Learning To Invert: Simple Adaptive Attacks for Gradient Inversion in Federated Learning

Ruihan Wu[1 *] **Xiangyu Chen**[1 *] **Chuan Guo**[2] **Kilian Q. Weinberger**[1]

[1]Cornell University, USA
[2]Meta AI, USA
[*]equal contribution

Abstract

Gradient inversion attack enables the recovery of training samples from model gradients in federated learning (FL), and constitutes a serious threat to data privacy. To mitigate this vulnerability, prior work proposed both principled defenses based on differential privacy, as well as heuristic defenses based on gradient compression as countermeasures. These defenses have so far been very effective, in particular those based on gradient compression that allow the model to maintain high accuracy while greatly reducing the effectiveness of attacks. In this work, we argue that such findings underestimate the privacy risk in FL. As a counterexample, we show that existing defenses can be broken by a simple adaptive attack, where a model trained on auxiliary data is able to invert gradients on both vision and language tasks.

1 INTRODUCTION

Federated learning (FL; [McMahan et al., 2017]) is a popular framework for distributed model training on sensitive user data. Instead of centrally storing the training data, FL operates in a server-client setting where the server hosts the model and has no direct access to clients' data. The clients can apply the model to their private data and send gradient updates back to the server. This learning regime promises data privacy as users share only gradients but never any raw data. However, recent work [Zhu et al., 2019, Zhao et al., 2020, Geiping et al., 2020] showed that despite these efforts, the server is still able to recover training data from gradient updates, violating the promise of data privacy in FL. These so-called *gradient inversion attacks* operate by optimizing over the input space to search for samples whose gradient matches that of the observed gradient, and such attacks remain effective even when clients utilize secure aggregation [Bonawitz et al., 2016] to avoid revealing individual gradients [Yin et al., 2021, Jeon et al., 2021].

As countermeasures against these gradient inversion attacks, prior work proposed both principled defenses based on differential privacy [Abadi et al., 2016], as well as heuristics that compress the gradient update through gradient pruning [Aji and Heafield, 2017] or sign compression [Bernstein et al., 2018]. In particular, gradient compression defenses have so far enjoyed great success, severely hindering the effectiveness of existing optimization-based attacks [Zhu et al., 2019, Jeon et al., 2021] while maintaining a similar level of model performance. As a result, these limitations seemingly diminish the threat of gradient inversion attacks in practical FL applications.

In this paper, we argue that evaluating defenses on existing optimization-based attacks may provide a false sense of security. To this end, we propose a simple *learning-based* attack—which we call *Learning To Invert* (LTI)—that trains a model to invert gradient updates and recover client samples; see Figure 1 for an illustration. We assume that the adversary (*i.e.*, the server) has access to an *auxiliary dataset* whose distribution is similar to that of the private data. The gradient inversion model trains on samples in the auxiliary dataset, with corresponding gradients provided by the global model. Our attack is highly adaptable to different defense schemes, since applying a defense simply amounts to training data augmentation for the gradient inversion model.

We empirically demonstrate that LTI can successfully circumvent defenses based on gradient perturbation (*i.e.*, using differential privacy; [Abadi et al., 2016]), gradient pruning [Aji and Heafield, 2017] and sign compression [Bernstein et al., 2018] on both vision and language tasks.

- Vision: We evaluate on the CIFAR10 [Krizhevsky et al., 2009] classification dataset for both LeNet and ResNet20. LTI attains recovery accuracy close to that of the best optimization-based method when no defense is applied, and significantly outperforms all prior attacks under defense settings.

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

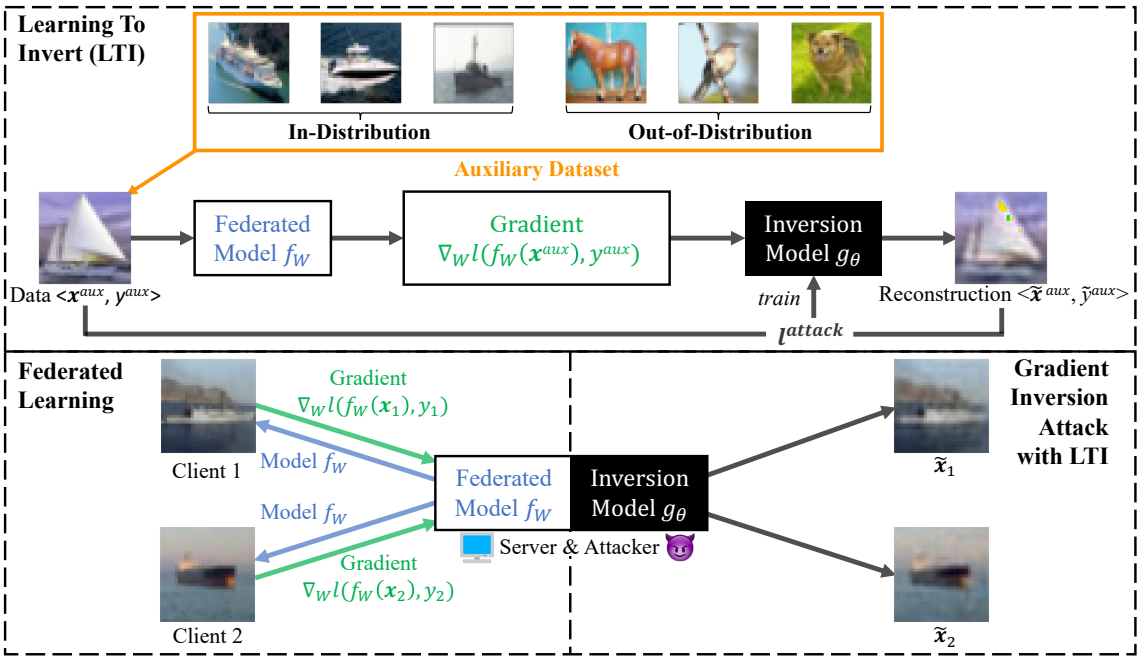

Figure 1: Illustration of federated learning (FL) and gradient inversion methods. The goal of gradient inversion is to recover training data $(\mathbf{x}, y)$ from the observed gradient $\nabla_{\mathbf{w}}\ell(f_{\mathbf{w}}(\mathbf{x}), y)$. Optimization-based methods (*e.g.*, [Zhu et al., 2019, Geiping et al., 2020, Yin et al., 2021, Jeon et al., 2021]) directly optimize $(\tilde{\mathbf{x}}, \tilde{y})$ in search for a sample that produces gradient similar to that of $(\mathbf{x}, y)$. Our proposed learning-based approach, which we call *Learning to Invert*, instead trains an inversion model g_θ to reconstruct training samples from their gradient.

- NLP: We experiment with both text classification task on CoLA [Warstadt et al., 2018] and causal language model training on the WikiText [Merity et al., 2016] dataset, where LTI attains state-of-the-art performance in all settings, with or without defense.

Given the strong empirical performance of LTI and its adaptability to different learning tasks and defense mechanisms, we advocate for its use as a simple baseline for future studies on gradient inversion attacks in FL.

2 BACKGROUND

Federated learning. The objective of federated learning [McMahan et al., 2017] is to train a machine learning model in a distributed fashion without centralized collection of training data. In detail, let $f_{\mathbf{w}}$ be the *global federated model* parameterized by $\mathbf{w}$, and consider a supervised learning setting that optimizes $\mathbf{w}$ by minimizing a loss function ℓ over the training set $\mathcal{D}_{\text{train}}$: $\sum_{(\mathbf{x},y)\in\mathcal{D}_{\text{train}}} \ell(f_{\mathbf{w}}(\mathbf{x}), y)$. In centralized learning this is typically done by computing a stochastic gradient $\frac{1}{B}\sum_{i=1}^{B}\nabla_{\mathbf{w}}\ell(f_{\mathbf{w}}(\mathbf{x}_i), y_i)$ over a randomly drawn batch of data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_B, y_B)$ and minimizing ℓ using stochastic gradient descent.

In FL, instead of centrally collecting $\mathcal{D}_{\text{train}}$, the training set $\mathcal{D}_{\text{train}}$ is distributed across multiple clients and the model $f_{\mathbf{w}}$ is stored on a central server. At each iteration, the model parameter $\mathbf{w}$ is transmitted to each client to compute the per-sample gradients $\{\nabla_{\mathbf{w}}\ell(f_{\mathbf{w}}(\mathbf{x}_i), y_i)\}_{i=1}^{B}$ locally over a set of B clients. The server and clients then execute a *federated aggregation* protocol to compute the average gradient for the gradient descent update. A major advantage of FL is data privacy since clients do not need to disclose their data explicitly, but rather only send their gradient $\nabla_{\mathbf{w}}\ell(f_{\mathbf{w}}(\mathbf{x}_i), y_i)$ to the server. Techniques such as secure aggregation [Bonawitz et al., 2016] and differential privacy [Dwork et al., 2006, 2014] can further reduce the risk of privacy leakage from sending this gradient update.

Gradient inversion attack. Despite the promise of data privacy in FL, recent work showed that the heuristic of sending gradient updates instead of training samples themselves in fact provides a false sense of security. Zhu et al. [2019] showed in their seminal paper that it is possible for the server to recover the full batch of training samples given aggregated gradients. These *optimization-based* gradient inversion attacks operate by optimizing a set of *dummy data* $\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_B$ and labels $\tilde{y}_1, \ldots, \tilde{y}_B$ to match their gradients to the observed gradients with cost function:

$$\min_{\tilde{\mathbf{x}},\tilde{\mathbf{y}}} \left\| \sum_{i=1}^{B} \nabla_{\mathbf{w}}\ell(f_{\mathbf{w}}(\tilde{\mathbf{x}}_i), \tilde{y}_i) - \sum_{i=1}^{B} \nabla_{\mathbf{w}}\ell(f_{\mathbf{w}}(\mathbf{x}_i), y_i) \right\|_2^2 \tag{1}$$

For image tasks, since Equation 1 is differentiable to $\tilde{\mathbf{x}}_i$ and $\tilde{y}_i$, and the model parameters $\mathbf{w}$ are accessible to the server, the server can simply optimize Equation 1 using gradient-based search. Doing so yields recovered samples $(\tilde{\mathbf{x}}_i, \tilde{y}_i)$ that closely resemble actual samples $(\mathbf{x}_i, y_i)$ in the batch. In practice this approach is highly effective, and follow-up works proposed several optimizations to further improve its recovery accuracy [Geiping et al., 2020, Yin et al., 2021, Jeon et al., 2021].

For language tasks this optimization problem is considerably more complex since the samples $\mathbf{x}_1, \ldots, \mathbf{x}_B$ are sequences of discrete tokens, and optimizing Equation 1 amounts to solving a discrete optimization problem. To circumvent this difficulty, Zhu et al. [2019] and Deng et al. [2021] instead optimize the *token embeddings* to match the observed gradient and then maps the recovered embeddings to their closest tokens in the embedding layer to recover the private text. In contrast, Gupta et al. [2022] leveraged the insight that the gradient of the token embedding layer can be used to recover exactly the set of tokens present in the training sample, and used beam search to optimize the ordering of tokens for fluency to recover the private text.

Gradient inversion under the malicious server setting. The aforementioned gradient inversion attacks operate under the *honest-but-curious* setting where the server faithfully executes the federated learning protocol, but attempts to extract private information from the observed gradients. Fowl et al. [2021], Boenisch et al. [2021] and Fowl et al. [2022] consider a stronger *malicious server* threat model, which allows the server to transmit arbitrary model parameters $\mathbf{w}$ to the clients. Under this threat model, it is possible to carefully craft the model parameters such that the training sample can be recovered exactly from its gradient even when the batch size B is large. While this setting is certainly realistic and relevant, our paper operates under the weaker honest-but-curious threat model.

3 LEARNING TO INVERT: LEARNING-BASED GRADIENT INVERSION ATTACKS

3.1 PROBLEM SET-UP

Motivation. The threat of gradient inversion attack has prompted prior work to employ defense mechanisms to mitigate this privacy risk in FL [Zhu et al., 2019, Jeon et al., 2021]. Intuitively, such defenses reduce the amount of information contained in the gradient about the training sample by either perturbing the gradient with noise [Abadi et al., 2016] or compressing them [Aji and Heafield, 2017, Bernstein et al., 2018], making recovery much more difficult. However, doing so also reduces the amount of information a sample can provide for training the global model, and hence has a negative impact on the model's performance. This is certainly true for principled defenses based on differential privacy [Dwork et al., 2006] such as gradient perturbation [Abadi et al., 2016]. However, defenses based on gradient compression seemingly provide a much better privacy-utility trade-off, effectively preventing the attack and reducing communication costs with minor reduction in model performance [Zhu et al., 2019].

The empirical success of existing defenses seemingly diminish the threat of gradient inversion attacks in FL. However, we argue that optimization-based attacks underestimate the power of the adversary: If the adversary has access to an auxiliary dataset $\mathcal{D}_{\text{aux}}$, they can train a *gradient inversion model* to recover $\mathcal{D}_{\text{aux}}$ from its gradients computed on the global model. As we will establish later, this greatly empowers the adversary, exposing considerable risks to federate learning.

Threat model. We consider the setting where the adversary is an *honest-but-curious* server, who executes the learning protocol faithfully but aims to extract private training data from the observed gradients. Hence, in each FL iteration, the adversary has the knowledge of model weights $\mathbf{w}$ and aggregated gradients. Moreover, we assume the adversary has an auxiliary dataset $\mathcal{D}_{\text{aux}}$, which could be in-distribution or a mixture of in-distribution and out-of-distribution data. This assumption is similar to the setting in Jeon et al. [2021], which assumes a generative model that is trained from the in-distribution data, and is common in the study of other privacy attacks such as membership inference [Shokri et al., 2017].

In this paper, we focus on the attack against defense mechanisms (DM) in prior work [Zhu et al., 2019, Jeon et al., 2021]. Thus, we assume the adversary receives the aggregated gradients $\sum_{i=1}^{B} \mathrm{DM}\left[\nabla_{\mathbf{w}} \ell(f_{\mathbf{w}}(\mathbf{x}_i), y_i)\right]$ at one of following DM settings:

1. *Gradient without defense.* The gradient before the aggregation is the original gradient without any defense. Most previous papers focus on this common setting.
2. *Sign compression.* [Bernstein et al., 2018] applies a element-wise sign function to gradient before the aggregation, which compresses the gradient to *one bit per dimension*.
3. *Gradient pruning with pruning rate* α [Aji and Heafield, 2017] zeroes out the bottom $1-\alpha$ fraction of coordinates of $\nabla_{\mathbf{w}} \ell(f_{\mathbf{w}}(\mathbf{x}), y)$ in terms of absolute value, which effectively compresses the gradient to $(1-\alpha)m$ dimensions, where m denotes the model size.
4. *Gradient perturbation with Gaussian standard deviation* σ [Abadi et al., 2016] is a differentially private mechanism used commonly for training private models. A Gaussian random vector $\mathcal{N}(\mathbf{0}, \sigma^2 I)$ is added to the gradient, which one can show achieves ϵ-local differential privacy [Kasiviswanathan et al., 2011] with $\epsilon = O(1/\sigma)$.

3.2 LEARNING TO INVERT (LTI)

Definition of the learning problem. Having knowledge of the model weights and the defense mechanism DM, the adversary is able to generate the gradient $\text{grad}_{S_B}^{\text{DM}} = \sum_{i=1}^{B} \text{DM}\left[\nabla_{\mathbf{w}} \ell(f_{\mathbf{w}}(\mathbf{x}_i^{\text{aux}}), y_i^{\text{aux}})\right]$ for any batch of samples $S_B = \{(\mathbf{x}_1^{\text{aux}}, y_1^{\text{aux}}) \cdots (\mathbf{x}_B^{\text{aux}}, y_B^{\text{aux}})\}$ in the auxiliary dataset. This allows the adversary to learn a *gradient inversion model* $g_\theta : \mathbb{R}^m \to \mathbb{R}^{B \times d}$ (d denotes data dimension), parameterized by θ , to predict this batch of data point S_B from the aggregated gradient $\text{grad}_{S_B}^{\text{DM}}$. The learning goal is to minimize the reconstruction error ℓ^{attack} of g_θ on the auxiliary dataset $\mathcal{D}^{\text{aux}}$:

$$\min_{\theta} \mathbb{E}_{S_B \sim \mathcal{D}^{\text{aux}}} \ell^{attack}\left(g_\theta\left(\text{grad}_{S_B}^{\text{DM}}\right), S_B\right). \quad (2)$$

We hereby explain the choice of the loss function ℓ^{attack} and the inversion model g_θ. Since g_θ needs to reconstruct data in batches, ℓ^{attack} should be permutation invariant w.r.t. the S_B. A common solution [Zhang et al., 2019] is to define ℓ^{attack} as:

$$\ell^{attack}\left(g_\theta\left(\text{grad}_{S_B}^{\text{DM}}\right), S_B\right) = \min_{\pi} \sum_{i=1}^{B} \ell_{single}^{attack}\left(\left(\text{grad}_{S_B}^{\text{DM}}\right)_i, \left(\mathbf{x}_{\pi(i)}^{\text{aux}}, y_{\pi(i)}^{\text{aux}}\right)\right), \quad (3)$$

where the minimization is over all possible permutation π. ℓ_{single}^{attack} is the loss function for a single pair of the prediction and target data. In practice, ℓ_{single}^{attack} can be a cross-entropy loss for discrete inputs or a L2 loss for continuous-valued inputs. As for the choice of the inversion model g_θ, we empirically find that a multi-layer perceptron (MLP) [Bishop et al., 1995] is sufficiently effective for the tasks in our experiments.

Comparison to optimization-based attacks. LTI is superior in its simplicity to optimization-based methods on generalization, for the following two aspects. Firstly, LTI doesn't explicitly have any terms relevant to data prior. It will learn the data property from the auxiliary dataset. However, optimization-based attacks usually manually encode the data prior in their objective functions, e.g. the total variation term in most optimization-based attacks to reconstruct image samples. Secondly, there's no need for careful adaptation to different defense mechanisms. As we know, in optimization-based attacks, for any FL defense mechanism, it is crucial to carefully design a corresponding objective function for gradient matching. In section 4, we will show that our simple approach is surprisingly effective at circumventing existing defenses for both language and vision data.

Dimensionality reduction for large models $f_{\mathbf{w}}$. One potential problem for LTI is that the gradients $\sum_{i=1}^{B} \text{DM}\left[\nabla_{\mathbf{w}} \ell(f_{\mathbf{w}}(\mathbf{x}_i^{\text{aux}}), y_i^{\text{aux}})\right]$ can be extremely high-dimensional. For example, ResNet20 [He et al., 2016] for vision tasks has $270K$ parameters and BERT [Devlin et al., 2019] for language tasks have approximately $110M$ trainable parameters. Such high-dimensional input to the model g_θ can lead to memory issues, as the first layer of the MLP would have $110M \times h$ parameters, where h denotes the size of the first hidden layer.

To address this issue, we use feature hashing [Weinberger et al., 2009] to reduce the dimensionality of the input gradient. In feature hashing, each gradient dimension $i \in [m]$ is randomly assigned to one of k bins (k is much smaller than the size of gradient m), formalized as $r(i) \in [k]$. We then sum up all gradient values in each bin, producing a compressed feature vector of size k. In other words, we project the aggregated gradient $\sum_{i=1}^{B} \text{DM}\left[\nabla_{\mathbf{w}} \ell(f_{\mathbf{w}}(\mathbf{x}_i^{\text{aux}}), y_i^{\text{aux}})\right]$ to $P\left(\sum_{i=1}^{B} \text{DM}\left[\nabla_{\mathbf{w}} \ell(f_{\mathbf{w}}(\mathbf{x}_i^{\text{aux}}), y_i^{\text{aux}})\right]\right)$ using the random projection matrix P given by:

$$P \in \{0,1\}^{k \times m} s.t.\ \forall i,\ P_{j,i} = 0\ (\forall j \neq r(i)),\ P_{r(i),i} = 1.$$

P in the definition is a sparse matrix with m nonzero element that can be saved in a memory efficient way. In this way, g_θ 's the memory footprint can be reduced to a constant independent from the gradient dimension.

4 EXPERIMENT

We evaluate LTI on both vision and language tasks. The evaluation results demonstrate that it vastly outperforms prior gradient inversion attacks, especially when *gradient defenses are applied*. Moreover, we show that LTI is able to perform surprisingly well even when the auxiliary data is out-of-distribution, which makes LTI more applicable in the real scenario[1].

4.1 EVALUATION ON VISION TASK

Federate learning tasks. For evaluating LTI on vision tasks, we experiment with image classification on CIFAR10 [Krizhevsky et al., 2009] and the training loss is the cross-entropy loss. The *original test split of CIFAR10* is used for FL training. For the generalization propose, we test the attacks on two different architectures as the FL model $f_{\mathbf{w}}$, which are LeNet [LeCun et al., 1998] and ResNet20 [He et al., 2016] with $15K$ and $270K$ parameters.

Defense mechanisms set-up. The adversary will receive the gradient aggregated from $B = 1$ or 4 clients, applied with no defense, sign compression, gradient pruning ($\alpha = 0.99$), or Gaussian perturbation ($\sigma = 0.1$).

Baselines. We compare our method with two gradient inversion attack baseline methods: *Inverting Gradients* (IG;

[1] Our code is released at `https://github.com/wrh14/Learning_to_Invert`.

Geiping et al. [2020]), a representative optimization-based method with limited data prior, and *Gradient Inversion with Generative Image Prior* (GI-GIP; Jeon et al. [2021]), the state-of-the-art optimization-based method that uses a generative model to encode the data prior. We make minor modifications to these attacks to adapt them to various defenses; see appendix for details. The threat model of LTI is most similar to GI-GIP since both use an auxiliary dataset to encode the data prior.

Set-up of LTI. We introduce the training set-up of LTI.

- *Auxiliary dataset.* We use the *original train split of CIFAR10* as the auxiliary dataset of the adversary. Notice that under this set-up, the auxiliary dataset is different from the dataset that the FL tasks are trained on, i.e. the one from which the aggregated gradients are computed.
- *Inversion model architecture.* Our inversion model g_θ is a three-layer MLP with hidden size 3K or 10K upon the memory limitation. The MLP takes the flattened gradient vector as input and outputs a $B \times 3072$-dimensional vector representing the flattened images. Because the size ResNet20 is large, we use feature hashing (see subsection 3.2) to reduce the target model gradient to 50% of its original dimensionality as input to the inversion model.
- *Training details.* The training objective ℓ_{single}^{attack} in Equation 3 is the mean squared error (MSE) between the output vector from MLP and the flattened ground truth image. We use the Adam [Kingma and Ba, 2014] optimizer for training g_θ. The model is trained for 200 epochs using training batch size 256. The initial learning rate is 10^{-4} with learning rate drop to 10^{-5} after 150 epochs.
- *Computation cost.* Our experiments are conducted using NVIDIA GeForce RTX 2080 GPUs and each training run takes about 1.5 hours.

Evaluation methodology. We evaluate LTI and the aforementioned baselines on $1,000$ random images from the CIFAR10 test split. To measure reconstruction quality, we use three common metrics: 1. *Mean squared error* (MSE) measures the average pixel-wise (squared) distance between the reconstructed image and the ground truth image. 2. *Peak signal-to-noise ratio* (PSNR) measures the ratio between the maximum image pixel value and MSE. 3. *Learned perceptual image patch similarity* (LPIPS) measures distance in the features space of a VGG [Simonyan and Zisserman, 2014] model trained on ImageNet. 4. *Structural similarity index measure* (SSIM) measures the perceived change in structural information

4.1.1 Main Results

Quantitative evaluation. Table 1 gives quantitative comparisons in the metric of MSE for IG, GI-GIP, and LTI against various defense mechanisms on CIFAR10; Tables of PSNR, LPIPS and SSIM are in the appendix due to space limit. When no defense mechanism is applied, GI-GIP achieves the best performance. It is not surprising because GI-GIP, explicitly encodes image-prior in an image generator, which is more tailored than LTI to image data. However, when the gradient is augmented with a defense mechanism that is underexplored, both IG and GI-GIP have considerably worse performance with MSE close to or above 0.1. By comparison, LTI outperforms both baselines significantly and consistently across all three defense mechanisms. For example, under gradient perturbation with $\sigma = 0.1$, which prior work believed is sufficient for preventing gradient inversion attacks [Zhu et al., 2019, Jeon et al., 2021], MSE can be as low as 0.012 for LTI. Our result, therefore, provides considerable additional insight for the level of empirical privacy achieved by DP-SGD [Abadi et al., 2016], and suggests that the theoretical privacy leakage as predicted by DP ϵ may be tighter than previously thought. These results validate that LTI has strong adaptation performance in various settings and can be a great baseline to show the vulnerability in those underexplored settings.

Qualitative evaluation. Figure 2 shows 4 random CIFAR10 test samples and their reconstructions under different defense mechanisms when the FL model is LeNet and $B = 1$. Without any defense in place, all three methods recover a considerable amount of semantic information about the object of interest, with both GI-GIP and LTI faithfully reconstructing the training sample. Under the sign compression defense, IG completely fails to reconstruct all 4 samples, while GI-GIP only successfully reconstructs the second image. In contrast, LTI is able to recover the semantic information in all 4 samples. Results for gradient pruning and gradient perturbation yield similar conclusions. More examples are given in the appendix.

4.1.2 Ablation Studies for Auxiliary Dataset

Since LTI learns to invert gradients using the auxiliary dataset, its performance depends on the quantity and quality of data available to the adversary. We perform ablation studies to better understand this dependence by changing the auxiliary dataset size and its distribution. All ablation studies are conducted in the setting where the FL model is LeNet and $B = 1$.

Varying the auxiliary dataset size. We randomly subsample the CIFAR10 training set to construct auxiliary datasets of size $\{500, 5000, 15000, 25000, 35000, 45000, 50000\}$ and evaluate the performance of LTI under various defenses. Figure 3(a) plots reconstruction MSE as a function of the auxiliary dataset size, which is monotonically decreasing as expected. Moreover, with just $5,000$ samples for training the inversion model (second point in each curve), the performance is nearly as good as when training using the

Table 1: MSE for baselines (IG and GI-GIP) and our method LTI on CIFAR10. As shown in the table, neither IG nor GIGIP works well when the defense mechanism is applied, while our method has the power to break the privacy protection from the compression and randomness.

FL model	Methods	$B = 1$				$B = 4$			
		None	**Sign Comp.**	**Grad. Prun.**	**Gauss. Pert.**	**None**	**Sign Comp.**	**Grad. Prun.**	**Gauss. Pert.**
LeNet	IG	0.022	0.116	0.138	0.150	0.105	0.265	0.169	0.206
	GI-GIP	**0.001**	0.091	0.043	0.124	**0.009**	0.082	0.180	0.157
	LTI (Ours)	0.004	**0.014**	**0.029**	**0.012**	0.015	**0.023**	**0.031**	**0.026**
ResNet20	IG	0.120	0.154	0.171	0.133	0.125	0.272	0.195	0.123
	GI-GIP	0.062	0.099	0.238	0.233	0.086	0.236	0.231	0.229
	LTI (Ours)	**0.018**	**0.013**	**0.023**	**0.021**	**0.038**	**0.035**	**0.038**	**0.039**

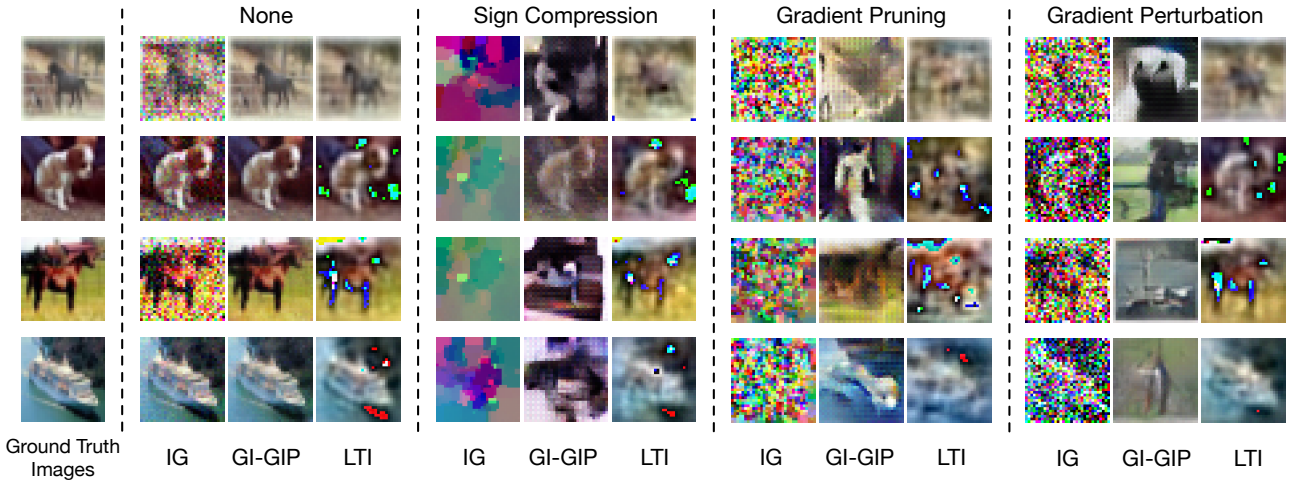

Figure 2: Comparison of LTI with IG and GI-GIP for reconstructing 4 random images in CIFAR10 when the FL model is LeNet and $B = 1$. Under sign compression, only LTI can partially reconstruct the images to recover the object of interest whereas both IG and GI-GIP fail to do so on most samples.

full CIFAR10 training set. Notably, even with the auxiliary dataset size as small as 500, the reconstruction MSE is *still lower than that of IG and GI-GIP* in Table 1. Corresponding figures for PSNR, LPIPS, and SSIM in the appendix show similar findings.

Varying the auxiliary data distribution. Although access to a large set of in-distribution data may be unavailable in practice, the adversary may still collect out-of-distribution samples for the auxiliary dataset. This is beneficial for the adversary since a model learning on out-of-distribution samples may transfer its knowledge to in-distribution data as well. To simulate this scenario, we divide CIFAR10 into two halves with disjoint classes and construct the auxiliary dataset by combining a β fraction of samples from the first half and a $1 - \beta$ fraction of samples from the second half for $\beta \in \{0, 0.01, 0.1, 0.3, 0.5, 0.7, 0.9, 1\}$. The target model $f_{\mathbf{w}}$ is trained only on samples from the first half, and hence the auxiliary set has the exact same distribution as the target model's data when $\beta = 1$ and only has out-of-distribution data when $\beta = 0$.

Figure 3(b) shows reconstruction MSE as a function of β. We make the following observations:

1. Even if the auxiliary dataset only contains 250 in-distribution samples ($\beta = 0.01$; second point in each curve), MSE of the inversion model is *still lower than that of the best baseline* in Table 1. For example, with the sign compression defense, LTI attains an MSE of ≤ 0.02, which is much lower than the MSE of 0.116 for IG and 0.091 for GI-GIP.
2. When the auxiliary dataset contains only out-of-distribution data ($\beta = 0$), the inversion model has a very high reconstruction MSE. In the next paragraph, we will propose a data augmentation method to improve the out-of-distribution generalization.

Out-of-distribution (OOD) auxiliary data. We further consider the auxiliary dataset that only has out-of-distribution data. Suppose the auxiliary data are images of the second half classes in CIFAR10 and the target model $f_{\mathbf{w}}$ is trained only on images from the first half (i.e. the setting of $\beta = 0$ when studying the data distribution). Instead

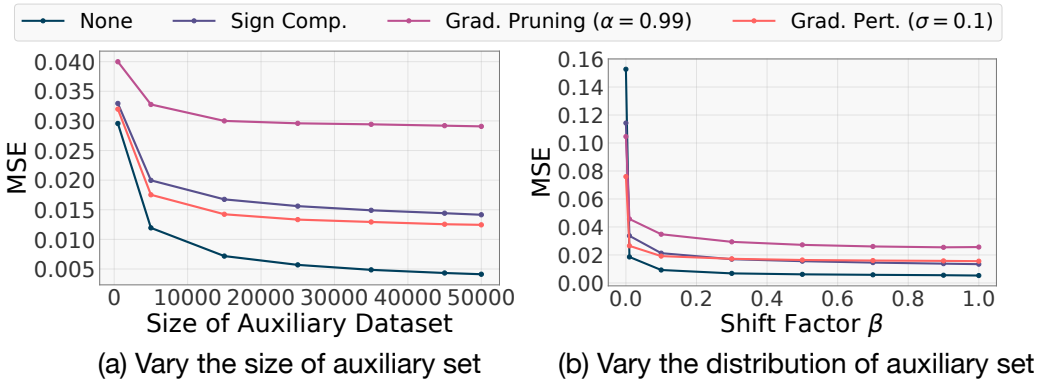

Figure 3: Ablation studies on size and distribution of the auxiliary dataset $\mathcal{D}_{\text{aux}}$. Under both severe data size limitation (left) and data distribution shift ($\beta = 0.01$; right), LTI is able to outperform both baselines in Table 1 when a defense is applied.

Table 2: MSE of LTI when the auxiliary dataset is out-of-distribution. LIT-OOD outperforms GI-GIP for all defense mechanism settings.

	None	Sign Comp.	Grad. Prune.	Gauss. Pert.
LTI-OOD	0.015	0.036	0.045	0.029
GI-GIP	0.001	0.091	0.043	0.124

of performing LTI with only the out-of-distribution data, we further augment the auxiliary dataset with the following steps:

1. Convert OOD data into the frequency domain by the discrete cosine transform (DCT).
2. Compute the mean and variance of OOD data in the DCT space.
3. Sample new data from a Gaussian with the mean and variance computed in step 2.
4. Convert new data back to the original image space.

Then we can train LTI with the OOD data and the augmented data from the steps above and name this method as LTI-OOD. Table 2 presents its MSE. By comparing it with baselines in Table 1, LTI-OOD is better or not worse than the baselines when the defense mechanisms are applied. Although LTI-OOD is worse than GI-GIP when no defense mechanism is applied, this is fair because GI-GIP utilizes the in-distribution data and this is a stronger data assumption than LTI-OOD.

To better understand this data augmentation, we also test the data augmentation where we estimate a Gaussian in the original image space and the MSE will increase from 0.015 to 0.045 when no defense is applied. We hypothesize this is because by fitting a Gaussian in the DCT domain, the frequency property as an image is kept so that the distribution is closer to the target image distribution.

4.2 EVALUATION ON LANGUAGE TASK

Federate learning tasks. For the evaluation on language data, we consider two common language tasks: text classifier training and causal language model training[2].

In the task of text classification, the classifier $f_{\mathbf{w}}$ is the BERT model [Devlin et al., 2019] with *frozen token embedding layer*. Fixing the token embedding layer is a common technique for language model fine-tuning [Sun et al., 2019], which also has privacy benefits since direct privacy leakage from the gradient magnitude of the token embedding layer can be prevented [Fowl et al., 2022, Gupta et al., 2022]. As a result, the trainable model contains about $86M$ parameters. The BERT classier is trained on CoLA [Warstadt et al., 2018] dataset using the cross-entropy loss.

In the task of causal language model, the language model $f_{\mathbf{w}}$ is a three-layer transformer [Vaswani et al., 2017] with *frozen token embedding layer*. The trainable model contains about $1.1M$ parameters. We train the language model on WikiText [Merity et al., 2016], where each training sample is limited to $L = 16$ tokens and the language model is trained to predict the next token $\mathbf{x}_l$ given $\mathbf{x}_{:l-1}$ for $l = 1, \ldots, L$ using the cross-entropy loss.

We set the *original test split of CoLA / WikiText dataset* as the dataset for the FL training, i.e. the dataset that the attacks will be test on.

Defense mechanisms set-up. The adversary will receive the gradient applied with no defense, sign compression, gradient pruning ($\alpha = 0.99$) and gaussian perturbation ($\sigma = 0.001$ for text classificatier training task and $\sigma = 0.01$ for causal language model training task) when $B = 1$.

Baseline. We compare LTI with TAG [Deng et al., 2021]—the state-of-the-art language model gradient inversion attack

[2]We follow the task setup and code in https://github.com/JonasGeiping/breaching

without utilizing the token embedding layer gradient[3]. The objective function for TAG is a slight modification of Equation 1 that uses both the ℓ_2 and ℓ_1 distance between the observed gradient and the gradient of dummy data. We also modify TAG slightly to adapt it to different defenses; see appendix for details.

Set-up of LTI. We follow the setup below for training the gradient inversion model g_θ.

- *Auxiliary dataset.* We use 8551 samples from the train split of CoLA or $\sim 1.8 \times 10^5$ samples from the train split of Wikitext as the auxiliary dataset.
- *Inversion model architecture.* For both FL tasks, we train a two-layer MLP with ReLU activation and first hidden-layer size 600 and second hidden-layer size $1,000$. The inversion model outputs L probability vectors each with size equal to the vocabulary size ($\sim 50,000$), and we train it using the cross-entropy loss to predict the L tokens given the target model gradient. We use feature hashing (see subsection 3.2) to reduce the target model gradient to 1% or 10% of its original dimensions as input to the inversion model when $f_{\mathbf{w}}$ is BERT or three-layer transformer.
- *Training details.* We use Adam [Kingma and Ba, 2014] to train the inversion model over 100 epochs with batch size 64. Learning rates are selected separately for each defense from $\{10^{-3}, 10^{-4}, 10^{-5}\}$.
- *Computation cost.* Our experiments are conducted using NVIDIA GeForce RTX 3090 GPUs and each training run takes about 3 hours.

Evaluation methodology. We evaluate LTI and the TAG baseline on $1,000$ samples from each task. To measure the quality of inverted text from attacks, we use four metrics: 1. *Accuracy*$(\%)$ measures the average token-wise zero-one accuracy. 2. *Rouge-1*$(\%)$, *Rouge-2*$(\%)$ and *Rouge-L*$(\%)$ measure the overlap of unigram, bigram, and length of longest common subsequence between the ground truth and the reconstructed text.

We also check the reconstructed texts from both TAG and LTI to see how the semantic meaning of the text is recovered and analyze the type of reconstruction error. This part is put in the appendix.

Results. Table 3 shows the quantitative comparison between LTI and TAG against various defenses. The overall trend is remarkably consistent: in all 4 metrics, LTI significantly outperforms TAG across different settings (7 out of 8). This shows that our method is easily adapted to the discrete language data and different defenses and is able to achieve great attack performance.

One observation is that the accuracy of inverted texts when the FL task is the causal language model training is overall much higher than the accuracy when the FL task is the text classifier training. We hypothesize this is because in the task of causal language model, the label in the cross entropy loss is the input sequence itself. On the other hand, The literature [Yin et al., 2021, Zhao et al., 2020] shows how easy it is to reconstruct the labels.

Another observation is that TAG has a relative low performance at most settings, it achieves the perfect accuracy at the setting of the sign compression when the FL task is the causal language model training. At the first impression, this perfect accuracy is very suspicious. By our carefully check, the explanation is that: if we treat the objective function when the gradient is applied sign compression as a special objective function when the adversary receives the full gradient, the result simply suggests that this special objective function is coincidently better than the one designed for the full gradient. Nevertheless, this phenomenon is not generalized to TAG for the other FL task. This demonstrates that the optimization-based method is very sensitive to the design of the object function.

Out-of-distribution (OOD) auxiliary data. Instead of assuming the adversary has in-distribution auxiliary texts, we relax this to only assuming the knowledge of the word frequency. Then, we can independently sample the word token for each position in the sentence and get a set of pseudo data. The distribution of pseudo data is out-of-distribution, because the pseudo data loses the inner dependency between different positions of a sentence. We train LTI with the pseudo data and name it as LTI-OOD.

The results of LTI-OOD are presented in Table 3. LTI outperforms TAG on both CoLA and WikiText dataset at most metrics for all settings of gradient defenses. Moreover, we can observe that LTI-OOD is even better than LTI on WikiText dataset. We hope this promising OOD results can motivate the exploration of OOD generalization of LTI in the future work.

5 CONCLUSION AND FUTURE WORK

We demonstrated the effectiveness of LTI—a simple learning-based gradient inversion attack—under realistic federated learning settings. For both vision and language tasks, LTI can match or exceed the performance of state-of-the-art optimization-based methods when no defense is applied, and significantly outperform all prior works under defenses based on gradient perturbation and gradient compression. Given its simplicity and versatility, we advocate the use of LTI as both a strong baseline for future research and a diagnostic tool for evaluating privacy leakage in FL.

Future work. This paper serves as preliminary work towards understanding the effectiveness of learning-based

[3] We do not compare against a more recent attack by Gupta et al. [2022] since it crucially depends on access to the token embedding layer gradient.

Table 3: Results for gradient inversion attack on two language tasks. The overall trend is remarkably consistent: in all 4 metrics, LTI significantly outperforms TAG across different settings (7 out of 8). This shows that our method is easily adapted and is able to achieve great attack performance.

(a) Text classifier training on CoLA dataset.

Defense	**None**				**Sign Compression**			
Method	Acc.	Rouge-1	Rouge-2	Rouge-L	Acc.	Rouge-1	Rouge-2	Rouge-L
TAG	8.38	51.23	6.88	29.35	1.62	8.81	0.00	8.09
LTI (Ours)	61.87	65.23	44.46	63.34	63.89	69.92	49.79	67.86
LTI-OOD (Ours)	52.03	45.86	29.46	45.79	50.77	49.07	30.86	48.80
Defense	**Gradient Pruning** ($\alpha = 0.99$)				**Gaussian Perturbation** ($\sigma = 0.001$)			
Method	Acc.	Rouge-1	Rouge-2	Rouge-L	Acc.	Rouge-1	Rouge-2	Rouge-L
TAG	5.69	43.30	6.90	26.96	5.12	33.85	2.94	22.01
LTI (Ours)	58.93	60.12	37.96	58.17	53.96	53.09	32.41	52.35
LTI-OOD (Ours)	38.68	35.66	23.11	35.46	37.96	33.75	21.85	33.55

(b) Causal language model training on WikiText dataset.

Defense	**None**				**Sign Compression**			
Method	Acc.	Rouge-1	Rouge-2	Rouge-L	Acc.	Rouge-1	Rouge-2	Rouge-L
TAG	74.13	71.92	50.64	68.46	100.00	100.00	100.00	100.00
LTI (Ours)	89.61	86.91	80.68	86.90	71.15	64.35	45.40	64.29
LTI-OOD (Ours)	91.14	89.43	85.11	89.41	88.06	84.66	76.46	84.64
Defense	**Gradient Pruning** ($\alpha = 0.99$)				**Gaussian Perturbation** ($\sigma = 0.01$)			
Method	Acc.	Rouge-1	Rouge-2	Rouge-L	Acc.	Rouge-1	Rouge-2	Rouge-L
TAG	34.34	48.50	10.21	35.60	64.34	66.19	37.86	59.55
LTI (Ours)	70.80	64.24	45.79	64.15	82.49	78.75	67.06	78.71
LTI-OOD (Ours)	86.19	82.56	73.04	82.50	90.25	87.39	81.94	87.34

gradient inversion attacks, and our method can be further improved in several directions. **1.** For large models, our current approach is to hash the gradients into a lower-dimensional space to reduce memory cost. It may be possible to leverage model architectures to design more effective dimensionality reduction techniques to further scale up the method. **2.** Currently we only focus on the setting with batch size 4 for vision tasks and batch size 1 for language tasks. In practice, the batch size could be larger. For LTI, the complexity of MLP would increase when the batch size increases, which makes learning harder. More advanced model architectures and loss designs may help with the large batch case. **3.** LTI in its current form does not leverage additional data priors such as image smoothness or text fluency. We can readily incorporate these priors by modifying the inversion model's loss function with total variation (for image data) or perplexity on a trained language model (for text data), which may further improve the performance of LTI.

Acknowledgements

RW, XC and KQW are supported by grants from the National Science Foundation NSF (IIS-2107161, III-1526012, IIS-1149882, and IIS-1724282), and the Cornell Center for Materials Research with funding from the NSF MRSEC program (DMR-1719875), and SAP America. RW is also supported by LinkedIn PhD Award.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

Alham Fikri Aji and Kenneth Heafield. Sparse communi-

cation for distributed gradient descent. *arXiv preprint arXiv:1704.05021*, 2017.

Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signsgd: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, pages 560–569. PMLR, 2018.

Christopher M Bishop et al. *Neural networks for pattern recognition*. Oxford university press, 1995.

Franziska Boenisch, Adam Dziedzic, Roei Schuster, Ali Shahin Shamsabadi, Ilia Shumailov, and Nicolas Papernot. When the curious abandon honesty: Federated learning is not private. *arXiv preprint arXiv:2112.02918*, 2021.

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for federated learning on user-held data. *arXiv preprint arXiv:1611.04482*, 2016.

Jieren Deng, Yijue Wang, Ji Li, Chao Shang, Hang Liu, Sanguthevar Rajasekaran, and Caiwen Ding. Tag: Gradient attack on transformer-based language models. *arXiv preprint arXiv:2103.06819*, 2021.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

Liam Fowl, Jonas Geiping, Wojtek Czaja, Micah Goldblum, and Tom Goldstein. Robbing the fed: Directly obtaining private data in federated learning with modified models. *arXiv preprint arXiv:2110.13057*, 2021.

Liam Fowl, Jonas Geiping, Steven Reich, Yuxin Wen, Wojtek Czaja, Micah Goldblum, and Tom Goldstein. Decepticons: Corrupted transformers breach privacy in federated learning for language models. *arXiv preprint arXiv:2201.12675*, 2022.

Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients-how easy is it to break privacy in federated learning? *Advances in Neural Information Processing Systems*, 33:16937–16947, 2020.

Samyak Gupta, Yangsibo Huang, Zexuan Zhong, Tianyu Gao, Kai Li, and Danqi Chen. Recovering private text in federated learning of language models. *arXiv preprint arXiv:2205.08514*, 2022.

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

Jinwoo Jeon, Kangwook Lee, Sewoong Oh, Jungseul Ok, et al. Gradient inversion with generative image prior. *Advances in Neural Information Processing Systems*, 34: 29898–29908, 2021.

Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3): 793–826, 2011.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016.

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

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune bert for text classification? In *China national conference on Chinese computational linguistics*, pages 194–206. Springer, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*, 2018.

Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg. Feature hashing for large scale multitask learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 1113–1120, 2009.

Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M Alvarez, Jan Kautz, and Pavlo Molchanov. See through gradients: Image batch recovery via gradinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16337–16346, 2021.

Yan Zhang, Jonathon Hare, and Adam Prugel-Bennett. Deep set prediction networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. idlg: Improved deep leakage from gradients. *arXiv preprint arXiv:2001.02610*, 2020.

Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. *Advances in Neural Information Processing Systems*, 32, 2019.