Spatio-Temporal Gradient Matching for Federated Continual Learning

Minh-Duong Nguyen [*1,2] **Le-Tuan Nguyen** [*1] **Quoc-Viet Pham** [3]

Abstract

Federated Continual Learning (FCL) has emerged as an important research area, as data from distributed clients often arrives in a streaming manner and requires sequential learning. In this paper, we consider a more practical and challenging FCL setting where clients may have unrelated or even conflicting tasks. In such scenarios, statistical heterogeneity and data noise can lead to spurious correlations, biased feature learning, and severe catastrophic forgetting. Existing FCL methods often rely on generative replay to reconstruct previous tasks, but these approaches suffer from task divergence and forgetting themselves, which results in overfitting and degraded performance. To address these challenges, we propose a novel approach called **S**patio-**T**emporal gr**A**dient **M**atching with rehearsal data**P**ool (STAMP). Our key idea is to perform unified gradient matching across both the spatial and temporal dimensions of FCL. Spatial matching aligns gradients across clients at the same time step, while temporal matching aligns gradients across sequential tasks within each client. This dual perspective mitigates negative transfer and improves knowledge retention across diverse and evolving tasks. Extensive experiments show that STAMP outperforms existing FCL methods under heterogeneous conditions.

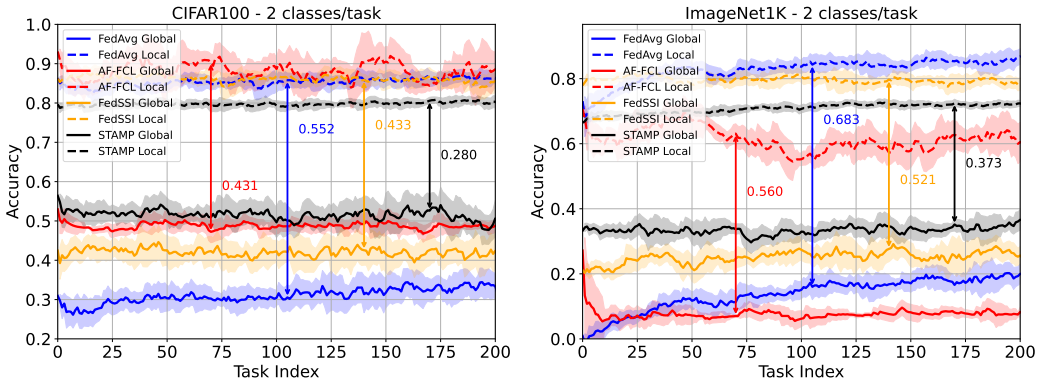
1. Introduction

Federated Learning (FL) is a distributed and privacy-preserving learning paradigm that enables collaboration among multiple entities, such as devices or organizations, without sharing raw data (Lim et al., 2020; Le et al., 2025). Due to the robustness of distributed machine learning, this approach supports the capabilities of the emerging Internet of Agents (IoA) (Wang et al., 2025). In FL, a central server coordinates the training of a global model by aggregating updates from distributed clients. This approach has gained widespread adoption across various domains, including healthcare (Nguyen et al., 2023a), Internet-of-Things (Nguyen et al., 2023b; 2022), autonomous driving (Fantauzzo et al., 2022; Tung et al., 2025), Internet-of-Agents (Wang et al., 2025), and large language models (Tran et al., 2025). However, most existing FL methods assume a fixed set of classes and stable data distributions, which rarely hold in real-world applications where new classes and shifts in data naturally occur over time (Elsayed & Mahmood, 2024). Continuously training new models from scratch to adapt to these changes is computationally inefficient. While transfer learning offers a more efficient alternative by leveraging pre-trained models, it often suffers from catastrophic forgetting (Dong et al., 2024), where the model loses previously acquired knowledge. To tackle this issue, recent studies (Luo et al., 2023) have proposed Federated Continual Learning (FCL), which combines the strengths of FL and Continual Learning (CL) (Yang et al., 2023) to support sequential learning in distributed environments.

In FCL, clients collaboratively learn models for their own private, sequential tasks while ensuring data privacy. However, the inherently sequential nature of these tasks means that each client typically only has access to a limited amount of data for the current task. This limitation often leads to forgetting previously learned knowledge, a phenomenon known as catastrophic forgetting. In contrast to conventional settings, we investigate a more demanding variant of FCL, referred to as heterogeneous FCL. This scenario introduces two key challenges not addressed by standard FCL methods. First, clients in heterogeneous FCL often work on entirely different tasks simultaneously, resulting in a highly non-uniform learning environment (see Figure 1b). This variability causes domain shifts in the aggregated global model after each communication round, hindering stable convergence. Second, existing Federated Class-Incremental Learning (FCIL) approaches struggle under heterogeneous FCL conditions. These methods typically rely on the assumption of a shared, consistent class distribution across clients, allowing for the incremental addition of task-specific output

[*]Equal contribution [1]College of Engineering and Computer Science, VinUniversity, Hanoi, Vietnam [2]a part of this work was done when Minh-Duong Nguyen was with Pusan National University [3]School of Computer Science and Statistics, Trinity College Dublin, Dublin 2, D02PN40, Ireland. Correspondence to: Quoc-Viet Pham <viet.pham@tcd.ie>.

CFAgentic@ICML2025, Vancouver, Canada. PMLR 267, 2025.



(a) Generalization gap on CIFAR100

(b) Generalization gap on ImageNet1K

Figure 1: The illustration highlights the challenges encountered by current FCL methods (e.g., AF-FCL (Wuerkaixi et al., 2024), FedSSI (Li et al., 2025a)) when applied in heterogeneous settings. A notable gap between local and global test accuracy arises due to client-specific data heterogeneity within each task at every time step. STAMP demonstrates superior robustness over current baselines by mitigating inter-client divergence throughout the learning process, leading to a reduced local-global generalization gap.

heads. However, in heterogeneous FCL, clients encounter distinct class sets at each stage, leading to misaligned local model architectures. This mismatch significantly reduces the effectiveness of current FCIL techniques.

To address the aforementioned challenges, we propose a novel method, dubbed Federated Continual Learning via **S**patio-**T**emporal gr**A**dient **M**atching with rehearsal data**P**ool (STAMP). Our key contributions are as follows:

- We address a more practical and challenging Federated Continual Learning (FCL) scenario where clients learn from non-identical and sequential tasks under significant statistical heterogeneity.
- We propose **S**patio-**T**emporal gr**A**dient **M**atching with rehearsal data**P**ool (STAMP), a novel model-agnostic framework that explicitly aligns gradients across both spatial (inter-client) and temporal (intra-client) dimensions. STAMP facilitates stable and generalizable learning by reducing negative transfer across clients and tasks, enabling more reliable knowledge accumulation.
- We validate STAMP through extensive experiments on a series of benchmark datasets under heterogeneous task distributions. Our results demonstrate the effectiveness and superiority of STAMP over prior state-of-the-art FCL methods.

2. Backgrounds & Preliminaries

2.1. Federated Continual Learning

Federated Continual Learning (FCL) describes a realistic learning paradigm that integrates the principles of Federated Learning (FL) and Continual Learning (CL). Consider a system with U clients. Each client u learns over a sequence of T tasks. At any step indexed by $t \times R + r$, where R denotes the number of communication rounds allocated per task and r is the current round within task t, client u maintains local model parameters $\theta_u^{t,r}$ and has access only to data associated with task t. On client u, the dataset for task t is denoted as $\mathcal{D}_u^t = \{(x_i^t, y_i^t)\}_{i=1}^{N_u^t}$, where N_u^t is the number of labeled samples available for that task.

Most prior works have primarily focused on a task reshuffling setting, where all clients share the same task set but encounter the tasks in different orders (Yoon et al., 2021). However, in real-world applications, the task distributions across clients may not be correlated. That is, the sequence of tasks $\{\mathcal{D}_u^1, \mathcal{D}_u^2, \ldots, \mathcal{D}_u^T\}$ for client u may not exhibit any clear relationship, nor is there any guaranteed similarity among tasks $\{\mathcal{D}_1^t, \mathcal{D}_2^t, \ldots, \mathcal{D}_U^t\}$ across clients. Therefore, we adopt a more practical and generalized scenario called the Limitless Task Pool (LTP).

Limitless Task Pool. In the Limitless Task Pool (LTP) setting, tasks are randomly sampled from a large and diverse repository, leading to scenarios where two clients may have entirely disjoint task sets (i.e., $\{\mathcal{D}_u^i\}_{i=1}^{t_u} \cap \{\mathcal{D}_v^i\}_{i=1}^{t_v} = \varnothing,\ \forall u, v \in \{1, 2, \ldots, U\}$). Furthermore, due to statistical heterogeneity, each client may have a distinct joint data-label distribution $p(x, y)$, meaning that knowledge transferred from one client may introduce biases when applied

to another client's tasks.

At each task t, the objective is to collaboratively train a global model with parameters θ^t while preserving performance on prior tasks. Under the privacy-preserving constraints of FL and CL, the goal is to balance learning the current task effectively while retaining knowledge of previous tasks across all clients. This can be formalized as:

$$\min_{\theta^t} [\mathcal{S}_1^t, \mathcal{S}_2^t, \dots, \mathcal{S}_U^t],$$
$$\text{where } \mathcal{S}_u^t = \left[\mathcal{L}(\theta^t; \mathcal{D}_u^1), \mathcal{L}(\theta^t; \mathcal{D}_u^2), \dots, \mathcal{L}(\theta^t; \mathcal{D}_u^t)\right]. \quad (1)$$

However, due to the limited storage and computational capabilities of client devices, maintaining full access to previous task data $\mathcal{D}_u^{[1:t-1]}$ is impractical. Consequently, while learning task t, client u cannot directly minimize the cumulative empirical loss over all past tasks, i.e., $\sum_{i=1}^{t} \mathcal{L}(\theta_u^t; \mathcal{D}_u^i)$. Additionally, task-specific and client-specific data heterogeneity often leads to domain and label distribution shifts, resulting in conflicting gradients during training (Nguyen et al., 2025).

2.2. Gradient Matching

When learning from diverse and non-identical tasks, one of the most critical challenges is the presence of gradient conflict.

Definition 2.1 (Gradient Conflict)**.** Given two task-specific gradients g_i and g_j $(i \neq j)$, a gradient conflict occurs if their cosine similarity is negative, i.e., $\cos(g_i, g_j) = \frac{g_i \cdot g_j}{\|g_i\| \cdot \|g_j\|} < 0$. In such cases, updating the model in the direction of g_i would negatively impact the performance with respect to task j, and vice versa.

To alleviate gradient conflict as described in Definition 2.1, we adopt the Gradient Matching (GM) technique proposed in (Nguyen et al., 2025). This method computes a task-aligned gradient that preserves consistency across tasks while reducing interference:

$$\text{GM}(\mathbf{g}^{(r)}) = \frac{\kappa\|\bar{g}^{(r)}\|}{\|\Gamma^* \mathbf{g}^{(r)}\|}\Gamma^* \mathbf{g}^{(r)}$$
$$\text{s.t.} \quad \Gamma^* = \arg\min_{\Gamma} \Gamma \mathbf{g}^{(r)} \cdot \bar{g}^{(r)} + \kappa\|\bar{g}^{(r)}\|\|g_\Gamma^{(r)}\|, \quad (2)$$
$$\bar{g}^{(r)} = \frac{1}{|\mathcal{T}|}\sum_{t \in \mathcal{T}} g_t^{(r)},$$

where $\mathbf{g}^{(r)} = [g_t^{(r)} \mid t \in \mathcal{T}]$ denotes the set of task-specific gradients participating in the current training round. The aggregated gradient $g_G = \text{GM}(\mathbf{g}^{(r)})$ leverages the directional information from multiple tasks to maintain the task-invariant components of individual gradients. By ensuring that $g_G \cdot g_i \leq 0$ for all $i \in \mathcal{T}$, the method guarantees that the resulting gradient avoids negative transfer. As a result, the GM approach enables better generalization across tasks within the continual learning framework.

3. Proposed Method

We introduce a novel framework, **STAMP**, designed for heterogeneous federated continual learning (FCL). The STAMP framework consists of two key components: (1) *on-client temporal gradient matching*, (2) *on-server spatial gradient matching*.

3.1. Temporal Gradient Matching

Temporal Gradient Matching is performed locally on each client during training. Specifically, the client leverages task-specific gradients from both past and current tasks to solve a gradient matching optimization problem. The local model update is defined as follows:

$$\theta_u^{t,r+1} = \theta_u^{t,r} - \text{GM}(\mathbf{g}_u^{[0:t]}), \quad (3)$$

where $\mathbf{g}_u^{[0:t]} = [g_u^i \mid i \in \{1, 2, \dots, t\}]$ denotes the set of gradients corresponding to all tasks learned so far by client u, including the current task gradient g_u^t and historical gradients $\mathbf{g}_u^{[0:t-1]}$. Conventionally, such historical gradients are estimated using stored exemplars from previous tasks (Lopez-Paz & Ranzato, 2017; Luo et al., 2023; Wu et al., 2024). By explicitly incorporating gradient information from prior tasks, this component reduces the risk of catastrophic forgetting in continual learning.

3.2. Spatial Gradient Matching

Inspired by (Nguyen et al., 2025), the Spatial Gradient Matching component is executed on the server to identify a unified gradient direction that remains consistent across heterogeneous tasks received from multiple clients. This approach helps the global model stabilize its learning trajectory by mitigating the effects of task diversity and reducing negative transfer. The server-side update rule is given by:

$$\theta^{t,r+1} = \theta^{t,r} - \text{GM}(\mathbf{g}^t), \quad (4)$$
$$\text{where} \quad \mathbf{g}^t = [g_u^t \mid u \in \{1, 2, \dots, U\}], \quad (5)$$

and each local gradient g_u^t is implicitly computed via the model update, i.e., $g_u^t = \theta_u^{t,r+1} - \theta_u^{t,r}$. Notably, this formulation avoids additional communication cost, as the required gradients are derived from already transmitted model weights. By aligning the gradient directions across participating clients, spatial gradient matching alleviates the impact of client drift and enhances global model consistency under heterogeneous FCL settings.

4. Experimental Evaluations

In this section, we present extensive experiments to evaluate the effectiveness of the proposed **STAMP** framework. Implementation details and additional results are provided in Appendix B. To ensure a fair evaluation of FCL baselines

Table 1: We report the average per-task performance of FCL under a setting where each task is assigned 20 classes. Evaluations are conducted using 10 clients (fraction = 1.0) across 5 independent trials. OOM refers to the out of memory in GPU. ↑ and ↓ indicate that higher and lower values are better, respectively. C→S and S→C denote communication from the client to the server and from the server to the client, respectively.

Methods	**Accuracy ↑**	**AF ↓**	**Avg. Comp. ↓ (Sec/Round)**	**Comm. Cost ↓ C→S**	**S → C**	**GPU (Peak) ↓**	**Disk ↓**
CIFAR100 ($U = 10, C = 20$)							
FedAvg	27.2 (± 2.2)	5.9 (± 0.9)	27.6 sec	44.6 MB	44.6 MB	1.92 GB	N/A
FedALA	28.5 (± 2.4)	6.5 (± 1.2)	28.2 sec	44.6 MB	44.6 MB	1.93 GB	N/A
FedDBE	28.3 (± 1.6)	5.5 (± 0.7)	28.3 sec	44.6 MB	44.6 MB	1.91 GB	N/A
FedAS	40.2 (± 1.1)	30.7 (± 0.3)	135.7 sec	44.6 MB	44.6 MB	1.92 GB	N/A
FedOMG	36.8 (± 1.4)	8.5 (± 0.6)	32.7 sec	44.6 MB	44.6 MB	1.92 GB	N/A
GLFC	29.8 (± 2.1)	7.5 (± 0.4)	167.8 sec	88.2 MB	46.5 MB	3.83 GB	22.1 MB
FedCIL	32.4 (± 1.7)	6.3 (± 1.2)	199.3 sec	95.3 MB	44.6 MB	4.21 GB	18.5 MB
LANDER	45.1 (± 1.3)	5.4 (± 0.8)	153.6 sec	112.4 MB	138.7 MB	4.83 GB	131.5 MB
TARGET	32.1 (± 2.3)	5.9 (± 1.6)	236.4 sec	112.4 MB	44.6 MB	3.65 GB	18.5 MB
FedL2P	30.2 (± 1.8)	6.3 (± 1.3)	78.1 sec	56.3 MB	56.3 MB	2.56 GB	N/A
FedWeIT	37.3 (± 2.3)	4.7 (± 0.8)	38.7 sec	44.2 MB	44.2 MB	7.21 GB	N/A
AF-FCL	35.6 (± 0.4)	5.2 (± 0.5)	45.3 sec	156.3 MB	121.3 MB	8.93 GB	N/A
STAMP	41.3 (± 0.9)	5.4 (± 0.6)	56.3 sec	44.6 MB	44.6 MB	1.92 GB	16.3 MB
ImageNet1K ($U = 10, C = 20$)							
FedAvg	17.3 (± 3.3)	14.1 (± 0.2)	1485.2 sec	112.5 MB	112.5 MB	16.11 GB	N/A
FedALA	17.6 (± 5.6)	14.9 (± 0.8)	1556.6 sec	112.5 MB	112.5 MB	16.12 GB	N/A
FedDBE	18.8 (± 5.2)	13.9 (± 0.3)	1572.7 sec	112.5 MB	112.5 MB	16.11 GB	N/A
FedAS	22.3 (± 5.0)	18.2 (± 0.6)	5108.5 sec	112.5 MB	112.5 MB	16.11 GB	N/A
FedOMG	21.2 (± 3.3)	11.3 (± 0.7)	1821.2 sec	112.5 MB	112.5 MB	16.11 GB	N/A
GLFC	22.5 (± 2.1)	6.3 (± 0.2)	5647.3 sec	225.3 MB	121.2 MB	20.24 GB	112.6 MB
FedCIL	24.1 (± 2.8)	7.3 (± 0.4)	7120.3 sec	245.5 MB	112.5 MB	23.47 GB	184.3 MB
LANDER	31.8 (± 1.4)	7.8 (± 0.9)	6825.8 sec	267.4 MB	453.6 MB	26.54 GB	1.31 GB
TARGET	25.8 (± 3.8)	6.7 (± 0.4)	9958.2 sec	287.4 MB	112.5 MB	21.08 GB	184.3 MB
FedL2P	22.3 (± 3.7)	9.4 (± 0.6)	3278.7 sec	146.6 MB	146.6 MB	18.21 GB	N/A
FedWeIT	24.8 (± 1.3)	5.1 (± 0.8)	1763.8 sec	110.4 MB	110.4 MB	41.23 GB	61.7 GB
AF-FCL	21.3 (± 5.1)	4.5 (± 0.6)	1823.7 sec	421.3 MB	336.8 MB	46.81 GB	N/A
STAMP	26.8 (± 2.3)	5.8 (± 0.4)	3041.2 sec	112.5 MB	112.5 MB	16.11 GB	152.6 MB

under heterogeneous task settings and to accurately measure catastrophic forgetting, we deliberately avoid the use of pretrained models. This decision is motivated by the fact that commonly used pretrained backbones (e.g., those trained on ImageNet-1K) share significant overlap with our datasets, which could otherwise introduce evaluation bias.

4.1. Benchmarking with Baselines

Tables 1 and 2 present the results on the CIFAR100 and ImageNet1K datasets, both of which involve varying class distributions across tasks. In addition to average accuracy and average forgetting (AF), we evaluate several key system-level metrics: computational overhead, communication cost, GPU utilization, and disk usage. Computational overhead is defined as the average time per round, capturing the cost of client-side training, particularly for methods involving generative models. Communication cost represents the average amount of data transferred (in GB) per client-server round. GPU utilization measures peak memory usage, which is crucial in scenarios with limited hardware resources. Disk usage indicates the total client-side storage consumption, including replay buffers and task-specific model parameters. The standard FL baselines, such as FedAvg, FedALA, FedAS, FedDBE, and FedOMG, often cause the model to forget previously learned knowledge, as reflected by their high average forgetting scores. FedWeIT stores task-specific head parameters in GPU memory. However, when the number of classes (e.g., 1000 classes in ImageNet1K) and the number of tasks (e.g., 500 tasks in our ImageNet setup) are both large, the total number of parameters increases significantly. Consequently, storing all task-specific parameters in GPU memory becomes impractical, and they must be offloaded to disk, which in turn causes a notable increase in average training time. LANDER stores all generated pseudo task-specific data on disk, resulting in a client-side storage

Table 2: We report the average per-task performance of FCL under a setting where each task is assigned 2 classes. Evaluations are conducted using 10 clients (fraction = 1.0) across 5 independent trials. OOM refers to the out of memory in GPU. ↑ and ↓ indicate that higher and lower values are better, respectively. C→S and S→C denote communication from the client to the server and from the server to the client, respectively.

CIFAR100 ($U = 10, C = 2$)							
Methods	**Accuracy ↑**	**AF ↓**	**Avg. Comp. ↓ (Sec/Round)**	**Comm. Cost ↓ C→S**	**S → C**	**GPU (Peak) ↓**	**Disk ↓**
FedAvg	31.7 (± 1.7)	25.2 (± 1.3)	3.3 sec	44.6 MB	44.6 MB	1.92 GB	N/A
FedALA	36.5 (± 2.4)	27.3 (± 0.5)	3.6 sec	44.6 MB	44.6 MB	1.93 GB	N/A
FedDBE	37.0 (± 1.6)	26.1 (± 0.7)	3.6 sec	44.6 MB	44.6 MB	1.91 GB	N/A
FedAS	58.2 (± 0.1)	56.1 (± 0.1)	13.7 sec	44.6 MB	44.6 MB	1.92 GB	N/A
FedOMG	39.1 (± 1.3)	24.5 (± 0.4)	4.1 sec	44.6 MB	44.6 MB	1.92 GB	N/A
GLFC	44.8 (± 2.1)	29.5 (± 0.4)	18.3 sec	88.2 MB	46.5 MB	4.33 GB	22.1 MB
FedCIL	46.5 (± 2.2)	28.8 (± 1.2)	22.3 sec	95.3 MB	44.6 MB	4.81 GB	18.5 MB
LANDER	50.8 (± 1.3)	22.6 (± 0.4)	15.8 sec	88.2 MB	104.3 MB	5.26 GB	131.5 MB
TARGET	45.1 (± 2.4)	28.6 (± 1.6)	25.6 sec	112.4 MB	44.6 MB	3.65 GB	18.5 MB
FedL2P	48.2 (± 1.8)	28.1 (± 0.6)	8.6 sec	56.3 MB	56.3 MB	2.56 GB	N/A
FedWeIT	52.6 (± 1.3)	25.7 (± 0.9)	5.4 sec	44.5 MB	44.5 MB	5.83 GB	61.7 GB
AF-FCL	51.4 (± 0.7)	48.7 (± 1.2)	4.9 sec	156.3 MB	121.3 MB	8.93 GB	N/A
STAMP	52.8 (± 0.9)	24.3 (± 0.8)	9.1 sec	44.6 MB	44.6 MB	1.92 GB	16.3 MB
ImageNet1K ($U = 10, C = 2$)							
FedAvg	24.3 (± 5.1)	19.6 (± 0.1)	133.2 sec	112.5 MB	112.5 MB	16.11 GB	N/A
FedALA	27.2 (± 9.1)	20.3 (± 0.2)	141.6 sec	112.5 MB	112.5 MB	16.12 GB	N/A
FedDBE	29.2 (± 7.2)	19.4 (± 0.2)	142.7 sec	112.5 MB	112.5 MB	16.11 GB	N/A
FedAS	43.5 (± 4.4)	40.2 (± 0.4)	498.5 sec	112.5 MB	112.5 MB	16.11 GB	N/A
FedOMG	30.4 (± 3.8)	21.1 (± 0.7)	171.3 sec	112.5 MB	112.5 MB	16.11 GB	N/A
GLFC	31.4 (± 3.1)	27.4 (± 0.6)	466.7 sec	225.3 MB	121.2 MB	20.24 GB	112.6 MB
FedCIL	33.8 (± 3.6)	25.8 (± 0.7)	652.3 sec	245.5 MB	112.5 MB	23.47 GB	184.3 MB
LANDER	34.9 (± 2.7)	26.1 (± 0.9)	573.8 sec	267.4 MB	453.6 MB	26.54 GB	1.31 GB
TARGET	33.2 (± 4.2)	25.2 (± 0.4)	913.2 sec	287.4 MB	112.5 MB	21.08 GB	184.3 MB
FedL2P	34.5 (± 4.8)	26.4 (± 0.2)	303.7 sec	146.6 MB	146.6 MB	18.21 GB	N/A
FedWeIT	39.7 (± 3.1)	21.5 (± -)	194.2 sec	111.8 MB	111.8 MB	62.7 GB	640 GB
AF-FCL	8.3 (± 5.3)	46.6 (± 0.3)	176.7 sec	421.3 MB	336.8 MB	46.81 GB	N/A
STAMP	41.5 (± 2.4)	24.2 (± 0.8)	321.2 sec	112.5 MB	112.5 MB	16.11 GB	152.6 MB

burden similar to conventional CL methods relying on replay memory. Moreover, transmitting synthetic data from the server to clients imposes considerable communication overhead.

Key insights from Tables 1 and 2 show that more difficult scenarios, particularly those with only two classes per task, are more prone to catastrophic forgetting. This arises because each task conveys limited information about the entire dataset, leading to higher average forgetting (AF) scores. STAMP achieves the most notable improvements in both accuracy and forgetting. In addition, its communication cost remains on par with standard FL methods. STAMP also demands relatively low RAM and disk usage, making it well-suited for deployment on devices with constrained resources.

4.2. Performance under tasks with non-IID settings

Figure 2 illustrates the test accuracy across varying levels of data heterogeneity for CIFAR10, CIFAR100, Digit10, and Office31 datasets. As shown in the figure, all methods achieve higher test accuracy as data heterogeneity decreases (i.e., larger α). Notably, STAMP consistently delivers superior and stable performance across different heterogeneity levels, demonstrating its robustness under non-IID conditions.

4.3. Analysis on STAMP

4.3.1. Analysis on Temporal Gradient Matching

To assess the effectiveness of temporal gradient matching on the client side, we analyze the gradient angles produced by STAMP on CIFAR100 and ImageNet1K datasets and compare them against two sets of baseline methods: FedAvg

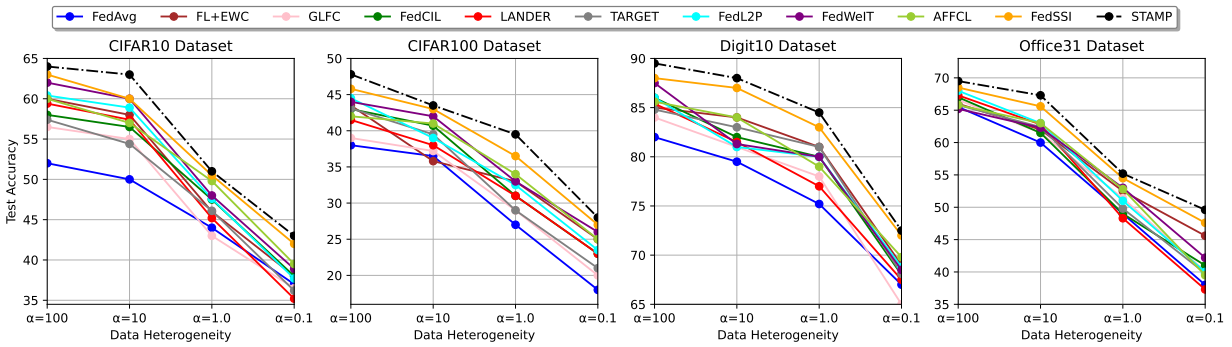
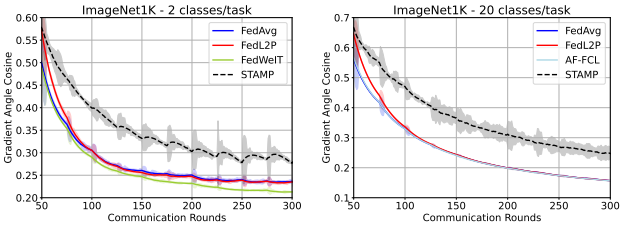
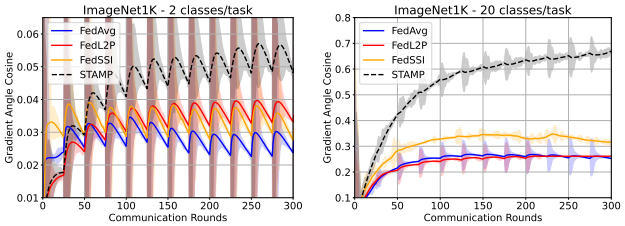


Figure 2: Performance w.r.t data heterogeneity α for four datasets.



(a) (b)

Figure 3: The figures illustrate the average temporal gradient angles across different baseline methods.



(a) (b)

Figure 4: The figures illustrate the average spatio gradient angles across different baseline methods.

and FedL2P for standard FL, and FedWeIT and AF-FCL for FCL. Figure 3 shows that STAMP achieves superior gradient alignment with previously learned tasks. This enhanced alignment indicates that STAMP is less susceptible to catastrophic forgetting compared to existing methods.

4.3.2. Efficiency of Spatio Gradient Matching

Figure 4 reports the gradient divergence for various baseline methods on CIFAR100 and ImageNet1K under two scenarios: 20 classes per task and the more challenging 2 classes per task. Unlike existing baselines that typically neglect alignment among client gradients, STAMP attains significantly better gradient alignment. This improvement enables model updates to better approximate invariant aggregated gradient directions across clients for specific tasks, thus improving the generalization of the aggregated model. This finding aligns with the reduced global-local generalization gap observed in Figure 1b.

4.4. Ablation Study on STAMP

Table 3 shows the ablation results for each component of the framework. The findings indicate that both Spatio grAdient Matching (SAM) and Temporal grAdient Matching (TAM) consistently improve average classification accuracy. In particular, SAM has a more pronounced impact on accuracy by enhancing generalization across tasks within a single communication round, whereas TAM is more crucial for lowering average forgetting by alleviating catastrophic forgetting through aligning learned gradients with those from previous tasks on the same client.

Table 3: We conduct ablation studies on the CIFAR100 and ImageNet1K datasets, using 10 clients and 2 classes per task. Specifically, **(1)** refers to spatio-temporal gradient matching performed on the server side, **(2)** denotes temporal gradient matching executed on the client side.

Method	CIFAR100		ImageNet1K	
	Acc.	AF	Acc.	AF
FedAvg	31.7 ($\pm$ 1.7)	22.1 ($\pm$ 1.3)	24.3 ($\pm$ 5.1)	19.6 ($\pm$ 0.1)
(1)	38.1 ($\pm$ 1.3)	23.8 ($\pm$ 0.4)	30.5 ($\pm$ 2.8)	26.1 ($\pm$ 0.7)
(2)	37.8 ($\pm$ 0.6)	21.7 ($\pm$ 0.9)	28.3 ($\pm$ 2.6)	23.8 ($\pm$ 0.6)
STAMP	52.8 ($\pm$ 0.9)	24.3 ($\pm$ 0.8)	41.5 ($\pm$ 2.8)	24.2 ($\pm$ 0.8)

5. Conclusion

This paper addresses the challenges of federated continual learning (FCL) in realistic scenarios marked by client data heterogeneity and task conflicts. To overcome the limitations of current generative replay-based approaches, we proposed a novel model-agnostic framework: Spatio-Temporal Gradient Matching. Our method effectively reduces catastrophic forgetting and data bias by performing gradient matching in both temporal and spatial domains. Extensive experimental results demonstrate that our approach consistently surpasses existing baselines, underscoring its effectiveness as a robust solution for FCL in diverse and dynamic settings.

References

Deng, D., Chen, G., Hao, J., Wang, Q., and Heng, P.-A. Flattening sharpness for dynamic gradient projection memory benefits continual learning. In *Adv. Neural Inform. Process. Syst.*, Dec. 2021.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *IEEE Conf. Comput. Vis. Pattern Recog.*, Aug. 2009.

Dohare, S., Hernandez-Garcia, J. F., Lan, Q., et al. Loss of plasticity in deep continual learning. *Nature*, Aug. 2024.

Dong, J., Wang, L., Fang, Z., Sun, G., Xu, S., Wang, X., and Zhu, Q. Federated class-incremental learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, Apr. 2022.

Dong, J., Zhang, D., Cong, Y., Cong, W., Ding, H., and Dai, D. Federated incremental semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 3934–3943, Jun. 2023.

Dong, J., Li, H., Cong, Y., Sun, G., Zhang, Y., and Van Gool, L. No One Left Behind: Real-World Federated Class-Incremental Learning. *IEEE Patt. Ana. and Mach. Intell.*, 46(04):2054–2070, Apr. 2024. ISSN 1939-3539.

Elsayed, M. and Mahmood, A. R. Addressing loss of plasticity and catastrophic forgetting in continual learning. In *Int. Conf. Learn. Represent.*, May 2024.

Fantauzzo, L., Fanì, E., Caldarola, D., Tavera, A., Cermelli, F., Ciccone, M., and Caputo, B. FedDrive: Generalizing federated learning to semantic segmentation in autonomous driving. Oct. 2022.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. Technical Report.

Le, M., Huynh-The, T., Do-Duy, T., Vu, T.-H., Hwang, W.-J., and Pham, Q.-V. Applications of distributed machine learning for the internet-of-things: A comprehensive survey. *IEEE Comm. Surveys & Tutorials*, 27(2):1053–1100, 2025.

Lee, R., Kim, M., Li, D., Qiu, X., Hospedales, T., Huszár, F., and Lane, N. D. Fedl2p: Federated learning to personalize. In *Adv. Neural Inform. Process. Syst.*, Dec. 2023.

Li, Y., Li, Q., Wang, H., Li, R., Zhong, W., and Zhang, G. Towards efficient replay in federated incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12820–12829, Jun. 2024.

Li, Y., Wang, Y., Xiao, T., Wang, H., Qi, Y., and Li, R. FedSSI: Rehearsal-free continual federated learning with synergistic synaptic intelligence. In *Int. Conf. Mach. Learn.*, Jul. 2025a.

Li, Y., Xu, W., Wang, H., Qi, Y., Guo, J., and Li, R. Personalized federated domain-incremental learning based on adaptive knowledge matching. In *Eur. Conf. Comput. Vis.*, Apr. 2025b.

Lim, W. Y. B., Luong, N. C., Hoang, D. T., Jiao, Y., Liang, Y.-C., Yang, Q., Niyato, D., and Miao, C. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Comm. Surveys & Tutorials*, Jun. 2020.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10012–10022, 2021.

Lopez-Paz, D. and Ranzato, M. A. Gradient episodic memory for continual learning. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Adv. Neural Inform. Process. Syst.*, Dec. 2017.

Luo, K., Li, X., Lan, Y., and Gao, M. Gradma: A gradient-memory-based accelerated federated learning with alleviated catastrophic forgetting. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 3708–3717, Jun. 2023.

Luu, M. N., Nguyen, M.-D., Bedeer, E., Nguyen, V. D., Hoang, D. T., Nguyen, D. N., and Pham, Q.-V. Sample-driven federated learning for energy-efficient and real-time IoT sensing, 2023. URL https://arxiv.org/abs/2310.07497.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and Arcas, B. A. y. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Int. Conf. on AISTATS*, Apr. 2017.

Mirzadeh, S.-I., Farajtabar, M., Görür, D., Pascanu, R., and Ghasemzadeh, H. Linear mode connectivity in multitask and continual learning. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2021.

Nguyen, D. C., Pham, Q.-V., Pathirana, P. N., Ding, M., Seneviratne, A., Lin, Z., Dobre, O., and Hwang, W.-J. Federated learning for smart healthcare: A survey. *ACM Comput. Surv.*, 55, Mar. 2023a.

Nguyen, M.-D., Pham, Q.-V., Hoang, D. T., Tran-Thanh, L., Nguyen, D. N., and Hwang, W.-J. Label driven knowledge distillation for federated learning with non-IID data, 2022. URL https://arxiv.org/abs/2209.14520.

Nguyen, M.-D., Lee, S.-M., Pham, Q.-V., Hoang, D. T., Nguyen, D. N., and Hwang, W.-J. HCFL: A high compression approach for communication-efficient federated learning in very large scale IoT networks. *IEEE Trans. on Mob. Comp.*, Jul. 2023b.

Nguyen, T.-B., Nguyen, M.-D., Park, J., Pham, Q.-V., and Hwang, W. J. Federated domain generalization with data-free on-server gradient matching. In *Int. Conf. Learn. Represent.*, May 2025.

Qi, D., Zhao, H., and Li, S. Better generative replay for continual federated learning. In *Int. Conf. Learn. Represent.*, May 2023.

Saha, G., Garg, I., and Roy, K. Gradient projection memory for continual learning. In *Int. Conf. Learn. Represent.*, May 2021.

Tran, K.-T., Dao, D., Nguyen, M.-D., Pham, Q.-V., O'Sullivan, B., and Nguyen, H. D. Multi-agent collaboration mechanisms: A survey of LLMs, 2025. URL https://arxiv.org/abs/2501.06322.

Tran, M.-T., Le, T., Le, X.-M., Harandi, M., and Phung, D. Text-enhanced data-free approach for federated class-incremental learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2024.

Tung, N. X., Giang, L. T., Son, B. D., Jeong, S. G., Chien, T. V., Hwang, W. J., and Hanzo, L. Graph neural networks for next-generation-IoT: Recent advances and open challenges, 2025. URL https://arxiv.org/abs/2412.20634.

Vu, T.-H., Kumar Jagatheesaperumal, S., Nguyen, M.-D., Van Huynh, N., Kim, S., and Pham, Q.-V. Applications of generative ai (gai) for mobile and wireless networking: A survey. *IEEE Internet of Things Journal*, 2025.

Wang, Y., Guo, S., Pan, Y., Su, Z., Chen, F., Luan, T. H., Li, P., Kang, J., and Niyato, D. Internet of agents: Fundamentals, applications, and challenges, 2025. URL https://arxiv.org/abs/2505.07176.

Wu, Y., Huang, L.-K., Wang, R., Meng, D., and Wei, Y. Meta continual learning revisited: Implicitly enhancing online hessian approximation via variance reduction. In *Int. Conf. Learn. Represent.*, Dec. 2024.

Wuerkaixi, A., Cui, S., Zhang, J., Yan, K., Han, B., Niu, G., Fang, L., Zhang, C., and Sugiyama, M. Accurate forgetting for heterogeneous federated continual learning. In *Int. Conf. Learn. Represent.*, May 2024.

Yang, E., Shen, L., Wang, Z., Liu, S., Guo, G., and Wang, X. Data augmented flatness-aware gradient projection for continual learning. In *Int. Conf. Comput. Vis.*, pp. 5630–5639, Oct. 2023.

Yang, X., Huang, W., and Ye, M. Fedas: Bridging inconsistency in personalized federated learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2024.

Yoon, J., Jeong, W., Lee, G., Yang, E., and Hwang, S. J. Federated continual learning with weighted inter-client transfer. In *Int. Conf. Mach. Learn.*, Jul. 2021.

Zhang, J., Chen, C., Zhuang, W., and Lyu, L. TARGET: Federated Class-Continual Learning via Exemplar-Free Distillation . In *Int. Conf. Comput. Vis.*, Oct. 2023a.

Zhang, J., Hua, Y., Cao, J., Wang, H., Song, T., Xue, Z., Ma, R., and Guan, H. Eliminating domain bias for federated learning in representation space. In *Adv. Neural Inform. Process. Syst.*, Dec. 2023b.

Zhang, J., Hua, Y., Wang, H., Song, T., Xue, Z., Ma, R., and Guan, H. Fedala: Adaptive local aggregation for personalized federated learning. In *AAAI*, volume 37, Jun. 2023c.

Zhang, J., Liu, Y., Hua, Y., Wang, H., Song, T., Xue, Z., Ma, R., and Cao, J. PFLlib: A beginner-friendly and comprehensive personalized federated learning library and benchmark. *Journal of Machine Learning Research*, Feb. 2025.

A. Related Works

A.1. Importance-based Sampling

LGA (Dong et al., 2024) proposes balancing class contributions to the gradient to reduce forgetting caused by class imbalance in incremental tasks. Re-Fed (Li et al., 2024) defines an importance score to selectively store key samples in replay memory. SCFL (Luu et al., 2023) removes samples that has the high affinity score with the previous samples. FedWeIT (Yoon et al., 2021) splits model weights into global and task-specific parts, letting clients combine task-specific knowledge from others in a weighted manner. FedSSI (Li et al., 2025a) adds a regularization term that tracks how important each weight is during training. It prevents important weights from changing too much, helping to preserve past knowledge.

A.2. Gradient Memory

GradMA (Luo et al., 2023) adjusts gradients locally by projecting them using stored gradients from other clients, optimized through quadrature methods.

A.3. Generative Replay Memory

Generative models contributes significantly to the design of generative replay memory, by introducing models that generate pseudo data (Vu et al., 2025). FedCIL (Qi et al., 2023) offers an efficient way to train GAN-based replay memory in distributed settings. TARGET (Zhang et al., 2023a) trains a server-side generative model to create data aligned with the global model, then uses it to update local models through knowledge distillation. AF-FCL (Wuerkaixi et al., 2024) applies a normalizing flow to model the core data distribution while removing biased features. pFedDIL (Li et al., 2025b) uses a small auxiliary classifier in each personalized model to distinguish its own task from others, aiding knowledge transfer across tasks. FBL (Dong et al., 2023) generates trustworthy pseudo labels using adaptive class balancing, semantic correction, and relation consistency, which helps correct background shifts and stabilize gradient updates.

A.4. Episodic Replay Memory for Continual Learning

GEM (Lopez-Paz & Ranzato, 2017) introduces episodic memory to store selected past samples and compute task-specific gradients, enabling gradient projection to reduce forgetting. VR-MCL (Wu et al., 2024) presents a meta-learning approach that leverages stored data in memory for continual learning.

(Qi et al., 2023) shows that client-specific feature shifts can harm GAN-based replay in federated learning. FedCIL addresses this with a distillation strategy to reduce domain discrepancies. GPM (Saha et al., 2021) stores gradient directions in memory instead of raw data, supporting continual learning. FS-DGPM (Deng et al., 2021) improves GPM by flattening gradient projections, leading to better generalization and resistance to noisy loss landscapes.

B. Experimental Details

Our experiments are built upon the **pFLLib framework** (Zhang et al., 2025) and conducted using a system equipped with four NVIDIA GeForce RTX 4090 GPUs and two NVIDIA GeForce RTX 3090 GPUs. Detailed configurations are described below.

B.1. Datasets

We investigate heterogeneous Federated Continual Learning (FCL) settings to simulate realistic, challenging non-IID scenarios. Following (Dohare et al., 2024), we create sequential classification tasks by grouping classes. For instance, one task might involve differentiating chickens from llamas, while another might focus on distinguishing phones from computers.

To assess performance under varying levels of heterogeneity, we test two distinct task configurations. In the first, each task comprises 20 distinct classes, representing a conventional setup commonly found in existing literature (Wuerkaixi et al., 2024). In the second, each task contains only 2 classes, creating a more challenging environment where models are more prone to overfitting and catastrophic forgetting, leading to increased client divergence.

We utilize two widely recognized benchmark datasets:

Non-Overlapped-CIFAR100. This dataset (Krizhevsky, 2009) contains 60,000 32×32 images across 100 object categories. For the 2-class per task setup, 4950 unique tasks can be formed. For 20 classes per task, over 5×10^{20} tasks are possible.

Non-Overlapped-ImageNet1K. This dataset (Deng et al., 2009) features over 1.3 million high-resolution images (224×224 after resizing) from 1,000 categories. With 2 classes per task, we can form half a million tasks. For 20 classes per task, over 3×10^{41} tasks can be created. ImageNet1K's large scale and diversity present significant challenges in terms of memory, computation, and model scalability.

B.2. Baselines

We compare our approach against several established methods from both traditional Federated Learning (FL) and Federated Continual Learning (FCL). For **conventional FL**, we include FedAvg (McMahan et al., 2017) as the foundational baseline. We also incorporate personalized FL methods like FedALA (Zhang et al., 2023c), FedL2P (Lee et al., 2023), and FedAS (Yang et al., 2024), which help models adapt to client-specific data. Furthermore, we include methods focused on constructing more robust global models by reducing inter-client bias, such as FedDBE (Zhang et al., 2023b) and FedOMG (Nguyen et al., 2025).

For **FCL**, we assess several state-of-the-art methods. FedWeIT (Yoon et al., 2021) exemplifies approaches that use specialized modules for task-specific adaptation. GLFC (Dong et al., 2022) employs a distillation-based method to combat forgetting. FedCIL (Qi et al., 2023), LANDER (Tran et al., 2024), TARGET (Zhang et al., 2023a), and AF-FCL (Wuerkaixi et al., 2024) adopt generative replay strategies, training generative models on each client to synthesize pseudo-data for previously encountered tasks. Among these, AF-FCL is particularly relevant as it directly addresses heterogeneous FCL. Lastly, FedSSI (Li et al., 2025a) is included for its approach of estimating the importance of weight changes to preserve prior knowledge.

B.3. Evaluation Metrics

We use two standard metrics from the Continual Learning (CL) literature (Yoon et al., 2021; Mirzadeh et al., 2021) to evaluate global model performance in FCL: accuracy and averaged forgetting.

Averaged Forgetting (AF). This metric quantifies how much a model forgets previously learned tasks. It measures the decline from a task's highest accuracy (typically right after training) to its final accuracy after all tasks have been learned. For T tasks, it's calculated as:

$$AF = \frac{1}{T-1} \sum_{i=1}^{T-1} \max_{t \in 1,\ldots,T-1} (a_{t,i} - a_{T,i}). \tag{6}$$

Minimizing AF is crucial for maintaining overall performance as the model shifts focus to new tasks.

B.4. Architecture Details

For CIFAR-10, CIFAR100, Digit10, and Office31, we use **ResNet-18** (He et al., 2016) as the backbone network architecture for all validation experiments. For ImageNet1K, we employ **Swin Transformer Tiny (Swin-T)** (Liu et al., 2021) as the backbone. It's important to note that FCIL, LANDER, TARGET, FedL2P, FedWeIT, and AF-FCL use additional generative networks or modify their network architectures, with details summarized in Table 4.

Table 4: Architectural details of methods with modified models.

Method	**CIFAR-10, CIFAR100, Digit10, Office31**		**ImageNet1K**	
	Model	**#Params**	**Model**	**#Params**
FedAvg	ResNet-18	11.7 M	Swin-T	28.8 M
FedSSI	ResNet-18	11.7 M	Swin-T	28.8 M
FCIL	ResNet-18 + GAN	16.1 M	Swin-T + GAN	49.7 M
LANDER	ResNet-18 + GAN	16.1 M	Swin-T + GAN	49.7 M
TARGET	ResNet-18 + GAN	16.1 M	Swin-T + GAN	49.7 M
FedL2P	ResNet-18 + Meta-Net	13.5 M	Swin-T + Meta-Net	32.6 M
FedWeIT (T)	Modified ResNet-18	596.2 M	Modified Swin-T	7192.3 M
FedWeIT (C)	Modified LeNet	171.8 B		
AF-FCL	ResNet-18 + NFlow	21.3 M	Swin-T + NFlow	53.4 M

Specifically, FedWeIT augments the base model with sparse task-adaptive parameters, task-specific masks over local base parameters, and attention weights for inter-client knowledge transfer. FCIL, LANDER, and TARGET incorporate additional GANs to learn past task features. FedL2P introduces a meta-network that generates personalized hyperparameters, such as batch normalization statistics and learning rates, adapted to each client's local data distribution to improve learning on non-IID data. AF-FCL additionally requires a normalizing flow generative model (NFlow[1]) for credibility estimation and a generative replay mechanism, which guide selective retention and forgetting.

Table 5: Experimental Details. Settings for heterogeneous and non-IID distributed FCL.

Attributes	**Heterogeneous FCL**		**Non-IID distributed FCL**			
	CIFAR100	**ImageNet1K**	**CIFAR10**	**CIFAR100**	**Digit10**	**Office31**
Task size	141 MB / 14 MB	8 GB / 0.8 GB	141 MB	141 MB	480 MB	88 MB
Image number	60K	1.3M	60K	60K	110K	4.6K
Image Size	$3 \times 32 \times 32$	$3 \times 224 \times 224$	$3 \times 32 \times 32$	$3 \times 32 \times 32$	$1 \times 28 \times 28$	$3 \times 300 \times 300$
Task number	5/50	50/500	5	10	4	3
Batch Size	128	128	64	64	64	32
ACC metrics	Top-1	Top-1	Top-1	Top-1	Top-1	Top-1
Learning Rate	0.005	0.005	0.01	0.01	0.001	0.01
Data heterogeneity	N/A	N/A	0.1	10.0	0.1	1.0
Client numbers	10	10	10	10	10	10
Local training epoch	5	5	5	5	5	5
Client selection ratio	1.0	1.0	1.0	1.0	1.0	1.0
Communication Round	25	25	80	100	60	60

B.5. Training Details

In our proposed heterogeneous federated continual learning framework for the CIFAR100 and ImageNet1K datasets, we consider a setting involving 10 clients with a client participation fraction of 1.0. We don't adopt a conventional non-IID

[1]NFlow refers to the normalizing flow model, where the example is provided in https://github.com/zaocan666/AF-FCL/blob/main/FLAlgorithms/PreciseFCLNet/model.py

distribution in this scenario; instead, each client is assigned distinct classes, which introduces a level of heterogeneity that's more challenging than typical non-IID configurations.

Additionally, we evaluate the proposed approach under non-IID conditions using four benchmark datasets: CIFAR-10, CIFAR100, Digit-10, and Office-31. For these experiments, we simulate data heterogeneity using the Dirichlet distribution with varying concentration parameters (e.g., $\alpha = 0.1, 1.0, 10.0,$ and 100.0) to control the degree of non-IID-ness. Table 5 provides the complete experimental details.