LOPA: LEARNING LOCAL SUBSPACES FOR PRIVACY PROTECTION IN SPLIT FEDERATED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Split Federated Learning (SFL) reduces the computational burden on clients through collaborative training between clients and a server. However, an honest-but-curious server can still reconstruct private images from uploaded intermediate representations. Existing defenses limit information exposure through activation dropping, quantization, or perturbation, but these operations do not necessarily suppress reconstruction-relevant information while preserving information needed for classification, and stronger restrictions may degrade task performance. We observe that low-rank channel projection combined with normalization and spatial pooling can reduce reconstruction quality, but random selection of subspace directions does not consistently preserve classification accuracy. To address this issue, we propose LOPA (Local Privacy-Aware Subspace Adaptation), which uses only local client data to initialize a task-relevant subspace based on between-class and within-class scatter statistics. It then calibrates the subspace using classification loss and distance correlation to preserve task-discriminative information while reducing the statistical dependence of representations on raw inputs. The clients' orthogonal projection matrices are aggregated into a shared subspace before training, which remains fixed during online training to filter uploaded representations without further subspace optimization. Experiments on CelebA, CIFAR-10, and FMNIST show that LOPA substantially degrades reconstruction quality under both the FORA and Inverse-Network attacks. Unlike the compared defenses, which trade accuracy for protection, LOPA matches undefended SFL on all three datasets. Furthermore, subspace initialization takes only seconds and requires less memory than client-side training itself.

1 INTRODUCTION

When training data are distributed across multiple clients and cannot be collected centrally, Federated Learning (FL) enables collaborative learning through local model training and model update aggregation, without directly transmitting raw data (McMahan et al., 2017). However, conventional FL requires each client to train a complete model, imposing substantial computational and memory overhead on resource-constrained devices. To alleviate this burden, Split Learning (SL) partitions the model into a client-side subnetwork and a server-side subnetwork. Specifically, the client processes local inputs and uploads intermediate activations, commonly referred to as smashed data, to the server for subsequent computation. The two sides jointly train the model by exchanging activations and gradients (Gupta & Raskar, 2018). To overcome the sequential client training in the original SL protocol, Split Federated Learning (SFL) further incorporates federated aggregation, allowing multiple clients to train their client-side subnetworks in parallel (Thapa et al., 2022). Nevertheless, keeping raw data locally does not prevent information leakage through the uploaded intermediate representations. Even when faithfully following the prescribed training protocol, an honest-but-curious (Paverd et al., 2014) server may record the intermediate activations and use publicly available auxiliary images drawn from the same distribution to train a reconstruction model, thereby recovering clients' private images (Gao & Zhang, 2023; Xu et al., 2024). Therefore, an effective defense for SFL should reduce the risk of reconstructing private images from uploaded representations while remaining within the limited computational budgets of clients.

In recent years, researchers have proposed a series of defense methods to reduce reconstruction risks by modifying intermediate representations or the client-side training process (Vepakomma

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

et al., 2020; Li et al., 2022a; He et al., 2020; Pham et al., 2023; Yue et al., 2025). For example, Dropout (He et al., 2020) randomly zeros out a subset of activations to hinder attackers from reconstructing raw images from intermediate features. B-SL (Pham et al., 2023) binarizes client-side weights and activations, reducing the numerical precision of representations as well as client-side computation and storage costs. AdapLDP-FL (Yue et al., 2025) adaptively perturbs model updates to limit information leakage during federated training. These methods restrict information exposure through random dropping, quantization, or update perturbation. However, these operations do not necessarily suppress information useful for reconstruction while preserving information needed for classification. Previous studies have shown that protected intermediate representations may still contain information that reconstruction models can exploit (Xu et al., 2024). Stronger restrictions may reduce reconstruction quality, but they may also remove information needed for classification, thereby degrading task performance. Therefore, it is necessary to investigate how to selectively preserve task-relevant information in uploaded representations while reducing information useful for image reconstruction, within the limited computational budgets of clients.

To answer the above question, we first examine the effect of imposing structural constraints along the channel and spatial dimensions. Specifically, spatial pooling reduces fine-grained spatial information in intermediate representations, while low-rank channel projection restricts the feature directions that are retained. Our experiments show that, even when the subspace is constructed from a random orthonormal basis, this structural bottleneck combined with normalization and spatial pooling can noticeably reduce the quality of images reconstructed by FORA (Xu et al., 2024). For example, on CelebA (Liu et al., 2015), LPIPS (Zhang et al., 2018), which measures the perceptual distance between reconstructed and original images, increases from 0.195 for SFL to 0.623 for the random-subspace configuration. However, random subspace selection does not consider the contribution of different directions to classification, and its effect on accuracy is inconsistent across datasets, yielding both slight gains and losses relative to undefended SFL. These results suggest that, beyond imposing structural constraints, it is necessary to further investigate how to select the retained directions according to the classification task. Moreover, this selection process should accommodate limited client-side computational resources without relying on additional external data or costly online optimization. Therefore, it's a challenging task to select classification-relevant subspace directions using only local client data and limited additional computation, thereby improving task utility while maintaining reconstruction resistance.

In this paper, we propose LOPA (Local Privacy-Aware Subspace Adaptation), an SFL defense method that learns a privacy-preserving subspace from local client data. LOPA first initializes the subspace using local class statistics. Specifically, it constructs a scoring matrix from the between-class and within-class scatter matrices and uses eigendecomposition to select channel directions that favor class separation while penalizing within-class variation. This initialization requires no iterative gradient optimization. LOPA then calibrates the initial subspace under a joint objective combining classification loss and distance correlation between inputs and representations, aiming to retain task-discriminative information while reducing the statistical dependence between the representations and raw inputs. Since the same subspace can be represented by different orthonormal bases, LOPA aggregates the clients' orthogonal projection matrices rather than their basis matrices. This makes the aggregation invariant to sign changes and orthogonal rotations of the local bases. The resulting aggregate is used to construct a shared subspace before SFL training begins. During online training, the shared subspace remains fixed and filters intermediate representations before transmission, while the client-side and server-side models continue to be optimized under this constraint. This design has three characteristics. First, subspace fitting uses only the client's local data, requiring neither an additional auxiliary dataset nor an expert node with access to source-task data. Second, subspace initialization and calibration are performed only before training; during the online stage, there is no need to continuously optimize the subspace, train a temporary classification head, or compute the distance correlation required for calibration, and the additional computation mainly comes from representation filtering and its backward pass. Third, the shared subspace is determined before the first upload, so that the complete filtering mechanism operates at full strength from the start of training, without gradually learning the subspace during online training.

To evaluate the effectiveness of LOPA, we use three widely used (Kulkarni et al., 2025; Yi et al., 2025) datasets, including CelebA (Liu et al., 2015), CIFAR-10 (Krizhevsky et al., 2009), and FMNIST (Xiao et al., 2017). To assess reconstruction risk, we consider two attack settings. FORA (Xu et al., 2024) trains a substitute client and a reconstruction model concurrently with SFL train-

ing, while Inverse-Network (He et al., 2019) learns an inverse mapping after training, using auxiliary images from the same distribution together with their actual uploaded representations. Experimental results show that LOPA substantially reduces reconstruction quality under both attacks. On CelebA, LPIPS of the images reconstructed by FORA increases from 0.195 for SFL to 0.624, and identity similarity decreases from 0.595 to 0.110. At the same time, LOPA achieves classification accuracy no lower than undefended SFL on all three datasets, whereas Dropout and B-SL incur accuracy loss on all three datasets. In addition, subspace initialization takes at most 5.16 seconds on the client side, with a peak memory footprint lower than that of client-side training itself, indicating that this procedure can be executed on resource-constrained clients.

In summary, the main contributions of this paper are:

- We propose LOPA, a client-side defense for split federated learning that constrains uploaded representations through low-rank channel projection and selects the retained directions under this constraint, without requiring an auxiliary dataset or an expert node.
- We design a subspace construction mechanism that combines class-aware initialization, distance-correlation calibration, and projection-matrix aggregation. The subspace is learned offline before training and remains frozen, requiring no further subspace optimization during online training.
- We conduct extensive experiments on three datasets under two reconstruction attacks. The results show that, among the compared split-learning defenses, LOPA reduces reconstruction quality while maintaining classification accuracy comparable to undefended SFL.

2 Background and Related Works

Federated Learning (FL) enables collaborative training by keeping raw data on clients and exchanging model updates (McMahan et al., 2017). However, these updates may still leak private information. To address this risk, AdapLDP-FL (Yue et al., 2025) employs adaptive local differential privacy to protect model updates while mitigating noise-induced performance degradation. Nevertheless, FL typically requires clients to train the complete task model, imposing a substantial computational burden. Split Learning (SL) (Gupta & Raskar, 2018) partitions the model into subnetworks deployed across different participating entities, allowing clients to train only part of the model and thereby reducing their computational requirements. SL has been applied to multimodal learning, language model fine-tuning, and edge computing (Sun & Ochiai, 2024; Shen et al., 2025; Kim et al., 2024; Tirana et al., 2024). To improve the training efficiency of SL, Split Federated Learning (SFL) (Thapa et al., 2022) combines model partitioning with federated aggregation, enabling multiple clients to train in parallel. Clients exchange intermediate representations and their gradients with the main server to perform training, while a federation server aggregates and distributes client-side model parameters. SFL includes schemes such as SFL-V1 and SFL-V2, and this paper focuses primarily on the SFL-V2 scheme.

The intermediate representations uploaded in SFL may still be exploited by the server to reconstruct raw data. Dropout (He et al., 2020) randomly masks activations during inference to impede image reconstruction. B-SL (Pham et al., 2023) binarizes client-side weights and activations to reduce both information exposure and computational overhead. NoPeek (Vepakomma et al., 2020) jointly optimizes classification loss and input–representation distance correlation, encouraging the feature extractor to reduce the dependence of intermediate representations on raw inputs while retaining predictive utility. ResSFL (Li et al., 2022a) transfers reconstruction resistance from an expert node to clients through attacker-aware pretraining. Different from the existing work, LOPA builds a privacy-preserving subspace from local client data within the SFL paradigm, retaining the resource advantage of deploying only part of the task model on each client without requiring an additional auxiliary dataset or an expert node. LOPA jointly constrains information across multiple channels through the subspace and uses class-aware initialization and local calibration to retain task-discriminative information, balancing reconstruction risk and task utility. The shared subspace is determined before the first upload, enabling the complete filtering mechanism to operate from the start of training. During training, the shared subspace remains fixed while the client-side and server-side models continue to be optimized under this constraint. This keeps the additional computation limited to representation filtering and its backward pass.

3 PROBLEM FORMULATION

In SFL-V2, the complete network F is partitioned into a client-side subnetwork F_c and a server-side subnetwork F_s. Consider a system comprising N clients, where the i-th client holds a private dataset $\mathcal{D}_i$ and maintains a local replica of the client subnetwork, denoted as $F_c^{(i)}$. Meanwhile, the main server maintains the shared server subnetwork F_s, and the federation server is responsible for federated aggregation of the client-side models.

During each training iteration, a client feeds its local dataset x into $F_c^{(i)}$ to produce an intermediate feature $\mathcal{A}_t^{(i)}$ at the cut layer. This feature is transmitted to the main server as smashed data, where F_s completes the forward pass and computes the task loss. Subsequently, the main server updates its model parameters and backpropagates the cut-layer gradients to the client, enabling the client subnetwork to compute gradients and update its parameters. At the conclusion of each local training epoch, the federation server executes federated aggregation over the client-side model parameters and broadcasts the aggregated weights back to all clients, which serve as the initialization for the subsequent round.

Task Performance Objective. The primary objective of SFL is to jointly optimize the client and main server subnetworks such that the integrated model achieves competitive predictive performance over the target data distribution. In this work, we focus on image classification tasks, aiming to train a model with high classification accuracy.

Threat Model. We assume an honest-but-curious (HBC) (Paverd et al., 2014) adversary at the main server, a threat model that is widely adopted in privacy studies of split learning (Li et al., 2022b; Xu et al., 2024). The server adheres to the prescribed training protocol, executes model updates, and returns the requisite gradients to the clients. However, it may log and analyze the received smashed data throughout the training process to infer the private inputs of clients. Specifically, the server can harvest the set of intermediate activations $\mathcal{A}_t^{(i)}$ uploaded by the i-th client at round t, and leverage these cross-client and cross-round observations to train a data reconstruction model. The server's objective is to reconstruct the sensitive raw images from the observed intermediate representations without direct access to the client-side training data.

Privacy Protection Objective. To address the threat of an honest-but-curious server reconstructing private images from the observed intermediate representations $\mathcal{A}_t^{(i)}$, we use a client-side filter Φ_U to transform intermediate features Z into uploaded representations $\widetilde{Z}$, aiming to preserve classification information while reducing the server's ability to reconstruct private images.

4 METHODOLOGY

As illustrated in Fig. 1, LOPA proceeds through five stages. First, each client initializes its privacy subspace using class-aware scatter. Second, LOPA refines the initial subspace through tuning of privacy and utility. Third, the federated server aggregates the local subspaces into a shared privacy subspace. Fourth, clients use the shared subspace to filter their intermediate features. Finally, the client and server models are optimized through split federated learning.

4.1 PRIVACY SUBSPACE INITIALIZATION

Inspired by the Maximum Margin Criterion (MMC) (Li et al., 2006), we leverage local class statistics to select channel directions characterized by high inter-class separability and low intra-class variation, establishing an initial subspace for subsequent joint privacy-utility fine-tuning.

Local images are fed into the frozen client feature extractor to yield intermediate representations $Z \in \mathbb{R}^{C\times H\times W}$. We sequentially apply position-wise channel normalization, subtract the spatial mean, and perform adaptive average pooling to obtain $Q \in \mathbb{R}^{C\times g\times g}$. These operations dampen feature magnitudes while preserving coarse-grained structural layouts.

Each Q_i contains $S = g^2$ spatial positions, where $q_{i,s} \in \mathbb{R}^C$ denotes the feature vector at position s. For a local fitting set of n samples, let n_c denote the number of samples in class c, and let $\mu_{c,s}$

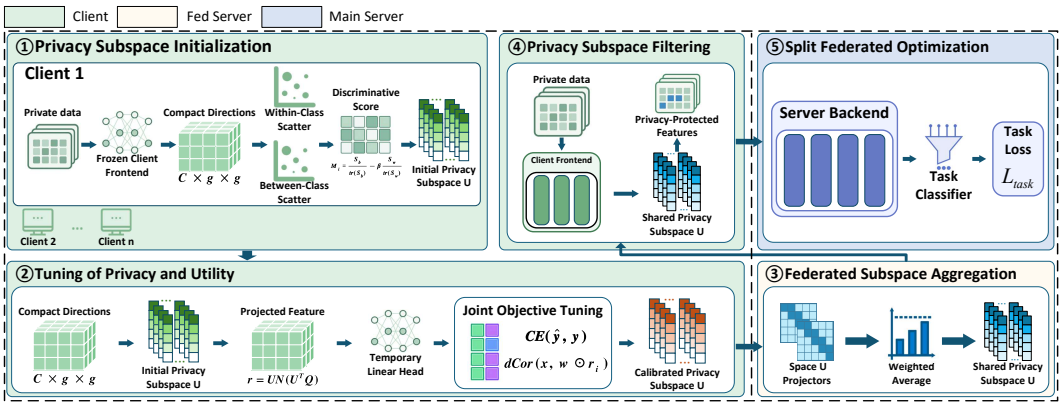


Figure 1: Overview of the proposed LOPA.

and μ_s denote the class-conditional mean and global mean at position s, respectively. We formulate the between-class scatter matrix S_b and within-class scatter matrix S_w as:

$$S_b = \frac{1}{nS}\sum_{c} n_c \sum_{s=1}^{S}(\mu_{c,s}-\mu_s)(\mu_{c,s}-\mu_s)^\top,$$
$$S_w = \frac{1}{nS}\sum_{i=1}^{n}\sum_{s=1}^{S}(q_{i,s}-\mu_{y_i,s})(q_{i,s}-\mu_{y_i,s})^\top, \tag{1}$$

where y_i is the label of sample i. Specifically, S_b characterizes the dispersion among class centroids, while S_w quantifies intra-class feature variance; computing statistics across individual spatial locations prevents layout information loss incurred by global spatial averaging. In practice, sufficient statistics are accumulated in a single pass, and S_w is computed efficiently as the difference between the total scatter and between-class scatter matrices.

To balance scales, both scatter matrices undergo trace normalization to form a scoring matrix:

$$M = \frac{S_b}{\max(\mathrm{tr}(S_b),\epsilon)} - \lambda\frac{S_w}{\max(\mathrm{tr}(S_w),\epsilon)}, \tag{2}$$

where ϵ is introduced for numerical stability, and $\lambda = 0.1$ controls the penalty strength on intra-class scatter. We select the orthonormal eigenvectors corresponding to the top $k = 32$ eigenvalues of M to form $U_0 \in \mathbb{R}^{C\times k}$.

4.2 Tuning of Privacy and Utility

Starting from the initial basis matrix U_0, we leverage locally cached images, pooled features, and labels (x_i, Q_i, y_i) to further adapt the subspace to balance task utility and privacy. The client feature extractor remains frozen, updating only the basis matrix U and a linear classification head.

We project Q_i onto the subspace, apply position-wise normalization to the projection coefficients, and map them back to the original channel space via U, yielding the representation $V_i \in \mathbb{R}^{C\times g\times g}$. Taking the flattened V_i as input, the auxiliary classification head preserves class-discriminative information via cross-entropy loss. Concurrently, the distance correlation between the input and the representation serves as a privacy regularizer to mitigate their statistical dependence.

To align with the distance metric structure of the actually transmitted representation, we weight positions by the spatial replication count of nearest-neighbor upsampling: $\overline{V}_{i,:,s} = \sqrt{m_s/\overline{m}}\,V_{i,:,s}$, where m_s denotes the replication count at position s and $\overline{m}$ is its mean. The joint optimization objective is formulated as:

$$\min_{U,\phi} \mathcal{L}_{\mathrm{CE}} + \beta \, \mathrm{dCor}\left(\{x_i\}_{i=1}^{B}, \{\overline{V}_i\}_{i=1}^{B}\right), \qquad \text{s.t.} \quad U^\top U = I_k, \tag{3}$$

where ϕ denotes the parameters of the classification head, B is the batch size, and $\beta = 0.02$ governs the privacy constraint strength. The classification head operates on the unweighted V_i, whereas spatial weighting is applied exclusively to the distance correlation computation.

We first warm up the classification head for 32 steps while keeping $U = U_0$ fixed, followed by 32 steps of joint updates for both U and the classification head. After each joint update step, QR decomposition is performed on U to restore its orthonormality. Upon completing fine-tuning, the auxiliary classification head is discarded, and the adapted local basis matrix is retained for subsequent federated subspace aggregation.

4.3 Federated Subspace Aggregation

Upon completing local fine-tuning, each client obtains an orthonormal basis matrix $U_i \in \mathbb{R}^{C\times k}$. Leveraging the federated server in SFL, we aggregate the projection matrices that uniquely characterize these subspaces:

$$\overline{P} = \sum_{i=1}^{N} \frac{n_i}{\sum_{j=1}^{N} n_j} U_i U_i^\top, \tag{4}$$

where N denotes the number of clients, and n_i is the number of local samples at client i. The server then extracts the orthonormal eigenvectors corresponding to the top k eigenvalues of $\overline{P}$ to construct the shared global basis matrix $U_\star$.

4.4 Privacy Subspace Filtering

During online training, each client utilizes the frozen global basis matrix $U_\star$ to filter intermediate features. For an incoming input, the client-side model generates feature representations Z, which are processed into Q via channel normalization, partial spatial centering, and average pooling, following the identical preprocessing pipeline in the initialization stage. Subsequently, subspace projection is applied to synthesize the uploaded representation:

$$\widetilde{Z} = \mathrm{Up}_{H\times W}\left[U_\star \mathcal{N}(U_\star^\top Q)\right] + \delta, \tag{5}$$

where $\mathcal{N}(\cdot)$ denotes position-wise ℓ_2 normalization, Up represents nearest-neighbor upsampling, and $\delta = 1.0001$ ensures non-negative encoding. This procedure suppresses the transmission of out-of-subspace components and dampens magnitude as well as fine-grained spatial details, while maintaining the transmitted feature dimensions at $C \times H \times W$. The filtered representation $\widetilde{Z}$ is subsequently forwarded to the server to complete the downstream task computation.

4.5 Split Federated Optimization

The server receives the filtered representation $\widetilde{Z}$, conducts downstream prediction, computes the task loss, and updates the server-side model before backpropagating the gradients at the cut layer back to the client. The client then propagates these gradients through the fixed filter to update its local model parameters. At the end of each round, the federated server aggregates the client-side model parameters via sample-weighted averaging and broadcasts the updated model for the subsequent training round. Concurrently, the task server maintains a continuously updated shared model without performing server-side model aggregation.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETTINGS

Datasets. We conduct our experiments on three widely adopted (Kulkarni et al., 2025; Yi et al., 2025) datasets: CelebA (Liu et al., 2015), CIFAR-10 (Krizhevsky et al., 2009), and FMNIST (Xiao et al., 2017).

Metrics. To verify that our method preserves primary task performance, we employ classification accuracy as the evaluation metric. To assess its privacy-preserving capability, we utilize Learned Perceptual Image Similarity (LPIPS) (Zhang et al., 2018), Structural Similarity (SSIM) (Wang et al., 2004), Peak Signal-to-Noise Ratio (PSNR) (Hore & Ziou, 2010), and Mean Squared Error (MSE) to quantify the reconstruction quality of client-side images. Furthermore, for the CelebA face dataset, we leverage ID similarity[1] to evaluate identity leakage risks. These metrics are widely used in prior work (Xu et al., 2024; Qian et al., 2025).

Baselines. We compare LOPA with three SL baselines, namely SFL (Thapa et al., 2022), Dropout (He et al., 2020), and B-SL (Pham et al., 2023), in terms of classification accuracy and privacy protection, and with the federated learning baseline AdapLDP-FL (Yue et al., 2025) in terms of classification accuracy. AdapLDP-FL is excluded from the privacy evaluation because it does not transmit smashed data. Furthermore, we evaluate reconstruction privacy using FORA (Xu et al., 2024) and Inverse-Network (He et al., 2019). The privacy preservation performance of LOPA against FORA is detailed in Section 5.3, while its defense efficacy under Inverse-Network attacks is evaluated in Appendix A.2.1.

Implementation Details. To ensure the attack efficacy, we adopt the same backbone architectures as FORA (Xu et al., 2024), namely MobileNet (Howard et al., 2017). Furthermore, we follow an identical model partitioning strategy that divides the network into layers 1 through 4. Then, layers 1 and 2 are deployed on the client side, while layers 3 and 4 are assigned to the server side. We partition the training sets across three clients using a label-stratified Dirichlet distribution with $\alpha = 1.0$. All experiments are repeated across three random seeds, and we report the average results.

Table 1: Accuracy (%) comparison of LOPA and baselines.

Method	CelebA	CIFAR-10	FMNIST
SFL	89.02	66.78	89.47
Dropout	85.57	55.88	82.68
AdapLDP-FL	88.57	66.82	89.10
B-SL	87.68	56.68	87.58
LOPA	**89.78**	**67.56**	**89.70**

5.2 EFFECTIVENESS OF LOPA FOR CLASSIFICATION TASK

Existing defenses reduce reconstruction risks by limiting information exposure, but may also remove information needed for classification. Although structural bottlenecks can degrade reconstruction quality, random selection of subspace directions does not preserve classification accuracy across all datasets. To assess whether LOPA can maintain classification utility while protecting uploaded representations, we evaluate its classification performance on CelebA, CIFAR-10, and FMNIST, using the Smiling attribute as the classification target for CelebA. The experimental results are presented in Table 1. LOPA achieves the highest classification accuracy among the compared methods on all three datasets, consistently outperforming undefended SFL. A plausible explanation is that initialization based on between-class and within-class scatter statistics helps identify directions that favor class separation, while the classification loss during calibration further guides the subspace toward preserving task-discriminative information.

[1] https://github.com/timesler/facenet-pytorch.git

[2] Since CIFAR-10 and FMNIST do not involve facial identities, the ID Similarity metric is not applicable to these datasets.

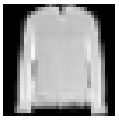
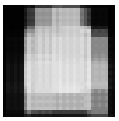
Table 2: Comparison of defensive performance against the FORA between LOPA and baselines

Dataset	Method	ID Similarity [2] ↓	LPIPS ↑	SSIM ↓	PSNR ↓	MSE ↑
CelebA	SFL	0.595	0.195	0.867	22.65	0.006
	Dropout	0.429	0.340	0.796	20.65	0.009
	B-SL	0.148	0.554	0.594	15.58	0.029
	LOPA	**0.110**	**0.624**	**0.403**	**14.86**	**0.035**
CIFAR-10	SFL	–	0.226	0.810	23.16	0.005
	Dropout	–	0.360	0.685	20.09	0.011
	B-SL	–	0.446	0.617	16.99	0.023
	LOPA	–	**0.556**	**0.341**	**15.78**	**0.029**
FMNIST	SFL	–	0.110	0.831	25.04	0.003
	Dropout	–	0.272	0.697	21.49	0.008
	B-SL	–	0.297	0.571	15.64	0.031
	LOPA	–	**0.437**	**0.373**	**15.08**	**0.034**

5.3 Effectiveness of LOPA for Client Data Privacy Protection

We train FORA concurrently with SFL and evaluate privacy protection through the quality of its reconstructed images. As shown in Table 2, LOPA provides competitive protection across all three datasets. Specifically, LOPA reduces identity similarity on CelebA from 0.595 under undefended SFL to 0.110, indicating a substantial decline in identity fidelity. Moreover, LPIPS, SSIM, PSNR, and MSE across the three datasets indicate that FORA reconstructs lower-quality images when attacking LOPA than when attacking undefended SFL and the other baselines. These results demonstrate that LOPA effectively mitigates image reconstruction risks under the evaluated FORA setting. This effectiveness may stem from the structural bottleneck jointly imposed by low-rank channel projection, normalization, and spatial pooling, which limits reconstruction-relevant information in uploaded representations.

Table 3: Visualization of FORA attack reconstructed images

	FMNIST		CIFAR-10		CelebA	
Ground Truth		-		-		-
SFL		SSIM: 0.84 PSNR: 25.5 LPIPS: 0.11		SSIM: 0.89 PSNR: 21.4 LPIPS: 0.23		SSIM: 0.93 PSNR: 24.3 LPIPS: 0.11
B-SL		SSIM: 0.54 PSNR: 17.1 LPIPS: 0.32		SSIM: 0.78 PSNR: 16.5 LPIPS: 0.38		SSIM: 0.72 PSNR: 14.8 LPIPS: 0.47
LOPA		SSIM: 0.32 PSNR: 13.6 LPIPS: 0.43		SSIM: 0.40 PSNR: 13.6 LPIPS: 0.60		SSIM: 0.51 PSNR: 14.8 LPIPS: 0.60

We also employ the FORA model trained concurrently with each baseline to reconstruct one correctly classified image selected from each of the three datasets. The reconstruction results are shown in Table 3. The first row displays the raw images, the second row corresponds to the reconstructed images from SFL, the third row shows those from B-SL, and the fourth row presents the results from LOPA. The reconstruction metrics are displayed to the right of each reconstructed image. Visually, the reconstruction quality for LOPA is noticeably worse.

Table 4: Ablation study of class-aware initialization and local fine-tuning of the privacy subspace U

Dataset	Method	ACC ↑	ID Similarity [3] ↓	LPIPS ↑	SSIM ↓	PSNR ↓	MSE ↑
CelebA	SFL	89.02	0.595	0.195	0.867	22.65	0.006
	$\text{Model}_{\text{woU}}$	88.48	0.106	0.631	0.401	14.77	0.035
	$\text{Model}_{\text{woT}}$	87.94	0.120	0.618	0.403	14.82	0.034
	$\text{Model}_{\text{woUT}}$	89.03	0.107	0.623	0.411	14.92	0.034
	LOPA	89.78	0.110	0.624	0.403	14.86	0.035
CIFAR-10	SFL	66.78	–	0.226	0.810	23.16	0.005
	$\text{Model}_{\text{woU}}$	66.41	–	0.554	0.334	15.66	0.029
	$\text{Model}_{\text{woT}}$	67.21	–	0.570	0.340	15.97	0.028
	$\text{Model}_{\text{woUT}}$	67.12	–	0.546	0.370	16.45	0.025
	LOPA	67.56	–	0.556	0.341	15.78	0.029
FMNIST	SFL	89.47	–	0.110	0.830	25.04	0.003
	$\text{Model}_{\text{woU}}$	89.62	–	0.431	0.364	15.15	0.034
	$\text{Model}_{\text{woT}}$	89.55	–	0.437	0.369	15.50	0.031
	$\text{Model}_{\text{woUT}}$	89.04	–	0.432	0.361	15.28	0.033
	LOPA	89.70	–	0.437	0.373	15.08	0.034

6 Ablation Study

To dissect the individual contributions of LOPA's core components to privacy preservation and task utility, we construct three ablation variants: (1) $\text{Model}_{\text{woU}}$, which replaces the class-statistics-based subspace initialization with random orthogonal bases; (2) $\text{Model}_{\text{woT}}$, which omits the joint fine-tuning stage and directly aggregates the initialized local subspaces; and (3) $\text{Model}_{\text{woUT}}$, which initializes with random orthogonal bases while skipping the fine-tuning phase altogether. All variants are evaluated under identical experimental settings to LOPA.

Table 4 shows that even without class-statistics-based initialization and joint calibration, ModelwoUT substantially reduces reconstruction quality relative to undefended SFL and achieves reconstruction resistance comparable to that of full LOPA. For example, their LPIPS scores on CelebA are 0.623 and 0.624, respectively. This indicates that the structural bottleneck formed by low-rank channel projection, normalization, and spatial pooling can effectively hinder image reconstruction, and that this effect does not entirely depend on task-relevant subspace selection. However, random subspaces do not preserve classification utility across all datasets. On FMNIST, ModelwoUT achieves an accuracy of 89.04%, below the 89.47% achieved by undefended SFL. Full LOPA outperforms $\text{Model}_{\text{woUT}}$ by 0.75, 0.44, and 0.66 percentage points on CelebA, CIFAR-10, and FMNIST, respectively, and exceeds undefended SFL on all three datasets. A plausible explanation is that class-statistics-based initialization helps select directions that favor class separation, while joint calibration preserves task-discriminative information through the classification loss and reduces statistical dependence between representations and raw inputs through the distance correlation constraint.

7 Conclusion

We propose LOPA, an SFL privacy defense for clients that aims to mitigate image reconstruction risks from uploaded representations while preserving classification utility. LOPA combines low-rank channel projection with normalization and spatial pooling. Using only local client data, it selects task-relevant subspace directions through class-statistics-based initialization and joint calibration, and constructs a shared subspace by aggregating orthogonal projection matrices. This subspace is determined before the first upload and remains fixed during online training, enabling the complete filtering mechanism to operate from the outset while avoiding the overhead of continuous subspace optimization. Experiments on three datasets demonstrate that LOPA effectively reduces reconstruction quality under both FORA and Inverse-Network attacks while achieving classification accuracy no lower than undefended SFL across all three datasets.

AI USE STATEMENT

In this work, we used generative AI tools for assisting with translation. We have not used generative AI tools for helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, implementing methods, cleaning and reformatting datasets, supporting qualitative and thematic data analysis, or interpreting results, and tasks involving the generation of synthetic data sets are not applicable to this work. Additionally, we used generative AI tools for sourcing/searching for information. We have reviewed all AI-assisted work. We have confirmed the accuracy of the translated content and the collected information. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Our code is available at `https://anonymous.4open.science/r/LOPA-221C`

REFERENCES

Xinben Gao and Lan Zhang. Pcat: Functionality and data stealing from split learning by pseudo-client attack. In *USENIX Security Symposium*, pp. 5271–5288, 2023.

Otkrist Gupta and Ramesh Raskar. Distributed learning of deep neural network over multiple agents. *Journal of Network and Computer Applications*, 116:1–8, 2018.

Zecheng He, Tianwei Zhang, and Ruby B Lee. Model inversion attacks against collaborative inference. In *Annual Computer Security Applications Conference*, pp. 148–162, 2019.

Zecheng He, Tianwei Zhang, and Ruby B Lee. Attacking and protecting data privacy in edge–cloud collaborative inference systems. *IEEE Internet of Things Journal*, 8(12):9706–9716, 2020.

Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *International Conference on Pattern Recognition*, pp. 2366–2369, 2010.

Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

Taejin Kim, Jinhang Zuo, Xiaoxi Zhang, and Carlee Joe-Wong. Edge-msl: Split learning on the mobile edge via multi-armed bandits. In *IEEE International Conference on Computer Communications*, pp. 391–400, 2024.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Akshay Kulkarni, Ge Yan, Chung-En Sun, Tuomas Oikarinen, and Tsui-Wei Weng. Interpretable generative models through post-hoc concept bottlenecks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8162–8171. IEEE, 2025.

Haifeng Li, Tao Jiang, and Keshu Zhang. Efficient and robust feature extraction by maximum margin criterion. *IEEE Transactions on Neural Networks*, 17(1):157–165, 2006.

Jingtao Li, Adnan Siraj Rakin, Xing Chen, Zhezhi He, Deliang Fan, and Chaitali Chakrabarti. Ressfl: A resistance transfer framework for defending model inversion attack in split federated learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10184–10192, 2022a.

Oscar Li, Jiankai Sun, Xin Yang, Weihao Gao, Hongyi Zhang, Junyuan Xie, Virginia Smith, and Chong Wang. Label leakage and protection in two-party split learning. In *International Conference on Learning Representations*, pp. 1–27, 2022b.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *IEEE International Conference on Computer Vision*, pp. 3730–3738, 2015.

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

Andrew Paverd, Andrew Martin, and Ian Brown. Modelling and automatically analysing privacy properties for honest-but-curious adversaries. Technical report, Tech. rep, 2014.

Ngoc Duy Pham, Alsharif Abuadbba, Yansong Gao, Khoa Tran Phan, and Naveen Chilamkurti. Binarizing split learning for data privacy enhancement and computation reduction. *IEEE Transactions on Information Forensics and Security*, 18:3088–3100, 2023.

Guocheng Qian, Kuan-Chieh Wang, Or Patashnik, Negin Heravi, Daniil Ostashev, Sergey Tulyakov, Daniel Cohen-Or, and Kfir Aberman. Omni-id: Holistic identity representation designed for generative tasks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8786–8795, 2025.

Xicong Shen, Yang Liu, Yi Liu, Peiran Wang, Huiqi Liu, Jue Hong, Bing Duan, Zirui Huang, Yunlong Mao, Ye Wu, and Sheng Zhong. Sap: Privacy-preserving fine-tuning on language models with split-and-privatize framework. In *International Joint Conference on Artificial Intelligence*, pp. 502–510, 2025.

Yuwei Sun and Hideya Ochiai. Bidirectional contrastive split learning for visual question answering. In *AAAI Conference on Artificial Intelligence*, volume 38, pp. 21602–21609, 2024.

Chandra Thapa, Pathum Chamikara Mahawaga Arachchige, Seyit Camtepe, and Lichao Sun. Splitfed: When federated learning meets split learning. In *AAAI Conference on Artificial Intelligence*, volume 36, pp. 8485–8493, 2022.

Joana Tirana, Dimitra Tsigkari, George Iosifidis, and Dimitris Chatzopoulos. Workflow optimization for parallel split learning. In *IEEE International Conference on Computer Communications*, pp. 1331–1340, 2024.

Praneeth Vepakomma, Abhishek Singh, Otkrist Gupta, and Ramesh Raskar. Nopeek: Information leakage reduction to share activations in distributed deep learning. In *International Conference on Data Mining Workshops*, pp. 933–942, 2020.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Xiaoyang Xu, Mengda Yang, Wenzhe Yi, Ziang Li, Juan Wang, Hongxin Hu, Yong Zhuang, and Yaxin Liu. A stealthy wrongdoer: Feature-oriented reconstruction attack against split learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12130–12139, 2024.

Liping Yi, Han Yu, Gang Wang, Xiaoguang Liu, and Xiaoxiao Li. Federated representation angle learning. In *IEEE/CVF International Conference on Computer Vision*, pp. 1314–1324. IEEE, 2025.

Gaofeng Yue, Li Yan, Liuwang Kang, and Chao Shen. Adapldp-fl: An adaptive local differential privacy for federated learning. *IEEE Transactions on Mobile Computing*, 24(6):5569–5583, 2025.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.

A Appendix

A.1 Experimental Setup Details

A.1.1 Datasets

We elaborate on the three datasets used in our experiments.

CelebA. CelebA is a facial attribute dataset that provides 202,599 face images of 10,177 celebrities, with 40 attribute annotations available per image. Before being fed into the model, the images are resized to a resolution of 3×64×64.

CIFAR-10. CIFAR-10 is an image classification benchmark consisting of 60,000 color images of size 32×32 pixels across 10 categories. The dataset is partitioned into a training set of 50,000 images and a test set of 10,000 images, with each class containing an equal number of samples.

FMNIST. Fashion-MNIST is an image classification benchmark consisting of 70,000 grayscale images of size 28×28 pixels across 10 categories of fashion items. The dataset is partitioned into a training set of 60,000 images and a test set of 10,000 images, with each class containing an equal number of samples.

A.1.2 More setups of LOPA

In addition to the setups mentioned in the main paper, we provide more setups of LOPA here. During the tuning of privacy and utility, we train a temporary linear head using the Adam optimizer, with a learning rate of 0.001 and a weight decay of 0.0001. We also optimize the subspace with Adam using a learning rate of 0.0001. Finally, task training is conducted with the Adam optimizer for 30 epochs. It uses a batch size of 64, a learning rate of 0.001, and a momentum of 0.9. The dimension of the private subspace is set to $k = 32$.

A.2 Additional Experiments

A.2.1 Effectiveness of LOPA in Preserving Privacy Against Inverse-Network

In this section, we further employ the Inverse-Network attack (He et al., 2019) to evaluate the privacy protection provided by LOPA under different reconstruction attack mechanisms. Experiments are conducted on FMNIST, CIFAR-10, and CelebA, using the same MobileNet architecture as in the main experiments, with the model split at level2. Inverse-Network keeps the trained client-side model and its defense module fixed and separately learns an inverse mapping from transmitted features to input images. Specifically, we assume that the main server has access to a set of auxiliary images drawn from the same distribution, excluding the target images to be reconstructed. The main server can feed these auxiliary images into the client-side model and obtain the features actually transmitted to the server after the corresponding defense is applied, thereby constructing paired auxiliary images and transmitted features for training. We divide 5,000 auxiliary images into 4,500 training images and 500 validation images, and evaluate reconstruction performance on a separate, fixed set of 2,000 images that does not overlap with the auxiliary pool. This experiment assesses whether LOPA can still reduce reconstruction quality when the server learns a reconstruction mapping for a fixed model.

Table 5 compares the privacy protection provided by LOPA and the baseline methods against Inverse-Network attacks. As shown, LOPA achieves the best results across all reported metrics. Its LPIPS scores on CelebA, CIFAR-10, and FMNIST are 0.582, 0.509, and 0.465, respectively. The corresponding SSIM scores are 0.456, 0.431, and 0.396. On the same datasets, LOPA yields PSNR values of 16.34, 17.97, and 16.37, and MSE values of 0.025, 0.018, and 0.025, respectively. On CelebA, LOPA also achieves an identity similarity score of 0.144, indicating reduced identity leakage. A possible explanation is that LOPA's subspace projection suppresses texture, spatial detail, and magnitude information in the transmitted representations. Consequently, even when an attacker trains an inverse network using auxiliary images and their corresponding genuine smashed data, accurately reconstructing the original images and recovering identity-related features remains difficult. These results demonstrate that LOPA provides effective privacy protection even when the attacker has access to genuine smashed data from an auxiliary dataset.

[3] Since CIFAR-10 and FMNIST do not involve facial identities, the ID Similarity metric is not applicable to these datasets.

[4] ID Similarity is not applicable to CIFAR-10 and FMNIST because these datasets do not involve facial identities.

Table 5: Comparison of defensive performance against the Inverse-Network attack between LOPA and baselines

Dataset	Method	ID Similarity[4] ↓	LPIPS ↑	SSIM ↓	PSNR ↓	MSE ↑
CelebA	SFL	0.780	0.107	0.931	26.91	0.002
	Dropout	0.604	0.210	0.878	23.90	0.004
	B-SL	0.319	0.388	0.778	20.19	0.010
	LOPA	**0.144**	**0.582**	**0.456**	**16.34**	**0.025**
CIFAR-10	SFL	–	0.135	0.902	26.95	0.002
	Dropout	–	0.229	0.841	24.35	0.004
	B-SL	–	0.303	0.757	21.98	0.007
	LOPA	–	**0.509**	**0.431**	**17.97**	**0.018**
FMNIST	SFL	–	0.068	0.947	30.29	0.001
	Dropout	–	0.146	0.886	26.66	0.002
	B-SL	–	0.193	0.807	23.31	0.005
	LOPA	–	**0.465**	**0.396**	**16.37**	**0.025**

Table 6: Comparison of computational overhead between client and privacy subspace U

Dataset	Time (s)		Peak Allocated Memory (MiB)	
	Client	Privacy Subspace U	Client	Privacy Subspace U
CelebA	207.19	5.16	144.80	135.44
CIFAR-10	72.81	1.83	144.80	135.49
FMNIST	105.93	2.67	144.30	134.82

A.2.2 Cost Analysis

Since both the subspace initialization and joint fine-tuning in LOPA are conducted entirely on local clients, we further evaluate the computational overhead and memory footprint of this pipeline to assess its feasibility on resource-constrained client devices. As shown in Table 6, the initialization time of the privacy subspace U varies with the dataset scale, taking up to 5.16 seconds, which accounts for only 2.5% of the total training time on the corresponding client. Across all datasets, the peak GPU memory footprint of initialization is lower than that of the corresponding client training process, reaching a maximum of 135.49 MiB. These results indicate that subspace initialization introduces low computational overhead and can be executed locally on clients.