Avoid Forgetting by Preserving Global Knowledge Gradients in Federated Learning with Non-IID Data

Abhijit Chunduru[*1], **Majid Morafah**[*2], **Mahdi Morafah**[3], **Vishnu Pandi Chellapandi**[4], **Ang Li**[5]

[1]University of Massachusetts at Amherst, [2]Islamic Azad University,
[3]University of California San Diego, [4]Cummins, [5]University of Maryland College Park
[*]Equal first authorship and contribution

Abstract

The inevitable presence of data heterogeneity has made federated learning very challenging. There are numerous methods to deal with this issue, such as local regularization, better model fusion techniques, and data sharing. Though effective, they lack a deep understanding of how data heterogeneity can affect the global decision boundary. In this paper, we bridge this gap by performing an experimental analysis of the learned decision boundary using a toy example. Our observations are surprising: (1) we find that the existing methods suffer from forgetting and clients forget the global decision boundary and only learn the perfect local one, and (2) this happens regardless of the initial weights, and clients forget the global decision boundary even starting from pre-trained optimal weights. In this paper, we present FedProj, a federated learning framework that robustly learns the global decision boundary and avoids its forgetting during local training. To achieve better ensemble knowledge fusion, we design a novel server-side ensemble knowledge transfer loss to further calibrate the learned global decision boundary. To alleviate the issue of learned global decision boundary forgetting, we further propose leveraging an episodic memory of average ensemble logits on a public unlabeled dataset to regulate the gradient updates at each step of local training. Experimental results demonstrate that FedProj outperforms state-of-the-art methods by a large margin.

1 Introduction

Federated Learning (FL) [20] has emerged as a privacy-preserving framework that trains a shared global model across multiple clients without exchanging raw data, under the coordination of a central server. The most widely adopted FL method, FedAvg [32], aggregates local weight or gradient updates to learn the global model. Although FedAvg has demonstrated promise across various applications, its performance is significantly hindered by data heterogeneity [3–6, 13, 28, 34–36, 45] among clients—commonly referred to as Non-IID data—which can lead to poor convergence and degraded performance.

Numerous studies have sought to address the challenges posed by Non-IID data in FL by mitigating client drift [21, 26], enhancing server-side model fusion and distillation [7, 24, 51], and refining local training protocols [9]. Despite these efforts, a critical issue remains underexplored: during local training, clients tend to overfit to their individual objectives, thereby *catastrophically forgetting the global knowledge and decision boundaries learned by the shared aggregated model*. Although prior works have acknowledged this phenomenon [47], a comprehensive experimental investigation quantifying global knowledge forgetting is lacking. In our work, we first present an empirical pilot

Preprint. Under review.

study of standard federated learning, i.e. FedAvg, under Non-IID settings, demonstrating that local clients often completely forget global knowledge and converge to models that are fine-tuned solely to their local objectives. Consequently, *averaging such divergent models merely restores the original performance rather than enhancing it*, and the server-side fusion is further compromised as client models have become increasingly dissimilar. Notably, this catastrophic forgetting occurs even when clients begin training from a pre-trained global model.

Motivated by these insights we propose a novel federated learning method called *FedProj*. Our approach mitigates global knowledge forgetting by imposing explicit constraints on local gradient updates where the actual forgetting happens, thereby preserving the learned global knowledge. Specifically, we maintain a small episodic memory of global knowledge using a public dataset and formulate an optimization problem that constrains new gradient directions to prevent an increase in the losses associated with the global knowledge. At the server side, we further enhance model fusion through ensemble knowledge distillation using both logits and feature representations, and we incorporate a novel weight divergence regularization term to alleviate the adverse effects of noisy distillation. Extensive experiments on both computer vision and natural language processing tasks demonstrate that FedProj significantly outperforms state-of-the-art methods across multiple datasets, achieving superior performance in Non-IID federated learning scenarios.

Contributions. Our work makes the following contributions:

- We present the first empirical pilot study that provides new insights into the phenomenon of catastrophic forgetting in standard federated learning under data heterogeneity.
- We introduce a novel federated learning method, *FedProj*, which mitigates global knowledge forgetting by imposing explicit constraints on local gradient updates and preserving global knowledge.
- We conduct extensive experiments on both CV and NLP tasks and demonstrate that our proposed method outperforms state-of-the-art approaches on various benchmarks and datasets under Non-IID settings.

Organization. The remainder of this paper is organized as follows. In Section 2, we review the related work. Section 3 presents our empirical pilot study. In Section 4, we describe the proposed FedProj methodology. Section 5 details our experimental results, and Section 6 concludes the paper.

2 Related Works

Continual Learning. In continual learning, gradient projection methods mitigate catastrophic forgetting by constraining parameter updates to preserve past knowledge. These approaches identify critical directions in weight space and restrict updates to be orthogonal or complementary to them [12, 48]. In particular, Gradient Projection Memory (GPM) [38] focuses on key gradient subspaces and explicitly preserves the past gradient subspaces. More recent methods relax strict orthogonality to balance stability and plasticity [30, 37]. These methods frame continual learning as a stability-plasticity tradeoff in linear algebra, ensuring knowledge retention while allowing adaptability. Our work draws intuition from these body of works in continual learning to allieviate the catastrophic forgetting in FL.

FL with Non-IID Data. Federated learning with non-IID data presents significant challenges, prompting diverse strategies to improve convergence and accuracy. FedAvg [32] established the foundation for FL by averaging local model updates, but it struggles with statistical heterogeneity. To address this, FedProx [26] introduces a proximal term to stabilize local updates, while FedNova [42] normalizes local contributions, mitigating objective inconsistency. SCAFFOLD [21] introduces control variates to correct local updates, effectively reducing drift by anchoring clients to a global direction. FedGen [50] tackles non-IID challenges by generating synthetic data at the server using generative models, improving generalization across clients. For personalization, Ditto [27] maintains dual local-global models, improving client-specific performance, while pFedHN [40] uses hypernetworks to generate personalized models, enhancing adaptability. FedBN [29] addresses feature

shift by keeping batch normalization layers local, while sharing other parameters, boosting robustness in non-IID settings. MOON [25] employs contrastive learning to align local and global models, reducing divergence.

Comparison with Close Works. FedDF [31] enhances server-side performance through knowledge distillation, refining the global model by distilling knowledge from local models. FedET [8] employs ensemble distillation, combining local knowledge into a generalized global model, improving accuracy on heterogeneous data. FedGKT [17] introduces group knowledge transfer, where lightweight client models distill knowledge into a larger server model, boosting efficiency and handling model heterogeneity. Researchers in [15] propose a privacy-preserving FL framework with one-shot offline knowledge distillation using unlabeled public data, reducing communication overhead while enhancing privacy guarantees.

In contrast, FedProj directly applies gradient projection onto the global knowledge gradient subspace, effectively mitigating client drift and preserving global knowledge across non-IID clients. Moreover, while FedDF, FedET, and FedGKT focus on logit-level or ensemble-based distillation, FedProj retains knowledge at the gradient level, ensuring more effective knowledge alignment in non-IID FL settings.

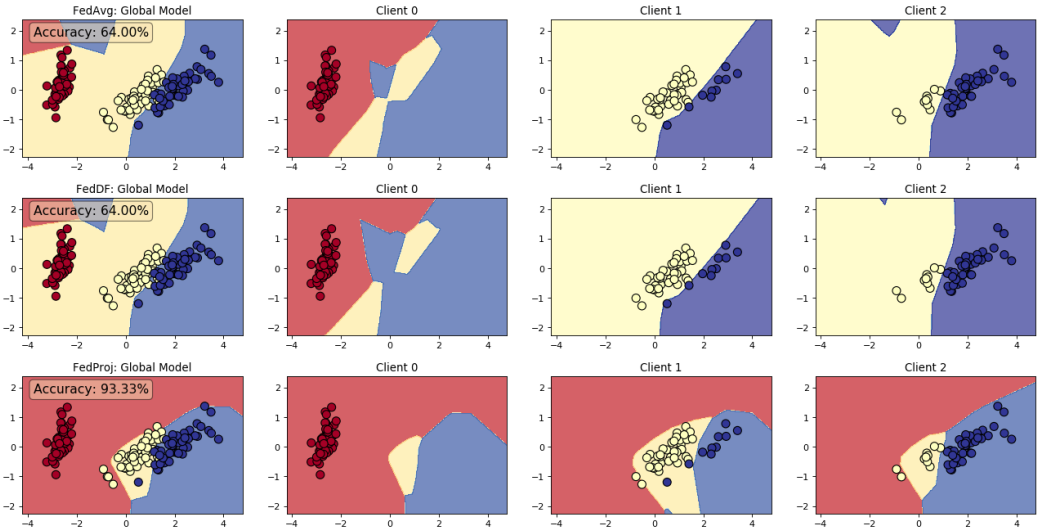


Figure 1: **Visualization of Catastrophic Forgetting of Global Decision Boundaries under Non-IID Federated Learning.** This figure illustrates that standard federated learning methods (FedAvg and FedDF) experience significant catastrophic forgetting of global decision boundaries after local training on Non-IID client data, leading to poor global model performance (64%). In comparison, our proposed FedProj method successfully preserves global decision boundaries during local updates, resulting in a robust and highly accurate global model (93.33%).

3 Pilot Study: Catastrophic Forgetting of Global Knowledge

In this section, we conduct a systematic empirical study to demonstrate and analyze catastrophic forgetting of global knowledge, specifically global decision boundaries, in federated learning under Non-IID data conditions.

Experimental Setup. To clearly visualize decision boundaries and intuitively illustrate global knowledge forgetting, we utilize the Iris dataset and apply Principal Component Analysis (PCA) to reduce its dimensionality to 2D. The Iris dataset comprises 150 samples evenly distributed among three distinct classes: Setosa, Versicolor, and Virginica, each characterized by four numerical features. We partition the data into three Non-IID clients, each predominantly containing data from different subsets of the available classes to simulate realistic heterogeneous conditions.

We employ a simple 3-layer Multi-Layer Perceptron (MLP) neural network as the local model at each client. For optimization, we use Stochastic Gradient Descent (SGD) with a learning rate of 1×10^{-3} and momentum of 0.9. To compare standard FL methods with our proposed method, we run federated learning using FedAvg, FedDF, and FedProj. We set the number of communication rounds to 20 and allow each client to perform local training for 5 epochs per round.

Results and Analysis. Figure 1 presents visualizations of decision boundaries generated by each method. The decision boundaries for each class—red, yellow, and blue—are indicated by their corresponding colors. The client-specific decision boundaries are plotted after the completion of local training for each communication round.

Our observations clearly indicate catastrophic forgetting in standard federated learning methods (FedAvg and FedDF). After local training, individual clients drastically diverge from the global decision boundaries, particularly evident in Client 1 and Client 2. This divergence signifies the loss of global knowledge as clients overfit to their localized data distribution, ignoring global classification objectives. Consequently, aggregating these diverged local models leads to poor global accuracy and unstable decision boundaries.

In contrast, our proposed method, FedProj, demonstrates significant robustness against global knowledge forgetting. By explicitly incorporating constraints via gradient projection to retain global decision boundaries, FedProj effectively preserves global knowledge throughout the local training process. As a result, the aggregated global model consistently maintains high classification accuracy and stable decision boundaries across all clients.

4 Methodology

In this section, we detail our proposed *FedProj* methodology, a novel federated learning gradient projection algorithm that addresses *global knowledge catastrophic forgetting* in the presence of Non-IID data. Our approach combines explicit constraint-based gradient projection in client-side local training with server-side knowledge distillation. Below, we first describe the basic problem setting and provide a mathematical formulation of our goal, then detail the proposed method.

4.1 Problem Formulation and Preliminaries

Consider a federated learning system with N clients, indexed by $k = 1, \ldots, N$, each with a private local dataset $\mathcal{D}_k$. Let $\boldsymbol{\theta} \in \mathbb{R}^p$ denote the parameters of the global model to be learned. In standard federated learning, the server initializes $\boldsymbol{\theta}_g^{(0)}$ and, at each communication round t, sends the current global parameters $\boldsymbol{\theta}_g^{(t)}$ to a subset of selected clients $\mathcal{S}_t \subseteq \{1, \ldots, N\}$. Each client $k \in \mathcal{S}_t$ then performs local training using its private data $\mathcal{D}_k$, yielding a locally updated model $\boldsymbol{\theta}_k^{(t+1)}$. The server then aggregates these updates (e.g., via simple averaging):

$$\boldsymbol{\theta}_g^{(t+1)} \;=\; \sum_{k \in \mathcal{S}_t} \frac{|\mathcal{D}_k|}{\sum_{j \in \mathcal{S}_t} |\mathcal{D}_j|} \, \boldsymbol{\theta}_k^{(t+1)}. \tag{1}$$

While such aggregation works well under IID settings, it suffers from severe performance drops under data heterogeneity (Non-IID). Our *FedProj* algorithm addresses this by incorporating:

1. **Client-Side Gradient Projection:** During local updates, we impose explicit gradient constraints that preserve *global knowledge*, thereby preventing the local model from overfitting exclusively to client-specific objectives.
2. **Server-Side Knowledge Distillation:** Once local updates are sent back, the server fuses them more effectively through ensemble distillation, using a small, auxiliary *public dataset* to align their logits and feature representations. Furthermore, to avoid the negative impact of noisy ensemble distillation we further add a new weight divergence regularizer.

4.2 Local Training with Gradient Projection

To illustrate the core idea of our gradient projection, we first define the local objective and then show how to preserve global knowledge via an explicit constraint on the gradient.

Local Objective. Let k be a particular client in the selected subset $\mathcal{S}_t$ at round t. The client receives $\boldsymbol{\theta}_g^{(t)}$ from the server and initializes $\boldsymbol{\theta}_k$ with it. The local training objective typically involves empirical risk minimization on $\mathcal{D}_k$, for instance:

$$\min_{\boldsymbol{\theta}_k} \quad \frac{1}{|\mathcal{D}_k|} \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}_k} \ell\big(f(\mathbf{x};\,\boldsymbol{\theta}_k),\ \mathbf{y}\big), \tag{2}$$

where $\ell(\cdot,\cdot)$ is a loss function (e.g., cross-entropy) and $f(\mathbf{x};\,\boldsymbol{\theta}_k)$ is the client-side model output.

Global Knowledge Memory. Recall that our objective is to avoid *global knowledge forgetting* during each client's local updates. To achieve this, we introduce a *memory loss* $\mathcal{L}_{\text{mem}}$ that measures how well the current local model $\boldsymbol{\theta}_k$ preserves the global knowledge. Specifically, the global knowledge is distilled via the server's ensemble logits on a small public dataset $\mathcal{D}_{\text{pub}}$. Let

$$\mathcal{Z}_{\text{server}}(\mathbf{x}_m) \;=\; \frac{1}{|\mathcal{S}_t|} \sum_{\ell\in\mathcal{S}_t} f_\ell(\mathbf{x}_m;\,\boldsymbol{\theta}_\ell^{(t+1)}),$$

be the server-aggregated logits (averaged over the selected clients $\mathcal{S}_t$) for a sample $\mathbf{x}_m \in \mathcal{D}_{\text{pub}}$. We denote the local model logits by $\mathcal{Z}_k(\mathbf{x}_m;\,\boldsymbol{\theta}_k)$. Then we define

$$\begin{aligned}\mathcal{L}_{\text{mem}}(\boldsymbol{\theta}_k) = \frac{1}{|\mathcal{M}|} \sum_{(\mathbf{x}_m)\in\mathcal{M}} \mathrm{KL}\big(\sigma(\mathcal{Z}_{\text{server}}(\mathbf{x}_m)),\\ \sigma(\mathcal{Z}_k(\mathbf{x}_m;\,\boldsymbol{\theta}_k))\big)\end{aligned} \tag{3}$$

where $\mathcal{M} \subseteq \mathcal{D}_{\text{pub}}$ is a small *memory buffer* for preserving global knowledge, $\sigma(\cdot)$ is the softmax function, and KL is the Kullback–Leibler divergence. Minimizing $\mathcal{L}_{\text{mem}}$ ensures that local updates remain aligned with global model knowledge, preventing catastrophic forgetting.

Constrained Local Objective and Gradient Projection. Let $\mathcal{L}_{\text{local}}(\boldsymbol{\theta}_k)$ be the ordinary local objective (e.g., cross-entropy on the client's private data $\mathcal{D}_k$). Our goal is to minimize $\mathcal{L}_{\text{local}}$ *without* increasing $\mathcal{L}_{\text{mem}}$. Formally, we can write a constraint-based objective:

$$\min_{\boldsymbol{\theta}_k} \;\; \mathcal{L}_{\text{local}}(\boldsymbol{\theta}_k) \quad \text{subject to} \quad \mathcal{L}_{\text{mem}}(\boldsymbol{\theta}_k) \;\le\; \mathcal{L}_{\text{mem}}\big(\boldsymbol{\theta}_k^{\text{old}}\big), \tag{4}$$

where $\boldsymbol{\theta}_k^{\text{old}}$ is the local model state prior to the new update (i.e., just received from the server). Intuitively, the local model's memory loss must not exceed its old memory loss, ensuring that the updated local parameters $\boldsymbol{\theta}_k$ do not degrade global knowledge.

Although (4) is straightforward conceptually, it is challenging to solve directly in the high-dimensional parameter space of neural networks. Instead, we locally approximate $\mathcal{L}_{\text{mem}}(\boldsymbol{\theta}_k)$ around $\boldsymbol{\theta}_k^{\text{old}}$. Concretely, let:

$$g_{\text{new}} \;=\; \nabla_{\boldsymbol{\theta}_k}\mathcal{L}_{\text{local}}(\boldsymbol{\theta}_k), \quad g_{\text{glob}} \;=\; \nabla_{\boldsymbol{\theta}_k}\mathcal{L}_{\text{mem}}(\boldsymbol{\theta}_k).$$

Requiring $\mathcal{L}_{\text{mem}}(\boldsymbol{\theta}_k) \le \mathcal{L}_{\text{mem}}(\boldsymbol{\theta}_k^{\text{old}})$ to first order amounts to ensuring $\langle g_{\text{proj}},\, g_{\text{glob}}\rangle \ge 0$. Therefore, we rewrite the constraint in terms of the projected gradient update g_{proj}, leading to the following equivalent quadratic program:

$$\min_{g_{\text{proj}}} \quad \frac{1}{2}\big\| g_{\text{new}} \;-\; g_{\text{proj}}\big\|_2^2, \quad \text{subject to} \quad \langle g_{\text{proj}},\, g_{\text{glob}}\rangle \;\ge\; 0. \tag{5}$$

Solving (5) enforces that our final update direction does not *negatively correlate* with g_{glob}, thus preventing increases in the memory-based loss and preserving the global knowledge as a result.

The optimal g_{proj} can be derived via standard Lagrangian methods, yielding:

$$g_{\text{proj}} = \begin{cases} g_{\text{new}}, & \text{if } \langle g_{\text{new}}, g_{\text{glob}} \rangle \geq 0, \\ g_{\text{new}} - \frac{\langle g_{\text{new}}, g_{\text{glob}} \rangle}{\| g_{\text{glob}} \|^2 + \epsilon} g_{\text{glob}}, & \text{otherwise,} \end{cases}$$

where $\epsilon > 0$ is added for numerical stability. Finally, the local model parameters are updated as:

$$\boldsymbol{\theta}_k \leftarrow \boldsymbol{\theta}_k - \eta_{\text{local}} g_{\text{proj}}, \tag{6}$$

where η_{local} is the local learning rate. This ensures that we locally minimize $\mathcal{L}_{\text{local}}$ while preserving global knowledge encapsulated by $\mathcal{L}_{\text{mem}}$.

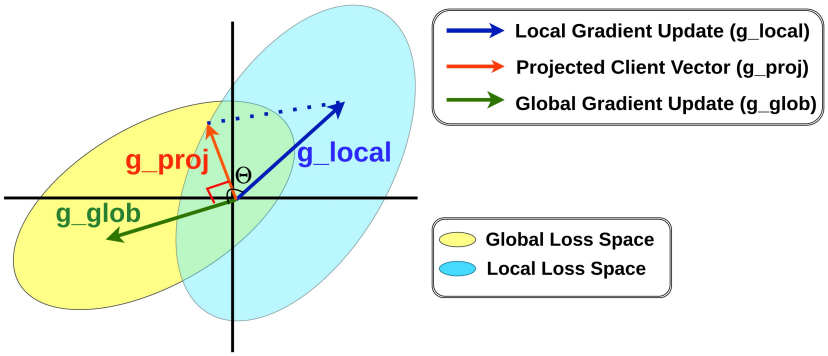


Figure 2: **Gradient Projection in FedProj:** Local gradient updates (g_{local}) are projected onto a subspace orthogonal to the global gradient (g_{glob}), resulting in the projected vector (g_{proj}) for better knowledge retention.

Figure 2 illustrates the gradient update process in heterogeneous local and global loss spaces. The yellow and blue regions represent the global and local loss spaces, respectively. The local gradient update, g_{local} (solid blue arrow), may deviate from the global objective due to data heterogeneity. To correct this, it is projected onto the global loss space, yielding g_{proj} (solid red arrow), which better aligns with the global objective. The final update, g_{glob} (green arrow), integrates the projected gradient, preserving global knowledge. The angle θ highlights the deviation, while the red arc indicates the correction applied.

The full details of FedProj algorithm (Algorithm 1) and mathematical derivations are presented in the Appendix.

4.3 Server-Side Distillation and Model Fusion

After all participating clients finish their local updates, the server aggregates them. Instead of a simple weight average stated in equation (1), FedProj further *distills* knowledge across clients via a small *public* or *proxy* dataset. Let $\{\boldsymbol{\theta}_k^{(t+1)}\}_{k \in \mathcal{S}_t}$ be the client-updated parameters. We form an *ensemble of teachers*

$$\{ f_k(\cdot; \boldsymbol{\theta}_k^{(t+1)}) \mid k \in \mathcal{S}_t \},$$

and a *student model* $f_g(\cdot; \boldsymbol{\theta}_g)$ on the server, initially set to the simple FedAvg result $\boldsymbol{\theta}_g = \boldsymbol{\theta}_g^{\text{FedAvg}}$ obtained via equation (1).

Logit Distillation. For each mini-batch $\mathbf{X}$ sampled from $\mathcal{D}_{\text{pub}}$, we compute teacher logits:

$$\mathcal{Z}_{\text{teacher}} = \frac{1}{|\mathcal{S}_t|} \sum_{k \in \mathcal{S}_t} f_k(\mathbf{X}; \boldsymbol{\theta}_k^{(t+1)}),$$

then align the student's logits $\mathcal{Z}_{\text{student}} = f_g(\mathbf{X}; \boldsymbol{\theta}_g)$ to these teacher logits through a knowledge-distillation loss:

$$\mathcal{L}_{\text{KD}} = \text{KL}\Big(\sigma\big(\mathcal{Z}_{\text{student}}\big) \,,\, \sigma\big(\mathcal{Z}_{\text{teacher}}\big)\Big), \tag{7}$$

where $\text{KL}(\cdot, \cdot)$ is the Kullback–Leibler divergence, and $\sigma(\cdot)$ is the softmax function.

Weight Divergence Regularization. The ensemble distillation process is noisy. This noise is primarily due to utilizing public data that is different from the actual learning private dataset. To scaffold the global model from the nosiy ensemble distillation process, we introduce a weight divergence regularization term. This term encourages the global model parameters to remain close to the averaged model in order to not forget the learned knowledge during the noisy distillation process. In particular, this regularization term allows for learning new knowledge while not being negatively impacted by the noise. Formally, given the global model parameters $\boldsymbol{\theta}_g$ and the local model parameters $\boldsymbol{\theta}_k$, one can impose:

$$\mathcal{L}_{\text{div}} = \big\|\boldsymbol{\theta}_k - \boldsymbol{\theta}_g\big\|_2^2.$$

Overall Server-Side Objective. Combining logit distillation, feature distillation, and weight divergence regularization, the server update solves:

$$\min_{\boldsymbol{\theta}_g} \quad \mathcal{L}_{\text{KD}}(\boldsymbol{\theta}_g) \;+\; \alpha \mathcal{L}_{\text{div}}(\boldsymbol{\theta}_g), \tag{8}$$

using gradient-based optimization for E_d distillation epochs at the server, with learning rate η_{distill}. The final server parameters become $\boldsymbol{\theta}_g^{(t+1)}$.

5 Experiments

5.1 Main Experimental Setup

Datasets and Architecture We evaluate our approach across computer vision (CV) and natural language processing (NLP) tasks. For CV, we perform image classification on CIFAR-10/100 [23] and CINIC-10 [10]. For NLP, we fine-tune pre-trained models on MNLI [44], SST-2 [41] and MARC [22]. We employ ResNet-8 for CIFAR-10, ResNet-18 for CIFAR-100 and CINIC-10, and Tiny BERT [19] for NLP tasks. Data heterogeneity is simulated via Dirichlet distribution [18] with concentration parameters $\beta \in \{0.3, 0.5\}$.

Federated Learning Setup. For CV tasks, we use 100 clients for CIFAR-10, and CINIC-10 datasets and 50 clients for CIFAR-100 dataset, fixing client sampling rate to 10% for all cases. Training spans 100 rounds for CIFAR datasets and 60 for CINIC-10, and local epoch is fixed to 20 for all cases. NLP experiments involve 15 clients with 30% client sample rate, 1 local epoch, and 15 communication rounds. For server-side distillation, we utilized auxiliary datasets: CIFAR-100 for CIFAR-10/CINIC-10, ImageNet-100 [11] for CIFAR-100, SNLI [2] for MNLI, Sentiment140 [14] for SST-2, and Yelp [49] for MARC.

Implementation Details. The code is implemented in PyTorch 2.4.1 and executed on NVIDIA RTX 3090 GPUs, using the FedZoo benchmark [33]. The implementation is anonymously available at `https://anonymous.4open.science/r/FedProj_Neurips-63C1`. We use Adam optimizer with learning rate 0.001 for CV and 3×10^{-5} for NLP tasks. Server-side distillation employs KL divergence loss with temperature $T = 3$, performed for 1 epoch (CIFAR-10, CINIC-10, all NLP) or 3 epochs (CIFAR-100) with batch sizes of 256 (CV) and 128 (NLP).

Baselines and Evaluation. We compare FEDPROJ against established methods: FEDAVG [32], FEDPROX [26], FEDNOVA [42], FEDDF [31], FEDET [8], MOON [25], FEDDYN [1], and FEDRCL [39]. Results report average performance and standard deviation across three independent runs with different random seeds, evaluated using global classification accuracy on held-out test sets.

5.2 Main Experimental Results

Performance on CV Task. Table 1 presents performance across datasets under non-i.i.d. conditions ($\beta = 0.3$ and $\beta = 0.5$). FedProj consistently outperforms all baselines, achieving 65.52% and 69.88% on CIFAR-10, 35.27% and 38.06% on CIFAR-100, and 41.46% and 41.63% on CINIC-10, respectively. While MOON performs well under moderate heterogeneity ($\beta = 0.5$), it degrades sharply under higher skew ($\beta = 0.3$). FedProj maintains robust performance through its projection-based update rule, which aligns local gradients with global descent direction. On the more complex CIFAR-100, alternatives like FedNova and FedDyn exhibit lower accuracy and higher variance. For CINIC-10, which introduces domain shift, FedProj outperforms distillation-based methods like FedDF and FedET. These results demonstrate that FedProj's integration of global memory, gradient projection, and dual-mode distillation effectively addresses client heterogeneity.

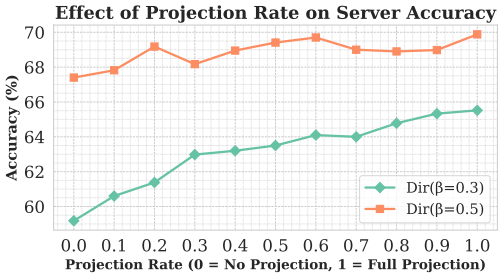
Table 1: **Performance on CIFAR-10, CIFAR-100, and CINIC-10** under Dirichlet($\beta = 0.3, 0.5$).

Baseline	CIFAR-10		CIFAR-100		CINIC-10	
	Dir(β=0.3)	Dir(β=0.5)	Dir(β=0.3)	Dir(β=0.5)	Dir(β=0.3)	Dir(β=0.5)
FedAvg [32]	63.19 $\pm$ 1.48	66.41 $\pm$ 0.56	33.72 $\pm$ 0.17	37.18 $\pm$ 0.09	40.59 $\pm$ 0.12	40.70 $\pm$ 0.18
FedProx [26]	61.40 $\pm$ 0.92	67.34 $\pm$ 0.36	33.96 $\pm$ 0.88	36.66 $\pm$ 0.49	40.69 $\pm$ 0.07	40.80 $\pm$ 0.17
FedNova [42]	63.43 $\pm$ 0.99	67.93 $\pm$ 0.49	33.40 $\pm$ 0.55	36.40 $\pm$ 0.48	39.81 $\pm$ 0.25	40.01 $\pm$ 0.18
FedDyn [1]	63.35 $\pm$ 1.03	67.53 $\pm$ 0.71	33.61 $\pm$ 0.36	36.52 $\pm$ 0.39	40.43 $\pm$ 0.11	40.59 $\pm$ 0.10
MOON [25]	61.09 $\pm$ 1.36	68.83 $\pm$ 0.78	30.29 $\pm$ 0.71	34.49 $\pm$ 0.09	40.64 $\pm$ 0.10	40.79 $\pm$ 0.14
FedRCL [39]	62.14 $\pm$ 0.51	67.26 $\pm$ 0.87	33.91 $\pm$ 0.20	36.77 $\pm$ 0.11	40.72 $\pm$ 0.08	40.81 $\pm$ 0.16
FedDF [31]	61.32 $\pm$ 2.02	67.44 $\pm$ 0.83	33.65 $\pm$ 0.65	36.60 $\pm$ 0.30	39.34 $\pm$ 0.37	39.57 $\pm$ 0.11
FedET [8]	58.79 $\pm$ 1.20	66.46 $\pm$ 0.33	32.85 $\pm$ 0.31	36.21 $\pm$ 0.14	39.11 $\pm$ 0.21	39.20 $\pm$ 0.12
FedProj	**65.52** $\pm$ 0.86	**69.88** $\pm$ 0.03	**35.27** $\pm$ 0.11	**38.06** $\pm$ 0.21	**41.46** $\pm$ 0.55	**41.63** $\pm$ 0.21

Table 2: **Performance Results for NLP Task on MNLI, SST-2 and MARC.**

Private	Public	Baseline	Dir(β=0.3)	Dir(β=0.5)
MNLI [44]	SNLI [2]	FedAvg	35.67$\pm$1.21	41.91$\pm$3.98
		FedDF	36.65$\pm$1.32	41.07$\pm$5.88
		FedET	36.10$\pm$3.34	36.50$\pm$3.39
		FedProj	**44.38**$\pm$3.91	**45.13**$\pm$3.10
SST2 [41]	Sent140 [14]	FedAvg	56.96$\pm$1.36	55.08$\pm$6.46
		FedDF	51.43$\pm$2.19	54.45$\pm$3.86
		FedET	54.96$\pm$8.31	56.36$\pm$9.42
		FedProj	**64.80**$\pm$5.1	**65.98**$\pm$2.87
MARC [22]	Yelp [49]	FedAvg	37.21$\pm$2.85	40.86$\pm$2.89
		FedDF	40.74$\pm$2.91	38.40$\pm$6.05
		FedET	37.02$\pm$3.39	40.05$\pm$2.94
		FedProj	**45.15**$\pm$1.59	**46.52**$\pm$4.42

Performance on NLP Task. Table 2 presents results on three NLP private-public dataset pairs: MNLI–SNLI, SST-2–Sentiment140, and MARC–Yelp, under two Dirichlet non-IID settings (Dir(β=0.3) and Dir(β=0.5)). Similar to CV results, on NLP experiments FedProj consistently outperforms existing baselines, demonstrating superior robustness across heterogeneous client scenarios. On MNLI [44], paired with SNLI [2], FedProj delivers substantial gains. Under Dir(β=0.3), it achieves 44.38 accuracy, surpassing FedDF (36.65) and FedET (36.10), with an improvement exceeding 7% compared to FedAvg. These results expose the limitations of standard aggregation approaches under skewed data distributions. FedET underperforms FedAvg, revealing the downside of relying on uncertainty estimates from pretrained language models, which produce overconfident and poorly calibrated predictions [16, 43, 46]. For SST-2 [41]–Sentiment140 [14], FedProj delivers state-of-the-art accuracy across both settings, improving by nearly 9% over the strongest baseline. This proves FedProj's effectiveness in bridging domain gaps between curated sentiment labels and noisy social media text. On MARC [22]–Yelp [49], involving multilingual and domain-diverse reviews, FedProj maintains its dominance with 4–6% gains across both non-IID scenarios. These



(a) **Impact of Projection Dropout.**

Dataset	WD	Dir(β=0.3)	Dir(β=0.5)
CIFAR10	0.1	64.12±1.42	66.57±1.43
	0.3	63.31±0.29	**69.88**±0.03
	0.5	63.07±2.38	67.12±1.85
	w/o	**65.52**±0.86	67.23±0.66
CIFAR100	0.1	34.71±0.16	37.67±0.14
	0.3	33.49±0.24	36.99±0.16
	0.5	33.21±0.31	**38.06**±0.21
	w/o	**35.27**±0.11	36.98±0.41
CINIC10	0.1	39.85±0.34	40.97±0.22
	0.3	40.11±0.21	**41.63**±0.21
	0.5	40.22±0.36	41.25±0.18
	w/o	**41.46**±0.55	41.19±0.27

(b) **Impact of Weight Divergence (WD).**

results establish FedProj's effectiveness against both linguistic and domain shifts in realistic federated NLP applications.

5.3 Ablation Studies

Impact of Gradient Projection. We quantify client-side gradient projection's critical role on CIFAR-10, where projection is randomly omitted during client updates with varying rates. As shown in Fig. 3a, under high heterogeneity (Dir($\beta = 0.3$)), server accuracy increases sharply as projection rate increases, with full projection delivering superior performance. This proves that projection enforces alignment between local updates and the global objective, resulting into updates that directly minimize local loss while preserving the global knowledge. Even slight projection removal triggers substantial degradation as observed. Under lower data heterogeneity (Dir($\beta = 0.5$)), performance show stability at moderate projection levels but declines precipitously when projection is largely eliminated. These results establish projection as essential for global model alignment, especially in highly heterogeneous settings.

Impact of Weight Divergence Regularization. Table 3b presents the effect of L2 regularization on server-side performance for CIFAR-10, CIFAR-100, and CINIC-10. For CIFAR-10, the optional use of L2 regularization ($\lambda = 0$) achieves the highest accuracy for $\beta = 0.3$, while a moderate value ($\lambda = 0.3$) performs best for $\beta = 0.5$. In CIFAR-100, $\lambda = 0.5$ yields the best result for $\beta = 0.5$, whereas omitting L2 regularization is preferable for $\beta = 0.3$. CINIC-10 shows minimal variation across values, indicating lower sensitivity to L2 regularization. Overall, the improvements from L2 regularization are marginal; the primary performance gains are attributed to knowledge fusion and the use of client-side gradient projection.

6 Conclusion

In this work, we propose FedProj, a federated learning framework designed to tackle catastrophic forgetting and enhance knowledge retention in non-IID settings. Specifically, FedProj consists of two key components: (1) a client-side gradient projection mechanism that preserves global knowledge by constraining local updates, preventing overfitting to client-specific objectives, and (2) a server-side knowledge distillation process that fuses local models through ensemble distillation on a small, auxiliary public dataset, effectively aligning their logits and feature representations. We conduct extensive experiments on benchmark datasets and demonstrate that FedProj outperforms state-of-the-art FL methods in terms of accuracy and stability on non-IID data, validating its effectiveness in preserving global knowledge while accommodating local adaptations. The limitations of our work are lack of real-world implementation and relying on public dataset which we will pursue in the future work.

References

[1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. *arXiv preprint arXiv:2111.04263*, 2021.

[2] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.

[3] Vishnu Pandi Chellapandi, Antesh Upadhyay, Abolfazl Hashemi, and Stanislaw H Żak. On the convergence of decentralized federated learning under imperfect information sharing. *IEEE Control Systems Letters*, 2023.

[4] Vishnu Pandi Chellapandi, Liangqi Yuan, Christopher G. Brinton, Stanislaw H Zak, and Ziran Wang. Federated learning for connected and automated vehicles: A survey of existing approaches and challenges. *IEEE Transactions on Intelligent Vehicles*, November 2023.

[5] Vishnu Pandi Chellapandi, Liangqi Yuan, Stanislaw H Zak, and Ziran Wang. A survey of federated learning for connected and automated vehicles. *arXiv preprint arXiv:2303.10677*, 2023.

[6] Vishnu Pandi Chellapandi, Antesh Upadhyay, Abolfazl Hashemi, and Stanislaw H Żak. Fednmut – federated noisy model update tracking convergence analysis. *arXiv preprint arXiv:2403.13247*, 2024.

[7] Sijie Cheng, Jingwen Wu, Yanghua Xiao, and Yang Liu. Fedgems: Federated learning of larger server models via selective knowledge fusion. *arXiv preprint arXiv:2110.11027*, 2021.

[8] Yae Jee Cho, Andre Manoel, Gauri Joshi, Robert Sim, and Dimitrios Dimitriadis. Heterogeneous ensemble knowledge transfer for training large models in federated learning. *arXiv preprint arXiv:2204.12703*, 2022.

[9] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *International conference on machine learning*, pages 2089–2099. PMLR, 2021.

[10] Luke N Darlow, Elliot J Crowley, Antreas Antoniou, and Amos J Storkey. Cinic-10 is not imagenet or cifar-10. *arXiv preprint arXiv:1810.03505*, 2018.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[12] 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.

[13] Margalit R Glasgow, Honglin Yuan, and Tengyu Ma. Sharp bounds for federated averaging (local sgd) and continuous perspective. In *International Conference on Artificial Intelligence and Statistics*, pages 9050–9090. PMLR, 2022.

[14] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009, 2009.

[15] Xuan Gong, Abhishek Sharma, Srikrishna Karanam, Ziyan Wu, Terrence Chen, David Doermann, and Arun Innanje. Preserving privacy in federated learning with ensemble cross-domain knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11891–11899, 2022.

[16] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.

[17] Chaoyang He, Murali Annavaram, and Salman Avestimehr. Group knowledge transfer: Federated learning of large cnns at the edge. *Advances in neural information processing systems*, 33:14068–14080, 2020.

[18] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

[19] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*, 2019.

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

[21] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pages 5132–5143. PMLR, 2020.

[22] Phillip Keung, Yichao Lu, György Szarvas, and Noah A Smith. The multilingual amazon reviews corpus. *arXiv preprint arXiv:2010.02573*, 2020.

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

[24] Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019.

[25] 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.

[26] 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.

[27] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International conference on machine learning*, pages 6357–6368. PMLR, 2021.

[28] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

[29] Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. Fedbn: Federated learning on non-iid features via local batch normalization. *arXiv preprint arXiv:2102.07623*, 2021.

[30] Sen Lin, Li Yang, Deliang Fan, and Junshan Zhang. Trgp: Trust region gradient projection for continual learning. *arXiv preprint arXiv:2202.02931*, 2022.

[31] 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.

[32] 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.

[33] Mahdi Morafah, Weijia Wang, and Bill Lin. A practical recipe for federated learning under statistical heterogeneity experimental design. *IEEE Transactions on Artificial Intelligence*, 5 (4):1708–1717, 2023.

[34] Mahdi Morafah, Hojin Chang, and Bill Lin. Large scale delocalized federated learning over a huge diversity of devices in emerging next-generation edge intelligence environments. In *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*, pages 1–8, 2024.

[35] Mahdi Morafah, Matthias Reisser, Bill Lin, and Christos Louizos. Stable diffusion-based data augmentation for federated learning with non-iid data. *arXiv preprint arXiv:2405.07925*, 2024.

[36] Majid Morafah and Mahdi Morafah. Clustered federated learning: A review. *Federated Learning-A Systematic Review*, 2025.

[37] Gobinda Saha and Kaushik Roy. Continual learning with scaled gradient projection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 9677–9685, 2023.

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

[39] Seonguk Seo, Jinkyu Kim, Geeho Kim, and Bohyung Han. Relaxed contrastive learning for federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12279–12288, 2024.

[40] Aviv Shamsian, Aviv Navon, Ethan Fetaya, and Gal Chechik. Personalized federated learning using hypernetworks. In *International conference on machine learning*, pages 9489–9502. PMLR, 2021.

[41] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[42] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33:7611–7623, 2020.

[43] Yuxia Wang, Daniel Beck, Timothy Baldwin, and Karin Verspoor. Uncertainty estimation and reduction of pre-trained models for text regression. *Transactions of the Association for Computational Linguistics*, 10:680–696, 2022.

[44] Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017.

[45] Blake E Woodworth, Kumar Kshitij Patel, and Nati Srebro. Minibatch vs local sgd for heterogeneous distributed learning. *Advances in Neural Information Processing Systems*, 33: 6281–6292, 2020.

[46] Yuxin Xiao, Paul Pu Liang, Umang Bhatt, Willie Neiswanger, Ruslan Salakhutdinov, and Louis-Philippe Morency. Uncertainty quantification with pre-trained language models: A large-scale empirical analysis. *arXiv preprint arXiv:2210.04714*, 2022.

[47] Xin Yang, Hao Yu, Xin Gao, Hao Wang, Junbo Zhang, and Tianrui Li. Federated continual learning via knowledge fusion: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 36(8):3832–3850, 2024.

[48] Guanxiong Zeng, Yang Chen, Bo Cui, and Shan Yu. Continual learning of context-dependent processing in neural networks. *Nature Machine Intelligence*, 1(8):364–372, 2019.

[49] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

[50] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *International conference on machine learning*, pages 12878–12889. PMLR, 2021.

[51] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *International conference on machine learning*, pages 12878–12889. PMLR, 2021.

A Appendix

A.1 Detailed Derivation and Proof for Gradient Projection

We provide a detailed mathematical derivation of the gradient projection step used in FedProj.

Lemma 1 (Preservation of Global Knowledge). Given gradients g_{new} from the local loss and g_{glob} from the memory-based global knowledge loss, the projection given by (14) is the optimal solution to the constrained optimization problem stated in (4).

Proof. Consider the optimization problem:

$$\min_{g_{\text{proj}}} \quad \frac{1}{2}\|g_{\text{new}} - g_{\text{proj}}\|_2^2, \quad \text{s.t.} \quad \langle g_{\text{proj}}, g_{\text{glob}} \rangle \geq 0. \tag{9}$$

We introduce the Lagrangian:

$$\mathcal{L}(g_{\text{proj}}, \lambda) = \frac{1}{2}\|g_{\text{new}} - g_{\text{proj}}\|_2^2 - \lambda \langle g_{\text{proj}}, g_{\text{glob}} \rangle,$$

with the Karush-Kuhn-Tucker (KKT) optimality conditions:

$$\frac{\partial \mathcal{L}}{\partial g_{\text{proj}}} = g_{\text{proj}} - g_{\text{new}} - \lambda g_{\text{glob}} = 0, \tag{10}$$

$$\lambda \geq 0, \tag{11}$$

$$\lambda \langle g_{\text{proj}}, g_{\text{glob}} \rangle = 0, \quad \langle g_{\text{proj}}, g_{\text{glob}} \rangle \geq 0. \tag{12}$$

From the first KKT condition, we have:

$$g_{\text{proj}} = g_{\text{new}} + \lambda g_{\text{glob}}. \tag{13}$$

To find λ, we consider two cases:

- **Case 1 ($\lambda = 0$):** If $\langle g_{\text{new}}, g_{\text{glob}} \rangle \geq 0$, then $g_{\text{proj}} = g_{\text{new}}$ directly satisfies the constraint, making it the optimal solution.
- **Case 2:** If $\langle g_{\text{new}}, g_{\text{glob}} \rangle < 0$, the constraint becomes active:

$$\langle g_{\text{proj}}, g_{\text{glob}} \rangle = 0.$$

Substituting (13), we obtain:

$$\langle g_{\text{new}} + \lambda g_{\text{glob}}, g_{\text{glob}} \rangle = 0 \quad \Rightarrow \quad \lambda = -\frac{\langle g_{\text{new}}, g_{\text{glob}} \rangle}{\|g_{\text{glob}}\|^2 + \epsilon}.$$

Hence, the optimal projected gradient is:

$$g_{\text{proj}} = g_{\text{new}} - \frac{\langle g_{\text{new}}, g_{\text{glob}} \rangle}{\|g_{\text{glob}}\|^2 + \epsilon} g_{\text{glob}}, \tag{14}$$

matching Eq. (14). Thus, the derived gradient projection update guarantees that local model updates do not negatively affect global knowledge retention.

Algorithm 1 FedProj: Federated Learning with Gradient Projection and Distillation

Require: Number of rounds T, local learning rate η_{local}, distillation rate η_{distill}, local epochs E, distillation epochs E_d, public dataset $\mathcal{D}_{\text{pub}}$, memory-based gradient constraint.

Initialize server model parameters $\boldsymbol{\theta}_g^{(0)}$.
for $t = 0, \ldots, T-1$ **do**
  Sample a subset $\mathcal{S}_t$ of clients
  for all $k \in \mathcal{S}_t$ **in parallel do**
    $\boldsymbol{\theta}_k \leftarrow \boldsymbol{\theta}_g^{(t)}$ *Initialize local model*
    for epoch $e = 1, \ldots, E$ **do**
      for mini-batch $(\mathbf{x}, \mathbf{y}) \subseteq \mathcal{D}_k$ **do**
        $g_{\text{new}} \leftarrow \nabla_{\boldsymbol{\theta}_k} \ell\big(f(\mathbf{x}; \boldsymbol{\theta}_k),\, \mathbf{y}\big)$
        $g_{\text{glob}} \leftarrow \nabla_{\boldsymbol{\theta}_k} \Big[\frac{1}{|\mathcal{D}_{\text{pub}}|} \sum_{(\mathbf{x}_m, \mathbf{y}_m) \in \mathcal{D}_{\text{pub}}} \ell\big(f(\mathbf{x}_m; \boldsymbol{\theta}_k), \mathbf{y}_m\big)\Big]$
        Project as in (14): $g_{\text{proj}} \leftarrow g_{\text{new}} - \frac{\langle g_{\text{new}},\, g_{\text{glob}} \rangle}{\|g_{\text{glob}}\|^2 + \epsilon}\, g_{\text{glob}}$
        $\boldsymbol{\theta}_k \leftarrow \boldsymbol{\theta}_k - \eta_{\text{local}}\, g_{\text{proj}}$
      end for
    end for
    $\boldsymbol{\theta}_k^{(t+1)} \leftarrow \boldsymbol{\theta}_k$
  end for
  // Server aggregation & distillation
  $\boldsymbol{\theta}_g^{\text{FedAvg}} \leftarrow \sum_{k \in \mathcal{S}_t} \Big(\frac{|\mathcal{D}_k|}{\sum_{j \in \mathcal{S}_t} |\mathcal{D}_j|}\Big) \boldsymbol{\theta}_k^{(t+1)}$
  Initialize $\boldsymbol{\theta}_g \leftarrow \boldsymbol{\theta}_g^{\text{FedAvg}}$
  for epoch $e_d = 1, \ldots, E_d$ **do**
    for mini-batch $\mathbf{X} \subseteq \mathcal{D}_{\text{pub}}$ **do**
      Compute teacher logits:

$$\mathcal{Z}_{\text{teacher}} = \frac{1}{|\mathcal{S}_t|} \sum_{k \in \mathcal{S}_t} f(\mathbf{X};\, \boldsymbol{\theta}_k^{(t+1)})$$

      $\mathcal{Z}_{\text{student}} = f(\mathbf{X}; \boldsymbol{\theta}_g)$
      $\mathcal{L}_{\text{KD}} = T^2 \cdot \text{KL}\big(\sigma(\frac{\mathcal{Z}_{\text{student}}}{T}),\, \sigma(\frac{\mathcal{Z}_{\text{teacher}}}{T})\big)$
      $\mathcal{L}_{\text{div}} = \alpha\, \|\boldsymbol{\theta}_g - \boldsymbol{\theta}_g^{(t)}\|_2^2$
      $\boldsymbol{\theta}_g \leftarrow \boldsymbol{\theta}_g - \eta_{\text{distill}} \nabla_{\boldsymbol{\theta}_g} \big[\mathcal{L}_{\text{KD}} + \mathcal{L}_{\text{div}}\big]$
    end for
  end for
  $\boldsymbol{\theta}_g^{(t+1)} \leftarrow \boldsymbol{\theta}_g$
end for

A.2 Full Algorithm Description of FedProj

The full algorithm description of FedProj is presented in Algorithm 1.