Aegis: Generative Gradient Masking for Privacy-Preserving Medical Federated Learning

Chaoyu Zhang[1], Shanghao Shi[2], Heng Jin[1], Ning Wang[3], Y. Thomas Hou[1], Wenjing Lou[1]
[1]Virginia Tech [2]Washington University in St. Louis
[3]University of South Florida

Abstract

Federated learning (FL) has become a foundational paradigm for multi-institutional medical AI, allowing hospitals and research centers to jointly train diagnostic models without exchanging patient records. This privacy promise, however, is increasingly contested: a malicious or honest-but-curious server can launch model inversion attacks (MIAs) that reconstruct private patient images directly from shared model updates, and recent scalable, closed-form attacks penetrate even secure aggregation at clinically realistic batch sizes. Existing defenses face an unsatisfactory dilemma. Gradient-perturbation methods such as differential privacy and pruning trade away the diagnostic accuracy on which clinical reliability depends, while cryptographic protocols add system complexity yet still leave updates exposed to these scalable attacks. We propose Aegis, a principled client-side defense that breaks this dilemma without perturbing patient data or modifying the FL protocol. Our key insight is that the success of every known MIA is fundamentally bounded by the local batch size relative to the model's leakage capacity; once this limit is exceeded, distinct samples collide and reconstructions collapse into indistinguishable mixtures. Aegis turns this universal bottleneck into a defense: each client superimposes onto its real update a masking gradient computed on locally synthesized, task-relevant data, deliberately pushing the effective batch beyond the attack's recovery capacity. We complement the design with theoretical convergence guarantees under standard convex assumptions and evaluate Aegis on MNIST, CIFAR-10, and three MedMNIST modalities (chest X-ray, abdominal CT, colon pathology). Aegis neutralizes three state-of-the-art MIAs while preserving model utility and incurring only modest overhead, offering a practical privacy primitive for medical FL.

1 Introduction

FL has emerged as a foundational paradigm for multi-institutional medical collaboration: hospitals, research centers, and biotech labs can jointly train diagnostic and biomedical models without ever moving raw patient data, because only model updates (weights or gradients) are transmitted to a central server [23]. This decentralization is precisely what makes FL attractive for clinical AI, where data is fragmented across silos governed by HIPAA [25], GDPR [19], and institutional review boards. Yet the privacy guarantee of FL is increasingly contested. MIAs, launched by an honest-but-curious or actively malicious server, can reconstruct private client samples directly from the shared updates. A first generation of optimization-based MIAs [50, 47, 6, 49, 45, 22, 10] iteratively matches dummy gradients to real ones, raising particularly serious concerns when the data being reconstructed are sensitive medical images such as CT slices, chest X-rays, or histopathology tiles.

To mitigate MIAs, existing defenses fall into two categories: gradient perturbation and cryptographic protection. Gradient-perturbation methods, exemplified by differential privacy (DP) [24], inject

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

noise into local gradients to obscure sensitive signal at the cost of model accuracy, a cost that is particularly damaging in medical applications, where small accuracy drops can translate into clinically meaningful errors. Adaptive variants [39, 40] calibrate noise to estimated leakage risk, and pruning-based approaches [50, 38] selectively suppress recoverable gradient components. Cryptographic defenses based on multi-party computation [16] implement secure aggregation protocols [1, 9, 13, 42] so that the server only ever sees aggregated updates.

However, a recent and rapidly growing class of MIAs [48, 36, 41, 26, 5, 35] demonstrates strong reconstruction even *under* secure aggregation. By exploiting the *linear-leakage primitive* [5], these attacks bypass iterative optimization and recover inputs in closed form, scaling to large batches and remaining effective in production FL settings. Related leakage risks also arise in federated language-model fine-tuning [37] and split learning [12]. Consequently, current defenses face an unsatisfactory dilemma: perturbation-based methods sacrifice the model utility on which clinical reliability depends, while cryptographic approaches add system overhead yet do not eliminate the leakage. This gap is critical for medical FL: if linear-leakage MIAs can reconstruct patient scans from shared updates, the privacy promise that justifies federated training in the first place is broken. We therefore set out to design a defense that (i) is effective against scalable, linear-leakage MIAs, (ii) preserves model utility on clinical data, and (iii) requires no modification to existing FL protocols.

Our solution. We introduce Aegis, a generative-data-based defense against MIAs. Our key observation (Tab. A.9; Appendix H) is that *the reconstruction success of every known MIA is bounded by the local batch size relative to model capacity*. Optimization-based MIAs degrade rapidly beyond very small batches; linear-leakage attacks scale further but, by their own mathematical structure, succeed only when the local batch size does not exceed the number of neurons in the first fully connected layer [36]. Once that capacity is exceeded, multiple inputs collide in the same leakage bins and reconstructions collapse into mixed, indistinguishable images.

Aegis turns this fundamental bottleneck into a defense. Each client uses an off-the-shelf generative model to synthesize an auxiliary *defense dataset* large enough that, when combined with the real local batch, the effective batch size exceeds the leakage capacity of the targeted layer. After standard local training, the client computes an additional *masking gradient* on the defense data and superimposes it onto its real update before sharing. Crucially, Aegis does not perturb real patient samples, drop informative gradients, or alter the FL protocol; it exploits the structural limit of linear leakage so that any closed-form reconstruction the server attempts collapses into a blended mixture. We trade a moderate, one-time generation cost and an extra masking pass per local epoch for strong empirical privacy at a small utility cost, yielding a practical privacy-preserving building block for medical FL.

Contributions. (i) We identify local batch size relative to first-FC-layer capacity as a unifying bottleneck for state-of-the-art MIAs, including those that defeat secure aggregation. (ii) Building on this observation, we design Aegis, a principled, protocol-compatible defense that masks real gradients with large-batch gradients computed on locally synthesized task-relevant data. (iii) We provide a convergence analysis under standard convex assumptions that quantifies how the discrepancy between real and synthetic data enters the convergence bound; we are explicit that this is not a non-convex deep-learning guarantee. (iv) On MNIST, CIFAR-10, and three MedMNIST modalities (chest X-ray, abdominal CT, colon pathology), Aegis suppresses Robbing-the-Fed, LOKI, and Scale-MIA while matching or exceeding the model utility of state-of-the-art defenses, with running time competitive with lightweight baselines.

2 Preliminaries and Related Work

Federated learning. An FL system trains a global model G_θ across a set of distributed clients $\mathcal{C} = \{c_i\}_{i=1}^{N}$, each holding a private dataset D_i (e.g., a hospital's patient images). At each round t, a subset $\mathcal{C}^t \subseteq \mathcal{C}$ either uploads gradients (*FedSGD*), $\theta^{t+1} = \theta^t - \eta \sum_{c_i \in \mathcal{C}^t} p_i \nabla \mathcal{L}\big(\theta^t, D_i\big)$, or post-training local-parameter deltas δ_i^t (*FedAvg*), $\theta^{t+1} = \theta^t + \sum_{c_i \in \mathcal{C}^t} p_i \, \delta_i^t$, where η is the global learning rate, $\{p_i\}$ are non-negative client weights summing to one over $\mathcal{C}^t$, and $\mathcal{L}$ is the loss. While only updates leave each client, recent work shows that these updates are themselves an attack surface: adversaries can reconstruct D_i via MIAs.

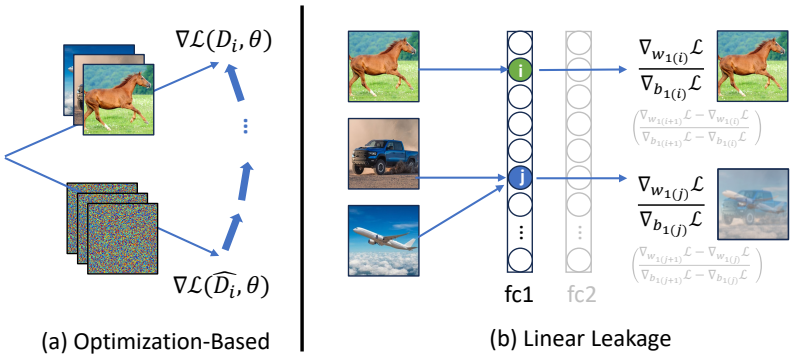


Figure 1: Two families of MIAs in FL: (a) optimization-based attacks iteratively minimize the distance between dummy and target gradients; (b) linear-leakage-based attacks recover inputs in closed form from the weight and bias gradients of a fully connected layer.

2.1 Model Inversion Attacks in Federated Learning

MIAs in FL fall into two families.

Optimization-based attacks reconstruct local data by solving $D_i^* = \arg\min_{\hat{D}_i} \|\nabla\mathcal{L}(\hat{D}_i, \theta) - \nabla\mathcal{L}(D_i, \theta)\|$ to match dummy gradients to the observed ones (Fig. 1(a)). DLG [50] pioneered this line, refined by iDLG [47], Inverting Gradients [6] for higher fidelity, and GradInversion [45] for batched inputs on ResNet. In production-scale FL settings (large local batch sizes, multiple local epochs, realistic dataset sizes, and non-IID partitions), the success of these attacks degrades sharply, and a wide range of defenses can counter them [39].

Linear-leakage-based attacks bypass iterative optimization and recover inputs in closed form, dramatically improving feasibility under realistic FL conditions and even under secure aggregation (Fig. 1(b)).

The linear-leakage primitive [27]. For a fully connected (FC) layer $y = w_1 x + b_1$ with $x \in \mathbb{R}^d$, $y \in \mathbb{R}^k$, $w_1 \in \mathbb{R}^{k\times d}$, $b_1 \in \mathbb{R}^k$, if a single input x activates neuron i in isolation, then it can be recovered exactly as $x = \nabla_{w_1(i)}\mathcal{L}/\nabla_{b_1(i)}\mathcal{L}$. When multiple inputs activate the same neuron, the corresponding gradient becomes a mixture, and the recovered image is a superposition of all activating inputs (Fig. 1(b), blue arrows).

Robbing-the-Fed [5] turns this primitive into a scalable attack by treating each neuron of a crafted FC layer as a *leakage bin*: weights w_1 encode a known cumulative distribution function over input statistics (e.g., image brightness), and biases b_1 act as thresholds, so that with negative biases and positive weights the ReLU activates exactly the bins whose threshold is crossed by a given input. Two consecutive FC layers $y = \mathrm{ReLU}(w_1 x + b_1)$ and $z = w_2 y + b_2$ enable the closed-form recovery $x_i = \big(\nabla_{w_1(i+1)}\mathcal{L} - \nabla_{w_1(i)}\mathcal{L}\big)/\big(\nabla_{b_1(i+1)}\mathcal{L} - \nabla_{b_1(i)}\mathcal{L}\big)$, which holds whenever the second FC layer is configured so that $\nabla_{w_2(i)}\mathcal{L}$ and $\nabla_{b_2(i)}\mathcal{L}$ are equal across neurons activated by the same input. This primitive has since been extended to break secure aggregation (Fishing for Data [41], LOKI [48], Scale-MIA [36]), to convolutional architectures (R-GAP [49]), and to vision transformers (APRIL [22]).

The structural bottleneck. Critically, linear-leakage attacks succeed only while the local batch size B does not exceed the number of leakage bins, i.e., the number of neurons k in the first crafted FC layer [36]. Once $B > k$, multiple inputs unavoidably activate the same bin, the gradient ratios above become mixtures, and reconstructions degrade into blurry, ghost-laden composites. *We therefore identify the ratio of effective batch size to first-FC-layer capacity as the key factor governing the success of linear-leakage MIAs, and we take this bottleneck as the target of our defense.*

2.2 Defenses against MIAs

Perturbation-based defenses. Early defenses applied gradient compression [50] or DP [24]; subsequent work [39, 40] argued that such generic mechanisms are not specifically targeted at MIAs and incur a noticeable privacy and utility trade-off. Soteria [38] perturbs FC-layer representations based

on the observation that leakage originates in latent space; GradDefense [40] extends perturbation to all layers using a sensitivity metric; Outpost [39] provides a lightweight, calibrated noise-addition scheme. All of these defenses, however, alter the very gradients used for learning and therefore degrade utility, an outcome that is especially problematic for clinical models.

Cryptographic defenses. Secure aggregation (SA) protocols [3, 1, 9, 13, 42] built on multi-party computation [16] mask local updates with random bits while preserving the aggregate, $\mathrm{SA}(\delta_1, \ldots, \delta_N) = (u_1, \ldots, u_N)$ s.t. $\sum_{i=1}^{N} u_i = \sum_{i=1}^{N} \delta_i$. SA preserves utility but adds nontrivial system and communication overhead. More importantly, recent attacks defeat it directly: gradient-disaggregation [41] infers individual updates from aggregates with crafted models, gradient-suppression [26] zeros out all but a target client's update, and large-batch-reconstruction [5] together with LOKI [48] (which extends linear leakage to convolutional modules under FedAvg) reconstruct global batches in closed form even after aggregation.

Generation-based defenses. Gong et al. [7] explored generated-data-based defenses against GAN-based model inversion attacks in the centralized setting, biasing the adversary to recover synthesized rather than original samples. Their setup, however, is centralized rather than federated and targets a different attack surface; we are not aware of prior FL-specific defenses that exploit linear-leakage's structural bottleneck via auxiliary generated data.

2.3 Generative Models for Image Synthesis

A generative model trained or queried with class- or text-conditioning provides the auxiliary in-distribution data that Aegis requires. Modern conditional image generators include GANs [8] (e.g., StackGAN [46], AttnGAN [43]), VAEs [15], and diffusion-based models [11, 33, 34, 31, 2], the latter often combined with text encoders such as CLIP [29] or T5 [30]. Aegis is not tied to a specific generative architecture; it requires that the synthesized auxiliary samples are sufficiently task-relevant to avoid harming utility while still inducing gradient collisions in the leakage bins (see Sec. 4).

3 Threat Model

We study a multi-institutional medical FL system where clients $\{c_i\}_{i=1}^{N}$, such as hospitals or research centers, jointly train a diagnostic model under a server. Each client keeps private data D_i (e.g., CT slices, chest X-rays, or histopathology tiles) locally and sends only updates δ_i. The server defines the task and protocol, coordinates training, and may act as the strongest known active MIA adversary [5, 48, 36]. Its goal is to reconstruct the private data union $\bigcup_{i=1}^{N} D_i$ from received updates.

Adversary capabilities. The server knows and controls the global model G and parameters θ at every round. It may alter the architecture or initialization to plant leakage primitives, such as crafted FC layers calibrated for linear leakage [5], and may attack at any round, especially early rounds when random parameters hide malicious structure and the model has not converged. It can also approximate the global data distribution using public data, self-collected samples, or colluding clients, yielding an auxiliary set D_{Adv} for bin calibration. The adversary knows standard FL primitives, including FedSGD, FedAvg, and secure aggregation, and mounts linear leakage on top of them as in current state-of-the-art MIAs [5, 48, 36].

Defender model. The defenders are the legitimate medical clients. They aim to protect patient data while sending useful updates, so diagnostic accuracy should stay close to the no-defense baseline. Each client sees only the broadcast model and cannot reliably tell whether it is benign or crafted as G_{adv}, nor which MIA variant is being used. Existing defenses such as SA, DP, and gradient pruning either remain vulnerable to scalable linear leakage or incur utility losses that are problematic for clinical use (Sec. 5.3).

Why local hyperparameters alone do not suffice. One might try to stop MIAs by changing local training. However, with Robbing-the-Fed [5] as a representative attack, varying batch size, learning rate, and local epochs within utility-preserving ranges does not meaningfully reduce reconstruction quality (Tab. A.2; Appendix C). This motivates an explicit *defense*, not reliance on training hyperparameters.

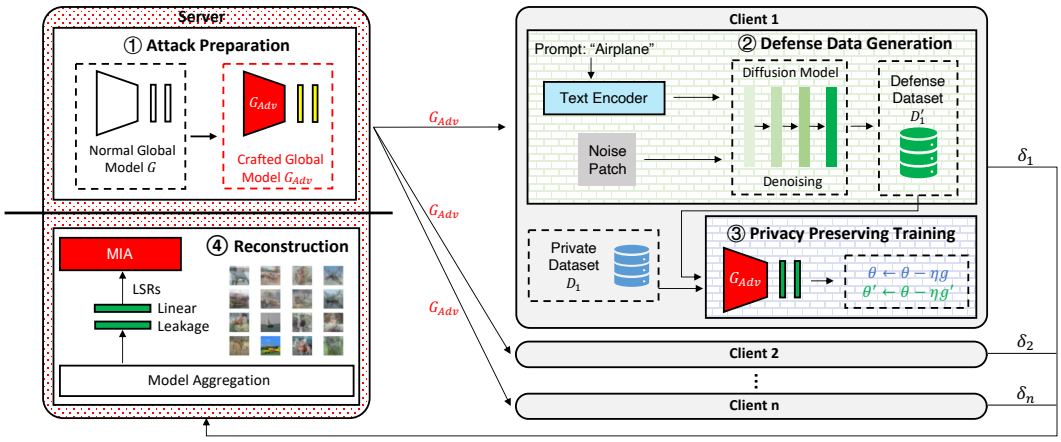


Figure 2: Overview of Aegis. Each client uses a local generative model to synthesize an in-distribution defense dataset, computes a large-batch *masking gradient* from it, and superimposes that masking gradient on its real model update once per local epoch. The combined update drives the effective batch size beyond the leakage capacity of the first crafted FC layer; multiple inputs collide in every leakage bin and the server's reconstruction collapses into a blended mixture rather than identifiable patient samples.

4 System Design

Threat-model recap (Sec. 3). An active server may craft the broadcast model G_{adv} to enable closed-form linear-leakage reconstruction of client images; the client controls only its local training and what it transmits to the server, and cannot reliably distinguish a benign global model from a maliciously crafted one. Aegis is a client-side mechanism that operates entirely within this constraint.

4.1 Defense Principle and Intuition

Section 2 established that every known MIA is bounded by the local batch size: optimization-based attacks fail at moderate batches, and linear-leakage attacks succeed only while the local batch size B does not exceed the number of bins k in the first crafted FC layer. We turn this universal bottleneck into a defense by deliberately driving the *effective* batch size that contributes to the shared gradient above k, without modifying the real training data, the FL protocol, or the loss objective. Because both private and synthetic samples activate the same leakage bins, every reconstruction the server attempts is forced to collapse into a superposition of multiple inputs, and individual patient samples cannot be recovered.

In medical FL, real local datasets are typically small per institution and non-IID across institutions, so simply enlarging the real training batch is rarely possible. Aegis instead constructs an auxiliary defense dataset on the client side using a generative model, computes a separate masking gradient on it, and adds this masking gradient to the legitimate update before sharing. Because the masking gradient is computed from in-distribution synthetic data and added to the model's parameters, the masking step is itself a valid optimization step on a slightly perturbed loss landscape. The result is a structural defense that does not inject random noise (unlike DP or Outpost), does not discard informative gradient components (unlike GC or Soteria), and does not require any cryptographic protocol (unlike SA).

4.2 Detailed Design

We describe Aegis from the perspective of one client c_i in round t (Fig. 2); all participating clients run the same procedure in parallel. Algorithm 1 gives the complete client-side procedure.

Step ①: malicious global model. The server turns the benign global model G into G_{adv} by inserting or re-parameterizing crafted FC layers with k leakage bins, then broadcasts it to clients. From a client's view, G_{adv} is indistinguishable from a benign model and must be used as received.

Step ②: defense dataset construction. The client holds a local class- or text-conditional generator $\mathcal{G}$, pretrained on public data and optionally fine-tuned on a small in-domain corpus. Using local labels Φ_i (e.g., disease or organ labels), it samples $D'_i = \{\, \tilde{x}_j \sim \mathcal{G}(\cdot \mid \phi) : \phi \in \Phi_i,\ j = 1, \ldots, M_i \,\}$. The defense size M_i is chosen so that $B + M_i > k$; in our experiments, $M_i = 2048 \gg B = 64$. The generated samples must be task-relevant: out-of-distribution samples may miss the same leakage bins as D_i or bias training. We therefore filter samples by the client-side in-distribution budget $\mathbb{E}\|D_i - D'_i\|^2 \leq H$, computed over matched private and synthetic samples. Samples that violate the budget are discarded, and $\mathcal{G}$ can be fine-tuned on D_i if needed. The same budget appears in Assumption 6. This construction is offline, cached across rounds, and never transmitted.

Step ③: privacy-preserving local training. Each round has E local epochs. In epoch e, the client first trains normally on mini-batches from D_i with gradient $g_i^{(e)} = \nabla_\theta \mathcal{L}\big(G_{\text{adv}}(D_i; \theta_i^{(e)})\big)$. Once per epoch, not once per round, it then computes a masking gradient on D'_i, $g_i'^{(e)} = \nabla_\theta \mathcal{L}\big(G_{\text{adv}}(D'_i; \theta_i^{(e)})\big)$, using one large logical batch or gradient accumulation over micro-batches when memory is limited. The client applies the combined step $\theta_i^{(e+1)} \leftarrow \theta_i^{(e)} - \eta\, g_i^{(e)} - \eta\, g_i'^{(e)}$. Thus each epoch contributes both learning on D_i and masking on D'_i. After E epochs, the client sends $\delta_i^t = \theta_i^{(E)} - \theta^t$. The effective per-epoch direction is $\tilde{g}_i^{(e)} = (1 - \lambda)\, g_i^{(e)} + \lambda\, g_i'^{(e)}$ with $\lambda = M_i/(B + M_i)$, as used in Appendix G.

Step ④: server-side reconstruction collapses. The server attacks the received δ_i (or the post-aggregation analogue under SA) as in Robbing-the-Fed [5], LOKI [48], or Scale-MIA [36]. Because masking adds M_i synthetic inputs to the same leakage bins, each bin mixes multiple $\big(\nabla_{w_1}\mathcal{L}, \nabla_{b_1}\mathcal{L}\big)$ contributions. Closed-form recovery therefore returns blended images rather than identifiable patient samples.

Why this is safe and principled. Aegis does not perturb D_i, does not drop or down-weight any component of $g_i^{(e)}$, and does not modify the FL protocol; the server still receives a valid update that contains the same learning signal as standard local training plus an additional gradient step on task-relevant synthetic data. The privacy argument follows from the structural collision above, and empirical utility on deep, non-convex models is reported in Sec. 5.

Algorithm 1 summarizes the complete client-side procedure; the full convergence analysis is provided in Appendix G, while Appendix A consolidates notation and implementation details.

5 Experiments

We evaluate Aegis as a privacy-preserving primitive for medical FL, focusing on two questions: (i) does Aegis suppress state-of-the-art linear-leakage MIAs on both standard benchmarks and clinical imaging modalities, and (ii) does it preserve global model utility competitively with state-of-the-art defenses, without imposing prohibitive overhead?

5.1 Experimental Setup

Attack Baselines: To evaluate the effectiveness of our mechanism, we select three representative closed-form attacks: Robbing-the-Fed [5], LOKI [48], Scale-MIA [36]. These are evaluated in a production FL setting where optimization-based methods fail beyond their feasibility.

Defense Baselines: We compare the effectiveness of Aegis with various state-of-the-art defense schemes. Among them, DP [24] adds artificial noise to the local training gradients. Outpost [39] and GradDefense (GD) [40] adaptively add noise to gradients based on privacy leakage risk. Soteria [38] and Gradient Compression (GC) [50] prune parts of the gradients to hide the recoverable ground truth.

FL configuration. We use non-IID FedAvg with $N = 100$ clients, disjoint local data, 10% client participation per round, $E = 3$ local epochs, and batch size $B = 64$.

Datasets. We use MNIST and CIFAR-10 for standard utility/privacy comparisons, and ChestMNIST, OrganAMNIST, and PathMNIST from MedMNIST v2 [44] for medical FL evaluation.

Algorithm 1: Aegis client-side procedure (one communication round)

Input: global model G_{adv} with parameters θ^t; private data D_i; conditioning set Φ_i; generator $\mathcal{G}$; defense size target M_i; in-distribution budget H; local hyperparameters E, B, η
Output: client update δ_i^t to send to the server

; // *Stage 1: Defense dataset construction (offline; cached across rounds)*
if D_i' *has not been constructed yet, or the modality has changed* **then**
  Initialize $D_i' \leftarrow \emptyset$;
  while $|D_i'| < M_i$ **do**
    Sample a label/prompt $\phi \sim \text{Uniform}(\Phi_i)$;
    Draw a candidate sample $\tilde{x} \sim \mathcal{G}(\cdot \mid \phi)$ **if** $\tilde{x}$ *is in-distribution:* $\mathbb{E}\|D_i - (D_i' \cup \{\tilde{x}\})\|^2 \leq H$ **then**
      $D_i' \leftarrow D_i' \cup \{\tilde{x}\}$;
    else
      Discard $\tilde{x}$; if rejection rate is high, fine-tune $\mathcal{G}$ on a sanitized subset of D_i and resume sampling;
  Cache D_i' locally; reuse in subsequent rounds;

; // *Stage 2: Local training interleaved with per-epoch masking*
$\theta_i \leftarrow \theta^t$; // initialize local copy from received global model
for *epoch* $e = 1, \ldots, E$ **do**
  Partition D_i into mini-batches of size B;
  for *each mini-batch* $B_j \subseteq D_i$ **do**
    $g_{i,j} \leftarrow \nabla_\theta \mathcal{L}\big(G_{\text{adv}}(B_j; \theta_i)\big)$;
    $\theta_i \leftarrow \theta_i - \eta\, g_{i,j}$; // standard SGD step
  ; // *Per-epoch masking: once per local epoch, not once per round*
  Form a large batch from D_i' of size M_i (or accumulate gradients over micro-batches when memory-limited);
  $g_i'^{(e)} \leftarrow \nabla_\theta \mathcal{L}\big(G_{\text{adv}}(D_i'; \theta_i)\big)$;
  $\theta_i \leftarrow \theta_i - \eta\, g_i'^{(e)}$; // drives effective batch size $> k$

; // *Stage 3: Form and transmit the masked update*
$\delta_i^t \leftarrow \theta_i - \theta^t$; // accumulated parameter delta after all E epochs of interleaved learning and masking
Send δ_i^t to the server (through SA or directly, per the FL protocol);

Defense data. Each client builds D_i' with a class- or text-conditional latent diffusion model [33]; generated samples are filtered by $\mathbb{E}\|D_i - D_i'\|^2 \leq H$ and used only for masking gradients. Unless swept, Aegis uses defense batch size $M_i = 2048$.

Evaluation Metrics: We assess reconstruction performance using peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM). PSNR quantifies the ratio of an image signal's maximum possible power to noise, measured in decibels (dB). SSIM evaluates perceived image quality by comparing luminance, contrast, and structure, with values ranging from -1 to 1, where scores near 1 indicate high similarity and those below 0.5 suggest significant differences. The reconstruction rate is defined as the fraction of reconstructed images with PSNR above the threshold $th = 18$. These metrics are widely adopted in prior work [36, 5, 41, 48]. To assess Aegis 's impact on FL performance, we track test accuracy across FL rounds and compare it against other defense baselines.

Details. Appendix B gives model architectures, optimization, data partitioning, generation prompts/filtering, baseline hyperparameters, and attack protocol; Appendix I states trust assumptions and limitations.

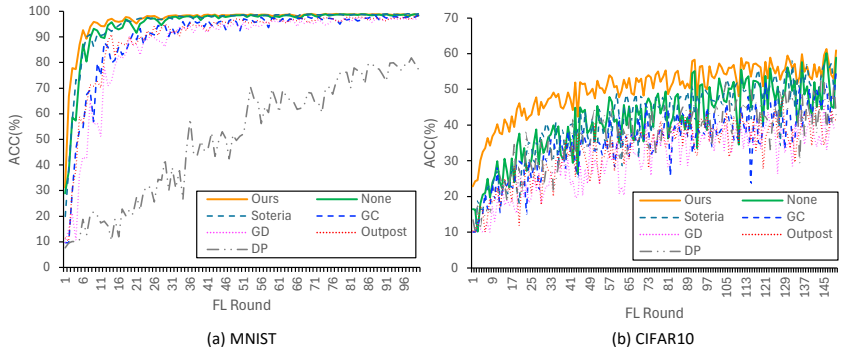


Figure 3: Round-to-accuracy comparison of Aegis (defense batch size 2048) against state-of-the-art defenses on (a) MNIST and (b) CIFAR-10.

Table 1: Wall-clock training time (seconds) for 100 FL rounds on MNIST and 150 FL rounds on CIFAR-10.

Dataset	None	Aegis	Soteria	GC	GD	Outpost	DP
MNIST	290.5	1890.1	19816.2	334.5	4622.4	512.4	330.9
CIFAR-10	433.1	2339.3	32482.0	512.1	8356.8	781.7	517.0

5.2 Round-to-Accuracy under Defense

We first evaluate how Aegis affects the global model's convergence trajectory. We set the Aegis defense batch size to 2048; for baselines, the Soteria pruning rate is 80%, the GC pruning rate is 80%, DP uses Gaussian noise with $\varepsilon = 1.0$ and $\Delta = 10^{-4}$, and GD/Outpost use the configurations recommended in their original papers [40, 39].

On MNIST (Fig. 3(a)), DP is the most damaging baseline: its calibrated noise visibly slows convergence and lowers final accuracy. Adaptive-noise schemes (GD, Outpost) attenuate this effect by injecting smaller, risk-calibrated perturbations. Pruning-based defenses (GC, Soteria) only remove small-magnitude gradients and therefore incur smaller utility hits. Aegis tracks the no-defense baseline most closely and achieves the highest final test accuracy among all defenses. CIFAR-10 (Fig. 3(b)) shows the same ordering: Aegis's masking gradients are valid optimization steps on in-distribution data, and therefore do not destabilize convergence the way noise- or pruning-based defenses do. The convergence bound in Appendix G is stated under standard convex assumptions and explains how the auxiliary-data discrepancy enters constants; it is not a non-convex guarantee for the deep networks here, so Fig. 3 is the primary utility evidence in the deep regime.

Tab. 1 reports wall-clock training times. Noise- and pruning-based defenses (DP, Outpost, GC) introduce minimal overhead. GD and Soteria, by contrast, require additional per-epoch computations: Soteria learns perturbed representations every epoch, and GD computes a model-sensitivity metric to calibrate Gaussian noise. Aegis adds only the cost of one large-batch forward-backward pass over D_i' per local epoch and remains substantially faster than GD and Soteria, while offering substantially stronger privacy protection (Sec. 5.3). When GPU memory is smaller than the logical masking batch M_i, the client can accumulate gradients over micro-batches of size b: the mean gradient matches a single full-batch pass, so bin collisions and the privacy argument are unchanged, while wall-clock grows roughly with M_i/b. This overhead is modest on our workstation but can matter for resource-constrained clients (Appendix I).

5.3 Effectiveness against Linear-Leakage MIAs

We now evaluate Aegis's defensive effectiveness against the three linear-leakage MIAs. Experiments use the same FL setup ($E = 3$, $B = 64$) for direct comparability. Fig. 4 shows that, across all three attacks, Aegis systematically suppresses reconstruction quality: as the defense batch size grows, RR, PSNR, and SSIM all decrease together, and reconstructed samples become progressively indistinguishable from noise. This empirically confirms our central claim that enlarging the effective

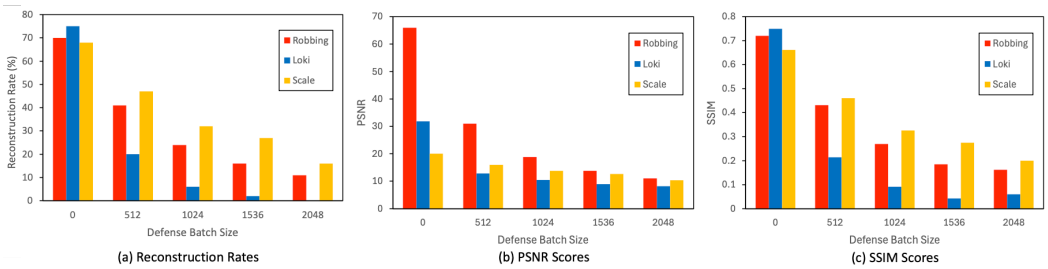
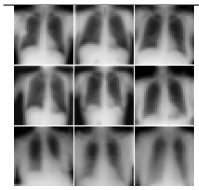
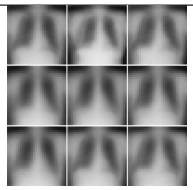
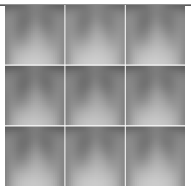
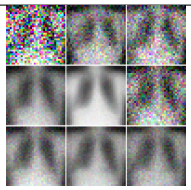
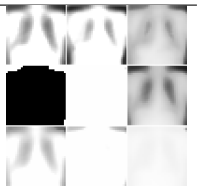
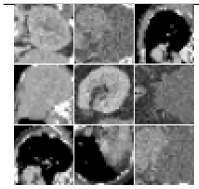
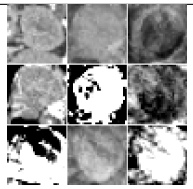
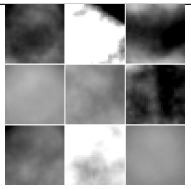
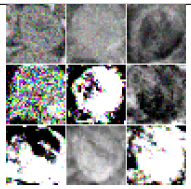
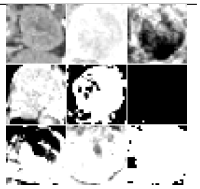
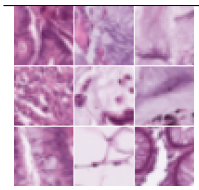
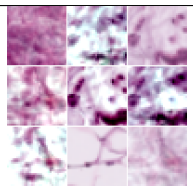
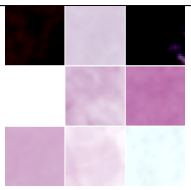
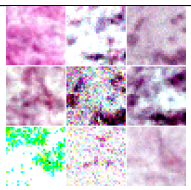
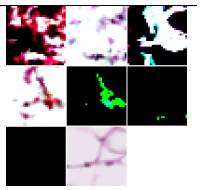


Figure 4: Effectiveness of Aegis against three state-of-the-art linear-leakage MIAs (Robbing-the-Fed, LOKI, Scale-MIA) on CIFAR-10. As the defense batch size grows, RR, PSNR, and SSIM all decrease consistently.

Table 2: Robbing-the-Fed reconstructions on three MedMNIST modalities (ChestMNIST, OrganAMNIST, PathMNIST) under no defense, Aegis, DP, and GC. Each block shows qualitative samples (top) and quantitative reconstruction metrics (bottom). Blue scores in parentheses are computed only over successfully reconstructed samples (PSNR>18); RR is the overall reconstruction rate (fraction of samples with PSNR>18).

Ground Truth	**No defense**	**Aegis**	**DP**	**GC**
ChestMNIST (chest X-ray)				
				
RR (%)	89.06	9.50	71.88	59.38
PSNR	62.40 (105.40)	18.70 (29.80)	36.80 (58.60)	42.50 (76.20)
SSIM	0.94 (0.99)	0.46 (0.91)	0.82 (0.96)	0.78 (0.97)
OrganAMNIST (abdominal CT)				
				
RR (%)	82.81	11.63	65.63	53.13
PSNR	50.60 (98.70)	20.10 (31.40)	33.40 (54.20)	38.90 (70.50)
SSIM	0.89 (0.98)	0.51 (0.88)	0.76 (0.94)	0.72 (0.95)
PathMNIST (colon pathology)				
				
RR (%)	78.13	9.38	59.38	46.88
PSNR	45.20 (91.30)	17.30 (27.90)	30.60 (48.70)	34.20 (61.80)
SSIM	0.86 (0.97)	0.38 (0.84)	0.69 (0.92)	0.64 (0.93)

batch size beyond the linear-leakage capacity, via a synthetic-data masking gradient, is sufficient to disrupt scalable, closed-form reconstructions. Quantitative summaries and additional reconstructed samples comparing Aegis against DP, GC, Soteria, and Outpost on CIFAR-10 are provided in Ap-

pendix D. The experiments instantiate the standard linear-leakage construction; if a server increases the first-FC width k, Aegis correspondingly requires a larger M_i, so sizing the defense batch under unknown or adaptive k is a deployment consideration rather than a protocol change. A server might also attempt to demix bins using public knowledge of the generator (typically ill-posed when real and synthetic contributions mix), or pursue primitives outside linear leakage (not among today's scalable MIAs). Appendix I expands these adaptive-adversary considerations.

Tab. 2 reports Robbing-the-Fed reconstructions on the three MedMNIST modalities with both qualitative samples and quantitative RR/PSNR/SSIM (computed as in Tab. A.3), providing the most safety-relevant evidence: anatomical and pathological detail in real chest X-rays, abdominal CTs, and colon pathology tiles. Without any defense, RR reaches 78–89% with high PSNR/SSIM; DP lowers RR but leaves strong conditional scores on surviving reconstructions; GC yields intermediate RR with similarly high PSNR on recovered tiles when pruning misses informative gradients. Aegis drives RR down to 9.38–11.63% across modalities while collapsing aggregate PSNR and SSIM relative to no defense and both baselines, matching the blended visual outputs: gradient superposition forces multiple medical samples into the same leakage bins and suppresses identifiable anatomical or pathological detail. Crucially, this protection is achieved without perturbing or discarding any informative gradient component; the model still learns from the full, untouched real medical data, so model utility (Tab. A.4) is preserved while privacy is restored. These medical studies use three MedMNIST modalities at benchmark resolution; extending to full-resolution CT volumes or whole-slide pathology would require correspondingly stronger generators and is outside the experiments reported here. The quantitative claims target FC-layer linear leakage as in Robbing-the-Fed-class attacks; defenses against hypothetical primitives that bypass this bottleneck are outside our scope (Appendix I).

6 Conclusion

We presented Aegis, a principled, protocol-compatible defense against linear-leakage model inversion attacks in medical federated learning. By identifying that all known scalable MIAs are bounded by the local batch size relative to the first-FC-layer capacity, Aegis turns this structural bottleneck into a defense: each client computes a large-batch masking gradient on locally synthesized task-relevant data and superimposes it onto its real update once per local epoch, forcing the server's closed-form reconstructions to collapse into indistinguishable mixtures. Aegis is not tied to a specific generative architecture; it requires only that the auxiliary samples are sufficiently task-relevant to avoid harming utility while inducing gradient collisions. We complement the design with stability analysis under standard convex assumptions (not a non-convex guarantee) and show, on MNIST, CIFAR-10, and three MedMNIST modalities (chest X-ray, abdominal CT, colon pathology), that Aegis suppresses three state-of-the-art MIAs while preserving model utility and incurring competitive overhead, providing a practical privacy-preserving primitive for AI/ML for health and biotechnology.

Acknowledgements

This work was supported in part by the Office of Naval Research under grants N00014-24-1-2730, and the National Science Foundation under grants 2312447, 2247560, and 2235232.

References

[1] James Henry Bell, Kallista A. Bonawitz, Adrià Gascón, Tancrède Lepoint, and Mariana Raykova. Secure single-server aggregation with (poly)logarithmic overhead. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 1253–1269, 2020. doi: 10.1145/3372297.3417885.

[2] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, Wesam Manassra, Prafulla Dhariwal, Casey Chu, Yunxin Jiao, and Aditya Ramesh. Improving image generation with better captions. Technical report, OpenAI, 2023. URL `https://cdn.openai.com/papers/dall-e-3.pdf`.

[3] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191, 2017. doi: 10.1145/3133956.3133982.

[4] Shaltiel Eloul, Fran Silavong, Sanket Kamthe, Antonios Georgiadis, and Sean J. Moran. Mixing gradients in neural networks as a strategy to enhance privacy in federated learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3956–3965, 2024. URL `https://openaccess.thecvf.com/content/WACV2024/html/Eloul_Mixing_Gradients_in_Neural_Networks_as_a_Strategy_To_Enhance_WACV_2024_paper.html`.

[5] Liam Fowl, Jonas Geiping, Wojciech Czaja, Micah Goldblum, and Tom Goldstein. Robbing the Fed: Directly obtaining private data in federated learning with modified models. In *International Conference on Learning Representations*, 2022. URL `https://iclr.cc/virtual/2022/poster/7067`.

[6] 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. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/hash/c4ede56bbd98819ae6112b20ac6bf145-Abstract.html`.

[7] Xueluan Gong, Ziyao Wang, Shuaike Li, Yanjiao Chen, and Qian Wang. A GAN-based defense framework against model inversion attacks. *IEEE Transactions on Information Forensics and Security*, 18: 4475–4487, 2023. doi: 10.1109/tifs.2023.3295944.

[8] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL `https://proceedings.neurips.cc/paper_files/paper/2014/hash/f033ed80deb0234979a61f95710dbe25-Abstract.html`.

[9] Xiaojie Guo, Zheli Liu, Jin Li, Jiqiang Gao, Boyu Hou, Changyu Dong, and Thar Baker. VeriFL: Communication-efficient and fast verifiable aggregation for federated learning. *IEEE Transactions on Information Forensics and Security*, 16:1736–1751, 2021. doi: 10.1109/tifs.2020.3043139.

[10] 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. URL `https://openaccess.thecvf.com/content/CVPR2022/html/Hatamizadeh_GradViT_Gradient_Inversion_of_Vision_Transformers_CVPR_2022_paper.html`.

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html`.

[12] Heng Jin, Chaoyu Zhang, Hexuan Yu, Wenjing Lou, and Y. Thomas Hou. Privacy-preserving split learning for federated LLM fine-tuning. *arXiv preprint arXiv:2609.09794*, 2026. URL `https://arxiv.org/abs/2609.09794`.

[13] Swanand Kadhe, Nived Rajaraman, O. Ozan Koyluoglu, and Kannan Ramchandran. FastSecAgg: Scalable secure aggregation for privacy-preserving federated learning. *arXiv preprint arXiv:2009.11248*, 2020. URL `https://arxiv.org/abs/2009.11248`.

[14] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Advances in Neural Information Processing Systems*, volume 33, pages 12104–12114. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/hash/8d30aa96e72440759f74bd2306c1fa3d-Abstract.html`.

[15] Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014. URL https://arxiv.org/abs/1312.6114.

[16] Brian Knott, Shobha Venkataraman, Awni Hannun, Shubho Sengupta, Mark Ibrahim, and Laurens van der Maaten. CrypTen: Secure multi-party computation meets machine learning. In *Advances in Neural Information Processing Systems*, volume 34, pages 4961–4973. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/2754518221cfbc8d25c13a06a4cb8421-Abstract.html.

[17] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

[18] Yann LeCun, Corinna Cortes, and Christopher J. C. Burges. The MNIST database of handwritten digits. Dataset webpage, n.d. URL https://web.archive.org/web/20200430193701/http://yann.lecun.com/exdb/mnist/. Archived April 30, 2020.

[19] He Li, Lu Yu, and Wu He. The impact of GDPR on global technology development. *Journal of Global Information Technology Management*, 22(1):1–6, 2019. doi: 10.1080/1097198x.2019.1569186.

[20] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on Non-IID data silos: An experimental study. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 965–978, 2022. doi: 10.1109/icde53745.2022.00077.

[21] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of FedAvg on Non-IID data. In *International Conference on Learning Representations*, 2020. URL https://iclr.cc/virtual/2020/poster/1421.

[22] Jiahao Lu, Xi Sheryl Zhang, Tianli Zhao, Xiangyu He, and Jian Cheng. APRIL: Finding the Achilles' heel on privacy for vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10051–10060, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Lu_APRIL_Finding_the_Achilles_Heel_on_Privacy_for_Vision_Transformers_CVPR_2022_paper.html.

[23] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR, 2017. URL https://proceedings.mlr.press/v54/mcmahan17a.html.

[24] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *International Conference on Learning Representations*, 2018. URL https://iclr.cc/virtual/2018/poster/187.

[25] Wilnellys Moore and Sarah Frye. Review of HIPAA, part 1: History, protected health information, and privacy and security rules. *Journal of Nuclear Medicine Technology*, 47(4):269–272, 2019. doi: 10.2967/jnmt.119.227819.

[26] Dario Pasquini, Danilo Francati, and Giuseppe Ateniese. Eluding secure aggregation in federated learning via model inconsistency. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 2429–2443, 2022. doi: 10.1145/3548606.3560557.

[27] Le Trieu Phong, Yoshinori Aono, Takuya Hayashi, Lihua Wang, and Shiho Moriai. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Transactions on Information Forensics and Security*, 13(5):1333–1345, 2018. doi: 10.1109/tifs.2017.2787987.

[28] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations*, 2024. URL https://iclr.cc/virtual/2024/poster/18250.

[29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

[30] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

[31] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *arXiv preprint arXiv:2204.06125*, 2022. URL https://arxiv.org/abs/2204.06125.

[32] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/5f8e2fa1718d1bbcadf1cd9c7a54fb8c-Abstract.html.

[33] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Rombach_High-Resolution_Image_Synthesis_With_Latent_Diffusion_Models_CVPR_2022_paper.html.

[34] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, volume 35, pages 36479–36494. Curran Associates, Inc., 2022. doi: 10.52202/068431-2643. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/ec795aeadae0b7d230fa35cbaf04c041-Abstract-Conference.html.

[35] Shanghao Shi, Md Shahedul Haque, Abhijeet Parida, Chaoyu Zhang, Marius George Linguraru, Y. Thomas Hou, Syed Muhammad Anwar, and Wenjing Lou. MedLeak: Multimodal medical data leakage in secure federated learning with crafted models. In *Proceedings of the ACM/IEEE International Conference on Connected Health: Applications, Systems and Engineering Technologies*, pages 245–256, 2025. doi: 10.1145/3721201.3721375.

[36] Shanghao Shi, Ning Wang, Yang Xiao, Chaoyu Zhang, Yi Shi, Y. Thomas Hou, and Wenjing Lou. ScaleMIA: A scalable model inversion attack against secure federated learning via latent space reconstruction. In *Network and Distributed System Security (NDSS) Symposium*, 2025. doi: 10.14722/ndss.2025.240644.

[37] Shanghao Shi, Chaoyu Zhang, Heng Jin, Yang Xiao, Yevgeniy Vorobeychik, William Yeoh, Ning Zhang, Y. Thomas Hou, and Wenjing Lou. From efficiency to leakage – privacy backdoor in federated language model fine-tuning. *arXiv preprint arXiv:2606.20553*, 2026. URL https://arxiv.org/abs/2606.20553.

[38] 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. URL https://openaccess.thecvf.com/content/CVPR2021/html/Sun_Soteria_Provable_Defense_Against_Privacy_Leakage_in_Federated_Learning_From_CVPR_2021_paper.html.

[39] 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, 2023. doi: 10.1109/infocom53939.2023.10228919.

[40] Junxiao Wang, Song Guo, Xin Xie, and Heng Qi. Protect privacy from gradient leakage attack in federated learning. In *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*, pages 580–589, 2022. doi: 10.1109/infocom48880.2022.9796841.

[41] Yuxin Wen, Jonas A. Geiping, Liam Fowl, Micah Goldblum, and Tom Goldstein. Fishing for user data in large-batch federated learning via gradient magnification. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 23668–23684. PMLR, 2022. URL https://proceedings.mlr.press/v162/wen22a.html.

[42] Guowen Xu, Hongwei Li, Sen Liu, Kan Yang, and Xiaodong Lin. VerifyNet: Secure and verifiable federated learning. *IEEE Transactions on Information Forensics and Security*, 15:911–926, 2020. doi: 10.1109/tifs.2019.2929409.

[43] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1316–1324, 2018. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Xu_AttnGAN_Fine-Grained_Text_CVPR_2018_paper.html.

[44] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. MedMNIST v2 - a large-scale lightweight benchmark for 2D and 3D biomedical image classification. *Scientific Data*, 10(1):41, 2023. doi: 10.1038/s41597-022-01721-8.

[45] 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. URL https://openaccess.thecvf.com/content/CVPR2021/html/Yin_See_Through_Gradients_Image_Batch_Recovery_via_GradInversion_CVPR_2021_paper.html.

[46] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N. Metaxas. StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5907–5915, 2017. URL https://openaccess.thecvf.com/content_iccv_2017/html/Zhang_StackGAN_Text_to_ICCV_2017_paper.html.

[47] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. iDLG: Improved deep leakage from gradients. *arXiv preprint arXiv:2001.02610*, 2020. URL https://arxiv.org/abs/2001.02610.

[48] Joshua C. Zhao, Atul Sharma, Ahmed Roushdy Elkordy, Yahya H. Ezzeldin, Salman Avestimehr, and Saurabh Bagchi. Loki: Large-scale data reconstruction attack against federated learning through model manipulation. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 1287–1305, 2024. doi: 10.1109/sp54263.2024.00030.

[49] Junyi Zhu and Matthew Blaschko. R-GAP: Recursive gradient attack on privacy. In *International Conference on Learning Representations*, 2021. URL https://iclr.cc/virtual/2021/poster/3194.

[50] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/60a6c4002cc7b29142def8871531281a-Abstract.html.

Appendix

The appendix is organized as follows. Appendix A gives a consolidated notation table and implementation details for Algorithm 1. Appendix B records the experimental setup needed for reproducibility, including data partitioning, models, optimization, defense generation, baseline settings, and attack protocol. Appendix C reports the local-hyperparameter sensitivity study referenced from Sec. 3. Appendix D provides additional CIFAR-10 reconstruction comparisons against five defense baselines. Appendix G gives the standard convex assumptions and convergence theorem used in [21], adapted here. Appendix H summarizes representative gradient-based MIAs in FL. Appendix I collects extended discussion of deployment assumptions and scope.

Additional medical utility and privacy–utility results appear in Appendix E; generator, non-IID, and backbone experiments appear in Appendix F.

A Aegis Notation and Implementation Details

This appendix supplements the client-side procedure in Algorithm 1. The same procedure runs in parallel on every participating client, and the server-side aggregation step is unchanged from the underlying FL protocol (FedAvg or FedSGD with or without secure aggregation).

A.1 Notation and Inputs

Table A.1 consolidates the symbols used in the client-side algorithm and the convergence analysis.

Table A.1: Notation used by Aegis. The left column lists the mathematical symbol and the right column gives its role in the FL protocol, defense mechanism, or convergence analysis.

Notation	Explanation
c_i, N	Client i and the total number of clients.
t, E, B	Communication round, number of local epochs, and local mini-batch size.
K, S_t	Number and set of clients selected in round t.
G, G_{adv}	Benign global model and potentially adversarial broadcast model.
θ^t, θ_i	Global parameters at round t and the local copy maintained by client c_i.
δ_i^t	Client update sent to the server after local training and masking.
D_i, D_i'	Private client dataset and synthetic defense dataset.
Φ_i, $\mathcal{G}$	Local label/prompt set and class- or text-conditional generator.
M_i, k	Defense dataset size and leakage-bin capacity of the first crafted FC layer.
H	In-distribution budget, $\mathbb{E}\|D_i - D_i'\|^2 \le H$.
η, $g_{i,j}$	Local learning rate and mini-batch gradient on private data.
$g_i^{(e)}$, $g_i'^{(e)}$	Epoch-level private gradient and masking gradient.
λ	Mixing weight $M_i/(B+M_i)$ used in the effective-gradient view.
F_i, F, p_i	Local objective, global objective, and client aggregation weight.
L, μ, σ_i^2, G^2	Smoothness, strong-convexity, gradient-variance, and gradient-norm constants.
κ, γ, Γ	Condition number, learning-rate schedule constant, and client heterogeneity term.

A.2 Implementation Notes

Cost amortization. Stage 1 is performed once and cached, so Aegis adds one masking pass over D_i' (size M_i) per local epoch, i.e., E extra forward-backward passes per communication round. This is empirically much cheaper than the per-epoch costs incurred by GD and Soteria (Tab. 1).

Memory considerations and gradient accumulation. The masking step requires forming a single backward pass over a logical batch of size $M_i = 2048$, which is well above the typical mini-batch size used during training. On clients whose accelerators cannot hold M_i samples in memory, the masking step is implemented with gradient accumulation: D_i' is processed as M_i/b micro-batches of size b, each contributes $\nabla_\theta \mathcal{L}\big(G_{\text{adv}}(B_j'; \theta_i)\big)$, and the accumulated sum is applied as a single masking step before the next epoch begins. Because the mean-loss gradient is linear in the per-sample gradients, gradient accumulation yields the same $g_i'^{(e)}$ as a single large-batch pass,

and the linear-leakage collision argument is unchanged: each accumulated micro-batch deposits its $M_i/(M_i/b) = b$ contributions into the same leakage bins as the synchronous formulation. In practice this only changes wall-clock time, not the privacy and utility behavior of the defense.

Generator requirements (not architecture-agnostic). The procedure does not depend on a specific generative-model family, but it does depend on the auxiliary samples being sufficiently task-relevant: samples that are far out-of-distribution either fail to enter the same leakage bins as D_i (so they do not induce collisions) or pull the model away from the real data distribution (hurting utility). In our experiments we instantiate $\mathcal{G}$ with a publicly available latent diffusion model conditioned on locally known classes; conditional GANs, conditional VAEs, or domain-specific clinical generators are also viable as long as the in-distribution budget can be satisfied.

In-distribution check. The constraint $\mathbb{E}\|D_i - D_i'\|^2 \leq H$ is implemented per-client using a small validation split of D_i and a simple distance estimator (e.g., feature-space MSE in a frozen feature extractor); samples whose addition would exceed H are rejected. In practice this filter triggers infrequently when $\mathcal{G}$ is reasonably well matched to the modality.

Compatibility with secure aggregation. Aegis operates entirely client-side and does not modify the FL communication protocol. When SA is enabled, δ_i^t is fed through the SA mask exactly as a standard update, and all guarantees of the underlying SA protocol continue to hold; the privacy benefit from Aegis is additive.

B Additional Experimental Details

This section records the implementation choices abbreviated in Sec. 5.1. The main text keeps only the experimental essentials; the details below specify the exact simulation, models, generation pipeline, baselines, and metrics.

Evaluation scope and compute. We evaluate the active-server linear-leakage threat model of [5, 48, 36]. Unless noted otherwise, quantitative results average multiple independent runs with different random seeds, including randomness from initialization and client participation. All timing and training experiments use a workstation with an Intel Core i9-11900K CPU and one NVIDIA GeForce RTX 3080 GPU; peak memory is governed by the logical masking batch M_i when gradient accumulation is used.

Federated data and optimization. We simulate $N = 100$ clients with non-overlapping data partitions. To induce non-IID label skew, each client is restricted to examples from at most five classes, and the server samples 10% of clients uniformly without replacement per FedAvg round. Local training uses Adam with learning rate $\eta = 10^{-3}$, local epochs $E = 3$, and mini-batch size $B = 64$; gradient clipping is used only where required by a baseline implementation, such as DP calibration.

Datasets and classifiers. MNIST [18] and CIFAR-10 [17] use a shared CNN with three 3×3 convolutional layers (32 filters, ReLU, 2×2 max-pooling), a 512-unit fully connected layer, and a 10-class output layer. For ChestMNIST, OrganAMNIST, and PathMNIST [44], we use the same backbone, adapt the input channel count to the dataset, and set the output width to the dataset-native label cardinality.

Defense data and hyperparameters. Defense images are generated with Stable Diffusion 2.0 [33] using OpenCLIP ViT-H/14. Clients use class-aligned prompts derived from locally visible labels, resize or crop generated candidates to match the dataset tensor layout, and filter them by the in-distribution budget $\mathbb{E}\|D_i - D_i'\|^2 \leq H$ estimated on a small held-out slice of D_i. Unless swept, Aegis uses $M_i = 2048$; Soteria and GC use pruning ratio 0.8; DP uses Gaussian noise with $\varepsilon = 1.0$ and sensitivity $\Delta = 10^{-4}$; GD and Outpost follow the hyperparameters released with their reference implementations [40, 39]. One accepted defense candidate takes roughly five seconds to generate on our workstation, including decoding and filtering.

Attack protocol and metrics. Robbing-the-Fed, LOKI, and Scale-MIA use the same FL schedule (E, B) as the utility experiments. Reconstructions are evaluated from the aggregated global model after the first FL round unless a figure explicitly sweeps another variable; crafted FC leakage layers use the attack papers' recommended width, $k = 1024$, unless otherwise stated. We report PSNR, SSIM, and reconstruction rate (RR), where RR counts images with $\mathrm{PSNR} > 18$ dB as successful

reconstructions [36, 5]. Parenthetical PSNR/SSIM values in tables average only over successful reconstructions.

C Robbing-the-Fed under Varying Local Training Hyperparameters

Tab. A.2 supplements Sec. 3 by varying local batch size, learning rate, and number of local epochs under Robbing-the-Fed [5]. The tested utility-preserving hyperparameter changes do not materially reduce reconstruction quality, which motivates a defense mechanism that directly targets the leakage-capacity bottleneck.

Table A.2: Sensitivity of Robbing-the-Fed [5] to local training hyperparameters. The rows compare batch size, learning rate, and local-epoch choices that preserve normal training utility; attack quality remains high across these settings, showing that benign hyperparameter tuning alone is not a reliable defense.

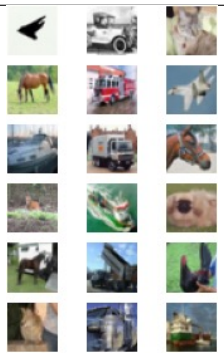
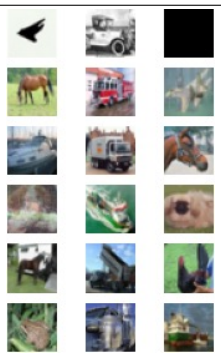
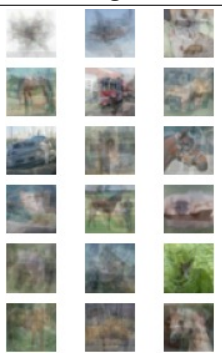
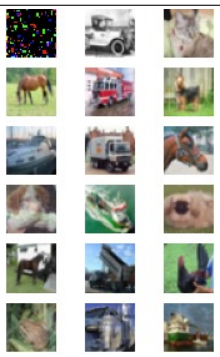
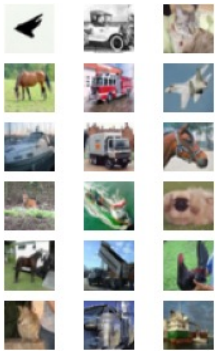
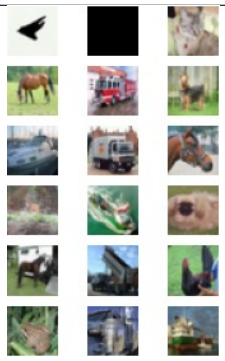
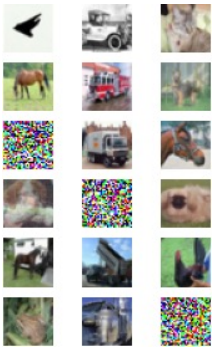
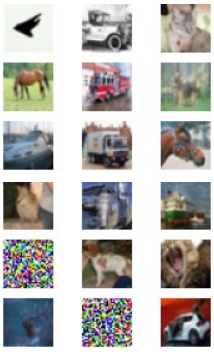
Ground Truth	Local Batch Size			Local Learning Rate (Adam)			Local Training Epoch		
	32	*64*	*128*	0.001	0.003	0.005	*1*	*3*	*5*
									
									
									
PSNR	*119.51*	*109.69*	*102.18*	*111.10*	*113.37*	*112.90*	*109.33*	*112.84*	*106.00*
SSIM	*0.99*	*0.99*	*0.99*	*0.99*	*0.99*	*0.99*	*0.99*	*0.99*	*0.99*

D Additional Experiment: Multi-baseline Comparisons on CIFAR-10

This appendix complements the medical-imaging results in Sec. 5.3 with CIFAR-10 reconstructions against a broader defense set. We compare Aegis with DP [24], Outpost [39], GD [40], Soteria [38], and GC [50], using Robbing-the-Fed [5] as the representative high-fidelity linear-leakage attack.

Tab. A.3 reports the comparison. Aegis reduces RR to 12.50% from 87.50% without defense, and the remaining reconstructions are visually blended. DP, GC, Soteria, and Outpost reduce some aggregate metrics, but their successful reconstructions retain high conditional PSNR/SSIM, indicating that recognizable inputs can still survive. These CIFAR-10 results are consistent with the medical-imaging results in Sec. 5.3: increasing the effective batch size through synthetic masking directly disrupts the closed-form leakage mechanism.

Table A.3: CIFAR-10 reconstructions by Robbing-the-Fed under six settings: no defense, Aegis, DP, GC, Soteria, and Outpost. RR is the fraction of samples with PSNR>18. Blue parenthetical PSNR/SSIM values average only over successfully reconstructed samples, separating overall protection from the fidelity of the few surviving reconstructions.

Ground Truth	**No defense**	**Aegis**	**DP**
			
RR (%)	87.50	12.50	84.38
PSNR	99.24 (112.02)	16.30 (28.30)	58.99 (63.01)
SSIM	0.96 (0.99)	0.40 (0.99)	0.96 (0.99)
Ground Truth	**GC**	**Soteria**	**Outpost**
			
RR (%)	64.06	81.25	79.69
PSNR	75.03 (112.20)	53.57 (57.14)	54.44 (57.58)
SSIM	0.95 (0.99)	0.95 (0.99)	0.95 (0.99)

E Additional Medical Utility and Privacy–Utility Results

E.1 Medical Task Accuracy

Table A.4 reports final medical test accuracy under the same 100-client non-IID FedAvg setting as Appendix B. The Eloul et al. [4] baseline uses MSE loss with same-label batching under the same FL and Robbing-the-Fed protocol. With $M = 2048$, Aegis's accuracy differs from no defense by at most 0.3 percentage points across the three datasets.

Table A.4: Final medical test accuracy (%). Aegis uses $M = 2048$.

Defense	ChestMNIST	OrganAMNIST	PathMNIST
No defense	55.8	58.6	57.3
Eloul et al.	55.7	58.5	57.2
Aegis	55.5	58.8	57.2
Soteria	53.8	56.6	55.3
GC	51.9	54.7	53.4
GD	47.2	50.0	48.7
Outpost	49.0	51.8	50.5
DP	44.7	47.5	46.2

E.2 Privacy–Utility Trade-off

Table A.5 reports the masking-batch sweep and the Eloul et al. baseline against Robbing-the-Fed, with $B = 64$, $E = 3$, and $k = 1024$. Increasing M reduces reconstruction rate, while accuracy stays within 0.4 percentage points of no defense.

Table A.5: Medical privacy–utility trade-off against Robbing-the-Fed. PSNR (dB) and SSIM are overall reconstruction metrics.

Setting	ACC (%) ↑	RR (%) ↓	PSNR ↓	SSIM ↓
ChestMNIST				
No defense	55.8	89.06	62.40	0.94
Eloul et al.	55.7	85.94	58.90	0.92
Aegis, $M = 512$	56.0	49.95	34.59	0.69
Aegis, $M = 1024$	55.9	27.03	25.06	0.55
Aegis, $M = 1536$	55.7	16.24	21.08	0.49
Aegis, $M = 2048$	55.5	9.50	18.70	0.46
OrganAMNIST				
No defense	58.6	82.81	50.60	0.89
Eloul et al.	58.5	79.69	47.90	0.87
Aegis, $M = 512$	58.8	47.82	31.19	0.69
Aegis, $M = 1024$	59.0	27.31	24.54	0.58
Aegis, $M = 1536$	58.9	17.66	21.76	0.53
Aegis, $M = 2048$	58.8	11.63	20.10	0.51
PathMNIST				
No defense	57.3	78.13	45.20	0.86
Eloul et al.	57.2	75.00	42.60	0.83
Aegis, $M = 512$	57.5	44.34	27.45	0.61
Aegis, $M = 1024$	57.6	24.53	21.36	0.47
Aegis, $M = 1536$	57.4	15.21	18.82	0.41
Aegis, $M = 2048$	57.2	9.38	17.30	0.38

F Additional Sensitivity Experiments

F.1 Generator Ablation

Table A.6 compares Stable Diffusion 2.0 with SDXL 1.0 [28], VQ-VAE-2 [32], and StyleGAN2-ADA [14], using the same $M = 2048$ cache and FL/Robbing-the-Fed configuration ($B = 64$, $E = 3$, $k = 1024$). The default public Stable Diffusion checkpoint is used without dataset-specific fine-tuning; alternatives are conditioned or adapted on a sanctioned client-side training split where needed. All variants use the same locally visible labels, preprocessing, and acceptance filter, with no test images used for generation or adaptation. Accuracy and reconstruction metrics remain close across the evaluated generators.

Table A.6: Generator ablation under the same Aegis and Robbing-the-Fed configuration.

Setting	ACC (%) ↑	RR (%) ↓	PSNR ↓	SSIM ↓
ChestMNIST				
No defense	55.8	89.06	62.40	0.94
Stable Diffusion 2.0	55.5	9.50	18.70	0.46
SDXL 1.0	55.6	9.06	18.30	0.44
VQ-VAE-2	55.4	10.63	19.20	0.48
StyleGAN2-ADA	55.3	11.25	19.60	0.49
OrganAMNIST				
No defense	58.6	82.81	50.60	0.89
Stable Diffusion 2.0	58.8	11.63	20.10	0.51
SDXL 1.0	58.7	10.94	19.60	0.49
VQ-VAE-2	58.6	12.50	20.70	0.53
StyleGAN2-ADA	58.5	12.81	21.10	0.54
PathMNIST				
No defense	57.3	78.13	45.20	0.86
Stable Diffusion 2.0	57.2	9.38	17.30	0.38
SDXL 1.0	57.3	8.75	16.90	0.36
VQ-VAE-2	57.1	10.31	17.90	0.41
StyleGAN2-ADA	57.0	10.63	18.20	0.42

F.2 Non-IID Partitions

Following Li et al. [20], Table A.7 reports distribution-based label skew, two-class label skew, feature skew, and quantity skew. All settings retain $B = 64$, $M = 2048$, $k = 1024$, and 10% client participation. The two-class constraint yields the lowest accuracy, while reconstruction rates remain close across partitions.

Table A.7: Non-IID sensitivity of Aegis against Robbing-the-Fed. Partition labels follow the experimental settings.

Setting	ACC (%) ↑	RR (%) ↓	PSNR ↓	SSIM ↓
ChestMNIST				
Original: at most 5 classes	55.5	9.50	18.70	0.46
Label Dir(0.5)	58.8	9.38	18.61	0.45
Label Dir(0.1)	53.8	9.62	18.82	0.47
Label quantity: 2 classes	43.6	9.71	18.91	0.47
Feature Gau(0.1)	59.5	9.94	19.05	0.48
Quantity Dir(0.5)	60.2	9.44	18.66	0.46
OrganAMNIST				
Original: at most 5 classes	58.8	11.63	20.10	0.51
Label Dir(0.5)	62.1	11.48	19.95	0.50
Label Dir(0.1)	57.0	11.78	20.25	0.52
Label quantity: 2 classes	46.4	11.91	20.37	0.52
Feature Gau(0.1)	62.8	12.08	20.51	0.53
Quantity Dir(0.5)	63.5	11.57	20.04	0.51
PathMNIST				
Original: at most 5 classes	57.2	9.38	17.30	0.38
Label Dir(0.5)	60.5	9.25	17.18	0.37
Label Dir(0.1)	55.4	9.51	17.42	0.39
Label quantity: 2 classes	44.8	9.62	17.54	0.39
Feature Gau(0.1)	61.2	9.78	17.72	0.40
Quantity Dir(0.5)	61.9	9.34	17.28	0.38

F.3 Classifier Backbones

Table A.8 compares classifier backbones under the original non-IID partition, with $M = 2048$ and the same crafted FC leakage front-end. All classifiers use dataset-native resolution without classifier pretraining; ViT-S/4 uses 4×4 patches. Backbone changes affect accuracy, while the reported reconstruction metrics remain close.

Table A.8: Backbone sensitivity under the same Aegis and Robbing-the-Fed setting.

Setting	ACC (%) ↑	RR (%) ↓	PSNR ↓	SSIM ↓
ChestMNIST				
3-layer CNN	55.5	9.50	18.70	0.46
ResNet-18	57.8	9.47	18.68	0.46
ViT-S/4	56.4	9.54	18.74	0.46
Swin-T	57.2	9.49	18.69	0.46
OrganAMNIST				
3-layer CNN	58.8	11.63	20.10	0.51
ResNet-18	61.4	11.59	20.07	0.51
ViT-S/4	59.9	11.67	20.14	0.51
Swin-T	60.7	11.61	20.09	0.51
PathMNIST				
3-layer CNN	57.2	9.38	17.30	0.38
ResNet-18	60.1	9.35	17.27	0.38
ViT-S/4	58.4	9.42	17.35	0.38
Swin-T	59.4	9.37	17.29	0.38

G Convergence Analysis: Assumptions and Theorem

This section contains the full assumptions, intermediate variance bounds, and convergence theorem referenced from the system design. The key convergence issue is the discrepancy between private data D_i and synthetic defense data D_i': under standard convex federated-learning assumptions [21], the budget $\mathbb{E}\|D_i - D_i'\|^2 \leq H$ enters the variance and norm constants but does not change the asymptotic $\mathcal{O}(1/T)$ rate. Our analysis extends the FedAvg framework of [21] to the Aegis setting; the only structural addition is the data-discrepancy bound between the private dataset D_i and the synthetic defense dataset D_i', formalized in Assumption 6.

Scope. *This analysis is not intended as a full convergence guarantee for the non-convex neural networks used in our experiments.* It is a stability argument under the same standard convex assumptions used in [21] (smoothness, strong convexity, bounded stochastic gradients, uniform client sampling). Its purpose is to make explicit how the auxiliary-gradient discrepancy $H = \mathbb{E}\|D_i - D_i'\|^2$ propagates into the variance and uniform-norm constants of the bound. Strong convexity is well known not to hold for deep neural networks, so the theorem below should be read as informing the design choice (smaller H tightens constants) rather than as a deep-learning convergence result. Empirical utility on deep models is reported in Sec. 5.

Assumptions

Assumption 1. *Local objective functions $F_1, \ldots, F_N$ are all L-smooth: for all v and w, $F_i(v) \leq F_i(w) + (v - w)^T \nabla F_i(w) + \frac{L}{2} \|v - w\|_2^2$.*

Assumption 2. *$F_1, \ldots, F_N$ are all μ-strongly convex: for all v and w, $F_i(v) \geq F_i(w) + (v - w)^T \nabla F_i(w) + \frac{\mu}{2} \|v - w\|_2^2$.*

Assumption 3. *Let D_i be sampled uniformly at random from the i-th device's local data. The variance of stochastic gradients on each device is bounded: $\mathbb{E}\|\nabla F_i(D_i; \theta_i^t) - \nabla F_i(\theta_i^t)\|^2 \leq \sigma_i^2$ for $i = 1, \ldots, N$.*

Assumption 4. *The expected squared norm of stochastic gradients is uniformly bounded: $\mathbb{E}\|\nabla F_i(D_i; \theta_i^t)\|^2 \leq G^2$ for all $i = 1, \ldots, N$ and $t = 1, \ldots, T - 1$.*

Assumption 5. *Let S_t contain K indices sampled uniformly without replacement from $[N]$. Assume balanced data: $p_1 = \cdots = p_N = 1/N$. The aggregation step of FedAvg is $\theta_t \leftarrow \frac{N}{K} \sum_{i \in S_t} p_i \theta_i^t$.*

Assumption 6. *(Defense-data discrepancy.) The expected squared distance between the private data D_i and the synthesized defense data D_i' is bounded: $\mathbb{E}\|D_i - D_i'\|^2 \leq H$ for all $i = 1, \ldots, N$ and $t = 1, \ldots, T-1$.*

Gradient variance under mixed private and defense data

We derive the new variance bound replacing Assumption 3. Write the per-round effective update direction as $\delta_i^t = (1-\lambda)\nabla F_i(D_i; \theta_i^t) + \lambda \nabla F_i(D_i'; \theta_i^t)$ with $\lambda \in [0, 1]$ controlled by $M_i/(B + M_i)$. Using Assumptions 1 and 6,

$$\begin{aligned}
&\mathbb{E}\left\|\delta_i^t - \nabla F_i(\theta_i^t)\right\|^2 \\
&\leq \mathbb{E}\left\|\nabla F_i(D_i; \theta_i^t) - \nabla F_i(\theta_i^t)\right\|^2 + \mathbb{E}\left\|\delta_i^t - \nabla F_i(D_i; \theta_i^t)\right\|^2 \\
&\leq \sigma_i^2 + \mathbb{E}\left\|\delta_i^t - \nabla F_i(D_i; \theta_i^t)\right\|^2 \\
&= \sigma_i^2 + \mathbb{E}\left\|\lambda(\nabla F_i(D_i'; \theta_i^t) - \nabla F_i(D_i; \theta_i^t))\right\|^2 \\
&\leq \sigma_i^2 + \mathbb{E}\left\|\nabla F_i(D_i'; \theta_i^t) - \nabla F_i(D_i; \theta_i^t)\right\|^2 \\
&\leq \sigma_i^2 + L\,\mathbb{E}\left\|D_i - D_i'\right\|^2 \\
&\leq \sigma_i^2 + LH. \qquad \text{(A.1)}
\end{aligned}$$

By the same derivation, the uniform bound in Assumption 4 becomes $\mathbb{E}\|\nabla F_i(D_i; \theta_i^t)\|^2 \leq G^2 + LH$. Define F^* and F_i^* as the minimum of F and F_i, and set $\Gamma = F^* - \sum_{i=1}^N p_i F_i^*$.

Convergence theorem

Theorem 1. *Under Assumptions 1–6, with $\kappa = L/\mu$, $\gamma = \max\{8\kappa, E\}$, and the diminishing learning-rate schedule $\eta_t = 2/(\mu(\gamma + t))$, after T communication rounds*

$$\mathbb{E}[F(\theta^T)] - F^* \;\leq\; \frac{\kappa}{\gamma + T - 1}\left(\frac{2(B+C)}{\mu} + \frac{\mu\gamma}{2}\,\mathbb{E}\left\|\theta^1 - \theta^*\right\|^2\right), \qquad \text{(A.2)}$$

where

$$B = \sum_{i=1}^N p_i^2(\sigma_i^2 + LH) + 6L\Gamma + 8(E-1)^2(G^2 + LH), \qquad \text{(A.3)}$$

$$C = \frac{N-K}{N-1}\,\frac{4}{K}\,E^2(G^2 + LH). \qquad \text{(A.4)}$$

Remark. Setting $H = 0$ (i.e., D_i' identical in distribution to D_i) recovers the bound of [21], so Aegis inherits the standard $\mathcal{O}(1/T)$ rate; $H > 0$ enlarges the constants B and C by a controlled amount but does not affect the rate.

H Survey of Gradient-based MIAs in Federated Learning

Tab. A.9 summarizes representative MIAs by whether they break SA, the batch-size constraint at which they remain effective, and whether they rely on iterative optimization. The table substantiates the introduction's central observation: the local batch size relative to the first-FC-layer capacity is the unifying bottleneck across the entire literature, and Aegis turns this bottleneck into a defense.

I Extended Discussion and Limitations

Deployment assumptions. Aegis is designed for the server-side MIA threat model studied in recent linear-leakage attacks [5, 48, 36]: clients follow the FL protocol, while the server may broadcast a model crafted for reconstruction. Client-side poisoning, backdoors, and malicious client behavior are orthogonal FL security problems and can be combined with Aegis through existing robust-aggregation or poisoning-defense tools.

Table A.9: Representative gradient-based MIAs in FL. "Break SA" indicates attacks demonstrated under secure aggregation or equivalent aggregate-only settings. "Batch Size Constraint" reports the largest batch scale at which the attack is typically effective under a first crafted FC layer with 1024 neurons, highlighting why leakage capacity is the relevant bottleneck for Aegis.

MIA	Break SA	Batch Size	Optimization
DLG [50]	No	1	Yes
iDLG [47]	No	1	Yes
Inverting Gradients [6]	No	8	Yes
GradInversion [45]	No	48	Yes
GradViT [10]	No	8	Yes
APRIL-Optim [22]	No	1	Yes
APRIL-Analytic [22]	No	1	Yes
R-GAP [49]	No	1	Yes
Fishing for Data [41]	Yes	256	Yes
Eluding SA [26]	Yes	512	Yes
Robbing-the-Fed [5]	Yes	1024	No
LOKI [48]	Yes	1024	No
Scale-MIA [36]	Yes	1024	No

Generator choice. Aegis only requires a class- or text-conditional generator whose samples are close enough to the client modality to enter the same leakage bins as real data. Public pretrained generators, domain-adapted generators, or generators fine-tuned on sanctioned in-domain corpora are all compatible. The defense data remains local and is used only for masking-gradient computation, so it is not revealed to the server. Better-matched generators reduce the in-distribution budget H in Appendix G, which helps preserve utility while maintaining bin collisions.

Hyperparameters and resources. The main defense parameters are M_i and H. M_i should make $B + M_i$ exceed the leakage-bin capacity k, so the required defense batch scales with the capacity of the crafted layer; our default $M_i = 2048$ is larger than the $k = 1024$ setting used by the evaluated linear-leakage attacks. H is selected with a small client-side validation split to balance synthetic diversity and utility. When M_i does not fit in memory, clients can compute the same logical masking gradient by accumulating micro-batches; this preserves the collision argument and trades memory for wall-clock time.

Adaptive servers. The evaluated attacks cover the known scalable linear-leakage family. A server could increase the crafted FC width, in which case Aegis requires M_i to scale with the larger leakage capacity; choosing M_i under unknown or adaptive k is therefore a deployment consideration. A server could also try to estimate and remove synthetic contributions, but each leakage bin mixes unknown real and synthetic samples, making demixing underdetermined without client-side sample identities. These observations do not preclude future attack variants, but they show that Aegis targets the structural bottleneck shared by current closed-form attacks.

Scope of the evidence. The convergence result is a stability statement under standard convex assumptions for the federated setting; empirical utility in the non-convex deep-learning regime is evaluated directly in Sec. 5. The medical experiments cover three MedMNIST modalities at benchmark resolution. Applying the same mechanism to higher-resolution volumes or whole-slide pathology mainly changes the generator and compute requirements, not the client-side masking protocol.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Sec. 1 state the MIA threat model, the proposed Aegis defense, and the standard/medical benchmark evaluation. The empirical claims are supported in Sec. 5.3, and the theory claim is scoped in Appendix G.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Appendix I discusses deployment assumptions, generator requirements, adaptive-server considerations, resource constraints, and the scope of the medical experiments. Sec. 5 also notes that the reported medical studies use benchmark-resolution MedMNIST modalities.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Appendix G states Assumptions 1–6, derives the defense-specific variance terms, and gives the corresponding convergence theorem. We will provide any remaining standard proof details in the supplementary material while keeping the non-convex deep-learning scope explicit.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Sec. 5.1 and Appendix B specify the FL setup, datasets, model architecture, optimizer, baselines, defense batch size, attack protocol, and metrics. Sec. 5.3 and Appendix D report the main quantitative comparisons.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The datasets used in Sec. 5.1 are public benchmarks and are cited, and the implementation-level procedure is documented in Algorithm 1, with further details in Appendix A. We will provide anonymized code and step-by-step reproduction instructions in the supplementary material.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Sec. 5.1 gives the high-level experimental setup, and Appendix B gives data partitioning, optimizer, model architecture, attack protocol, generation setup, and baseline hyperparameters. Metric definitions are provided in Sec. 5.1 and Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: Appendix B states that quantitative metrics are averaged over multiple independent runs with different random seeds. We will report corresponding variability information, such as error bars or standard deviations, for the plotted and tabulated results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix B reports the CPU/GPU workstation used for timing and training experiments. Tab. 1 reports wall-clock training time for representative MNIST and CIFAR-10 FL runs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper studies a privacy-preserving FL defense using public benchmark datasets, as described in Sec. 5.1 and Appendix B. The threat model and limitations in Sec. 3 and Appendix I address misuse risks from reconstruction attacks.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Sec. 1 motivates the positive impact of protecting sensitive medical FL data. Sec. 3 and Appendix I discuss negative misuse risks from active servers and adaptive reconstruction attacks.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release a new high-capacity generative model, scraped dataset, or other high-risk asset. It uses public benchmarks and existing generative components, as described in Sec. 5.1 and Appendix B.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Existing datasets, models, attacks, and defense baselines are credited through citations in `main.bib` and discussed in Sec. 5.1 and Appendix B. We will explicitly list the relevant licenses and terms of use for each asset in the supplementary material or final artifact documentation.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper does not introduce a new public dataset, benchmark, or model asset. The contribution is a client-side defense mechanism described in Sec. 4 and Algorithm 1.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The experiments use public ML benchmarks and do not involve crowdsourcing or direct human-subjects studies. The datasets are listed in Sec. 5.1 and Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No new human-subjects data were collected, and the experiments use public benchmark datasets listed in Sec. 5.1. Therefore IRB approval is not applicable to this study.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not an original or non-standard component of the proposed defense or experiments. Defense-data synthesis uses conditional image generators, as described in Sec. 4 and Sec. 5.1.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.