Supplementary Material for "GIFD: A Generative Gradient Inversion Method with Feature Domain Optimization"

Hao Fang[1,2] Bin Chen[1,3,4*] Xuan Wang[1,3,4] Zhi Wang[2,3] Shu-Tao Xia[2]
[1]Harbin Institute of Technology, Shenzhen
[2]Tsinghua Shenzhen International Graduate School, Tsinghua University [3]Peng Cheng Laboratory
[4]Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies
190110304@stu.hit.edu.cn, chenbin2021@hit.edu.cn, wangxuan@cs.hit.edu.cn
{wangzhi, xiast}@sz.tsinghua.edu.cn

A. Larger Batch Sizes on FFHQ

We provide the results of different batch sizes on the FFHQ dataset. Since the label extraction algorithm[14] requires non-repeating labels in a batch, the batch size cannot exceed the number of categories. Therefore, the maximum batch size of our experiment is 8 on FFHQ. Note that the latent vector of StyleGAN2 has a relatively large number of parameters to be optimized and the CMA-ES optimizer, adopted by GGL, does not support large-scale optimization. Thus, GGL is unable to operate when $B > 2$.

Table 1: PSNR mean of different methods for different batch sizes on FFHQ.

Method	Batch Size			
	1	2	4	8
IG [3]	19.07606	16.27659	13.87481	12.24488
GI [14]	17.35061	15.55461	13.42360	12.19926
GGL [9]	14.74791	13.35473	——	——
GIAS [6]	20.07786	16.95568	13.67158	12.49889
GIFD	**21.13338**	**17.96191**	**14.34927**	**12.74023**

As shown in Table 1, GIFD outperforms all previous methods at every batch size we considered. During the label extraction process, the error rate of inferred labels is relatively high when $B > 2$, leading to degraded performance of all methods because of losing the significant information brought by the correct labels.

B. Inference Speed Comparison

GIAS [6], searching the latent and parameter space of the generative model in turn, generally performs best among the previous methods. A series of experiments have demonstrated that our method achieves consistent improvement over GIAS. Besides, GIFD only searches the feature domain, whose optimized parameters are far less compared with the generator's parameters. And GIAS requires a specific generator to be trained for each reconstructed image, consuming great inference time and GPU memory. Thus GIFD should have an advantage in inference speed. In Figure 1, we draw the cost-time curve for the intermediate feature searching phase of GIFD and the parameter space searching phase of GIAS. The corresponding PSNR values of the results are also annotated in the figure.

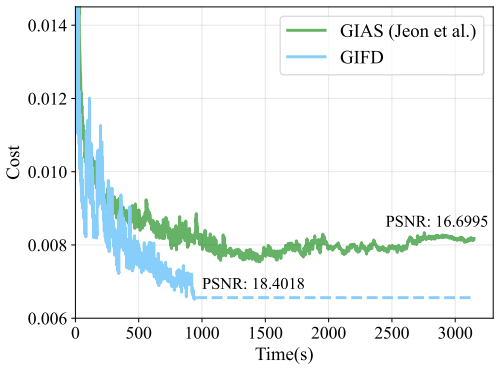


Figure 1: The cost function over time of GIAS and GIFD with $B = 4$. We give 4 trials and calculate the average values. For a fair comparison, both methods execute a total of 8000 iterations.

As expected, GIFD completes the optimization task only with less than $1/3$ of the time for GIAS and further reduces the loss. We also find that every time the optimizer moves to the next feature space, the loss comes to a peak and quickly decreases over several iterations. After getting stable, the loss value of each layer is always below the curve of GIAS.

Further, we conduct experiments on methods with the same distance metric as GIFD in Table 2. Although IG con-

*Corresponding Author

verges faster, an attacker can perform attacks offline with a copy of the historical global model and observed gradients, thus the attack effect is more essential than inference speed. In this case, GIFD achieves a good trade-off between effectiveness and time cost.

Table 2: Converge time and results at batch size 4.

Method	IG [3]	GIAS [6]	GIFD
Time(s)↓	**726.8008**	1360.7808	907.727
PSNR↑	14.1896	16.6995	**18.4018**
Loss↓	0.008029	0.007803	**0.006865**

C. Ablation Study

We conduct ablation experiments on both two datasets to further verify the effectiveness of each proposed technique. There are three variants of GIFD. GIFD-z only searches the latent space. GIFD-f starts to search the intermediate feature domain without the l_1 ball limitation and outputs the final results from the last searched intermediate layer. Based on GIFD-f, GIFD-e selects outputs from the layer with the least matching error. And GIFD is GIFD-e plus the l_1 ball limitation. The results of Table 3 show that each aforementioned technique can further improve the performance.

Table 3: Ablation study of GIFD and its three variants on every 1000th image of ImageNet and FFHQ validation set.

Method	Metric			
	PSNR↑	LPIPS↓	SSIM↑	MSE↓
GIFD-z	13.9451	0.3445	0.1463	0.0488
GIFD-f	18.6457	0.2320	0.3916	0.0180
GIFD-e	19.4662	0.1900	0.4383	0.0161
GIFD	**20.0534**	**0.1559**	**0.4713**	**0.0141**

(a) ImageNet

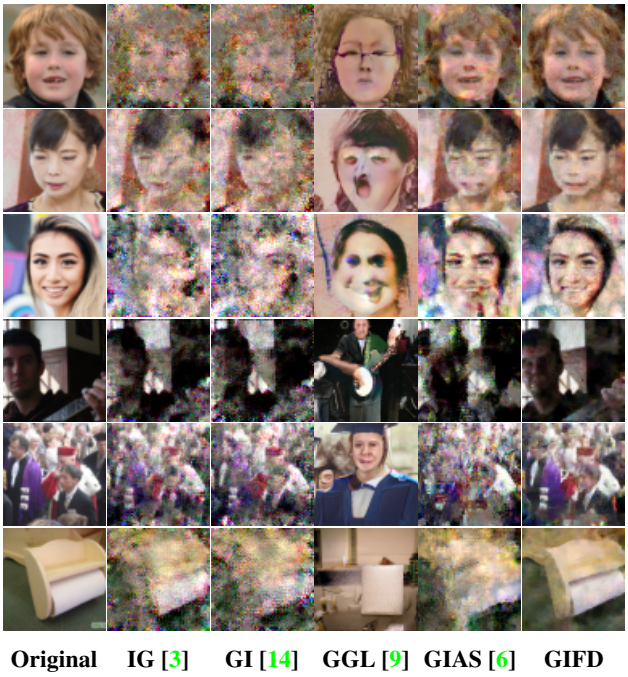
Method	Metric			
	PSNR↑	LPIPS↓	SSIM↑	MSE↓
GIFD-z	16.9947	0.1351	0.3931	0.0263
GIFD-f	20.2506	0.1462	0.5210	0.0123
GIFD-e	20.5839	0.1267	0.5412	0.0119
GIFD	**21.3368**	**0.1023**	**0.5768**	**0.0098**

(b) FFHQ

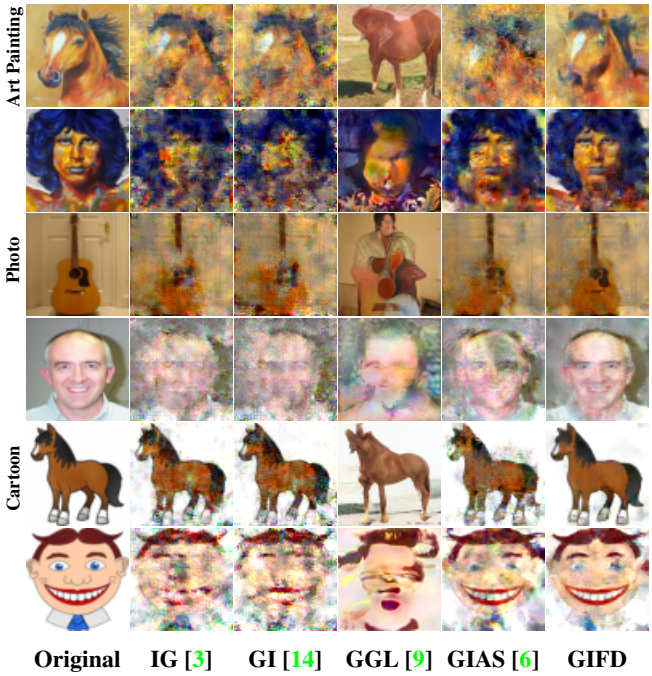
D. More Visual Comparison

We show more qualitative comparison on both in-distribution datasets (ImageNet[2] and FFHQ[7]) and out-of-distribution data (PACS [8]) in Figure 2.

We can easily find that our reconstructed images stay closer to the original manifold. This again provides evidence that our method fully exploits the generative model as an image prior and hence could reveal more sensitive information about the private data.



(a) In-distribution Data



(b) Out-of-distribution Data

Figure 2: More visual comparison of different methods on in-distribution and out-of-distribution data.

E. More FL Global Models.

To further validate our method, We provide numerical results of PSNR on more global models below.

Table 4: PSNR for different global models on ImageNet.

Global model	IG [3]	GI [14]	GGL [9]	GIAS [6]	**GIFD**
ConvNet	22.8043	21.3876	13.2582	24.1741	**25.5646**
AlexNet	15.3390	16.3423	13.6605	17.5883	**19.7926**
VGG-16	13.3505	13.3856	13.9808	14.5414	**16.0014**
ResNet-18	17.0756	16.5109	13.3885	17.4923	**20.0534**
DenseNet-121	18.0840	17.3253	14.3538	18.1686	**19.7376**

As shown in Table 4, the overall attack performance varied across different global models and GIFD always performs the best, convincing the superiority of our method. It also implies that the model structure is related to the defense effect. Further study can look into it and design more secure model structures.

F. Details about Gradient Transformation

More specifically, the adversary can infer three defense strategies as follows (denote the received gradients by g):

(1) *Gradient clipping.* Given a clipping bound c, gradient clipping transforms the gradients as $\mathcal{T}(g, c) = g \cdot \min(\frac{c}{\|g\|_2}, 1)$. Since this operation is always layer-wise, the attacker can compute the ℓ_2 norm at each layer of the received gradients as the estimated clipping bound.

(2) *Gradient sparsification.* Given a pruning rate $p \in (0, 1)$, the client only transmits the $(1-p)$ largest values of g (in absolute value) and the rest values are replaced by zero. Implemented by applying a layer-wise mask, the gradient sparsity can be estimated by observing the percentage of non-zero entries in the shared gradients.

(3) *Soteria.* Recently proposed by [12], it is an efficient and reliable defense strategy. This operation is actually equivalent to applying a mask only to the gradients of the defended layer. Once the global model f_θ and input $\mathbf{x}$ are given, this process becomes deterministic. Then, the attacker can inverse this mask according to the non-zero entries of the gradients from the defended layer.

G. Another Approach for OOD Problem

Motivated by previous work [4, 13], Jeon *et al*. [6] propose another method that can solve the out-of-distribution (OOD) problem through training a generative model only with the shared gradients, *i.e*., Gradient Inversion to Meta-Learn (GIML). They regard the global model in FL framework as a discriminator and train the generator by solving a set of gradient inversion tasks. However, assuming that the private labels are known, their experiments are limited to 32×32 images and the improvement is also limited. On the contrary, our method has impressive effects on OOD data and only requires a pre-trained GAN, which releases the computing costs of training a generative model.

H. Discussion

Reconstruction at large batch sizes. Although our method performs well in a range of experiments, the improvement at large batch sizes is still limited, implying that attacks in such a scenario are still a major challenge. Meanwhile, we make the assumption that there are no duplicate labels in each batch to infer the labels, which is also difficult to achieve in the real scenario. To relax the assumption that non-repeated labels in a batch, we notice a recent paper [10] has addressed the problem effectively, which can be perfectly combined with our GIFD to enhance the attack when there are duplicate labels.

Hypothesis about OOD data. In order to utilize the powerful information brought by labels, we assume that images from different distributions have the same label space. To tackle more realistic OOD problems where label spaces are different, subsequent research could consider using more powerful diffusion models [5, 11] as prior information, or using other related techniques to improve the expressiveness of generative models.

I. Experimental Details

For each intermediate feature domain, we use Adam optimizer with 0.1 as the initial learning rate and give 1000 iterations. We adopt the warm-up strategy, where the learning rate linearly warms up from 0 to 0.1 during the first $1/20$ of the optimization and gradually decays to 0 in the last $3/4$ stage using cosine decay.

Guided by the theory [1] that a sequence of increasing radii of the l_1 ball tends to provide better results, we gradually allow larger deviations and tune the r by experiment, obtaining an appropriate setting as follows.

(1) For BigGAN, we only need to constrain the intermediate features:

- Intermediate features: [2000, 2500, 3000, 3500, 4000, 4500, 5000, 5500, 6000].

(2) StyleGAN2 has more particularities we need to handle for feature domain optimization. In addition to the intermediate features, we optimize the noise vectors and apply the l_1 ball constraint to them at the same time. Involved in the generation of styles in StyleGAN2, the latent vectors also need to be optimized and we constrain their searching range within an l_1 ball as well:

- Intermediate features: [2000, 3000, 4000, 5000]
- Noises: [1000, 2000, 3000, 4000, 5000]
- Latent vectors: [1000, 2000, 3000, 4000, 5000]

For the image fidelity regularization, we use $\alpha_{TV} = 10^{-4}$, $\alpha_{l_2} = 10^{-6}$. We run all experiments on NVIDIA RTX 2080 Ti GPUs and A100 GPUs. The experiments on the effects of K and on defense strategies are each conducted on 30 randomly selected images and the numerical results for batch size are the averages of 10 batches.

References

[1] Giannis Daras, Joseph Dean, Ajil Jalal, and Alex Dimakis. Intermediate layer optimization for inverse problems using deep generative models. In *International Conference on Machine Learning*, pages 2421–2432. PMLR, 2021. 3

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2

[3] 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. 1, 2, 3

[4] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. Deep models under the gan: information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 603–618, 2017. 3

[5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 3

[6] 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. 1, 2, 3

[7] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 2

[8] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017. 2

[9] Zhuohang Li, Jiaxin Zhang, Luyang Liu, and Jian Liu. Auditing privacy defenses in federated learning via generative gradient leakage. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10132–10142, 2022. 1, 2, 3

[10] Kailang Ma, Yu Sun, Jian Cui, Dawei Li, Zhenyu Guan, and Jianwei Liu. Instance-wise batch label restoration via gradients in federated learning. In *The Eleventh International Conference on Learning Representations*, 2022. 3

[11] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. 3

[12] Jingwei Sun, Ang Li, Binghui Wang, Huanrui Yang, Hai Li, and Yiran Chen. Soteria: Provable defense against privacy leakage in federated learning from representation perspective. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9311–9319, 2021. 3

[13] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. Beyond inferring class representatives: User-level privacy leakage from federated learning. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 2512–2520. IEEE, 2019. 3

[14] 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. 1, 2, 3