SVDefense: Effective Defense against Gradient Inversion Attacks via Singular Value Decomposition

Chenxiang Luo
City University of Hong Kong
chenxiluo8-c@my.cityu.edu.hk

David K.Y. Yau
Singapore University of Technology and Design
david_yau@sutd.edu.sg

Qun Song*
City University of Hong Kong
qunsong@cityu.edu.hk

***Abstract*—Federated learning (FL) enables collaborative model training without sharing raw data but is vulnerable to gradient inversion attacks (GIAs), where adversaries reconstruct private data from shared gradients. Existing defenses either incur impractical computational overhead for embedded platforms or fail to achieve privacy protection and good model utility at the same time. Moreover, many defenses can be easily bypassed by adaptive adversaries who have obtained the defense details. To address these limitations, we propose *SVDefense*, a novel defense framework against GIAs that leverages the truncated Singular Value Decomposition (SVD) to obfuscate gradient updates. *SVDefense* introduces three key innovations, a Self-Adaptive Energy Threshold that adapts to client vulnerability, a Channel-Wise Weighted Approximation that selectively preserves essential gradient information for effective model training while enhancing privacy protection, and a Layer-Wise Weighted Aggregation for effective model aggregation under class imbalance. Our extensive evaluation shows that *SVDefense* outperforms existing defenses across multiple applications, including image classification, human activity recognition, and keyword spotting, by offering robust privacy protection with minimal impact on model accuracy. Furthermore, *SVDefense* is practical for deployment on various resource-constrained embedded platforms. We will make our code publicly available upon paper acceptance.**

I. INTRODUCTION

Federated learning (FL) has emerged as a promising paradigm for collaborative model training that preserves user privacy in domains such as healthcare [1], finance [2], and social safety [3]. In FL, distributed clients train models on their local datasets and only share model updates with a central server, eliminating the need to expose sensitive user data [4]. However, recent research has revealed that FL systems remain vulnerable to gradient inversion attacks (GIAs) [5]–[7], where adversaries can reconstruct private training data by exploiting the gradients shared during client-server communication. These attacks have shown the ability to reconstruct private training data with high fidelity [7]–[9], which raises substantial privacy risks in FL systems.

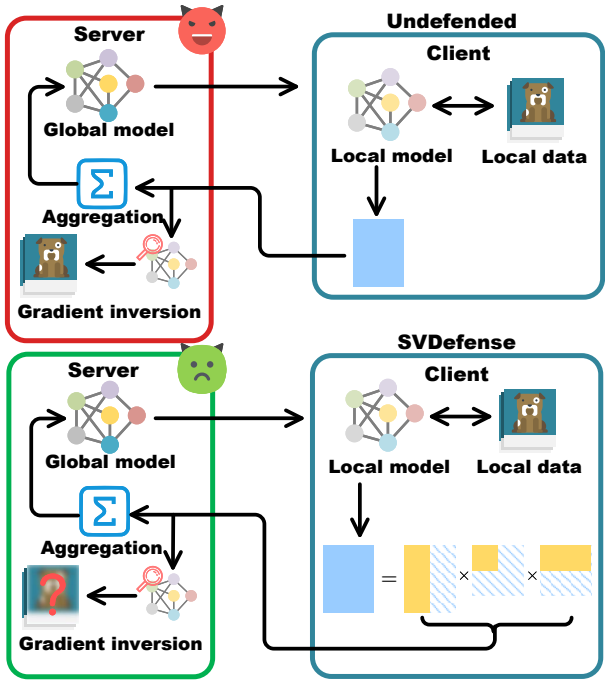


Fig. 1: An illustration of *SVDefense*. While the adversary may attempt to reconstruct user data using the gradients uploaded by an undefended client, our defense hinders the data reconstruction by uploading the gradients protected by *SVDefense*. The blue and yellow blocks represent the original and truncated gradients in our defense, respectively.

While various defense mechanisms against GIAs have been proposed, they face significant limitations in practical FL deployments. Encryption-based methods such as secure multi-party computation (SMC) [10], [11] and homomorphic encryption (HE) [12], [13] provide strong theoretical privacy guarantees but introduce prohibitive computational overhead for resource-constrained devices. Perturbation-based defenses counteract GIAs by modifying local inputs [14]–[17], gradients [5], [18]–[21], or training processes [22]–[24]. Defenses that perturb local inputs [14]–[17] or add noise to local

*Corresponding author.

Network and Distributed System Security (NDSS) Symposium 2026
23-27 February 2026, San Diego, CA, USA
ISBN 979-8-9919276-8-0
https://dx.doi.org/10.14722/ndss.2026.230114
www.ndss-symposium.org

gradients, such as differential privacy (DP) and its variants [5], [18]–[20], often fail to balance privacy protection and model utility [20], [25]. In our analysis, methods that perturb local gradients [21] and training processes [23], [24] can be bypassed by adaptive adversaries who have obtained details of the defense mechanisms. Pruning-based defenses [5], [25], [26] that selectively remove gradient components to counteract GIAs can also be bypassed by adaptive attackers, as shown in our analysis. Unlike previous work [9], [27] that considers less practical adaptive attacks relying on strong assumptions, we demonstrate the vulnerability of these defenses under GIAs augmented with practical adaptive operations.

This paper introduces *SVDefense*, a novel defense framework against GIAs based on the truncated Singular Value Decomposition (SVD). Truncated SVD is a matrix factorization technique that approximates a matrix with the goal of effectively reducing its dimensionality while preserving important information. Our design is motivated by key insights from the analysis of existing defenses under practical adaptive attacks, which suggests that affecting all the gradient components and doing so irreversibly are desirable properties that offer improved robustness against adaptive GIAs. Moreover, it is crucial for defense mechanisms to balance defense performance with model utility. Truncated SVD has the potential to counteract adaptive GIAs via gradient decomposition and truncation that irreversibly modifies the entire gradient space while preserving model utility. However, applying truncated SVD against GIAs presents practical challenges. Our preliminary study finds that clients with higher degrees of class imbalance are more vulnerable to attacks. Thus, more imbalanced clients should be given stronger protection from GIAs through lower energy thresholds $\mathcal{T}$ that reduce the information in truncated gradients for data reconstruction. This strategy raises three key questions during the SVD truncation: (1) How to effectively quantify the varying degrees of class imbalance and adaptively adjust the energy thresholds $\mathcal{T}$ for different clients; (2) How to preserve information critical for model training while suppressing sensitive information leakage; and (3) How to effectively aggregate SVD truncated client updates and thereby improve global model utility under the class imbalance?

To address these challenges, *SVDefense* introduces three key innovations. First, we observe that the distribution of the singular values obtained in SVD strongly correlates with the degree of class imbalance. Hence, we propose a *Self-Adaptive Energy Threshold* that adaptively adjusts the energy threshold for each client based on its singular value distribution, providing stronger protection to class-imbalanced clients who are more susceptible to GIAs. Second, whereas lowering the energy threshold in SVD truncation provides stronger protection against GIAs, it also reduces the information needed for effective model training. To address this issue, we propose a *Channel-Wise Weighted Approximation* that strategically assigns weights to gradients during the SVD truncation, which preserves gradients that are critical for model performance while suppressing potential sensitive information leakages, leading to better model accuracy and defense performance. Third, non-IID local data distributions, such as class imbalance across the clients, lead to degraded global model accuracy due to client drift [24]. Existing studies have not addressed how to effectively aggregate SVD truncated updates to improve the global model's utility under such heterogeneous data distributions. Our *Layer-Wise Weighted Aggregation* addresses this gap by leveraging key correlation between singular value distributions and local class imbalance. By strategically assigning layer-wise aggregation weights to client updates based on their singular value distributions, we effectively improve the global model's accuracy under class-imbalanced data. Together, the three components enable *SVDefense* to achieve strong privacy protection while maintaining model utility across diverse FL scenarios. An illustration of how *SVDefense* works is shown in Fig. 1.

Our extensive evaluation on the EMNIST [28], CIFAR-10 [29], HAR [30], and Google Speech Commands [31] datasets demonstrates that *SVDefense* achieves superior performance in both model accuracy and defense effectiveness compared with various representative defenses. Moreover, we implement a real-world FL testbed on various embedded platforms, including Raspberry Pi, Nvidia Jetson Orin Nano, and Nvidia Jetson TX2, to validate the practicality of *SVDefense*. Our experiments show that *SVDefense* has high computational efficiency on various resource-constrained embedded platforms and significantly reduces communication cost.

Our contributions are summarized as follows.

- We systematically analyze various representative defenses against GIAs, demonstrating their vulnerability to practical adaptive attacks. We derive key insights from our analysis to identify truncated SVD as a promising technique for leveraging these insights.
- We develop *SVDefense*, a novel defense framework against adaptive GIAs based on truncated SVD. To the best of our knowledge, ours is the first comprehensive solution to address practical challenges of defending against GIAs under non-IID data distributions caused by class imbalance across FL clients. We introduce the Self-Adaptive Energy Threshold to adapt the privacy protection for clients based on their varying degrees of vulnerability to GIAs caused by their respective levels of class imbalance, the Channel-Wise Weighted Approximation to enhance both accuracy and defense performance, and the Layer-Wise Weighted Aggregation for effective aggregation of SVD truncated client updates to improve global model accuracy under the class imbalance.
- Our extensive evaluation on various datasets demonstrates that *SVDefense* outperforms existing defenses in both model accuracy and defense effectiveness. We also implement our solution in a real-world FL testbed using various embedded platforms. Experimental results show that *SVDefense* achieves practical computational cost and significantly reduces communication cost.

II. BACKGROUND AND RELATED WORK

A. Gradient Inversion Attacks

Existing GIAs can be categorized into optimization-based and GAN-based attacks.

Optimization-based Attacks: Deep Leakage from Gradients (DLG) [5] is the first work that demonstrates the feasibility of reconstructing local data and corresponding labels from the shared gradients by iteratively optimizing dummy inputs to match the shared gradients using L-BFGS optimization. Improved DLG (iDLG) [32] enhances data reconstruction effectiveness by extracting ground-truth labels from gradient signs. Inverting Gradients (IG) [6] attack improves over the early studies of DLG and iDLG by employing the Adam optimizer to stabilize convergence and introduces cosine similarity as a more effective gradient matching objective. GradInversion [33] can reconstruct high-fidelity input image batches using the gradient matching objective with fidelity regularization and group consistency regularization terms to improve reconstruction quality. However, its reliance on input batch normalization statistics makes it impractical in typical FL settings.

GAN-based Attacks: Generative adversarial networks (GANs) [34] are generative models capable of capturing the probability distribution of images from the training set. Recent advances in GIAs leverage GANs as *image* priors to compensate for information loss and enhance reconstruction quality. GIAS [35] alternately optimizes latent vectors and generator parameters to improve image reconstruction fidelity. GGL [8] employs pre-trained GANs as priors to constrain the image reconstruction. GIFD [36] sequentially explores the latent space and intermediate features of the generator under an l_1-ball constraint to address limitations of expressiveness and generalization in pre-trained GANs. A recent strong attack ROG [7] encodes raw images into low-dimensional representations to improve attack optimization efficiency, followed by GAN-based post-processing to enhance image reconstruction quality.

Adaptive Attacks: Research has shown that an *adaptive adversary* [9], [37], [38] with knowledge of the defense (e.g., an honest-but-curious aggregator in FL who has legitimate access to the defense mechanism) can design targeted attacks against it. By formulating GIAs within a Bayesian framework, the work [27] demonstrates how a Bayes optimal adversary can break several heuristic defenses. However, the theoretical analysis is based on specific neural network architectures. Learning To Invert (LTI) [9] trains a gradient inversion model to invert gradients protected by defenses including sign compression, gradient pruning, and gradient perturbation. However, LTI assumes that the adversary has access to the private data distribution. The work [22] shows how Dropout's effectiveness as a defense against GIAs can be mitigated by modeling dropout-induced stochasticity during attack optimization. While prior efforts [9], [27] shed light on how adaptive adversaries can circumvent certain defenses, their findings are limited by strong assumptions about specific model architectures or attacker's capabilities. In this paper, we extensively investigate the vulnerability of various representative defenses under realistic adaptive adversaries by augmenting existing GIAs with practical adaptive operations and further propose a novel defense framework.

B. Gradient Inversion Defenses

Existing defenses against GIAs can be categorized into encryption-based, perturbation-based, pruning-based, and compression-based methods.

Encryption-based defenses employ cryptographic techniques to protect client updates in FL. SMC protocols [10], [11] enable secure aggregation of client updates without revealing individual contributions. Some studies [12], [13] leverage HE to perform arbitrary computations on encrypted gradients. Several efforts [39], [40] combine DP with HE or SMC to provide formal privacy guarantees while allowing encrypted gradient aggregation. Although these defenses offer theoretical privacy guarantees, they introduce significant computational, communication, and storage overhead and often necessitate modifications to FL architectures, making them less practical.

Perturbation-based defenses can be further divided into three sub-categories. *Input perturbation* modifies the local training data. Approaches include creating composite images through linear combinations [14], [16], applying strategic data augmentation [15], [41], and synthesizing visually distinct concealed samples to mimic sensitive data at the gradient level [17]. However, these methods often compromise classification performance, provide insufficient protection, or incur high computational overhead [25]. *Gradient perturbation* modifies the local gradients. Early defenses [5] leverage DP by adding Gaussian and Laplace noises to the gradients. The work [18] applies per-example gradient clipping and DP noise injection during local training. The work [19] adds layer-wise random perturbations to gradients based on information leakage risk. Outpost [20] adaptively adds Gaussian noise combined with gradient pruning during each local training iteration based on privacy leakage risks. However, these noise injection-based defenses struggle to balance good defense performance and model utility [6], [25]. CENSOR [21] samples gradients from a subspace orthogonal to the original gradients while using cold Bayesian posteriors aiming to improve model utility. *Training perturbation* perturbs local training processes. The work [22] adds dropout layers in local models during training aiming to mitigate GIAs. PRECODE [23] adds a variational bottleneck prior to the output layer of the local model to counteract GIAs while maintaining classification performance. The learning-rate-perturbation (LRP) [24] randomly perturbs each client's learning rate to prevent accurate data reconstruction while preserving model accuracy. However, LRP only modifies the gradient scale without affecting the direction, making it vulnerable to strong attacks like IG that employ cosine similarity loss.

Pruning-based defenses selectively remove gradient components. Prune [5] sets gradients with small magnitudes to zero. Soteria [26] identifies that the data representations, i.e.,

the outputs of the layers after the feature extractor, inferred from the gradients reveal significant information about the input data. It then prunes the selected gradients to perturb the data representations. The work [42] proposes pruning large gradients to defend against GIAs. Dual Gradient Pruning (DGP) [25] prunes both large and small gradients to conceal label information while incorporating an error feedback mechanism [43] that adds back the pruned gradients in the next training step to mitigate information loss caused by pruning. **Compression-based** defenses mitigate information leakage by compressing gradients. pFGD [44] combines Discrete Cosine Transform and gradient pruning to suppress sensitive frequency components. Mixed Quantitation (MQ) [45] assigns varying quantization precisions across model layers. Existing defenses directly apply gradient compression techniques without tailoring them to defend against GIAs or considering realistic non-IID scenarios. In comparison, our truncated SVD-based defense adapts to client vulnerability, improves accuracy and defense performance via weighted approximation, and enhances aggregation effectiveness under class imbalance.

C. Singular Value Decomposition

Low-rank approximation is the process of approximating a matrix $\mathbf{W}$ by a matrix $\hat{\mathbf{W}}$ of lower rank. Formally, the objective is to minimize the approximation error $||\mathbf{W} - \hat{\mathbf{W}}||$ subject to $\text{rank}(\hat{\mathbf{W}}) \leq k$, where k is the desired reduced rank. SVD can solve this problem effectively [46]. For a matrix $\mathbf{W} \in \mathbb{R}^{p\times q}$ with $p \geq q$, SVD decomposes it as $\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, where $\mathbf{U} \in \mathbb{R}^{p\times r}$ is an orthogonal matrix of left singular vectors, $\mathbf{\Sigma} = \text{diag}(\sigma_1, \cdots, \sigma_r) \in \mathbb{R}^{r\times r}$ contains singular values in descending order $(\sigma_1 \geq \cdots \geq \sigma_r)$, $r \leq \min\{p, q\}$ is the rank of $\mathbf{W}$, and $\mathbf{V}^\top \in \mathbb{R}^{r\times q}$ is the transpose of an orthogonal matrix of right singular vectors. Truncated SVD approximates $\mathbf{W}$ by retaining only the k largest singular values and their corresponding singular vectors as $\hat{\mathbf{W}} = \mathbf{U}'\mathbf{\Sigma}'\mathbf{V}'^\top$, where $\mathbf{U}' \in \mathbb{R}^{p\times k}$, $\mathbf{\Sigma}' \in \mathbb{R}^{k\times k}$, and $\mathbf{V}'^\top \in \mathbb{R}^{k\times q}$. The number of retained singular values is determined by the energy threshold $\mathcal{T}$, where energy refers to the sum of the squared singular values representing the amount of information captured by the singular values [47]. Specifically, k is chosen to satisfy $\min_k \frac{\sum_{i=1}^{k}\sigma_i^2}{\sum_{i=1}^{r}\sigma_i^2} > \mathcal{T}$. If $pk+k+kq < pq$, the truncated matrices contain fewer parameters than the original matrix $\mathbf{W}$, thus reducing communication cost. SVD is commonly used in principal component analysis (PCA) for dimensionality reduction [48], latent semantic analysis (LSA) for feature extraction in natural language processing [49], and addressing the challenge of sharing large-scale model updates in FL [50]–[52]. Unlike prior works, we explore truncated SVD to defend against adaptive GIAs.

III. THREAT MODEL

FL Setting. We consider a standard FL setting with a central server and M distributed clients. Each client m has local training data $D_m = \{(\mathbf{x}_{m,n}, y_{m,n})\}_{n=1}^{N_m}$ with N_m samples. In communication round t, the server first sends the global model parameters $\mathbf{\Theta}_g^{t-1}$ to a batch of B^t selected clients and sets their local model parameters to $\mathbf{\Theta}_m^{t-1} = \mathbf{\Theta}_g^{t-1}$. Then, the clients perform local training on D_m and update their respective local models from $\mathbf{\Theta}_m^{t-1}$ to $\mathbf{\Theta}_m^t$. In our setting, each client m transmits its local model gradients $\nabla\mathbf{\Theta}_m$ to the server, which has the same effect as sending the updated local model parameters since $\nabla\mathbf{\Theta}_m = \mathbf{\Theta}_g^{t-1} - \mathbf{\Theta}_m^t$. Finally, the server updates the global model as $\mathbf{\Theta}_g^t = \mathbf{\Theta}_g^{t-1} - \sum_{b=1}^{B^t} p_b \nabla\mathbf{\Theta}_b$, where p_b is the normalized aggregation weight of the b-th selected client. The objective is to collaboratively train a global model $\mathbf{\Theta}_g$ by aggregating client updates. We follow the FedAvg algorithm [4] and formulate the objective as $\arg\min_{\mathbf{\Theta}_g} \sum_{m=1}^{M}\sum_{n=1}^{N_m} \mathcal{L}(F_{\mathbf{\Theta}_g}(\mathbf{x}_{m,n}), y_{m,n})$, where $F_{\mathbf{\Theta}_g}(\cdot)$ is the neural network with parameters $\mathbf{\Theta}_g$ and $\mathcal{L}(\cdot,\cdot)$ is the loss function used to train it.

Adversary Model and Capabilities. We consider an *honest-but-curious* central server as the adversary. Such an adversary is common in FL research [5], [6], [20], [26]. This adversary follows the FL protocol properly but attempts to reconstruct clients' private training data from their uploaded updates. The adversary has access to the global model parameters $\mathbf{\Theta}_g$ and local model gradients $\nabla\mathbf{\Theta}_m$ but cannot modify the training process or tamper with model parameters. Furthermore, we assume an *adaptive adversary* who knows the deployed defense mechanisms and can design targeted attacks against these defenses. We assume that the adversary has enough computational resources to perform attacks.

Adversary Goals. The adversary aims to reconstruct private data from local model gradients via GIAs. Given client m's gradients $\nabla\mathbf{\Theta}_m$, the adversary first initializes pairs of dummy input data $\mathbf{x}'_m$ and label y'_m, which are the optimizable parameters for data recovery. After forward and backward propagation on the global model, the dummy gradients $\nabla\mathbf{\Theta}'_m = \nabla\mathcal{L}(F_{\Theta_g}(\mathbf{x}'_m), y'_m)$ can be generated. The reconstruction of private data $\mathbf{x}_m$ can be viewed as an iterative optimization process with the objective of minimizing the distance between the dummy gradients and the ground-truth gradients of the victim client, which can be formulated as $\arg\min_{\mathbf{x}'_m, y'_m} \text{Dist}(\nabla\mathbf{\Theta}'_m, \nabla\mathbf{\Theta}_m) + \mathcal{R}(\mathbf{x}'_m)$, where $\mathcal{R}(\cdot)$ is the regularization term.

IV. MOTIVATION STUDY

This section presents a preliminary study on the MNIST dataset [53], demonstrating that existing GIA defenses can be circumvented by practical adaptive adversaries. We summarize key insights from our study to motivate our defense design.

Experimental Setup. We consider five representative defenses, including two perturbation-based defenses, *CENSOR* [21] and *PRECODE* [23], and three pruning-based defenses, *Prune* [5], *Soteria* [26], and *DGP* [25]. In this section, we omit perturbation-based defenses that rely on noise injection, including DP [5] and Outpost [20]. This is because Expectation over Transformation (EoT), a practical adaptive attack operation commonly considered for mitigating random effects induced by the defenders [21], [37], [54], [55], is

TABLE I: Defense Performance of Different Methods Under Non-adaptive and Adaptive GIAs. Higher PSNR and Lower LPIPS Values Mean Stronger Attack Performance.

Defense \ Metric	Non-adaptive		Adaptive	
	PSNR	LPIPS	PSNR	LPIPS
CENSOR [21]	8.1940	0.6958	16.4071	0.2881
PRECODE [23]	3.5659	0.7668	57.4165	0.0001
Prune [5]	12.9257	0.4993	36.1273	0.0221
Soteria [26]	10.8145	0.6481	38.7447	0.0161
DGP [25]	9.7334	0.6187	35.6383	0.0254

ineffective against these noise injection-based defenses, as theoretically analyzed in Appendix A. However, we show (in §VI-C) that under a more powerful, thus less practical, adaptive adversary, our defense still outperforms the existing defenses, which demonstrates the superior performance of our proposed solution even in stressful situations. We configure the defenses following the recommended settings in their respective publications to reproduce their best performance. For CENSOR, we activate the defense for the first five epochs, following [21]. For PRECODE, we adopt the same parameter settings as in [23]. For Prune, we set the pruning rate to 90%. For Soteria, we prune 80% of the gradients in the fully connected layers. For DGP, we prune the smallest 75% as well as the largest 5% of the gradients.

We evaluate the defenses against both non-adaptive and adaptive attacks. For the non-adaptive scenarios, we use the standard IG attack. For the adaptive scenarios, we apply defense-specific operations on the original IG attack. For CENSOR, the adversary is assumed to know that the defense is active for the initial epochs. We then perform the attack in the subsequent undefended epochs and report the best attack results. Note that if the defense keeps active for all the epochs in CENSOR, the model's utility will degrade significantly, as shown in §VI-B. For PRECODE, we initialize a dummy random vector and optimize it together with the dummy inputs during attack optimization. For the pruning-based defenses (i.e., Prune, Soteria, and DGP), we assume that the adversary knows the defense and details of the pruning operation by detecting the zero values in the ground-truth gradients. We then apply identical pruning operations to both the ground-truth and dummy gradients during attack optimization. Since these defense mechanisms and their parameters (e.g., the number of initial epochs where defense is deployed) are static information, the adversary can obtain them in an advanced persistent threat (APT) scenario [56], where the adversary may use, for instance, social engineering against the FL clients to exfiltrate the needed knowledge. Note that we omit GAN-based attacks in this discussion because they yield results similar to optimization-based attacks (validated in §VI-C) due to the two approaches' shared optimization objective, as discussed in §III. Consequently, the adaptive operations in this section are compatible with both attack types.

Experimental Results. Table I summarizes the peak signal-to-noise ratio (PSNR) [57] and the learned perceptual image patch similarity (LPIPS) [58] between the ground-truth and reconstructed images under the non-adaptive and adaptive GIAs. Higher PSNR values mean better reconstruction quality and lower LPIPS values mean smaller perceptual differences between the original and reconstructed images, both indicating stronger attack performance. Although LRP is not effective against the strong IG attack, we still show that its defense performance drops under the adaptive DLG attack [5] in Appendix B. These results demonstrate that the effectiveness of the subject defenses drops significantly under adaptive attacks. Fig. 2 exemplifies how an adaptive attack works against the Prune defense. From the figure, while the non-adaptive attack directly matches gradients without considering the defense, the adaptive attack applies the same pruning operation to both the ground-truth and dummy gradients during optimization, thereby achieving better reconstruction performance.

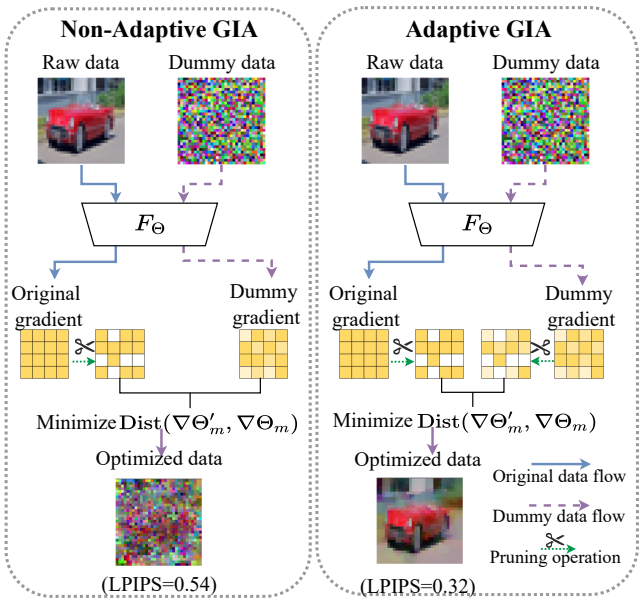


Fig. 2: Comparison of non-adaptive and adaptive GIAs against the Prune defense [5].

Key Insights. Our experimental results demonstrate key vulnerabilities of the existing defenses under practical adaptive adversaries. First, pruning-based defenses can be adaptively attacked by identifying zeroed gradient components and exploiting the unaffected ones for reconstruction. Second, defenses relying on random variables (e.g., PRECODE and LRP) can be bypassed through variable recovery. These empirical findings suggest that affecting all the gradient components and doing so irreversibly are desirable properties for improved robustness against adaptive GIAs. Third, by applying random orthogonal projection to gradients in the initial epochs only, CENSOR is vulnerable to attacks during the later undefended epochs, where the adversary can still extract sufficient information to reconstruct much of the input data, as evidenced by our experiments. If the random projection were to be kept active to maintain privacy, the totality of projected gradients would

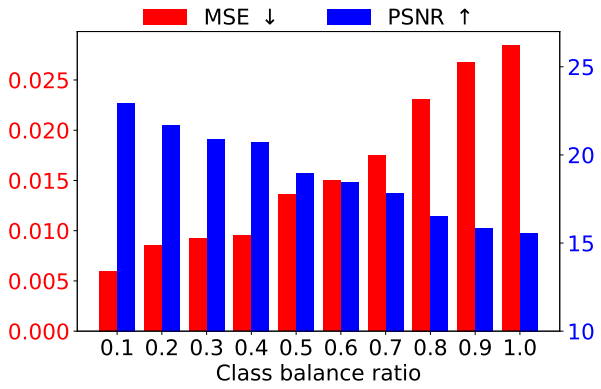


Fig. 3: Impact of class imbalance on attack effectiveness. Lower MSE and higher PSNR indicate stronger attack performance.

retain so little information that it significantly degrades the model's utility. Therefore, it is essential for practical defenses to achieve both privacy protection and good model utility. These insights motivate our proposed solution based on truncated SVD. On the one hand, truncated SVD irreversibly affects all the gradient components. On the other hand, it prudently truncates the gradients while preserving critical information for model utility at low computational and communication overheads. However, applying truncated SVD as a robust GIA defense presents practical challenges, which we will address in the next section.

V. System Design

A. Challenges and Design Goals

In real-world FL deployments, data distributions among clients are often not independent and identically distributed (non-IID). One of the most common non-IID scenarios is the imbalanced distribution of classes [59]–[61], where clients possess varying proportions of data samples across different classes. For example, different hospitals may observe different frequencies of disease types based on their specialties and patient demographics. While prior efforts [59]–[62] have focused on improving model utility under non-IID data in FL, the impact of such data heterogeneity on gradient inversion attacks and defenses remains unexplored.

In our investigation, we simulate the non-IID scenario of class imbalance using the MNIST dataset and adopt ResNet-18 [63] as our target model. We first randomly shuffle the order of the data classes and retain $N_{cls\_i} \times \rho^{cls\_i}$ samples for each class, where cls_i, ($cls\_i = 0, \cdots 9$) is the shuffled class index, N_{cls_i} is the original number of samples in class cls_i, and ρ is our defined *class balance ratio*. By varying ρ from 0 to 1 with a step size of 0.1, we simulate different degrees of class imbalance, where a larger ρ indicates a more balanced class distribution. For each ρ, we generate 128 batches of input samples with a batch size of 10. Then, we launch the IG attack. As shown in Fig. 3, as ρ increases, the mean squared error (MSE) between the original and reconstructed images increases and the PSNR value decreases. This suggests that clients subject to higher degrees of class imbalance are more vulnerable to attacks. This is potentially because, when trained on class-imbalanced data, the model's gradients primarily reflect patterns from the dominant classes. Therefore, it becomes easier for attackers to reconstruct private data from the less diverse gradients.

Based on the above observations, it is inadvisable to treat clients with varying degrees of class imbalance uniformly when counteracting GIAs through SVD truncation. Since more imbalanced clients are more vulnerable to attacks, they require stronger protection through lower energy thresholds that consequently reduce available information in the truncated gradients for data reconstruction, as illustrated in Appendix C. This strategy raises three practical challenges during SVD truncation: **Challenge C1:** How to effectively quantify the degree of class imbalance and thereby adaptively adjust the energy threshold $\mathcal{T}$ under heterogeneous clients; **Challenge C2:** How to preserve information critical for model training while suppressing sensitive information leakage; and **Challenge C3:** How to effectively aggregate the SVD truncated client updates and improve global model utility under class imbalance? Our design aims to address these three challenges.

B. Overview of SVDefense

Fig. 4 gives an overview of the proposed *SVDefense*. We also provide a detailed description of our approach in Alg. 1. The workflow is as follows. ① Each client receives the global model and trains its local model (lines 4-6). ② Each client computes channel-wise weights based on its gradient magnitude information (line 17). ③ Each client applies the channel-wise weights to perform SVD on the gradients, measures the entropy of the squared singular value distribution, and derives the self-adaptive energy threshold (lines 18-20). ④ The factorized gradients are then truncated according to the calculated energy threshold for each client (line 21). ⑤ Each client transmits channel-wise weights, truncated gradients, and entropy value to the server (line 9). ⑥ The server reconstructs the local gradients with the truncated gradients (line 26) and calculates layer-wise aggregation weights based on the entropy value (line 27) to update the global model (line 13).

C. Self-Adaptive Energy Threshold

Our analysis in §V-A indicates that the clients with higher degrees of class imbalance require stronger protection against GIAs by lowering the energy threshold. To address **Challenge C1** and quantify the degree of class imbalance for adapting each client's energy threshold, we follow the experimental protocol in §V-A and apply SVD to decompose the gradients obtained at the end of each training epoch for each setting of ρ. Fig. 5 shows the entropy of the squared singular value distribution for a linear layer of the target model versus the class balance ratio ρ. We can see that the entropy value increases with the class balance ratio, indicating that the singular value distribution can serve as an effective indicator of the degree of class imbalance. This observation motivates

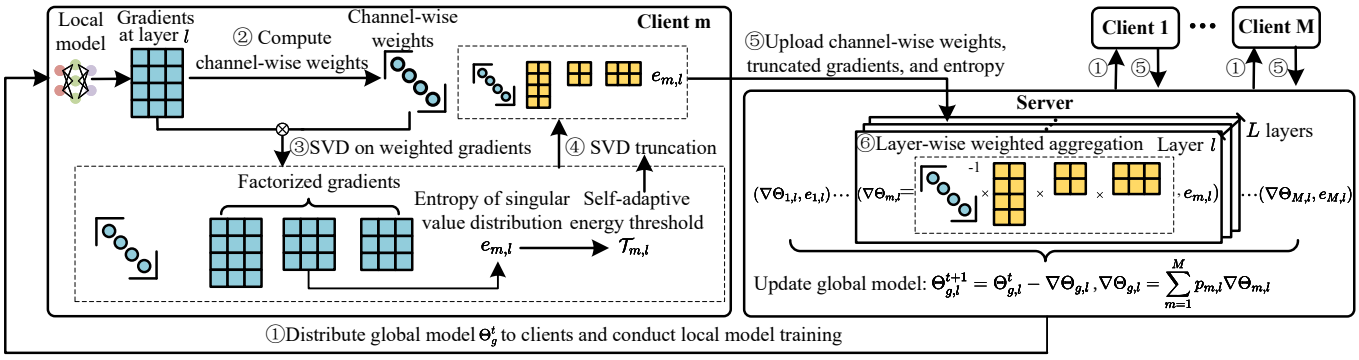


Fig. 4: Overview of the proposed *SVDefense*. On the server side, $\nabla\Theta_{m,l}$ denotes client m's reconstructed gradients at layer l, $\nabla\Theta_{g,l}$ is the global gradients at layer l, and $p_{m,l}$ represents the layer-wise aggregation weight.

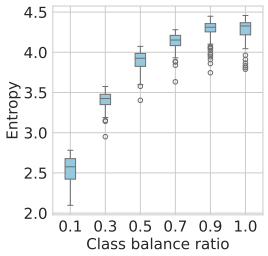


Fig. 5: Entropy of singular value distribution vs. class balance ratio.

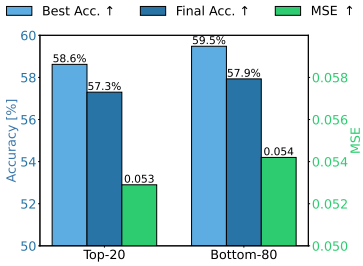


Fig. 6: Comparison of different gradient perturbation-based defense strategies under GIAs.

us to adapt the energy threshold $\mathcal{T}_{m,l}$ for layer l of client m's local model based on the entropy $e_{m,l}$ of the normalized squared singular values. Specifically, we set

$$\mathcal{T}_{m,l} = 1 - \exp(-\beta e_{m,l}), \tag{1}$$

where $e_{m,l} = -\sum_{i=1}^{r_{m,l}} \tilde{\sigma}_{m,l,i}\log(\tilde{\sigma}_{m,l,i})$ with $\tilde{\sigma}_{m,l,i} = \frac{(\sigma_{m,l,i})^2}{\sum_{j=1}^{r_{m,l}}(\sigma_{m,l,j})^2}$ as the i-th normalized squared singular value, $r_{m,l}$ is the rank of the gradients of layer l in client m's model, and the sensitivity parameter β controls the sensitivity of privacy protection. A larger β means a faster decrease in threshold values as the entropy decreases. This exponential function ensures that clients with more imbalanced class distributions (indicated by lower entropy values) receive lower energy thresholds, resulting in stronger privacy protection.

D. Channel-Wise Weighted Approximation

Lowering the energy threshold $\mathcal{T}$ in SVD truncation not only suppresses sensitive information that can be exploited for data reconstruction, but it also reduces useful information for training the model. Traditional SVD treats all the elements of the matrix uniformly, which does not align with our objective of addressing **Challenge C2** to preserve more information critical for model training while suppressing sensitive information leakage. Recent studies [5], [25], [26] suggest that larger gradients, which capture the primary direction of model updates, contain more critical information for classification, while smaller gradients often carry redundant information. Additionally, as proven in [25], the effectiveness of GIAs measured by the data reconstruction error is bounded by the overall gradient error, regardless of whether it originates from large or small gradients. Inspired by these insights, our design preserves larger gradients while applying stronger perturbations to smaller gradients during SVD truncation, aiming to improve both defense performance and model utility.

We conduct a toy experiment on CIFAR-10 using the IG attack to illustrate the effectiveness of the aforementioned strategy. We consider two perturbation-based defenses: 1) *top-20* that applies Laplace noise (scale = 0.03) to the top 20% largest gradients and 2) *bottom-80* that applies Laplace noise (scale = 0.03) to the bottom 80% smallest gradients. As shown in Fig. 6, the bottom-80 defense strategy achieves higher best and final classification accuracies and higher MSE between the ground truth and reconstructed inputs, indicating a better classification and defense performance at the same time.

To this end, we propose a weighted truncated SVD approach that incorporates the gradient magnitude information. Intuitively, the weighted optimization objective is formulated as: $\arg\min_{\hat{\mathbf{W}}_l} \sum_{c=1}^{C}\sum_{i=1}^{N_{l,c}} \omega_{l,c,i}(\mathbf{W}_{l,c,i} - \hat{\mathbf{W}}_{l,c,i})^2$, where $\mathbf{W}_l$ and $\hat{\mathbf{W}}_l$ denote respectively the ground-truth and approximated gradients of the l-th layer in the client model, $\omega_{l,c,i}$ is the weight for the i-th element in the c-th output channel of the gradients, C is the number of total output channels, and $N_{l,c}$ is the number of elements in the c-th channel of $\mathbf{W}_l$. For simplicity, we use squared gradients as weights, where $\omega_{l,c,i} = (\mathbf{W}_{l,c,i})^2$. In §VI-E, we consider an alternative method of using absolute gradient values as weights and demonstrate that squared gradients as weights achieves better accuracy and defense performance. However, such element-wise weighted low-rank approximation is a nonlinear optimization problem, which does not have a closed-form solution [64]. To make the problem tractable, we employ a diagonal weight matrix. Specifically, we sum the weights along the output channel axis and define the channel-wise weight matrix as $\mathbf{I}_l = \text{diag}(\sqrt{\omega_{l,1}}, \ldots, \sqrt{\omega_{l,C}})$, where $\omega_{l,c} = \sum_{i=1}^{N_{l,c}} \omega_{l,c,i}$.

Algorithm 1 Federated learning with *SVDefense*

Input: Initial global model $\mathbf{\Theta}_g^0$, local datasets $\{D_m\}_{m=1}^M$, number of rounds T, sensitivity parameter β
Output: Final global model $\mathbf{\Theta}_g^T$
for each round $t = 0, 1, \ldots, T-1$ **do**
  for each selected client m in parallel **do**
    Initialize local model: $\mathbf{\Theta}_m \leftarrow \mathbf{\Theta}_g^t$
    Train local model on D_m to get gradients $\nabla\mathbf{\Theta}_m$
    for each layer l of client model **do**
      $\mathbf{P}_{m,l} \leftarrow \text{DEFEND\_GRAD}(\nabla\mathbf{\Theta}_{m,l}, \beta)$
      Send $\mathbf{P}_{m,l}$ to server
    end for
  end for
  for each layer l of global model in server **do**
    $\mathbf{\Theta}_{g,l}^{t+1} \leftarrow \text{AGGREGATE}(\mathbf{\Theta}_{g,l}^t, \{\mathbf{P}_{m,l}\}_{m=1}^M)$
  end for
end for
function DEFEND_GRAD($\nabla\mathbf{\Theta}_{m,l}, \beta$)
  Compute channel-wise weight matrix $\mathbf{I}_l$ from $\nabla\mathbf{\Theta}_{m,l}$
  $\{\mathbf{U}_l, \mathbf{\Sigma}_l, {\mathbf{V}_l}^\top\} \leftarrow \text{SVD}(\mathbf{I}_l\nabla\mathbf{\Theta}_{m,l})$
  Compute entropy $e_{m,l}$ from $\mathbf{\Sigma}_l$
  Compute threshold $\mathcal{T}_{m,l} \leftarrow 1 - \exp(-\beta e_{m,l})$
  Truncate singular values to get $\{\mathbf{U}_l^*, \mathbf{\Sigma}_l^*, {\mathbf{V}_l^*}^\top\}$
  return $\mathbf{P}_{m,l} \leftarrow \{\mathbf{I}_l, \mathbf{U}_l^*, \mathbf{\Sigma}_l^*, {\mathbf{V}_l^*}^\top, e_{m,l}\}$
end function
function AGGREGATE($\mathbf{\Theta}_{g,l}^t, \{\mathbf{P}_{m,l}\}_{m=1}^M$)
  for each client m **do**
    $\nabla\mathbf{\Theta}_{m,l} \leftarrow (\mathbf{I}_{m,l})^{-1}\mathbf{U}_{m,l}^*\mathbf{\Sigma}_{m,l}^*{\mathbf{V}_{m,l}^*}^\top$
    Compute layer-wise aggregation weight $p_{m,l}$ based on $e_{m,l}$
  end for
  $\nabla\mathbf{\Theta}_{g,l} \leftarrow \sum_{m=1}^M p_{m,l}\nabla\mathbf{\Theta}_{m,l}$
  return $\mathbf{\Theta}_{g,l}^t - \nabla\mathbf{\Theta}_{g,l}$
end function

The optimization problem then transforms to:

$$\arg\min_{\hat{\mathbf{W}}_l} \|\mathbf{I}_l\mathbf{W}_l - \mathbf{I}_l\hat{\mathbf{W}}_l\|_F, \tag{2}$$

where $\|\cdot\|_F$ is the Frobenius norm. The optimization tends to reduce the error in high-weight regions due to $\mathbf{I}_l$.

The optimization process is as follows. For a rank k derived from the energy threshold $\mathcal{T}$, we obtain the optimal $\hat{\mathbf{W}}_l^*$ by applying truncated SVD to $\mathbf{I}_l\mathbf{W}_l$, yielding $\mathbf{U}_l^*$, $\mathbf{\Sigma}_l^*$, and ${\mathbf{V}_l^*}^\top$ that satisfy Eq. 2. The optimal solution is denoted by $\hat{\mathbf{W}}_l^* = (\mathbf{I}_l)^{-1}\mathbf{U}_l^*\mathbf{\Sigma}_l^*{\mathbf{V}_l^*}^\top$, which maintains the same rank k since $(\mathbf{I}_l)^{-1}$ is a diagonal matrix. This weighted optimization strategy effectively improves both the model accuracy and the privacy protection by selectively preserving larger gradients while applying stronger perturbations to smaller gradients.

We theoretically demonstrate that our Channel-Wise Weighted Approximation enhances the defense performance of truncated SVD. A *passive attacker* is defined as an adversary who attempts to reconstruct private input data while honestly adhering to the FL protocol [25]. As discussed in §III, the honest-but-curious adversary we consider belongs to the class of passive attackers. Specifically, we have the following definition and theorem:

Definition 1. *A passive attack $\mathcal{A}$ is an (ε, δ)-passive attack, if it satisfies*:

$$\mathbb{P}(\mathbb{E}(\mathcal{D}_{\mathcal{A}}(\nabla\mathbf{\Theta}, \nabla\mathbf{\Theta}^*)) \leq \varepsilon) \geq 1 - \delta. \tag{3}$$

where $\mathbb{P}$ represents the probability, $\mathbb{E}$ represents the expectation, and $\mathcal{D}_{\mathcal{A}}$ is the distance estimated under $\mathcal{A}$.

Theorem 1. *For any (ε, δ)-passive attack $\mathcal{A}$, under the presence of truncated SVD, it will degenerate to $(\varepsilon + \sqrt{\gamma_1}\|\nabla\mathbf{\Theta}\|_F, \delta)$, where $\gamma_1 = 1 - \mathcal{T}$. Under the presence of truncated SVD with Channel-Wise Weighted Approximation, it will degenerate to $(\varepsilon + \sqrt{\gamma_2}\|\nabla\mathbf{\Theta}\|_F, \delta)$, where $\gamma_2 = (\frac{\sigma_{max}(\mathbf{I})}{\sigma_{min}(\mathbf{I})})^2(1 - \mathcal{T})$, $\sigma_{max}(\cdot)$ and $\sigma_{min}(\cdot)$ mean the maximum and minimum singular value of the input matrix.*

Since $\frac{\sigma_{max}(\mathbf{I})}{\sigma_{min}(\mathbf{I})} \geq 1$, we have $\gamma_2 \geq \gamma_1$. This indicates that truncated SVD with Channel-Wise Weighted Approximation provides stronger protection than the original truncated SVD. The detailed proof can be found in Appendix D. By our proof in Appendix D and experimental results in §VI, Channel-Wise Weighted Approximation improves both the defense performance and model accuracy.

E. Layer-Wise Weighted Aggregation

Under non-IID local data distributions, the aggregated local optima deviates from the global optimum, leading to degraded global accuracy [24], [61], [65], [66]. To address this, weighted aggregation strategies have been proposed [66], [67], where local clients contribute differently to the global model updates based on their heterogeneous data distributions. However, no existing work investigates appropriate weights for aggregating SVD truncated weights and addresses **Challenge C3**.

This work assumes the global data distribution to be class-balanced, which aligns with standard FL benchmarks [28]–[31]. Under this assumption, clients with more balanced local data distributions will have local optima closer to the global optimum, which should be assigned higher weights during aggregation [61]. Our analysis in §V-C reveals that the entropy of the squared singular value distribution can serve as an indicator of the degree of class imbalance. Therefore, we assign layer-wise aggregation weights to different clients as:

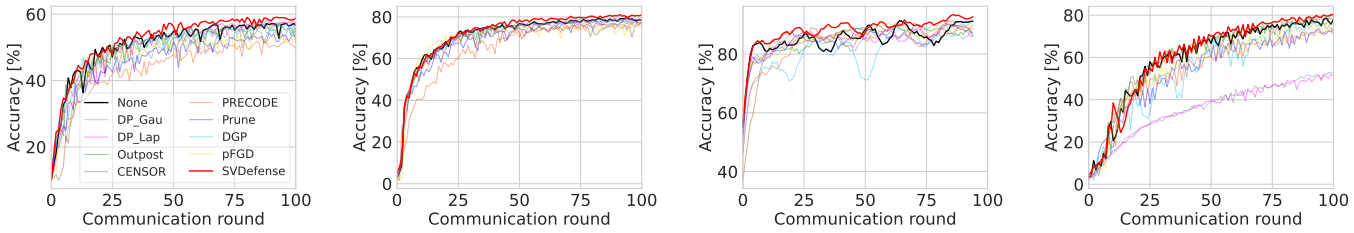
$$p_{m,l} = \frac{e_{m,l} \times N_m}{\sum_{i=1}^{M} e_{i,l} \times N_i}, \tag{4}$$

where $p_{m,l}$ represents the aggregation weight. The server then aggregates the received client updates to update the global model $\mathbf{\Theta}_{g,l}^{t+1} = \mathbf{\Theta}_{g,l}^t - \nabla\mathbf{\Theta}_{g,l}$, where $\mathbf{\Theta}_{g,l}^t$ is the weights of layer l of the global model, $\nabla\mathbf{\Theta}_{g,l} = \sum_{m=1}^M p_{m,l}\nabla\mathbf{\Theta}_{m,l}$, and $\nabla\mathbf{\Theta}_{m,l}$ denotes the layer l of client m's gradients.

VI. EVALUATION

A. Experimental Setup.

Datasets and Models: We evaluate the effectiveness of our defense using the following datasets and applications.



(a) IC-CIFAR10 (b) IC-EMNIST (c) HAR (d) KWS

Fig. 7: Comparison of classification accuracy across different defense methods.

TABLE II: Number of Total and Participating Clients per Round.

Dataset	# total client	# client per round	# class	non-IID
EMNIST [28]	196	19	64	✓
CIFAR-10 [29]	100	10	10	✓
HAR [30]	30	2	6	✓
KWS [31]	489	48	36	✓

- **Image Classification (IC)**. We consider two image classification datasets: the extended MNIST (EMNIST) [28] with 28×28 grey-scale handwritten letters collected from different writers with imbalanced class distributions and CIFAR-10 [29] with 32×32 color images. We randomly select 196 writers from the EMNIST dataset and assign each writer's data samples to a client. For CIFAR-10, we create 100 class-imbalanced clients by splitting the data samples of each class based on proportions sampled from a Dirichlet distribution with $\alpha = 0.5$. We denote the two applications as IC-EMNIST and IC-CIFAR10.
- **Human Activity Recognition (HAR)**. HAR identifies daily activities like walking or sitting, leveraging sensor signals such as Inertial Measurement Unit (IMU) data. We adopt a public IMU dataset [30] consisting of six daily activities collected from 30 class-imbalanced participants. We assign each participant's data samples to a client.
- **Keyword Spotting (KWS)**. KWS captures specific commands using device microphones to enable voice-based human-computer interaction. We use the Google Speech Commands Dataset [31], which contains 105,829 one-second utterances of 35 keywords collected from different speakers. Each voice sample is processed into an 81×40 Mel-Frequency Cepstral Coefficients (MFCC) tensor. We randomly select 489 speakers that are class-imbalanced and assign each speaker's data samples to a client.

For IC-CIFAR10, IC-EMNIST, and KWS, we adopt the ResNet-18 architecture. For HAR, we use the 1D ConvNet [68]. We also consider the larger-scale Vision Transformer (ViT) [69] trained on ImageNet [70] in §VI-C. The total number of clients and the number of participating clients per communication round are summarized in Table II. Note that the data splits in all the applications are class-imbalanced.

Attack and Defense Baselines: We implement the widely considered IG attack as our primary evaluation attack due to its broad applicability across different model architectures and data types. §VI-C also evaluates our defense using a recent strong GAN-based attack, ROG [7], on high-resolution image data. We consider eight representative defense baselines, including perturbation-based (DP [5], Outpost [20], CENSOR [21], and PRECODE [23]), pruning-based (Prune [5], Soteria [26], and DGP [25]), and compression-based (pFGD [44]) defenses. We implement adaptive attacks as defined in §III and employ the same attack settings as in §IV for all the defenses except for noise injection-based (i.e., DP and Outpost) and compression-based (i.e., pFGD) defenses. This is because noise injection-based defenses are theoretically proven effective against practical adaptive attacks in Appendix A, while pFGD compresses the entire gradient space by discarding low-frequency components, similar to truncated SVD. However, §VI-C compares our defense with noise injection-based and compression-based defenses under a less practical adaptive adversary to demonstrate our defense's robustness even under extreme threat conditions. We implement two DP baseline variants, DP-Gaussian and DP-Laplace, which employ Gaussian and Laplacian noises, respectively.

For IC-CIFAR10, IC-EMNIST, and KWS, we set the noise scale to 0.03. For HAR, we set the noise scale to 0.5. We follow [44] and set the pruning rate to 0.01 for pFGD. We randomly sample 128 input samples from each dataset to launch attacks. We set the local training batch size, number of epochs, and the number of steps all to be 1 and perform the attack in the first communication round. This setting is ideal for the adversary and most challenging for the defender, which is commonly adopted by existing defenses [7], [9], [22], [25]. By default, we set β to be 0.3 in *SVDefense* as our experiments in §VI-G show that this value achieves a favorable trade-off among defense, accuracy, and communication cost.

Evaluation Metrics: We use accuracy to evaluate classification performance. To evaluate defense effectiveness, we use metrics including the MSE, PSNR, structural similarity index measure (SSIM) [71], and LPIPS. Note that we use all four metrics for image classification and only use MSE for the remaining applications, since PSNR, SSIM, and LPIPS are specific to image data. Higher MSE and LPIPS and lower

TABLE III: Comparison of Defense Effectiveness Across Different Defense Methods.

Dataset	Metric	None	DP-Gau	DP-Lap	Outpost	CENSOR	PRECODE	Prune	Soteria	DGP	pFGD	*SVDefense*
CIFAR-10	MSE (↑)	0.0056	0.0546	0.0514	0.0177	0.0141	0.0000	0.0136	0.0050	0.0108	0.0584	**0.0619**
	PSNR (↓)	23.8755	12.8280	13.1080	18.0419	19.2682	inf	19.3477	24.0950	21.1468	12.9291	**12.5278**
	SSIM (↓)	0.8411	0.2478	0.2718	0.3780	0.6908	0.9998	0.6915	0.8469	0.7579	0.2122	**0.1375**
	LPIPS (↑)	0.1894	0.5830	0.5754	0.6347	0.2747	0.0001	0.3223	0.1780	0.2631	0.5821	**0.5866**
EMNIST	MSE (↑)	0.0003	0.0633	0.0575	0.0057	0.0017	0.0000	0.0006	0.0003	0.0006	0.0968	**0.1429**
	PSNR (↓)	36.8783	12.1025	12.5235	23.2789	40.8229	inf	35.7690	37.0652	33.6131	10.3058	**8.5792**
	SSIM (↓)	0.9516	0.5376	0.5522	0.8178	0.9833	0.9968	0.9550	0.9553	0.9264	0.3084	**0.2025**
	LPIPS (↑)	0.0111	0.5453	0.5310	0.1223	0.0098	0.0003	0.0135	0.0103	0.0176	0.6494	**0.6651**
HAR	MSE (↑)	0.1953	0.2198	0.2907	0.2627	0.2034	0.000	0.2930	0.2493	0.2247	0.3561	**0.4156**
KWS	MSE (↑)	0.0978	0.1286	0.1542	0.1129	0.1194	0.000	0.1638	0.1385	0.1068	0.1634	**0.1676**

PSNR and SSIM indicate more effective defense performance. To evaluate system overhead, we define the *normalized on-device latency* and *communication cost reduction*. The normalized on-device latency is defined as the ratio between the total on-device local training time with and without defense. A value of 1.0 for this metric indicates no additional computational overhead compared with the baseline without defense. The communication cost reduction measures the percentage decrease in total communication latency (including both uploading and downloading latency) achieved by the defense relative to the baseline without defense.

Implementation Details: To validate the practicality of *SVDefense*, we implement a real-world FL testbed as shown in Appendix E. The testbed contains heterogeneous embedded platforms, including two NVIDIA Jetson TX2, two NVIDIA Jetson Nano, and six Raspberry Pi 4, as client devices. The server is equipped with an AMD EPYC 7543@ 3.7GHz, 256G RAM, and 4 RTX A5000 GPUs. We use TL-SG116 to connect the server and client devices. Since the number of clients participating in each communication round may exceed the number of available devices, we randomly assign a device for each client to train its local model. Each device trains one client model at a time due to resource constraints.

B. Accuracy

Fig. 7 presents the classification accuracy across communication rounds in the presence of different defenses and in the absence of defense, indicated by "None". From the results, we can see that *SVDefense* performs better than the undefended baseline "None" and the other defense methods across all the applications. As expected, DP-based defenses degrade the model utility. Specifically, DP-Laplace achieves a final accuracy of 51.0% and DP-Gaussian achieves 53.5% in KWS, while the "None" baseline achieves a final accuracy of 78.0% and *SVDefense* achieves 79.9%, respectively. Outpost achieves an accuracy similar to "None" by adaptively adjusting perturbations to preserve more information during the FL training. However, it has degraded defense performance, as shown in §VI-C. CENSOR maintains an accuracy close to "None" since its defense operations are only activated during the initial few training epochs [21]. When the defense is activated for all the training epochs, CENSOR reduces the final accuracy by 14%, 8%, 18%, and 9% for IC-CIFAR10, IC-EMNIST, HAR, and KWS, respectively, compared with "None". PRECODE has degraded accuracy in all the applications because it needs to sample a random vector in each training step, leading to extra noise. Note that Soteria is omitted from the accuracy comparison as its layer-wise defense operations become computationally intractable when applied in each communication round [20]. Prune achieves a lower accuracy due to its aggressive gradient pruning that discards potentially important update information. Although DGP performs well in IC applications, its accuracy fluctuates in HAR and KWS, potentially due to the per-step pruning of large gradients that may contain useful information. pFGD has degraded accuracy because it applies the Discrete Cosine Transform to the gradients and directly sets the coefficients of the low-frequency components to zero, which may discard critical gradient direction information.

TABLE IV: Comparison of Defense Effectiveness Across Different Defense Methods Under Adaptive LTI attack [9].

Metric	DP-Gau	DP-Lap	Outpost	pFGD	*SVDefense*
MSE (↑)	0.0292	0.0315	0.0220	0.0197	**0.0469**
PSNR (↓)	15.8955	15.5623	17.1465	17.6388	**14.3392**
SSIM (↓)	0.2547	0.2356	0.3369	0.3672	**0.1509**
LPIPS (↑)	0.5744	0.5834	0.5487	0.5362	**0.6521**

C. Defense Performance

Table III presents the defense performance of different methods, with the best and second-best results highlighted in bold and underlined, respectively. Note that the PSNR is computed by dividing the MSE. Thus, "inf" values in the table indicate near-zero MSE values. We can see from the table that *SVDefense* achieves the best defense performance in all the applications, compared with all the baselines. Although DP provides theoretical privacy guarantees, it significantly impacts the model utility, as shown in Fig. 7. In comparison, *SVDefense* achieves strong defense performance without compromising classification accuracy, due to the effectiveness of the Channel-Wise Weighted Approximation and Layer-Wise Weighted Aggregation mechanisms in our design. Examples of the reconstructed images under different defenses are provided in Appendix E.

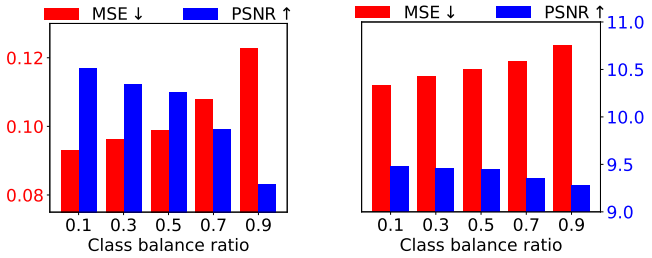
Adaptive Adversary against *SVDefense*. As analyzed in §IV, *SVDefense* is invulnerable to adaptive attack operations like identifying modified gradient components or recovering random variables used by the defender. Nevertheless, to further

TABLE V: Comparison of Defense Effectiveness Across Different Defense Methods on High-resolution ImageNet with LeNet [72].

Metric	None	DP-Gau	DP-Lap	Outpost	CENSOR	PRECODE	Prune	DGP	*p*FGD	*SVDefense*
MSE (↑)	0.0220	0.0381	0.0369	0.0273	0.0289	0.0029	0.0265	0.0247	0.0564	**0.0904**
PSNR (↓)	17.2417	14.6213	14.6889	16.3300	16.3367	28.6856	16.4031	16.8004	13.4950	**10.9315**
SSIM (↓)	0.5090	0.2613	0.2446	0.4253	0.4162	0.9287	0.4280	0.4952	0.4490	**0.1128**
LPIPS (↑)	0.4313	0.6175	0.6242	0.4908	0.5163	0.0236	0.5053	0.4498	0.5343	**0.7004**

TABLE VI: Comparison of Defense Effectiveness Across Different Defense Methods on High-resolution ImageNet with ViT [69].

Metric	None	DP-Gau	DP-Lap	Outpost	CENSOR	PRECODE	Prune	DGP	*p*FGD	*SVDefense*
MSE (↑)	0.0817	0.0796	0.0794	0.0848	0.0874	0.0834	0.1109	0.0859	0.0985	**0.1287**
PSNR (↓)	11.4922	11.6943	11.7168	11.3691	11.1634	11.4756	9.9646	11.2950	10.5598	**9.2805**
SSIM (↓)	0.4852	0.2795	0.2704	0.1925	0.4342	0.4847	0.3194	0.4586	0.1821	**0.0494**
LPIPS (↑)	0.3528	0.6552	0.6739	0.6939	0.3793	0.3536	0.5627	0.3834	0.6631	**0.7473**



(a) SVD (b) SVD_S

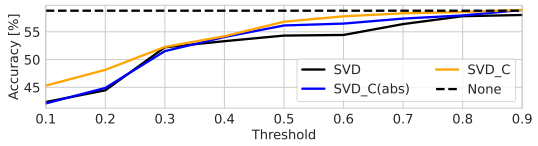
Fig. 8: Impact of Self-Adaptive Energy Threshold on defense performance under class imbalance. Lower MSE and higher PSNR indicate stronger attack effectiveness. $\mathcal{T} = 0.8$ is fixed for "SVD"; β for "SVD_S" is chosen to match the defense performance of "SVD" at class balance ratio of 0.9.

evaluate *SVDefense*'s performance in extreme situations, we simulate a powerful adaptive attacker based on the LTI attack [9]. Specifically, we assume that the attacker has obtained a surrogate training dataset that follows the same distribution as the client's local data. The attacker can then train a surrogate local model on this dataset, use the model to generate input-gradient pairs, and apply *SVDefense* to process these gradients. The attacker then trains a neural network to learn the mapping from the approximated gradients to the input samples. We can follow a similar procedure to train the gradient inversion models for the DP, Outpost, and *p*FGD baselines. During the attack phase, the gradient inversion model is applied to a victim client's defended gradients in order to attempt data reconstruction. Table IV presents the defense performance on CIFAR-10. From the results, we can observe that *SVDefense* achieves the best defense performance compared with the DP baselines (i.e., DP-Gaussian and DP-Laplace both with a noise scale of 0.1), Outpost, and *p*FGD. However, the assumption that the adversary can obtain the distribution of a client's real data can be impractically strong and work against the premise of GIAs. This is because in real-world FL scenarios, the private data distribution is precisely what a client aims to protect. If the attacker already had access to similarly distributed data, there would be little motivation to perform GIAs in the first place.

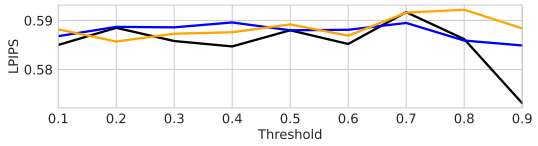
***SVDefense* Effectiveness for High-Resolution Images and on Large-Scale Model.** We evaluate our defense on ImageNet, a large-scale dataset consisting of 1,000 classes of high-resolution (224×224) color images. We train a LeNet [72] on ImageNet and implement a recent strong GAN-based attack, ROG. We follow the same evaluation setup as described in §VI-A except that we set the training batch size to 16 and the noise scale of the DP baselines to 0.1. As shown in Table V, *SVDefense* achieves superior defense performance compared with all the baselines. Note that Soteria is excluded from this comparison due to its computational infeasibility for high-dimensional data. We then evaluate *SVDefense*'s performance on the large-scale ViT model using the IG attack. As shown in Table VI, *SVDefense* still outperforms all the baselines.

D. Impact of Self-Adaptive Energy Threshold under Class Imbalance

This section evaluates the impact of the Self-Adaptive Energy Threshold on the defense performance of truncated SVD under class imbalance. We follow the method described in §V-A to simulate varying degrees of class imbalance on CIFAR-10. Fig. 8 illustrates the defense performance of truncated SVD with and without Self-Adaptive Energy Threshold, denoted by "SVD" and "SVD_S", respectively, under varying degrees of class imbalance. For a fair comparison, we fix the energy threshold $\mathcal{T}$ to be 0.8 for "SVD" and select a β for "SVD_S" such that its defense performance matches that of the "SVD" when the class balance ratio is 0.9. We can observe that the defense performance of "SVD" deteriorates as the class balance ratio decreases. In comparison, "SVD_S" effectively adapts to varying degrees of class imbalance and maintains more stable defense performance. The defense performance for "SVD_S" is always better than that of "SVD". This is because class-imbalanced inputs produce gradients with a more skewed distribution of squared singular values, leading the Self-Adaptive Energy Threshold to adaptively

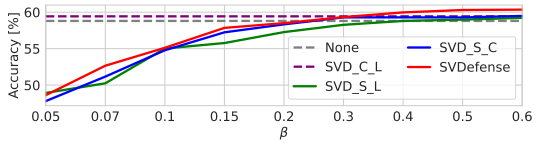


(a) Classification accuracy vs. energy threshold.

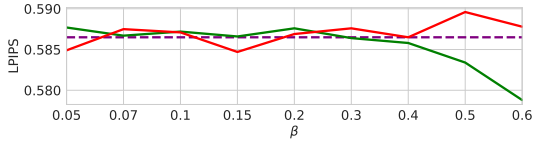


(b) Defense performance vs. energy threshold.

Fig. 9: Impact of varying energy threshold on accuracy and defense performance for SVD_C.



(a) Classification accuracy vs. β.



(b) Defense performance vs. β.

Fig. 10: Impact of varying β on accuracy and defense performance for *SVDefense*.

derive a lower energy threshold that provides stronger protection.

E. Impact of Channel-Wise Weighted Approximation

This section compares the original truncated SVD ("SVD") and two variants of truncated SVD with Channel-Wise Weighted Approximation, i.e., "SVD_C" that uses squared gradients as weights and "SVD_C(abs)" that uses absolute values of gradients as weights, on CIFAR-10. Fig. 9a shows the classification accuracy when varying the energy threshold $\mathcal{T}$ from 0.1 to 0.9 with a step size of 0.1. We can observe that the accuracy increases with the threshold, and that "SVD_C" outperforms both "SVD_C(abs)" and "SVD". Fig. 9b shows the defense performance. We can see that, when $\mathcal{T} < 0.7$, the LPIPS value fluctuates and all the methods perform similarly. When $\mathcal{T}$ is greater than 0.7, "SVD_C" outperforms both "SVD_C(abs)" and "SVD". These results demonstrate the effectiveness of the Channel-Wise Weighted Approximation module in enhancing both the accuracy and defense performance. They also show that, compared with absolute values of gradients as weights, square gradients as weights better emphasize larger singular components, which contributes to improved classification and defense performance.

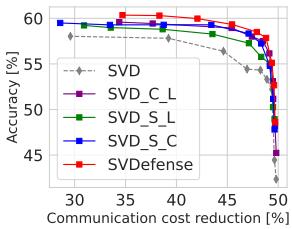


Fig. 11: Classification accuracy vs. communication cost reduction.

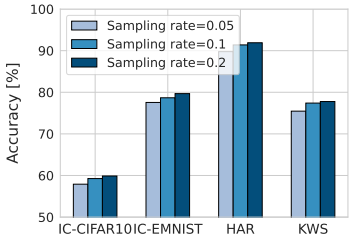


Fig. 12: Impact of client sampling rate on *SVDefense* classification accuracy.

F. Ablation Study

We evaluate three key components of *SVDefense*, namely Self-Adaptive Energy Threshold, Channel-Wise Weighted Approximation, and Layer-Wise Weighted Aggregation, on CIFAR-10. We denote different variants as "SVD_{}_{}", where each placeholder in brackets contains the component's initial letter if included. For example, "SVD_S_C" represents *SVDefense* without the Layer-Wise Weighted Aggregation. "SVDefense" denotes our full proposed method.

Fig. 10 presents the accuracy and defense performance of different variants. For the variant without the Self-Adaptive Energy Threshold, we set a fixed energy threshold $\mathcal{T}$ of 0.8, which achieves a balanced accuracy and defense performance based on empirical results. For variants with the Self-Adaptive Energy Threshold, we vary the sensitivity parameter β within $\{0.05, 0.07, 0.1, 0.15, 0.2, 0.3, 0.4, 0.5, 0.6\}$. Since the undefended baseline "None" and "SVD_C_L" are not affected by the variation of β, their performance is represented by horizontal lines in Fig. 10. The Layer-Wise Weighted Aggregation aims to improve the global model's utility and does not affect the defense performance. Therefore, in Fig. 10b, we only present "SVDefense" (equivalent to "SVD_S_C"), "SVD_S_L" (equivalent to "SVD_S"), and "SVD_C_L" (equivalent to "SVD_C"). From Fig. 10a, the complete version of *SVDefense* achieves the best accuracy compared with all the other variants. From Fig. 10b, the defense performance of all the methods is similar when $\beta < 0.3$. When $\beta \geq 0.3$, *SVDefense* outperforms the other variants. In conclusion, *SVDefense* that combines all three components achieves the best accuracy and defense performance when setting β at appropriate values. This is because the Self-Adaptive Energy Threshold and Channel-Wise Weighted Approximation effectively suppress sensitive information leakage under class imbalance while preserving more information critical for the model training. The Layer-Wise Weighted Aggregation further enhances the global model accuracy.

Fig. 11 shows the classification accuracy versus the communication cost reduction of the different variants. For the variants without Self-Adaptive Energy Threshold, we vary $\mathcal{T}$ from 0.1 to 0.9 with a step size of 0.1. For the variants with Self-Adaptive Energy Threshold, we vary β to be $\{0.05, 0.07, 0.1, 0.15, 0.2, 0.3, 0.4, 0.5, 0.6\}$. We can see

that *SVDefense* achieves better accuracy and communication efficiency, compared with all the other variants. This validates the effectiveness of the three key components in balancing accuracy and communication efficiency.

G. Sensitivity Analysis

This section evaluates the impact of varying different parameters on *SVDefense*'s performance using the CIFAR-10 dataset.

Sensitivity Parameter β. First, we analyze the impact of varying the sensitivity parameter β on *SVDefense*'s performance. We vary the value of β to be $\{0.05, 0.07, 0.1, 0.15, 0.2, 0.3, 0.4, 0.5, 0.6\}$. The red line in Fig. 10a shows the classification accuracy of *SVDefense* across different β values. The result shows that the accuracy increases with β. This is because for the same entropy value, a larger value of β leads to a higher energy threshold $\mathcal{T}$, retaining more gradient information. This improvement in accuracy plateaus around $\beta \approx 0.3$, suggesting that additional gradient information beyond this point contributes minimally to the classification performance. The red line in Fig. 10b illustrates the defense performance of *SVDefense* across different β values. The performance fluctuates because *SVDefense* tends to retain larger gradients for higher accuracy, which can also be primarily exploited for data reconstruction [25]. However, by perturbing the gradients with channel-wise weights and aggregating the local models with layer-wise weights, *SVDefense* can effectively improve both the accuracy and defense performance. The red line in Fig. 11 illustrates the trade-off between the accuracy and communication cost reduction of *SVDefense* across different β values. The result shows that the accuracy decreases as the communication cost reduction rate increases. *SVDefense* achieves a favorable trade-off when $\beta \in \{0.3,0.2,0.15\}$. In summary, *SVDefense* achieves an optimal balance at $\beta \approx 0.3$, where it maintains strong classification performance, robust privacy protection, and high communication efficiency. Similar sensitivity analysis can be applied to determine the optimal value of β for other applications, considering their respective requirements for model accuracy, privacy protection, and communication cost.

Number of Participating Clients. We analyze the impact of the per-round client sampling rate on the classification accuracy of *SVDefense*. We vary the client sampling rate $f \in \{0.05,0.1,0.2\}$, where f represents the ratio of sampled clients in each communication round. As shown in Fig. 12, increasing the client sampling rate yields modest accuracy improvements. For example, for IC-CIFAR10, the accuracy increases by 1.97% when f increases from 0.05 to 0.2. However, this marginal performance gain comes with 4x higher client-server communication costs. The results suggest that moderate client sampling rates can achieve a good model performance while maintaining communication efficiency.

H. System Overhead

Normalized On-Device Latency. Fig. 13 compares the normalized on-device latency of different defenses on three embedded platforms. The absolute on-device latency of the undefended baseline "None" over one epoch can be found in Table VII. First, on the Raspberry Pi 4, all the defenses incur minimal additional computation overhead. This is because the CPU-based model training on the Raspberry Pi 4 is time-intensive, making the defense operation time relatively insignificant. Second, the DP-based defenses show a 2-3× slowdown compared with "None" on both the Orin Nano and TX2. This is because random sampling operations are costly on resource-constrained hardware. Third, pruning-based defenses perform notably slower on Orin Nano due to their computation-intensive matrix operations like sorting. Fourth, CENSOR and pFGD have negligible extra computational overhead but struggle to achieve a good balance between model utility and privacy protection. Lastly, *SVDefense* incurs limited additional on-device computational overhead on all the platforms across all the applications except IC-EMNIST. This is because *SVDefense* is only applied once at the end of each communication round, reducing the overall computational cost. The higher extra computational cost for EMNIST is because the dataset is relatively simple. Thus, the total local training time becomes comparable to the SVD operation time. Note that Soteria is omitted from the latency comparison as its layer-wise defense operations applied in each training step become computationally infeasible on these embedded platforms [20].

TABLE VII: Absolute On-device Latency (Seconds) of "None".

Device	IC-CIFAR10	IC-EMNIST	HAR	KWS
RPi	130.1	31.3	2.1	62.5
Orin Nano	3.6	0.8	0.2	1.7
TX2	4.5	1.1	0.3	2.6

TABLE VIII: Communication Cost Reduction (%) for *SVDefense*.

Application	IC-CIFAR10	IC-EMNIST	HAR	KWS
Comm. cost reduction (%)	42.0	30.3	48.6	23.7

Communication Cost Reduction. *SVDefense* achieves communication cost reduction by decomposing and truncating the model updates into matrices with fewer parameters than the original updates. As shown in Table VIII, *SVDefense* significantly reduces the communication costs in all the applications. Note that the other defense baselines have similar communication cost as "None" because these defenses do not have specific mechanisms for communication time reduction.

VII. Discussion

SVDefense can be potentially extended to deep learning architectures like recurrent neural networks (RNNs), graph neural networks (GNNs), and other emerging model architectures. For example, SVD can be applied to the recurrent layers in RNNs and the message-passing layers in GNNs. Doing so may require developing new SVD strategies that

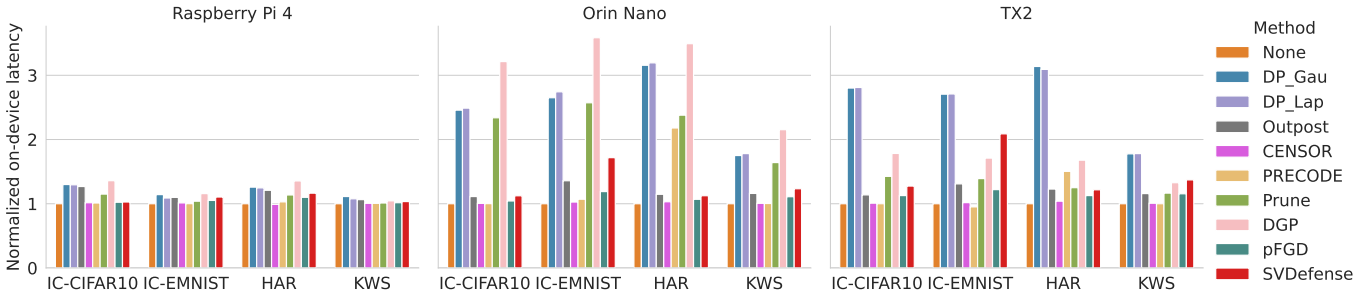


Fig. 13: Comparison of normalized on-device latency across different defense methods on three embedded platforms.

account for the unique characteristics of these architectures. Beyond the applications discussed in this paper, *SVDefense* can be employed in other domains such as natural language processing [73]–[75] and medical image analysis.

VIII. Conclusion

This paper presents *SVDefense*, a novel defense framework based on truncated SVD against adaptive GIAs in FL. Our framework introduces three key innovations: the Self-Adaptive Energy Threshold that adapts to client vulnerability, the Channel-Wise Weighted Approximation to enhance accuracy and defense performance, and the Layer-Wise Weighted Aggregation for effective aggregation under class imbalance. Extensive experiments demonstrate that *SVDefense* outperforms existing defenses in both model accuracy and defense effectiveness. Furthermore, *SVDefense* achieves practical computational cost and much reduced communication cost on a real-world FL testbed.

Acknowledgment

We thank the anonymous reviewers for their valuable comments and suggestions.

References

[1] I. Dayan, H. R. Roth, A. Zhong, A. Harouni, A. Gentili, A. Z. Abidin, A. Liu, A. B. Costa, B. J. Wood, C.-S. Tsai *et al.*, "Federated learning for predicting clinical outcomes in patients with covid-19," *Nature medicine*, vol. 27, no. 10, pp. 1735–1743, 2021.

[2] R. Ye, W. Wang, J. Chai, D. Li, Z. Li, Y. Xu, Y. Du, Y. Wang, and S. Chen, "Openfedllm: Training large language models on decentralized private data via federated learning," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 6137–6147.

[3] Q. Meng, F. Zhou, H. Ren, T. Feng, G. Liu, and Y. Lin, "Improving federated learning face recognition via privacy-agnostic clusters," in *International Conference on Learning Representations*, 2022.

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

[5] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," *Advances in neural information processing systems*, vol. 32, 2019.

[6] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients-how easy is it to break privacy in federated learning?" *Advances in neural information processing systems*, vol. 33, pp. 16 937–16 947, 2020.

[7] K. Yue, R. Jin, C.-W. Wong, D. Baron, and H. Dai, "Gradient obfuscation gives a false sense of security in federated learning," in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 6381–6398.

[8] Z. Li, J. Zhang, L. Liu, and J. Liu, "Auditing privacy defenses in federated learning via generative gradient leakage," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 132–10 142.

[9] R. Wu, X. Chen, C. Guo, and K. Q. Weinberger, "Learning to invert: Simple adaptive attacks for gradient inversion in federated learning," in *Uncertainty in Artificial Intelligence*. PMLR, 2023, pp. 2293–2303.

[10] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1175–1191.

[11] D. Lia and M. Togan, "Privacy-preserving machine learning using federated learning and secure aggregation," in *2020 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*. IEEE, 2020, pp. 1–6.

[12] J. Kim, D. Koo, Y. Kim, H. Yoon, J. Shin, and S. Kim, "Efficient privacy-preserving matrix factorization for recommendation via fully homomorphic encryption," *ACM Transactions on Privacy and Security (TOPS)*, vol. 21, no. 4, pp. 1–30, 2018.

[13] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, and Y. Liu, "{BatchCrypt}: Efficient homomorphic encryption for {Cross-Silo} federated learning," in *2020 USENIX annual technical conference (USENIX ATC 20)*, 2020, pp. 493–506.

[14] Y. N. D. Zhang Hongyi, Moustapha Cisse and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2018.

[15] W. Gao, X. Zhang, S. Guo, T. Zhang, T. Xiang, H. Qiu, Y. Wen, and Y. Liu, "Automatic transformation search against deep leakage from gradients," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 9, pp. 10 650–10 668, 2023.

[16] Y. Huang, Z. Song, K. Li, and S. Arora, "Instahide: Instance-hiding schemes for private distributed learning," in *International conference on machine learning*. PMLR, 2020, pp. 4507–4518.

[17] J. Wu, M. Hayat, M. Zhou, and M. Harandi, "Concealing sensitive samples against gradient leakage in federated learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 19, 2024, pp. 21 717–21 725.

[18] W. Wei, L. Liu, Y. Wu, G. Su, and A. Iyengar, "Gradient-leakage resilient federated learning," in *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2021, pp. 797–807.

[19] J. Wang, S. Guo, X. Xie, and H. Qi, "Protect privacy from gradient leakage attack in federated learning," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 2022, pp. 580–589.

[20] F. Wang, E. Hugh, and B. 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*. IEEE, 2023, pp. 1–10.

[21] K. Zhang, S. Cheng, G. Shen, B. Ribeiro, S. An, P.-Y. Chen, X. Zhang, and N. Li, "Censor: Defense against gradient inversion via orthogonal subspace bayesian sampling," in *32nd Annual Network and Distributed System Security Symposium, NDSS 2025*, 2025.

[22] D. Scheliga, P. Mäder, and M. Seeland, "Dropout is not all you need to prevent gradient leakage," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 8, 2023, pp. 9733–9741.

[23] ——, "Precode-a generic model extension to prevent deep gradient leakage," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 1849–1858.

[24] G. Wan, H. Du, X. Yuan, J. Yang, M. Chen, and J. Xu, "Enhancing privacy preservation in federated learning via learning rate perturbation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4772–4781.

[25] L. Xue, S. Hu, R. Zhao, L. Y. Zhang, S. Hu, L. Sun, and D. Yao, "Revisiting gradient pruning: A dual realization for defending against gradient attacks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 6, 2024, pp. 6404–6412.

[26] J. Sun, A. Li, B. Wang, H. Yang, H. Li, and Y. 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*, 2021, pp. 9311–9319.

[27] M. Balunovic, D. I. Dimitrov, R. Staab, and M. Vechev, "Bayesian framework for gradient leakage," in *International Conference on Learning Representations*, 2022.

[28] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik, "Emnist: Extending mnist to handwritten letters," in *2017 international joint conference on neural networks (IJCNN)*. IEEE, 2017, pp. 2921–2926.

[29] A. Krizhevsky, V. Nair, and G. Hinton, "The cifar-10 dataset. online: http://www. cs. toronto. edu/kriz/cifar. html, 55, 2014," *Cited on pages 73, 117, and*, vol. 120, 2014.

[30] D. Anguita, A. Ghio, L. Oneto, X. Parra, J. L. Reyes-Ortiz *et al.*, "A public domain dataset for human activity recognition using smartphones." in *Esann*, vol. 3, 2013, p. 3.

[31] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv preprint arXiv:1804.03209*, 2018.

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

[33] H. Yin, A. Mallya, A. Vahdat, J. M. Alvarez, J. Kautz, and P. Molchanov, "See through gradients: Image batch recovery via gradinversion," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 16 337–16 346.

[34] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[35] J. Jeon, K. Lee, S. Oh, J. Ok *et al.*, "Gradient inversion with generative image prior," *Advances in neural information processing systems*, vol. 34, pp. 29 898–29 908, 2021.

[36] H. Fang, B. Chen, X. Wang, Z. Wang, and S.-T. Xia, "Gifd: A generative gradient inversion method with feature domain optimization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4967–4976.

[37] F. Tramer, N. Carlini, W. Brendel, and A. Madry, "On adaptive attacks to adversarial example defenses," *Advances in neural information processing systems*, vol. 33, pp. 1633–1645, 2020.

[38] L. Jiang, Q. Song, R. Tan, and M. Li, "Primask: Cascadable and collusion-resilient data masking for mobile cloud inference," in *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, 2022, pp. 164–178.

[39] A. G. Sébert, M. Checri, O. Stan, R. Sirdey, and C. Gouy-Pailler, "Combining homomorphic encryption and differential privacy in federated learning," in *2023 20th Annual International Conference on Privacy, Security and Trust (PST)*. IEEE, 2023, pp. 1–7.

[40] W.-N. Chen, A. Ozgur, and P. Kairouz, "The poisson binomial mechanism for unbiased federated learning with secure aggregation," in *International Conference on Machine Learning*. PMLR, 2022, pp. 3490–3506.

[41] W. Gao, S. Guo, T. Zhang, H. Qiu, Y. Wen, and Y. Liu, "Privacy-preserving collaborative learning with automatic transformation search," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 114–123.

[42] Z. Zhang, Z. Tianqing, W. Ren, P. Xiong, and K.-K. R. Choo, "Preserving data privacy in federated learning through large gradient pruning," *Computers & Security*, vol. 125, p. 103039, 2023.

[43] S. P. Karimireddy, Q. Rebjock, S. Stich, and M. Jaggi, "Error feedback fixes signsgd and other gradient compression schemes," in *International Conference on Machine Learning*. PMLR, 2019, pp. 3252–3261.

[44] C. Palihawadana, N. Wiratunga, H. Kalutarage, and A. Wijekoon, "Mitigating gradient inversion attacks in federated learning with frequency transformation," in *European Symposium on Research in Computer Security*. Springer, 2023, pp. 750–760.

[45] P. R. Ovi, E. Dey, N. Roy, and A. Gangopadhyay, "Mixed quantization enabled federated learning to tackle gradient inversion attacks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 5046–5054.

[46] Y.-C. Hsu, T. Hua, S. Chang, Q. Lou, Y. Shen, and H. Jin, "Language model compression with weighted low-rank factorization," in *International Conference on Learning Representations*, 2022.

[47] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of massive data sets*. Cambridge university press, 2020.

[48] J. Shlens, "A tutorial on principal component analysis."

[49] R. Peter, G. Shivapratap, G. Divya, and K. Soman, "Evaluation of svd and nmf methods for latent semantic analysis," *International Journal of Recent Trends in Engineering*, vol. 1, no. 3, p. 308, 2009.

[50] C. Wu, F. Wu, L. Lyu, Y. Huang, and X. Xie, "Communication-efficient federated learning via knowledge distillation," *Nature communications*, vol. 13, no. 1, p. 2032, 2022.

[51] J. Kwon and H. Park, "Efficient low-rank federated learning based on singular value decomposition," in *Proceedings of the Twenty-Third International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, 2022, pp. 285–286.

[52] Y. Shin, K. Lee, S. Lee, Y. R. Choi, H.-S. Kim, and J. Ko, "Effective heterogeneous federated learning via efficient hypernetwork-based weight generation," pp. 112–125, 2024.

[53] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE signal processing magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[54] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," in *International conference on machine learning*. PMLR, 2018, pp. 284–293.

[55] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *International conference on machine learning*. PMLR, 2018, pp. 274–283.

[56] A. Alshamrani, S. Myneni, A. Chowdhary, and D. Huang, "A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1851–1877, 2019.

[57] R. Gonzalez and R. Woods, "Digital image processing 3rd ed," 2020.

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

[59] Y. Deng, W. Chen, J. Ren, F. Lyu, Y. Liu, Y. Liu, and Y. Zhang, "Tailorfl: Dual-personalized federated learning under system and data heterogeneity," in *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, 2022, pp. 592–606.

[60] K. Hsieh, A. Phanishayee, O. Mutlu, and P. Gibbons, "The non-iid data quagmire of decentralized machine learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 4387–4398.

[61] Y. Xu, Y. Liao, L. Wang, H. Xu, Z. Jiang, and W. Zhang, "Overcoming noisy labels and non-iid data in edge federated learning," *IEEE Transactions on Mobile Computing*, 2024.

[62] Z. Lu, H. Pan, Y. Dai, X. Si, and Y. Zhang, "Federated learning with non-iid data: A survey," *IEEE Internet of Things Journal*, 2024.

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

[64] N. Srebro and T. Jaakkola, "Weighted low-rank approximations," in *Proceedings of the 20th international conference on machine learning (ICML-03)*, 2003, pp. 720–727.

[65] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *International conference on machine learning*. PMLR, 2020, pp. 5132–5143.

[66] D. Chen, J. Hu, V. J. Tan, X. Wei, and E. Wu, "Elastic aggregation for federated optimization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 12 187–12 197.

[67] A. Ahmad, W. Luo, and A. Robles-Kelly, "Robust federated learning under statistical heterogeneity via hessian-weighted aggregation," *Machine Learning*, vol. 112, no. 2, pp. 633–654, 2023.

[68] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1d convolutional neural networks and applications: A survey," *Mechanical systems and signal processing*, vol. 151, p. 107398, 2021.

[69] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.
[70] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, pp. 211–252, 2015.
[71] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.
[72] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[73] J. Deng, Y. Wang, J. Li, C. Shang, H. Liu, S. Rajasekaran, and C. Ding, "Tag: Gradient attack on transformer-based language models," *arXiv preprint arXiv:2103.06819*, 2021.
[74] M. Balunovic, D. Dimitrov, N. Jovanović, and M. Vechev, "Lamp: Extracting text from gradients with language model priors," *Advances in Neural Information Processing Systems*, vol. 35, pp. 7641–7654, 2022.
[75] X. Feng, Z. Ma, Z. Wang, E. J. Chegne, M. Ma, A. Abuadbba, and G. Bai, "Uncovering gradient inversion risks in practical language model training," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2024, pp. 3525–3539.

APPENDIX

A. Analysis of Noise Injection-based Defenses under EoT Attack

In GIAs, the adversary optimizes the dummy input $\mathbf{x}'$ by minimizing $\mathcal{D}(\nabla\mathbf{\Theta}, \nabla\mathbf{\Theta}')$, where $\nabla\mathbf{\Theta}$ and $\nabla\mathbf{\Theta}'$ represent the ground truth and dummy gradients, respectively, and $\mathcal{D}$ represents the distance metric. The distance metric $\mathcal{D}$ quantifies the distance between the two gradients. To simplify our analysis, we consider $\mathcal{D}$ to be the Euclidean distance. Under defense, the GIA objective is:

$$\min \|\varphi(\nabla\mathbf{\Theta}) - \nabla\mathbf{\Theta}'\|_F, \tag{5}$$

where $\varphi(\cdot)$ denotes the defense operation and $\|\cdot\|_F$ is the Frobenius norm.

Take DP-Gau [5] as an example, $\varphi(\cdot)$ represents injecting noise η sampled from a Gaussian distribution $\mathcal{N}(0, \sigma^2)$ to input data. Then, the observed gradients can be denoted by $\varphi(\nabla\mathbf{\Theta}) = \nabla\mathbf{\Theta} + \eta$ and the optimization objective of GIA under DP-Gau becomes:

$$\min \|\nabla\mathbf{\Theta} + \eta - \nabla\mathbf{\Theta}'\|_F. \tag{6}$$

When an adaptive adversary applies the Expectation over Transformation (EoT) [21] by sampling n times from the Gaussian distribution $\mathcal{N}(0, \sigma^2)$ to add noise to and average the perturbed input, the dummy gradients become $\nabla\mathbf{\Theta}'+\eta'$, where $\eta' \sim \mathcal{N}(0, \frac{\sigma^2}{n})$. Consequently, the optimization objective becomes:

$$\min \|\nabla\mathbf{\Theta} - \nabla\mathbf{\Theta}' + \eta - \eta'\|_F, \tag{7}$$

where $\eta - \eta'$ follows a Gaussian distribution $\mathcal{N}(0, \frac{n+1}{n}\sigma^2)$. Given the objectives in Eqs. 6 and 7, the optimized dummy gradients $\nabla\mathbf{\Theta}^*$ is approximated as $\nabla\mathbf{\Theta} + \eta$ and $\nabla\mathbf{\Theta} + \eta - \eta'$, respectively. As theoretically proven in [25], the attack effectiveness of GIAs, which is characterized by the data reconstruction error, is lower bounded by the overall gradient error between the ground-truth and optimized dummy gradients, which is $\|\nabla\mathbf{\Theta} - \nabla\mathbf{\Theta}^*\|_F$. By applying the EoT operation, the data reconstruction error lower bound for DP-Gau changes from $\|\eta\|_F$ to $\|\eta-\eta'\|_F$, with increased variance. Consequently, the attack becomes less effective with EoT operation. A similar analysis can be done on Outpost that injects Gaussian noise in the gradients based on leakage risks.

For DP-Lap, the injected noise η follows Laplace distribution $Laplace(0, b)$ with mean 0 and variance $2b^2$. The noise η' introduced in the dummy gradients when the adversary applies EoT follows a distribution with mean 0 and variance $\frac{2b^2}{n}$. Thus, $\eta - \eta'$, follows a new distribution with mean 0 and variance $\frac{(2+2n)b^2}{n}$, which is greater than the variance of η, leading to an increased variance in the lower bound of the data reconstruction error. We can conclude that the EoT operation also deteriorates the attack effectiveness against DP-Lap.

In conclusion, under noise injection-based defenses including DP-Gau, Outpost, and DP-Lap, the adaptive attack operation of EoT increases the variance of the lower bound of data reconstruction error, leading to reduced attack effectiveness.

TABLE IX: Defense Performance of LRP and MQ Under Non-adaptive and Adaptive GIAs.

Defense \ Metric	Non-adaptive		Adaptive	
	PSNR	LPIPS	PSNR	LPIPS
LRP [24]	12.8634	0.4995	31.2368	0.1459
MQ [45]	4.3438	0.7567	45.7481	0.0032

B. Effectiveness of Adaptive Attack under LRP and MQ

This section evaluates the defense performance of LRP [24] and MQ [45] under both non-adaptive and adaptive attackers.

LRP defends against GIAs by assigning randomly sampled learning rates to clients, concealing them from the attacker. In our experiments, we set the client learning rate to be 0.1. For the non-adaptive attack, we implement the DLG attack [5], as LRP has been shown to be resistant to DLG but vulnerable to the IG attack. We then configure the adversary's dummy learning rate to be 0.2, simulating the LRP defense. For the adaptive attack against LRP, we initialize a trainable dummy learning rate and optimize it together with the dummy input during the attack optimization process.

MQ defends against GIAs by hiding the gradient range information. For the non-adaptive attack, we implement the IG attack and directly use quantized gradients for input reconstruction. For the adaptive attack, similar to LRP, the adversary can initialize trainable dummy minimum and maximum gradient vectors, which are jointly optimized together with the dummy input during attack optimization process. The recovered hidden gradient range can be used to dequantize the gradients.

Table IX presents the defense performance of LRP and MQ. The results show that the adaptive attacks significantly weaken the defense effectiveness of both methods.

C. Impact of Energy Threshold on Attack Effectiveness

Fig. 14 illustrates the effectiveness of IG attack under truncated SVD with varying energy thresholds using an image from the MNIST dataset. As the energy threshold increases,

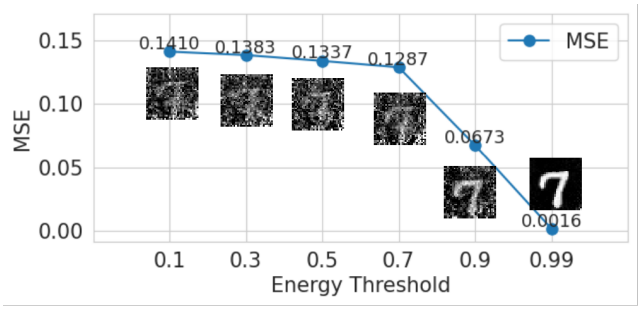


Fig. 14: The attack performance under different energy thresholds.

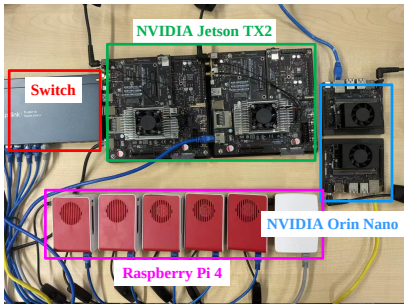


Fig. 15: An illustration of our federated learning testbed with various embedded platforms.

the MSE decreases and image reconstruction quality also increases. This is because truncated SVD with higher energy threshold retains more singular values and corresponding vectors, preserving more information for data reconstruction. This exemplifies that clients with smaller energy thresholds during SVD truncation receive stronger protection against GIAs.

D. Theoretical Analysis of truncated SVD with Channel-Wise Weighted Approximation

Theorem 1. *For any (ε, δ)-passive attack $\mathcal{A}$, Under the presence of truncated SVD, it will degenerate to $(\varepsilon + \sqrt{\gamma_1}\|\nabla\mathbf{\Theta}\|_F, \delta)$, where $\gamma_1 = 1 - \mathcal{T}$. Under the presence of truncated SVD with Channel-Wise Weighted Approximation, it will degenerate to $(\varepsilon + \sqrt{\gamma_2}\|\nabla\mathbf{\Theta}\|_F, \delta)$, where $\gamma_2 = (\frac{\sigma_{max}(\mathbf{I})}{\sigma_{min}(\mathbf{I})})^2(1-\mathcal{T})$, $\sigma_{max}(\cdot)$ and $\sigma_{min}(\cdot)$ mean the maximum and minimum singular value of the input matrix.*

Poof. By definition, an (ε, δ)-passive attack allows the attacker to achieve:

$$\mathbb{E}\|\nabla\mathbf{\Theta} - \nabla\mathbf{\Theta}^*\|_F \leq \varepsilon, \tag{8}$$

where $\nabla\mathbf{\Theta}^*$ is the attacker's optimized gradients of the ground-truth gradients $\nabla\mathbf{\Theta}$. The Frobenius norm of a matrix can be expressed as $\|\mathbf{A}\|_F^2 = \sum_{i=1}^{r}\sigma_i^2(\mathbf{A})$, where $\sigma_i(\cdot)$ denotes the i-th singular value of $\mathbf{A}$ and r is the rank of $\mathbf{A}$.

By applying truncated SVD (tSVD) on $\nabla\mathbf{\Theta}$ to retain the top-k singular values based on the energy threshold $\mathcal{T}$, we have:

$$\sum_{i=1}^{k}\sigma_i^2(\nabla\mathbf{\Theta}) \geq \mathcal{T}\cdot\|\nabla\mathbf{\Theta}\|_F^2. \tag{9}$$

With tSVD, the attacker observes only the truncated gradients. Therefore, the error between the ground-truth gradients and optimized gradients becomes:

$$\begin{aligned}
&\mathbb{E}\|\nabla\mathbf{\Theta} - \nabla\mathbf{\Theta}^*\|_F \\
&= \mathbb{E}\|\nabla\mathbf{\Theta} - \text{tSVD}(\nabla\mathbf{\Theta}, \mathcal{T}) + \text{tSVD}(\nabla\mathbf{\Theta}, \mathcal{T}) - \nabla\mathbf{\Theta}^*\|_F \\
&\overset{(a)}{\leq} \|\text{tSVD}(\nabla\mathbf{\Theta}, \mathcal{T}) - \nabla\mathbf{\Theta}^*\|_F + \|\nabla\mathbf{\Theta} - \text{tSVD}(\nabla\mathbf{\Theta}, \mathcal{T})\|_F \\
&\overset{(8)}{\leq} \varepsilon + \|\nabla\mathbf{\Theta} - \text{tSVD}(\nabla\mathbf{\Theta}, \mathcal{T})\|_F \\
&= \varepsilon + \sqrt{\sum_{i=k+1}^{r}\frac{\sigma_i(\nabla\mathbf{\Theta})^2}{\sum_{j=1}^{r}\sigma_j(\nabla\mathbf{\Theta})^2}}\|\nabla\mathbf{\Theta}\|_F \\
&\overset{(9)}{\leq} \varepsilon + \sqrt{(1-\mathcal{T})}\|\nabla\mathbf{\Theta}\|_F \\
&= \varepsilon + \sqrt{\gamma_1}\|\nabla\mathbf{\Theta}\|_F,
\end{aligned} \tag{10}$$

where $\gamma_1 = 1 - \mathcal{T}$. (a) is based on the Frobenius norm triangle inequality, which states that for any two matrices A and B, the inequality $\|A + B\|_F \leq \|A\|_F + \|B\|_F$ holds.

By augmenting tSVD with Channel-Wise Weighted Approximation, the error between the ground-truth gradients and optimized gradients becomes:

$$\begin{aligned}
&\mathbb{E}\|\nabla\mathbf{\Theta} - \nabla\mathbf{\Theta}^*\|_F \\
&= \mathbb{E}\|\nabla\mathbf{\Theta} - \mathbf{I}^{-1}\text{tSVD}(\mathbf{I}\nabla\mathbf{\Theta}, \mathcal{T}) \\
&\quad + \mathbf{I}^{-1}\text{tSVD}(\mathbf{I}\nabla\mathbf{\Theta}, \mathcal{T}) - \nabla\mathbf{\Theta}^*\|_F \\
&\overset{(a)}{\leq} \|\mathbf{I}^{-1}\text{tSVD}(\mathbf{I}\nabla\mathbf{\Theta}, \mathcal{T}) - \nabla\mathbf{\Theta}^*\|_F \\
&\quad + \|\nabla\mathbf{\Theta} - \mathbf{I}^{-1}\text{tSVD}(\mathbf{I}\nabla\mathbf{\Theta}, \mathcal{T})\|_F \\
&\overset{(8)}{\leq} \varepsilon + \|\nabla\mathbf{\Theta} - \mathbf{I}^{-1}\text{tSVD}(\mathbf{I}\nabla\mathbf{\Theta}, \mathcal{T})\|_F \\
&= \varepsilon + \|\mathbf{I}^{-1}(\mathbf{I}\nabla\mathbf{\Theta} - \text{tSVD}(\mathbf{I}\nabla\mathbf{\Theta}, \mathcal{T}))\|_F \\
&\overset{(b)}{\leq} \varepsilon + \|\mathbf{I}^{-1}\|_2\|\mathbf{I}\nabla\mathbf{\Theta} - \text{tSVD}(\mathbf{I}\nabla\mathbf{\Theta}, \mathcal{T})\|_F \\
&\overset{(9)}{\leq} \varepsilon + \sigma_{max}(\mathbf{I}^{-1})\sqrt{(1-\mathcal{T})}\|\mathbf{I}\nabla\mathbf{\Theta}\|_F \\
&\overset{(b)}{\leq} \varepsilon + \sigma_{max}(\mathbf{I}^{-1})\sqrt{(1-\mathcal{T})}\|\mathbf{I}\|_2\|\nabla\mathbf{\Theta}\|_F \\
&= \varepsilon + \sigma_{max}(\mathbf{I}^{-1})\sigma_{max}(\mathbf{I})\sqrt{(1-\mathcal{T})}\|\nabla\mathbf{\Theta}\|_F \\
&= \varepsilon + \frac{\sigma_{max}(\mathbf{I})}{\sigma_{min}(\mathbf{I})}\sqrt{(1-\mathcal{T})}\|\nabla\mathbf{\Theta}\|_F \\
&= \varepsilon + \sqrt{\gamma_2}\|\nabla\mathbf{\Theta}\|_F,
\end{aligned} \tag{11}$$

where $\|\cdot\|_2$ represents the spectral norm, i.e., the maximum singular value of the matrix, and $\gamma_2 = (\frac{\sigma_{max}(\mathbf{I})}{\sigma_{min}(\mathbf{I})})^2(1-\mathcal{T})$. (b) is based on the submultiplicativity of matrix norms, which states that for any matrices A and B, the inequality $\|AB\|_F \leq \|A\|_2 \cdot \|B\|_F$ holds. Since $\frac{\sigma_{max}(\mathbf{I})}{\sigma_{min}(\mathbf{I})} \geq 1$, we have $\gamma_2 \geq \gamma_1$ under the same energy threshold $\mathcal{T}$. This indicates that tSVD with Channel-Wise Weighted Approximation provides stronger protection than the original tSVD. Hence, this theorem holds.

E. Visualization of our FL Testbed and Reconstructed Examples

Fig. 15 illustrates our FL testbed. To visualize defense

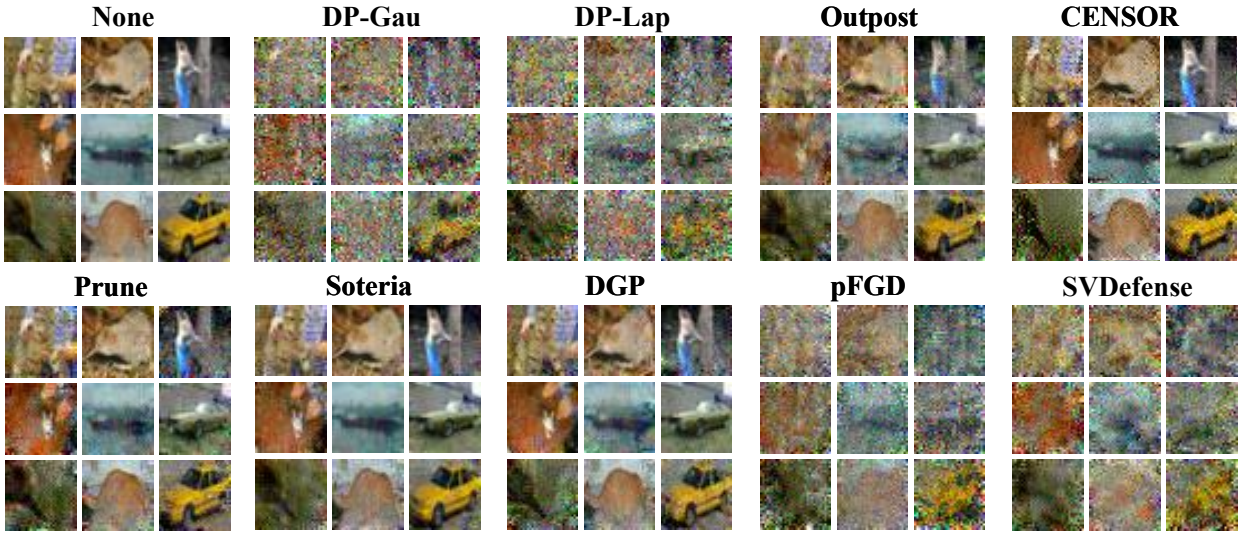


Fig. 16: Visual examples of reconstructed inputs on CIFAR-10.

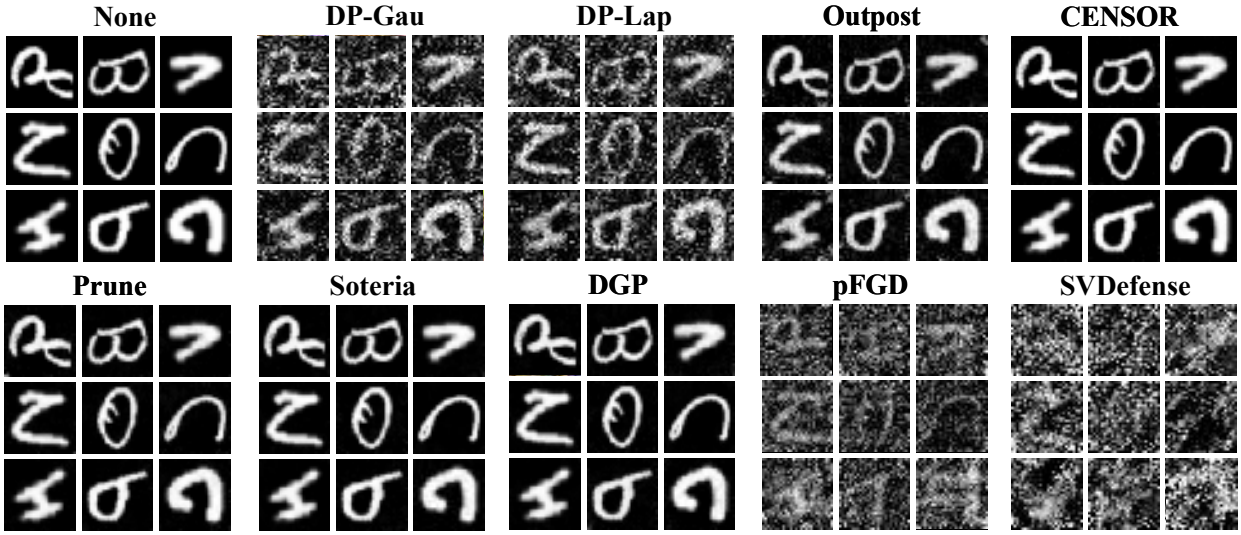


Fig. 17: Visual examples of reconstructed inputs on EMNIST.

effectiveness, we present examples of reconstructed CIFAR-10 and EMNIST images under different defense methods in Fig. 16 and Fig. 17, respectively. We can observe that the reconstructed examples from *SVDefense* are not recognizable for both datasets.

F. ARTIFACT APPENDIX

1) Description & Requirements: We provide *SVDefense*, a novel defense framework against GIAs that leverages the truncated Singular Value Decomposition (SVD) to obfuscate gradient updates. *SVDefense* integrates three core functionalities: *SVDefense* introduces three key innovations, the Self-Adaptive Energy Threshold that adapts the privacy protection for clients with different vulnerability to GIAs caused by their varying degrees of class imbalance, the Channel-Wise Weighted Approximation that selectively preserves essential gradient information for model training while enhancing privacy protection, and the Layer-Wise Weighted Aggregation for effective aggregation of client updates under class imbalance. We have developed a comprehensive platform that evaluate the defense performance and classification performance of existing defenses and *SVDefense*. In this artifact, we take the cifar10 dataset as an example.

- **How to access**: Our implementation is available on Zenodo with DOI: https://doi.org/10.5281/zenodo.16948135.
- **Hardware dependencies**: GPU: 8GB (optional), CPU: 8 cores, RAM: 16GB and Disk space: 100 GB of space.
- **Software dependencies**:
  1) Operation System: Ubuntu 22.04.
  2) Package management system: Conda (or Miniconda).
- **Benchmarks**: cifar10.

2) Artifact Installation & Configuration:

- **Create workspace**: Download the code repository from GitHub, and name as `~/svdefense` workspace.
- **Environment setup**: Navigate to `~/svdefense/scripts` and run the setup script `./setup.sh` to configure the development environment.
- **Software**: Install the parallel using `sudo apt-get install parallel` to run experiments in parallel.

3) Experiment E1: Defense performance under IG attack: about [20 human-minutes + 100 compute-hours] This experiment aims to:

- Assess the functionality of the *Svdefense*.
- Evaluate *Svdefense*'s effectiveness under the IG attack, as Table III details.

[Preparation]

- Change the `defense` parameter in the `~/svdefense/scripts/defense_IG.sh` to evaluate different defense methods. Here we provide some examples including 'none', 'dp', 'outpost', 'prune', 'dgp', 'pfgd', and 'svdefense'. We can comment out the command to test the performance of the defense.
- Using the 'parallel' command can reconstruct multiple images concurrently. The usage of the 'parallel' command can be found in the `defense_IG.sh`. The default command is to reconstruct one image for testing.

[Execution]

- Main Script: Navigate to `~/svdefense/scripts` and run the setup script `./defense_IG.sh`.

[Results] The reconstructed images can be found in the folder `~/svdefense/IG_attack/recon_data/`, and the ground-truth images can be found in the `~/svdefense/IG_attack/gt_data/`. Then we can use the `cal_matric.py` to output the metrics in Table III.

4) Experiment E2: Training perturbation-based defense performance under IG attack: about [2 human-minutes + 1 compute-hours] This experiment aims to:

- Evaluate existing training perturbation-based defenses' effectiveness under the adaptive attack, as Table III details.
- We take the PRECODE as an example.

[Preparation]

- Comment out the code in the 137 line of `PRECODE/invertinggradients/inversefed/reconstruction_algorithms.py` to enable the adaptive attack and vice versa.

[Execution]

- Main Script: Navigate to `~/svdefense/scripts` and run the setup script `./defense_PRECODE.sh`.

[Results] The reconstructed images can be found in the folder `~/svdefense/PRECODE/recon_data/`, and the ground-truth images can be found in the `~/svdefense/PRECODE/gt_data/`. Then `cal_matric.py` can output the metrics in Table III

5) Experiment E3: Defense performance under ROG attack:

[Preparation]

- Download the needed pretrained weights following the public GitHub repository https://github.com/KAI-YUE/rog.
  1) Download the pretrained models [1] and put them under `ROG_attack/model_zoos/`.
  2) Download the csv file [2] and put it under *ROG_attack/data* folder.
- Change the `defense` parameter in the configuration file `ROG_attack/utils/config_fedavg.yaml` to evaluate different defenses. Here we provide some examples including 'none','dp', 'outpost', 'prune', 'dgp', 'pfgd', and 'svdefense'.

[Execution]

- Main Script: Navigate to `~/svdefense/scripts` and run the script `./defense_ROG.sh`.

[Results] The reconstructed images can be found in the folder `~/svdefense/ROG_attack/recon_data/`, and the ground-truth images can be found in the `~/svdefense/ROG_attack/gt_data/`. Then `python cal_matric.py` can output the metrics in Table V.

6) Experiment E4: Classification performance on cifar10 dataset:

[Preparation]

- Here we provide some examples including 'none', 'outpost', 'prune', 'dgp', 'pfgd', and 'svdefense'. We can update the parameters in `~/svdefense/svd-defense/pyproject.toml` to test the performance of the defense. Noted that, to control for confounding factors, the layer-wise weighted aggregation is omitted so that the analysis focuses solely on the defense applied to gradients.
- We can change the 'local-defense' parameter in the configuration file to evaluate different defense methods in the Federated learning setting.

[Execution]

- Main Script: Navigate to `~/svdefense/scripts` and run the script `./fl.sh`.
- Change the parameters 'server-device' and 'client-device' to 'cuda' to accelerate the training process.

[Results] The accuracy across epochs can be found in `~/svdefense/svd-defense/{defense}_acc.txt`. Then `python draw.py` can output one line of results of the corresponding defense in Fig.7.

[1] https://huggingface.co/erickyue/rog_modelzoo/tree/main

[2] https://storage.googleapis.com/openimages/v6/oidv6-class-descriptions.csv