Foreseeing Privacy Threats from Gradient Inversion Through the Lens of Angular Lipschitz Smoothness

Anonymous Author(s)
Affiliation
Address
email

Abstract

Recent works proposed server-side *input recovery attacks* in federated learning (FL), in which an honest-but-curious server can recover clients' data (e.g., images) using shared model gradients, thus raising doubts regarding the safety of FL. However, the attack methods are typically demonstrated on only a few models or focus heavily on the reconstruction of a single image, which is easier than that of a batch (multiple images). Thus, in this study, we systematically re-evaluated state-of-the-art (SOTA) attack methods on a variety of models in the context of batch reconstruction. For a broad spectrum of models, we considered two types of model variations: *implicit* (i.e., *without any change in architecture*) and *explicit* (i.e., *with architectural changes*). Motivated by the re-evaluation results that the quality of reconstructed image batch differs per model, we propose *angular Lipschitz constant of a model gradient function with respect to an input* as a measure that explains the vulnerability of a model against input recovery attacks. The prototype of the proposed measure is derived from our theorem on the convergence of attackers' gradient matching optimization, and re-designed into the *scale-invariant* form to prevent trivial server-side loss scaling trick. We demonstrated the predictability of the proposed measure on the vulnerability under recovery attacks by empirically showing its strong monotonic correlation with not only loss drop during gradient-matching optimization but also the quality of the reconstructed image batch. We expect our measure to be a key factor for developing client-side defensive strategies against privacy threats in our proposed realistic FL setting called *black-box* setting, where the server deliberately conceals global model information from clients excluding model gradients.

1 Introduction

Federated learning (FL) is a cooperative machine learning between clients as local trainers and a central server as a global aggregator [14, 21]. Participants in FL cannot access raw data from others and only communicate with one another through gradients, which were believed to leak little information of the original data in the past.

However, recent studies [31, 30, 6, 26, 12] challenge *inverting* gradients back to original data, suggesting that there is potential for an honest-but-curious server to attack by sneakily recovering clients' data from gradients in FL. Their algorithms, so-called gradient inversion attacks, aim at optimizing input variables (e.g., images) to match the given gradients under the condition of fixed model weights. For better reconstruction quality, state-of-the-art (SOTA) attacks assume that both batch normalization (BN) [11] layers' statistics and private labels are known [6, 26, 12, 8]. However,

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

they are demonstrated on a limited range of global models. Thus, we systematically re-evaluated SOTA gradient inversion attacks on a variety of models in the context of *batch (or multiple images) reconstruction*, the recovery of input batch from the averaged gradients over itself, which is more difficult to solve than single image reconstruction, the recovery of single image from its gradient. In this paper, two kinds of model variations are considered, namely *implicit* and *explicit*.

Implicit model variations refer to a collection of different models with the same architecture. In this paper, we consider two types of implicit model variations: *BN modes* and *training epochs*.

- As mentioned previously, SOTA gradient inversion attack methods are demonstrated on models with BN layers to assume shared BN statistics. Note that there are two modes of a BN layer, namely, *train mode* and *eval mode*. In the reality of FL, the server can choose any mode among them. Therefore, we re-evaluated SOTA attacks by considering both modes of BN. This paper is the first to consider BN modes for the evaluation of gradient inversion attacks. We empirically found that the quality of reconstructed batch significantly changes by switching BN modes even for the same model weights.
- By reflecting the reality that clients can encounter global model from the server at any time, we consider models with different training epochs for the re-evaluation. This scheme extends the scope of previous works' training epoch choices of *black-and-white* manner: zero training epoch (untrained) and maximum training epochs (fully trained). We empirically found that the best reconstruction result was usually found at earlier training epochs, not untrained nor fully trained, thus raising the need to expand the evaluation criterion for attack methods.

Meanwhile, explicit model variations are more straightforward than implicit model variations as they only involve architectural changes. In this study, we consider two types of explicit model variations: *skip connections* and *channel size*.

- Residual networks (ResNets) [9] are frequently employed in previous works [26, 6, 31, 12] even for batch reconstruction, while networks *without skip connection* are introduced for only for the recovery of single image from its gradient [6]. Therefore, we explored how a skip connection affects the quality of SOTA gradient inversion attacks in the context of batch reconstruction. Our empirical findings suggest that models without skip connection are more robust against the gradient inversion attack than residual networks.
- The reconstruction quality is known to increase with the number of channels, but this property is demonstrated on single image reconstruction [30, 6]. Thus, we recap how the number of channels affects the attack quality in the context of batch reconstruction.

By re-evaluating SOTA attacks in a variety of models, we found that the vulnerability against gradient inversion attack significantly differs per model, implying the need of more strict evaluation criteria for attack methods. Then, clients are required to judge whether a shared model from the server is safe or not *before sending* locally computed gradients back for their privacy. In this study, we consider two settings on the transparency of global model information to clients: *white-box* and *black-box*. In a white-box setting, clients have an absolute control over global model such as the server; thus, clients can directly apply SOTA attacks to the model to assess its vulnerability.

On the other hand, a *black-box* setting only allows clients control over model gradients to restrict access to the global model possibly due to companies' secrets. For the client-side measurement of privacy leakage in this practical and difficult setting, we propose *angular Lipschitz constant of model gradients with respect to an input* as a predictive measure for the quality of reconstructed samples inverted from model gradients.

This measure is derived from our theorem in Sec. 4 that an attacker's gradient matching loss function drops more abruptly with a smaller L in a particular range, where L is Lipschitz constant of model gradients with respect to an input. However, using L as a measure for privacy leakage would be inappropriate as L can be any nonnegative value by loss function scaling. Therefore, inspired by scale-invariant cosine similarity loss function, we propose the angular Lipschitz constant, a *loss scaling-invariant* alternative to L. We experimentally found that both measure motonically correlates

with not only total loss drop during an attacker's optimization but also the reconstruction quality than the norm of gradients. These findings are expected to support the construction of client-side defense algorithms particularly for *black-box* setting, where only model gradients are given to clients as minimal information of the model as described in Fig. 5.

2 Prior Art in the Gradient Inversion Attack

Given the neural network function $f_w : \mathbb{R}^{b\times d} \rightarrow \mathbb{R}^{b\times c}$ (w, b, d, c being the model weights, batch size, image size, and the number of classes, respectively), and the gradient $g^* = \frac{\partial \mathcal{L}(f_w(x^*),y^*)}{\partial w}$ computed with ground truth input batch $(x^*, y^*) \in \mathbb{R}^{b\times d} \times \mathbb{R}^b$ (x^*, y^* being the image batch, and corresponding label batch) and the loss function $\mathcal{L} : \mathbb{R}^{b\times c} \times \mathbb{R}^b \rightarrow \mathbb{R}$ (e.g., cross-entropy loss), the goal of gradient inversion attack is to reconstruct an image batch $x \in \mathbb{R}^{b\times d}$, a resemblance of ground truth image batch x^*. In the context of federated learning (FL), f_w is the global model, and g^* is the gradient computed from a client. Then, a honest-but-curious server aims to recover the client's private data x^*.

A general method to tackle the problem of inverting gradients is to solve an optimization problem formulated as follows:

$$\arg\min_{x,y} \mathcal{L}_{grad}(\frac{\partial \mathcal{L}(f_w(x), y)}{\partial w}, \frac{\partial \mathcal{L}(f_w(x^*), y^*)}{\partial w}) + \alpha_{prior}\mathcal{R}_{prior}(x), \tag{1}$$

where $\mathcal{L}_{grad} : \mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}$ (N is the size of weights w) is the loss function for gradient matching (which closes the distance between current gradients and target gradients), $\mathcal{R}_{prior} : \mathbb{R}^{b\times d} \rightarrow \mathbb{R}$ is the regularization loss for image prior, with α_{prior} being its coefficient.

Prior to the advent of packages for automatic differentiation, the gradient term $g = \frac{\partial \mathcal{L}(f_w(x),y)}{\partial w}$ was computed as a function of (x, y) in a closed form. For the computation to be tractable, $\mathcal{L}_{grad}$ was set to a squared loss ($\mathcal{L}(g, g^*) = ||g - g^*||_2^2$), and f_w was also slightly modified from the original design of contemporary neural networks. For example, ReLU activation functions were replaced with Sigmoid, and all the strides in convolution modules were excluded from the original ResNet in [31]. Consequently, the choice of f_w was limited.

Currently, with the advantages of automatic differentiation [22] and advanced deep learning optimization algorithms [13, 23, 5], solving for optimization problem in (1) becomes tractable for most contemporary deep neural networks without the need for modification. Further, the gradient matching loss is selected in a broad range from cosine similarity loss ($\mathcal{L}(g, g^*) = 1 - \frac{<g,g^*>}{||g||||g^*||}$) [6, 12, 10, 26] to L2 loss ($\mathcal{L}(g, g^*) = ||g - g^*||_2^2$) [31, 29, 26]. The liberation from the limited choice of loss functions and neural network architectures became the trigger of state-of-the-art attack methods.

State-of-the-art attack methods provide several assumptions which enable the baseline, which is only gradient-based, to be expanded.

First, the server is supposed to know the private labels of clients' images. Currently, estimating x and y becomes a sequential process, in which y is estimated first, after which x is approximated with the estimated $y = y^*_{approx}$ given. Rather than jointly learning x and y in (1), prior works suggest estimating y directly by seeing the gradients from ground truth data g^* before optimization [26, 29]. Therefore, the problem of estimating labels from gradients is separated from the original optimization problem in (1) [3, 25, 16] and some works, which focus on reconstruction of images rather than labels, assume that private labels are known [6, 12].

Second, the local batch statistics $\{\mu_l(x^*; w), {\sigma_l}^2(x^*; w)\}_{l=1}^M$ ($\mu_l(x^*; w)$, $\sigma_l(x^*; w)$, and M being the batch mean of the l^{th} batch normalization layer, batch standard deviation of the l^{th} batch normalization (BN) layer, and number of the BN layers, respectively), computed with client's data batch, is given to the server. This assumption reflects a naive approach of a FL algorithm called FedAvg [21] on the global model with BN layers [19, 17]. When $\{\mu_l(x^*; w), \sigma_l(x^*; w)^2\}_{l=1}^M$ is shared from a client to the server for the update of population statistics in the global model's BN layers, the server as an honest-but-curious adversary would work to add up the batch statistics matching loss term to (1) to ensure a stronger attack.

Then, optimization problem in (1) can be rewritten by considering both assumptions mentioned previously as follows:

$$\arg\min_{x} \mathcal{L}_{grad}\left(\frac{\partial \mathcal{L}(f_w(x), y^*)}{\partial w}, \frac{\partial \mathcal{L}(f_w(x^*), y^*)}{\partial w}\right) + \alpha_{prior}\mathcal{R}_{prior}(x) + \alpha_{BN}\sum_{l=1}^{M}\mathcal{R}_{BN}((\mu_l, {\sigma_l}^2), (\mu_l^*, {\sigma_l^*}^2)) \tag{2}$$

, where $\mathcal{R}_{BN}$ is the BN statistics matching loss and α_{BN} being its coefficient with $(\mu_l, \sigma_l) = (\mu_l(x; w), {\sigma_l}^2(x; w))$ and $(\mu_l^*, \sigma_l^*) = (\mu_l(x^*; w), {\sigma_l}^2(x^*; w))$.

By solving the optimization problem in (2), high resolution images (e.g. ImageNet [4]) with a batch size of up to 40 can be constructed in [26]. However, f_w is only considered for three models: ImageNet pre-trained ResNet18 model, ImageNet pre-trained ResNet50 model, and MOCO v2 [2] pre-trained ResNet50 model fine-tuned with ImageNet. However, there are various choices of f_w. Although a broad spectrum of f_w choices is introduced in [6] (e.g., increasing channel size, models with or without skip connection), the authors of the work verified the effect from model variations on single image reconstruction as well as considered the optimization problem of the form (1) rather than (2). Thus, in this paper, we recap how model variations considered in [6] affect reconstruction of multiple images in a batch by solving optimization problem of the form (2) to achieve a better quality of reconstructed samples.

3 Re-evaluation of SOTA Gradient Attacks on a Broad Spectrum of Models

Prior works in gradient inversion attacks properly select limited range of models with vulnerability under the proposed attack methods to demonstrate their effectiveness [26, 12, 30, 6]. Therefore, this study aims to re-evaluate state-of-the-art attack methods on a broad spectrum of models. The target of our evaluation is attack methods that can solve the optimization problem of the form (2) assuming that the server as an honest-but-curious attacker desires to reconstruct multiple private images from batch gradients given, which is rarely studied previously. The model variations we considered are twofold: *implicit* and *explicit*.

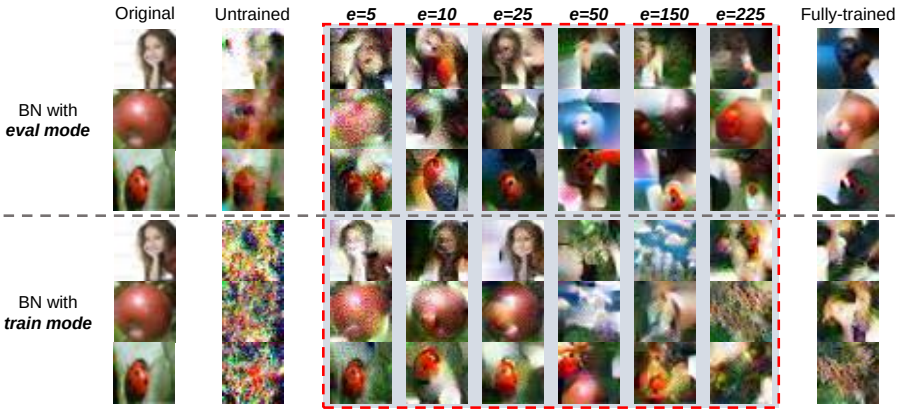


Figure 1: **Visualization of reconstructed images from implicit model variations of ResNet18.** Here e denotes training epochs. Then, "Untrained" means $e = 0$, and "Fully-trained" means $e = 300$ as the ResNet18 model is trained on CIFAR100 training set up to 300 epochs. Reconstructed images in the red dotted line box come from our choices of e. Original images (a woman image, an apple image, a beetle image) were randomly sampled from the CIFAR100 validation set.

3.1 Implicit model variation: BN modes and training epochs

While *explicit model variation* refers to an architectural change such as increasing channel sizes of the model, as suggested in [6], *implicit model variation* is invisible in the architectural level. However, changes arise internally within the same architecture such as applying different weights with different training epochs or switching the mode of normalization layers (e.g., switching between train and eval modes for BN). This is the first work to introduce the concept of implicit model variation. More

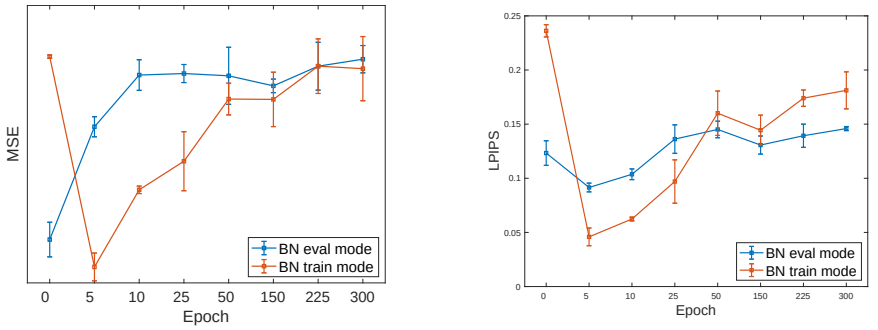


Figure 2: **Plotting the quality of reconstructed samples from implicit model variations of ResNet18 in terms of MSE (↓, left) and LPIPS (↓, right).**

specifically, this is the first time implicit model variation is considered for the evaluation of gradient inversion attacks. Interestingly, we experimentally found that the reconstruction quality ranges in a broad spectrum over *implicit model variations*.

3.1.1 BN modes: motivation

State-of-the-art gradient inversion attack methods elevate the quality of reconstructed samples by introducing batch statistics matching loss to the original problem of gradient matching as in (2). Therefore, we adopt a global model with BN layers to realize shared batch statistics in FL. BN layer has two modes of operation: train mode and eval mode [11]. However, recent works have not specified which mode is set for their demonstration while the malicious server, at least as an honest-but-curious attacker, can send a global model with BN layers set to any mode. Therefore, this study considers both BN train mode and BN eval mode for the re-evaluation of SOTA gradient attacks. Our re-evaluation results show that reconstruction results from different BN modes can be significantly different from each other even in terms of the same model weights as in Tab. 9.

Epoch (e)	MSE ↓	PSNR ↑	LPIPS ↓
0	**0.8499 ± 0.1996**	**12.8833 ± 1.241**	0.1233 ± 0.0227
5	1.5033 ± 0.1157	10.4366 ± 0.43	**0.0915 ± 0.0081**
10	1.7985 ± 0.1766	9.87 ± 0.2749	0.1037 ± 0.0099
25	1.8072 ± 0.1042	10.02 ± 0.5716	0.1362 ± 0.0263
50	1.7941 ± 0.3291	9.8666 ± 0.5507	0.1451 ± 0.0153
150	1.7361 ± 0.0783	10.34 ± 0.3732	0.1307 ± 0.0167
225	1.8495 ± 0.2759	10.1866 ± 0.4878	0.1393 ± 0.0214
300	1.8899 ± 0.1575	9.75 ± 0.4313	0.1459 ± 0.0038

(a) BN with eval mode

Epoch (e)	MSE ↓	PSNR ↑	LPIPS ↓
0	1.9045 ± 0.0195	8.9265 ± 0.0665	0.2362 ± 0.0113
5	**0.6921 ± 0.1601**	**14.9733 ± 0.9168**	**0.0459 ± 0.0164**
10	1.1367 ± 0.0434	12.5 ± 0.2861	0.0624 ± 0.0035
25	1.3015 ± 0.3402	11.6733 ± 1.4027	0.097 ± 0.04
50	1.66 ± 0.1831	10.0433 ± 0.6354	0.1601 ± 0.041
150	1.6581 ± 0.3138	10.2066 ± 1.1074	0.1444 ± 0.0279
225	1.8497 ± 0.315	9.49 ± 1.008	0.174 ± 0.0151
300	1.8353 ± 0.3703	9.4333 ± 0.8832	0.1812 ± 0.0342

(b) BN with train mode

Table 1: Quantitative comparison between reconstruction results for 50 CIFAR100 images from ResNet18 model with BN set to (a) eval mode and (b) train mode. MSE (↓), PSNR (↑), and LPIPS (↓) are used as evaluation metrics. We highlight the best performance for each column in **bold**.

3.1.2 Training epochs: motivation

In a scenario of FL, a client can participate at any time during training. Then, a client can encounter the global model with arbitrary performance. This fact contradicts previous works' experimental setup, where the global model is chosen in a dichotomous manner: an untrained (or initialized) model or a model fully trained on the training set [6, 26]. Therefore, we re-evaluated SOTA inversion attacks on models with a broad spectrum of training epochs. We empirically found that the best reconstruction quality is usually obtained at earlier training epochs.

3.1.3 BN modes and training epochs: experimental results

Setup We trained a ResNet18 model on CIFAR100 [15] training set for 300 epochs using SGD optimizer with initial learning rate 0.1, momentum 0.9, and learning rate decay 0.1 applied when

$e = 150$ and $e = 225$ for the training epoch e. During training, we saved checkpoints of model weights when $e \in \{0, 5, 10, 25, 50, 150, 225, 300\}$ to consider the models from different training epochs. We oversampled model weights before the first learning decay ($0 < e < 150$) to cover the whole set of dynamically changing model weights in the beginning of training. On the other hand, hyperparameters and loss function choices for input reconstruction attacks are borrowed from [10].

Results As expected from their difference in batch statistics computation, BN with train mode and BN with eval mode show different reconstruction results both qualitatively (see Fig. 1.) and quantitatively (see Fig. 2 and Tab. 9.). When BN is set to eval mode, partial information (e.g. colors or shapes) is barely leaked in reconstructed images only for the cases $e = 0$ and $e = 5$ as described in Fig. 1 and Fig. 2. On the other hand, for BN with train mode, the quality of reconstructed images were sufficient enough to identify the object in each image only for the cases $e = 5, 10, 25$. Unlike the BN mode set to eval mode, it is remarkable that reconstructed images from BN with train mode in Fig. 1 are noisy images for $e = 0$. For the cases $e \geq 50$, input reconstruction failed for both BN modes and reconstructed images even from the same target gradients look significantly different for different BN modes. However, both BN with train mode and BN with eval mode have similar reconstruction quality in terms of both mean squared error (MSE) and Learned Perceptual Image Patch Similarity (LPIPS) [28] in Fig. 2 and Tab. 9. Therefore, *in the early stage of training, a global model would be privacy threatening with high probability.*

3.2 Explicit model variation: skip connection and channel size

Explicit model variations involve *change in architecture level* like removing skip connections in residual blocks or increasing the number of channels in convolution module, which are the kinds considered in previous works but on single image reconstruction. Therefore, we re-explore the effect of skip connection and channel size on the model's vulnerability against gradient inversion attack but in the context of batch reconstruction. Skip connection helps information flow both forward and backward through the network, thus input reconstruction is expected to be easier for residual networks but harder for models without skip connection [9]. On the other hand, increasing channel size implies increasing dimension of gradients, which is the capacity of gradients to store information. Therefore, we expect that more information about input would be compressed in gradients when the number of channels increase.

Figure 3: **Visualization of reconstructed images from ConvNet on CIFAR100.** In each image block, the images at the positions of the red, pink, and green borders denote the original image, the reconstruction with BN (*eval mode*), and the reconstruction with BN (*train mode*), respectively. Original images were randomly sampled from CIFAR100 validation set.

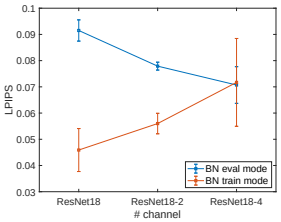


Figure 4: **Plotting the best reconstruction quality in terms of LPIPS (↓) among model variations through training epochs for ResNet18, ResNet18-2, ResNet18-4 models with BN *eval* (orange) and *train* (blue) modes.** $e = 5$ or $e = 10$ usually result in the best reconstruction quality. As channel size increases, the reconstruction quality increases for BN *eval* but decreases for BN *train*.

3.2.1 Skip connection and channel size: experimental results

Setup Instead of ResNet18, we trained a ConvNet model, ResNet18-2 model, and ResNet18-4 model for explicit model variations. ConvNet, which is introduced in [6] for the first time, is a convolutional neural network without skip connection and ResNet18-2 and ResNet18-4 being ResNet with channel size doubled and quadrupled, respectively. Note that we apply implicit variations considered in the Sec. 3.1 to the models. Training conditions and hyperparameters for both model training and attack methods are kept the same with the setup in the previous section.

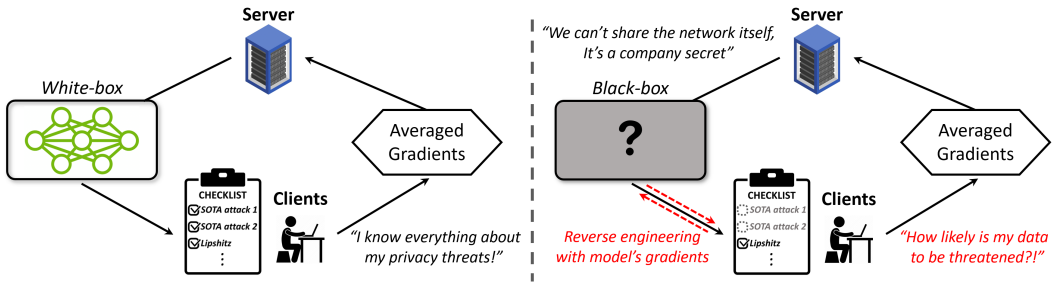
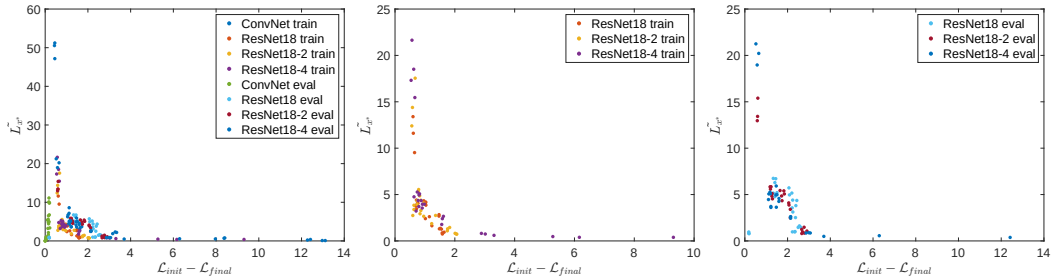


Figure 5: **White-box (left) and black-box (right) FL settings.**



(a) All models ($r_s = -0.78$) (b) ResNets, BN *train* ($r_s = -0.87$) (c) ResNets, BN *eval* ($r_s = -0.66$)

Figure 6: **Proposed measure $\tilde{L_{x^*}}$ is approximately a monotonic decreasing function with respect to $\mathcal{L}_{init} - \mathcal{L}_{final}$, the difference between initial ($\mathcal{L}_{init}$) and final losses ($\mathcal{L}_{final}$) among (a) all models, (b) ResNet models with BN *train* mode, and (c) ResNet models with BN *eval* mode considered in Sec. 3.**

Results Reconstructed images from ConvNet models with the best quality, in terms of LPIPS, are listed in Fig. 3. For ConvNet models, reconstructed images, even with the best quality, are far from original images visually due to severe artifacts. Therefore, as expected from the role of skip connection in residual networks, a network without skip connection like ConvNet seems to be robust against input recovery attacks. Then, ConvNet models would be considered as global model candidates for privacy protection in FL despite of their worse performance than that of residual networks.

By contrast, the best averaged reconstruction results among the sampled training epochs $e \in \{0, 5, 10, 25, 50, 150, 225, 300\}$ are plotted in Fig. 4 for ResNet18, ResNet18-2, and ResNet18-4 models with varied BN modes. When BN is set to the eval mode, the reconstruction quality increases as the number of channels increases as expected. However, the reconstruction quality worsens as the number of channels increases for BN set to the train mode, which breaks the belief from previous works that increasing channel size makes input recovery attack easier [6, 30]. However, the reconstruction quality obtained with BN train mode is better than that with BN eval mode for all models considered except ResNet18-4, where their LPIPS range overlaps, implying that BN train mode is vulnerable against input recovery attacks than BN eval mode. The quantitative results for ConvNet, ResNet18-2, and ResNet18-4 are provided in Appendix A1.

4 Lipschitz Smoothness for Client-Side Privacy Leakage Detection

For privacy-preserving FL, choosing global model robust against any server-side input reconstruction attack method would be important. At the least, global model should be robust against well-known SOTA gradient inversion attack methods to alleviate clients' anxiety about any potential leakage from gradient sharing with the server. If clients can access the global model with the same level of a central server (*white-box*), applying SOTA attack methods directly to the global model with private images would be the best way for assessing whether or not the global model presents a risk to the client's privacy. However, in general, global model information would be opaque to clients due to company

secrets. As clients should communicate with the server via locally computed gradients, we suppose the *black-box* setting, where model gradients are given to clients as minimal information of the global model. Therefore, we provide a helpful measure for developing the system for clients to examine whether the given global model is safe in terms of privacy by using gradients computed with their self-controlled inputs. Note that *white-box* and *black-box* are described in Fig. 5.

4.1 Angular Lipschitz smoothness: motivation

If a function $f : \mathbb{R}^n \to \mathbb{R}^m$ is Lipschitz smooth (or the derivative of f is Lipschitz continuous) with constant L, then the following holds: $||\nabla f(x) - \nabla f(y)|| \leq L||x - y|| \; \forall x, y \in \mathbb{R}^n$. The concept of Lipschitz smoothness or Lipschitz continuity is frequently employed to prove convergence theorem of gradient descent methods for optimization [24, 7, 27, 20, 18, 1]. This study employs the concept of Lipschitz smoothness to prove the following theorem in the context of gradient matching problem.

Theorem 1. (Monotonic decreasing loss function). Suppose $\nabla_w \mathcal{L}(f(x), y)$ is Lipschitz continuous with respect to x with constant L and $\mathcal{L}^x_{grad} = ||\nabla_w \mathcal{L}(f(x), y) - g^*||^2_2$ is given as a gradient matching loss. Then, when gradient descent $\triangle x$ is applied with step size $\mu = \frac{1}{2L^2} > 0$ and $L > \epsilon$ for some $\epsilon > 0$, the following holds:

$$\mathcal{L}^{x+\triangle x}_{grad} \leq \mathcal{L}^x_{grad} - \frac{1}{4L^2}||\frac{\partial \mathcal{L}^x_{grad}}{\partial x}||^2_2. \tag{3}$$

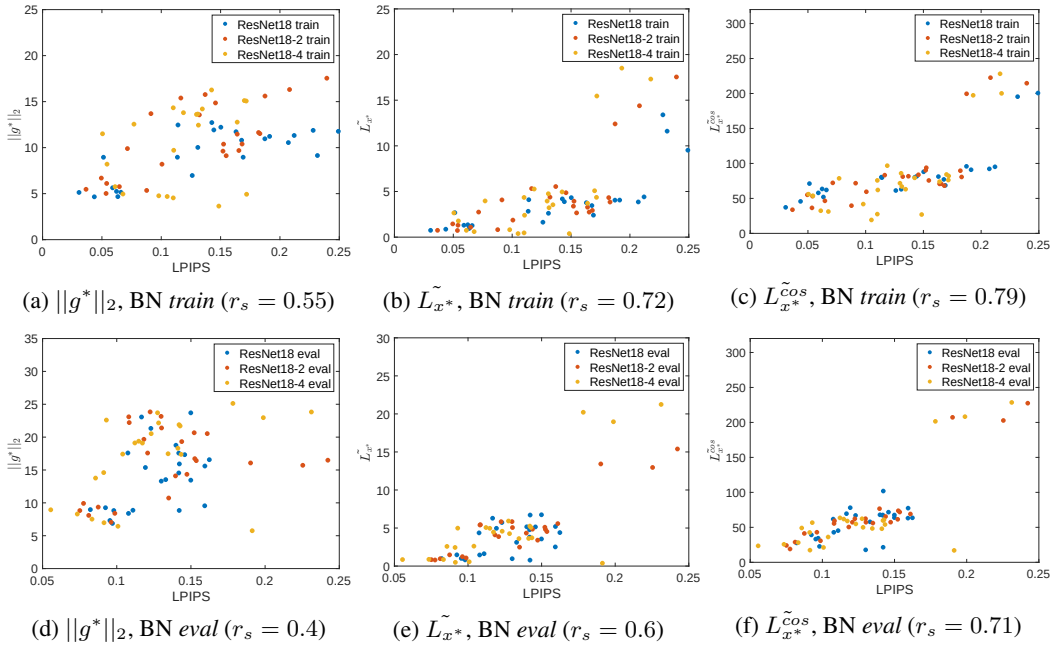
Inequality (3) implies that gradient matching loss strictly decreases as the gradient descent steps unless the gradient term $\frac{\partial \mathcal{L}^x_{grad}}{\partial x}$ is zero (i.e. gradient matching loss already converges). Furthermore, a gradient descent with a small L (or large $\frac{1}{L^2}$) can accelerate the convergence of gradient matching optimization but with the premise that $L > \epsilon$ for $\epsilon > 0$. This premise is required to ensure the first-order Taylor approximation for $\nabla_w \mathcal{L}(f(x + \triangle x), y)$ in the proof in Appendix A2. Therefore, in a particular range of L (i.e., $L > \epsilon$), we hypothesize that a global model with smaller L experiences a sharper loss drop in gradient matching optimization. *We empirically found that L is not too small for most models, thus meeting the premise in reality*.

For the empirical verification of the hypothesis in the context of input reconstruction, we desire to compute Lipschitz smoothness constant locally around x^*, $L_{x^*,\epsilon} = \sup_{||x-x^*||<\epsilon, x\neq x^*} \frac{||\nabla_w \mathcal{L}(f(x),y) - \nabla_w \mathcal{L}(f(x^*),y)||}{||x-x^*||}$, with small ϵ, for the models considered in Sections 3.1 and 3.2. Recent works on computing precise upper bound of L only focus on multi-layer perceptrons (MLP) due to the difficulty of computing L for normalization layers or residual layers. Therefore, $L_{x^*,\epsilon}$ is estimated as $\tilde{L_{x^*}} = \max_{n\neq \mathbf{0}} \frac{||\nabla_w \mathcal{L}(f(x^*+n),y) - \nabla_w \mathcal{L}(f(x^*),y)||}{||n||}$ by sampling 1,000 noises (n) from the Gaussian distribution $\mathcal{N}(0, 0.001^2)$ in our experiments.

However, $\tilde{L_{x^*}}$ can be any nonnegative value by scaling loss function $\mathcal{L}$. If $\mathcal{L}$ is scaled by nonnegative scalar k, then $\tilde{L_{x^*}}$ is scaled by k too, allowing $\tilde{L_{x^*}}$ to be manipulated by the server using simple loss scaling. Therefore, inspired by the cosine similarity loss function, which is scale-invariant, we propose *the angular Lipschitz constant* $\tilde{L^{cos}_{x^*}} = \max_{n\neq \mathbf{0}} \frac{1-cs(\nabla_w \mathcal{L}(f(x^*+n),y), \nabla_w \mathcal{L}(f(x^*),y))}{1-cs(x^*, x^*+n)}$ (cs being the cosine similarity loss) as a loss scaling-invariant alternative to $\tilde{L_{x^*}}$. We find hat $\tilde{L^{cos}_{x^*}}$ shows a strong monotonic correlation with the quality of reconstructed samples, demonstrating the potential of $\tilde{L^{cos}_{x^*}}$ to be imperative for client-side defense methods.

4.2 Angular Lipschitz smoothness: experimental results

We computed $\tilde{L_{x^*}}$ and the attacker's loss drop $\mathcal{L}_{init} - \mathcal{L}_{final}$ ($\mathcal{L}_{init}$ and $\mathcal{L}_{final}$ being the initial and final losses, respectively) for the models and input batches considered in Sec. 3 (Fig. 22a). We also quantified their correlation using the Spearman's rank correlation coefficient r_s, which quantifies how two variables are in a monotonic relationship. $r_s = 1$ ($r_s = -1$) means that one variable is a completely monotonic increasing (decreasing) function with respect to the other one. Then, $\tilde{L_{x^*}}$ is almost a monotonic decreasing function with respect to $\mathcal{L}_{init} - \mathcal{L}_{final}$ with $r_s = -0.78$, thus validating our hypothesis. For ResNet models with BN *train* (Fig. 22b), $\tilde{L_{x^*}}$ and $\mathcal{L}_{init} - \mathcal{L}_{final}$ show a stronger monotonic correlation than that for ResNet models with BN *eval* (Fig. 22c) with



(a) $||g^*||_2$, BN *train* ($r_s = 0.55$) (b) $\tilde{L_{x^*}}$, BN *train* ($r_s = 0.72$) (c) $\tilde{L^{cos}_{x^*}}$, BN *train* ($r_s = 0.79$)

(d) $||g^*||_2$, BN *eval* ($r_s = 0.4$) (e) $\tilde{L_{x^*}}$, BN *eval* ($r_s = 0.6$) (f) $\tilde{L^{cos}_{x^*}}$, BN *eval* ($r_s = 0.71$)

Figure 7: **Comparison of $||g^*||_2$, $\tilde{L_{x^*}}$, and $\tilde{L^{cos}_{x^*}}$ in terms of the correlation between LPIPS of reconstructed samples for ResNet models with BN *train* (top) and BN *eval* (bottom)**

$r_s = -0.85$. As in Tab. 9 and Fig. 1, reconstructed samples are closer to their original images in BN *train mode*, thus $\tilde{L_{x^*}}$, which is computed around the ground truth x^*, seems to fit more to BN *train* while L should be estimated around the solution from the attack method rather than x^* for the case of BN *eval*. However, clients cannot access to the solution from the the attack method in the *black-box* setting. The plot of $\tilde{L_{x^*}}$ and $\mathcal{L}_{init} - \mathcal{L}_{final}$ for the ConvNet models is provided in Appendix A3.

5 Limitations and Future Work

Our hypothesis can be extended to the correlation between Lipschitz constant and the quality of reconstructed samples, rather than loss drop. Zero gradient matching loss does not mean complete recovery of original images due to the existence of *twin data* [30], two different data input with identical model gradients. However, we empirically found that both $\tilde{L_{x^*}}$ and $\tilde{L^{cos}_{x^*}}$ show positive monotonic correlations with the quality of reconstructed samples, in terms of LPIPS (lower value is better) (Fig. 7). In particular, they beat the baseline measure, the norm of given gradients ($||g^*||_2$), which was implicitly believed to be the amount of information within the gradients in previous works, by a wide margin, in terms of r_s. Therefore, we expect $\tilde{L_{x^*}}$ and $\tilde{L^{cos}_{x^*}}$ to be the key factors for developing future client-side defense strategies.

6 Conclusions

Here, we re-evaluated the SOTA attack method on a broad spectrum of models in the context of batch reconstruction, which is rarely studied in previous works. We considered model variations of two types: *implicit*, which changes in model weights or BN modes within the same architecture, and *explicit*, with changes in architecture. The re-evaluation results indicate that the quality of the reconstruction attack varies depending on the implicit or explicit model changes. Therefore, inspired by our theorem related to the convergence of gradient matching optimization and scale-invariance of the cosine similarity loss function, we propose an explainable and predictive measure for privacy leakage, an angular Lipschitz constant L^{cos}, which is invariant to trivial loss scaling attacks from malicious servers. We empirically find that L^{cos} shows a strong monotonic correlation with the quality of reconstructed samples, thus expecting the potential of L^{cos} to be a key factor for clients' defense strategies in a *black-box* setting, where only model gradients are given as minimal information about the global model to clients.

References

[1] H. H. Bauschke, J. Bolte, and M. Teboulle. A descent lemma beyond lipschitz gradient continuity: first-order methods revisited and applications. *Mathematics of Operations Research*, 42(2):330–348, 2017.

[2] X. Chen, H. Fan, R. Girshick, and K. He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[3] T. Dang, O. Thakkar, S. Ramaswamy, R. Mathews, P. Chin, and F. Beaufays. Revealing and protecting labels in distributed training. *Advances in Neural Information Processing Systems*, 34, 2021.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[5] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

[6] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller. Inverting gradients-how easy is it to break privacy in federated learning? *Advances in Neural Information Processing Systems*, 33:16937–16947, 2020.

[7] H. Gouk, E. Frank, B. Pfahringer, and M. J. Cree. Regularisation of neural networks by enforcing lipschitz continuity. *Machine Learning*, 110(2):393–416, 2021.

[8] A. Hatamizadeh, H. Yin, P. Molchanov, A. Myronenko, W. Li, P. Dogra, A. Feng, M. G. Flores, J. Kautz, D. Xu, et al. Do gradient inversion attacks make federated learning unsafe? *arXiv preprint arXiv:2202.06924*, 2022.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] Y. Huang, S. Gupta, Z. Song, K. Li, and S. Arora. Evaluating gradient inversion attacks and defenses in federated learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[11] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[12] J. Jeon, K. Lee, S. Oh, J. Ok, et al. Gradient inversion with generative image prior. *Advances in Neural Information Processing Systems*, 34:29898–29908, 2021.

[13] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015.

[14] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[15] A. Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[16] O. Li, J. Sun, X. Yang, W. Gao, H. Zhang, J. Xie, V. Smith, and C. Wang. Label leakage and protection in two-party split learning. 2022.

[17] Q. Li, Y. Diao, Q. Chen, and B. He. Federated learning on non-iid data silos: An experimental study. 2021.

[18] X. Li and F. Orabona. On the convergence of stochastic gradient descent with adaptive stepsizes. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 983–992. PMLR, 2019.

[19] X. Li, M. JIANG, X. Zhang, M. Kamp, and Q. Dou. Fedbn: Federated learning on non-iid features via local batch normalization. In *International Conference on Learning Representations*, 2020.

[20] V. V. Mai and M. Johansson. Stability and convergence of stochastic gradient clipping: Beyond lipschitz continuity and smoothness. In *International Conference on Machine Learning*, pages 7325–7335. PMLR, 2021.

[21] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[22] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017.

[23] S. Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[24] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry. How does batch normalization help optimization? *Advances in neural information processing systems*, 31, 2018.

[25] D. Ye, T. Zhu, S. Zhou, B. Liu, and W. Zhou. Label-only model inversion attack: The attack that requires the least information. *arXiv preprint arXiv:2203.06555*, 2022.

[26] H. Yin, A. Mallya, A. Vahdat, J. M. Alvarez, J. Kautz, and P. 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.

[27] J. Zeng, T. T.-K. Lau, S. Lin, and Y. Yao. Global convergence of block coordinate descent in deep learning. In *International conference on machine learning*, pages 7313–7323. PMLR, 2019.

[28] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[29] B. Zhao, K. R. Mopuri, and H. Bilen. idlg: Improved deep leakage from gradients. *arXiv preprint arXiv:2001.02610*, 2020.

[30] J. Zhu and M. B. Blaschko. R-gap: Recursive gradient attack on privacy. In *International Conference on Learning Representations*, 2020.

[31] L. Zhu, Z. Liu, and S. Han. Deep leakage from gradients. *Advances in Neural Information Processing Systems*, 32, 2019.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] We thoroughly listed the contributions of our work in both abstract and introduction.
   (b) Did you describe the limitations of your work? [Yes] See Sec. 5 for limitations of our work and future research direction.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] In Theorem 1, all the assumptions are included in the statement. The details are described in Appendix A2.

(b) Did you include complete proofs of all theoretical results? [Yes] We include complete proofs in Appendix A2.

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The details for experiments are partially described in Results section in Sec. 3 including Appendix.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] All the training details for experiments are described in Results section in Sec. 3.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We report standard deviation of our results in Tab. 9 and Fig. 2.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We report them in Appendix A4.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We cited the creators for CIFAR100, ImageNet, ConvNet, and ResNet18.
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] We use data widely used in the field of research related to our work
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Appendix

A1 Quantitative Results for Experiments in Sec 3

Epoch (e)	MSE ↓	PSNR ↑	LPIPS ↓
0	1.8965 ± 0.1382	9.2833 ± 0.3121	0.2901 ± 0.0179
5	2.1649 ± 0.0848	8.3266 ± 0.1929	0.1688 ± 0.0186
10	2.031 ± 0.017	8.62 ± 0.0264	0.1582 ± 0.044
25	1.8544 ± 0.171	9.2333 ± 0.6753	**0.1359 ± 0.017**
50	1.9994 ± 0.1924	9.1366 ± 0.1106	0.1513 ± 0.0236
150	1.8748 ± 0.2704	9.5066 ± 0.3957	0.1587 ± 0.0071
225	1.6861 ± 0.2622	10.1333 ± 0.6735	0.1469 ± 0.0086
300	**1.5614 ± 0.1468**	**10.3866 ± 0.5901**	0.1458 ± 0.0072

(a) BN with eval mode

Epoch (e)	MSE ↓	PSNR ↑	LPIPS ↓
0	1.6316 ± 0.0592	9.5933 ± 0.068	0.2008 ± 0.0253
5	2.1189 ± 0.2271	8.46 ± 0.5179	0.1642 ± 0.0372
10	1.8955 ± 0.0588	8.95 ± 0.1928	0.145 ± 0.0296
25	**1.397 ± 0.0289**	**11.4133 ± 0.24**	**0.0944 ± 0.0136**
50	1.5079 ± 0.1185	11.1133 ± 0.4005	0.1295 ± 0.037
150	1.8097 ± 0.1404	10.13 ± 0.2816	0.1522 ± 0.0384
225	1.6551 ± 0.1594	10.34 ± 0.6295	0.1454 ± 0.0081
300	1.5598 ± 0.0409	10.5666 ± 0.467	0.1396 ± 0.0056

(b) BN with train mode

Table 2: Quantitative comparison between reconstruction results for 50 CIFAR100 images from ConvNet model with BN set to (a) eval mode and (b) train mode. MSE (↓), PSNR (↑), and LPIPS (↓) are used as evaluation metrics. We highlight the best performance for each column in **bold**.

Epoch (e)	MSE ↓	PSNR ↑	LPIPS ↓
0	1.6762 ± 0.0929	9.5333 ± 0.1607	0.2193 ± 0.0265
5	**1.4252 ± 0.0947**	10.8566 ± 0.5804	**0.0779 ± 0.003**
10	1.56 ± 0.1788	10.2066 ± 0.3253	0.0941 ± 0.0058
25	1.8473 ± 0.1458	10.1333 ± 0.8367	0.1343 ± 0.0132
50	1.6196 ± 0.0972	**11.1133 ± 0.4523**	0.1188 ± 0.011
150	1.6408 ± 0.2033	10.9233 ± 1.3664	0.1306 ± 0.0193
225	1.74 ± 0.1739	10.4233 ± 0.8991	0.1452 ± 0.0132
300	1.8682 ± 0.1614	10.2166 ± 1.0142	0.1453 ± 0.0202

(a) BN with eval mode

Epoch (e)	MSE ↓	PSNR ↑	LPIPS ↓
0	1.8183 ± 0.0549	9.25 ± 0.1479	0.2116 ± 0.0262
5	**0.8339 ± 0.1301**	**14.32 ± 0.4611**	0.0592 ± 0.026
10	1.0023 ± 0.2293	13.14 ± 1.0834	**0.056 ± 0.0078**
25	1.4296 ± 0.2193	11.2933 ± 0.62	0.1089 ± 0.0422
50	1.5804 ± 0.4373	10.97 ± 1.8166	0.1411 ± 0.0432
150	1.6429 ± 0.4505	10.35 ± 1.432	0.1434 ± 0.0103
225	1.6529 ± 0.2715	10.1466 ± 1.3	0.1505 ± 0.0333
300	1.7069 ± 0.2039	10.1633 ± 1.1359	0.1614 ± 0.0227

(b) BN with train mode

Table 3: Quantitative comparison between reconstruction results for 50 CIFAR100 images from ResNet18-2 model with BN set to (a) eval mode and (b) train mode. MSE (↓), PSNR (↑), and LPIPS (↓) are used as evaluation metrics. We highlight the best performance for each column in **bold**.

Epoch (e)	MSE ↓	PSNR ↑	LPIPS ↓
0	1.7153 ± 0.2172	9.4366 ± 0.6727	0.2027 ± 0.0266
5	1.5792 ± 0.2495	10.0766 ± 0.7902	0.1278 ± 0.0551
10	**1.1437 ± 0.1481**	**12.26 ± 0.8729**	**0.0707 ± 0.014**
25	1.458 ± 0.212	12.0933 ± 0.6028	0.0936 ± 0.0093
50	1.4843 ± 0.1045	12.2066 ± 0.5685	0.1148 ± 0.019
150	1.6002 ± 0.3407	11.05 ± 1.3352	0.1278 ± 0.0132
225	1.5096 ± 0.2374	10.9733 ± 0.8712	0.1298 ± 0.0157
300	1.6437 ± 0.1456	10.7066 ± 0.79	0.1342 ± 0.0146

(a) BN with eval mode

Epoch (e)	MSE ↓	PSNR ↑	LPIPS ↓
0	1.5248 ± 0.1184	10.0066 ± 0.5387	0.209 ± 0.0138
5	1.5976 ± 0.0279	10.3033 ± 0.67	0.1212 ± 0.0237
10	**1.1173 ± 0.1554**	12.6366 ± 1.1192	0.0755 ± 0.0196
25	1.119 ± 0.3125	**12.8 ± 1.3452**	**0.0717 ± 0.0335**
50	1.4036 ± 0.1523	11.5366 ± 0.6013	0.1362 ± 0.0269
150	1.6145 ± 0.0317	10.6033 ± 0.352	0.1479 ± 0.0214
225	1.4926 ± 0.2147	10.6633 ± 0.8298	0.144 ± 0.0225
300	1.3546 ± 0.1617	11.1466 ± 0.3066	0.1225 ± 0.0475

(b) BN with train mode

Table 4: Quantitative comparison between reconstruction results for 50 CIFAR100 images from ResNet18-4 model with BN set to (a) eval mode and (b) train mode. MSE (↓), PSNR (↑), and LPIPS (↓) are used as evaluation metrics. We highlight the best performance for each column in **bold**.

A2 Proof of Theorem 1

Theorem. (Monotonic decreasing loss function). Suppose $\nabla_w \mathcal{L}(f(x), y)$ is Lipschitz continuous with respect to x with constant L and $\mathcal{L}^x_{grad} = ||\nabla_w \mathcal{L}(f(x), y) - g^*||_2^2$ is given as a gradient matching loss. Then, when gradient descent $\triangle x$ is applied with step size $\mu > 0$ and $L > \epsilon$ for some $\epsilon > 0$, the following holds:

$$\mathcal{L}^{x+\triangle x}_{grad} \leq \mathcal{L}^x_{grad} - \frac{1}{4L^2}||\frac{\partial \mathcal{L}^x_{grad}}{\partial x}||_2^2.$$

Proof. First, we will compute the vector for gradient descent, $\triangle x = -\mu \frac{\partial \mathcal{L}_{grad}}{\partial x}$ by chain rule as follows:

$\triangle x = -\mu \frac{\partial \mathcal{L}_{grad}}{\partial x} = -\mu \frac{\partial ||\nabla_w \mathcal{L}(f(x),y) - g^*||_2^2}{\partial x} = -2\mu \nabla_x \nabla_w \mathcal{L}(f(x),y)(\nabla_w \mathcal{L}(f(x),y) - g^*)$.

Then, $\mathcal{L}_{grad}^{x+\triangle x}$ can be separated into three terms by summation like the following:

$$\begin{aligned}\mathcal{L}_{grad}^{x+\triangle x} &= ||\nabla_w \mathcal{L}(f(x+\triangle x),y) - g^*||_2^2 \\ &= ||\nabla_w \mathcal{L}(f(x+\triangle x),y) - \nabla_w \mathcal{L}(f(x),y) + \nabla_w \mathcal{L}(f(x),y) - g^*||_2^2 \\ &= ||u||_2^2 + 2u^T v + ||v||_2^2\end{aligned}$$

, where $u = \nabla_w \mathcal{L}(f(x+\triangle x),y) - \nabla_w \mathcal{L}(f(x),y)$ and $v = \nabla_w \mathcal{L}(f(x),y) - g^*$.

For the first term, $||u||_2^2 = ||\nabla_w \mathcal{L}(f(x+\triangle x),y) - \nabla_w \mathcal{L}(f(x),y)||_2^2 \leq L^2 ||\triangle x||^2 = L^2\mu^2 ||\frac{\partial \mathcal{L}_{grad}}{\partial x}||_2^2$ due to the L-Lipschitz continuity condition of $\nabla_w \mathcal{L}(f(x),y)$ with respect to an input.

For the second term, $2u^T v = 2(\nabla_w \mathcal{L}(f(x+\triangle x),y) - \nabla_w \mathcal{L}(f(x),y))^T (\nabla_w \mathcal{L}(f(x),y) - g^*)$
$\approx 2\triangle x^T \nabla_x \nabla_w \mathcal{L}(f(x),y)(\nabla_w \mathcal{L}(f(x),y) - g^*)(\because$ Taylor's first-order approximation with $\mu < \delta)$
$= -\mu(\frac{\partial \mathcal{L}_{grad}}{\partial x})^T(\frac{\partial \mathcal{L}_{grad}}{\partial x}) = -\mu||\frac{\partial \mathcal{L}_{grad}}{\partial x}||_2^2(\because$ Recall that how $\triangle x$ is computed in the first step).

For the third term, $||v||_2^2 = ||\nabla_w \mathcal{L}(f(x),y) - g^*||_2^2 = \mathcal{L}_{grad}^x$.

Then, summing up the three terms considered above will lead to the following inequality:

$$\begin{aligned}\mathcal{L}_{grad}^{x+\triangle x} &\leq \mathcal{L}_{grad}^x + (L^2\mu^2 - \mu)||\frac{\partial \mathcal{L}_{grad}}{\partial x}||_2^2 \\ &= \mathcal{L}_{grad}^x - \frac{1}{4L^2}||\frac{\partial \mathcal{L}_{grad}}{\partial x}||_2^2(\because \text{minimized at } \mu = \frac{1}{2L^2}).\end{aligned}$$

For both $\mu = \frac{1}{2L^2}$ and $\mu < \delta$ to be met, $\frac{1}{2L^2} < \delta$ should be satisfied and this is why the premise that $L > \epsilon = \frac{1}{\sqrt{2\delta}}$ is required for the theorem. We empirically found that $\tilde{L}$, the estimated value for L, turned out to be not small, thus meeting the premise in reality. We will derive the theorem for small L to explain outliers in future work.

□

A3 The correlation between $\tilde{\mathcal{L}}_{x^*}$ and $\mathcal{L}_{init} - \mathcal{L}_{final}$

A4 GPU Information

We used TITAN X GPUs and TITAN V GPUs for all our experiments.

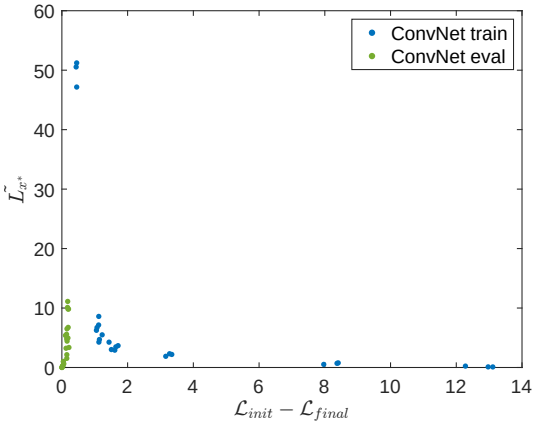


Figure 8: **Proposed measure $\tilde{L_{x^*}}$ is a monotonic decreasing function with respect to $\mathcal{L}_{init} - \mathcal{L}_{final}$, the difference between initial ($\mathcal{L}_{init}$) and final losses ($\mathcal{L}_{final}$) among ConvNet models with BN *train* while there is no monotonic correlation between them for ConvNet models with BN *eval*.** We attribute the absence of monotonic correlation among ConvNet models with BN *eval* to their reconstructed samples, which are far from ground-truth, as $\tilde{L_{x^*}}$ is computed locally around ground truth x^*.

Epoch (e)	MSE ↓	LPIPS (VGG) ↓	$L_{x^*}^{\tilde{cos}}$ ↓
0	**0.5719 ± 0.0652**	0.5691 ± 0.0175	**20.9183 ± 3.2255**
5	1.3485 ± 0.0655	**0.4852 ± 0.0143**	27.0683 ± 5.5657
10	1.6405 ± 0.0571	0.5248 ± 0.041	46.0818 ± 2.7603
25	1.7825 ± 0.1158	0.5738 ± 0.0254	57.4093 ± 5.4613
50	1.7479 ± 0.1583	0.5639 ± 0.0303	64.0587 ± 10.9156
150	1.688 ± 0.1404	0.5427 ± 0.0436	71.5169 ± 10.1408
225	1.7872 ± 0.087	0.5724 ± 0.0383	81.7608 ± 5.6223
300	1.806 ± 0.0877	0.5798 ± 0.44	76.3331 ± 3.1129

(a) BN with eval mode

Epoch (e)	MSE ↓	LPIPS (VGG) ↓	$L_{x^*}^{\tilde{cos}}$ ↓
0	1.948 ± 0.1515	0.7326 ± 0.0127	199.1924 ± 17.68
5	**0.7951 ± 0.08**	**0.3547 ± 0.0247**	**41.12 ± 3.4926**
10	1.0326 ± 0.1589	0.391 ± 0.0313	56.7727 ± 5.3159
25	1.757 ± 0.0843	0.611 ± 0.0687	68.1433 ± 7.4578
50	1.6903 ± 0.263	0.6305 ± 0.1033	82.2794 ± 6.7557
150	1.7062 ± 0.064	0.6322 ± 0.0466	79.3744 ± 5.6671
225	1.4934 ± 0.1529	0.5953 ± 0.0537	82.9744 ± 10.5295
300	1.4084 ± 0.1174	0.5408 ± 0.0467	86.2595 ± 8.7986

(b) BN with train mode

Table 5: Quantitative comparison between reconstruction results for 64 CIFAR100 images (4 batches, 16 images per batch) from ResNet18 model with BN set to (a) eval mode and (b) train mode. MSE (↓) and LPIPS (↓) are used as evaluation metrics. Our proposed measure, $L_{x^*}^{\tilde{cos}}$ (↓), is also reported. We highlight the best performance for each column in **bold**.

Epoch (e)	MSE ↓	LPIPS (VGG) ↓	$L_{x^*}^{\tilde{cos}}$ ↓
0	**0.3156 ± 0.0825**	**0.4502 ± 0.043**	**6.2153 ± 0.0825**
5	1.693 ± 0.1596	0.6523 ± 0.0108	28.7715 ± 9.821
10	1.7111 ± 0.1043	0.6122 ± 0.0116	65.3085 ± 37.3436
25	1.8316 ± 0.1657	0.5998 ± 0.0183	62.1694 ± 10.6263
50	1.7095 ± 0.086	0.5898 ± 0.0211	99.9578 ± 5.3933
150	1.4483 ± 0.0746	0.536 ± 0.0325	99.2213 ± 11.0362
225	1.4582 ± 0.1248	0.542 ± 0.0458	97.4186 ± 13.8522
300	1.7293 ± 0.1651	0.5862 ± 0.027	101.3323 ± 14.381

(a) BN with eval mode

Epoch (e)	MSE ↓	LPIPS (VGG) ↓	$L_{x^*}^{\tilde{cos}}$ ↓
0	1.4859 ± 0.1365	0.7046 ± 0.0142	194.9052 ± 9.3291
5	1.4703 ± 0.1091	0.6701 ± 0.0271	57.2143 ± 68.38
10	1.1802 ± 0.0651	0.5867 ± 0.0255	**58.6197 ± 17.6834**
25	1.3613 ± 0.0947	0.524 ± 0.032	97.5486 ± 19.549
50	1.6709 ± 0.112	0.5896 ± 0.0378	114.8647 ± 6.5291
150	1.4543 ± 0.0539	0.5191 ± 0.0741	112.95164 ± 7.3345
225	**1.3505 ± 0.0877**	0.5236 ± 0.0832	115.0326 ± 16.07
300	1.432 ± 0.066	**0.5039 ± 0.0449**	121.9739 ± 13.3709

(b) BN with train mode

Table 6: Quantitative comparison between reconstruction results for 64 CIFAR100 images (4 batches, 16 images per batch) from ConvNet model with BN set to (a) eval mode and (b) train mode. MSE (↓) and LPIPS (↓) are used as evaluation metrics. Our proposed measure, $L_{x^*}^{\tilde{cos}}$ (↓), is also reported. We highlight the best performance for each column in **bold**.

Epoch (e)	MSE ↓	LPIPS (VGG) ↓	$L_{x^*}^{\widetilde{cos}}$ ↓
0	**0.2474 ± 0.0062**	**0.4054 ± 0.0167**	**19.9073 ± 2.3527**
5	1.534 ± 0.1027	0.5183 ± 0.0244	23.8754 ± 1.9732
10	1.5702 ± 0.1286	0.5012 ± 0.0223	37.9359 ± 2.4721
25	1.6132 ± 0.0808	0.5293 ± 0.0389	53.1483 ± 6.4762
50	1.6138 ± 0.1309	0.5732 ± 0.0382	59.1139 ± 2.615
150	1.6 ± 0.0635	0.5313 ± 0.0479	71.7673 ± 3.4096
225	1.7086 ± 0.0565	0.5484 ± 0.0465	78.7679 ± 3.7287
300	1.8235 ± 0.0687	0.5714 ± 0.0416	74.4335 ± 7.5889

(a) BN with eval mode

Epoch (e)	MSE ↓	LPIPS (VGG) ↓	$L_{x^*}^{\widetilde{cos}}$ ↓
0	1.7373 ± 0.1201	0.7237 ± 0.0105	204.3889 ± 11.7899
5	**0.7048 ± 0.149**	**0.3756 ± 0.0221**	**35.9896 ± 4.009**
10	1.0816 ± 0.1969	0.4228 ± 0.1458	48.5864 ± 4.6507
25	1.6825 ± 0.1391	0.6626 ± 0.0363	65.5451 ± 5.354
50	1.7025 ± 7.5396	0.7109 ± 0.0269	74.8524 ± 7.5396
150	1.6581 ± 0.0715	0.6562 ± 0.0203	78.8464 ± 11.376
225	1.419 ± 0.1081	0.5998 ± 0.0707	83.8286 ± 5.8781
300	1.3323 ± 0.0559	0.5404 ± 0.0509	79.4811 ± 4.6781

(b) BN with train mode

Table 7: Quantitative comparison between reconstruction results for 64 CIFAR100 images (4 batches, 16 images per batch) from ResNet18-2 model with BN set to (a) eval mode and (b) train mode. MSE (↓) and LPIPS (↓) are used as evaluation metrics. Our proposed measure, $L_{x^*}^{\widetilde{cos}}$ (↓), is also reported. We highlight the best performance for each column in **bold**.

Epoch (e)	MSE ↓	LPIPS (VGG) ↓	$L_{x^*}^{\widetilde{cos}}$ ↓
0	**0.0613 ± 0.007**	**0.1962 ± 0.021**	21.3437 ± 1.1459
5	0.8951 ± 0.2044	0.4654 ± 0.0658	**15 ± 5.221**
10	1.4287 ± 0.2041	0.4613 ± 0.0142	22.1449 ± 2.7226
25	1.4744 ± 0.0824	0.5176 ± 0.0688	47.8112 ± 4.7598
50	1.5319 ± 0.1064	0.5279 ± 0.0604	52.626 ± 1.9861
150	1.8181 ± 0.1554	0.5717 ± 0.05	67.4925 ± 3.9216
225	1.6169 ± 0.1111	0.5286 ± 0.0357	70.7561 ± 10.5865
300	1.8231 ± 0.1274	0.5423 ± 0.045	67.3849 ± 4.0188

(a) BN with eval mode

Epoch (e)	MSE ↓	LPIPS (VGG) ↓	$L_{x^*}^{\widetilde{cos}}$ ↓
0	1.7054 ± 0.065	0.7258 ± 0.0151	190.876 ± 8.3158
5	**0.6054 ± 0.1936**	**0.4616 ± 0.0806**	**19.4512 ± 3.0999**
10	1.3525 ± 0.9591	0.4865 ± 0.1597	29.3387 ± 3.5174
25	1.6787 ± 0.3834	0.6079 ± 0.0348	58.525 ± 2.2005
50	1.4906 ± 0.073	0.6285 ± 0.0573	66.9798 ± 7.6932
150	1.6027 ± 0.1728	0.6474 ± 0.0609	74.2522 ± 8.0542
225	1.3665 ± 0.099	0.581 ± 0.072	76.8094 ± 10.6026
300	1.1961 ± 0.2369	0.5091 ± 0.0941	76.7829 ± 11.1112

(b) BN with train mode

Table 8: Quantitative comparison between reconstruction results for 64 CIFAR100 images (4 batches, 16 images per batch) from ResNet18-4 model with BN set to (a) eval mode and (b) train mode. MSE (↓) and LPIPS (↓) are used as evaluation metrics. Our proposed measure, $L_{x^*}^{\widetilde{cos}}$ (↓), is also reported. We highlight the best performance for each column in **bold**.

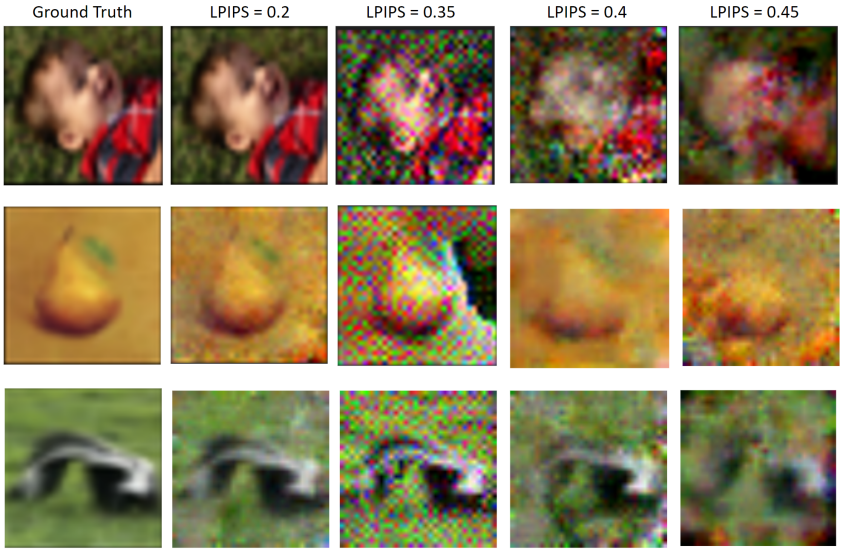


Figure 9: **LPIPS (vgg) of reconstructed images on CIFAR100.** The detailed information seems to be lost when LPIPS exceeds 0.4.

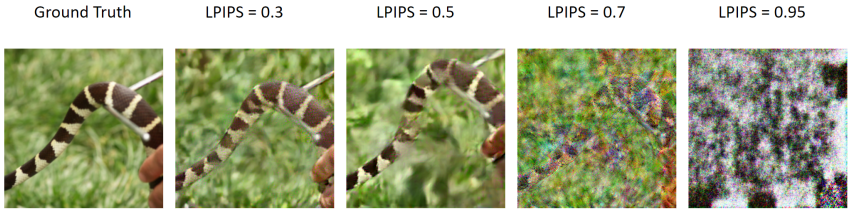
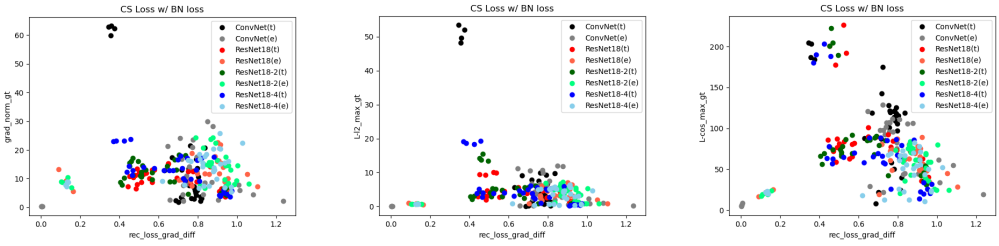
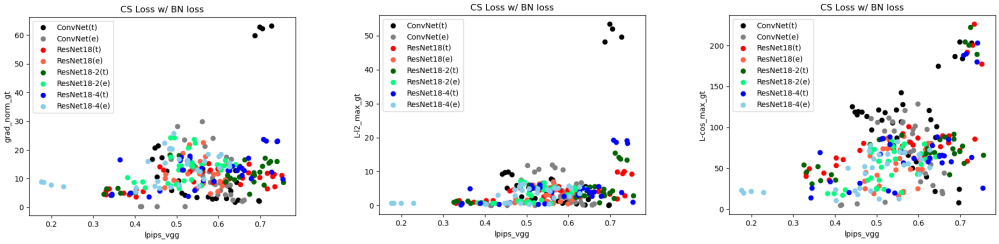


Figure 10: **LPIPS (vgg) of reconstructed images on ImageNet.** The detailed information seems to be lost when LPIPS exceeds 0.7.



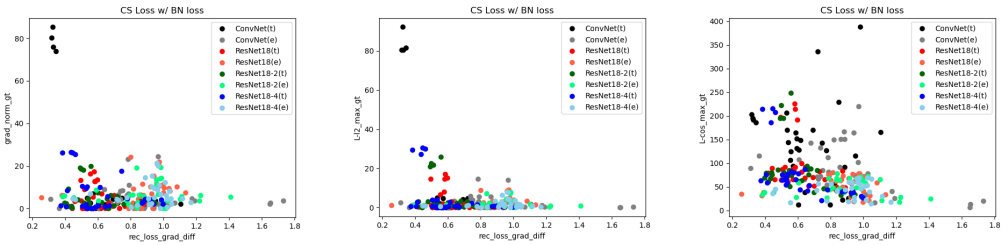
(a) $||g^*||_2$ vs $\mathcal{L}_{grad}^{x_i} - \mathcal{L}_{grad}^{x_f}$ (b) $\tilde{L_{x^*}}$ vs $\mathcal{L}_{grad}^{x_i} - \mathcal{L}_{grad}^{x_f}$ (c) $\tilde{L_{x^*}^{cos}}$ vs $\mathcal{L}_{grad}^{x_i} - \mathcal{L}_{grad}^{x_f}$

Figure 11: **The correlation between gradient loss drop and each of $||g^*||_2$, $\tilde{L_{x^*}}$, and $\tilde{L_{x^*}^{cos}}$ on 64 CIFAR100 images.** x_i means attacker's initialized image and x_f is the final solution by attacker. (t) means BN train mode while (e) means BN eval mode.



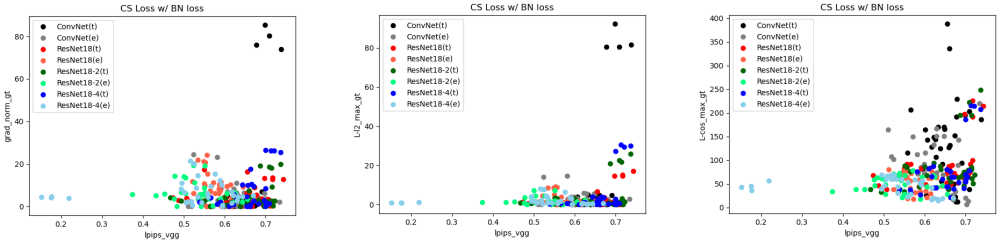
(a) $||g^*||_2$ vs LPIPS(x_i, x_f) (b) $\tilde{L_{x^*}}$ vs LPIPS(x_i, x_f) (c) $\tilde{L_{x^*}^{cos}}$ vs LPIPS(x_i, x_f)

Figure 12: **The correlation between reconstruction quality (LPIPS) and each of $||g^*||_2$, $\tilde{L_{x^*}}$, and $\tilde{L_{x^*}^{cos}}$ on 64 CIFAR100 images.** x_i means attacker's initialized image and x_f is the final solution by attacker. (t) means BN train mode while (e) means BN eval mode.



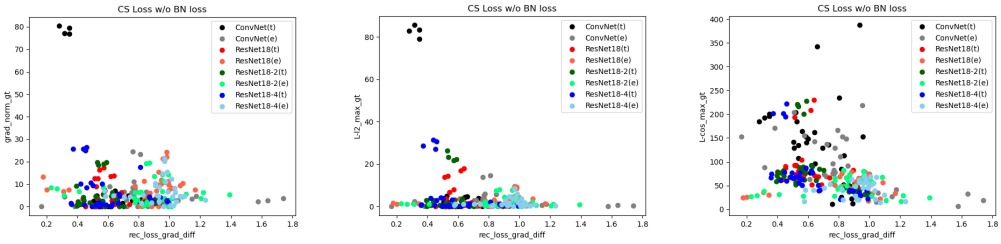
(a) $||g^*||_2$ vs $\mathcal{L}_{grad}^{x_i} - \mathcal{L}_{grad}^{x_f}$ (b) $\tilde{L_{x^*}}$ vs $\mathcal{L}_{grad}^{x_i} - \mathcal{L}_{grad}^{x_f}$ (c) $\tilde{L_{x^*}^{cos}}$ vs $\mathcal{L}_{grad}^{x_i} - \mathcal{L}_{grad}^{x_f}$

Figure 13: **The correlation between gradient loss drop and each of $||g^*||_2$, $\tilde{L_{x^*}}$, and $\tilde{L_{x^*}^{cos}}$ on 40 CIFAR10 images.** x_i means attacker's initialized image and x_f is the final solution by attacker. (t) means BN train mode while (e) means BN eval mode.



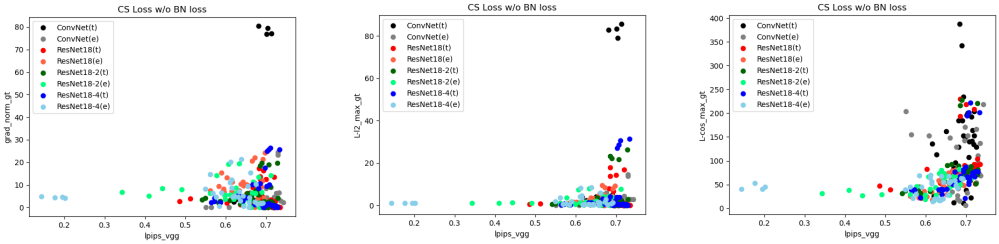
(a) $||g^*||_2$ vs LPIPS(x_i, x_f) (b) $\tilde{L_{x^*}}$ vs LPIPS(x_i, x_f) (c) $\tilde{L_{x^*}^{cos}}$ vs LPIPS(x_i, x_f)

Figure 14: **The correlation between reconstruction quality (LPIPS) and each of $||g^*||_2$, $\tilde{L_{x^*}}$, and $\tilde{L_{x^*}^{cos}}$ on 40 CIFAR10 images.** x_i means attacker's initialized image and x_f is the final solution by attacker. (t) means BN train mode while (e) means BN eval mode.



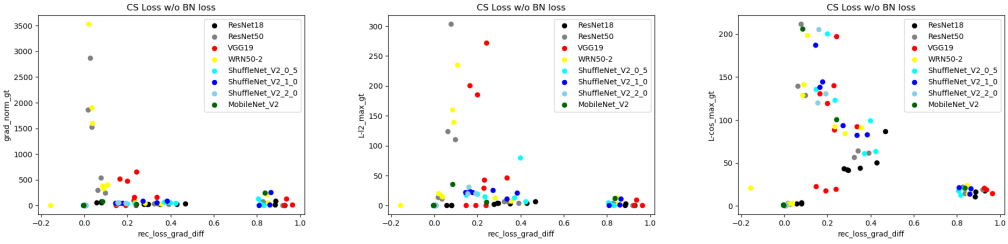
(a) $||g^*||_2$ vs $\mathcal{L}_{grad}^{x_i} - \mathcal{L}_{grad}^{x_f}$ (b) $\tilde{L_{x^*}}$ vs $\mathcal{L}_{grad}^{x_i} - \mathcal{L}_{grad}^{x_f}$ (c) $\tilde{L_{x^*}^{cos}}$ vs $\mathcal{L}_{grad}^{x_i} - \mathcal{L}_{grad}^{x_f}$

Figure 15: **The correlation between gradient loss drop and each of $||g^*||_2$, $\tilde{L_{x^*}}$, and $\tilde{L_{x^*}^{cos}}$ on 40 CIFAR10 images, without BN statistics matching loss.** x_i means attacker's initialized image and x_f is the final solution by attacker. (t) means BN train mode while (e) means BN eval mode.



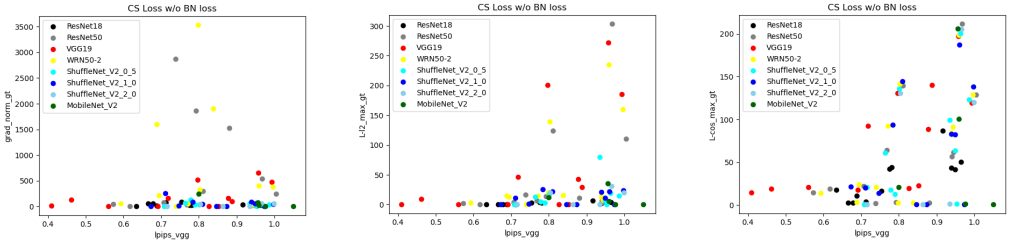
(a) $||g^*||_2$ vs LPIPS(x_i, x_f) (b) $\tilde{L_{x^*}}$ vs LPIPS(x_i, x_f) (c) $\tilde{L_{x^*}^{cos}}$ vs LPIPS(x_i, x_f)

Figure 16: **The correlation between reconstruction quality (LPIPS) and each of $||g^*||_2$, $\tilde{L_{x^*}}$, and $\tilde{L_{x^*}^{cos}}$ on 40 CIFAR10 images, but without BN statistics matching loss.** x_i means attacker's initialized image and x_f is the final solution by attacker. (t) means BN train mode while (e) means BN eval mode.



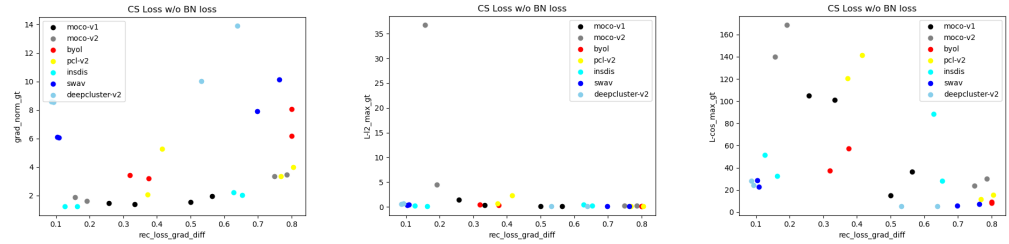
(a) $||g^*||_2$ vs $\mathcal{L}_{grad}^{x_i} - \mathcal{L}_{grad}^{x_f}$ (b) $\tilde{L_{x^*}}$ vs $\mathcal{L}_{grad}^{x_i} - \mathcal{L}_{grad}^{x_f}$ (c) $\tilde{L_{x^*}^{cos}}$ vs $\mathcal{L}_{grad}^{x_i} - \mathcal{L}_{grad}^{x_f}$

Figure 17: **The correlation between gradient loss drop and each of $||g^*||_2$, $\tilde{L_{x^*}}$, and $\tilde{L_{x^*}^{cos}}$ on 3 ImageNet images.** x_i means attacker's initialized image and x_f is the final solution by attacker.



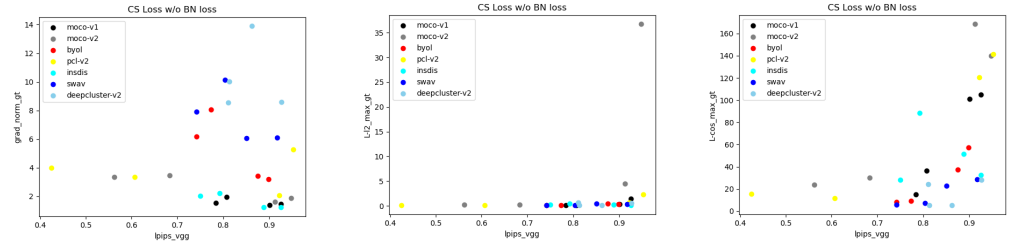
(a) $||g^*||_2$ vs LPIPS(x_i, x_f) (b) $\tilde{L_{x^*}}$ vs LPIPS(x_i, x_f) (c) $\tilde{L_{x^*}^{cos}}$ vs LPIPS(x_i, x_f)

Figure 18: **The correlation between reconstruction quality (LPIPS) and each of $||g^*||_2$, $\tilde{L_{x^*}}$, and $\tilde{L_{x^*}^{cos}}$ on 3 ImageNet images.** x_i means attacker's initialized image and x_f is the final solution by attacker.



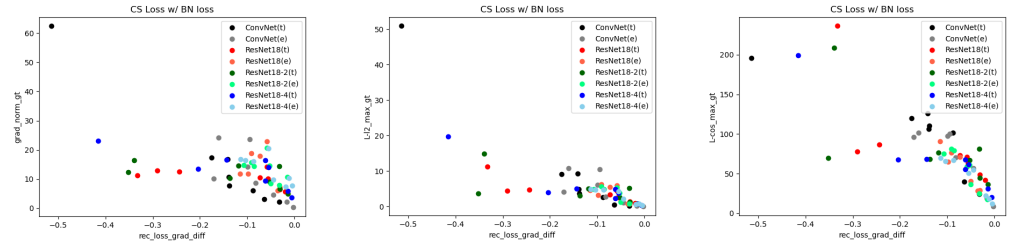
(a) $||g^*||_2$ vs $\mathcal{L}_{grad}^{x_i} - \mathcal{L}_{grad}^{x_f}$ (b) $\tilde{L_{x^*}}$ vs $\mathcal{L}_{grad}^{x_i} - \mathcal{L}_{grad}^{x_f}$ (c) $\tilde{L_{x^*}^{cos}}$ vs $\mathcal{L}_{grad}^{x_i} - \mathcal{L}_{grad}^{x_f}$

Figure 19: **The correlation between gradient loss drop and each of $||g^*||_2$, $\tilde{L_{x^*}}$, and $\tilde{L_{x^*}^{cos}}$ on self-supervised models and 3 ImageNet images.** x_i means attacker's initialized image and x_f is the final solution by attacker.



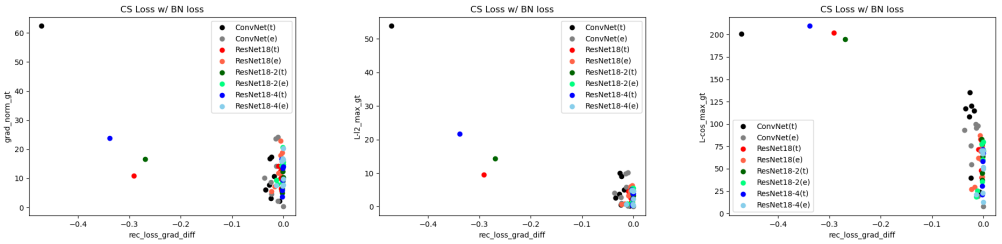
(a) $||g^*||_2$ vs LPIPS(x_i, x_f) (b) $\tilde{L_{x^*}}$ vs LPIPS(x_i, x_f) (c) $\tilde{L_{x^*}^{cos}}$ vs LPIPS(x_i, x_f)

Figure 20: **The correlation between reconstruction quality (LPIPS) and each of $||g^*||_2$, $\tilde{L_{x^*}}$, and $\tilde{L_{x^*}^{cos}}$ on self-supervised models and 3 ImageNet images.** x_i means attacker's initialized image and x_f is the final solution by attacker.



(a) $||g^*||_2$ vs $\mathcal{L}_{grad}^{x_i} - \mathcal{L}_{grad}^{x_f}$ (b) $\tilde{L_{x^*}}$ vs $\mathcal{L}_{grad}^{x_i} - \mathcal{L}_{grad}^{x_f}$ (c) $\tilde{L_{x^*}^{cos}}$ vs $\mathcal{L}_{grad}^{x_i} - \mathcal{L}_{grad}^{x_f}$

Figure 21: **The correlation between gradient loss drop and each of $||g^*||_2$, $\tilde{L_{x^*}}$, and $\tilde{L_{x^*}^{cos}}$ when attacker's image is initialized to near ground-truth and optimizer is SGD with initial learning rate 1e-3.** x_i means attacker's initialized image and x_f is the final solution by attacker.



(a) $||g^*||_2$ vs $\mathcal{L}_{grad}^{x_i} - \mathcal{L}_{grad}^{x_f}$ (b) $\tilde{L_{x^*}}$ vs $\mathcal{L}_{grad}^{x_i} - \mathcal{L}_{grad}^{x_f}$ (c) $\tilde{L_{x^*}^{cos}}$ vs $\mathcal{L}_{grad}^{x_i} - \mathcal{L}_{grad}^{x_f}$

Figure 22: **The correlation between gradient loss drop and each of $||g^*||_2$, $\tilde{L_{x^*}}$, and $\tilde{L_{x^*}^{cos}}$ when attacker's image is initialized to near ground-truth and optimizer is adam with initial learning rate 1e-3.** x_i means attacker's initialized image and x_f is the final solution by attacker.

Epoch (e)	MSE ↓	LPIPS (VGG) ↓	$\tilde{L_{x^*}^{cos}}$ ↓
0	**0.453 ± 0.0042**	0.5326 ± 0.0236	25.3089 ± 1.1828
5	1.4512 ± 0.0372	0.577 ± 0.0714	**18.9778 ± 3.6686**
10	1.5152 ± 0.0979	0.531 ± 0.0079	29.5007 ± 3.6333
25	1.7654 ± 0.1141	0.5708 ± 0.0222	48.2371 ± 5.7139
50	1.6698 ± 0.0836	**0.531 ± 0.0378**	60.6711 ± 13.9046
150	1.6758 ± 0.1248	0.5397 ± 0.0504	73.0855 ± 15.083
225	1.643 ± 0.1164	0.5533 ± 0.0332	80.2256 ± 19.026
300	1.6104 ± 0.455	0.6203 ± 0.0232	67.2345 ± 20.1765

(a) BN with eval mode

Epoch (e)	MSE ↓	LPIPS (VGG) ↓	$\tilde{L_{x^*}^{cos}}$ ↓
0	1.772 ± 0.107	0.7016 ± 0.0148	154.5375 ± 8.1019
5	**1.0375 ± 0.1094**	0.5225 ± 0.0275	**28.31 ± 3.2312**
10	1.1714 ± 0.1072	**0.4998 ± 0.0227**	40.1853 ± 4.653
25	1.4137 ± 0.1374	0.5219 ± 0.0366	62.4392 ± 7.0658
50	1.4419 ± 0.0523	0.5377 ± 0.0347	69.4821 ± 5.0685
150	1.3895 ± 0.2779	0.527 ± 0.0921	78.7187 ± 13.1556
225	1.5137 ± 0.0959	0.6054 ± 0.0308	81.9572 ± 16.688
300	1.5985 ± 0.0734	0.6055 ± 0.0213	83.9432 ± 14.4981

(b) BN with train mode

Table 9: Quantitative comparison between reconstruction results for 64 CIFAR100 images (4 batches, 16 images per batch) from skip connection-removed ResNet18 model with BN set to (a) eval mode and (b) train mode. MSE (↓) and LPIPS (↓) are used as evaluation metrics. Our proposed measure, $\tilde{L_{x^*}^{cos}}$ (↓), is also reported. We highlight the best performance for each column in **bold**.