Appendix

A. Proof for Propositions

A.1. Proof for Proposition 1

Proof: Demonstration of the existence proposition can be accomplished by providing a singular example. Given a batch of "white images $\hat{x}$", we have $\text{TV}(\hat{x}) = 0$, whereas $\text{TV}(x^*) \neq 0$ for most natural images. Therefore, if

$$\lambda > \frac{L_{\text{grad}}\left(\hat{x}, W, \nabla W\right)}{\text{TV}(x^*)},$$

we have $L_{\text{sum}}\left(\hat{x}, W, \nabla W\right) < L_{\text{sum}}\left(x^*, W, \nabla W\right)$. This concludes the proof. □

A.2. Proof for Proposition 2

Proof: Consider a 1-layer neural network of weight $W = [w_1, w_2, \cdots, w_C] \in \mathbb{R}^{D\times C}$.

If only one sample $\hat{x} \in \mathbb{R}^{D\times 1}$ is generated, the linear layer outputs $\hat{a} = w^T\hat{x} = [w_1^T\hat{x}, w_2^T\hat{x}, \cdots, w_C^T\hat{x}]^T \in \mathbb{R}^{C\times 1}$ and the softmax probability is computed as

$$\hat{p}_k = \frac{e^{\hat{a}_k}}{\sum e^{\hat{a}_j}}.$$

Compute gradient w.r.t. each w_k:

1. If $k \neq y$, then

$$\begin{aligned}
\frac{\partial \ell}{\partial w_k} &= \frac{\partial \ell}{\partial \hat{p}_y} \cdot \frac{\partial \hat{p}_y}{\partial \hat{a}_k} \cdot \frac{\partial \hat{a}_k}{\partial w_k} \\
&= \frac{1}{\hat{p}_y} \cdot \frac{e^{\hat{a}_y}}{\left(\sum e^{\hat{a}_j}\right)^2} \cdot e^{\hat{a}_k} \cdot \hat{x} \\
&= \frac{e^{\hat{a}_k}}{\sum e^{\hat{a}_j}} \cdot \hat{x} \\
&= \hat{p}_k \cdot \hat{x}
\end{aligned}$$

2. If $k = y$, then

$$\begin{aligned}
\frac{\partial \ell}{\partial w_y} &= \frac{\partial \ell}{\partial \hat{p}_y} \cdot \frac{\partial \hat{p}_y}{\partial \hat{a}_y} \cdot \frac{\partial \hat{a}_y}{\partial w_y} \\
&= -\frac{1}{\hat{p}_y} \cdot \frac{e^{\hat{a}_y} \cdot \sum e^{\hat{a}_j} - e^{\hat{a}_y} e^{\hat{a}_y}}{\left(\sum e^{\hat{a}_j}\right)^2} \cdot \hat{x} \\
&= -\frac{\sum e^{\hat{a}_j} - e^{\hat{a}_y}}{\sum e^{\hat{a}_j}} \cdot \hat{x} \\
&= (\hat{a}_y - 1) \cdot \hat{x}
\end{aligned}$$

Consider one-hot encoding for $y^* = [y_1, y_2, \cdots, y_C]$, then the above two cases can be unified into $\frac{\partial \ell}{\partial w_k} = (\hat{p}_k - y_k) \cdot \hat{x}$.

For a mini-batch of N samples $\hat{x} = [\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_N]$, note

$$\nabla w_i := \frac{1}{N}\sum_{j=1}^{N} \frac{\partial \ell_j}{\partial w_i} = \frac{1}{N}\sum_{j=1}^{N} (\hat{p}_{j,i} - y_{j,i}) \cdot \hat{x}_j$$

Rearranging the gradient leads us to

$$
\begin{aligned}
N \cdot \nabla W &= \begin{bmatrix} N\nabla w_1 & N\nabla w_2 & \cdots & N\nabla w_C \end{bmatrix} \\
&= \begin{bmatrix} \hat{x}_1 & \hat{x}_2 & \cdots & \hat{x}_N \end{bmatrix} \cdot \begin{bmatrix} \hat{p}_{1,1} - y_{1,1} & \hat{p}_{1,2} - y_{1,2} & \cdots, \hat{p}_{1,C} - y_{1,C} \\ \hat{p}_{2,1} - y_{2,1} & \hat{p}_{2,2} - y_{2,2} & \cdots, \hat{p}_{2,C} - y_{2,C} \\ \cdots & \cdots & \cdots, \cdots \\ \hat{p}_{N,1} - y_{N,1} & \hat{p}_{N,2} - y_{N,2} & \cdots, \hat{p}_{N,C} - y_{N,C} \end{bmatrix} \\
&= \hat{x} \cdot \hat{P}
\end{aligned}
$$

This concludes the proof. □

A.3. Proof for Proposition 3

Proposition 4. *Given a latent vector z_0 and its corresponding generative model space denoted as $\hat{X}_G$, let its intersection with the gradient constraints be defined as*

$$\hat{X}_\Lambda := \hat{X}_G \cap \hat{X}_{grad}.$$

A necessary condition for successful generative gradient inversion is that $\hat{X}_\Lambda \neq \emptyset$. Furthermore, in the case where $\hat{X}_G$ is sufficiently large, such a non-emptiness condition always holds.

Proof: Given the universal approximation theory of generative models [15], it follows that the generative space $\hat{X}_{\mathrm{G}}$ can be arbitrarily vast in the absence of constraints on the model. In such a scenario, the groundtruth images will always lie within this expansive space, namely $x^* \in \hat{X}_{\mathrm{G}}$.

As the groundtruth images also satisfy Eq (2) and (3) with zero loss, it follows that $x^* \in \hat{X}_{\mathrm{grad}}$. This non-emptiness assertion is therefore supported by the following observation:

$$x^* \in \hat{X}_{\mathrm{G}} \cap \hat{X}_{\mathrm{grad}}.$$

This concludes the proof. □

B. Details of CI-Net

B.1. Tailoring Progressive Model

Our justification for customizing the expanding model proposed by Karras et al.[12] is presented as follows. While the generative model is primarily intended to produce images of high-fidelity, we have identified several inadequacies that arise when it is directly applied to the gradient inversion problem. As demonstrated in Figure 9, the utilization of linear interpolation (represented as the blue line) as prescribed in the original publication presents certain limitations when compared with the nearest interpolation method presented in this work. Additionally, it has been observed that the adoption of linear interpolation may lead to algorithmic failure under specific conditions, particularly in scenarios where the parameters fail to satisfy the requisite threshold for CI-Net. Hence, it is essential to tailor the original model to better accommodate the gradient inversion problem.

In order to enhance performance, we have made several specific modifications, as outlined below:

1. The original pixel-norm, which has been demonstrated to impair performance, has been replaced with the spectral normalization technique [21].

2. The original linear-interpolation method, which was found to result in experiment failure in certain scenarios, has been substituted with the nearest-interpolation method to improve pixel similarity.

3. We have replaced the two To-RGB layers with a single direct output layer, for the sake of simplicity. Furthermore, the Resnet-block is no longer employed in our network.

4. The channel number has been set as a hyper-parameter to facilitate over-parameterization.

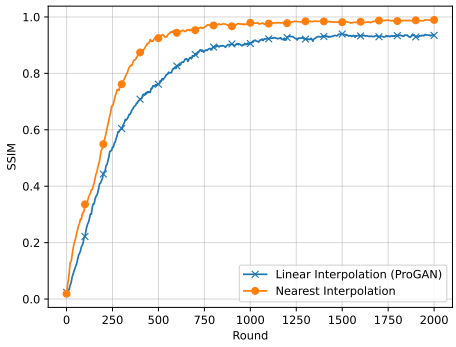

Figure 9: Image reconstruction with different interpolation methods.

5. The ReLU layer has been replaced with the LeakyReLU, which has demonstrated superior performance in our experiments.

B.2. Algorithm

The complete algorithm is depicted in Alg 1. Note the keys steps are: 1) It is necessary for the system to calculate $P(F)$ to enable over-parameterization in subsequent steps; 2) In place of traditional gradient descent, a signed counterpart has been employed..

Algorithm 1 Gradient Inversion with CI-Net

Input: Differential machine learing model F and uploaded gradient ∇W, learning rate η

Set $P(F) = P(\nabla W)$ ▷ Obtain parameter number
Load CI-Net $G(\theta)$ and select a channel number so that $P(G) > P(F)$
Obtain y^* from ∇W
$\theta_0 \leftarrow \mathcal{N}(0,1)$ and $z_0 \leftarrow \mathcal{N}(0,1)$ ▷ Set initial values for CI-Net
for $t \leftarrow 0$ to $T-1$ **do**
    Generate images $\hat{x}_t = G(z_0, \theta_t)$ ▷ Generate fake images
    Obtain gradient $\nabla_W L(\hat{x}_t, y^*)$
    Compute $L_{\text{grad}}\left(\hat{x}, W, \nabla W\right) = \|\nabla_W L(\hat{x}_t, y^*) - \nabla_W L(x^*, y^*)\|^2$
    Compute $\nabla_\theta L_{\text{grad}}$
    $\nabla\theta_t = \text{sign}(\nabla_\theta L_{\text{grad}})$ ▷ Replace gradient with signed gradient
    Update parameters $\theta_{t+1} = \theta_t - \eta\nabla\theta_t$ ▷ Update parameters
end for

Output: Generated Images $\hat{x} = G(z_0, \theta_T)$

B.3. Parameter Setting

As elucidated in the manuscript, it is generally not imperative to explicitly specify a particular z_0. In our experiments, a customary selection for z_0 would entail a stochastic Gaussian vector of length 128. The learning rate η for CI-Net is held at a fixed value of 5×10^{-3}. In regards to all competing algorithms, the learning rate has been configured to adhere to the default value as defined within the authors' source code.

C. CIFAR-10 Additional Experiment

C.1. Batch size=64

For the purpose of validation, we conduct numerical experiments to assess the performance of both GIAS and the proposed method by setting the batch size to 64. Table 4 presents the experimental outcomes by exhibiting four similarity metrics. In contrast to the results obtained with a batch size of 128, GIAS demonstrates acceptable performance in this scenario, attaining a mean SSIM value of 0.87 and a mean LPIPS value of 0.04. These values suggest that this seminal approach is capable of accurately recovering the groundtruth when the batch size is relatively small. Nonetheless, the proposed method introduced in this work still exhibits clear advantages by achieving superior results across all four metrics, culminating in an SSIM value of 1.00, indicating that the reconstructed images are virtually indistinguishable from the groundtruth.

Table 4: Algorithm performance of gradient inversion on 64 CIFAR-10 images.

Algorithm	SSIM↑	FSIM↑	PSNR↑	LPIPS (VGG)↓
GIAS [11]	0.87 ± 0.01	0.94 ± 0.01	16.06 ± 0.38	0.04 ± 0.01
Ours	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{33.72 \pm 0.05}$	$\mathbf{0.01 \pm 0.00}$

C.2. Batch size=128

In Figure 6, we present a visual summary of the 128 images obtained from CI-Net, wherein 12 representative samples are showcased for better visualization. The performance evaluation of the CI-Net is also provided in Table 1.

For completeness, we present all reconstructed images of CI-Net in Figure 10, accompanied by bounding boxes for the first 4 images. As evident from the results, the obtained images exhibit a high degree of similarity with the groundtruth, thereby offering potential for revealing the true data that lie hidden.

C.3. Batch size=256

The performance of the proposed method was further evaluated with a larger batch size of 256, which is the maximum size that the GPU can support. As conventional methods have already failed at a batch size of 128, only the performance of CI-Net after 150k rounds of training was reported.

The results in Table 5 demonstrate that the proposed method can effectively reconstruct groundtruth images even with a larger batch size, highlighting its robustness and high fidelity. These results also address concerns regarding the potential for third-party reconstruction of images from such an averaged gradient obtained from the server. Importantly, our findings suggest that using a large batch size may not necessarily provide a safety guarantee in the context of federated learning. It is worth noting that in federated learning, local training typically does not utilize such a large batch size, but the server may aggregate local gradients into an averaged value, which can serve as a proxy for a large batch of images. Moreover, if the local client employs the Multi-Party Computation (MPC) technique to protect its gradient, the server or any third-party can obtain the averaged gradient of all local images, where the batch size can be very large.

Table 5: Algorithm performance of gradient inversion on 256 CIFAR-10 images.

Algorithm	SSIM↑	FSIM↑	PSNR↑	LPIPS (VGG)↓
Ours	0.98 ± 0.01	0.99 ± 0.01	34.11 ± 0.13	0.02 ± 0.01

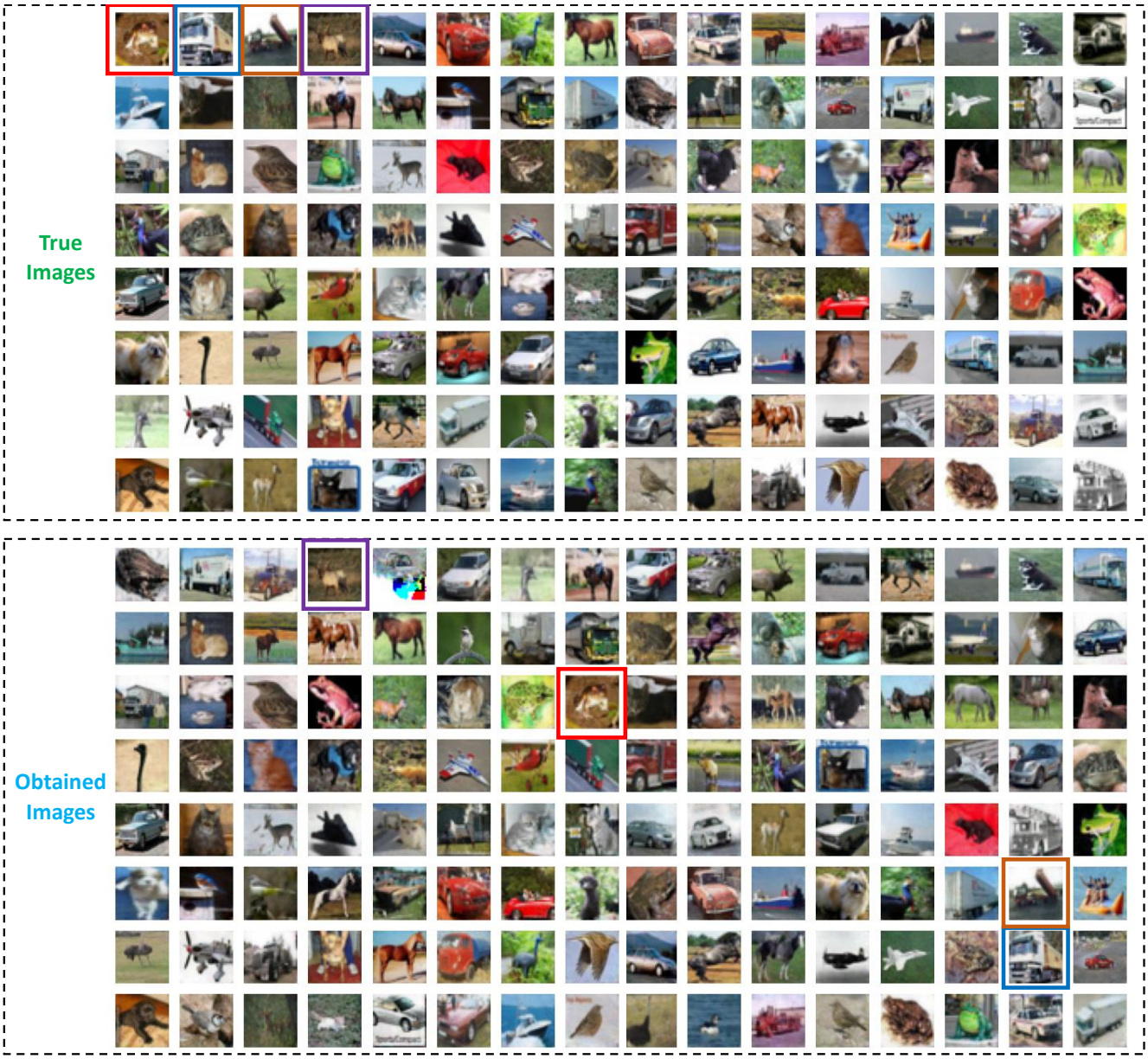

Figure 10: 128 Groundtruth images and the obtained results of CI-Net. Bounding boxes for the first 4 images are plotted for better visualization.

D. ImageNet Additional Experiment

D.1. Batch size=24

We present the complete set of 24 reconstructed images in Figure 11 obtained through the use of CI-Net on the ImageNet dataset. For the purpose of visual comparison, the corresponding groundtruth images are also included on the left-hand side.

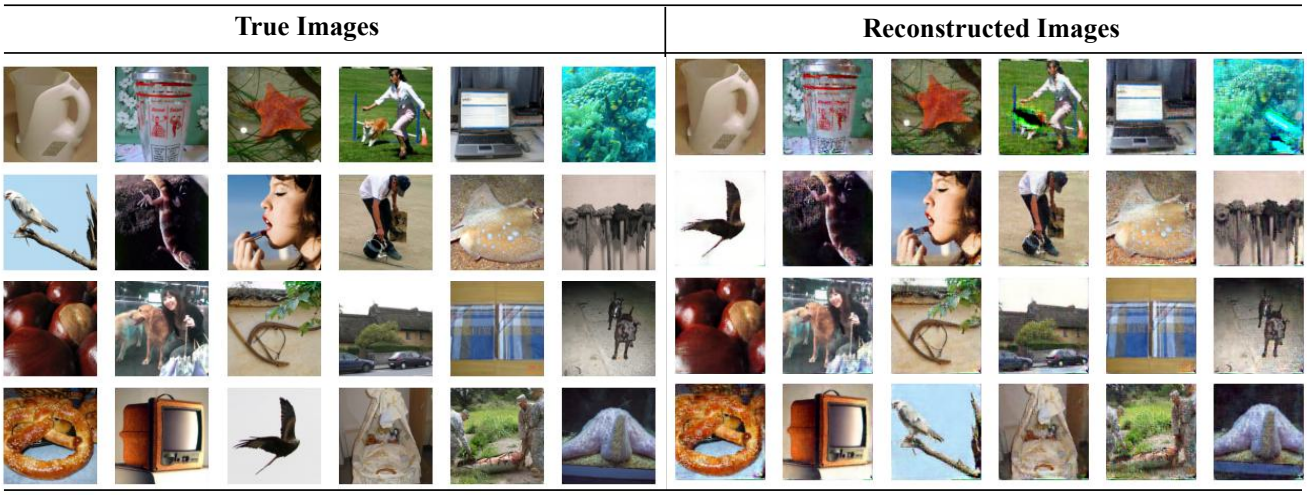

Figure 11: The groundtruth images and reconstructed images from CI-Net. Batch size equals 24.

D.2. Batch size=32

We further expand our experimental analysis to include 32 images on ImageNet, which entails a large batch size and an over-parameterized network, pushing the limits of our hardware device.

The obtained results are consistent with our prior findings when the batch size was 24. Despite some images appearing slightly blurred, potential attackers can still uncover the groundtruth from the reconstructed images, highlighting the vulnerability of the system.

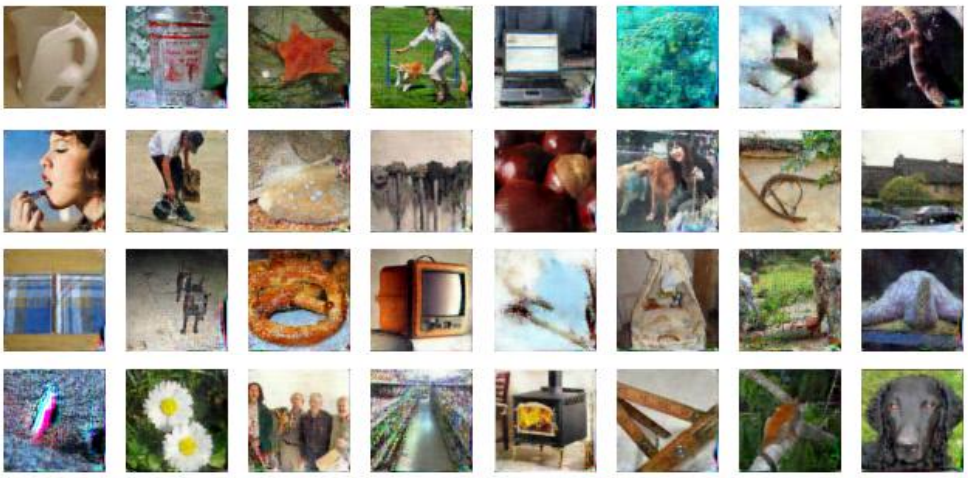

Figure 12: Gradient inversion with CI-Net on ImageNet. Batch size equals 32.