Cracking Federated Privacy: Initialization-Resilient Gradient Inversion with Fine-Grained Reconstruction

Kaiming Zhu Jinsheng Yang Siyang Guo Huaqian Qin Taiyu Wang
Junbo Wang* Yuhong Nan Zibin Zheng

Sun Yat-sen University, P.R. China
{zhukm3, yangjsh27, qinhq5, wangty56}@mail2.sysu.edu.cn, gsiyang210906@gmail.com
{wangjb33, nanyh, zhzibin}@mail.sysu.edu.cn

Abstract

Federated Learning (FL) remains vulnerable to Gradient Inversion Attacks (GIA), where shared gradients can reveal clients' private data. Existing attacks struggle under early-stage initialization variations and often produce coarse reconstructions. In this paper, we identify sparsity changes in shared gradients as the primary source of this sensitivity and propose an initialization-resilient GIA with a coarse-to-fine design, achieving fine-grained recovery. The coarse stage aligns gradient directions and constrains non-zero entries to mitigate sparsity changes, while the fine stage refines magnitude alignment by a hybrid metric combining Cosine distance with a deformed Manhattan term. Extensive experiments against five baselines show up to 200% PSNR gain (25.4 $\rightarrow$ 47.7 dB) under sensitive initializations on CIFAR-10/100, with consistently delivering fine-grained recovery across four datasets and the entire FL lifecycle. Our method maintains competitive performance with SOTA baselines across batch sizes and local steps and reveals persistent leakage on several popular models and insufficient defenses, underscoring the urgent need for stronger privacy-preserving mechanisms.

1 Introduction

As a collaborative machine learning model training framework on distributed data, Federated Learning (FL) is regarded as a promising solution to balance model performance and privacy concerns [20, 23]. A classic FL scenario follows an iterative gradient-sharing process in a client-server setup: the server distributes a global model to clients, each client computes gradients from its private data and sends them to the server, which updates the model by aggregating the client gradients. Through iterative gradient computation and aggregation, the global model can learn from clients' private data without directly accessing it.

While FL offers a decentralized approach to model training, it is not as privacy-preserving as initially believed. Gradients, which encode private data in high-dimensional space, can be exploited to reconstruct sensitive information. Gradient Inversion Attacks (GIA) exemplify this threat, enabling the recovery of private data from gradient observations [48]. As shown in Fig. 1, attackers can reconstruct the underlying input from randomly initialized dummy data by minimizing the Euclidean distance between their gradients [50, 52]. However, the practicality of GIA remains controversial. A key limitation is its instability over the FL lifecycle. Specifically, after several rounds of local updates, GIA performance deteriorates sharply, yielding noisy and uninformative reconstructions [34, 43]. Given FL's dynamic client participation, clients can mitigate leakage risks by simply joining later, further weakening GIA's effectiveness in practice.

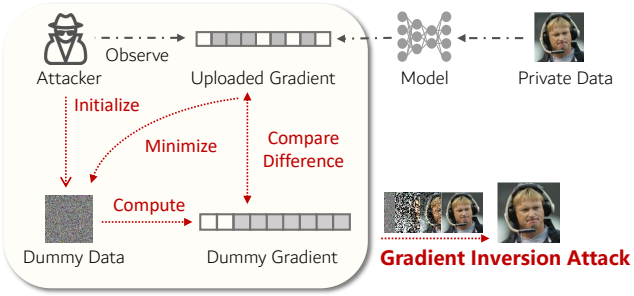


Figure 1: Workflow of Gradient Inversion Attack(GIA): reconstructing private data by matching gradients.

To improve attack stability, Geiping et al. proposed a widely adopted solution: Cosine-distance-based GIA [13]. They observed that as FL progresses, model accuracy improves while gradient norms shrink, making traditional Euclidean-distance-based GIA unstable due to its sensitivity to magnitude changes. In contrast, cosine distance emphasizes gradient direction over magnitude, thereby mitigating this instability. However, this approach still faces two key limitations. First, it does not address instability in the early stages of FL, where variations in model initialization can degrade attack performance, particularly for models initialized with normal distri-

*Corresponding author.

butions [34, 37]. Since such variations do not affect gradient magnitude, Cosine-distance-based GIA remains vulnerable. Second, by focusing solely on gradient direction, this method often yields coarse reconstructions, akin to a blurred image that captures overall structure but lacks fine details. Although later work improved this by assigning different weights to gradient distances across layers [40, 42], these methods require extensive empirical tuning, increasing costs to deploy GIA.

To address the above limitations, we propose an initialization-resilient, coarse-to-fine GIA that enables fine-grained reconstructions. Our method is motivated by an empirical analysis showing that variations in model initialization induce changes in the sparsity of clients' shared gradients, which negatively correlate with the performance of existing GIA methods. To mitigate this issue, we design a two-stage optimization framework that explicitly stabilizes inversion under sparsity variations. Our contributions are summarized as follows:

- We propose an initialization-resilient, coarse-to-fine GIA achieving fine-grained recovery across the FL lifecycle. The coarse stage aligns gradient directions and constrains non-zero entries to mitigate sparsity changes, while the fine stage refines magnitude alignment along the pre-aligned directions with a deformed Manhattan term.

- We conduct extensive experiments with six state-of-the-art baselines and limit the attacker's knowledge to simulate attackers in realistic FL systems. Our method achieves up to 200% PSNR improvement (from 25.4 $\rightarrow$ 47.7 dB) under sensitive initializations on CIFAR-10/100. It consistently delivers fine-grained recovery across five datasets, demonstrating competitive performance across varying batch sizes and local steps, underscoring an underestimated leakage from gradients.

- We evaluate the method against two popular perturbation-based defenses-Differential Privacy and Gradient Pruning-commonly used to mitigate gradient leakage. Our results show that insufficient application of these defenses leads to significant privacy leakage, even in high-sensitivity scenarios, highlighting the need for stronger privacy-preserving mechanisms

2 Related Work

2.1 Gradient Inversion: Attacks

GIA reconstructs private data by matching gradients [48]. The choice of distance metric is crucial. Based on the metric used to compare gradients, prior GIA work largely falls into two groups: **Euclidean-distance-based GIAs** and **Cosine-distance-based GIAs**.

Euclidean-distance-based GIAs. Magnitude- distance-based GIA measures the degree of gradient matching by directly calculating the magnitude difference between two sets of gradients. Magnitude- distance-based GIA measures the degree of gradient matching by directly calculating the magnitude difference between two sets of gradients. Zhu et al. [52] introduced the first Euclidean-distance-based GIA. Attackers minimize Euclidean distance by jointly optimizing the data and the label. Later, Zhao et al. [50] showed that the label can be inferred directly from gradients. This led to *improved Deep Leakge in Gradient (iDLG)*, which splits the attack into label inference and data recovery. This simplification reduces computational complexity and leads to a significant quality improvement. Yin et al. [44] used meta-information from specific layers (e.g., Batch Normalization statistics) as regularization. This enabled high-resolution recovery on well-trained models. However, Euclidean distance is sensitive to gradient magnitude changes. During FL, gradient norms shrink over global epochs [2, 13, 34]. As a result, these methods tend to degrade in later stages of FL unless additional information is available.

Cosine-distance-based GIAs. Cosine distance is another common metric for comparing gradients. Since cosine distance measures the difference in gradient directions, it is less affected by changes in gradient magnitudes, ensuring stable performance throughout the FL lifecycle [13]. However, two key issues persist. First, their stability is compromised under initialization variations. Recent studies highlight the sensitivity of GIA to changes in initialization [34, 37, 47]. Since initialization variations often do not affect gradient magnitudes, Cosine-distance-based GIAs also degrade. Second, their reconstruction quality is often suboptimal. Since cosine distance only measures the direction of gradients, it leads to coarse reconstruction. This typically results in a blurry image with unclear facial features or missing background details. While later works attempted to address this by assigning layer-specific weights to gradients [40, 42], these approaches require extensive empirical tuning, significantly increasing the deployment cost of GIA.

Our Positioning. We present a GIA that is resilient to initialization variation and recovers fine-grained details across the FL lifecycle. Building on observations of initialization sensitivity [37], we identify *gradient-sparsity shift* in shared gradients as a key factor and design an initialization-resilient reconstruction procedure to mitigate it. Unlike Cosine–distance–based GIAs, which align only gradient directions and yield coarse reconstructions [13], our coarse-to-fine method first aligns directions and then magnitudes, yielding consistent gains while avoiding manually tuned, per-layer distance weights [40, 42]. These improvements are achieved under a minimum-knowledge, honest-but-curious attacker with no auxiliary information about the private data [44], underscoring practical risks in real-world FL deployments.

2.2 Gradient Inversion: Defenses

From an implementation perspective, defenses against GIA fall into two families:

(i) Encryption-/Protocol-based defenses. These methods are based on Homomorphic Encryption (HE) and Secure Multiparty Computation (SMC) [39, 49]. HE encrypts per-client gradients and supports aggregation on ciphertexts, while SMC masks/secret-shares per-client gradients and ensures only the aggregated results are accessible. These mechanisms provide strong privacy guarantees by encrypting per-client gradients, but incur additional computation and communication overhead [45]. Additionally, since clients' gradients are hidden, server-side detection of poisoned or arbitrary updates is precluded without additional defenses [36]; if aggregated, such updates can degrade model performance.

(ii) Perturbation-Based Defenses. These methods perturb gradients before sharing, most commonly via Differential Privacy (DP) [17, 34] and Gradient Pruning (GP) [31, 32, 34, 52]. DP aims to ensure that gradients computed from different data are statistically indistinguishable, by clipping them into a bounded threshold and perturbing with Gaussian noise. GP aims to reduces the information encoded in the shared gradients, which will sparsify a fraction of gradients entries with small magnitudes, and attackers are only visible to the ground truth value of large entries. As their perturbation strengths increases (e.g., DP with smaller clipping bound and higher noise variance, GC with larger pruning ratios), the observed gradients deviate further from the true private-data gradients, and GIA performance declines [47]. However, training on these perturbed gradients typically reduces model accuracy and slows convergence, raising significant privacy–utility trade-offs in practice [19, 36]. But without sufficient perturbations, residual leakage to private data might persist [52]. Therefore, designing such practical defenses that balance both privacy and utility remains a critical open challenge.

Our Positioning. We study the privacy leakage of gradient in practical FL and argue that prior failures of GIAs can lead to an underestimation of risk. Specifically, we present a GIA that is resilient to initialization variation and reconstructs fine-grained details across the FL lifecycle. Our coarse-to-fine approach yields consistent gains over prior work under a minimum-knowledge, honest-but-curious threat model, indicating residual risk in realistic settings. We further evaluate Perturbation-Based Defenses, including DP (gradient clipping and Gaussian noise injection) and GP (gradient sparsification). We find that when these perturbations are insufficiently strong, significant leakage can persist. This observation illustrates the risk of several defenses popular for gradient leakages [32, 34], which weakens perturbations to achieve better privacy–utility trade-offs.

3 Preliminaries

3.1 Threat Model

Our threat model strictly follows established standards for evaluating privacy leakage in practical FL systems [10, 34] and extends them to a more constrained scenario for quantifying practical leakage. Details of the threat model are outlined below.

Envisioned Attacker(s). We assume that the server in the FL system is a *minimum-knowledge-limited*, *honest-but-curious* attacker, targeting a randomly selected client as the victim. The attacker's profile is outlined in three parts.

Firstly, Goal and Methodology. The attacker seeks to reconstruct the victim's private data. The attacker achieves this goal by performing GIA on the victim's shared gradients.

Secondly, Timing. GIA occurs during a single, randomly chosen global training round, with the attacker observing shared gradients for only that round. This restricts the attacker's available knowledge, as multi-round observations [40] or targeting performance-boosting rounds [34] could enhance the attack.

Thirdly, Abilities and knowledge. The attacker adheres strictly to the FL protocol, without altering any components or protocols. The attacker's knowledge is limited to the essentials required for performing GIA, such as models, loss functions, data dimensions, and the number of samples in the gradients [10, 34]. [23]. Privacy-sensitive meta-information, such as Batch Normalization statistics, is assumed to be unavailable, as clients can omit it without affecting FL training [34, 43].

Threat surfaces. Our threat surface primarily arises from the gradient-sharing mechanism inherent to FL [20, 23]. Shared gradients from clients act as a potential entry point for the privacy leakage to their private data. This threat is particularly relevant in practical FL systems, where gradient accessibility is essential for detecting malicious clients. The server must validate gradients from all clients to prevent harmful updates, such as poisoned or randomly generated gradients, from being aggregated. Such updates could significantly degrade model performance [33, 36].

Generality & Practicality. The generality and practicality of our approach are demonstrated in three aspects. First, we consider a standard FL system driven by clients' gradient sharing [20, 23], which inherently exposes entry points for privacy leakage. Second, we assume a minimum-knowledge-limited, honest-but-curious attacker, which strongly adheres to standards to evaluate the practical privacy risks in the FL system [10, 34]. Third, our attack does not rely on assumptions about specific model structures or data complexity (e.g., image resolution), making it applicable across a wide range of FL tasks.

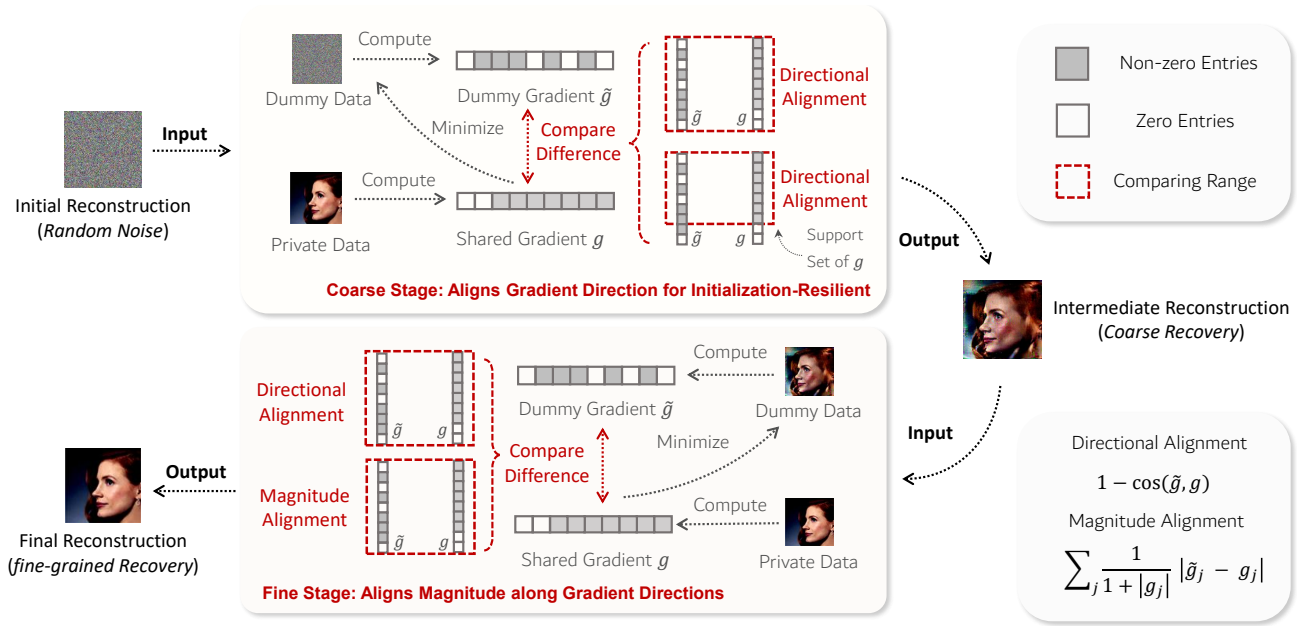


Figure 2: Overview of coarse-to-fine GIA. The coarse stage stabilizes direction alignment with non-zero entries constraints by Eq. (1); the fine stage refines magnitudes with directional alignments by Eq. (2).

3.2 Problem Formulation

Given a machine learning model $F(x;\theta)$ and a loss function $L(F(x;\theta),y)$, a random client in FL holds $n \geq 1$ private datapoints (x,y) , where data $x \in \mathbb{R}^{n\times d}$ and label $y \in \mathbb{R}^n$. The client calculates the gradient $g = \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta L(F(x_i;\theta),y_i)$ and shares it to the attacker. The attacker aims to reconstruct these private datapoints (x,y) by observing the gradient g. Specifically, the GIA attacker first initializes dummy data $\tilde{x} \in \mathbb{R}^{n\times d}$ and then optimizes it to obtain $\tilde{x}^*$ by minimizing the following objective:

$$\tilde{x}^* = \arg\min_{\tilde{x}} \; d(\tilde{g},g) + R(\tilde{x})$$

Where $\tilde{g} = \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta L(F(\tilde{x}_i;\theta),y_i)$ is the gradient of the dummy data $\tilde{x}$, and the label of private data y could be directly inferred from observing the gradient g [5, 13, 41]. $d(\cdot,\cdot)$ is the distance measure between gradients, and $R(\tilde{x})$ is the regularizing term of $\tilde{x}$ in GIA.

4 Coarse-to-fine Gradient Inversion Attack

We propose a Coarse-to-fine GIA that mitigates initialization variability and enables fine-grained reconstruction of client data. The whole scheme is illustrated in Fig. 2, consisting of a two-stage optimization on randomly initialized dummy data.

In the following subsections, we first state the motivation of this proposal by investigating two empirical observations regarding on reconstruction performance, and then describe the proposed methods in details, and finally show the integrated algorithm.

4.1 Observation and Motivation

Our motivation derives from two empirical observations regarding the impact of initialization changes on reconstruction accuracy.

1. We observed that variations in model initialization lead to changes in the sparsity of shared gradients, which in turn induce instability in GIA.

2. We compared the reconstruction accuracy under Cosine-distance-based and Euclidean-distance-based attack objectives. The results indicate that the Euclidean-distance-based loss exhibits more unstable performance than the Cosine-distance-based loss when facing changes in sparsity.

Specifically, the details are shown as follows:
Observation 1. Figure 3 presents our empirical study on InvG [13], which is used here as a representative GIA. We randomly sample 20 datapoints from each of the CIFAR-10 and CelebA-HQ datasets and evaluate them under Kaiming-Normal [16], Kaiming-Uniform [16], Orthogonal [29], and Transferred [25] initialization schemes. We find that the pixel-wise reconstruction fidelity (PSNR↑) decreases as gradient sparsity increases, with an average Pearson correlation of -0.677 on CelebA-HQ and -0.478 on CIFAR-10. According to commonly used statistical conventions [6], Pearson

Dataset	Metrics	Initialization Methods				Pearson Coefficient
		Uniform	Orthogonal	Normal	Transfer	
CelebA-HQ	Sparsity	0.203	0.213	0.240	0.344	-0.677
	PSNR↑	18.91	17.03	13.32	11.16	
CIFAR-10	Sparsity	0.241	0.266	0.284	/	-0.478
	PSNR↑	37.90	28.20	22.65	/	

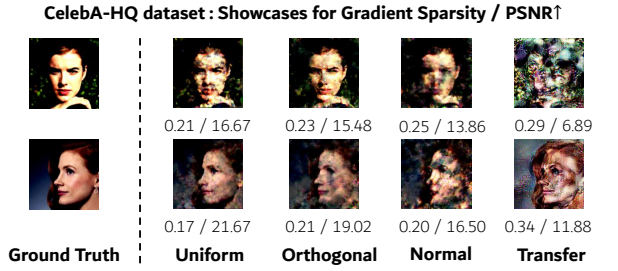


Figure 3: Qualitative comparison of GIA [13] under varying gradient sparsity on ResNet-18. The figure reports reconstruction fidelity measured by PSNR (↑) across different initialization methods on CIFAR-10 and CelebA-HQ. Pearson correlation coefficients ($[-1,1]$) between gradient sparsity and PSNR are shown to summarize their relationship.

correlation coefficients with absolute values in the range of approximately 0.3–0.5 are typically regarded as moderate, while values above 0.5 indicate a stronger association. These observations indicate a clear negative relationship between gradient sparsity and reconstruction fidelity.

Observation 2. Figure 4 presents our empirical study under the same setup as Fig. 3. We consider two representative GIAs: iDLG [50], which adopts a magnitude-based alignment objective, and InvG [13], which is based on directional alignment. To ensure a fair comparison, we focus on the CIFAR-10 dataset, where both methods are applicable, and evaluate them using their original implementations. Across different initialization schemes, we observe a consistently stronger negative correlation between gradient sparsity and reconstruction fidelity for magnitude-based alignment than for directional alignment. Specifically, with higher gradient sparsity (e.g., Kaiming-Normal [16]), InvG [13] is still able to produce meaningful reconstructions in most cases, while iDLG [50] more frequently yields noise-perturbed reconstructions. These observations suggest that, in terms of GIAs' objectives, directional alignment is relatively more robust to increased gradient sparsity compared to magnitude-based alignment.

Motivations. Existing work has demonstrated that combining cosine-based and magnitude-based losses typically yields superior gradient recovery [12], indicating their complementary potential. However, they typically adopt a fixed-weight linear combination of the two losses throughout the attack process, with limited analysis of their behavioral differences under varying initializations. Our empirical results reveal that as shared gradients become sparser, the reconstruction quality of magnitude-based loss degrades more significantly. In contrast, cosine-based loss demonstrates greater robustness under identical settings. Based on these observations, we argue that simply "combining both losses simultaneously" fails to leverage their respective strengths. To address this, we propose a two-stage optimization strategy. In the first stage, we use a purely directional objective which yields a coarse reconstruction and is more robust to initialization. In the second stage, we introduce a reweighted L_1 loss to finely align gradient magnitudes. Overall, the two-stage design achieves a better trade-off between optimization stability and fine-detail recovery.

Figure 4: Qualitative comparison of GIAs under varying gradient sparsity on CIFAR-10. Following the setup in Fig. 3, we visualize InvG [13] and iDLG [50], whose objectives are based on directional and magnitude alignment, respectively, across different initialization schemes. Pearson correlation coefficients between gradient sparsity and PSNR are $r = -0.478$ (directional) and $r = -0.555$ (magnitude).

4.2 Coarse Stage: Aligns Gradient Direction for Initialization-Resilient

In the coarse stage, we target initialization sensitivity, which is associated with gradient sparsity changes and degraded inversion quality. We focus on directional alignments to alleviate it, which takes benefits from the relative robustness of direction alignments towards initializations changes. Specifically, our optimizing target is formulated as Eq. (1):

$$\begin{aligned} d_1(\tilde{g},g) &= 1-\cos(\tilde{g},g)+\lambda_1\big(1-\cos(\tilde{g}_S,g_S)\big),\\ S &:= \{i \mid g_i \neq 0\},\\ \tilde{g}_S &:= (\tilde{g}_i)_{i\in S},\\ g_S &:= (g_i)_{i\in S}. \end{aligned} \tag{1}$$

Here, the set S indicates the support set of the gradient g, with $S = \mathrm{supp}(g) = \{i | g_i \neq 0\}$. The first term constraints $\tilde{g}$ to align with g in the global direction over all coordinates. The second term computes cosine distance restricted to the support of g, focusing directional matching on coordinates where the target gradient carries non-zero signals. This complements the global cosine term by focusing on the signal-bearing subspace, which reduces the influence of off-support components of $\tilde{g}$ when g becomes sparse.

Our experiment results in Sec. 5.6 indicates a stabler reconstructions of the overall structure (SSIM↑, structural similarity) of the image. This aligns with the target of this stage, which is to provide a stabler reconstructions by coarse output, ensuring a better start-point of next stage reconstruction.

4.3 Fine Stage: Aligns Magnitude along Gradient Directions

In the fine stage, we refine the coarse reconstruction into a fine-grained recovery by matching gradient magnitudes while preserving the already-aligned direction. Specifically, our optimization objective can be formalized as:

$$d_2(\tilde{g},g) = 1-\cos(\tilde{g},g) + \lambda_2 \sum_j \frac{1}{1+|g_j|}|\tilde{g}_j - g_j|, \quad (2)$$

where λ_2 controls the relative strength of the magnitude term. The first term $1-\cos(\tilde{g},g)$ retains the directional constraint, preventing the optimization process from deviating from the true direction after introducing the magnitude term.

The second term is a reweighted L_1 loss, designed to prevent the overall objective from being dominated by large-magnitude coordinates. When $|g_j| \gg 1$, we have $1+|g_j| \approx |g_j|$, and the contribution of the j-th coordinate is approximately

$$\frac{1}{1+|g_j|}|\tilde{g}_j - g_j| \approx \frac{|\tilde{g}_j - g_j|}{|g_j|},$$

which corresponds to a relative-error matching: by normalizing the discrepancy with the magnitude $|g_j|$, it effectively removes the impact of numerical scale. This normalization forces the optimizer to attend to every coordinate in the gradient vector, regardless of its absolute size, instead of being biased toward a few large entries. When $|g_j| \approx 0$, the weight reduces to $1/(1+|g_j|) \approx 1$, and the term degenerates to the standard absolute error $|\tilde{g}_j - g_j|$. Here, the constant "1" in the denominator acts as a safeguard to maintain numerical stability and avoid division by values close to zero.

4.4 Algorithm and Regularizer

Algorithm. In summary, our method follows a progressive coarse-to-fine design, first achieving robust coarse reconstruction via directional alignment and then refining fine-grained details through magnitude alignment, thereby mitigating initialization sensitivity while enabling fine-grained recovery. The full procedure is summarized in Algorithm 1.

Regularizer. Following prior work [13, 40, 42], we adopt Total Variation (TV) loss, a common regularization term that improves image recovery quality by enforcing smoothness. It is formulated as follows, with β as a rescaling hyperparameter:

$$R_{tv}(x) = \sum_{i,j} \left((x_{i,j+1} - x_{i,j})^2 + (x_{i+1,j} - x_{i,j})^2 \right)^{\beta/2}.$$

Algorithm 1 Coarse-to-fine Gradient Inversion Attack

Input: Distributed model $F(x;\theta)$,
Loss function $L(F(x;\theta),y)$,
Client's shared gradient g,
Learning rate of the coarse stage α_1,
Learning rate of the fine Stage α_2,
Iterations to optimize in the coarse stage T_1,
Iterations to optimize in the fine Stage T_2,
$R(x)$ Regularization terms for data
Output: Reconstructed private data point $(\tilde{x}^*, \tilde{y}^*)$

// Initialize dummy data point
$\tilde{y} \leftarrow$ directly inferring from g ▷ *by prior arts [44]*
$\tilde{x} \leftarrow \mathcal{N} \sim (0, \sigma^2)$ ▷ *random noises*
// Coarse Stage
for i in $1,2,...,T_1$ **do**
  $\tilde{g} \leftarrow \nabla_\theta L(F(\tilde{x};\theta),\tilde{y})$
  $S \leftarrow \{i \mid g_i \neq 0\}$
  $\tilde{g}_S \leftarrow (\tilde{g}_i)_{i\in S}$
  $g_S \leftarrow (g_i)_{i\in S}$
  $d_1(\tilde{g},g) \leftarrow 1-\cos(\tilde{g},g) + \lambda_1(1-\cos(\tilde{g}_S, g_S))$
  $\tilde{x} \leftarrow \tilde{x} - \alpha_1 \nabla_{\tilde{x}}[d_1(\tilde{g},g) + R(\tilde{x})]$
end for
// Fine Stage
for i in $1,2,...,T_2$ **do**
  $g^* \leftarrow \nabla_\theta L(F(\tilde{x};\theta),\tilde{y})$
  $d_2(\tilde{g},g) \leftarrow 1-\cos(\tilde{g},g) + \lambda_2 \sum_i \frac{1}{1+|g_i|}|\tilde{g}_i - g_i|$
  $\tilde{x} \leftarrow \tilde{x} - \alpha_2 \nabla_{\tilde{x}}[d_2(\tilde{g},g) + R(\tilde{x})]$
end for
$\tilde{x}^*, \tilde{y}^* \leftarrow \tilde{x}, \tilde{y}$
Return $\tilde{x}^*, \tilde{y}^*$

The regularization is applied uniformly across stages as:

$$R(\tilde{x}) = R_1(\tilde{x}) = R_2(\tilde{x}) = \lambda_{tv} R_{tv}(\tilde{x}).$$

Adaptive Scaling on Regularizer. We set $\beta = 4$, consistent with prior work [13, 40, 42]. To avoid manual fine-tuning, we scale λ_{tv} adaptively as:

$$\lambda_{tv} = \lambda_{tv}(x) = -0.00008 \log(\text{scale}(x)) + 0.0005. \quad (3)$$

This design is motivated by the multiscale nature of images: coarse outlines dominate at small scales and induce larger TV losses , while finer details emerge at larger scales and are reflected in smaller-magnitude regularization terms. Thus, λ_{tv} should increase with image scale. Following [40], we tested linear, exponential, and logarithmic scaling, and found the logarithmic form in Eq. (3) yielded the best reconstruction quality.

5 Experiment

We comprehensively evaluate our coarse-to-fine cosine-distance GIA against six state-of-the-art baselines across five diverse datasets , addressing the following questions:

Q1. (Initialization Stability): Can our method overcome sensitivity to model initialization in the early stages of FL?

Q2. (Lifecycle Stability): Does it remain effective as models converge throughout the FL lifecycle?

Q3. (Training Generality): How does the method perform in common FL settings involving batched gradient averaging or multi-step aggregated updates?

Q4. (Scenario Generality): Does it generalize across model architectures? Does it enlarge the unsafe operating regime, where popular defenses have not offered sufficient protection?

How does the method affect the operating regimes under which existing defenses provide limited protection?

Q1–Q2 evaluate the persistence of the threat across the FL lifecycle, while Q3–Q4 assess its practical viability. Experiments are conducted on a server equipped with an Intel(R) Xeon(R) Gold 5218 CPU and five NVIDIA GeForce RTX 3090 GPUs, using PyTorch 1.13.1, and CUDA 11.7.

5.1 Experimental settings

Dataset. We evaluate on five standard 3-channel image datasets, normalized to zero mean and unit variance, spanning low- and high-resolution settings. The low-resolution datasets are **CIFAR-10** (10 classes) [21] and **CIFAR-100** (100 classes) [21], with resolutions $32 \times 32 \times 3$. The high-resolution datasets are **ImageNet ILSVRC 2012** (1000 classes) [8] and **CelebA-HQ** [22] (307 classes), **Lung and Colon Cancer** [22] (LC25000, 5 Classes) [3], with resolutions $224 \times 224 \times 3$. For CelebA-HQ, we follow the preprocessing of previous work [25], segmenting it by identity, removing identities with fewer than 15 images, and applying a $4:1$ train–test split.

Models. We conduct experiments on **ResNet-18** [16], **ResNet-34** [16], **DenseNet-121** [18], and **MobileNet-V2** [28]. For MobileNet-V2 , we adopt the the architecture configuration used in prior privacy attacks on CelebA-HQ [26]. To further validate performance on well-fitted models, we also evaluate on open-source pretrained ResNet-18 models and reproduce the training process by their provided scripts, including one trained on CIFAR-10 from the Detector Library [7] and another trained on CelebA-HQ from Latent-HSJA [25].

All comparative experiments are conducted on ResNet-18, a commonly adopted backbone in GIA [40, 42] and defense evaluations [32, 34]. We adopt ResNet-18 with the above open-source configurations, as it achieves strong test accuracy (77.3% on CelebA-HQ [25] and 95.3% on CIFAR-10 [7]), ensuring that the evaluated models are applicable and free from degenerate training regimes.

Baselines. We adopt the Breaching framework as the implementation backbone. Breaching [1], together with its predecessor InverseFed [2],has been widely used in prior GIA-related research, supporting both attack development [13, 40, 42] and evaluation [10, 32, 34]. Using this framework, we benchmark six representative GIA methods, grouped by their distance measures. For Euclidean-distance-based GIA, we include **iDLG** [50], an early approach, and **SAPAG** [37], which improves stability under varying initializations. For Cosine-distance-based GIA, we consider **InvG** [13], a foundational method, along with **AGIC** [40] and **HFGradInv** [42], which enhance reconstruction fidelity through layer-wise reweighting of gradient distances. For the hybrid-distance-based GIA, we evaluate **FedLeak** [12], which combines directional and magnitude alignments. Baseline-specific implementation and adaptation details are provided in the Appendix A and our released code [51].

Evaluation Setup. In our evaluation, we follow the experimental setup used by prior baselines that (i) assume the same threat model as ours and (ii) report their performance under this evaluation pipeline [13, 40]. All experiments are conducted under a unified setup, including data partitioning, gradient computation, and gradient matching.

For GIA configurations, all attacks use the same label inference strategy [44], constrained to the same attacker knowledge and regularized with only TV loss. Each method is optimized for up to 30,000 iterations and the best intermediate result is reported. Each experiment is repeated 20 times with randomly sampled private data in each trial, and reported metrics are averaged across trials. Optimization is retried 5 times per trial to retain the best outcome.

For FL settings, we assume a conceptual system with $K = 40$ clients. Following common practice in prior GIA studies that target single-round participation [12, 13, 42], we use a held-out test split as private client data and perform GIA on the corresponding gradients or updates, emulating an attacker targeting a newly participating client at an arbitrary training stage. At each attack instance, a conceptual client samples a mini-batch of size B from the test set and performs E steps of local SGD on this batch before sharing the resulting gradient or accumulated update with the server. When $E = 1$, the shared signal reduces to the single-step per-client gradient. Each attack instance independently samples both the client and its mini-batch, without assuming persistent client identities or fixed client datasets.

We note that the baseline FedLeak [12] also has a released protocol that differs from the Breaching setup used in our study. For consistent comparison across all considered attacks and defenses, our main results evaluate all methods under the same Breaching setup. Separately, we reproduce FedLeak under its released implementations [11] in Appendix B by running the authors' code without modification.

[1] https://github.com/JonasGeiping/breaching

[2] https://github.com/JonasGeiping/invertinggradients

Dataset(Resolutions) & Metrics		Ours	FedLeak	HFGradInv	AGIC	SAPAG	InvG	iDLG
CelebA-HQ ($224 \times 224 \times 3$)	PSNR ↑	**20.593**	11.107	14.843	10.962	9.918	12.632	13.325
	LPIPS ↓	**0.445**	0.712	0.608	1.153	1.230	0.902	0.957
	SSIM ↑	**0.549**	0.269	0.382	0.282	0.193	0.345	0.306
ImageNet ($224 \times 224 \times 3$)	PSNR ↑	**20.460**	11.919	15.579	11.367	10.679	14.412	13.488
	LPIPS ↓	**0.418**	0.772	0.597	1.076	1.182	0.771	0.838
	SSIM ↑	**0.540**	0.242	0.357	0.283	0.222	0.359	0.325
LC25000 ($224 \times 224 \times 3$)	PSNR ↑	**21.212**	12.831	11.760	11.367	8.451	14.210	12.093
	LPIPS ↓	**0.274**	0.673	0.509	1.000	1.210	0.646	0.826
	SSIM ↑	**0.562**	0.248	0.258	0.281	0.181	0.361	0.306
CIFAR-100 ($32 \times 32 \times 3$)	PSNR ↑	**45.671**	12.127	16.718	23.327	10.655	22.128	25.860
	LPIPS ↓	**0.016**	0.273	0.084	**0.016**	0.299	0.064	**0.016**
	SSIM ↑	**0.889**	0.189	0.617	0.831	0.260	0.741	0.841
CIFAR-10 ($32 \times 32 \times 3$)	PSNR ↑	**47.737**	14.048	16.516	22.613	11.376	22.650	25.495
	LPIPS ↓	0.026	0.243	0.121	**0.022**	0.350	0.056	0.069
	SSIM ↑	**0.894**	0.275	0.524	0.807	0.259	0.785	0.779

Table 1: Quantitative comparisons (Batch Size = 1) on ResNet-18 with Kaiming-Normal initialization [16].

Metrics. Following prior work [40, 42], we use three metrics to evaluate attack performance: peak signal-to-noise ratio (**PSNR**↑), perceptual image similarity (**LPIPS**↓), and structural similarity index measure (**SSIM**↑). PSNR (↑) measures pixel-wise distortion. LPIPS(↓) is a perceptual similarity metric that measures the distance between deep feature representations of two images, where lower values indicate higher perceptual similarity. SSIM (↑) measures the similarity of overall image structure by jointly considering luminance, contrast, and structural components.

Implementations Details. Consistent with other baselines, each stage is also optimized for $T_1 = T_2 = 30{,}000$ iterations, with the best intermediate result from the coarse stage used for the next refinement. In the coarse stage, we use the Adam optimizer with a learning rate of 0.1 with learning rate scheduling and signed-gradient optimization as in InvG [13]. λ_1 is fixed to 0.05 in all experiments and applied with a delayed schedule, activated only in the later phase of the coarse stage ($T \geq 18{,}000$). This follows a coarse-to-fine optimization strategy [42], where early iterations focus on reconstructing dominant content, and regularization is introduced later to refine details. In the fine stage, we apply Adam with a learning rate of 0.01, scheduled by a cosine strategy from epoch 10,000. λ_2 is set to the reciprocal of the amount of gradients, and λ_{tv} is calculated via Eq. (3), taking values of 0.0002 for CIFAR-10/100 and 0.005 for ImageNet, LC25000 and CelebA-HQ reconstructions.

5.2 Evaluations across Initialization Variations

We first analyze the initialization sensitivity and lifecycle reconstruction performance of GIAs under batch size 1 and local step 1. This is an ablation setting that reduces confounding factors from multiple samples or local updates [10, 13]. We test all GIAs under four common initialization strategies: **Normal** (Kaiming-Normal [16]), **Uniform** (Kaiming-Uniform [16]), **Orthogonal** [29], and **Transfer** [25]. For the Transfer initialization, model parameters are taken from the model pre-trained on ImageNet.

Susceptible Initializations. We begin with the well-known and challenging **Normal** initialization scenario [34,37], which constitutes the primary focus of this paper. Table 1 reports the average performance of 20 trials across four datasets. Our method consistently outperforms all baselines, achieving up to a 200% gain on CIFAR-10 (25.5 $\rightarrow$ 47.7 dB) and a similar improvement on CIFAR-100 (25.9 $\rightarrow$ 45.7 dB). We present case studies on CelebA-HQ (Fig. 5) and LC25000 (Fig. 5) in this section, while results on ImageNet (Fig. 23) are provided in the appendix. Most methods yield noisy reconstructions, making it difficult for attackers to discern the original content. HFGradInv reduces noise in its reconstructions but collapses contrast and color fidelity, hindering attackers from discerning the original content in darker images (third result from the right in Figure 5). In contrast, our method preserves brightness, contrast, and color temperature, achieving nearly noise-free reconstructions and thereby effectively addressing GIA's sensitivity in this scenario. Our improvements are first evident in face recognition scenarios (Fig. 5). While prior work

Figure 5: Qualitative comparisons on CelebA-HQ (Batch Size = 1) using ResNet-18 with Kaiming-Normal initialization [16] .This figure corresponds to Table 1.

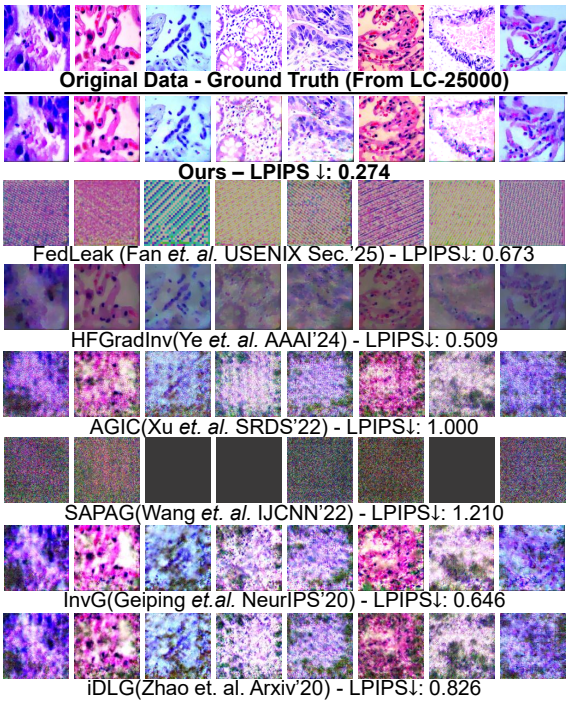


Figure 6: Qualitative comparisons on LC25000 (Batch Size = 1) using ResNet-18 with Kaiming-Normal initialization [16] . This figure corresponds to Table 1.

(e.g., HFGradInv [42]) can recover coarse and blurry facial structures in such sensitive settings, substantial inconsistencies in global appearance—such as brightness, contrast, and color temperature—remain in their reconstructions. These inconsistencies significantly reduce facial recognizability, particularly in low-illumination cases (e.g., the third image from the right in Fig. 5). In contrast, our method produces reconstructions with more consistent global appearance, yielding facial images that are more visually recognizable and therefore indicative of higher privacy leakage risk. This effect is even more pronounced in medical imaging scenarios (Fig. 6). Under well-aligned global appearance factors, our method can clearly delineate lesion boundaries, making clinically relevant pathological structures directly observable.

Initialization Variations. We evaluate robustness under four initialization strategies on CIFAR-10 and CelebA-HQ. Figure 7 compares reconstructions of the same image across methods. Baselines show large performance fluctuations. While Cosine-distance-based GIAs could recover facial features under specific cases (e.g., **Transfer**), they yield unstable or noisy outputs in most cases. HFGradInv and AGIC provide partial improvements through gradient subset selection, but fail to generalize. For instance, HFGradInv produces a misaligned face (first image, third row). In contrast, our method yields consistent reconstructions with far fewer variations across initializations. Table 2 corroborates this result: our approach outperforms all cosine-distance-based GIAs in PSNR, SSIM, and LPIPS under all evaluated initializations, demonstrating strong resilience to initialization variations.

Dataset & Metrics		Ours	HFGradInv	AGIC	InvG
CelebA-HQ	PSNR ↑	**19.468**	17.978	12.802	15.107
	LPIPS ↓	**0.374**	0.395	0.829	0.784
	SSIM ↑	**0.535**	0.508	0.362	0.410
CIFAR-10	PSNR ↑	**41.439**	22.901	26.656	32.917
	LPIPS ↓	**0.009**	0.053	0.011	0.019
	SSIM ↑	**0.950**	0.754	0.874	0.912

Table 2: Quantitative comparisons (Batch Size = 1) averaged over Kaiming-Normal [16], Kaiming-Uniform [16], Orthogonal [29], and Transfer [25] initializations on ResNet-18.

5.3 Evaluations throughout FL's Lifecycle

Since GIA instability in FL is largely driven by increasing model accuracy [13, 37, 40], we evaluate privacy leakage throughout model training, following prior practices [13, 40]. We evaluate on CIFAR-10 and CelebA-HQ at five accuracy checkpoints: initialization, 25%, 50%, 75%, and the maximum (open-sourced trained model). CIFAR-10 uses **Normal** initialization and CelebA-HQ uses **Transfer** initialization. Figures 9 and 10 report reconstruction results: right subfig-

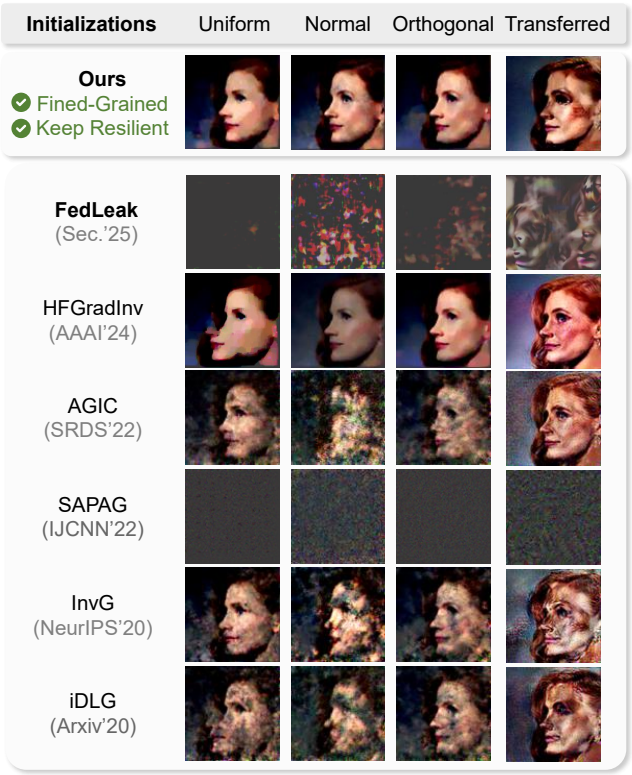


Figure 7: Qualitative comparisons on CelebA-HQ with Batch Size = 1, attacking on ResNet-18 with initializations varying.

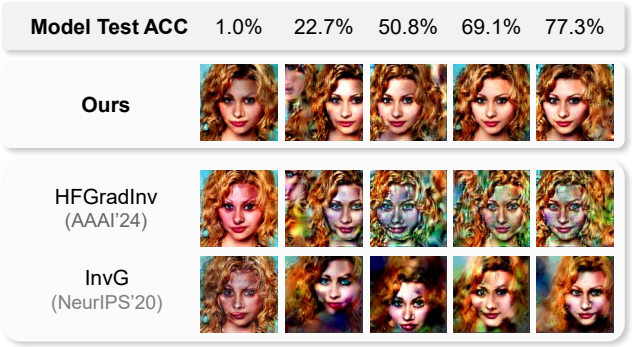
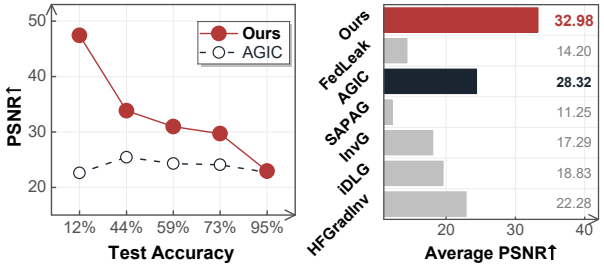


Figure 8: Qualitative comparisons on CelebA-HQ (Batch Size = 1) using ResNet-18 under varying test accuracy, with the two most competitive methods reported in Fig. 10 shown for conciseness.

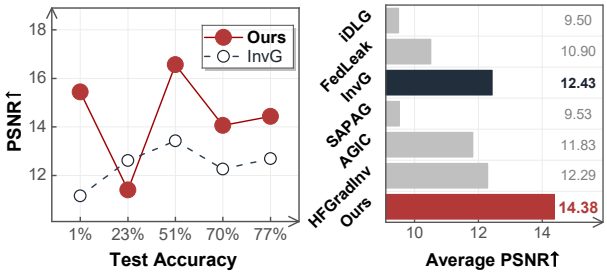
ures show average PSNR across methods, and left subfigures compare our method with the strongest baseline in detail. Our method achieves higher restoration accuracy across both datasets in most cases.

We observe a slight PSNR drop at certain intermediate stages (e.g., CelebA-HQ at 23% accuracy), where PSNR is comparable to or slightly below the strongest baseline. Such behavior is consistent with prior observations that gradient



(a) (b)

Figure 9: Quantitative comparisons on CIFAR-10 (Batch Size = 1) using ResNet-18 under varying test accuracy [7]. (a) Detailed comparison with the strongest baseline. (b) Average performance across all methods.



(a) (b)

Figure 10: Quantitative comparisons on CIFAR-10 (Batch Size = 1) using ResNet-18 under varying test accuracy [25]. (a) Detailed comparison with the strongest baseline. (b) Average performance across all methods.

inversion performance can fluctuate across training stages [10, 37]. During these stages, reconstructions may exhibit minor spatial shifts [44], which are heavily penalized by pixel-wise metrics such as PSNR. This effect is also reflected in our qualitative results (Fig. 8), where reconstructions from both our method and baselines show varying degrees of spatial misalignment across training stages. Importantly, fine-grained facial structures remain well preserved despite these fluctuations, and this advantage also extends to the later training stage (Fig. 24 in appendix). Overall, these results support fine-grained recovery from single-round gradients across different stages of the FL lifecycle.

5.4 Evaluations across Batch Sizes and Steps

We next evaluate performance under various FL training settings, where clients either share averaged gradients from batches (Batch Size $B \geq 1$) or models locally trained for ≥ 1 steps (steps $E \geq 1$). Since GIAs cannot directly access gradients of each private data in these cases, their reconstruction quality typically degrades [34, 38, 43]. Experiments are conducted on CelebA-HQ and CIFAR-10 with **Normal** initializa-

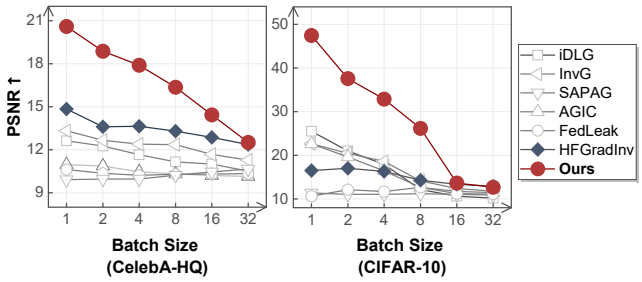


Figure 11: Quantitative comparisons on CelebA-HQ and CIFAR-10 under varying batch sizes, using ResNet-18 with Kaiming-Normal initialization [16].

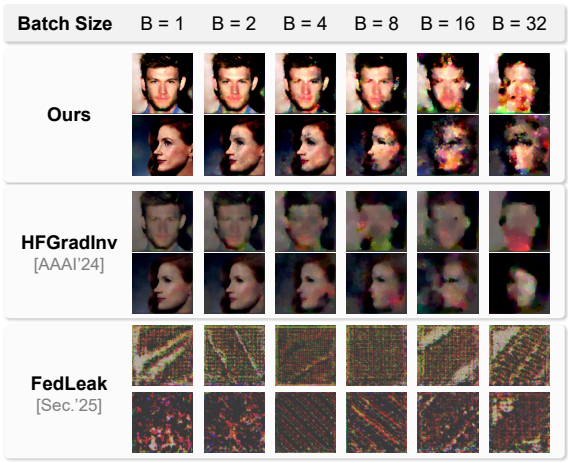


Figure 12: Qualitative comparisons on CelebA-HQ under varying batch sizes, using ResNet-18 with Kaiming-Normal initialization [16]. For conciseness, only the two most competitive methods reported in Fig. 11 are shown.

tion, optimized by SGD with learning rate 0.0001 to simulate multi-step local updates sharing as prior arts [40].

Batch Size Variations. Fig. 11 reports PSNR as B varies. Our method achieves improvements over prior baselines when ($B \leq 8$). As the batch size further increases ($B \geq 16$), the performance margin diminishes, and our approach remains comparable to the best-performing baseline.

This trend is expected: GIA tasks become increasingly difficult when gradients are aggregated over more samples [13], since per-sample signals must be disentangled from averaged updates [48]. While side information can partially mitigate this effect [15, 44], it is excluded by our threat model. As visualized in Fig. 12, increasing B leads reconstructions to transition from fine-grained details toward coarser, structure-level representations, resulting in a reduced performance margin. This convergence reflects the intrinsic difficulty of GIA under heavy aggregation, rather than the instability of our approach.

A further evaluation of the performances in these harder aggregation regimes ($B = 16, 32$) is provided in our appendix. Table 5 and Table 6 show that, under large batch

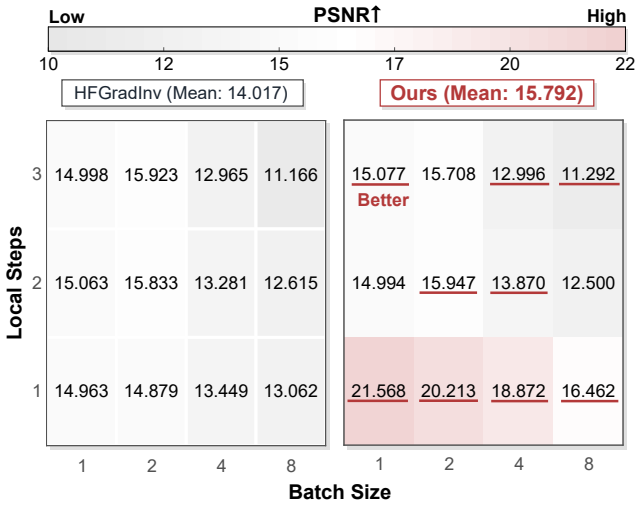


Figure 13: Quantitative comparisons on CelebA-HQ using ResNet-18 with Kaiming-Normal initialization [16], under varying batch size and local steps, comparing against the best-performing baseline from Fig. 11.

sizes ($B = 16, 32$), the achievable PSNR decreases relative to small-batch settings, and the performance differences between methods become less pronounced. In this regime, our method achieves PSNR comparable to the strongest baseline, while yielding modest but consistent average gains in LPIPS/SSIM, despite occasional regressions in individual settings. Fig. 21 and 22 provide qualitative case studies on CelebA-HQ. Despite blurred outputs, our reconstructions retain coherent global structures. Overall, our method remains competitive under batch aggregation without extra assumptions.

Batch Size & Step Variations. Figure 13 further evaluates batched gradients and multi-step local updates on CelebA-HQ ($B \in \{1,2,4,8\}$ and $E \in \{1,2,4,8\}$) with **Normal** initialization. Our method achieves a higher average PSNR than the strongest baseline (15.79 vs. 14.02), while PSNR decreases with larger B or E for all methods due to stronger aggregation.

5.5 Evaluations across Models and Defenses

Model Variations. Although our main evaluations are conducted on ResNet-18, we additionally examine whether similar leakage behaviors arise across other commonly used FL architectures. Figure 15 reports results on MobileNet-v2, ResNet-34, ResNet-18, and DenseNet-121, all initialized with **Normal** [16]. Our method successfully reconstructs recognizable facial features on ResNet-18/34, and exhibits clear leakage on MobileNet-v2. Even on DenseNet-121, which yields the lowest PSNR among the evaluated models, our reconstructions still preserve coarse structural information of the private data. These results suggest that gradient-based leakage is not confined to a specific architecture, but can man-

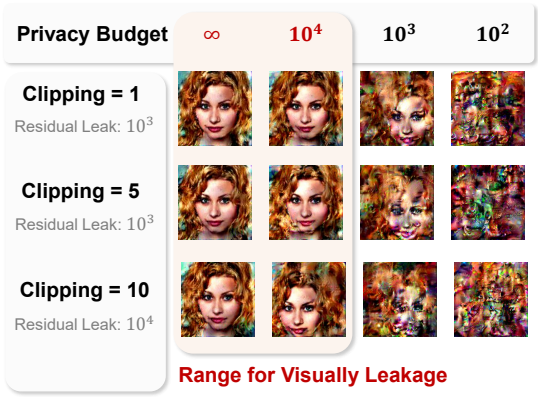


Figure 14: Defenses evaluations: qualitative comparisons on CelebA-HQ (Batch Size = 1) using ResNet-18 (test accuracy 53%) under Differential Privacy [1] , with varying clipping bounds (FClip [24]) and privacy budgets ε.

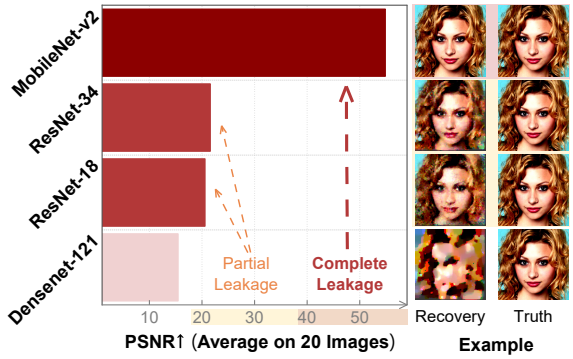


Figure 15: Models evaluations: comparisons on CelebA-HQ (Batch Size = 1) using several popular models with Kaiming-Normal initialization [16].

ifest across diverse model designs commonly adopted in FL.

Defenses Evaluations. Motivated by the underestimated leakage we observe, we evaluate several widely studied privacy-preserving defenses [4, 10]. Our goal is to assess whether these defenses provide sufficient protection against gradient leakage. We focus on the single-sample gradient setting (batch size $B = 1$) for this sanity check, where leakage is strongest and where our attack shows the largest improvements in our experiments. Prior work also evaluates defenses under $B = 1$ as a worst-case setting [32, 34]. As batch size increases, GIA performance tends to degrade, and defense effectiveness is expected to increase [10, 34], which is also consistent with our observations.

DP Evaluations. We observe that clipping substantially reduces reconstruction quality at early stages, while its impact diminishes at later stages. Figure 25 in Appendix illustrates the effect of flat clipping (FClip) with different clipping bounds across training stages. This behavior is consistent with prior findings that gradient norms are typically larger early in

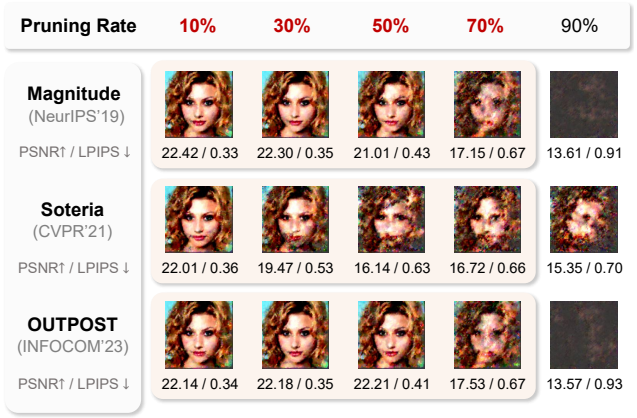


Figure 16: Defense evaluations: comparisons on CelebA-HQ (Batch Size = 1) using ResNet-18 with Kaiming-Normal initialization. Gradient pruning defenses (Magnitude [52], Soteria [32], OUTPOST [34]) are evaluated under varying pruning rates.

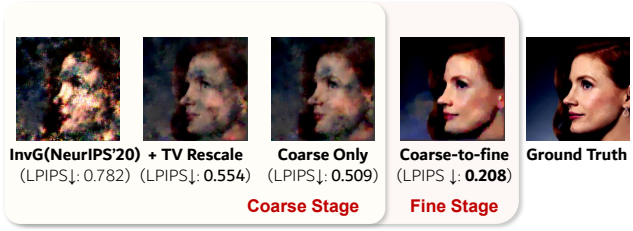


Figure 17: Ablation study: qualitative comparisons on CelebA-HQ (Batch Size = 1), using ResNet-18 initialized with Kaiming-Normal initialization [16].

training, so clipping is triggered more frequently and distorts the shared signal more strongly [17]. As training progresses and gradient norms shrink, clipping is activated less often, making it insufficient as a standalone defense in our setting.

We further vary the privacy budget in an intermediate stage (Fig. 14). Even with noise addition, we observe residual leakage of facial attributes when $\varepsilon \leq 10^3$, and visually concerning leakage up to $\varepsilon \leq 10^4$, consistent with prior observations on visually recognizable leakage ranges [12]. Overall, DP reduces but does not eliminate gradient-based leakage, reflecting the inherent privacy–utility trade-off.

Gradient Pruning Evaluations. We also evaluate three pruning-based defenses in Fig. 16: Magnitude [52], Soteria [32], and OUTPOST [34]. Using the pruning rates suggested by the original papers, we find that reconstructions can remain visually recognizable at moderate pruning levels. In particular, under pruning rates up to 70% (global pruning in our implementation), our method still exhibits non-negligible leakage for Magnitude and Soteria. OUTPOST provides stronger suppression in our experiments, although its protection weakens as training progresses in our setting.

5.6 Ablation Study

We evaluate each component's contribution to enhancing GIA through stepwise modifications of prior work [13]. Quantitative results are in Table 3, with a case study in Fig. 17.

1. Adaptive Scaling TV Loss (regularizing by Eq. (3)): Adaptive scaling TV Loss reduces noise production in reconstructions. As shown in Table 3 and Fig. 17, PSNR increases from 13.325 to 14.821.

2. Coarse stage GIA (using objective in Eq. (1)): Directional aligning with Eq. (1) improves the alignments in key facial features and overall structures of the image (i.e., image structure, contrast). Table 3 shows improvements in LPIPS and SSIM, with minimal PSNR loss. With an increase in the structural scores, this method highlights the reconstruction quality of key content, and stabilizes the reconstruction quality under scenarios suffering strong gradient sparsity.

3. Coarse-to-fine GIA (adding Fine Stage): The fine stage produces the largest PSNR improvement (14.716 $\rightarrow$ 20.593), enabling fine-grained recovery. It also improves LPIPS (0.445) and SSIM (0.455), enhancing perceptual quality and maintaining structural integrity, demonstrating its effectiveness in providing a fine-grained recovery.

Method	PSNR ↑	LPIPS ↓	SSIM ↑
InvG [13]	13.325	0.902	0.382
+ TV rescale (by Eq. (3))	14.821	0.664	0.388
Coarse Stages GIA	14.716	0.654	0.397
Coarse-to-fine GIA	**20.593**	**0.445**	**0.455**

Table 3: Ablation Study: quantitative comparison on CelebA-HQ (Batch Size = 1) using ResNet-18 with Kaiming-Normal initialization [16].

6 Limitations and Future Works

While this work demonstrates a stronger privacy leakage through GIA in practical FL settings, it also highlights the boundaries of what current methods can achieve under certain realistic FL conditions. These limitations point to important areas for further research.

Limitations: Leakage in Limited Training Settings. In FL, clients typically share batch-averaged gradients ($B \geq 1$) or updates with multiple local steps ($E \geq 1$). Although our method remain competitive performances under these settings, reconstruction quality degrades substantially for both GIA methods . This suggests that gradient leakages are most pronounced in computation-constrained scenarios (e.g., personalized FL or mobile devices with limited data per round), while attackers may face difficulties in large-scale FL systems with heavier local training and aggregation, limiting applicability in enterprise-level deployments.

Limitations: Sensitivity to Model Architecture. Though our method reveals persistent gradient leakage on several popular models, its reconstruction quality varies across model structures and can degrade or fail for certain architectures, indicating that attack effectiveness is architecture-dependent. This variability may limit consistency across different FL setups, motivating future work on improving robustness to architectural differences.

Future directions. Rather than diminishing the contribution of this work, these limitations highlight important research challenges and open the door for future advancements: (i) Develop stronger attacks that are robust to various training settings, especially for larger batch sizes and steps in large-scale FL systems. (ii) Improve GIA's resilience to model structure variations, enabling more consistent and reliable privacy leakage across models and architectures. (iii) Leverage insights from these limitations to design more effective privacy defenses that address the leakage from gradient, ultimately strengthening privacy-preserving mechanisms in FL.

7 Conclusion

We presented a coarse-to-fine gradient inversion attack that integrates rescaled total variation and non-zero component alignment to improve stability and fidelity of reconstructions. Our method achieves initialization-resilient and fine-grained recovery in both early-stage and across the FL lifecycle under observations to small batch gradients, consistently outperforming prior approaches. Our experiments exposing privacy risks across several common FL models, and highlight the privacy risk for insufficient defense adopting.

At the same time, our evaluation shows that leakage diminishes rapidly when observing to gradients in larger batch sizes and updates trained for multiple local steps, indicating that the strongest risks are concentrated in computation-constrained settings. By demonstrating both lifecycle-wide recovery under gradient observations and the limitations of its applications, our study advances the understanding of when and how gradient leakage arises, and highlights the vulnerable scenarios for strong leakage persisting. Our implementation and evaluation framework would be released for promoting further researches.

8 Acknowledgments

The authors are with Guangdong Provincial Key Laboratory of Fire Science and Intelligent Emergency Technology. This work was supported by the National Natural Science Foundation of China NO. 62072485 and Guangdong Basic and Applied Basic Research Foundation No. 2025A1515010248.

The author thank the authors of AGIC [40] for kindly sharing their implementation, which facilitated our experimental evaluation.

Ethical Considerations

We adhered to the USENIX Security Ethics Guidelines in conducting this research. All experiments were performed exclusively on publicly available datasets, including CelebA-HQ, ImageNet, CIFAR-10, and CIFAR-100. No private, sensitive, or real-world user data was accessed or processed, ensuring that no individual was directly affected by our work.

All evaluations were carried out in simulated FL environments. We did not deploy the proposed attacks on production systems, nor did we attempt to extract data from operational services. Our methodology is limited to controlled experimental settings, which prevents any potential harm to real users or organizations.

Instead of proposing specific countermeasures, by demonstrating an improved attack, we aim to motivate the development of more robust defenses in future work. We systematically study the stability of GIAs to provide a clearer understanding of realistic privacy risks. We believe that disclosing these findings is necessary to raise awareness within the community and to highlight scenarios where current defenses—specifically Differential Privacy and Gradient Pruning—may be insufficient if applied with weak perturbation parameters.

We further consider the implications of our findings for key stakeholders, including developers, policymakers, and users. For developers: Our results indicate that reliance on the natural instability of gradients during the early training stages (initialization) offers a false sense of security. Consequently, deploying FL without rigorous privacy auditing—even in the initial epochs—is risky. For policymakers and users: It is crucial to recognize that while FL mitigates data centralization risks, it does not strictly guarantee privacy against advanced gradient leakage in its current form. We recommend that FL systems should be deployed with caution and must be paired with sufficiently strong defense mechanisms to protect participants from the potential privacy breaches highlighted in this study.

Open Science

To support transparency and reproducibility, we archive our implementation on Zenodo [51]. The release includes artifact instructions, dependencies, datasets, models, initialization schemes, attacks, and defenses used in our experiments. All materials are complied with the USENIX open science policy and are made publicly available to facilitate independent validation.

References

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

[2] Mislav Balunovic, Dimitar Iliev Dimitrov, Robin Staab, and Martin Vechev. Bayesian framework for gradient leakage. In International Conference on Learning Representations, 2022.

[3] Andrew A Borkowski, Marilyn M Bui, L Brannon Thomas, Catherine P Wilson, Lauren A DeLand, and Stephen M Mastorides. Lung and colon cancer histopathological image dataset (lc25000). arXiv preprint arXiv:1912.12142, 2019.

[4] Vincenzo Carletti, Pasquale Foggia, Carlo Mazzocca, Giuseppe Parrella, and Mario Vento. {SoK}: Gradient inversion attacks in federated learning. In 34th USENIX Security Symposium (USENIX Security 25), pages 6439–6459, 2025.

[5] Huancheng Chen and Haris Vikalo. Recovering labels from local updates in federated learning. In Forty-first International Conference on Machine Learning, 2024.

[6] Jacob Cohen. Statistical Power Analysis for the Behavioral Sciences. Lawrence Erlbaum Associates, Hillsdale, NJ, 2 edition, 1988.

[7] Eduardo Dadalto. edadaltocg/detectors: v0.1.14. https://doi.org/10.5281/zenodo.7883596, January 2024.

[8] 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, 2009.

[9] Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. In Proceedings of the 33rd International Conference on Neural Information Processing Systems, pages 10542–10552, 2019.

[10] Jiacheng Du, Jiahui Hu, Zhibo Wang, Peng Sun, Neil Zhenqiang Gong, Kui Ren, and Chun Chen. Sok: On gradient leakage in federated learning. In 34th USENIX Security Symposium (USENIX Security 25), 2025.

[11] Mingyuan Fan. Code of fedleak. https://doi.org/10.5281/zenodo.15532456, May 2025.

[12] Mingyuan Fan, Fuyi Wang, Cen Chen, Jianying Zhou, and Reviewing Model. Boosting gradient leakage attacks: Data reconstruction in realistic fl settings. In 34th USENIX Security Symposium (USENIX Security 25), 2025.

[13] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients-how easy is it to break privacy in federated learning? In Advances in neural information processing systems, volume 33, pages 16937–16947, 2020.

[14] Pengxin Guo, Runxi Wang, Shuang Zeng, Jinjing Zhu, Haoning Jiang, Yanran Wang, Yuyin Zhou, Feifei Wang, Hui Xiong, and Liangqiong Qu. Exploring the vulnerabilities of federated learning: A deep dive into gradient inversion attacks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2025.

[15] Ali Hatamizadeh, Hongxu Yin, Holger R Roth, Wenqi Li, Jan Kautz, Daguang Xu, and Pavlo Molchanov. Gradvit: Gradient inversion of vision transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10021–10030, 2022.

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

[17] Jiahui Hu, Jiacheng Du, Zhibo Wang, Xiaoyi Pang, Yajie Zhou, Peng Sun, and Kui Ren. Does differential privacy really protect federated learning from gradient leakage attacks? IEEE Transactions on Mobile Computing, 23(12):12635–12649, 2024.

[18] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4700–4708, 2017.

[19] Bargav Jayaraman and David Evans. Evaluating differentially private machine learning in practice. In 28th USENIX security symposium (USENIX security 19), pages 1895–1912, 2019.

[20] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. Foundations and trends® in machine learning, 14(1–2):1–210, 2021.

[21] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.

[22] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In Proceedings of International Conference on Computer Vision (ICCV), December 2015.

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

[24] H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. arXiv preprint arXiv:1710.06963, 2017.

[25] Dongbin Na, Sangwoo Ji, and Jong Kim. Unrestricted black-box adversarial attack using gan with limited queries. In European Conference on Computer Vision, pages 467–482. Springer, 2022.

[26] Ngoc-Bao Nguyen, Keshigeyan Chandrasegaran, Milad Abdollahzadeh, and Ngai-Man Cheung. Rethinking model inversion attacks against deep neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 16384–16393, 2023.

[27] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 11410–11420, 2022.

[28] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4510–4520, 2018.

[29] A Saxe, J McClelland, and S Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In Proceedings of the International Conference on Learning Represenatations 2014. International Conference on Learning Represenatations 2014, 2014.

[30] Daniel Scheliga, Patrick Mäder, and Marco Seeland. Dropout is not all you need to prevent gradient leakage. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 37, pages 9733–9741, 2023.

[31] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In Proceedings of the 22nd ACM SIGSAC conference on computer and communications security, pages 1310–1321, 2015.

[32] 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.

[33] Vale Tolpegin, Stacey Truex, Mehmet Emre Gursoy, and Ling Liu. Data poisoning attacks against federated learning systems. In European symposium on research in computer security, pages 480–501. Springer, 2020.

[34] Fei Wang, Ethan Hugh, and Baochun Li. More than enough is too much: Adaptive defenses against gradient leakage in production federated learning. In IEEE INFOCOM 2023-IEEE Conference on Computer Communications, pages 1–10. IEEE, 2023.

[35] Fei Wang and Baochun Li. Data reconstruction and protection in federated learning for fine-tuning large language models. IEEE Transactions on Big Data, 2024.

[36] Junbo Wang, Amitangshu Pal, Qinglin Yang, Krishna Kant, Kaiming Zhu, and Song Guo. Collaborative machine learning: Schemes, robustness, and privacy. IEEE Transactions on Neural Networks and Learning Systems, 34(12):9625–9642, 2022.

[37] Yijue Wang, Jieren Deng, Dan Guo, Chenghong Wang, Xianrui Meng, Hang Liu, Chao Shang, Binghui Wang, Qin Cao, Caiwen Ding, et al. Variance of the gradient also matters: Privacy leakage from gradients. In 2022 International Joint Conference on Neural Networks (IJCNN), pages 1–8. IEEE, 2022.

[38] Wenqi Wei, Ling Liu, Margaret Loper, Ka-Ho Chow, Mehmet Emre Gursoy, Stacey Truex, and Yanzhao Wu. A framework for evaluating client privacy leakages in federated learning. In Computer Security–ESORICS 2020: 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14–18, 2020, Proceedings, Part I 25, pages 545–566. Springer, 2020.

[39] Haoran Xie, Yujue Wang, Yong Ding, Changsong Yang, Haibin Zheng, and Bo Qin. Verifiable federated learning with privacy-preserving data aggregation for consumer electronics. IEEE Transactions on Consumer Electronics, 70(1):2696–2707, 2023.

[40] Jin Xu, Chi Hong, Jiyue Huang, Lydia Y Chen, and Jérémie Decouchant. Agic: Approximate gradient inversion attack on federated learning. In 2022 41st International Symposium on Reliable Distributed Systems (SRDS), pages 12–22. IEEE, 2022.

[41] Xiangrui Xu, Pengrui Liu, Wei Wang, Hong-Liang Ma, Bin Wang, Zhen Han, and Yufei Han. Cgir: Conditional generative instance reconstruction attacks against federated learning. IEEE Transactions on Dependable and Secure Computing, 20(6):4551–4563, 2022.

[42] Zipeng Ye, Wenjian Luo, Qi Zhou, and Yubo Tang. High-fidelity gradient inversion in distributed learning. In Proceedings of the AAAI Conference on Artificial Intelligence, pages 19983–19991, 2024.

[43] Zipeng Ye, Wenjian Luo, Qi Zhou, Zhenqian Zhu, Yuhui Shi, and Yan Jia. Gradient inversion attacks: Impact factors analyses and privacy enhancement. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024.

[44] 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.

[45] Chengliang Zhang, Suyi Li, Junzhe Xia, Wei Wang, Feng Yan, and Yang Liu. BatchCrypt: Efficient homomorphic encryption for Cross-Silo federated learning. In 2020 USENIX Annual Technical Conference (USENIX ATC 20), pages 493–506. USENIX Association, July 2020.

[46] Chi Zhang, Zhang Xiaoman, Ekanut Sotthiwat, Yanyu Xu, Ping Liu, Liangli Zhen, and Yong Liu. Generative gradient inversion via over-parameterized networks in federated learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 5126–5135, 2023.

[47] Haobo Zhang, Junyuan Hong, Yuyang Deng, Mehrdad Mahdavi, and Jiayu Zhou. Understanding deep gradient leakage via inversion influence functions. Advances in neural information processing systems, 36:3921–3944, 2023.

[48] Rui Zhang, Song Guo, Junxiao Wang, Xin Xie, and Dacheng Tao. A survey on gradient inversion: Attacks, defenses and future directions. In International Joint Conference on Artificial Intelligence, volume 6, pages 5678–5685, july 2022.

[49] Zhuangzhuang Zhang, Libing Wu, Chuanguo Ma, Jianxin Li, Jing Wang, Qian Wang, and Shui Yu. Lsfl: A lightweight and secure federated learning scheme for edge computing. IEEE Transactions on Information Forensics and Security, 18:365–379, 2022.

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

[51] Kaiming Zhu. Coarse-to-fine gradient inversion attack. https://doi.org/10.5281/zenodo.17979356, December 2025.

[52] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In Advances in neural information processing systems, volume 32, 2019.

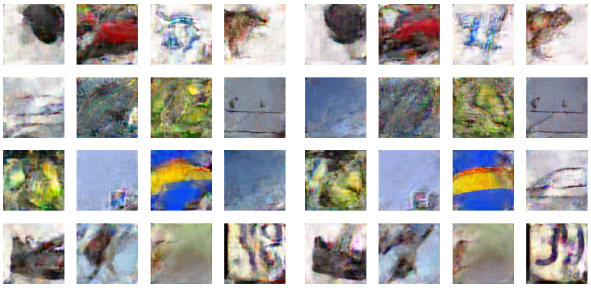
A Implementations Details

In our experiments, all attacks with available implementations are adapted to or integrated into the Breaching framework, ensuring that all GIAs are conducted and evaluated under a unified and consistent setup. For **HFGradInv** [42] and **InvG** [13], we reuse their implementations, as both are provided within the Breaching codebase. For **AGIC** [40], we adapt authors' shared implementation from InverseFed (an earlier version of Breaching) into the uniformed framework. For **iDLG** [50], we integrate from the public implementation and replace the original L-BFGS optimizer with Adam to improve optimization stability when attacking deeper models such as ResNet-18, following common practice in prior work [13, 40]. For **SAPAG** [37], we implement the method based on the algorithm described in the paper, as no public implementation was available at the time of our study.

FedLeak [12]. We use its publicly released Zenodo implementation (v1) [11]. Unlike other baselines, FedLeak is GAN-based: the attacker optimizes the generator $G(\hat{z};\hat{\theta})$ to produce reconstructions $\hat{x}$, rather than directly optimizing $\hat{x}$ as Breaching supported. Fully integrating FedLeak into Breaching would require framework-level changes that could affect other baselines. We therefore keep the original FedLeak implementation and apply only minimal interface adaptations to run it within the Breaching evaluation pipeline (e.g., same models and shared gradients) while preserving its attack design. All runs follow the released FedLeak behavior, with the following two adaptations applied for compatibility:

Adaptions 1. Resolution-consistent evaluation. FedLeak generates at $2\times$ lower resolution and upsamples for comparisons (e.g., $128\times128\times3 \rightarrow 256\times256\times3$) [11]. Breaching computes metrics without resizing and requires reconstructions at the target resolution. We therefore remove FedLeak's upsampling step and generate directly at the target resolution.

Adaptions 2. Supporting our target resolutions. FedLeak generates in a fixed set of image resolutions (i.e., $\{32, 64, 128, 256\}$) [11], whereas our evaluation includes $224\times224\times3$ datasets (ImageNet, CelebA-HQ, LC25000). We therefore extend FedLeak to output the target resolution while preserving the original generations in the resolution 32. For $224\times224\times3$, it generates $256\times256\times3$ and deterministically rescales to $224\times224\times3$ for gradient alignment and metric evaluation. We consider two commonly used GAN rescale strategies, namely resize [27] and center-crop [9]. We evaluate both strategies under the FedLeak evaluation setup using the images from the original FedLeak release. We also evaluate on LC-25000 and ImageNet, as in FedLeak [12]. For each dataset, we use two sampling schemes (first five mini-batches and five random mini-batches). This yields 21 instances with batch size $B = 16$; following FedLeak, we select the variant with higher PSNR. Qualitative results are shown in Fig. 18 and quantitative results in Table 4. We use Resize [27] as the default.



(a) Resize [27] (b) Center-crop [9]

Figure 18: Downsampling strategies: evaluation on the FedLeak release images (batch size = 16, best-of-5) [11].

Strategies	Best ↑	Median ↑	Average ↑
Resize [27]	13.021	12.470	12.357
Center-crop [9]	12.929	12.283	12.201

Table 4: Downsampling strategies. Quantitative comparison (Batch Size = 16, best-of-5) in terms of PSNR ↑ under three statistics. Metrics are computed with 5 independent retries per experiment and averaged over 21 evaluation instances.

B FedLeak Released-setup Reproduction

This appendix reports additional results under the original FedLeak evaluation setup using the authors' released implementation [11]. We summarize two observed differences between the FedLeak setup (this appendix) and the Breaching setup (main experiments), consistent with prior discussions [10, 35].

Implementations. Unless stated otherwise, we follow the protocol in Sec. 5.1 for our method. FedLeak results are produced by running the authors' released code [11]. When we report best-of-k, we use the same restart budget and selection rule for all methods (including FedLeak) for consistent reporting.

Evaluations. Under the FedLeak setup, FedLeak reconstructs substantially better than it does in our main experiments (Breaching setup). Our method is weaker than FedLeak in this setup, but still reveals recognizable semantic content. Figure 19 illustrates this with qualitative comparisons on the 16 images provided in the released FedLeak codebase [11], using batch size $B = 16$ and best-of-5 reporting.

To further evaluate both methods under this setup, we run additional experiments on LC-25000 and ImageNet using 10 randomly sampled batches per dataset. For these experiments, we use $B = 16$ and follow the FedLeak reporting protocol (single run per batch) [11]. Figure 20 summarizes the comparisons on our representative failure cases (i.e., the highest-LPIPS ↓ reconstructions for our method). Despite lower reconstruction quality, our method still reveals the edge of recognizable semantic content across the sampled batches.

(a) FedLeak [12] (b) Ours

Figure 19: FedLeak Released-setup: evaluation on the its release images [11] (batch size = 16, best-of-5).

These results suggest that GIA performance can vary across evaluation setups, consistent with prior observations of setup-dependent effects [10, 35]. We therefore summarize two observed differences between the FedLeak setup (this appendix) and the Breaching setup (main experiments). These setup differences can change gradient computation and are plausible contributors to the cross-setup performance gap.

Setup differences. In additional investigation, we found that the FedLeak setup is also adopted in some prior GIA studies [46, 52]. Compared to the Breaching setup used in our main experiments, we identify two differences in model and execution in the FedLeak setup:

- **Layer-level**: Both setups use the same backbone architecture, but the FedLeak setup additionally changes it by replacing ReLU activations with Sigmoid [11, 46].
- **Execution-level**: The FedLeak setup evaluates gradients in *train* mode [11, 46], whereas the Breaching-based setup adopted by our baselines typically evaluates in the *eval* mode [13, 40, 42].

Consistent with prior observations [10, 14, 35], these factors can change gradient computation in GIA, affecting both the victim's generation of shared gradients and the attacker's minimization of gradient distance. As a result, GIA performance can differ across setups that vary in these factors.

On the **layer level**, swapping ReLU for Sigmoid changes the gradient map (zeroing negatives vs. saturation at large magnitudes), and prior work reports that such activation changes can affect inversion behavior [10, 14]. On the **execution level**, for mode-dependent layers (e.g., Batch Normalization and Dropout), switching between *train* and *eval* mode changes forward propagation (output) and backward propagation (gradients), which can lead to different inversion outcomes across modes [30, 35].

Takeaway. We reproduce results under the released FedLeak setup by running the authors' implementation without modification [11]. Under this setup, both FedLeak and our method

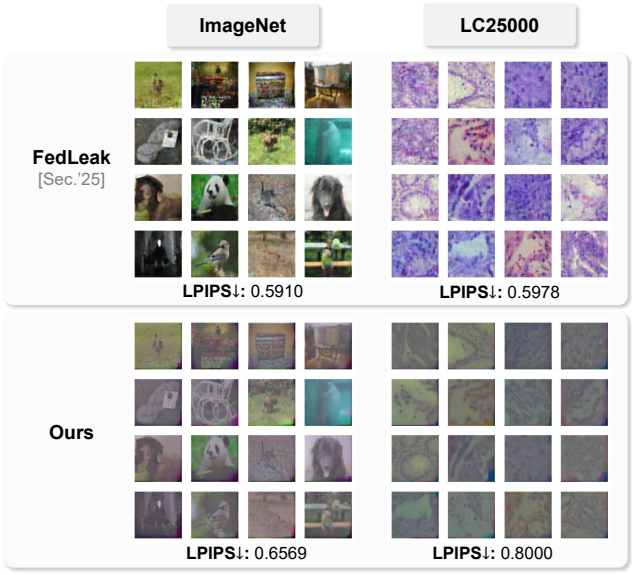


Figure 20: Cross-setup evaluation on LC-25000 and ImageNet with 10 randomly sampled batches per dataset (Batch Size = 16, one run per batch). We report our worst-case reconstructions, i.e., those with the highest LPIPS ↓.

behave differently than in our main Breaching-based experiments. Our method still recovers recognizable semantic content under both setups, but reconstruction quality varies across setups. This underscores setup sensitivity in GIA arising from deployment-level choices in FL (i.e. sensitivity to specific layers introduced [14, 30, 35]). Together with the main experiments, we include these results to show how GIA performance varies across different setups. This highlights its sensitivity to such variations and motivates future work on improving robustness.

C Experiment Results

Dataset	Methods	Average	Uniform [16]	Normal [16]	Orthogonal [29]	Transfer [25]
CIFAR-10	**Ours**	10.93 / **0.21** / **0.36**	8.92 / 0.23 / 0.31	13.63 / 0.16 / 0.44	11.27 / 0.22 / 0.33	9.90 / 0.21 / 0.34
	HFGradInv	11.81 / 0.24 / 0.32	10.74 / 0.26 / 0.29	11.83 / 0.24 / 0.32	11.33 / 0.20 / 0.38	13.36 / 0.27 / 0.28
	FedLeak	**12.35** / 0.40 / 0.15	12.87 / 0.53 / 0.05	11.76 / 0.25 / 0.26	12.75 / 0.50 / 0.06	12.04 / 0.31 / 0.23
LC-25000	**Ours**	10.34 / 0.73 / **0.27**	8.11 / 0.97 / 0.24	12.39 / 0.62 / 0.26	10.67 / 0.71 / 0.28	10.19 / 0.62 / 0.28
	HFGradInv	9.81 / 0.81 / 0.24	7.73 / 1.05 / 0.24	11.12 / 0.72 / 0.17	10.20 / 0.76 / 0.27	10.22 / 0.71 / 0.27
	FedLeak	**11.18** / **0.69** / 0.22	10.34 / 0.82 / 0.11	11.53 / 0.68 / 0.26	10.50 / 0.70 / 0.24	12.34 / 0.57 / 0.27
CelebA-HQ	**Ours**	13.21 / **0.65** / **0.36**	14.20 / 0.61 / 0.37	14.43 / 0.60 / 0.40	14.67 / 0.67 / 0.37	9.53 / 0.72 / 0.28
	HFGradInv	**13.36** / 0.66 / 0.34	14.57 / 0.60 / 0.38	13.36 / 0.67 / 0.29	15.31 / 0.65 / 0.39	10.19 / 0.73 / 0.29
	FedLeak	11.10 / 0.76 / 0.19	11.41 / 0.86 / 0.05	10.83 / 0.76 / 0.28	11.34 / 0.77 / 0.11	10.81 / 0.66 / 0.30

Table 5: Quantitative comparisons (Batch Size = 16) using ResNet-18 under four initializations (PSNR ↑ / LPIPS ↓ / SSIM ↑).

Dataset	Methods	Average	Uniform [16]	Normal [16]	Orthogonal [29]	Transfer [25]
CIFAR-10	**Ours**	10.44 / **0.20** / **0.35**	8.69 / 0.23 / 0.31	12.68 / 0.15 / 0.41	10.59 / 0.22 / 0.32	9.78 / 0.20 / 0.35
	HFGradInv	11.48 / 0.24 / 0.31	10.45 / 0.24 / 0.31	12.93 / 0.29 / 0.25	11.24 / 0.23 / 0.31	11.28 / 0.20 / 0.36
	FedLeak	**12.22** / 0.39 / 0.15	12.75 / 0.55 / 0.03	11.58 / 0.24 / 0.26	12.63 / 0.48 / 0.06	11.91 / 0.30 / 0.25
LC-25000	**Ours**	12.06 / 0.70 / **0.33**	8.44 / 0.92 / 0.25	12.61 / 0.62 / 0.25	10.57 / 0.72 / 0.28	10.63 / 0.58 / 0.30
	HFGradInv	**12.28** / 0.71 / 0.31	7.62 / 1.04 / 0.24	11.58 / 0.71 / 0.16	10.27 / 0.75 / 0.28	10.32 / 0.71 / 0.28
	FedLeak	11.49 / **0.68** / 0.23	10.00 / 0.84 / 0.11	12.66 / 0.68 / 0.27	10.96 / 0.64 / 0.29	12.36 / 0.57 / 0.26
CelebA-HQ	**Ours**	10.56 / **0.71** / **0.27**	13.10 / 0.64 / 0.35	12.51 / 0.65 / 0.36	13.48 / 0.72 / 0.34	9.15 / 0.79 / 0.26
	HFGradInv	9.95 / 0.80 / 0.24	13.33 / 0.64 / 0.35	12.37 / 0.68 / 0.27	13.88 / 0.71 / 0.35	9.56 / 0.81 / 0.27
	FedLeak	**11.10** / 0.77 / 0.19	11.37 / 0.87 / 0.05	10.86 / 0.76 / 0.29	11.31 / 0.76 / 0.13	10.84 / 0.68 / 0.28

Table 6: Quantitative comparisons (Batch Size = 32) using ResNet-18 under four initializations (PSNR ↑ / LPIPS ↓ / SSIM ↑).

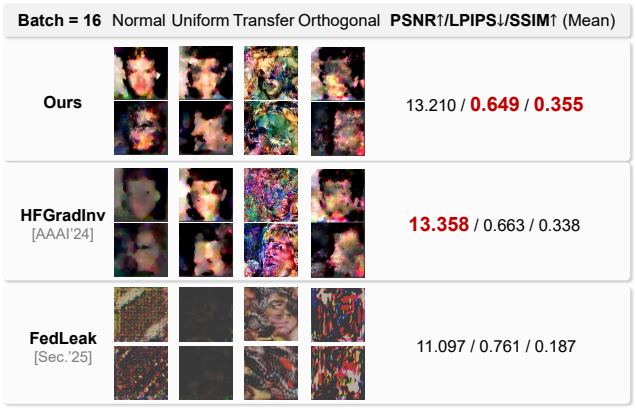


Figure 21: Qualitative comparisons on CelebA-HQ (Batch Size = 16), on ResNet-18 under four initializations, This figure corresponds to Table 5.

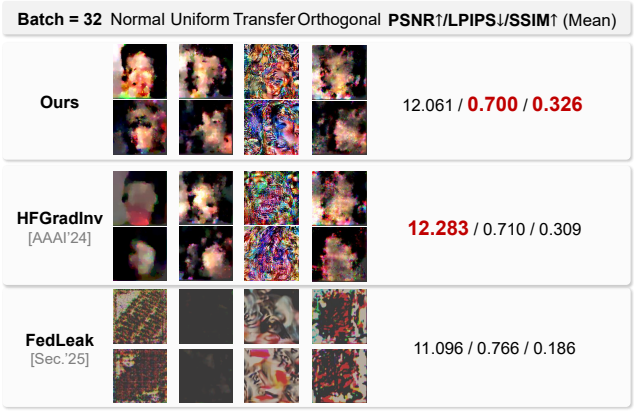


Figure 22: Qualitative comparisons on CelebA-HQ (Batch Size = 32), on ResNet-18 under four initializations, This figure corresponds to Table 6.

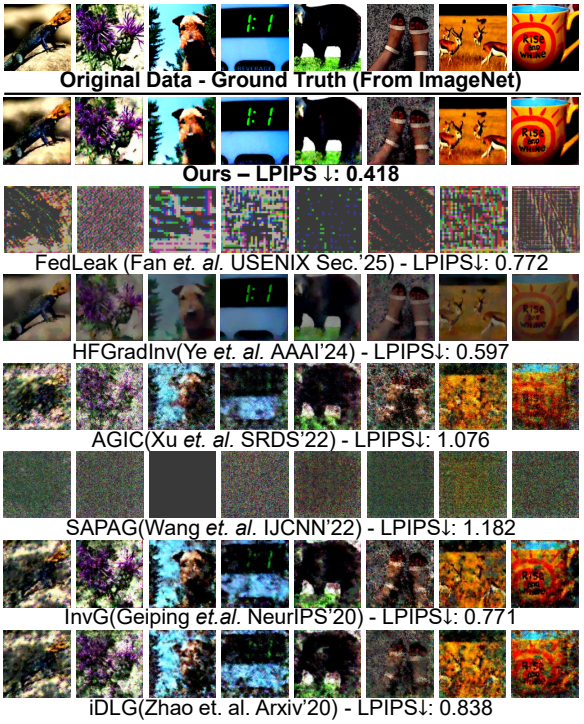


Figure 23: Qualitative comparisons on ImageNet (Batch Size = 1), on ResNet-18 with Kaiming-Normal initialization [16].

Figure 24: Qualitative comparisons on CelebA-HQ (Batch Size = 1) on ResNet-18 with test accuracy 77.3% [25].

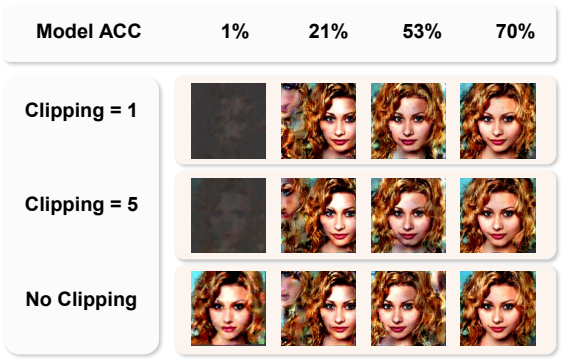


Figure 25: Defenses evaluations: qualitative comparisons on CelebA-HQ (Batch Size = 1) using ResNet-18 under varying test accuracy. We evaluate gradient clipping (FClip) [24] with varying clipping bounds.