Rehearsal-Free Statistical Prototype Regularization for Federated Incremental Learning

Xiuying Wang[1], **Yichen Li**[2], **Jiahua Cheng**[1], **Xiwei Liu**[3],
Chuang Zhao[4], **Imran Razzak**[3], **Bo Liu**[5*]
[1]International School, Beijing University of Posts and Telecommunications, Beijing, China
[2]School of Computer Science and Technology,
Huazhong University of Science and Technology, Wuhan, China
[3]Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, United Arab Emirates
[4]Tianjin University, Tianjin, China
[5]School of Computer Science and Artificial Intelligence,
Shenzhen University of Advanced Technology, Shenzhen, China
leowang980@bupt.edu.cn, liubo@suat-sz.edu.cn

Abstract

Federated Incremental Learning (FIL) aims to learn sequential tasks across distributed clients under strict privacy constraints while mitigating catastrophic forgetting. However, existing methods often rely on storing, transmitting, or generating previous data to retrain the model, which not only introduces potential privacy risks but also incurs additional memory, communication, and computational costs, thereby limiting their deployment on resource-constrained edge devices. To tackle this challenge, we propose *SPR-FIL*, a novel rehearsal-free and lightweight Statistical Prototype Regularization framework for Federated Incremental Learning that leverages Gaussian statistics and preserves previous knowledge through aligned prototypes and subspace-constrained encoder updates. More specifically, the server aggregates client-side statistics into global prototypes and dynamically aligns historical prototypes with the evolving feature space through anchor-based moment matching. Locally, clients utilize these aligned prototypes to induce virtual posteriors which serve as soft supervision for distillation while encoder updates are further regularized by projecting gradients onto the orthogonal complement of covariance-induced historical knowledge subspaces. Extensive experiments demonstrate that SPR-FIL is model-agnostic, communication-efficient, and consistently outperforms state-of-the-art methods across various FIL settings, achieving improvements of up to 9.73%.

1 Introduction

Federated Learning (FL) [31, 18, 17] has become a pivotal paradigm in privacy-sensitive domains such as healthcare [49] and finance [1, 3]. It enables multiple distributed clients to collaboratively train a shared model without directly exposing their raw data, thereby reducing privacy risks and communication overhead associated with centralized data collection.

However, real-world scenarios in these domains are rarely static. Data streams often arrive sequentially, and the underlying class distributions may continuously evolve as new diseases, user behaviors, or financial patterns emerge [12, 13]. This dynamic setting gives rise to Federated Incremental Learning (FIL), where the global model must continuously adapt to new tasks while preserving

*Corresponding author.

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

previously acquired knowledge. The key challenge is therefore to maintain a favorable balance between learning new concepts and mitigating catastrophic forgetting of old ones [30].

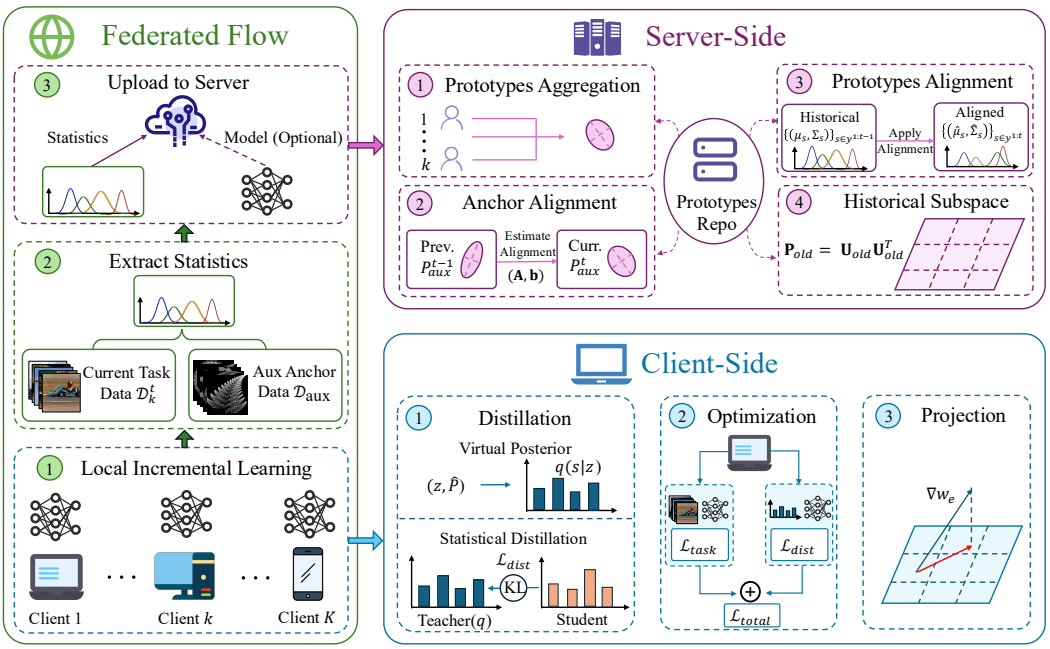


Figure 1: The SPR-FIL Framework. (i) The left side illustrates the federated flow, where clients perform local incremental learning, extract statistics from current-task data and auxiliary anchor data, and upload only statistics to the server. (ii) The upper-right side depicts server-side prototype aggregation, statistical space alignment, and historical subspace generation. (iii) The lower-right side shows client-side local training with dual constraints, including virtual posterior distillation and feature-gradient projection.

Extensive rehearsal methods have been explored in FIL. One prevalent strategy involves storing a limited number of representative samples from previous tasks for rehearsal. ReFed and its extended version ReFed+ [20, 21] propose a synergistic rehearsal framework that selects globally and locally important samples for rehearsal. Other methods avoid explicit data storage by generating pseudo-samples from earlier tasks. For example, FedCIL [35] employs generative models to synthesize historical data for rehearsal, while the authors in [39] further improve the quality of generation by leveraging client expertise and aligning them with task-aware representations.

While effective, these methods fundamentally undermine the privacy and efficiency principles of FL. In healthcare scenarios, rehearsal buffers or generators trained on historical records may implicitly encode sensitive clinical patterns, thereby violating data protection regulations such as the GDPR and HIPAA [49, 34, 9], which strictly regulate the retention and use of sensitive patient data. Moreover, the substantial storage and computational overheads required conflict directly with the resource-constrained nature of edge devices.

Reflecting on these limitations, an appealing alternative is to revisit regularization-based Incremental Learning (IL). However, standard methods (e.g. EWC [11], LwF [27]) are ill-suited for FIL. The heterogeneity inherent in decentralized data causes local estimates of parameter importance to diverge significantly from the global objective. Furthermore, the stochastic nature of partial client participation disrupts the aggregation process, leading to inconsistent regularization terms that fail to effectively constrain the model. Seeking a more representative proxy for global class distributions, we turn to statistical prototypes derived from the encoder's output, capturing the semantic essence of classes without sharing raw data. Additionally, their model-agnostic nature makes them particularly appealing for heterogeneous settings. However, applying them to FIL entails two fundamental hurdles. First, feature space drift causes historical prototypes derived from previous encoders to become misaligned with the current feature space in the incremental process, rendering them obsolete.

Second, effectively utilizing these prototypes requires designing a composite mechanism to translate statistical prototypes into supervision signals for both distillation and geometric constraints, thereby mitigating the catastrophic forgetting of historical knowledge.

To this end, we propose **SPR-FIL**, a novel rehearsal-free statistical prototype regularization framework tailored for privacy-preserving and resource-efficient FIL. On the server side, SPR-FIL maintains a global repository of lightweight Gaussian prototypes (comprising means and covariances) aggregated from clients' local feature statistics. To address feature space drift caused by model updates, the server employs a statistical space alignment mechanism, leveraging local statistics from a shared auxiliary anchor set, to linearly align these historical prototypes with the current feature space, and further extracts principal directions to construct a global historical knowledge subspace. On the client side, devices utilize the aligned prototypes to construct virtual posterior distributions for statistical distillation. Additionally, clients further constrain encoder updates by projecting gradients onto the orthogonal complement of the covariance-induced subspace. Through extensive experiments on various datasets in both homogeneous and heterogeneous settings, we show that SPR-FIL outperforms state-of-the-art approaches. We summarize our main contributions as:

- We identify critical privacy risks and resource limitations in rehearsal-based FIL, and reveal the inadequacy of existing regularization-based approaches.
- We propose SPR-FIL, a rehearsal-free framework utilizing aligned statistical prototypes. It preserves historical knowledge via server-side statistical space alignment and client-side dual constraints—virtual posterior distillation and orthogonal gradient projection—effectively balancing plasticity and stability.
- Extensive experiments on standard benchmark datasets demonstrate that the proposed model-agnostic SPR-FIL achieves state-of-the-art performance, outperforming existing methods by up to 9.73%.

2 Related Work

Federated Learning. FL is a distributed learning framework in which multiple clients jointly train a global model while keeping raw data local, typically coordinated through aggregation methods such as FedAvg [31]. To improve the efficiency and reliability of FL systems, prior studies have explored various techniques such as knowledge distillation [28, 23], model compression [7], gradient sparsification [50], and quantization [33] to reduce communication overhead. In addition, extensive efforts have been devoted to enhancing privacy, security, and robustness through approaches including heterogeneity-aware learning [47, 14], differential privacy [32, 29], secure aggregation [42], and adversarial training [19]. However, existing FL approaches generally assume static tasks and fixed data distributions, which limits their applicability in scenarios involving continuously evolving data streams.

Incremental Learning. IL aims to continuously learn from streaming data while mitigating catastrophic forgetting of previously acquired knowledge [30]. IL scenarios are generally categorized into Task-IL, Domain-IL, and Class-IL [44], depending on the availability of task identities and the nature of distribution shifts. To address catastrophic forgetting, existing IL methods can be broadly categorized into several representative families, including replay-based methods that retain past knowledge through exemplars [37, 38, 46], regularization-based methods that constrain updates to important parameters [11, 27, 45, 26], parameter-isolation approaches that allocate dedicated parameters for different tasks [6, 41], and architecture-expansion methods that dynamically enlarge the model capacity for new tasks [40, 4]. Despite their effectiveness, most of these approaches are designed for centralized settings, and their adaptation to federated environments remains challenging due to decentralized data distributions, privacy constraints, and communication overhead.

Federated Incremental Learning. FIL extends the static paradigm of FL [31] to dynamic environments, aiming to mitigate catastrophic forgetting while upholding privacy [22]. Early FIL frameworks primarily targeted the simpler Task-IL setting; for instance, [51] and [2] employed parameter isolation techniques, such as parameter decomposition with selective task-specific transfer or orthogonal subspaces, to prevent interference between distinct tasks. To address the more challenging Class-IL scenarios where task boundaries are ambiguous, recent research has diverged into two main streams. Rehearsal-based approaches [25, 20, 21, 35, 39] store or synthesize historical samples to approximate

the joint distribution. However, these methods inherently conflict with the resource constraints of edge devices and introduce potential privacy leaks. Conversely, regularization-based methods [24] constrain updates to important parameters or representation directions to preserve previous knowledge, though they often suffer from performance degradation due to non-IID data. In response, our work focuses on developing a statistical prototype regularization framework for FIL.

3 Methodology

3.1 Problem Formulation

We consider the FIL setting involving a central server and a set of K distributed clients, denoted as $\mathcal{K} = \{1, \ldots, K\}$. The learning process is organized into a sequence of T non-overlapping tasks. At any given task phase t, the k-th client holds a private local dataset $\mathcal{D}_k^t = \{(x_i, y_i)\}_{i=1}^{N_k^t}$ with size N_k^t, where $x \in \mathcal{X}$ represents the input sample and $y \in \mathcal{Y}^t$ denotes the corresponding label. Consistent with the Class-IL paradigm, the label space expands over time, such that $\mathcal{Y}^t \cap \mathcal{Y}^{t'} = \emptyset$ for $t \neq t'$. To comply with strict privacy regulations and resource constraints on edge devices, clients are prohibited from storing raw data from previous tasks $\{\mathcal{D}_k^j\}_{j=1}^{t-1}$ or sharing local data with the server.

The local model w_k^t comprises an encoder $w_{e,k}^t : \mathcal{X} \to \mathbb{R}^d$ and a classifier $w_{c,k}^t : \mathbb{R}^d \to \mathcal{Y}^{1:t}$. The fundamental challenge in FIL is the plasticity-stability dilemma: the model must update parameters $w_k^t = \{w_{e,k}^t, w_{c,k}^t\}$ to minimize the empirical risk on the current task while preserving the discriminative capability on all historical classes $\mathcal{Y}^{1:t-1}$. Formally, the ideal optimization objective is:

$$\min_{\{w_k^t\}_{k=1}^K} \underbrace{\sum_{j=1}^{t} \sum_{k=1}^{K} \frac{N_k^j}{\sum_{j=1}^{t} \sum_{k=1}^{K} N_k^j} \mathcal{L}_{CE}(w_k^t; \mathcal{D}_k^j)}_{\text{Cumulative Empirical Risk}}. \tag{1}$$

where $\mathcal{L}_{CE}(w_k^t; \mathcal{D}_k^j)$ is the mean cross-entropy loss computed by the local model w_k^t over samples in $\mathcal{D}_k^j$.

3.2 SPR-FIL: A Rehearsal-free Statistical Prototype Regularization Framework

In this section, we present SPR-FIL, a privacy-preserving and lightweight framework designed to mitigate catastrophic forgetting without rehearsal. SPR-FIL maintains historical knowledge on the server via statistical prototypes (multivariate Gaussians) and introduces an auxiliary anchor mechanism to actively align historical prototypes against feature space drift. Leveraging these aligned prototypes, the framework imposes dual constraints during local client training: a statistical distillation loss to align decision boundaries and a subspace-based gradient projection to further constrain encoder updates. The overall workflow is detailed in Algorithm. 1 and Fig. 1.

Server-Side Prototypes Repository. The server functions as a repository of historical knowledge. Instead of storing raw samples, it maintains a statistical prototype for each observed class to respect privacy. Specifically, we assume the feature distribution of class s follows a multivariate Gaussian distribution $\mathcal{N}(\mu_s, \Sigma_s)$.

At the end of the local training phase of task t, each client k computes local statistics for the newly learned classes. Additionally, to track global feature space evolution, we introduce a shared auxiliary anchor set $\mathcal{D}_{\text{aux}}$. Unlike raw Gaussian noise, which lacks the structural complexity to effectively probe the encoder, we utilize FractalDB [10] generated via Iterated Function Systems (IFS). This formula-driven approach guarantees that $\mathcal{D}_{\text{aux}}$ is inherently privacy-free and consistent across all clients and tasks (via a fixed seed), while providing the rich geometric patterns necessary for meaningful feature alignment. Importantly, we do not use any real public image dataset as the auxiliary anchor set, which avoids potential information leakage from semantic overlap with the evaluation data and prevents unfair advantages from introducing additional real-world training data. Let $f(w_{e,k}^t; x) \in \mathbb{R}^d$ denote the extracted feature of sample x using the encoder $w_{e,k}^t$. For a real class $s \in \mathcal{Y}^t$ on client k and the

Algorithm 1 SPR-FIL: Server-Client Pipeline for Task $t+1$. Operations marked with † are optional and used only when global model aggregation is applicable.

Require: Previous global model w^t †; prototype repository $\mathcal{P}^t$; anchor statistics P^t_{aux}; auxiliary anchor set $\mathcal{D}_{\text{aux}}$; client datasets $\{D_k^{t+1}\}_{k=1}^K$; communication rounds R; local epochs E

Ensure: Updated global model w^{t+1} † and prototype repository $\mathcal{P}^{t+1}$

Server Initialization:
  Align historical prototypes $\mathcal{P}^t$ using P^t_{aux} by Eq. (4)–(5), yielding $\hat{\mathcal{P}}^t$
  Construct the historical subspace basis $\mathbf{U}_{\text{old}}$
for $r = 1$ to R **do**
  Server: Sample client subset $\mathcal{S}_r$ and broadcast $(w_r^\dagger, \hat{\mathcal{P}}^t, \mathbf{U}_{\text{old}})$

Server Side

1. Send global model $w_r^\dagger$ and aligned historical prototypes $\hat{\mathcal{P}}^t$.
2. Receive local model $w_k^\dagger$, new-class statistics $\mathcal{P}_k^{t+1}$, and anchor statistics $P_{\text{aux},k}^{t+1}$.
3. Aggregate client models into the next global model $w_{r+1}^\dagger$.

Client Side

1. Initialize local model with $w_r^\dagger$ and train on D_k^{t+1} for E epochs.
2. Apply prototype distillation with $\hat{\mathcal{P}}^t$ by Eq. (9).
3. Project feature gradients using $\mathbf{U}_{\text{old}}$ by Eq. (11).
4. Upload $w_k^\dagger$, $\mathcal{P}_k^{t+1}$, and $P_{\text{aux},k}^{t+1}$.

  Server: Update $w_{r+1}^\dagger$ by Eq. (12)
end for
Server Update:
  Aggregate $\{\mathcal{P}_k^{t+1}\}$ into new-class prototypes $\mathcal{P}_{\text{new}}^{t+1}$
  Aggregate $\{P_{\text{aux},k}^{t+1}\}$ and store updated anchor statistics P_{aux}^{t+1}
  Update prototype repository $\mathcal{P}^{t+1} \leftarrow \hat{\mathcal{P}}^t \cup \mathcal{P}_{\text{new}}^{t+1}$
Note: Operations marked with † are enabled in homogeneous settings; in heterogeneous settings, they can be omitted while the prototype and anchor-statistics pipeline remains unchanged.
return w^{t+1} † and $\mathcal{P}^{t+1}$

anchor set, we define the feature sets:

$$Z_{s,k} = \{f(w_{e,k}^t; x) \mid (x, y) \in \mathcal{D}_k^t,\ y = s\},$$
$$Z_{\text{aux},k} = \{f(w_{e,k}^t; x) \mid x \in \mathcal{D}_{\text{aux}}\}. \tag{2}$$

The client calculates the local mean $\mu_{s,k}$ and covariance $\Sigma_{s,k}$ for each newly learned class, and similarly computes $(\mu_{\text{aux},k}, \Sigma_{\text{aux},k})$ for the anchor set. The server subsequently aggregates these local statistics into global prototypes $P_s = (\mu_s, \Sigma_s)$ (and global anchor statistics $P^t_{\text{aux}} = (\mu^t_{\text{aux}}, \Sigma^t_{\text{aux}})$) via weighted statistical aggregation:

$$\mu_s = \frac{\sum_k n_{s,k}\mu_{s,k}}{N_s}, \quad \Sigma_s = \frac{\sum_k n_{s,k}(\Sigma_{s,k} + \delta_{s,k}\delta_{s,k}^\top)}{N_s}. \tag{3}$$

where $n_{s,k} = |Z_{s,k}|$, $N_s = \sum_k n_{s,k}$, and $\delta_{s,k} = \mu_{s,k} - \mu_s$ represents the deviation of the local mean from the global mean. This aggregation ensures that the global prototypes accurately reflect the underlying data distribution across all clients.

However, a fundamental challenge in this setting is feature space drift. As the encoders $\{w_{e,k}^t\}_{k=1}^K$ are updated over tasks, the feature space evolves, and the historical prototypes $\{(\mu_s, \Sigma_s)\}_{s\in\mathcal{Y}^{1:t-1}}$ computed under previous encoder snapshots become misaligned with the current feature space. Directly using these obsolete prototypes may therefore mislead training.

To mitigate this issue, we propose a statistical space alignment mechanism utilizing the statistics of the auxiliary dataset. Since the input distribution of $\mathcal{D}_{\text{aux}}$ is fixed, any shift in its statistical prototypes from the previous global state P^{t-1}_{aux} to the current aggregated state P^t_{aux} reflects the evolution of the global feature space. We estimate the alignment parameters $(\mathbf{A}, \mathbf{b})$ by matching the first and second moments of these distributions:

$$\mathbf{A} = (\Sigma^t_{\text{aux}})^{\frac{1}{2}}(\Sigma^{t-1}_{\text{aux}})^{-\frac{1}{2}}, \quad \mathbf{b} = \mu^t_{\text{aux}} - \mathbf{A}\mu^{t-1}_{\text{aux}}. \tag{4}$$

With the estimated parameters, we align each historical prototype for $s \in \mathcal{Y}^{1:t-1}$ by

$$\hat{\mu}_s = \mathbf{A}\mu_s + \mathbf{b}, \qquad \hat{\Sigma}_s = \mathbf{A}\Sigma_s\mathbf{A}^\top. \tag{5}$$

and use $\hat{P}_s = (\hat{\mu}_s, \hat{\Sigma}_s)$ as the aligned repository of historical data in the current feature space.

Finally, to provide a geometric constraint for client updates, the server identifies the principal directions where historical knowledge resides using these aligned prototypes. We construct a cumulative covariance matrix $\Sigma_{\text{all}} = \sum_{s\in\mathcal{Y}^{1:t-1}} \hat{\Sigma}_s$ and perform eigen decomposition:

$$\Sigma_{\text{all}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top. \tag{6}$$

where $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_d]$ contains the eigenvectors and $\mathbf{\Lambda}$ the eigenvalues. We select the top-r eigenvectors corresponding to the largest eigenvalues to form the basis matrix $\mathbf{U}_{\text{old}} \in \mathbb{R}^{d\times r}$. This subspace $\mathcal{S}_{\text{old}}$ spanned by $\mathbf{U}_{\text{old}}$ captures the primary directions of variation for all historical classes.

Client-Side Dual-Constrained Local Training. Leveraging the aligned prototypes $\hat{P} = \{\hat{P}_s\}_{s\in\mathcal{Y}^{1:t-1}}$ and the historical subspace basis $\mathbf{U}_{\text{old}}$, clients perform local training. We introduce a dual-constraint mechanism to balance the learning of new tasks with the retention of historical knowledge.

Since clients lack access to historical data $\{\mathcal{D}_k^j\}_{j=1}^{t-1}$, we utilize the aligned prototypes to induce virtual posterior distributions. Specifically, we treat the collection of Gaussian distributions $\{\mathcal{N}(\hat{\mu}_s, \hat{\Sigma}_s)\}_{s\in\mathcal{Y}^{1:t-1}}$ as a probabilistic teacher model. For a sample x from the current task with extracted feature $z = f(w_{e,k}^t; x)$, we quantify its affinity to a historical class s using the squared Mahalanobis distance, effectively normalizing the feature space by the class-specific variance:

$$d_s(z) = (z - \hat{\mu}_s)^\top(\hat{\Sigma}_s + \epsilon\mathbf{I})^{-1}(z - \hat{\mu}_s). \tag{7}$$

where ϵ is a stabilization term. Assuming a uniform prior over historical classes, we derive the virtual posterior distribution $q(s|z)$ by applying a softmax function over the negative distances with a temperature scalar τ:

$$q(s|z) = \frac{\exp(-d_s(z)/\tau)}{\sum_{j\in\mathcal{Y}^{1:t-1}} \exp(-d_j(z)/\tau)}. \tag{8}$$

This distribution serves as a soft target that encapsulates the decision boundaries of previous tasks. To align the model's behavior with these historical boundaries, we introduce a statistical distillation constraint $\mathcal{L}_{\text{dist}}$. This term minimizes the Kullback-Leibler (KL) divergence between the virtual posterior $q(\cdot|z)$ and the model's predicted probability distribution over historical classes, denoted as $p_{\text{old}}(\cdot|x)$:

$$\mathcal{L}_{\text{dist}} = \sum_{s\in\mathcal{Y}^{1:t-1}} q(s|z) \log \frac{q(s|z)}{p_{\text{old}}(s|x)}. \tag{9}$$

By minimizing this divergence, the client model aligns its predictions with the aligned historical prototypes, mitigating catastrophic forgetting at the decision level.

Finally, we formulate the local optimization objective as a composite loss $\mathcal{L}_{\text{total}}$. It combines the standard Cross-Entropy loss $\mathcal{L}_{\text{task}}$ for the current task with the distillation constraint, weighted by a hyperparameter λ:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda\mathcal{L}_{\text{dist}}. \tag{10}$$

However, relying solely on decision-level distillation may not suffice to prevent the underlying feature space from drifting destructively under the strong supervision of new tasks. This limitation is underscored by recent studies on the anatomy of catastrophic forgetting [36], which reveal that the loss of historical knowledge is predominantly driven by the distortion of the encoder rather than the classifier. Motivated by this insight, we impose stricter constraints specifically on the encoder to explicitly preserve the geometric structure of historical representations. To this end, we integrate the global covariance information via subspace-constrained gradient projection. We posit that the principal eigenvectors captured by $\mathbf{U}_{\text{old}}$ define the protected directions of variation for historical knowledge. Let $\mathbf{P}_{\text{old}} = \mathbf{U}_{\text{old}}\mathbf{U}_{\text{old}}^\top$ denote the projection matrix onto the historical subspace. During backpropagation, the gradients for the classifier $w_{c,k}^t$ are updated via standard SGD. In contrast, for the encoder, we constrain the gradient propagated through its output feature representation. Specifically,

given the feature-level gradient $\nabla_z \mathcal{L}_{\text{total}} \in \mathbb{R}^d$, we project it onto the orthogonal complement of the historical subspace:

$$\nabla_z \mathcal{L}_{\text{total}} \leftarrow (\mathbf{I} - \mathbf{P}_{\text{old}}) \nabla_z \mathcal{L}_{\text{total}}. \tag{11}$$

The projected feature-level gradient is then backpropagated through the encoder to update $w_{e,k}^t$ via the chain rule. This operation suppresses encoder updates that would move features along the principal directions of old tasks, while preserving plasticity in the orthogonal complement of the historical subspace. As a result, new knowledge can be acquired with reduced disruption to the geometry of previously learned classes.

It is worth noting that SPR-FIL is model-agnostic. In homogeneous scenarios, where clients share the same architecture, the server performs standard weighted model aggregation as in [31]:

$$w^{t+1} = \sum_{k \in \mathcal{S}} \frac{N_k^{t+1}}{\sum_{j \in \mathcal{S}} N_j^{t+1}} w_k^{t+1}. \tag{12}$$

where $\mathcal{S}$ denotes the set of clients participating in the federated flow. In heterogeneous scenarios, where different client models induce different feature spaces, we assign a lightweight projector g_m to each model type and map local features into a unified space:

$$\tilde{z} = g_m\Big(f(w_{e,k}^{t+1}; x)\Big). \tag{13}$$

All prototype construction, alignment, distillation, and gradient projection are then performed on $\tilde{z}$, enabling SPR-FIL to support diverse client models without changing the overall pipeline.

4 Experiments

4.1 Setup

Datasets. To evaluate our proposed SPR-FIL, we conducted experiments under FIL settings on four widely used public datasets: **Fashion-MNIST** [48], **CIFAR-10**, **CIFAR-100** [15], and **Tiny-ImageNet** [16].

Baselines. For a fair comparison, we categorize the baselines into homogeneous and heterogeneous settings, distinguished by whether the server performs global model parameter aggregation. (i) Homogeneous: This includes standard **FedAvg** [31]; combinations of FedAvg with IL techniques, namely **FL+EWC** [11], **FL+SI** [52], and **FL+LwF** [27]; and rehearsal methods including buffer-based (**GLFC** [5], **ReFed+** [21]), and generative (**FedCIL** [35], **CAN**[39]). (ii) Heterogeneous: This includes **FedProto** [43] and its variants combined with standard IL techniques: **Proto+EWC**, **Proto+SI**, and **Proto+LwF**. Unless specified, experiments are conducted under the homogeneous setting.

Configurations. The Class-IL setting is structured as follows: 5 tasks $\times$ 2 classes for both Fashion-MNIST and CIFAR-10, 10 tasks $\times$ 10 classes for CIFAR-100, and 10 tasks $\times$ 20 classes for Tiny-ImageNet. We employ ResNet18 [8] as the backbone for all clients in the homogeneous setting. In the heterogeneous setting, we simulate system heterogeneity by assigning ResNet18 and a simple CNN model with the same feature dimension size to 50% of the clients each. By default, the system runs for $R = 150$ rounds with $E = 20$ local epochs, involving a random subset of clients ($C = 0.6$) at each round. Heterogeneous partitions are generated for 20 clients via a Dirichlet distribution, controlled by the parameter α. We set $\lambda = 1$ for Eq.(10) and $\tau = 2$ for Eq.(8) by default, and the detailed discussion is provided in the parameter sensitivity analysis. Finally, regarding the metrics, we report the final accuracy $A(f)$ and the average accuracy $\bar{A}$. Our evaluation uses Top-1 accuracy for all datasets except Tiny-ImageNet, where Top-5 accuracy is adopted. Specifically, we evaluate the global server model for the homogeneous setting, while reporting the averaged performance of all participating clients for the heterogeneous setting. All results are reported as mean $\pm$ standard deviation over five random seeds.

4.2 Performance Overview

Test Accuracy. Table 1 reports the $A(f)$ and $\bar{A}$ results. (i) Homogeneous: Regularization methods show only marginal improvements over FedAvg. This suggests that parameter constraints derived

Table 1: Performance comparison of different methods with $\alpha = 1.0$. The best results are highlighted in **bold**.

	Method	Fashion-MNIST		CIFAR-10		CIFAR-100		Tiny-ImageNet	
		$A(f)$	$\bar{A}$	$A(f)$	$\bar{A}$	$A(f)$	$\bar{A}$	$A(f)$	$\bar{A}$
Homogeneous	FedAvg	39.57±1.91	63.96±1.11	36.24±1.27	61.02±1.09	27.10±1.23	41.08±0.93	30.17±0.89	49.77±1.28
	FL+EWC	40.21±2.05	64.45±0.78	36.72±1.42	61.70±1.28	27.52±1.42	42.87±1.32	29.95±0.85	49.26±0.76
	FL+SI	39.49±1.23	64.16±0.92	37.09±0.67	61.55±1.76	29.47±0.77	42.50±0.58	30.19±1.22	51.10±1.42
	FL+LwF	39.63±1.06	64.81±1.05	36.54±2.15	61.29±0.62	29.23±2.08	43.06±1.64	31.76±1.27	51.87±1.18
	GLFC	41.29±1.25	64.87±0.73	36.41±1.76	61.35±1.52	31.71±1.09	42.96±1.01	34.21±1.22	52.03±1.30
	ReFed+	42.78±0.99	65.41±0.14	39.01±1.78	62.27±0.73	31.94±0.59	43.65±0.91	35.38±1.49	52.21±0.53
	FedCIL	42.08±1.31	65.29±1.24	37.20±0.93	62.15±1.50	30.35±1.86	42.17±1.12	33.69±0.82	51.39±0.44
	CAN	43.02±0.84	66.74±0.55	37.97±1.26	63.09±0.91	30.08±0.51	42.33±0.76	34.05±0.99	51.08±1.35
	SPR-FIL(homo)	**43.90**±1.53	**67.82**±0.81	**40.03**±1.05	**63.52**±1.36	**35.24**±1.74	**45.35**±1.27	**37.73**±0.92	**55.04**±0.82
Heterogeneous	FedProto	34.73±0.95	57.21±1.32	30.14±2.11	54.67±0.58	20.86±2.17	30.29±1.08	22.56±2.03	39.46±1.20
	Proto+EWC	36.92±2.07	60.41±0.87	32.39±0.57	57.18±0.92	24.58±2.03	33.01±0.55	28.63±1.64	45.37±1.27
	Proto+SI	37.15±1.12	61.32±2.19	33.17±1.63	56.28±0.97	26.03±2.11	35.07±0.85	27.92±1.71	45.79±1.58
	Proto+LwF	37.54±0.65	60.36±0.41	33.05±1.84	56.91±0.44	25.94±1.19	34.56±1.10	27.53±0.74	44.25±0.75
	SPR-FIL(hetero)	**39.43**±1.94	**62.98**±1.56	**36.99**±1.80	**59.24**±1.14	**29.70**±1.40	**37.17**±1.86	**32.29**±1.50	**49.11**±0.48

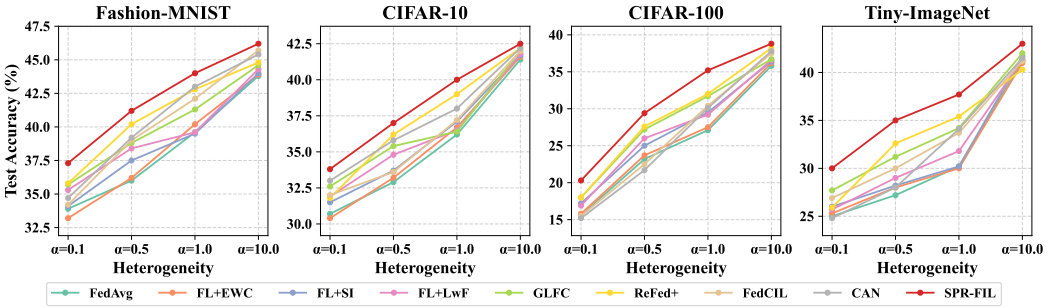


Figure 2: Performance of various methods with different levels of heterogeneity on four datasets.

from local datasets yield biased importance estimates, failing to approximate the global optimization objective. In contrast, rehearsal methods significantly boost performance by leveraging rehearsal buffers or generators. However, they encounter scalability bottlenecks as buffer-based approaches are constrained by limited buffer sizes, while generative baselines suffer from low-fidelity pseudo-samples as the class space expands. (ii) Heterogeneous: By aggregating prototypes from clients, regularization-based FedProto variants mitigate data heterogeneity to some extent and outperform FedProto. However, their performance is still limited by feature misalignment across clients.

Overall, SPR-FIL consistently outperforms state-of-the-art baselines across all datasets in both homogeneous and heterogeneous settings, achieving performance gains of up to 9.73%.

Data Heterogeneity. As shown in Fig. 2, we conducted experiments with four levels of data heterogeneity, and SPR-FIL consistently maintains the largest performance advantage, especially under high non-IID settings. This robustness stems from the statistical space alignment mechanism. In heterogeneous scenarios, relying solely on local statistics leads to biased estimates of global feature evolution. By utilizing the fixed shared auxiliary anchor set as a consistent reference, our method isolates the true feature space drift from local distribution shifts. This ensures that the aligned historical prototypes are accurately mapped to the current feature space, providing valid supervision signals regardless of client-side data skew.

Efficiency Analysis. We evaluate the efficiency of SPR-FIL from both communication and computational perspectives, as shown in Table 2. In terms of communication efficiency, SPR-FIL requires the fewest communication rounds among all compared methods. On CIFAR-

10, SPR-FIL converges in 112 rounds, reducing the communication rounds by 34.9% compared with FedAvg. Since communication in FL is mainly dominated by repeated model transmission, reducing the number of rounds directly lowers the overall communication cost. Although SPR-FIL introduces additional prototype and anchor statistics, these statistics are transmitted only once per task rather than in every communication round. Therefore, the extra communication overhead is negligible compared with the repeated model exchanges in FedAvg. For computational efficiency, SPR-FIL introduces only lightweight feature-space operations, including prototype distillation, gradient projection, and prototype alignment, without storing or replaying historical samples. As shown in Table 2, although SPR-FIL performs slightly slower than FedAvg, it is faster than GLFC and ReFed+ on both datasets. Together with the fewest communication rounds, these results show that SPR-FIL achieves a favorable trade-off between communication efficiency and computational cost.

Table 2: Efficiency comparison with $\alpha = 1.0$.

Method	CIFAR-10		CIFAR-100	
	Rounds	Time(s)	Rounds	Time(s)
FedAvg	172	90.8	230	53.2
GLFC	129	101.2	207	65.4
ReFed+	140	121.5	215	71.9
SPR-FIL	112	96.9	186	63.1

Ablation Study. We validate the effectiveness of each component by comparing SPR-FIL with two variants: (i) "w/o Align". removes the anchor-based statistical alignment, directly using raw historical prototypes which ignores feature drift; (ii) "w/o Ortho." removes the subspace-constrained gradient projection of encoder update, relying solely on the distillation loss.

Table 3: Ablation Study of SPR-FIL with $\alpha = 1.0$.

Method	CIFAR-10		CIFAR-100	
	$A(f)$	$\bar{A}$	$A(f)$	$\bar{A}$
SPR-FIL	40.03$\pm$1.05	63.52$\pm$1.36	35.24$\pm$1.74	45.35$\pm$1.27
- w/o Align.	36.71$\pm$1.42	61.39$\pm$0.72	30.01$\pm$1.86	40.90$\pm$1.15
- w/o Ortho.	38.92$\pm$0.87	62.38$\pm$2.11	34.16$\pm$1.59	43.97$\pm$1.73

Results in Table 3 confirm the necessity of both. "w/o Align" suffers significant degradation, particularly on CIFAR-100. This confirms that without actively compensating for feature space drift, the historical prototypes become misaligned with the current encoder state. Consequently, the distillation mechanism receives obsolete statistics, leading to error propagation. Similarly, the decline in "w/o Ortho" highlights the limitation of relying solely on statistical distillation. While distillation provides a soft target, it cannot strictly prevent encoder updates from disrupting the historical subspace. The gradient projection onto the orthogonal complement is thus essential to explicitly preserve the geometric structure of historical knowledge.

Parameter Sensitivity Analysis. We investigate the sensitivity of SPR-FIL to two key hyperparameters: the loss weight λ and the distillation temperature τ, as illustrated in Fig. 3, where box plots summarize the results over five random seeds. First, regarding the balancing weight λ, we observe a classic trade-off between stability and plasticity. A small λ fails to impose sufficient constraints on the historical decision boundaries, leading to forgetting, whereas an excessively large λ over-regularizes the model, hindering the acquisition of new concepts. The performance remains stable within a wide range (e.g., $\lambda \in [0.5, 2.0]$), achieving the best balance at $\lambda = 1.0$. Second, for the temperature τ, extremely low values result in sharp posterior distributions that may cause gradient vanishing, while overly high values lead to a uniform distribution, obscuring the informative structural relationships between classes. The results indicate that SPR-FIL is robust to τ, with optimal performance observed around $\tau = 2.0$.

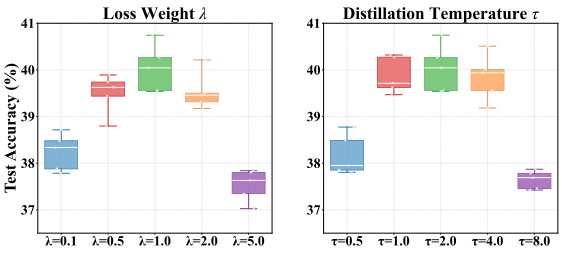


Figure 3: Parameter sensitivity analysis of the loss weight λ and distillation temperature τ with $\alpha = 1.0$ on CIFAR-10.

5 Conclusion

In this paper, we introduced SPR-FIL, a novel rehearsal-free and lightweight framework for FIL. By leveraging Gaussian statistics and preserving previous knowledge through aligned prototypes and subspace-constrained encoder updates, SPR-FIL effectively mitigates catastrophic forgetting and proves to be model-agnostic, making it highly suitable for deployment on resource-constrained edge devices.

Acknowledgements

This work was supported by the National Natural Science Foundation under grants 625B2073 and 62203309.

References

[1] Saif Khalifa Aljunaid, Saif Jasim Almheiri, Hussain Dawood, and Muhammad Adnan Khan. Secure and transparent banking: Explainable ai-driven federated learning model for financial fraud detection. *Journal of Risk and Financial Management*, 18(4):179, 2025.

[2] Yavuz Faruk Bakman, Duygu Nur Yaldiz, Yahya H Ezzeldin, and Salman Avestimehr. Federated orthogonal training: Mitigating global catastrophic forgetting in continual federated learning. *arXiv preprint arXiv:2309.01289*, 2023.

[3] Pushpita Chatterjee, Debashis Das, and Danda B Rawat. Federated learning empowered recommendation model for financial consumer services. *IEEE Transactions on Consumer Electronics*, 70(1):2508–2516, 2023.

[4] Chang Che, Ziqi Wang, Pengwan Yang, Cheems Wang, Hui Ma, and Zenglin Shi. Lora in lora: Towards parameter-efficient architecture expansion for continual visual instruction tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 19978–19986, 2026.

[5] Jiahua Dong, Lixu Wang, Zhen Fang, Gan Sun, Shichao Xu, Xiao Wang, and Qi Zhu. Federated class-incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10164–10173, 2022.

[6] Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *International conference on artificial intelligence and statistics*, pages 3762–3773. PMLR, 2020.

[7] Farzin Haddadpour, Mohammad Mahdi Kamani, Aryan Mokhtari, and Mehrdad Mahdavi. Federated learning with compression: Unified analysis and sharp guarantees. In *International Conference on Artificial Intelligence and Statistics*, pages 2350–2358. PMLR, 2021.

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

[9] Peter Kairouz and H Brendan McMahan. Advances and open problems in federated learning. *Foundations and trends in machine learning*, 14(1-2):1–210, 2021.

[10] Hirokatsu Kataoka, Kazushige Okayasu, Asato Matsumoto, Eisuke Yamagata, Ryosuke Yamada, Nakamasa Inoue, Akio Nakamura, and Yutaka Satoh. Pre-training without natural images. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[11] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[12] Likhitha Kolla, Kristin Linn, Amol S Navathe, Craig Kreisler, Christopher B Roberts, Sae-Hwan Park, Harvineet Singh, Jean Feng, Jinbo Chen, and Ravi B Parikh. Performance drift in a nationally deployed population health risk algorithm in the us veterans health administration. In *JAMA Health Forum*, volume 6, page e252717, 2025.

[13] Ali Kore, Elyar Abbasi Bavil, Vallijah Subasri, Moustafa Abdalla, Benjamin Fine, Elham Dolatabadi, and Mohamed Abdalla. Empirical data drift detection experiments on real-world medical imaging data. *Nature communications*, 15(1):1887, 2024.

[14] Zhiqiang Kou, Junxiang Wu, Wenke Huang, Wenwen He, Ming-Kun Xie, Changwei Wang, Yuheng Jia, Di Jiang, Yang Liu, Xin Geng, and Qiang Yang. Fedharmony: Harmonizing heterogeneous label correlations in federated multi-label learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

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

[16] Yann Le, Xuan Yang, et al. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

[17] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10713–10722, 2021.

[18] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

[19] Xiaoxiao Li, Zhao Song, and Jiaming Yang. Federated adversarial learning: A framework with convergence analysis. In *International conference on machine learning*, pages 19932–19959. PMLR, 2023.

[20] Yichen Li, Qunwei Li, Haozhao Wang, Ruixuan Li, Wenliang Zhong, and Guannan Zhang. Towards efficient replay in federated incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12820–12829, 2024.

[21] Yichen Li, Haozhao Wang, Yining Qi, Wei Liu, and Ruixuan Li. Re-fed+: A better replay strategy for federated incremental learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[22] Yichen Li, Haozhao Wang, Wenchao Xu, Tianzhe Xiao, Hong Liu, Minzhu Tu, Yuying Wang, Xin Yang, Rui Zhang, Shui Yu, et al. Unleashing the power of continual learning on non-centralized devices: A survey. *IEEE Communications Surveys & Tutorials*, 28:1059–1098, 2025.

[23] Yichen Li, Xiuying Wang, Wenchao Xu, Haozhao Wang, Yining Qi, Jiahua Dong, and Ruixuan Li. Feature distillation is the better choice for model-heterogeneous federated learning. *arXiv preprint arXiv:2507.10348*, 2025.

[24] Yichen Li, Yuying Wang, Haozhao Wang, Yining Qi, Tianzhe Xiao, and Ruixuan Li. Fedssi: Rehearsal-free continual federated learning with synergistic synaptic intelligence. In *Forty-second International Conference on Machine Learning*.

[25] Yichen Li, Wenchao Xu, Yining Qi, Haozhao Wang, Ruixuan Li, and Song Guo. Sr-fdil: Synergistic replay for federated domain-incremental learning. *IEEE Transactions on Parallel and Distributed Systems*, 35(11):1879–1890, 2024.

[26] Yichen Li, Wenchao Xu, Haozhao Wang, Yining Qi, Jingcai Guo, and Ruixuan Li. Personalized federated domain-incremental learning based on adaptive knowledge matching. In *European conference on computer vision*, pages 127–144. Springer, 2024.

[27] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

[28] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. *Advances in neural information processing systems*, 33:2351–2363, 2020.

[29] Yunbo Long, Jiaquan Zhang, Xi Chen, and Alexandra Brintrup. Topological federated clustering via gravitational potential fields under local differential privacy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 24044–24051, 2026.

[30] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

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

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

[33] Zihao Peng, Nan Zou, Jiandian Zeng, Boyuan Li, Shengbo Chen, Tian Wang, and Weijia Jia. Dynamic grouping and aggregation weight optimization for hierarchical federated learning with quantization. *IEEE Transactions on Networking*, 2026.

[34] Rajesh Poojari. Privacy-preserving generative ai in healthcare systems using federated learning approaches. *International Journal of Data Science and IoT Management System*, 5(1):78–88, 2026.

[35] Daiqing Qi, Handong Zhao, and Sheng Li. Better generative replay for continual federated learning. *arXiv preprint arXiv:2302.13001*, 2023.

[36] Vinay V Ramasesh, Ethan Dyer, and Maithra Raghu. Anatomy of catastrophic forgetting: Hidden representations and task semantics. *arXiv preprint arXiv:2007.07400*, 2020.

[37] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

[38] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in neural information processing systems*, 32, 2019.

[39] Xuankun Rong, Jianshu Zhang, Kun He, and Mang Ye. Can: Leveraging clients as navigators for generative replay in federated continual learning. In *Forty-second International Conference on Machine Learning*, 2025.

[40] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

[41] Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. *arXiv preprint arXiv:2103.09762*, 2021.

[42] Jinhyun So, Chaoyang He, Chien-Sheng Yang, Songze Li, Qian Yu, Ramy E Ali, Basak Guler, and Salman Avestimehr. Lightsecagg: a lightweight and versatile design for secure aggregation in federated learning. *Proceedings of Machine Learning and Systems*, 4:694–720, 2022.

[43] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 8432–8440, 2022.

[44] Gido M Van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019.

[45] Junda Wang, Minghui Hu, Ning Li, Abdulaziz Al-Ali, and Ponnuthurai Nagaratnam Suganthan. Incremental online learning of randomized neural network with forward regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

[46] Xiuying Wang, Yichen Li, Hang Su, Gaozhuo Liu, Shiwei Li, Chuang Zhao, Jiangming Shi, and Imran Razzak. Lightweight federated incremental learning via decoupled replay. In *Forty-third International Conference on Machine Learning*.

[47] Junxiang Wu, Zhiqiang Kou, Hongwei Zeng, Wenke Huang, Biao Liu, Hanlin Gu, Yuheng Jia, Di Jiang, Yang Liu, and Xin Geng. Trustworthy federated label distribution learning under annotation quality disparity, 2026.

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

[49] Jie Xu, Benjamin S Glicksberg, Chang Su, Peter Walker, Jiang Bian, and Fei Wang. Federated learning for healthcare informatics. *Journal of healthcare informatics research*, 5(1):1–19, 2021.

[50] Shuangqing Xu, Yifeng Zheng, and Zhongyun Hua. Harnessing sparsification in federated learning: A secure, efficient, and differentially private realization. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, pages 2354–2368, 2025.

[51] Jaehong Yoon, Wonyong Jeong, Giwoong Lee, Eunho Yang, and Sung Ju Hwang. Federated continual learning with weighted inter-client transfer. In *International conference on machine learning*, pages 12073–12086. PMLR, 2021.

[52] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. Pmlr, 2017.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The main contributions are summarized in the abstract and introduction sections.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discussed the limitations of our proposed method.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: We did not include new theorems.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We have included code required to reproduce the method and some important details in the paper.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: We include code in the supplemental material.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: All the brief introduction of the experimental setting are included in the paper. More detailed information could be found in the Code.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: The results are accompanied by error bars, confidence intervals, or statistical significance tests.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We provide the amount of compute required for experiments.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We have reviewed the Code of Ethics of NeurIPS.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: Our work does not have broader positive or negative societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: We don't pose such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We cited all the papers of code, dataset and model in our paper.

    Guidelines:

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

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: We only submit the code in the supplemental material without releasing to public.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: We haven't use any crowdsourcing and human efforts in our paper.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [N/A]

    Justification: No human subjects were involved in our paper.

    Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: We only use LLMs for writing, editing, or formatting.

Guidelines:

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