition behind using explanation variance in MIAs in Section 4.

**Differential Privacy** DP is a mathematically provable definition of privacy that provides a quantifiable metric of an algorithm's privacy loss, providing a computational method whose output is random enough to obscure any single participant's presence in the training data (Dwork & Roth, 2014). DP mechanisms have an $\varepsilon$ parameter that quantifies privacy strength (the lower the $\varepsilon$, the stronger the privacy strength) and a $\delta$ parameter that indicates the probability of privacy failure. In Appendix D, we explain the formal definition of DP and the optimized DP-stochastic gradient descent method by Bu et al. (2023) that we use in our experiments.

## 4. Our Membership Inference Attack Methods on Model Explanations

We present our new MIAs, drawn from Carlini et al.'s LiRA framework, that leverage the *variances, L1 norms, and L2 norms* of each example's feature attribution. We name these attacks VAR-LRT, L1-LRT, and L2-LRT, respectively. These black-box attacks assume that for every example, the adversary has access to the model's prediction on that example and a post-hoc explanation; unlike in loss-based attacks, adversarial access to true labels is not required.

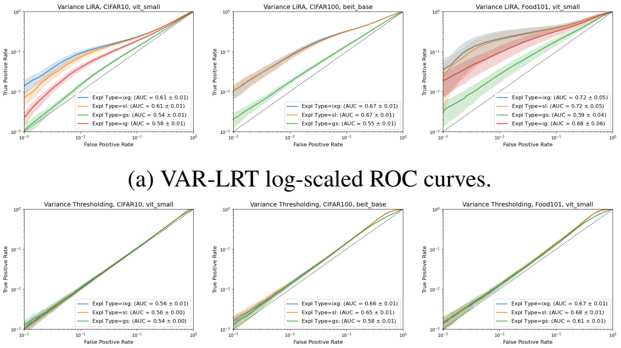
**Attack on Explanation Variances (VAR-LRT)** Shokri et al. (2021) thresholding attack on explanation variance follows the intuition that gradient descent pushes training set points further from the decision boundary, and non-training points are on average closer to the decision boundary. (This intuition is also leveraged in other adversarial ML work (Choquette-Choo et al., 2021; Yu et al., 2019).) The act of leveraging explanation variance is motivated by this idea—that for points closer to the decision boundary, changing a feature affects the prediction itself more strongly, which leads to higher explanation variance. If a point is farther from the decision boundary, that means the model is more certain about the point's prediction, and the model's behavior on the point is unlikely to change if we slightly perturb the point. Shokri's attack method directly thresholds on explanation variance in inferring training set membership of each example; **we use the attack's intuition but enhance the attack's design.** Our first attack, VAR-LRT, computes and thresholds on likelihood ratios of explanation variances. Algorithm 1 shows VAR-LRT in detail.

**Algorithm 1 VAR-LRT: LiRA on explanation variances.** The adversary trains shadow models on datasets with and without the target example, estimates parameters of the in- and out- distributions of sample variances of explanations (assuming Normal distributions of explanation variances), and runs a likelihood ratio test.

```
Require: model $f_\theta$, example $(\mathbf{x}, y) \in \mathbb{R}^d$, explanation vector $\varphi(f, (\mathbf{x}, y)) \in \mathbb{R}^d$, data distribution $\mathbb{D}$, number of shadow model iterations $N_S$
variances$_{\text{in}} = \{\}$, variances$_{\text{out}} = \{\}$
for $N_S$ times do
    $D_{\text{attack}} \xleftarrow{\$} \mathbb{D}$    ▷ sample a shadow dataset
    $f_{\text{in}} \leftarrow \mathcal{T}(D_{\text{attack}} \cup \{(\mathbf{x}, y)\})$    ▷ train IN model with $(\mathbf{x}, y)$ in training set
    $\varphi_{\text{in}} \leftarrow \varphi(f_{\text{in}}, (\mathbf{x}, y))$    ▷ generate post-hoc explanation of $f_{in}$'s behavior on $(\mathbf{x}, y)$
    $\bar{\varphi}_{\text{in}} \leftarrow \frac{1}{d}\sum_{i=1}^{d} \varphi_{\text{in},i}$
    variances$_{\text{in}} \leftarrow$ variances$_{\text{in}} \cup \{\frac{1}{d}\sum_{i=1}^{d}(\varphi_{\text{in},i} - \bar{\varphi}_{\text{in}})^2\}$    ▷ record sample variance of $\varphi_{in}$
    $f_{\text{out}} \leftarrow \mathcal{T}(D_{\text{attack}} \setminus \{(\mathbf{x}, y)\})$    ▷ train OUT model
    $\varphi_{\text{out}} \leftarrow \varphi(f_{\text{out}}, (\mathbf{x}, y))$    ▷ generate post-hoc explanation of $f_{out}$'s behavior on $(\mathbf{x}, y)$
    $\bar{\varphi}_{\text{out}} \leftarrow \frac{1}{d}\sum_{i=1}^{d} \varphi_{\text{out},i}$
    variances$_{\text{out}} \leftarrow$ variances$_{\text{out}} \cup \{\frac{1}{d}\sum_{i=1}^{d}(\varphi_{\text{out},i} - \bar{\varphi}_{\text{out}})^2\}$    ▷ record sample variance of $\varphi_{out}$
end for
$\hat{\mu}_{\text{in}} \leftarrow \texttt{mean}(\text{variances}_{\text{in}})$, $\hat{\mu}_{\text{out}} \leftarrow \texttt{mean}(\text{variances}_{\text{out}})$
$\hat{\sigma}^2_{\text{in}} \leftarrow \texttt{var}(\text{variances}_{\text{in}})$, $\hat{\sigma}^2_{\text{out}} \leftarrow \texttt{var}(\text{variances}_{\text{out}})$
$\varphi_{\text{obs}} \leftarrow \varphi(f_\theta, (\mathbf{x}, y))$, $\bar{\varphi}_{\text{obs}} \leftarrow \sum_{i=1}^{d} \varphi_{\text{obs},i}$
variance$_{\text{obs}} = \frac{1}{d}\sum_{i=1}^{d}(\varphi_{\text{obs},i} - \bar{\varphi}_{\text{obs}})^2$    ▷ query model
return $\hat{\Lambda} = \frac{p(\text{variance}_{\text{obs}} \mid \mathcal{N}(\hat{\mu}_{\text{in}}, \hat{\sigma}^2_{\text{in}}))}{p(\text{variance}_{\text{obs}} \mid \mathcal{N}(\hat{\mu}_{\text{out}}, \hat{\sigma}^2_{\text{out}}))}$
```

**Attacks on Explanation L1 and L2 Norms (L1-LRT/L2-LRT)** Nasr et al. (2019) previously highlighted disparities between gradient norm distributions of members and nonmembers, implying the efficacy of the *gradient norm* as an attack statistic. Recently, Wang et al. (2024) studied a white-box attack based on *gradient norms* on open-source large language models. *Explanation norms* are closely related to gradient norms, and we draw this connection—as well as explain our intuition behind constructing LiRAs based on explanation norms—in Appendix E. **As far as we know, there is no prior work leveraging norms of model explanations in membership inference attack.** Algorithm 3 in Appendix E shows our explanation L1 norm-based LiRA algorithm (L1-LRT), which does so. The L2-LRT attack is almost identical but is based instead on L2 norms.

**Baselines** Shokri et al.'s explanation-based attack, which we call the "thresholding attack," is the main *baseline attack* on which we improve. Moreover, in Section 5.4, we highlight that our attack methods perform competitively relative to the *loss-based LiRA* baseline. This implies that our black-box methods are comparable to strong attacks that allow adversarial access to true labels.



(a) VAR-LRT log-scaled ROC curves.

(b) Baseline thresholding attack log-scaled ROC curves.

*Figure 1.* VAR-LRT vs. baseline thresholding attack ROCs for the CIFAR-10 (left), CIFAR-100 (middle), and Food 101 (right) datasets. We present results for all explanation methods under each dataset's chosen model and hyperparameter setting.

## 5. Experimental Results

### 5.1. Setup

We give full details on experimental setups and implementation in Appendix F, but in short, we fine-tune and report attack results on the following five datasets: CIFAR-10, CIFAR-100, Street View House Numbers (SVHN), Food 101, and German Traffic Sign Recognition Benchmark (GTSRB). For each dataset, we "choose" a vision transformer model (out of 2-3 ImageNet pre-trained models analyzed per dataset) and hyperparameter setting to report in the main body, with additional and ablation experiment results in the appendices. Missing data in a few experimental setups is attributed to limitations in our compute resources. We release our code at https://github.com/catherinehuang82/explaining-model-protecting-data.

### 5.2. Evaluation of the VAR-LRT Attack

We first present results on VAR-LRT and do an apples-to-apples comparison of this attack with Shokri et al.'s thresholding attack. Figure 1 displays log-scaled ROC curves of the VAR-LRT versus baseline thresholding attacks for the CIFAR-10, CIFAR-100, and Food 101 datasets.

We observe from the log-scaled ROC curves in Figure 1 that across datasets and explanation methods, VAR-LRT performs significantly better than random guessing at low FPRs. This means it confidently captures a small, known subset of training data members. Across datasets and explanation methods, VAR-LRT is more successful than the baseline attack at this task. More thoroughly, we present numerical results comparing VAR-LRT with the thresholding attack for four datasets and all explanation methods in Table 1. We present each attack's performance on each metric and $\Delta$, the change between the two attacks' average performance. We encourage the reader to focus primarily on viewing the $\Delta$ columns but nonetheless provide complete metric values for reference. Almost all $\Delta$ values are green except for a few values for attacks on the GS explanation type, most of which correspond to the AUC metric—we explain in Appendix H.1 why these values do not undermine our conclusion throughout this section that VAR-LRT is a stronger attack than the thresholding attack.

### 5.3. Evaluation of the L1-LRT and L2-LRT Attacks

Figure 2 displays L1-LRT and L2-LRT attack ROCs for the CIFAR-10, CIFAR-100, and Food 101 datasets. L1-LRT and L2-LRT behave similarly to one another across the ROC curve, and both are highly successful, objectively and even relative to VAR-LRT. (We see that L1-LRT generally performs better than L2-LRT and hypothesize why in Appendix I.1.) To further highlight our most successful attack, L1-LRT, Table 2 shows numerical L1-LRT results for all five datasets; we see many bolded quantities highlighting where TPR at FPR $= x$ is at least $10 \cdot x$. We also observe that **across the whole table (every setting), the mean value of TPR at FPR $= x$ is higher than $x$.** This means that attacks perform reliably across the board, and a substantial number of attacks—especially on IXG and SL explanation types—perform exceedingly well at small FPR values.

**More Non-Private Attack Results and Ablation Experiments** More results on model and attack performance and ablation experiments are in Appendices G, H, I, and J.

### 5.4. Comparison with Loss-Based LiRA

Table 3 compares our attacks to the LiRA baseline based on per-example cross-entropy loss ("Loss LiRA"). L1-LRT and

*Table 1.* Comparing VAR-LRT vs. thresholding attack success. $\text{TPR}_x$ denotes the TPR at $\text{FPR} = x$ (i.e. $\text{FPR} = 100 \cdot x\%$). Green $\Delta$ values indicate metrics in which VAR-LRT has higher average value, and red $\Delta$ values indicate metrics in which VAR-LRT has lower average value.

| Exp Type | Metric | CIFAR-10 | | | SVHN | | |
|---|---|---|---|---|---|---|---|
| | | `Thres.` | `VAR-LRT` | $\Delta$ `(Avg.)` | `Thres.` | `VAR-LRT` | $\Delta$ `(Avg.)` |
| IXG | $\text{TPR}_{.001}$ | $0.0012 \pm 0.0006$ | $0.0252 \pm 0.0084$ | 0.0240 | $0.0012 \pm 0.0001$ | $0.0121 \pm 0.0103$ | 0.0471 |
| | $\text{TPR}_{.01}$ | $0.0112 \pm 0.0021$ | $0.1195 \pm 0.0152$ | 0.1083 | $0.0099 \pm 0.0190$ | $0.0570 \pm 0.0241$ | 0.0109 |
| | AUC | $0.5588 \pm 0.0055$ | $0.6133 \pm 0.0093$ | 0.0545 | $0.5448 \pm 0.0080$ | $0.5863 \pm 0.0182$ | 0.0415 |
| SL | $\text{TPR}_{.001}$ | $0.0012 \pm 0.0006$ | $0.0154 \pm 0.0062$ | 0.0142 | $0.0012 \pm 0.0008$ | $0.0139 \pm 0.0114$ | 0.0566 |
| | $\text{TPR}_{.01}$ | $0.0111 \pm 0.0023$ | $0.1032 \pm 0.0171$ | 0.0921 | $0.0103 \pm 0.0029$ | $0.0668 \pm 0.0258$ | 0.0126 |
| | AUC | $0.5593 \pm 0.0050$ | $0.6082 \pm 0.0106$ | 0.0489 | $0.5456 \pm 0.0079$ | $0.5889 \pm 0.0181$ | 0.0432 |
| IG | $\text{TPR}_{.001}$ | $0.0012 \pm 0.0007$ | $0.0038 \pm 0.0020$ | 0.0026 | $0.0013 \pm 0.0007$ | $0.0046 \pm 0.0027$ | 0.0152 |
| | $\text{TPR}_{.01}$ | $0.0110 \pm 0.0022$ | $0.0573 \pm 0.0103$ | 0.0464 | $0.0107 \pm 0.0023$ | $0.0260 \pm 0.0071$ | 0.0034 |
| | AUC | $0.5539 \pm 0.0068$ | $0.5872 \pm 0.0193$ | 0.0333 | $0.5233 \pm 0.0051$ | $0.5412 \pm 0.0091$ | 0.0180 |
| GS | $\text{TPR}_{.001}$ | $0.0016 \pm 0.0007$ | $0.0012 \pm 0.0007$ | 0.0003 | $0.0013 \pm 0.0008$ | $0.0024 \pm 0.0014$ | 0.0047 |
| | $\text{TPR}_{.01}$ | $0.0111 \pm 0.0019$ | $0.0181 \pm 0.0035$ | 0.0069 | $0.0103 \pm 0.0026$ | $0.0150 \pm 0.0036$ | 0.0011 |
| | AUC | $0.5404 \pm 0.0046$ | $0.5371 \pm 0.0056$ | $-0.0033$ | $0.5229 \pm 0.0054$ | $0.5206 \pm 0.0064$ | $-0.0023$ |

| Exp Type | Metric | CIFAR-100 | | | Food 101 | | |
|---|---|---|---|---|---|---|---|
| | | `Thres.` | `VAR-LRT` | $\Delta$ `(Avg.)` | `Thres.` | `VAR-LRT` | $\Delta$ `(Avg.)` |
| IXG | $\text{TPR}_{.001}$ | $0.0021 \pm 0.0010$ | $0.0200 \pm 0.0112$ | 0.0179 | $0.0012 \pm 0.0006$ | $0.0070 \pm 0.0018$ | 0.0057 |
| | $\text{TPR}_{.01}$ | $0.0158 \pm 0.0027$ | $0.1208 \pm 0.0271$ | 0.1050 | $0.0107 \pm 0.0021$ | $0.0225 \pm 0.0040$ | 0.0118 |
| | AUC | $0.6549 \pm 0.0100$ | $0.6708 \pm 0.0116$ | 0.0157 | $0.5106 \pm 0.0048$ | $0.5173 \pm 0.0050$ | 0.0067 |
| SL | $\text{TPR}_{.001}$ | $0.0018 \pm 0.0010$ | $0.0209 \pm 0.0109$ | 0.0191 | $0.0014 \pm 0.0007$ | $0.0021 \pm 0.0014$ | 0.0062 |
| | $\text{TPR}_{.01}$ | $0.0156 \pm 0.0029$ | $0.1176 \pm 0.0257$ | 0.1020 | $0.0106 \pm 0.0021$ | $0.0258 \pm 0.0041$ | 0.0152 |
| | AUC | $0.6522 \pm 0.0098$ | $0.6678 \pm 0.0112$ | 0.0156 | $0.5105 \pm 0.0043$ | $0.5170 \pm 0.0051$ | 0.0066 |
| IG | $\text{TPR}_{.001}$ | - | - | - | $0.0013 \pm 0.0009$ | $0.0028 \pm 0.0009$ | 0.0015 |
| | $\text{TPR}_{.01}$ | - | - | - | $0.0109 \pm 0.0026$ | $0.0147 \pm 0.0025$ | 0.0039 |
| | AUC | - | - | - | $0.5065 \pm 0.0048$ | $0.5074 \pm 0.0047$ | 0.0009 |
| GS | $\text{TPR}_{.001}$ | $0.0019 \pm 0.0010$ | $0.0027 \pm 0.0011$ | 0.0008 | $0.0012 \pm 0.0007$ | $0.0013 \pm 0.0007$ | 0.0002 |
| | $\text{TPR}_{.01}$ | $0.0152 \pm 0.0019$ | $0.0200 \pm 0.0036$ | 0.0049 | $0.0110 \pm 0.0022$ | $0.0116 \pm 0.0023$ | 0.0006 |
| | AUC | $0.5847 \pm 0.0065$ | $0.5572 \pm 0.0080$ | $-0.0275$ | $0.5057 \pm 0.0052$ | $0.5021 \pm 0.0033$ | $-0.0036$ |

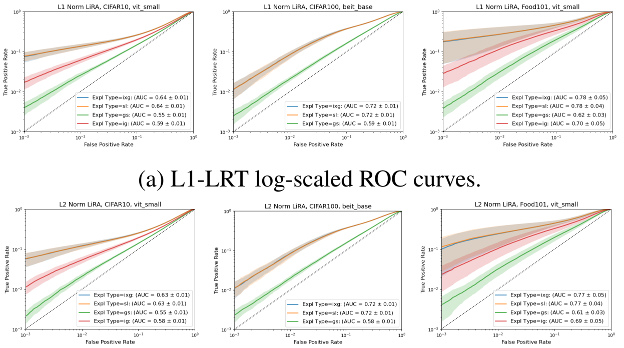
*Table 2.* Numerical attack results for L1-LRT. The bolded quantities show where TPR at FPR = $x$ is at least $10 \cdot x$.

| Exp Type | Metric | Dataset | | | | |
|---|---|---|---|---|---|---|
| | | `CIFAR-10` | `CIFAR-100` | `Food 101` | `SVHN` | `GTSRB` |
| IXG | $\text{TPR}_{.001}$ | $\mathbf{0.093} \pm 0.022$ | $\mathbf{0.022} \pm 0.013$ | $\mathbf{0.203} \pm 0.140$ | $\mathbf{0.015} \pm 0.014$ | $0.009 \pm 0.002$ |
| | $\text{TPR}_{.01}$ | $\mathbf{0.156} \pm 0.018$ | $\mathbf{0.130} \pm 0.037$ | $\mathbf{0.310} \pm 0.131$ | $0.065 \pm 0.029$ | $0.027 \pm 0.004$ |
| | AUC | $0.639 \pm 0.008$ | $0.716 \pm 0.012$ | $0.780 \pm 0.046$ | $0.603 \pm 0.018$ | $0.518 \pm 0.005$ |
| SL | $\text{TPR}_{.001}$ | $\mathbf{0.093} \pm 0.022$ | $\mathbf{0.021} \pm 0.011$ | $\mathbf{0.210} \pm 0.143$ | $\mathbf{0.017} \pm 0.015$ | $\mathbf{0.012} \pm 0.002$ |
| | $\text{TPR}_{.01}$ | $\mathbf{0.155} \pm 0.019$ | $\mathbf{0.128} \pm 0.035$ | $\mathbf{0.309} \pm 0.132$ | $0.077 \pm 0.030$ | $0.030 \pm 0.005$ |
| | AUC | $0.639 \pm 0.009$ | $0.716 \pm 0.011$ | $0.782 \pm 0.043$ | $0.605 \pm 0.018$ | $0.518 \pm 0.005$ |
| IG | $\text{TPR}_{.001}$ | $\mathbf{0.026} \pm 0.008$ | – | $\mathbf{0.044} \pm 0.024$ | $0.006 \pm 0.003$ | $0.004 \pm 0.001$ |
| | $\text{TPR}_{.01}$ | $0.080 \pm 0.012$ | – | $\mathbf{0.159} \pm 0.032$ | $0.008 \pm 0.007$ | $0.017 \pm 0.003$ |
| | AUC | $0.590 \pm 0.009$ | – | $0.700 \pm 0.051$ | $0.554 \pm 0.009$ | $0.508 \pm 0.004$ |
| GS | $\text{TPR}_{.001}$ | $0.006 \pm 0.002$ | $0.003 \pm 0.001$ | $0.006 \pm 0.003$ | $0.003 \pm 0.001$ | $0.002 \pm 0.001$ |
| | $\text{TPR}_{.01}$ | $0.033 \pm 0.005$ | $0.027 \pm 0.005$ | $0.044 \pm 0.016$ | $0.017 \pm 0.004$ | $0.012 \pm 0.003$ |
| | AUC | $0.554 \pm 0.006$ | $0.586 \pm 0.009$ | $0.616 \pm 0.032$ | $0.532 \pm 0.007$ | $0.502 \pm 0.004$ |

L2-LRT dominate Loss LiRA across all reported metrics, and VAR-LRT is also competitive. Given that Loss LiRA is a strong attack to begin with among attacks that allow adversarial access to true labels (as explained in Section 3 and shown in Carlini et al. (2021)), this result implies further privacy risk: our methods perform competitively even with black-box limitations on what the adversary has access to. This is a brief proof of concept—further work is helpful for benchmarking our attacks against attacks based on other, possibly more "traditional" signals, such as loss.

*Table 3.* Attack metrics compared with the loss LiRA attack on CIFAR-10, IXG explanations, `vit_small` model. Bold quantities indicate TPR @ low FPR metrics where our methods outperform loss LiRA.

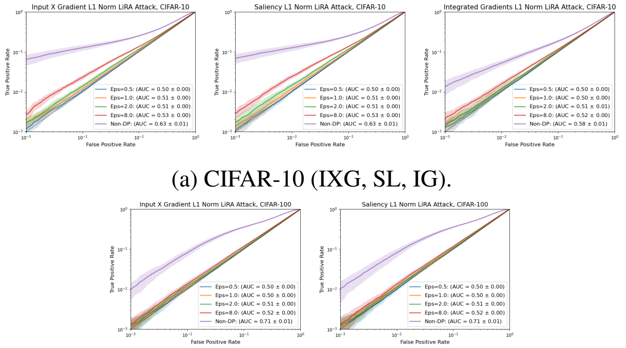
| Attack | $\text{TPR}_{.001}$ | $\text{TPR}_{.001}$ | AUC |
|---|---|---|---|
| Loss LiRA | $0.054 \pm 0.007$ | $0.095 \pm 0.012$ | $0.570 \pm 0.009$ |
| L1-LRT | $\mathbf{0.093} \pm 0.022$ | $\mathbf{0.156} \pm 0.018$ | $0.639 \pm 0.008$ |
| L2-LRT | $\mathbf{0.078} \pm 0.021$ | $\mathbf{0.146} \pm 0.016$ | $0.633 \pm 0.007$ |
| VAR-LRT | $0.025 \pm 0.008$ | $\mathbf{0.120} \pm 0.015$ | $0.613 \pm 0.009$ |



(a) L1-LRT log-scaled ROC curves.

(b) L2-LRT log-scaled ROC curves.

*Figure 2.* L1-LRT and L2-LRT attack results for the CIFAR-10 (left), CIFAR-100 (middle), and Food 101 (right) datasets.



(a) CIFAR-10 (IXG, SL, IG).

(b) CIFAR-100 (IXG, SL).

*Figure 3.* L1-LRT attack success of non-private (purple curves) vs. DP fine-tuned models (other curves). We show one plot per explanation method: IXG (left), SL (middle), and (with the exception of CIFAR-100) IG (right). Each subplot shows curves for $\epsilon = 0.5, 1.0, 2.0, 8.0$.

### 5.5. Mitigating Attack Success with Differential Privacy

We investigate whether Bu et al. (2023)'s accuracy-optimizing differentially private fine-tuning can successfully mitigate the success of these explanation-based attacks while preserving model accuracy. We approach this part by treating attack results on non-privately fine-tuned models as *baseline results* upon which we wish to improve, now aiming for *lower* attack success metrics.

Figure 3 shows ROC curves highlighting the impact of DP fine-tuning on our most powerful attack, L1-LRT. We observe that in each subplot—that is, across datasets and explanation types—the ROC curves corresponding to models fine-tuned with DP hug the "random guessing" diagonal line much more closely than the baseline ROC curves do. With CIFAR-10, the $\varepsilon = 8.0$ setting (and, for IXG and SL explanations, the $\varepsilon = 2.0$ setting) gives slight attack success. However, the other curves associated with DP fine-tuning in Figure 3 show attacks that behave no better than random guessing, both on average and in low FPR.

DP fine-tuned models give rise to explanation-based attacks that certainly do not demonstrate privacy risk at the level that the non-private attacks do. Thus, while we cannot guarantee that explanations that come from DP models will always deliver completely unsuccessful membership inference attacks, we still strongly conclude that DP fine-tuning substantially diminishes explanation-based MIA success to the point of doing only minimally better than "random guessing." Appendix L shows that this conclusion holds across the other attack types, and Appendix K shows, importantly, that DP mitigates attack success *while maintaining model accuracy*.

## 6. Discussion

The lack of trust in post-hoc explanations contributes to the slow adoption of machine learning in high-stakes domains. This paper reveals unforeseen vulnerabilities of feature attribution explanations to membership inference by introducing two new attacks that respectively leverage sample variances and norms of attribution vectors. These attacks are significantly more successful than existing attacks that leverage explanations, particularly at confidently identifying specific training set members.

We also show that when foundation models are fine-tuned with differential privacy, post-hoc explanations yield significantly lower explanation-based attack success compared to when models are fine-tuned without DP—all while preserving model accuracy. Our work patches a gap in literature—there is no prior empirical work in deep learning that thoroughly quantifies the relationship between DP training and the subsequent privacy risks of a broad class of post-hoc model explanations.

Within adversarial ML, there remain open research directions concerning how post-hoc model explanations may be leveraged to compromise data privacy. For example, can explanations be leveraged *alongside* traditional signals (e.g. loss, confidence) to compromise privacy to a greater extent? Can we get similar attack success without requiring adversarial training of shadow models? Can we formally quantify attack success (i.e. TPR at certain FPR)? How successful might other non-MIA attack types that leverage explanations be?

Keeping trade-offs in mind, we emphasize that there remains a barrier between the idea of differential privacy as a promising privacy risk mitigation approach, and its tangible deployment in contexts that necessitate model explanations. The latter entails adequate analysis on if privacy sacrifices on the *quality* or *usefulness* of explanations—this is out of the scope of this work but remains necessary to be studied.

## References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, oct 2016. doi: 10.1145/2976749.2978318. URL https://doi.org/10.1145%2F2976749.2978318.

Abascal, J., Wu, S., Oprea, A., and Ullman, J. Tmi! fine-tuned models leak private information from their pretraining data, 2023.

Azad, B., Azad, R., Eskandari, S., Bozorgpour, A., Kazerouni, A., Rekik, I., and Merhof, D. Foundational models in medical imaging: A comprehensive survey and future vision, 2023.

Beaujour, J.-M. Derivation of the gradient of the cross-entropy loss, Dec 2017. URL https://jmlb.github.io/ml/2017/12/26/Calculate_Gradient_Softmax/.

Bossard, L., Guillaumin, M., and Van Gool, L. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014.

Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Chen, X., Choromanski, K., Ding, T., Driess, D., Dubey, A., Finn, C., Florence, P., Fu, C., Arenas, M. G., Gopalakrishnan, K., Han, K., Hausman, K., Herzog, A., Hsu, J., Ichter, B., Irpan, A., Joshi, N., Julian, R., Kalashnikov, D., Kuang, Y., Leal, I., Lee, L., Lee, T.-W. E., Levine, S., Lu, Y., Michalewski, H., Mordatch, I., Pertsch, K., Rao, K., Reymann, K., Ryoo, M., Salazar, G., Sanketi, P., Sermanet, P., Singh, J., Singh, A., Soricut, R., Tran, H., Vanhoucke, V., Vuong, Q., Wahid, A., Welker, S., Wohlhart, P., Wu, J., Xia, F., Xiao, T., Xu, P., Xu, S., Yu, T., and Zitkovich, B. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023.

Bu, Z., Wang, Y.-X., Zha, S., and Karypis, G. Automatic clipping: Differentially private deep learning made easier and stronger, 2023.

Carlini, N., Chien, S., Nasr, M., Song, S., Terzis, A., and Tramèr, F. Membership inference attacks from first principles. *CoRR*, abs/2112.03570, 2021. URL https://arxiv.org/abs/2112.03570.

Choquette-Choo, C. A., Tramer, F., Carlini, N., and Papernot, N. Label-only membership inference attacks, 2021.

Clevert, D.-A., Unterthiner, T., and Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus), 2016.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. URL https://arxiv.org/abs/2010.11929.

Dwork, C. and Roth, A. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, aug 2014. ISSN 1551-305X. doi: 10.1561/0400000042. URL https://doi.org/10.1561/0400000042.

Grezes, F., Blanco-Cuaresma, S., Accomazzi, A., Kurtz, M. J., Shapurian, G., Henneken, E., Grant, C. S., Thompson, D. M., Chyla, R., McDonald, S., Hostetler, T. W., Templeton, M. R., Lockhart, K. E., Martinovic, N., Chen, S., Tanner, C., and Protopapas, P. Building astrobert, a language model for astronomy  astrophysics, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition, 2015.

Hendrycks, D. and Gimpel, K. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415, 2016. URL http://arxiv.org/abs/1606.08415.

Huang, C., Swoopes, C., Xiao, C., Ma, J., and Lakkaraju, H. Accurate, explainable, and private models: Providing recourse while minimizing training data leakage, 2023.

Jayaraman, B., Wang, L., Knipmeyer, K., Gu, Q., and Evans, D. Revisiting membership inference under realistic assumptions, 2021. URL https://arxiv.org/abs/2005.10881.

Kawaharazuka, K., Matsushima, T., Gambardella, A., Guo, J., Paxton, C., and Zeng, A. Real-world robot applications of foundation models: A review, 2024.

Ke, A., Ellsworth, W., Banerjee, O., Ng, A. Y., and Rajpurkar, P. Chextransfer: performance and parameter efficiency of imagenet models for chest x-ray interpretation. In *Proceedings of the Conference on Health, Inference, and Learning*, ACM CHIL '21. ACM, April 2021. doi: 10.1145/3450439.3451867. URL http://dx.doi.org/10.1145/3450439.3451867.

Khodabakhsh, H., LeCun, Y., Cortes, C., and Burges, C. J. Mnist dataset, 2019. URL https://www.kaggle.com/datasets/hojjatk/mnist-dataset.

Kokhlikyan, N., Miglani, V., Martin, M., Wang, E., Alsallakh, B., Reynolds, J., Melnikov, A., Kliushkina, N., Araya, C., Yan, S., and Reblitz-Richardson, O. Captum: A unified and generic model interpretability library for pytorch, 2020.

Krizhevsky, A. Learning multiple layers of features from tiny images. 2009. URL https://api.semanticscholar.org/CorpusID:18268744.

Liu, H., Wu, Y., Yu, Z., and Zhang, N. Please tell me more: Privacy impact of explainability through the lens of membership inference attack. In *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 123–123, Los Alamitos, CA, USA, may 2024. IEEE Computer Society. doi: 10.1109/SP54263.2024.00120. URL https://doi.ieeecomputersociety.org/10.1109/SP54263.2024.00120.

Long, Y., Wang, L., Bu, D., Bindschaedler, V., Wang, X., Tang, H., Gunter, C. A., and Chen, K. A pragmatic approach to membership inferences on machine learning models. In *2020 IEEE European Symposium on Security and Privacy (EuroSP)*, pp. 521–534, 2020. doi: 10.1109/EuroSP48549.2020.00040.

Lundberg, S. and Lee, S.-I. A unified approach to interpreting model predictions, 2017.

Marcel, S. and Rodriguez, Y. Torchvision the machine-vision package of torch. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, pp. 1485–1488, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781605589336. doi: 10.1145/1873951.1874254. URL https://doi.org/10.1145/1873951.1874254.

Moor, M., Moor, M., Banerjee, O., Abad, Z. S. H., Krumholz, H. M., Leskovec, J., Topol, E. J., and Rajpurkar, P. Foundation models for generalist medical artificial intelligence. *Nature*, 616(7956):259—265, April 2023. ISSN 0028-0836. doi: 10.1038/s41586-023-05881-4. URL https://www.nature.com/articles/s41586-023-05881-4.pdf.

Nasr, M., Shokri, R., and Houmansadr, A. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, May 2019. doi: 10.1109/sp.2019.00065. URL http://dx.doi.org/10.1109/SP.2019.00065.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. URL http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

Nguyen, T. D., Ting, Y.-S., Ciucă, I., O'Neill, C., Sun, Z.-C., Jabłońska, M., Kruk, S., Perkowski, E., Miller, J., Li, J., Peek, J., Iyer, K., Różański, T., Khetarpal, P., Zaman, S., Brodrick, D., Méndez, S. J. R., Bui, T., Goodman, A., Accomazzi, A., Naiman, J., Cranney, J., Schawinski, K., and UniverseTBD. Astrollama: Towards specialized foundation models in astronomy, 2023.

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., Bello, I., Berdine, J., Bernadett-Shapiro, G., Berner, C., Bogdonoff, L., Boiko, O., Boyd, M., Brakman, A.-L., Brockman, G., Brooks, T., Brundage, M., Button, K., Cai, T., Campbell, R., Cann, A., Carey, B., Carlson, C., Carmichael, R., Chan, B., Chang, C., Chantzis, F., Chen, D., Chen, S., Chen, R., Chen, J., Chen, M., Chess, B., Cho, C., Chu, C., Chung, H. W., Cummings, D., Currier, J., Dai, Y., Decareaux, C., Degry, T., Deutsch, N., Deville, D., Dhar, A., Dohan, D., Dowling, S., Dunning, S., Ecoffet, A., Eleti, A., Eloundou, T., Farhi, D., Fedus, L., Felix, N., Fishman, S. P., Forte, J., Fulford, I., Gao, L., Georges, E., Gibson, C., Goel, V., Gogineni, T., Goh, G., Gontijo-Lopes, R., Gordon, J., Grafstein, M., Gray, S., Greene, R., Gross, J., Gu, S. S., Guo, Y., Hallacy, C., Han, J., Harris, J., He, Y., Heaton, M., Heidecke, J., Hesse, C., Hickey, A., Hickey, W., Hoeschele, P., Houghton, B., Hsu, K., Hu, S., Hu, X., Huizinga, J., Jain, S., Jain, S., Jang, J., Jiang, A., Jiang, R., Jin, H., Jin, D., Jomoto, S., Jonn, B., Jun, H., Kaftan, T., Łukasz Kaiser, Kamali, A., Kanitscheider, I., Keskar, N. S., Khan, T., Kilpatrick, L., Kim, J. W., Kim, C., Kim, Y., Kirchner, J. H., Kiros, J., Knight, M., Kokotajlo, D., Łukasz Kondraciuk, Kondrich, A., Konstantinidis, A., Kosic, K., Krueger, G., Kuo, V., Lampe, M., Lan, I., Lee, T., Leike, J., Leung, J., Levy, D., Li, C. M., Lim, R., Lin, M., Lin, S., Litwin, M., Lopez, T., Lowe, R., Lue, P., Makanju, A., Malfacini, K., Manning, S., Markov, T., Markovski, Y., Martin, B., Mayer, K., Mayne, A., McGrew, B., McKinney, S. M., McLeavey, C., McMillan, P., McNeil, J., Medina, D., Mehta, A., Menick, J., Metz, L., Mishchenko, A., Mishkin, P., Monaco, V., Morikawa, E., Mossing, D., Mu, T., Murati, M., Murk, O., Mély, D., Nair, A., Nakano, R., Nayak, R., Neelakantan, A., Ngo, R., Noh, H., Ouyang, L., O'Keefe, C., Pachocki, J., Paino, A., Palermo, J., Pantuliano, A., Parascandolo, G., Parish, J., Parparita, E., Passos, A., Pavlov, M., Peng, A., Perelman, A., de Avila Belbute Peres, F., Petrov, M., de Oliveira Pinto, H. P., Michael, Pokorny, Pokrass, M., Pong, V. H., Powell, T., Power, A., Power, B., Proehl, E., Puri, R., Radford, A., Rae, J., Ramesh, A., Raymond, C., Real, F., Rimbach, K., Ross, C., Rotsted, B., Roussez, H., Ryder, N., Saltarelli, M., Sanders, T., Santurkar, S., Sastry, G., Schmidt, H., Schnurr, D., Schulman, J., Selsam, D., Sheppard, K., Sherbakov, T., Shieh, J., Shoker, S., Shyam, P., Sidor, S., Sigler, E., Simens, M., Sitkin, J., Slama, K., Sohl, I., Sokolowsky, B., Song, Y., Staudacher, N., Such, F. P., Summers, N., Sutskever, I., Tang, J., Tezak, N., Thompson, M. B., Tillet, P., Tootoonchian, A., Tseng, E., Tuggle, P., Turley, N., Tworek, J., Uribe, J. F. C., Vallone, A., Vijayvergiya, A., Voss, C., Wainwright, C., Wang, J. J., Wang, A., Wang, B., Ward, J., Wei, J., Weinmann, C., Welihinda, A., Welinder, P., Weng, J., Weng, L., Wiethoff, M., Willner, D., Winter, C., Wolrich, S., Wong, H., Workman, L., Wu, S., Wu, J., Wu, M., Xiao, K., Xu, T., Yoo, S., Yu, K., Yuan, Q., Zaremba, W., Zellers, R., Zhang, C., Zhang, M., Zhao, S., Zheng, T., Zhuang, J., Zhuk, W., and Zoph, B. Gpt-4 technical report, 2024.

Pawelczyk, M., Lakkaraju, H., and Neel, S. On the privacy risks of algorithmic recourse. *arXiv*, abs/2211.05427, 2022. URL https://arxiv.org/abs/2211.05427.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision, 2021.

Ribeiro, M. T., Singh, S., and Guestrin, C. ”why should I trust you?”: Explaining the predictions of any classifier. *CoRR*, abs/1602.04938, 2016. URL http://arxiv.org/abs/1602.04938.

Sablayrolles, A., Douze, M., Ollivier, Y., Schmid, C., and Jégou, H. White-box vs black-box: Bayes optimal strategies for membership inference, 2019. URL https://arxiv.org/abs/1908.11229.

Shapley, L. S. *A Value for N-Person Games*. RAND Corporation, Santa Monica, CA, 1952. doi: 10.7249/P0295.

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18, 2017. doi: 10.1109/SP.2017.41.

Shokri, R., Strobel, M., and Zick, Y. On the privacy risks of model explanations. *arXiv*, abs/1907.00164, 2021. URL https://arxiv.org/abs/1907.00164.

Shrikumar, A., Greenside, P., Shcherbina, A., and Kundaje, A. Not just a black box: Learning important features through propagating activation differences, 2017.

Song, L. and Mittal, P. Systematic evaluation of privacy risks of machine learning models. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2615–2632. USENIX Association, August 2021. ISBN 978-1-939133-24-3. URL https://www.usenix.org/conference/usenixsecurity21/presentation/song.

Sowrirajan, H., Yang, J., Ng, A. Y., and Rajpurkar, P. Mococxr: Moco pretraining improves representation and transferability of chest x-ray models, 2021.

Stallkamp, J., Schlipsing, M., Salmen, J., and Igel, C. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, (0): –, 2012. ISSN 0893-6080. doi: 10.1016/j.neunet.2012.02.016. URL http://www.sciencedirect.com/science/article/pii/S0893608012000457.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3319–3328. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/sundararajan17a.html.

Tian, L., Greer, H., Kwitt, R., Vialard, F.-X., Estepar, R. S. J., Bouix, S., Rushmore, R., and Niethammer, M. unigradicon: A foundation model for medical image registration, 2024.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL http://arxiv.org/abs/1706.03762.

Wang, J. G., Wang, J., Li, M., and Neel, S. Pandora's white-box: Increased training data leakage in open llms, 2024.

Watson, L., Guo, C., Cormode, G., and Sablayrolles, A. On the importance of difficulty calibration in membership inference attacks, 2022. URL https://arxiv.org/abs/2111.08440.

Weller, A. Transparency: Motivations and challenges, 2019.

Wightman, R. Pytorch image models. https://github.com/huggingface/pytorch-image-models, 2019.

Wu, C., Zhang, X., Zhang, Y., Wang, Y., and Xie, W. Towards generalist foundation model for radiology by leveraging web-scale 2d3d medical data, 2023.

Ye, J., Maddi, A., Murakonda, S. K., Bindschaedler, V., and Shokri, R. Enhanced membership inference attacks against machine learning models, 2022. URL https://arxiv.org/abs/2111.09679.

Yeom, S., Giacomelli, I., Fredrikson, M., and Jha, S. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pp. 268–282, 2018. doi: 10.1109/CSF.2018.00027.

Yu, T., Hu, S., Guo, C., Chao, W.-L., and Weinberger, K. Q. A new defense against adversarial images: Turning a weakness into a strength, 2019.

# A. Appendix

Our appendices are organized into the following parts:

- B: The Case for Foundation Models and Vision Transformers
- C: Post-Hoc Feature Attribution Explanations
- D: More on Differential Privacy, DP-SGD, and DP with Automatic Gradient Norm Clipping
- E: L1-LRT/L2-LRT Intuition and Algorithm
- F: Experimental Setups and Implementation Details
- G: Performance of Non-Private Models
- H: More Non-Private VAR-LRT Results
- I: More Non-Private L1-LRT/L2-LRT Results
- J: Non-Private Ablation Experiments
- K: Performance of Models Fine-Tuned with Differential Privacy
- L: More Results on the Impact of Differentially Private Fine-Tuning on Attack Success

## B. The Case for Foundation Models and Vision Transformers

Fine-tuning happens when a pre-trained foundation model is then trained on a smaller, more specific new task (Dosovitskiy et al., 2020). Foundation models in general are desirable for a variety of reasons with respect to our research questions:

- Foundation models are generally trained on data with public access, which means the models do not touch sensitive data (until possibly during downstream tasks). Hence, pre-trained foundation models adhere to the right to privacy.
- Foundation models perform remarkably well with complex problems on complex datasets. They are less prone to overfitting, having been pre-trained on broad data and able to effectively handle a wide range of inputs. Foundation models' architectures have large size, depth, and scale that allow for their state-of-the-art quality.
- Foundation models such as large language models (e.g. GPT (OpenAI et al., 2024)) and vision-language models (e.g. CLIP (Radford et al., 2021)) are widely applied to complex real-world settings. In terms of vision tasks, for example, foundation models are used for applications ranging from medical imaging (Azad et al., 2023; Moor et al., 2023; Sowrirajan et al., 2021; Ke et al., 2021; Tian et al., 2024; Wu et al., 2023) to astronomy (Grezes et al., 2021; Nguyen et al., 2023) to robotics (Kawaharazuka et al., 2024; Brohan et al., 2023). By virtue of their versatility and widespread use, foundation models are thus a viable choice of model type in settings involving sensitive personal data.
- Fine-tuning foundation models on downstream tasks requires substantially fewer computational resources than training the model from scratch. Fine-tuning generally requires fewer epochs than standard training. To conduct membership inference in this work, we fine-tune tens of shadow models per attack experiment, and doing so is computationally more feasible than training these numerous large models from scratch.

For these reasons, in particular for the first reason on privacy defense, we choose to evaluate our methods using large pre-trained foundation models on fine-tuned tasks.

### B.1. The Vision Transformer Architecture

In this work, we evaluate our methods on image classification tasks. We choose to focus on image classification tasks rather than text classification, since post-hoc explanations are conceptually better defined for images: each pixel is a feature, and post-hoc explanations reveal which pixels in an image are most influential to a model's prediction of that image's class. Text corpora are typically higher-dimensional and less standardized than images (which can readily be scaled to a fixed, standardized dimension), meaning that explanations on text-datasets are especially sparse. Furthermore, flagship papers on post-hoc explainability methods (Lundberg & Lee, 2017; Sundararajan et al., 2017; Shrikumar et al., 2017) typically evaluate their explanation methods on image data. Such image classification tasks are commonly trained with convolutional neutral network (CNN) foundation models; for example, He et al. (2015) famously introduced the state-of-the-art residual network ("ResNet") CNN architecture.

Separately, in the natural language processing domain, the Transformer architecture was proposed by Vaswani et al. Vaswani et al. (2017) for machine translation. Since then, Transformer-based architectures have become state-of-the-art in many NLP tasks. Transformers rely on a *self-attention* mechanism that is scalable, efficient, and captures both short-term and long-term dependencies among text sequences. Compared with previously prevalent CNN and recurrent neural network approaches to NLP tasks, Transformers have both higher performance and higher speed. Transformers are commonly pre-trained on large text corpora and then fine-tuned on smaller, more specific tasks, making them a suitable foundation model.

Applying Transformers to image classification tasks naively would require that each pixel attend to each other pixel; this is intractable. Dosovitskiy et al. (2020) propose the state-of-the-art solution in the *vision transformer (ViT)* architecture: the ViT reshapes each original input image $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$ into a sequence of flattened 2D patches $\mathbf{x}_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$, where $(H \times W)$ are the dimensions of the original image, $C$ is the number of (color) channels, $(P \times P)$ are the dimensions of each image patch, and $N = \frac{HW}{P^2}$ is the number of patches. After each image is split into fixed-size patches, ViT linearly embeds each image and adds positional embeddings to incorporate positional information (of the patches within each image). The embedded vectors are then fed into ViT's Transformer encoder, which is built with alternating layers of multiheaded self-attention units and multilayer perception (MLP) units. Each MLP block contains two layers with a Gaussian Error Linear Unit (GELU) activation function (Dosovitskiy et al., 2020). The GELU function is a high-performing neural network activation function that often yields a performance improvement upon the more vanilla ReLU activation function (Hendrycks

& Gimpel, 2016). For $Z \sim \mathcal{N}(0, 1)$ a Standard Normal random variable, GELU is defined as

$$\mathrm{GELU}(x) = xP(Z \leq x).$$

Figure 4 visualizes GELU compared with other common neural network activation functions. Figure 5 provides an overview of the vision transformer architecture.

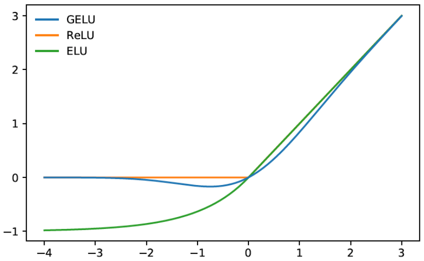


*Figure 4.* The GELU, ReLU, and ELU (Exponential Linear Unit) (Clevert et al., 2016) activation functions. The vision transformer architecture uses GELU activations.

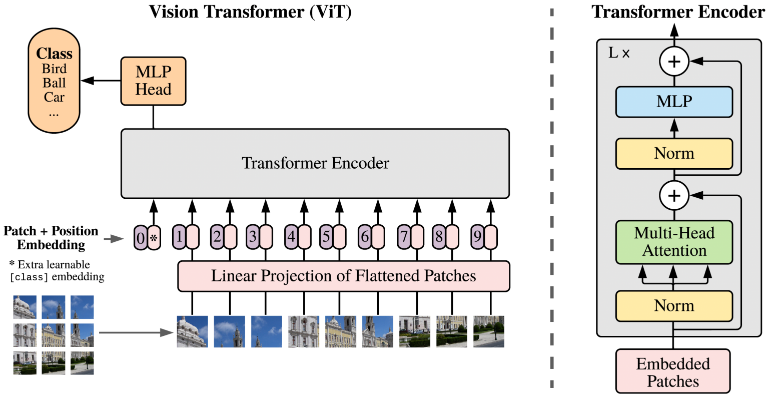


*Figure 5.* Overview of the vision transformer (ViT) model architecture. ViT splits an image into patches, embeds them (linearly and positionally), and feeds the embeddings into a Transformer encoder.

In this work, we use vision transformers in our experiments because not only are vision transformers state-of-the-art foundation models, but they perform better on fine-tuned downstream datasets than ResNet.

## C. Post-Hoc Feature Attribution Explanations

We study the following backpropagation-based feature attribution methods: Input $*$ Gradient (IXG), Saliency Maps (SL), Integrated Gradients (IG), and (a gradient-based approximation to) SHAP (GS). We describe each of these methods, as well as any desirable properties they exhibit (according to the pioneers of these methods).

### C.1. Input $*$ Gradient (IXG)

We first introduce the Input * Gradient technique (Shrikumar et al., 2017). This attribution vector is relatively simple to generate, computed by taking the partial derivatives of the output with respect to each input feature and multiplying them with the input itself:

$$\varphi_i(\mathbf{x}) = x_i \cdot \frac{\partial f_\theta(\mathbf{x})}{\partial x_i}.$$

### C.2. Saliency Maps (`SL`)

Saliency maps are almost equivalent to taking vanilla gradients. The only difference is that this method computes the *absolute value* of the gradient with respect to each input feature.

The interpretation of absolute value is that features with the highest absolute gradient need to be perturbed the least in order for the model's predicted output to change the most. The ensuing limitation of saliency maps is that they do not differentiate between features that contribute positively to prediction and features that contribute negatively. However, since our explanation-based attack methods are based on scalar summaries of attribution vectors (e.g. variance and norms), this limitation is not a direct concern.

### C.3. Integrated Gradients (`IG`)

Integrated gradients (Sundararajan et al., 2017) similarly computes the partial derivatives of the output with respect to each input feature. However, instead of only computing the gradient on the original input, IG computes the *average* gradient as the input varies along a linear path from a baseline $\mathbf{x}_{BL}$ to $\mathbf{x}$ (usually, $\mathbf{x}_{BL} = \mathbf{0}$). The mathematical definition of IG is

$$\varphi_{IG}(\mathbf{x})_i = (x_i - x_{BL,i}) \cdot \int_{\alpha=0}^{1} \frac{\partial f_\theta(\mathbf{x}^\alpha)}{\partial x_i^\alpha} d\alpha \bigg|_{\mathbf{x}^\alpha = \mathbf{x} + \alpha(\mathbf{x} - \mathbf{x}_{BL})}.$$

Through an axiomatic approach, IG is designed to satisfy three desirable properties of attribution methods: sensitivity, implementation invariance, and completeness. Implementation-wise, we can only approximate the integral by taking a Riemann sum over a discrete number of gradients along the linear path from baseline to input.

**Sensitivity** Sensitivity means that given a point $\mathbf{x} \in \mathcal{X}$ such that $x_i \neq x_{BL,i}$ and $f_\theta(\mathbf{x}) \neq f_\theta(\mathbf{x}_{BL})$, then $\varphi_i(\mathbf{x}) \neq 0$. In words, sensitivity asserts that for every input and baseline that differ in one feature but have different predictions, then the explanation method should give that feature a non-zero attribution.

**Completeness** Completeness means that $\sum_{i=1}^{n} \varphi_i(\mathbf{x}) = f_\theta(\mathbf{x}) - f_\theta(\mathbf{x}_{BL})$: the attributions sum up to the difference between the output of $f_\theta$ at the input $\mathbf{x}$ and the baseline $\mathbf{x}_{BL}$.

**Implementation invariance** Two models $f_1$ and $f_2$ are functionally equivalent if $f_1(\mathbf{x}) = f_2(\mathbf{x})$ for all inputs $\mathbf{x}$. The implementation invariance axiom asserts that explanations should be identical for functionally equivalent models.

### C.4. SHapley Additive exPlanations (SHAP; Abbreviated as GS)

In the original SHAP paper, Lundberg & Lee (2017) assume an additive *explanation model* $g$: $g$ is an interpretable approximation of the original model $f_\theta$ that is a linear combination of binary variables. The authors show that only one possible additive explanation model $g$ satisfies the three axiomatic properties of local accuracy, missingness, and monotonicity (details of which are beyond the scope of this work). Further, the corresponding feature attribution values $\phi_i$ of model $g$ coincide with Shapley values (1952) in cooperative game theory. The SHAP explanation framework is based on the Shapley values of a conditional expectation function derived from $f_\theta$, the details of which are beyond the scope of this work. At a high level, SHAP values set $\varphi_i$ to the change in the expected model prediction when conditioning on feature $i$.

SHAP values are difficult to compute exactly, and in this work, we use a *gradient-based approximation* to SHAP values that approximates the expectation of gradients * (inputs - baselines) (Kokhlikyan et al., 2020). The approximation works as follows: we add Gaussian random noise to each input sample multiple times, select random points along the path between the input $\mathbf{x}$ and a baseline $\mathbf{x}_{BL}$, and compute the gradient of outputs with respect to these points on the path. We use this approximation and refer to it as "Gradient SHAP" with abbreviation "GS."

## D. More on Differential Privacy, DP-SGD, and DP with Automatic Gradient Norm Clipping

Formally, a randomized mechanism $\mathcal{M}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ satisfies $(\varepsilon, \delta)$-**Differential Privacy** ($(\varepsilon, \delta)$-DP) if for any two adjacent input datasets $D, D' \sim \mathbb{D}$ differing by one row, and any subset of outputs $S \subseteq \mathcal{R}$, $Pr[\mathcal{M}(D) \in S] \leq e^\varepsilon \cdot Pr[\mathcal{M}(D') \in S] + \delta$. The lower the $\varepsilon$ (the privacy parameter), the stronger the privacy protection. A value of $\varepsilon = \infty$ represents that a mechanism lacks privacy.

**DP Stochastic Gradient Descent (DP-SGD)** Abadi et al. (2016) propose a differentially private stochastic gradient

**Algorithm 2** Differentially Private SGD (Outline)

```
Require: Examples $\{\mathbf{x_1}, \dots, \mathbf{x_N}\} \in \mathbb{R}^d$, labels $\{y_1, \dots, y_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(f_\theta(\mathbf{x_i}), y_i)$. Parameters: learning rate $\eta_t$, noise scale $\sigma$, group size $L$, gradient norm bound $C$.
  Initialize $\theta_0$ randomly
  for $t \in [T]$ do
    Take a random sample $L_t$ of examples with sampling probability $q = L/N$
    Compute gradient
    For each $i \in L_t$, compute $\mathbf{g}_t(\mathbf{x_i}) \leftarrow \nabla_{\theta_t}\mathcal{L}(f_{\theta_t}(\mathbf{x_i}), y_i)$
    Clip gradient
    $\bar{\mathbf{g}}_t(\mathbf{x_i}) \leftarrow \mathbf{g}_t(\mathbf{x_i}) / \max\left(1, \frac{\|\mathbf{g}_t(\mathbf{x_i})\|_2}{C}\right)$
    Add noise
    $\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L}\left(\sum_i \bar{\mathbf{g}}_t(\mathbf{x_i}) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$
    Descent
    $\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$
  end for
  Output $\theta_T$ and compute the overall privacy cost $(\varepsilon, \delta)$ using a privacy accounting method.
```

descent (DP-SGD) mechanism to train models with DP. It works by adding Gaussian distributed random noise (with standard deviation $\sigma$) to gradients. Higher $\sigma$ (i.e. adding random noise with higher variance) leads to lower $\varepsilon$.

Algorithm 2 outlines Abadi et al.'s method for training a model with parameters $\theta$. Before adding noise to gradients, DP-SGD upper-bounds the norm of the gradient to be within $C$ via gradient norm clipping; this is to enforce that the sensitivity of the gradients be exactly $C$ (instead of being unbounded). DP-SGD then involves adding $\mathcal{N}(\mathbf{0}, \sigma^2 C^2 \mathbf{I})$ distributed random noise, where $\sigma$ is a pre-specified noise scale. Abadi et al. use the *moments accountant* privacy loss budgeting technique and the Gaussian Mechanism to prove that for appropriately chosen $\sigma$, DP-SGD is $(O(q\varepsilon\sqrt{T}), \delta)$-DP, where $q$ is the sampling ratio per batch.

**Differential Privacy Under Post-Processing** If $M$ is $\varepsilon$-DP, and $G$ is an arbitrary deterministic mapping, then $G \circ M$ is also $\varepsilon$-DP. In other words, once a quantity is "made private" through DP, it cannot be subsequently "made un-private" (Dwork & Roth, 2014).

It is by post-processing of DP that an $(\varepsilon, \delta)$-DP fine-tuned model gives rise to $(\varepsilon, \delta)$-DP post-hoc explanations on the fine-tuning dataset, since explanations of the form $\varphi(\mathcal{X}, f_\theta, \mathbf{x}, \cdot)$ are functions of the model $f_\theta$. In this section, we present the existing DP optimiziations we use that improve upon the utility and computational efficiency of Abadi et al.'s DP-SGD algorithm.

**DP Utility and Efficiency Optimizations: Automatic Gradient Norm Clipping** Algorithm 2, which shows the DP-SGD algorithm, contains a gradient L2 norm clipping step in order to bound the global sensitivity of each batch gradient to a fixed, pre-determined parameter $C$: $\bar{\mathbf{g}}_t(\mathbf{x_i}) \leftarrow \mathbf{g}_t(\mathbf{x_i}) \cdot \min\left(1, \frac{C}{\|\mathbf{g}_t(\mathbf{x_i})\|_2}\right)$. There are two main downsides of this gradient norm clipping method:

1. We must tune $C$ (treated as a hyperparameter). This adds overhead to computational cost, especially since differentially private deep learning already also involves tuning parameters such as $\varepsilon$, $\delta$, batch size, sampling rate, and learning rate.
2. The vanilla clipping method makes gradients lose important information. Consider the per-sample gradient equivalent of Abadi et al.'s batch gradient clipping: $\bar{\mathbf{g}}(\mathbf{x_i}) \leftarrow \mathbf{g}(\mathbf{x_i}) \cdot \min\left(1, \frac{C}{\|\mathbf{g}(\mathbf{x_i})\|_2}\right)$. All per-sample gradients that have norm above $C$ are "punished equally" and end up with the same magnitude: $||\mathbf{g}(\mathbf{x_i}) \cdot \min\left(1, \frac{C}{\|\mathbf{g}(\mathbf{x_i})\|_2}\right)|| = C, \forall \mathbf{x_i}$.

In this work, we instead employ Bu et al. (2023) *automatic* gradient norm clipping algorithm to fine-tune models to improve model accuracy. The automatic per-sample gradient clipping works as follows:

$$\bar{\mathbf{g}}_t(\mathbf{x_i}) \leftarrow \mathbf{g}_t(\mathbf{x_i}) \cdot \frac{1}{\|\mathbf{g}_t(\mathbf{x_i})\|_2 + \gamma}, \tag{1}$$

where parameter $\gamma > 0$ is known as the "stability constant." This approach remedies the two aforementioned downsides of vanilla gradient norm clipping. This approach allows gradients with larger norms to still have larger norms after clipping, whereas clipping will diminish the magnitudes of gradients with smaller norms. Furthermore, this clipping method frees us of the parameter $C$. Bu et al. prove that clipping with multiplicative factor $\frac{C}{\|\mathbf{g}_t(\mathbf{x_i})\|_2 + \gamma}$ is functionally equivalent to choosing $C = 1$ (as in Equation 1), in the sense that in both cases, the gradient norm converges to zero at the same asymptotic rate.

We use Bu et al.'s Automatic Clipping to fine-tune vision transformers with DP.

## E. L1-LRT/L2-LRT Intuition and Algorithm

Before we discuss intuition on constructing likelihood ratio test statistics based on explanation norms, we first discuss intuition behind using gradient norms, since gradients are closely related to explanations.

**Intuition: Attacks on Gradient Norms** Broadly speaking, a model $f_\theta$ is trained to approximately minimize the loss that $f_\theta$ incurs on training examples. The gradient of the model loss with respect to model parameters reflects the magnitude and direction of the "step" that gradient descent takes during model training. The following intuition assumes a convex loss function. As the training process approaches a local minimum of the "loss landscape" (i.e. the structure of the loss function in the parameter space that the model traverses step-wise during training), the model takes smaller and smaller steps in each subsequent iteration of the gradient descent process, until it reaches convergence. A trained model is not as "well-fit" to non-members of the training set as it is to members. Hence, the model is more likely to take steeper and bigger gradient descent steps on unseen test set examples than on train set examples. The *L1 norm* of the gradient directly encodes the steepness of the descent step taken after a model "sees" an example. This intuition suggests that the gradient norms of training set members are on average smaller than gradient norms of non-members. Although the cross-entropy loss function we use is not convex in complex neural network settings, this intuition still motivates us to experiment with the gradient norm attack method.

**Intuition: From Gradient Norms to Explanation Norms** The backpropagation-based post-hoc explanation methods that we work with involve computing gradients of $f_\theta$'s model output with respect to input features. These gradients are not exactly the same as the gradients computed during training, which are gradients of the loss function with respect to model parameters. However, we can still leverage the aforementioned intuition, since model parameter values directly reflect—albeit in a non-linear manner—how input features contribute to model predictions. Furthermore, there is separate intuition on the type of gradient computed in post-hoc explanations: this gradient represents the extent to which $f_\theta$'s *prediction* changes if we were to perturb the input features. Since the training process pushes training set members further away from the decision boundary compared to non-members (behavior we previously explained in Section 4), it follows that perturbing the input features of an arbitrary *training* data point would scarcely change the model's behavior on or prediction for that point. Conceptually, this corresponds to a smaller gradient magnitude on training points—and magnitudes are equivalent to *L2 norms*.

**The Attack Algorithm** Algorithm 3 shows our explanation L1 norm-based LiRA method (L1-LRT). The L2-LRT attack is almost identical but instead based instead on L2 norms.

## F. Experimental Setups and Implementation Details

### F.1. Datasets

In this section, we discuss experimental setups and implementation details.

Across models and datasets, we sub-sample a smaller dataset of size 20000 for fine-tuning each shadow model and computing post-hoc explanations. In the membership inference attack literature, sub-sampling is commonplace. In our predecessor work, Shokri et al. (2021) employ sub-sampling in many of their experiments on explanation-based membership inference attacks, using sub-sample sizes of 5000, 10000, and 20000, among others. We generally use a $50\%/50\%$ train-test split across all attack and shadow model training procedures, since our attack success evaluation metrics are most straightforward to interpret when there is a balanced amount of training and test data given to the adversary; this is also the approach taken by Shokri et al.

We present results for models fine-tuned on the following datasets designed for image classification. Each dataset consists of color images in 3 color channels (red, green, and blue).

**Algorithm 3 L1-LRT: Likelihood ratio attack on the L1 norm of post-hoc explanations.** The adversary trains shadow models on datasets with and without the target example, generates post-hoc explanations on each example in their dataset, estimates parameters of the in- and out- distributions of sample variances of post-hoc explanations, and runs a likelihood ratio test.

```
Require: model $f_\theta$, example $(\mathbf{x}, y) \in \mathbb{R}^d$, explanation vector $\varphi(f, (\mathbf{x}, y)) \in \mathbb{R}^d$, data distribution $\mathbb{D}$,
    number of shadow model iterations $N_S$
norms$_{\text{in}}$ $= \{\}$
norms$_{\text{out}}$ $= \{\}$
for $N_S$ times do
    $D_{\text{attack}} \xleftarrow{\$} \mathbb{D}$                                          ▷ sample a shadow dataset
    $f_{\text{in}} \leftarrow \mathcal{T}(D_{\text{attack}} \cup \{(\mathbf{x}, y)\})$                  ▷ train IN model with $(\mathbf{x}, y)$ in training set
    $\varphi_{\text{in}} \leftarrow \varphi(f_{\text{in}}, (\mathbf{x}, y))$                 ▷ generate post-hoc explanation of $f_{in}$'s behavior on $(\mathbf{x}, y)$
    norms$_{\text{in}}$ $\leftarrow$ norms$_{\text{in}}$ $\cup \{\sum_{i=1}^{d} |\varphi_{\text{in},i}|\}$                              ▷ record L1 norm of $\varphi_{in}$
    $f_{\text{out}} \leftarrow \mathcal{T}(D_{\text{attack}} \backslash \{(\mathbf{x}, y)\})$              ▷ train OUT model with $(\mathbf{x}, y)$ not in training set
    $\varphi_{\text{out}} \leftarrow \varphi(f_{\text{out}}, (\mathbf{x}, y))$              ▷ generate post-hoc explanation of $f_{out}$'s behavior on $(\mathbf{x}, y)$
    norms$_{\text{out}}$ $\leftarrow$ norms$_{\text{out}}$ $\cup \{\sum_{i=1}^{d} |\varphi_{\text{out},i}|\}$                          ▷ record L1 norm of $\varphi_{out}$
end for
$\hat{\mu}_{\text{in}} \leftarrow \texttt{mean}(\text{norms}_{\text{in}})$
$\hat{\mu}_{\text{out}} \leftarrow \texttt{mean}(\text{norms}_{\text{out}})$
$\hat{\sigma}^2_{\text{in}} \leftarrow \texttt{var}(\text{norms}_{\text{in}})$
$\hat{\sigma}^2_{\text{out}} \leftarrow \texttt{var}(\text{norms}_{\text{out}})$
$\varphi_{\text{obs}} \leftarrow \varphi(f_\theta, (\mathbf{x}, y))$
norm$_{\text{obs}}$ $= \sum_{i=1}^{d} |\varphi_{\text{obs},i}|$                                          ▷ query target model
return $\hat{\Lambda} = \dfrac{p(\text{norm}_{\text{obs}} \mid \mathcal{N}(\hat{\mu}_{\text{in}}, \hat{\sigma}^2_{\text{in}}))}{p(\text{norm}_{\text{obs}} \mid \mathcal{N}(\hat{\mu}_{\text{out}}, \hat{\sigma}^2_{\text{out}}))}$
```

**CIFAR-10 and CIFAR-100** CIFAR-10 and CIFAR-100 (Krizhevsky, 2009) are well-known and widely used benchmark datasets for image classification. They consist of 10 and 100 classes, respectively, with 6000 and 600 images per class, respectively. The datasets are by default split into 50000 training images and 10000 test images, but for purposes of our membership inference attacks, we use a 50%/50% train-test split.

**Food 101** Food 101 (Bossard et al., 2014) is a dataset of 101 food categories with 101,000 images in total. For *each class*, there are 750 training and 250 test images. According to Boassard et al., "on purpose, the training images were not cleaned, and thus still contain some amount of noise. This comes mostly in the form of intense colors and sometimes wrong labels."

**Street View House Numbers (SVHN)** The SVHN dataset (Netzer et al., 2011) contains satellite images of house numbers in Google Street View. It is similar to MNIST (Khodabakhsh et al., 2019) in that images are of small cropped digits and that there are 10 classes, but it is a larger dataset (73257 train and 26032 test images) and contains color images (whereas MNIST images are black-and-white). SVHN's increased complexity (compared to MNIST) makes it an appropriate downstream task for pre-trained foundation models.

**German Traffic Sign Recognition Benchmark (GTSRB)** The GTSRB dataset (Stallkamp et al., 2012) features 43 classes of traffic signs split into 39209 training images and 12630 test images.

Each of our datasets is housed in Torchvision's datasets module (Marcel & Rodriguez, 2010) (see CIFAR-10, CIFAR-100, Food 101, SVHN, GTSRB).

## F.2. Model Architectures and Training

We import and fine-tune pre-trained models from `timm` (standing for PyTorch Image Models) (Wightman, 2019), a deep learning library that provides state-of-the-art computer vision models and helper utilities to work with them.

For each dataset, we experiment across the following model architectures. Each model has a patch size of 16, an input image dimension of 224, and is pre-trained on some ordered sample (possibly with replacement) of ImageNet-22k, ImageNet-21k, and ImageNet-1k.

- CIFAR-10: `timm`'s `vit_small_patch16_224` (30.1 M parameters), `vit_relpos_small_patch16_224.sw_in1k` (22.0 M parameters, with relative position embeddings), and `vit_relpos_base_patch16_224.sw_in1k` (86.4 M parameters, with relative position embeddings). In the main body, we report all attacks on the `vit_small_patch16_224` model.

- CIFAR-100: `timm`'s `beit_base_patch16_224.in22k_ft_in22k_in1k` (86.5 M parameters) and `beitv2_base_patch16_224.in1k_ft_in22k_in1k` (86.5 M parameters). For the main body, we report all attacks on `beit_base_patch16_224.in22k_ft_in22k_in1k`.

- Food 101, SVHN, and GTSRB: `vit_small_patch16_224` and `vit_relpos_small_patch16_224.sw_in1k`. For the main body, we report all attacks on `vit_small_patch16_224`.

### F.3. Data Pre-Processing

We employ the following pre-processing methods for each image in each dataset:

1. We resize each input image to have dimension $3 \times 224 \times 224$, where the first dimension corresponds to the three color channels (Red, Green, Blue). The per-color channel dimension is 224 because that is the input dimension expected from the model architectures we use.

2. We apply the transformation `torchvision.transforms.Normalize((0.5, 0.5, 0.5),(0.5, 0.5, 0.5))`. The first `(0.5, 0.5, 0.5)` corresponds to post-normalization mean of pixel values for each of the three (RGB) color channels, and the second `(0.5, 0.5, 0.5)` corresponds to post-normalization standard deviation. This operation centers the input image tensors around zero and scales them to a range of approximately $-1$ to $1$.

### F.4. Training Hyperparameters

Table 4 describes the chosen hyperparameter settings for each dataset, based on a combination of test accuracy (for model usefulness) and MIA attack success. "Mini-batch size" describes the number of samples in each mini-batch during training; the model is trained on each mini-batch separately. "Batch size" determines the sampling rate used in gradient descent. Sampling rate = (batch size) / (length of training data), and this quantity describes the proportion of the training data used for each parameter update step. This sampling rate is relevant to DP-SGD, where Gaussian noise is added to the gradients computed from only a subset of the training data at each update step.

*Table 4.* Training hyperparameters for each dataset.

| Dataset | Batch Size | Mini-Batch Size | Learning Rate | Epochs |
|---|---|---|---|---|
| CIFAR-10 | 1000 | 50 | 0.005 | 30 |
| CIFAR-100 | 1000 | 50 | 0.005 | 9 |
| Food 101 | 512 | 50 | 0.005 | 50 |
| SVHN | 512 | 50 | 0.005 | 50 |
| GTSRB | 512 | 50 | 0.005 | 50 |

### F.5. Post-Hoc Explanation Parameters

We use Captum, a model interpretability and understanding library for PyTorch (Kokhlikyan et al., 2020), to compute explanations in the form of attribution vectors. Captum supports all of the backpropagation-based methods we study (IXG, SL, IG, and GS), among others. Throughout this work, if an ROC curve or table omits results of a few particular settings of dataset and explanation type, that means it takes our computing resources too long to generate explanations of that type of 20000 data examples.

In Captum, each feature attribution method accepts a list of parameters. Each method requires as input the `target` parameter, which specifies the output indices for which we want gradients to be computed. Captum's documentation (Kokhlikyan et al., 2020) states that "for classification cases, this is usually the target class." We retain this default (`target` = predicted class), with the intuition that the explanations should capture the features important to the model's predictions on the predicted class, not on any other class.

Integrated gradients (IG) has a `baseline` parameter (see Section C.3). We set this $\mathbf{x}_{BL}$ quantity to the all-zero tensor, which is the default value in the Captum library. IG also has an `n_steps` parameter, which describes the number of approximation steps used in integration. Captum sets the default `n_steps` value to 50, but to speed up computation, we set `n_steps` $= 25$.

Gradient SHAP (GS) has a `baseline` parameter as well (see Section C.4), which we set to a tensor where each component is distributed $\mathcal{N}(0, 0.001^2)$. GS also has an `n_samples` parameter used for the following, according to Captum's documentation: "[GS] adds white noise to each input sample `n_samples` times, selects a random baseline from baselines' distribution and a random point along the path between the baseline and the input, and computes the gradient of outputs with respect to those selected random points." Captum sets `n_samples` to 5 by default, and we retain this setting.

### F.6. Likelihood Ratio Attack Implementation

For each attack setting, we train $N + 1$ total models, where $N$ is the total number of shadow models of each attack. We perform $N + 1$ runs of each attack, each time treating a different model as the target model and treating the remaining $N$ models as shadow models. Each of the $N + 1$ models is trained on a randomly selected 10000 points out of the subsampled dataset of size 20000, and the remaining 10000 points are used for testing. For each of the 20000 examples, we record whether that example is in the training set or the test set of each model and save that information as a vector of 0's and 1's. The *training set membership information* of all $N + 1$ models is saved in a matrix of dimension $20000 \times (N + 1)$. For each example, we also record the variance, L1 norm, and L2 norm "scores" of each model's post-hoc explanation of that example. The *explanation scores* of all examples on all $N + 1$ models are saved in three matrices, each of dimension $20000 \times (N + 1)$. Using the saved explanation scores (variances and L1/L2 norms) and training set membership statuses of each example, we run likelihood ratio attacks.

For the attacks on models fine-tuned *without* privacy, use $N = 32$, meaning that we train 32 shadow models per attack and perform 33 total runs of each attack setting. For the attacks on models fine-tuned *with* privacy, we use $N = 16$, meaning that we train 16 shadow models per attack and perform 17 total runs of each attack setting. We set $N$ to be lower here to respect fair and timely use of computational resources, considering the high amount of time it takes to generate post-hoc explanations and the multiple privacy settings (i.e. values of $\varepsilon$) we experiment on.

### F.7. Differential Privacy Parameters and Fine-Tuning

For all attacks on explanations coming from models fine-tuned with DP, we report results across the following values of privacy parameter $\varepsilon$: 0.5, 1.0, 2.0, 8.0, $\infty$. For conciseness, we report results involving DP on only the `vit_small_patch16_224` and `beit_base_patch16_224.in22k_ft_in22k_in1k` models and the CIFAR-10 and CIFAR-100 datasets. In addition, we found the GS explanation method and the GTSRB dataset to yield low baseline (non-private) attack success; in order to provide more interpretable and distinctive results, we thus exclude these settings from results shown in this work. With privacy, ROC curves and metric values are averaged over 17 evaluation runs rather than 33, and each attack uses 16 shadow models rather than 32. This holds too for results on non-private model-based attacks in this section, in order to do an apples-to-apples comparison of non-private and private fine-tuning. As a result, in results involving DP, the reported baseline (non-private model) ROC curves and metric values differ by an extremely small amount from those reported in results not involving DP. Nonetheless, the results preserve the success of baseline attacks as well as the relative success of baseline attacks across datasets and explanation types.

## G. Performance of Non-Private Models

In Table 5, we present train and test accuracies for non-privately fine-tuned models on all datasets. We observe that different models fine-tuned on the same datasets have similar test accuracies on these datasets.

## H. More Non-Private VAR-LRT Results

### H.1. Comparing VAR-LRT with Thresholding Attack: More Table 1 Analysis

With respect to Table 1, we observe that a majority of the red $\Delta$ values correspond to the AUC metric. This is the metric with which we are least concerned, for the following reasons:

*Table 5.* Model performance. We report average train and test accuracies for all non-privately finetuned models on all datasets. The "chosen" epoch counts are shown as bolded rows. The results are averaged over 33 evaluation runs and include ±1 standard deviation.

| CIFAR-10 | Train Accuracy (%) | Test Accuracy (%) |
|---|---|---|
| vit_small_patch26_224 | $100.000 \pm 0.000$ | $96.064 \pm 0.613$ |
| vit_relpos_small_patch16_224.sw_in1k | $99.938 \pm 0.0797$ | $95.404 \pm 0.691$ |
| vit_relpos_base_patch16_224.sw_in1k | $99.788 \pm 0.202$ | $95.508 \pm 0.831$ |
| CIFAR-100 | Train Accuracy (%) | Test Accuracy (%) |
| beit_base_patch16_224.in22k_ft_in22k_in1k | $98.722 \pm 0.321$ | $80.109 \pm 0.590$ |
| beitv2_base_patch16_224.in12k_ft_in22k_in1k | $94.928 \pm 0.748$ | $81.902 \pm 0.621$ |
| Food 101 | Train Accuracy (%) | Test Accuracy (%) |
| vit_small_patch26_224 | $99.761 \pm 0.325$ | $83.685 \pm 3.878$ |
| vit_relpos_small_patch16_224.sw_in1k | $99.630 \pm 0.429$ | $81.204 \pm 3.139$ |
| SVHN | Train Accuracy (%) | Test Accuracy (%) |
| vit_small_patch26_224 | $99.552 \pm 0.223$ | $91.558 \pm 1.056$ |
| vit_relpos_small_patch16_224.sw_in1k | $99.580 \pm 0.215$ | $91.823 \pm 1.177$ |
| GTSRB | Train Accuracy (%) | Test Accuracy (%) |
| vit_small_patch26_224 | $100.000 \pm 0.000$ | $99.899 \pm 0.037$ |
| vit_relpos_small_patch16_224.sw_in1k | $100.000 \pm 0.000$ | $99.912 \pm 0.030$ |

1. We established in Section 3 that AUC is an average-case metric that we inherently care about less than we care about TPR at low FPR.

2. All of the red $\Delta_{\texttt{AUC}}$ values correspond to Gradient SHAP (GS) attacks. With GS attacks, we observe that both attack types on GS generally have low success: in the GS rows, we see average $\texttt{TPR}_{.001}$ values closest to $0.001$ and average $\texttt{TPR}_{.01}$ values closest to $0.01$ (compared to attacks on other explanation types). The conclusion that both attacks are minimally successful is more salient than any conclusion made comparing the two attacks.

For these reasons, we are not concerned about negative $\Delta_{\texttt{AUC}}$ values, so we use a darker shade of red to color them. The singular other entry with a brighter shade of red indicates that the CIFAR-10 VAR-LRT attack based on Gradient SHAP (GS) performs weaker than the corresponding thresholding attack in the $\texttt{TPR}_{.001}$ metric. However, we claim that this entry *also* does not undermine the relative success of VAR-LRT: the bright red $\Delta_{\texttt{TPR}_{.001}}$ value shows a difference that is not statistically significant. The $t$-statistic of the two-sided hypothesis test comparing the mean $\Delta_{\texttt{TPR}_{.001}}$ values of the two attacks is $-1.946$, and the corresponding $p$-value is $0.0561$, which is considered not low, assuming a $0.05$-level test.

Almost all of the green-colored $\Delta$ values show statistically significant differences. In particular, the green $\Delta$ values corresponding to the CIFAR-10, SVHN, and CIFAR-100 datasets and the IXG, SL, and IG explanation methods are all significant. (The test statistics and $p$-values of the two-sample $t$-tests verifying this conclusion are in Table 6.) These are the very datasets and explanation methods we care about most, since both attack methods are more successful than random guessing in these settings.

Hence, the red values in Table 1 do not undermine the conclusion that VAR-LRT is a stronger attack than the thresholding attack, particularly at confidently identifying specific members of the training dataset. Further, this result holds across datasets and explanation methods.

### H.2. Results on More Models and Datasets

Figure 6 shows VAR-LRT ROCs for CIFAR-10, CIFAR-100, and Food 101 under the additional model architectures *not* shown in the main body. We observe that the VAR-LRT attack remains successful on this new set of vision transformer models—particularly at low FPR.

- CIFAR-10: `vit_relpos_small_patch16_224.sw_in1k, vit_relpos_base_patch16_224.sw_in1k`

- CIFAR-100: `beit_base_patch16_224.in22k_ft_in22k_in1k`

- Food 101: `vit_relpos_small_patch16_224.sw_in1k`

*Table 6.* Two-sample $t$-test results comparing VAR-LRT and thresholding attack performance. We report $t$ statistics and $p$-values for two-sample, two-sided $t$-tests of each metric under the CIFAR-10, CIFAR-100, Food 101, and SVHN datasets and the IXG, SL, and IG explanation types. In each setting, the mean difference in metric value between the VAR-LRT and thresholding attacks is positive, meaning VAR-LRT exhibits a statistically significant performance improvement compared to the thresholding attack. The sample size is 33 for each hypothesis test.

| Explanation Type | Metric | CIFAR-10 | | SVHN | | CIFAR-100 | | Food 101 | |
|---|---|---|---|---|---|---|---|---|---|
| | | $t$-statistic | $p$-value | $t$-statistic | $p$-value | $t$-statistic | $p$-value | $t$-statistic | $p$-value |
| IXG | $\text{TPR}_{.001}$ | 16.291 | 1.876E−24 | 6.092 | 7.050E−08 | 9.166 | 2.909E−13 | 7.300 | 5.533E−10 |
| | $\text{TPR}_{.01}$ | 40.594 | 2.146E−47 | 11.183 | 1.072E−16 | 22.176 | 9.645E−32 | 11.734 | 1.334E−17 |
| | AUC | 29.013 | 1.534E−38 | 12.007 | 4.815E−18 | 5.934 | 1.312E−07 | 5.082 | 3.486E−6 |
| SL | $\text{TPR}_{.001}$ | 13.216 | 5.914E−20 | 6.341 | 2.621E−08 | 9.977 | 1.156E−14 | 7.185 | 8.787E−10 |
| | $\text{TPR}_{.01}$ | 30.706 | 5.176E−40 | 12.507 | 7.639E−19 | 22.674 | 2.725E−32 | 11.345 | 5.796E−17 |
| | AUC | 24.003 | 1.039E−33 | 12.561 | 6.257E−19 | 6.029 | 9.032E−08 | 4.900 | 6.857E−6 |
| IG | $\text{TPR}_{.001}$ | 7.233 | 7.248E−10 | 7.049 | 1.526E−09 | - | - | 9.156 | 3.027E−13 |
| | $\text{TPR}_{.01}$ | 25.241 | 5.625E−35 | 11.745 | 1.279E−17 | - | - | 12.105 | 3.346E−18 |
| | AUC | 16.740 | 4.566E−25 | 9.931 | 1.382E−14 | - | - | 4.861 | 7.910E−6 |

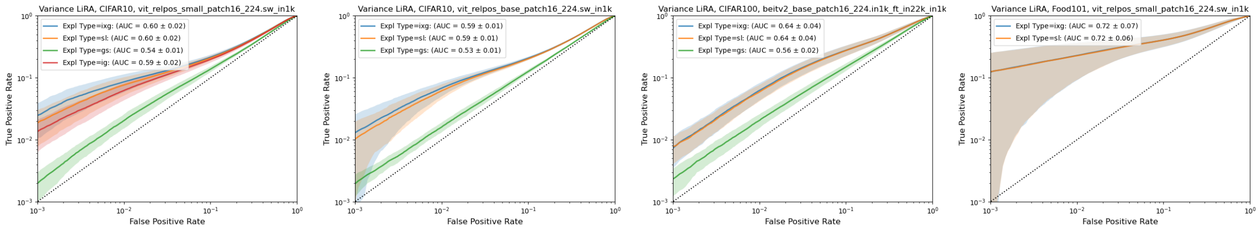


*Figure 6.* VAR-LRT log-scaled ROC curves for the CIFAR-10 (first and second from the left), CIFAR-100 (second from the right), and Food 101 (right) datasets, on different model architectures than are presented in the main body.

In the main text, we presented VAR-LRT attack ROCs for the CIFAR-10, CIFAR-100, and Food 101 datasets but excluded plots on the SVHN and GTSRB datasets. Figure 7 shows these excluded plots, using the `vit_small_patch16_224` architecture. VAR-LRT also performs better than random guessing on these datasets, particularly at low FPR.

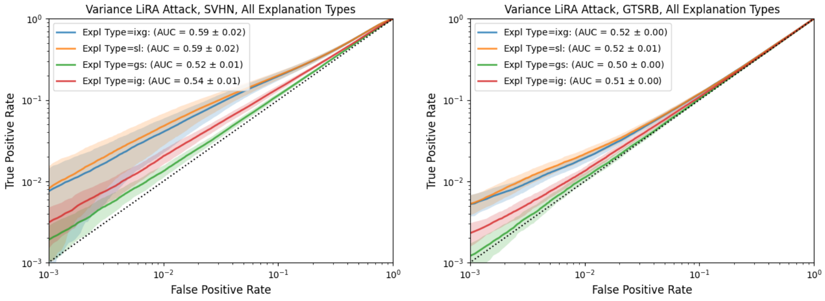


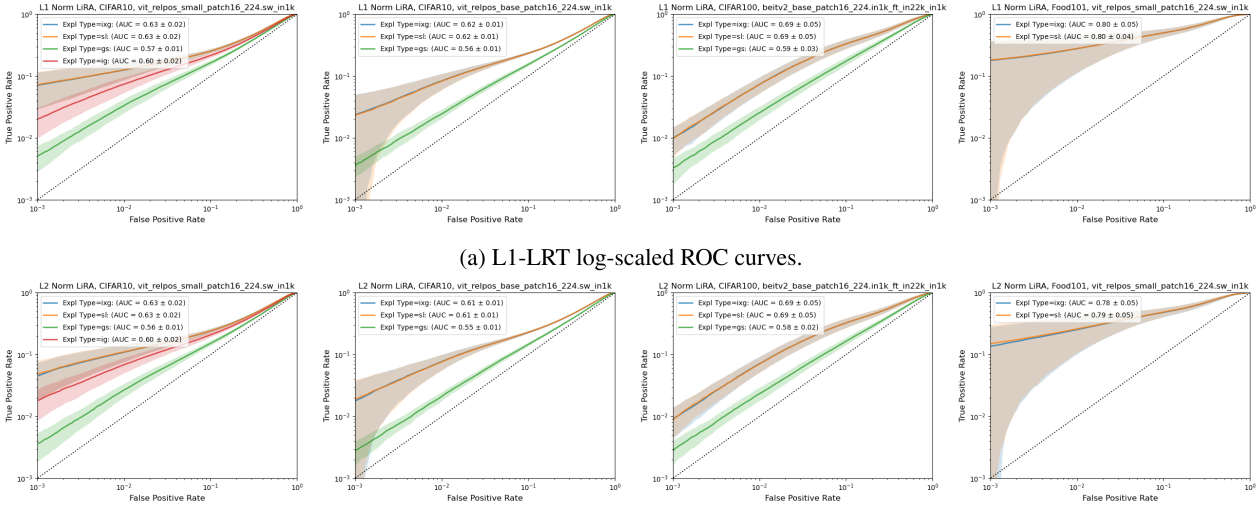
*Figure 7.* VAR-LRT ROCs for the SVHN (left) and GTSRB (right) datasets, `vit_small_patch16_224` model.

## I. More Non-Private L1-LRT/L2-LRT Results

In Figure 8, we present L1-LRT (top) and L2-LRT (bottom) results on the following datasets and architectures that were not featured in the main body:

- CIFAR-10: `vit_relpos_small_patch16_224.sw_in1k, vit_relpos_base_patch16_224.sw_in1k`
- CIFAR-100: `beit_base_patch16_224.in22k_ft_in22k_in1k`
- Food 101: `vit_relpos_small_patch16_224.sw_in1k`

We observe that the L1-LRT and L2-LRT attacks remain successful on this new set of vision transformer models.

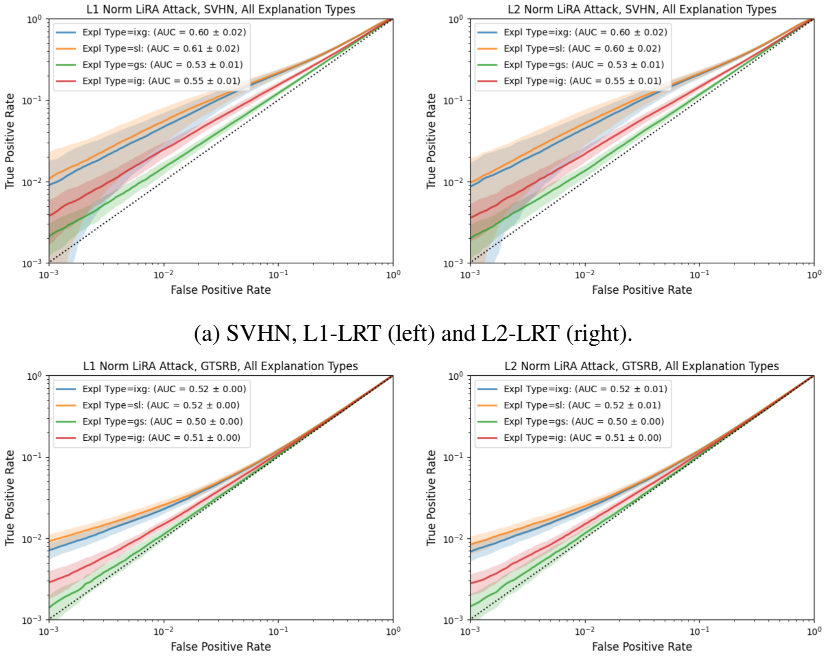


(a) L1-LRT log-scaled ROC curves.

(b) L2-LRT log-scaled ROC curves.

*Figure 8.* L1-LRT and L2-LRT attack results for the CIFAR-10 (first and second from the left), CIFAR-100 (second from the right), and Food 101 (right) datasets, on different model architectures than are presented in the main body.

In the main text, we presented L1-LRT and L2-LRT attack ROCs for the CIFAR-10, CIFAR-100, and Food 101 datasets but excluded plots on the SVHN and GTSRB datasets. In Figure 9, we show these excluded plots, coming from the `vit_small_patch16_224` model.



(a) SVHN, L1-LRT (left) and L2-LRT (right).

(b) GTSRB, L1-LRT (left) and L2-LRT (right).

*Figure 9.* L1-LRT and L2-LRT attack ROCs for the SVHN and GTSRB datasets, across all explanation types.

### I.1. Comparing L1-LRT and L2-LRT

We observe in the main text, as well as in this appendix, that L1-LRT attacks are more successful than L2-LRT attacks overall. We hypothesize that this may be related to the gradient of the cross-entropy loss with respect to weights in the last hidden layer of the underlying model. Let $w_{ji}$ be the weight linking hidden unit value $h_j$ to the (pre-activation) output $z_i$: this means $z_i = h_j w_{ji} + b_j$, where $b_j$ is a bias term. For feature vector $\mathbf{x}$, let $y_i$ be the $i$th element of the ground-truth one-hot encoded vector $\mathbf{y} \in \{0,1\}^k$. Let $\hat{y}_i = p(\mathbf{x})_i$ represent the $i$th element of the model's predicted probability distribution over the classes. Beaujour (2017) derives the gradient of cross-entropy loss with respect to weight $w_{ji}: \dfrac{\partial \mathcal{L}}{\partial w_{ji}} = h_j(\hat{y}_i - y_i)$. The gradient is linear in the distance between predicted and true class probabilities, so intuitively, gradient descent "travels linearly" through this probability vector space. The gradient of the model *output* with respect to the input features is closely related to the gradient of the model *loss* with respect to the final-layer weights, since model weights directly reflect how input features map to model predictions.

We thus hypothesize that the L1 norm of the gradient of the model output with respect to input features, which is also a "linear" distance metric, better reflects the linear behavior of gradient descent than does the L2 norm of the gradient. However, this is but a hypothesis, and we encourage future exploration into this result.

## J. Non-Private Ablation Experiments

### J.1. On the Computational Efficiency versus the Privacy Risk of Explanations

Our tables and figures present a salient observation that we have not yet verbalized: that across datasets, attacks on Input * Gradient (IXG) and Saliency (SL) generally perform best, while attacks on Integrated Gradients (IG) and Gradient SHAP (GS) tend to have lower success. This finding highlights a trade-off between the computational efficiency of an explanation method and its susceptibility to privacy attack; according to Table 7, IXG and SL attributions are much faster to compute than GS and IG explanations, at least with respect to our compute resources and platform.

An attacker can more readily leverage explanation methods that are computationally efficient: our attacks require computing a full set of feature attributions based on each shadow model, and this process is significantly easier if we use more efficient explanation methods. Thus, this trade-off is itself a sign of privacy risk.

*Table 7.* Comparing computational efficiency of explanation methods. Time (mm:ss) taken for each explanation method to generate attributions for 200 CIFAR-10 examples.

| Explanation Type | Time (200 Iters) |
|---|---|
| IXG | 1:07 |
| SL | 1:07 |
| GS | 6:24 |
| IG | 13:12 |

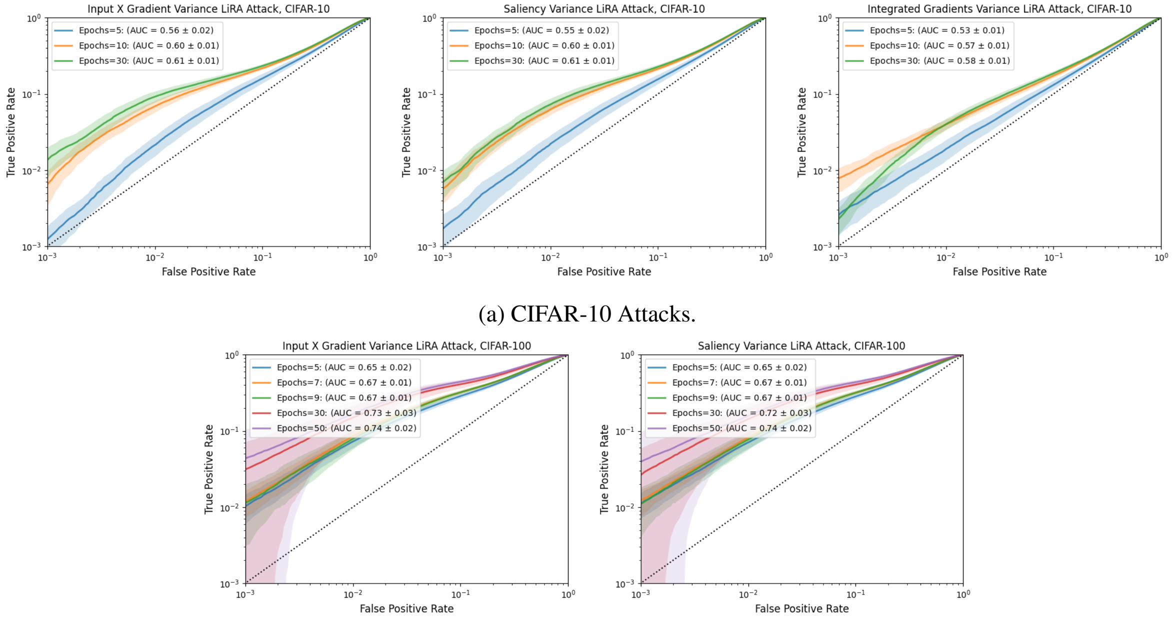
On the thread of comparing explanation methods with one another, one redeeming observation, however, is that because GS and IG *theoretically* satisfy desirable axiomatic properties that SL and IXG do not, the observation that GS and IG are less susceptible to privacy attack is auspicious from an axiomatic approach: the explanations with axiomatic properties are also better defended against privacy risk. (Recall that Appendix C highlights these axiomatic properties.)

### J.2. On the Impact of Overfitting and Underfitting

Figure 10 presents attack performance plots across different fine-tuning epoch counts on CIFAR-10 and CIFAR-100 data using the `vit_small_patch16_224` model. We experiment across the following epoch counts for the two datasets:

- CIFAR-10: 5, 10, 30 (30 is default)
- CIFAR-100: 5, 7, 9, 30, 50 (9 is default)

**CIFAR-10 and Underfitting** Figure 10 shows that even when the model is fine-tuned on CIFAR-10 for 10 epochs (well below the "chosen" 30 epoch setting), VAR-LRT still performs successfully, at least compared with the thresholding attack,



(a) CIFAR-10 Attacks.

(b) CIFAR-100 Attacks.

*Figure 10.* VAR-LRT on CIFAR-10 and CIFAR-100 data; multiple epoch settings. Each plot shows ROC curves of attacks for a single dataset and explanation type, with each curve within each plot corresponding to a different epoch setting.

not only on average (through improved AUC) but especially in the FPR=0.001 and FPR=0.01 regions. Although we cannot make such strong statements about statistical significance for the 5 epochs setting, the ROC curves and reported AUC values still show higher success for the VAR-LRT attack compared to the thresolding attack.

**CIFAR-100 and Overfitting** Figure 10 shows that even when the model is fine-tuned on CIFAR-100 for 30 or 50 epochs (well above the "chosen" 9 epoch setting), the ROC curves and reported AUC values show higher success for the VAR-LRT attack. Specifically, VAR-LRT captures significantly higher TPR than the thresholding attack when FPR is between 0.01 and 0.1. VAR-LRT shows improvement on average in other metrics (AUC, TPR  FPR=0.001) as well, albeit without statistical significance. Hence, VAR-LRT's performance exceeds that of the thresholding attack and is objectively successful even when the model is overfit or underfit.

**Overfitting and Training Data Leakage** This figure also shows that the longer we fine-tune a model for (that is, the more epochs the model is trained for), the more susceptible to privacy attack the ensuing explanations are; this result holds across datasets and explanation methods. Intuitively, the more epochs the model is trained for, the more "familiar" the model becomes on training points, and the further away the decision boundary moves from these points. Explanations, by design, capture model behavior, and model behavior varies more between training and non-training examples as it becomes more "familiar" with training examples. Hence, it follows intuitively that training data explanations will indeed on average differ more from non-training data explanations. This result more broadly reveals a downside to model overfitting beyond the more commonly discussed implication that overfitting leads to low model generalizability on unseen data: overfitting leads to *increased data privacy risk*, especially as we add transparency to models through explainability. Consequently, as researchers investigate privacy risk defenses in model training, it is important and promising to consider approaches that directly or indirectly avoid overfitting.

### J.3. On the Impact of More Shadow Models

For this investigation, we fix the number of evaluation runs per attack setup to 20. Figure 11 shows the IXG L1-LRT attack on CIFAR-10 and the `vit_small_patch16_224` model over $[32, 64, 128]$ shadow models. We observe that even quadrupling the number of shadow models from 32 to 128 has no impact on attack performance. Several membership inference attack works evaluate their results with more shadow models than the 32 and 16 used in this work. For example, Carlini et al. (2021) frequently use 64 and 128 shadow models in their experiments, and Abascal et al. (2023) use 128 shadow models. We use fewer shadow models out of respect for compute resource limitations, and Figure 11 shows that we do not sacrifice on attack performance in doing so.

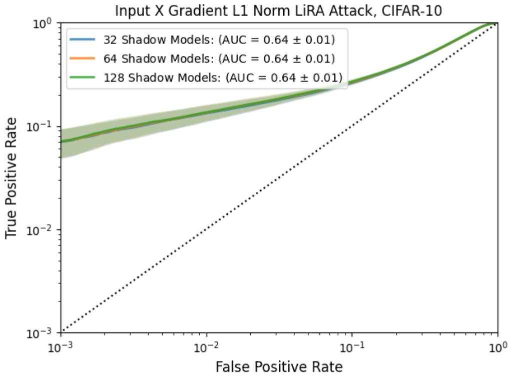


*Figure 11.* Impact of changing the number of shadow models. We show log-scaled ROC curves for the IXG L1-LRT attack on CIFAR-10 over $[32, 64, 128]$ shadow models, using the `vit_small_patch16_224` model. Each curve is taken across 20 evaluation runs. We observe no difference in attack performance after changing shadow model count.

## K. Performance of Models Fine-Tuned with Differential Privacy

Table 8 shows average train and test accuracies for each dataset at DP $\varepsilon$ values of $0.5, 1.0, 2.0, 8.0$, and $\infty$ (non-DP). The table brings out a surprising observation: the test accuracies of some of the $\varepsilon$-DP models with higher $\varepsilon$ values are *higher* than the test accuracy of the baseline non-private model. This is counter-intuitive and surprising: even at high values of $\varepsilon$, we inject noise into gradients during fine-tuning, so why would the model ever perform *better* than if no random noise were injected into gradient computations? Formally, this should not happen. This observation is precisely why we report train accuracies *alongside* test accuracies, even though we care about test accuracies more—train accuracies provide more perspective on this finding. The non-private models have considerably higher train accuracy than any of the private models but, perhaps as a result, they overfit more to the training data. The non-private models perform relatively poorly on training data, but it is *by virtue* of the injected random noise that these models generalize better, yielding high test accuracy. The private models' performance relative to one another is expected according to the well-known privacy-utility tradeoff: the lower the value of $\varepsilon$ (i.e. the stronger the privacy protection), the lower the model accuracy.

Objectively, however, the DP fine-tuned models have high test accuracy and strong performance. We chose to run experiments on large pre-trained foundation models using automatic gradient norm clipping (in the DP algorithm) in the first place for this purpose: so that the privately fine-tuned models we work with have high utility.

## L. More Results on the Impact of Differentially Private Fine-Tuning on Attack Success

### L.1. VAR-LRT DP Results

Figure 12 shows log-scaled ROC curves highlighting the impact of DP fine-tuning on VAR-LRT attack success. Each subplot features one dataset and explanation type over different privacy settings. We observe that in each subplot—that is, across datasets and explanation types—the ROC curves corresponding to models fine-tuned with DP hug the "random guessing" diagonal line much more closely than the baseline ROC curves do. For CIFAR-10, the $\varepsilon = 8.0$ setting shows minimal but nonzero attack success. However, even the least private $\varepsilon = 8.0$ setting yields unsuccessful attacks on CIFAR-100. Moreover, the other values of $\varepsilon = 0.5, 1.0, 2.0$ (corresponding to stronger privacy protection settings) yield attacks that behave no better than random guessing, both on average and when FPR is low.

### L.2. L1/L2-LRT DP Results

Table 9 shows the $\text{TPR}_{.001}$, $\text{TPR}_{.01}$, and AUC L1-LRT attack metrics for the $\varepsilon = 1.0, 2.0, 8.0$ privacy settings, the CIFAR-10 and CIFAR-100 datasets, and the IXG and SL explanation methods. For each metric, we also show results on non-private attacks ($\varepsilon = \infty$) over 16 shadow models per run (and 17 total runs). This table bolsters, from a numerical perspective, the

*Table 8.* Model train and test accuracies for each dataset at DP $\epsilon$ values of 0.5, 1.0, 2.0, 8.0, and $\infty$ (non-DP). Each model is fine-tuned with the chosen epoch setting highlighted in Table 5. We report the means computed over 17 independent runs, as well as the standard deviations.

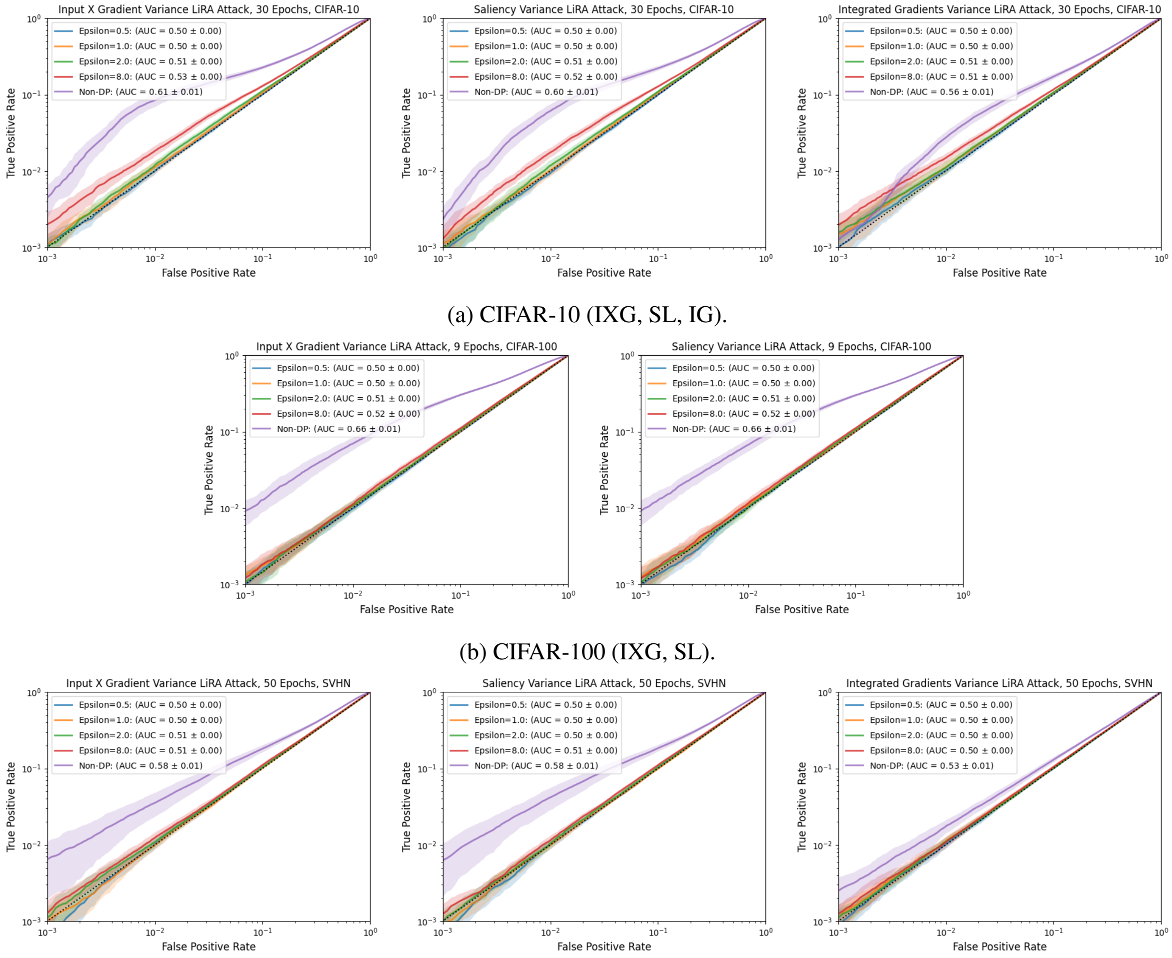
| | $\varepsilon$ | Accuracy | |
|---|---|---|---|
| | | Train Accuracy (%) | Test Accuracy (%) |
| CIFAR-10 | 0.5 | $96.313 \pm 0.301$ | $95.796 \pm 0.148$ |
| | 1.0 | $97.123 \pm 0.196$ | $96.234 \pm 0.157$ |
| | 2.0 | $97.972 \pm 0.147$ | $96.496 \pm 0.162$ |
| | 8.0 | $99.021 \pm 0.101$ | $96.611 \pm 0.162$ |
| | $\infty$ (non-DP) | $100.00 \pm 0.000$ | $96.064 \pm 0.613$ |
| CIFAR-100 | 0.5 | $73.754 \pm 1.262$ | $71.488 \pm 1.176$ |
| | 1.0 | $80.082 \pm 0.638$ | $77.346 \pm 0.767$ |
| | 2.0 | $83.305 \pm 0.566$ | $80.907 \pm 0.520$ |
| | 8.0 | $86.889 \pm 0.383$ | $84.343 \pm 0.321$ |
| | $\infty$ (non-DP) | $98.722 \pm 0.321$ | $80.109 \pm 0.590$ |
| SVHN | 0.5 | $76.708 \pm 2.066$ | $75.167 \pm 2.399$ |
| | 1.0 | $86.153 \pm 0.741$ | $84.149 \pm 0.989$ |
| | 2.0 | $89.694 \pm 0.378$ | $86.922 \pm 0.578$ |
| | 8.0 | $93.168 \pm 0.362$ | $88.922 \pm 0.406$ |
| | $\infty$ (non-DP) | $99.542 \pm 0.223$ | $91.558 \pm 1.056$ |

conclusions we have drawn from the ROC curves in the main body.

*Table 9.* L1-LRT attack success, DP vs. non-DP. We report the $\text{TPR}_{.001}$, $\text{TPR}_{.01}$, and AUC L1-LRT attack metrics for the $\varepsilon = 1.0, 2.0, 8.0$ privacy settings, the CIFAR-10 and CIFAR-100 datasets, and the IXG and SL explanation methods. For each metric, we also show results on non-private attacks ($\varepsilon = \infty$). Results are averaged over 17 evaluation runs models per run (with 16 shadow models used per run).

| Metric | Epsilon ($\varepsilon$) | CIFAR-10 | | CIFAR-100 | |
|---|---|---|---|---|---|
| | | `IXG` | `SL` | `IXG` | `SL` |
| $\text{TPR}_{.001} \downarrow$ | 1.0 | $0.0019 \pm 0.0008$ | $0.0016 \pm 0.0007$ | $0.0013 \pm 0.0008$ | $0.0013 \pm 0.0007$ |
| | 2.0 | $0.0019 \pm 0.0007$ | $0.0021 \pm 0.0011$ | $0.0012 \pm 0.0007$ | $0.0013 \pm 0.0009$ |
| | 8.0 | $0.0040 \pm 0.0014$ | $0.0042 \pm 0.0016$ | $0.0016 \pm 0.0008$ | $0.0015 \pm 0.0011$ |
| | $\infty$ (non-DP) | $0.0895 \pm 0.0213$ | $0.0902 \pm 0.0223$ | $0.0225 \pm 0.0089$ | $0.0195 \pm 0.0079$ |
| $\text{TPR}_{.01} \downarrow$ | 1.0 | $0.0128 \pm 0.0023$ | $0.0126 \pm 0.0022$ | $0.0109 \pm 0.0018$ | $0.0125 \pm 0.0022$ |
| | 2.0 | $0.0159 \pm 0.0025$ | $0.0164 \pm 0.0029$ | $0.0129 \pm 0.0025$ | $0.0115 \pm 0.0023$ |
| | 8.0 | $0.0302 \pm 0.0044$ | $0.0305 \pm 0.0047$ | $0.0153 \pm 0.0031$ | $0.0154 \pm 0.0035$ |
| | $\infty$ (non-DP) | $0.1530 \pm 0.0216$ | $0.1529 \pm 0.0230$ | $0.1332 \pm 0.0352$ | $0.1308 \pm 0.0033$ |
| AUC $\downarrow$ | 1.0 | $0.5087 \pm 0.0035$ | $0.5092 \pm 0.0031$ | $0.5018 \pm 0.0028$ | $0.5020 \pm 0.0032$ |
| | 2.0 | $0.5149 \pm 0.0040$ | $0.5142 \pm 0.0048$ | $0.5090 \pm 0.0048$ | $0.5100 \pm 0.0045$ |
| | 8.0 | $0.5327 \pm 0.0044$ | $0.5325 \pm 0.0043$ | $0.5219 \pm 0.0041$ | $0.5226 \pm 0.0042$ |
| | $\infty$ (non-DP) | $0.6383 \pm 0.0095$ | $0.6383 \pm 0.0095$ | $0.7171 \pm 0.0084$ | $0.7153 \pm 0.0094$ |

Figure 13 provides ROC curves showing L2-LRT attack success on non-private vs. DP fine-tuned models.

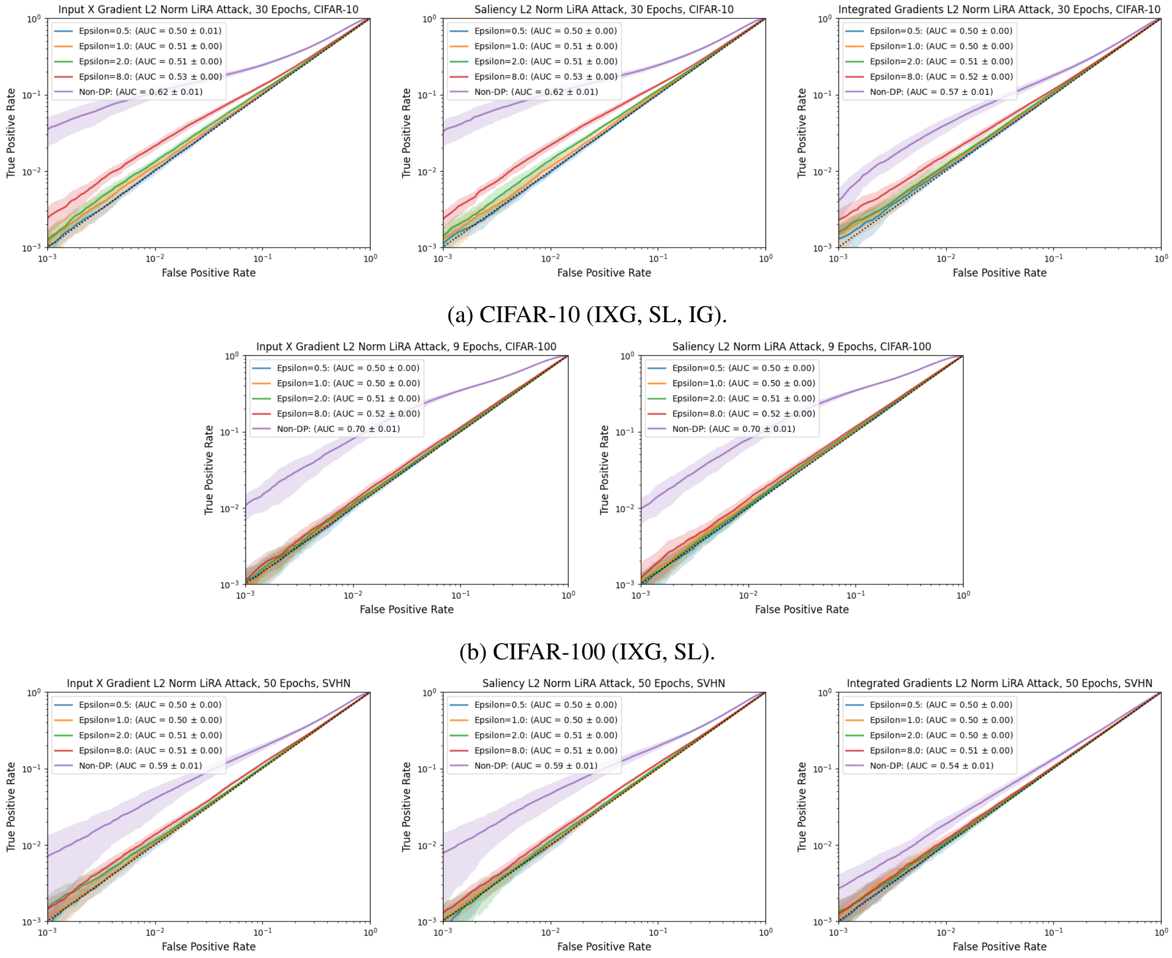


(a) CIFAR-10 (IXG, SL, IG).

(b) CIFAR-100 (IXG, SL).

(c) SVHN (IXG, SL, IG).

*Figure 12.* VAR-LRT attack success of non-private vs. DP fine-tuned models. We show one plot per explanation method: IXG (left), SL (middle), and (with the exception of CIFAR-100) IG (right). Each subplot shows curves for $\epsilon = 0.5, 1.0, 2.0, 8.0, \infty$.



(a) CIFAR-10 (IXG, SL, IG).

(b) CIFAR-100 (IXG, SL).

(c) SVHN (IXG, SL, IG).

*Figure 13.* L2-LRT attack success of non-private vs. DP fine-tuned models. We show one plot per explanation method: IXG (left), SL (middle), and (with the exception of CIFAR-100) IG (right). Each subplot shows curves for $\epsilon = 0.5, 1.0, 2.0, 8.0$.