Heterogeneity-aware Distillation for Federated Continual Learning

Gaozhuo Liu[1], **Xiuying Wang**[2], **Yulong Li**[3], **Chuang Zhao**[4], **Yankai Jiang**[5], **Imran Razzak**[3], **Yichen Li**[6]*

[1]School of Information and Communication Engineering, Beijing University of Posts and Telecommunications
[2]International School, Beijing University of Posts and Telecommunications
[3]Mohamed bin Zayed University of Artificial Intelligence
[4]Tianjin University [5]Shanghai Artificial Intelligence Laboratory
[6]School of Computer Science and Technology, Huazhong University of Science and Technology

3255946590@bupt.edu.cn, xiuying.wang@se23.qmul.ac.uk, ycli0204@hust.edu.cn

Abstract

Federated Continual Learning (FCL) requires distributed clients to acquire new tasks while retaining previously learned knowledge. In this paper, we revisit the heterogeneity challenge in FCL, which is composed of two distinct aspects: spatial static heterogeneity, e.g., initial data distribution and model architecture differences, and temporal dynamic heterogeneity, e.g., task shifts over time. While existing methods acknowledge the presence of heterogeneity, they typically treat it in a unified manner by employing a single adaptive strategy without explicitly disentangling their impacts. In contrast, we propose a heterogeneity-aware distillation framework for FCL called Ha-FCL, which is designed to handle spatial and temporal heterogeneity through tailored distillation strategies. Specifically, Ha-FCL combines width-aware federated training with server-side public-anchor distillation, integrating multi-layer feature distillation for architecture-aware alignment with historical-logit regularization and representation co-distillation. This forms a coordinated dual-dimensional distillation scheme that balances adaptation to new tasks with retention of prior knowledge. This enables more targeted knowledge transfer and leads to improved model performance under complex FCL scenarios. We evaluate our method in various heterogeneous settings. Compared with state-of-the-art baselines, Ha-FCL consistently improves test accuracy and shows better robustness under challenging heterogeneous FCL settings.

1 Introduction

Federated learning (FL) enables distributed devices to collaboratively train a global model by exchanging model updates rather than raw data, thereby preserving local data privacy [McMahan et al., 2017]. This paradigm has been widely explored in applications such as personalized recommendation, medical diagnosis [Rieke et al., 2020], and autonomous vehicles [Chellapandi et al., 2023]. However, conventional FL typically assumes stationary tasks, whereas real-world systems often face evolving data distributions and sequentially arriving tasks. Training on such streams can cause catastrophic forgetting [Kirkpatrick et al., 2017], where adaptation to new tasks erodes previously acquired knowledge and limits FL deployment in dynamic environments.

*Corresponding author.

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

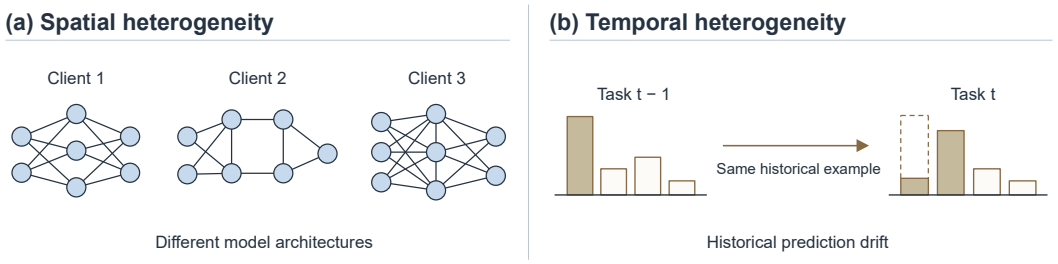


Figure 1: Conceptual illustration of spatial static and temporal dynamic heterogeneity in FCL. (a) Architectural differences across clients. (b) Prediction drift for the same historical example as tasks progress. Ha-FCL explicitly disentangles these factors through feature alignment for spatial consistency and historical regularization for temporal retention.

To address this challenge, Federated Continual Learning (FCL) has emerged as a promising paradigm, enabling clients to continuously learn from new tasks while preserving data privacy [Li et al., 2024a, 2025]. Existing FCL methods mainly mitigate forgetting through replay, regularization, or adaptive coordination. For example, GLFC [Dong et al., 2022] maintains client-side memory buffers, Re-Fed [Li et al., 2024a] coordinates global and local sample caching, and CAN [Rong et al., 2025] guides synthetic data generation and adaptive replay through client-side navigation. Regularization-based methods such as FedSSI [Li et al., 2024b] further adapt synaptic-intelligence-style constraints with personalized surrogate models to balance local and global information.

Despite their effectiveness, these FCL methods predominantly rely on heuristic designs or task-specific tuning, often overlooking the fundamental sources of heterogeneity that make FCL challenging. In particular, the heterogeneity in FCL arises not only from spatial static differences but also from temporal dynamics. These two dimensions of heterogeneity are conceptually distinct yet often entangle in practice, and their independent and joint effects on model performance remain underexplored. Most existing works adopt a unified treatment, applying a single mechanism to all forms of heterogeneity without explicitly tailoring the adaptation strategy to each source [Chen et al., 2025]. Figure 1 illustrates these two sources of heterogeneity.

This paper takes a step toward decomposing and systematically addressing the dual nature of heterogeneity in FCL. Such decoupling is non-trivial because spatial and temporal factors are intertwined during federated optimization, making it difficult to isolate their individual impacts on model performance. Inspired by recent advances in federated distillation [Wang et al., 2023], we design targeted knowledge distillation mechanisms that are aligned with the characteristics of each type of heterogeneity.

Spatial heterogeneity typically stems from architectural and statistical differences across clients [Jia et al., 2024, Chen et al., 2024]. These discrepancies can lead to misaligned intermediate representations, making naive aggregation or output-only distillation insufficient for transferring knowledge across heterogeneous models. To mitigate this, we introduce feature alignment, which explicitly aligns intermediate representations across clients [Qi et al., 2025]. By encouraging features to reside in a compatible latent space, feature alignment helps suppress client-specific biases and improves the robustness of knowledge fusion under spatial heterogeneity.

Temporal dynamic heterogeneity arises as clients sequentially receive new tasks over time, leading to catastrophic forgetting [Li et al., 2024a]. This form of non-stationarity requires the model to retain prior task knowledge while still adapting to incoming data. To this end, we incorporate historical-logit regularization, which preserves the model's functional behavior by aligning current predictions with cached soft targets on historical public anchors. Unlike feature alignment, which operates at the representation level, historical-logit regularization constrains output-level behavior and mitigates decision-boundary drift over time. This makes it particularly suitable for continual learning settings where raw historical private data cannot be centrally revisited.

Although these tailored strategies address distinct sources of heterogeneity, naively combining them may introduce additional computation and optimization sensitivity [Chen et al., 2024]. In particular, the relative strength of spatial alignment and temporal regularization must be carefully balanced; otherwise, one may dominate the other and impair either plasticity or stability.

To tackle the intertwined challenges of federated continual learning, we introduce an efficient, heterogeneity-aware framework called Ha-FCL that explicitly disentangles spatial static heterogeneity from temporal dynamic heterogeneity. Departing from prior methods that apply a uniform adaptation mechanism, Ha-FCL employs feature-alignment distillation to harmonize intermediate representations across structurally heterogeneous models, while historical-logit regularization safeguards previously acquired knowledge through cached soft predictions on historical anchors. The synergy of these two targeted strategies steers heterogeneous clients toward a compatible latent space without severely degrading performance on earlier tasks, a property essential for deployment in evolving environments. Furthermore, a multi-granularity representation co-distillation scheme is incorporated to coordinate feature-level and output-level supervision, thereby enhancing training stability under pronounced spatiotemporal shifts.

We evaluate Ha-FCL under a wide range of heterogeneous settings. Extensive experiments show that Ha-FCL consistently improves final accuracy compared to strong FCL baselines, with particularly notable gains under severe spatiotemporal shifts. These results demonstrate the effectiveness of heterogeneity-aware design in federated continual learning and suggest promising directions for more adaptive and modular FCL algorithms. To summarize, our contributions are threefold:

- We are the first to explicitly factor heterogeneity in federated continual learning into spatial static and temporal dynamic components, mitigating each with a dedicated adaptation method.
- We propose Ha-FCL, which unifies feature-alignment distillation and historical-logit regularization under a multi-granularity co-distillation scheme, harmonizing cross-model representations while suppressing catastrophic forgetting.
- Extensive experiments across various settings show that Ha-FCL delivers superior accuracy and stronger robustness than state-of-the-art FCL baselines under challenging heterogeneous federated continual learning scenarios.

2 Related Work

2.1 Federated Continual Learning

Federated Continual Learning (FCL) addresses catastrophic forgetting when distributed clients learn a sequence of tasks [Li et al., 2024a]. Existing methods are commonly categorized into replay-based, parameter-isolation, and regularization-based approaches. Replay-based methods such as GLFC and Re-Fed store or synthesize historical samples to rehearse prior knowledge [Dong et al., 2022, Li et al., 2024a], but may incur memory and privacy concerns. FedCIL and AF-FCL employ generative replay to retain historical knowledge [Qi et al., 2023, Wuerkaixi et al., 2024]. Parameter-isolation methods such as FedWeIT allocate task-specific parameters to reduce interference, yet their model size grows with tasks [Yoon et al., 2021]. Regularization-based approaches constrain updates or distill logits to balance plasticity and stability. Recent prompt-, adapter-, and LoRA-based continual learning methods provide rehearsal-free alternatives [Wang et al., 2022b,a, Liang and Li, 2024, Yu et al., 2024], but often depend on large pretrained backbones or prompt-adapted architectures. For asynchronous task arrivals, C^2-AFCL combines intra-client calibration with orthogonal inter-task interference correction to mitigate task drift [Li et al., 2026b]. Most existing FCL methods primarily focus on temporal forgetting and implicitly assume comparable client architectures, limiting their applicability to heterogeneous edge environments.

2.2 Heterogeneous Federated Learning

Federated heterogeneity includes statistical non-IID distributions and system-level differences such as architectural heterogeneity [Kairouz and McMahan, 2021]. Recent work also addresses heterogeneous label correlations through consensus-guided correction in FedHarmony [Kou et al., 2026b], and annotation-quality disparity through quality-adaptive calibration and reliability-aware aggregation in FedQual [Wu et al., 2026]. Width-scalable methods such as HeteroFL and FjORD allow clients to train variable-width sub-networks [Diao et al., 2020, Horvath et al., 2021], while distillation-based methods such as FedMD and FedDF transfer knowledge through public-data logits [Li and Wang, 2019, Lin et al., 2020]. However, output-only distillation may not sufficiently bridge feature-space discrepancies between structurally diverse models. FedFD aligns architecture-specific feature

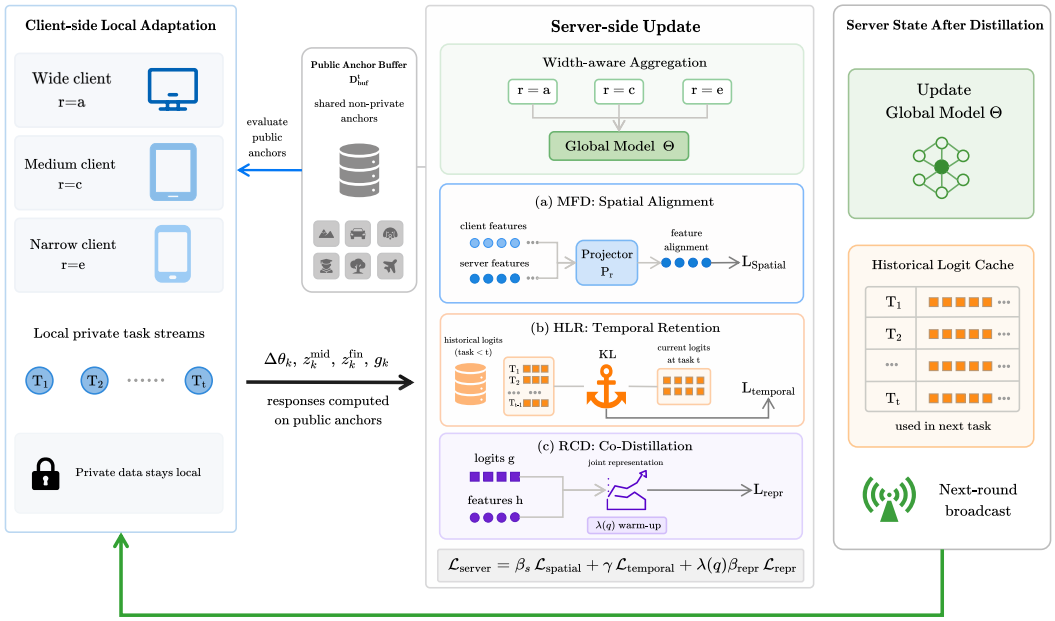


Figure 2: Overview of the Ha-FCL framework. In each communication round, heterogeneous clients first optimize their width-specific sub-networks on private task streams. The server then performs width-aware aggregation and collects feature/logit responses on a public distillation buffer. Based on these public-anchor responses, Ha-FCL constructs width-specific feature centroids for Multi-Layer Feature Distillation (MFD), preserves historical predictive behavior through Historical-Logit Regularization (HLR), and couples representation-level and output-level supervision via Representation Co-Distillation (RCD). All distillation modules are conducted on public anchors at the server side, enabling knowledge consolidation without centralizing private client data.

representations through orthogonal projections [Li et al., 2026c], while FedCD combines feature- and logit-based distillation through cross-layer attention and Gaussian mixture modeling [Li et al., 2026a]. Beyond federated settings, positive-unlabeled reinforcement learning distillation uses teacher-generated responses to guide local preference optimization of on-premise small models [Kou et al., 2026a]. The federated approaches discussed above primarily address cross-client heterogeneity under static task settings. Ha-FCL jointly addresses architectural heterogeneity and temporal forgetting through representation-level alignment and historical-logit regularization.

3 Methodology

In this section, we first formalize the problem of Federated Continual Learning (FCL) under heterogeneous settings. We then introduce **Ha-FCL**, a heterogeneity-aware framework designed to disentangle and address the dual challenges of spatial static heterogeneity (statistical and architectural differences) and temporal dynamic heterogeneity (catastrophic forgetting). Figure 2 illustrates the overall pipeline of Ha-FCL, including width-aware local training, public-anchor response aggregation, and the three server-side distillation modules.

3.1 Problem Formulation

We consider an FCL system with a central server and K clients, indexed by $k \in \{1, \ldots, K\}$, over sequential time steps $t = 1, \ldots, T$. At time t, client k receives a local task:

$$\mathcal{T}_k^t = \left\{ (x_k^{(i,t)}, y_k^{(i,t)}) \right\}_{i=1}^{N_k^t}, \tag{1}$$

where $x_k^{(i,t)} \in \mathcal{X}$ and $y_k^{(i,t)} \in \mathcal{Y}^t$ denote the input and label. The evolving label space $\mathcal{Y}^t$ and data distribution induce temporal heterogeneity. In the adopted width-adaptive training setup, the server

maintains a full-width global model f_Θ with parameters $\Theta \in \mathbb{R}^m$, while each client trains a width-specific sub-network $\theta_k = \mathcal{S}_{r_k}(\Theta)$ with assigned width $r_k \in \mathcal{R}$, where $\mathcal{R}$ denotes permissible widths and $\mathcal{S}_{r_k}(\cdot)$ is the extraction operator. This width-adaptive design captures spatial static heterogeneity across resource-constrained clients.

Each client k learns from its cumulative stream $\mathcal{D}_k^{\le t} = \bigcup_{n=1}^{t} \mathcal{T}_k^n$, with the ideal objective of minimizing empirical risk over $\mathcal{D}^{\le t} = \bigcup_{k=1}^{K} \mathcal{D}_k^{\le t}$. Since historical private data are distributed and cannot be centrally revisited, this objective is only an idealized continual-learning target. Ha-FCL additionally maintains a compact public distillation buffer $\mathcal{D}_{\text{buf}}^t$ of anchors associated with observed tasks/classes, disjoint from private data and used only for feature/logit response collection and distillation supervision. The resulting challenge is to jointly handle width-induced architectural mismatch and temporal forgetting, where learning $\mathcal{T}^t$ may degrade performance on $\mathcal{T}^{1:t-1}$.

3.2 Ha-FCL: Heterogeneity-Aware Distillation Framework

To tackle these challenges, Ha-FCL decouples heterogeneity into spatial and temporal components. In each communication round, active clients train width-specific sub-networks on private task streams, while the server performs coordinate-wise width-aware aggregation over compatible sub-network parameters (Appendix B.1) and constructs public-anchor distillation signals from client responses. Ha-FCL consists of three modules: Multi-Layer Feature Distillation (MFD) for architecture-aware representation alignment, Historical-Logit Regularization (HLR) for temporal knowledge retention, and Representation Co-Distillation (RCD) for coupling feature- and output-level supervision. To address statistical heterogeneity, public-anchor consensus guides server-side refinement, providing a common reference for subsequent client updates.

3.2.1 Multi-Layer Feature Distillation

A key challenge in heterogeneous federated learning is that cross-width clients produce feature maps with mismatched channel dimensions and semantic spaces, making naive parameter aggregation or output-only distillation insufficient. To address this spatial static heterogeneity, Multi-Layer Feature Distillation (MFD) constructs width-specific teacher centroids on public anchors and aligns server-side student features through learnable projectors. For a client k with width r_k and a public anchor $x \in \mathcal{D}_{\text{buf}}^t$, we denote its intermediate and final feature responses as $z_k^{(\text{mid})}(x)$ and $z_k^{(\text{fin})}(x)$, respectively. At communication round q, let $\mathcal{S}_q$ be the set of active clients, $\mathcal{S}_q^r = \{k \in \mathcal{S}_q : r_k = r\}$ be the active clients in width group r, and $\mathcal{R}_q = \{r \in \mathcal{R} : |\mathcal{S}_q^r| > 0\}$ be the set of active width groups. For each layer $\ell \in \{\text{mid}, \text{fin}\}$ and width group $r \in \mathcal{R}_q$, the server computes a teacher centroid by sample-weighted aggregation:

$$\bar{z}^{(\ell)}(r; x) = \sum_{k \in \mathcal{S}_q^r} \omega_k \, z_k^{(\ell)}(x), \quad \omega_k = \frac{n_k}{\sum_{j \in \mathcal{S}_q^r} n_j}, \tag{2}$$

where n_k is the local sample count. The resulting sample-weighted centroid $\bar{z}^{(\ell)}(r; x)$ serves as an aggregated semantic reference for width group r. To align server-side student features with these group-wise teacher centroids, we use server-side learnable projectors that map the global model's feature responses to width-specific dimensions. Let $\tilde{z}^{(\ell)}(r; x)$ denote the projected student feature for width group r at layer ℓ. The spatial alignment loss is defined as:

$$\mathcal{L}_{\text{spatial}} = \frac{1}{|\mathcal{D}_{\text{buf}}^t|} \sum_{x \in \mathcal{D}_{\text{buf}}^t} \sum_{\ell \in \{\text{mid},\text{fin}\}} \alpha_\ell \sum_{r \in \mathcal{R}_q} \left\| \tilde{z}^{(\ell)}(r; x) - \bar{z}^{(\ell)}(r; x) \right\|_2^2, \tag{3}$$

where α_ℓ balances the contributions of intermediate and final layers. Multi-layer alignment exploits the hierarchical nature of deep representations: intermediate features capture structural patterns, while final features encode task-level semantics. Together with learnable projectors, it provides coarse-to-fine guidance for reducing cross-width semantic gaps during server-side distillation.

3.2.2 Historical-Logit Regularization

As clients learn sequential task streams, the decision boundaries associated with previously learned classes may gradually drift, leading to catastrophic forgetting. Since privacy constraints prevent the

server from centrally storing or replaying historical private data, we use the public distillation buffer $\mathcal{D}_{\text{buf}}^t$ as a shared reference set and preserve prior knowledge through output-level distillation.

Specifically, Ha-FCL caches historical logits on public anchors and uses them to regularize the current model during later tasks. Let $\mathcal{D}_{\text{buf}}^{<t} \subseteq \mathcal{D}_{\text{buf}}^t$ denote the public anchors associated with previously observed tasks. For each anchor $x \in \mathcal{D}_{\text{buf}}^{<t}$ and width group $r \in \mathcal{R}$, we denote by $\tilde{y}_r^{t-1}(x)$ the cached logit target obtained from the preceding task stage. During the learning of the current task, these historical logits are kept fixed and serve as output-level anchors. Compared with directly replaying private samples or storing intermediate features that may shift as the encoder evolves, logits characterize the model's functional behavior on prior tasks and provide a stable supervision signal.

To preserve the soft class relations encoded in historical predictions, we apply temperature scaling and minimize the Kullback–Leibler (KL) divergence between the historical soft target and the current prediction:

$$\mathcal{L}_{\text{KL}}^{(r)}(x) = \tau^2 \,\text{KL}\left(\sigma_\tau\big(\tilde{y}_r^{t-1}(x)\big) \,\big\|\, \sigma_\tau\big(f_{\mathcal{S}_r(\Theta)}(x)\big)\right), \tag{4}$$

where $f_{\mathcal{S}_r(\Theta)}$ denotes the current global model instantiated with width r, and $\sigma_\tau(\cdot)$ is the softmax function with temperature τ. In class-incremental outputs, both the softmax and KL divergence are computed over previously observed classes, while τ softens inter-class relations and τ^2 rescales gradients.

The resulting historical-logit regularization term is formulated as:

$$\mathcal{L}_{\text{temporal}} = \frac{1}{|\mathcal{R}_q|} \sum_{r \in \mathcal{R}_q} \frac{1}{|\mathcal{D}_{\text{buf}}^{<t}|} \sum_{x \in \mathcal{D}_{\text{buf}}^{<t}} \mathcal{L}_{\text{KL}}^{(r)}(x). \tag{5}$$

When $t = 1$, no historical anchors are available and $\mathcal{L}_{\text{temporal}}$ is set to zero. At each round, HLR is evaluated only for active width groups; the historical targets of inactive groups remain cached for later rounds. By anchoring predictions on historical public anchors, this term mitigates temporal drift while allowing the model to adapt to new tasks without revisiting private historical data.

3.2.3 Representation Co-Distillation

Although MFD and HLR address spatial and temporal heterogeneity respectively, optimizing them in isolation may lead to inconsistent feature-level and output-level supervision. To bridge this gap, we introduce **Representation Co-Distillation (RCD)**, which couples features and logits in a unified embedding space on public anchors.

For width group r and anchor $x \in \mathcal{D}_{\text{buf}}^t$, let $h_s^{(r)}(x)$ and $g_s^{(r)}(x)$ be the feature and logit responses of the server global model f_Θ, and let $h_t^{(r)}(x)$ denote the normalized feature centroid derived from the MFD final-layer responses. The current-round logit teacher is the sample-weighted group consensus:

$$\bar{g}^{(r)}(x) = \sum_{k \in \mathcal{S}_q^r} \omega_k \, g_k(x), \quad \omega_k = \frac{n_k}{\sum_{j \in \mathcal{S}_q^r} n_j}, \tag{6}$$

where $g_k(x)$ is the client logit response. Unlike the cached HLR targets, $\bar{g}^{(r)}(x)$ captures current-round predictive consensus.

We employ group-specific projectors $Q^{(r)}(\cdot)$ and $R^{(r)}(\cdot)$ to map these modalities into a compatible space. The teacher feature centroid serves as the reference of the joint embedding space, while the logit projector maps the student and aggregated teacher logits into the same coordinate system. The RCD loss minimizes the element-wise discrepancy between the projected student embedding and the teacher embedding:

$$\mathcal{L}_{\text{repr}} = \frac{1}{|\mathcal{D}_{\text{buf}}^t| \, |\mathcal{R}_q|} \sum_{x \in \mathcal{D}_{\text{buf}}^t} \sum_{r \in \mathcal{R}_q} \frac{1}{d_r} \sum_{j=1}^{d_r} \phi\left(\Delta_j^{(r)}(x)\right), \tag{7}$$

where d_r is the embedding dimension for width group r, $\mathcal{R}_q = \{r \in \mathcal{R} : |\mathcal{S}_q^r| > 0\}$ denotes the set of active width groups at round q, and

$$\Delta_j^{(r)}(x) = \left[Q^{(r)}(h_s^{(r)}(x)) + R^{(r)}(g_s^{(r)}(x))\right]_j - \left[h_t^{(r)}(x) + R^{(r)}(\bar{g}^{(r)}(x))\right]_j. \tag{8}$$

We instantiate $\phi(\cdot)$ as the Smooth L1 loss. Compared with MSE, Smooth L1 is less sensitive to large early-stage discrepancies while still providing stable gradients near convergence, making it suitable for cross-architecture representation distillation.

To avoid over-constraining the model at the beginning of each task, we use a coarse-to-fine schedule $\lambda(q)$. Specifically, we define $\lambda(q)$ as a sigmoid-shaped warm-up function that evolves with the communication round q within the current task:

$$\lambda(q) = \frac{1}{1 + \exp\left[-\eta\left(\frac{q}{Q_t} - \frac{1}{2}\right)\right]}, \tag{9}$$

where η is a dimensionless warm-up schedule parameter controlling the steepness of the transition, and Q_t denotes the total communication rounds for task t. This dynamic scheduling keeps the model sufficiently plastic during early learning and gradually strengthens representation alignment as training stabilizes.

3.2.4 Overall Optimization Objective

To synergize the three complementary modules described above, Ha-FCL integrates them into a unified training framework following width-aware local federated learning. At task t and communication round q, active clients update $\theta_k = \mathcal{S}_{r_k}(\Theta)$ on current private task data by minimizing the standard cross-entropy loss $\mathcal{L}^k_{\text{local}}$. After width-aware aggregation, the server further refines the global model through public-anchor distillation. The resulting server-side optimization objective is formulated as:

$$\mathcal{L}_{\text{server}} = \beta_s \, \mathcal{L}_{\text{spatial}} + \gamma \, \mathcal{L}_{\text{temporal}} + \lambda(q) \, \beta_{\text{repr}} \, \mathcal{L}_{\text{repr}}, \tag{10}$$

where β_s, γ, and β_{repr} are hyperparameters governing the importance of spatial alignment, historical-logit regularization, and representation co-distillation, respectively. The scheduling term $\lambda(q)$ dynamically controls the strength of representation co-distillation along communication rounds within the current task.

The coefficient β_s enforces feature-manifold consistency, γ anchors historical predictive behavior, and $\lambda(q)\beta_{\text{repr}}$ gradually strengthens representation alignment. Together, these terms help Ha-FCL balance new-task adaptation and prior-knowledge retention in heterogeneous FCL.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate on CIFAR-10, CIFAR-100 [Krizhevsky et al., 2009], and Tiny-ImageNet [Le et al., 2015] under the standard class-incremental protocol, where the model is evaluated on all classes observed so far after each task. CIFAR-10, CIFAR-100, and Tiny-ImageNet are split into 5, 10, and 20 tasks with 2, 10, and 10 classes per task, respectively. To simulate non-IID data heterogeneity, private samples are partitioned by $\text{Dir}(\alpha)$ with $\alpha \in \{1, 5, 10\}$, corresponding to severe, moderate, and mild heterogeneity. All private training samples are distributed across clients, while the test set is centralized for evaluation.

Baselines. We compare Ha-FCL with representative heterogeneous FL and FCL methods under the same client partition, participation ratio, communication rounds, width assignments, and memory budget whenever applicable. Baselines without native width support use the same width-aware aggregation while retaining their original continual-learning mechanisms. HeteroFL [Diao et al., 2020] serves as the heterogeneous FL baseline with variable-width sub-networks. FedCIL [Qi et al., 2023], AF-FCL [Wuerkaixi et al., 2024], Re-Fed [Li et al., 2024a], MFCL [Babakniya et al., 2023], and FedWeIT [Yoon et al., 2021] cover replay-, regularization-, distillation-, and parameter-decomposition-based FCL strategies. Target denotes a final-task-only reference model trained without continual learning mechanisms.

Model Heterogeneity and Implementation. To simulate diverse edge resources, we follow the width-scalable heterogeneous setting of HeteroFL and FjORD [Diao et al., 2020, Horvath et al., 2021], using ResNet-18 backbones with different channel width multipliers. Following Sec. 3, we instantiate five shrinkage levels $\{a, b, c, d, e\}$ corresponding to $\{1, 1/2, 1/4, 1/8, 1/16\}\times$ channel widths. We evaluate two model-heterogeneity scenarios: (1) **Fully Mixed**, where clients are uniformly distributed

Table 1: Performance comparison of various methods in representative heterogeneous settings. The best results are **bold**.

Dataset	CIFAR-10				CIFAR-100				Tiny-ImageNet			
Heterogeneity	$\alpha = 1$		$\alpha = 10$		$\alpha = 1$		$\alpha = 10$		$\alpha = 1$		$\alpha = 10$	
Config	A-B-C-D-E	A-E	A-B-C-D-E	A-E	A-B-C-D-E	A-E	A-B-C-D-E	A-E	A-B-C-D-E	A-E	A-B-C-D-E	A-E
HeteroFL	30.91±0.15	42.23±0.19	37.24±0.37	44.65±0.19	12.24±0.05	13.00±0.05	14.49±0.06	16.08±0.19	5.49±0.03	6.96±0.04	7.26±0.05	7.95±0.13
FedCIL	34.84±0.17	37.41±0.21	37.30±0.24	38.08±0.15	12.40±0.17	14.87±0.03	12.66±0.09	16.28±0.07	5.44±0.06	7.52±0.07	6.79±0.04	8.25±0.03
AF-FCL	34.59±0.21	35.47±0.19	38.46±0.11	39.71±0.11	13.85±0.13	13.68±0.16	16.12±0.22	15.53±0.14	2.27±0.17	3.07±0.23	2.66±0.24	4.06±0.24
Re-Fed	33.12±0.05	36.36±0.05	42.63±0.06	48.34±0.21	14.76±0.03	17.54±0.04	15.24±0.01	19.44±0.02	4.19±0.20	4.66±0.07	6.02±0.02	7.49±0.03
Target	24.73±0.04	26.03±0.06	25.79±0.03	33.01±0.04	11.24±0.01	10.66±0.03	13.48±0.03	14.64±0.03	4.59±0.02	5.48±0.07	5.28±0.02	6.23±0.06
MFCL	22.69±0.07	30.09±0.18	25.41±0.29	32.95±0.05	12.87±0.24	15.58±0.01	13.29±0.17	17.00±0.12	2.30±0.21	3.25±0.03	3.41±0.06	4.17±0.13
FedWeIT	28.38±0.14	28.48±0.11	33.37±0.04	35.87±0.03	11.88±0.04	13.89±0.06	12.99±0.05	14.70±0.08	4.01±0.02	5.23±0.07	4.86±0.11	6.15±0.07
Ha-FCL	**47.21**±0.15	**49.31**±0.38	**48.24**±0.35	**49.43**±0.42	**20.85**±0.07	**24.12**±0.29	**22.19**±0.09	**26.25**±0.12	**8.01**±0.03	**9.36**±0.07	**9.03**±0.07	**9.60**±0.08

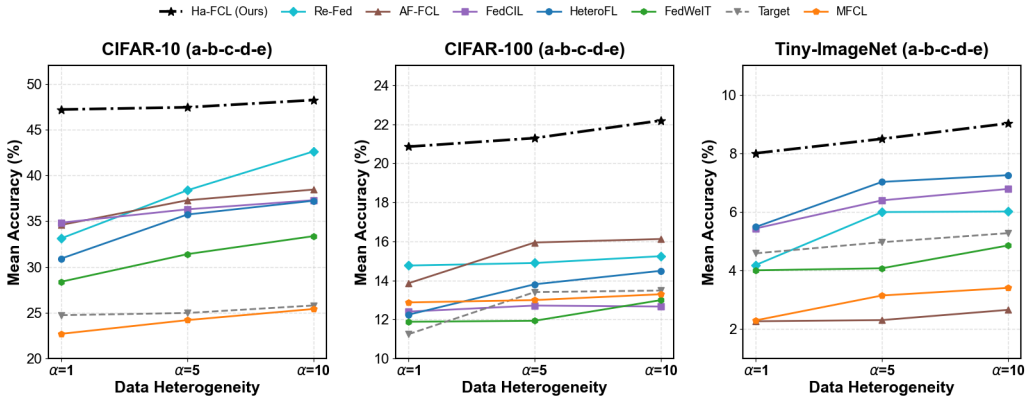


Figure 3: Performance of test accuracy under different datasets and heterogeneity settings.

across all five widths (a-e), and (2) **Extreme**, where clients are split only between the largest (a) and smallest (e) widths.

The federation consists of $K = 20$ clients with a participation ratio of 0.4, resulting in 8 active clients per round. Each incremental task is trained for 80 communication rounds. Clients perform $E = 10$ local epochs with batch size 64 and are optimized by SGD. The learning rate is tuned per dataset from $\{0.005, 0.01, 0.05\}$.

A memory budget of 500 samples is assigned to methods requiring replay or distillation [Rebuffi et al., 2017]. For Ha-FCL, this budget is instantiated as the public distillation buffer $\mathcal{D}^t_{\text{buf}}$, containing anchor samples associated with classes observed up to the current task. These public anchors are reserved before client partitioning, excluded from private local training data, and disjoint from the test set; thus, they do not increase the private training data available to clients. The buffer is used only for public-anchor response collection and server-side distillation, rather than centralized supervised training. Replay-based baselines follow their original protocols under the same 500-sample memory budget.

All experiments are conducted with three independent random seeds. We report the mean and standard deviation of Top-1 accuracy averaged over the last ten communication rounds of each run. For compactness, the main text reports representative results under selected heterogeneity levels, while complete results for all datasets, Dirichlet coefficients, and width configurations are provided in Appendix E.

4.2 Performance Overview

Test Accuracy. Table 1 reports the main Top-1 accuracy comparison between **Ha-FCL** and seven reference methods over three vision benchmarks, two representative Dirichlet coefficients ($\alpha \in$

$\{1, 10\}$), and two client-width compositions ("a-b-c-d-e" and "a-e"). Complete results covering all evaluated heterogeneity levels, including $\alpha = 5$, are provided in Appendix E. Across these settings, Ha-FCL consistently achieves the best overall performance. On CIFAR-10 under the mixed-width "a-b-c-d-e" configuration, Ha-FCL surpasses AF-FCL by **12.62pp** at $\alpha = 1$ and Re-Fed by **5.61pp** at $\alpha = 10$. The complete results further show that Ha-FCL reaches a peak CIFAR-10 accuracy of 50.98% under the "a-e" setting with $\alpha = 5$. On CIFAR-100, Ha-FCL also maintains a clear advantage, with a gain of **6.58pp** over Re-Fed under the "a-e" configuration at $\alpha = 1$. On Tiny-ImageNet, Ha-FCL consistently outperforms the compared methods, achieving **8.01%** under the severe mixed-width setting and up to **9.60%** under the polarized width configuration. These results demonstrate the effectiveness of jointly using multi-layer spatial alignment, historical-logit regularization, and representation co-distillation to handle coupled data and architectural heterogeneity in FCL.

Knowledge Retention. On CIFAR-100 under $\alpha = 1$ with mixed widths, Ha-FCL improves final accuracy over Re-Fed (20.85% vs. 14.76%), while achieving lower forgetting (0.2702 vs. 0.3219) and higher BWT (-0.2613 vs. -0.3128). Complete forgetting and BWT results are provided in Appendix F.1.

Robustness Analysis. To evaluate the applicability of Ha-FCL in realistic federated environments, we analyze its robustness along the two heterogeneity axes defined in Sec. 4.1: data heterogeneity and model heterogeneity. Regarding data heterogeneity, as the Dirichlet concentration parameter α decreases from 10 to 1, most baselines exhibit visible performance degradation. For example, FedWeIT shows a clear decline as the data distribution becomes more non-IID. In contrast, Ha-FCL maintains relatively stable performance across different α values, suggesting that public-anchor distillation and historical-logit regularization help stabilize predictive behavior under non-IID continual task streams. Regarding model heterogeneity, we compare the fully mixed "a-b-c-d-e" setting with the polarized "a-e" setting. Ha-FCL remains robust even when the capacity gap between clients is large, and in several cases achieves higher accuracy under the polarized configuration. This indicates that Multi-Layer Feature Distillation facilitates effective knowledge transfer across width groups and alleviates the negative impact of architectural disparity.

Table 2: Module ablation (a) and direct composition (b) under $\alpha = 1$. Accuracy (%).

(a) Module ablation

Method	CIFAR-10		CIFAR-100	
	a-b-c-d-e	a-e	a-b-c-d-e	a-e
Ha-FCL	**47.21**	**49.31**	**20.85**	**24.12**
-w/o MFD	41.21	43.15	20.61	23.83
-w/o HLR	39.08	42.27	18.83	19.67
-w/o RCD	41.22	43.23	19.76	22.31
HeteroFL	30.91	42.23	12.24	13.00

(b) Direct composition

Method	HeteroFL	+ KD	+ KD + HLR	Ha-FCL
Accuracy (%)	12.24	16.18	17.58	**20.85**

Ablation Study. Table 2 shows the performance of Ha-FCL after removing individual components. The variants "-w/o MFD", "-w/o HLR", and "-w/o RCD" remove Multi-Layer Feature Distillation, Historical-Logit Regularization, and Representation Co-Distillation, respectively, while HeteroFL serves as a reference baseline. All experiments are conducted under the severe non-IID setting ($\alpha = 1$) with two heterogeneous width configurations. The complete Ha-FCL consistently achieves the highest accuracy. Removing MFD weakens the alignment of mid- and high-level semantics across width-varying models, causing clear degradation on CIFAR-10 and modest but consistent drops on CIFAR-100. Removing HLR leads to the most pronounced decline in several settings, confirming the importance of preserving historical predictive behavior for mitigating catastrophic forgetting. Removing RCD also reduces performance, indicating that coupling representation-level and output-level supervision helps integrate new and prior knowledge. Although the ablated variants generally outperform HeteroFL, the full combination of MFD, HLR, and RCD yields the best overall

performance, validating their complementary roles in addressing spatial static and temporal dynamic heterogeneity. Table 2(b) compares Ha-FCL with HeteroFL and its "+" variants on CIFAR-100 with mixed widths, where KD denotes current public-logit distillation. Ha-FCL exceeds the strongest direct combination by 3.27 percentage points, supporting the benefit of its coordinated distillation design.

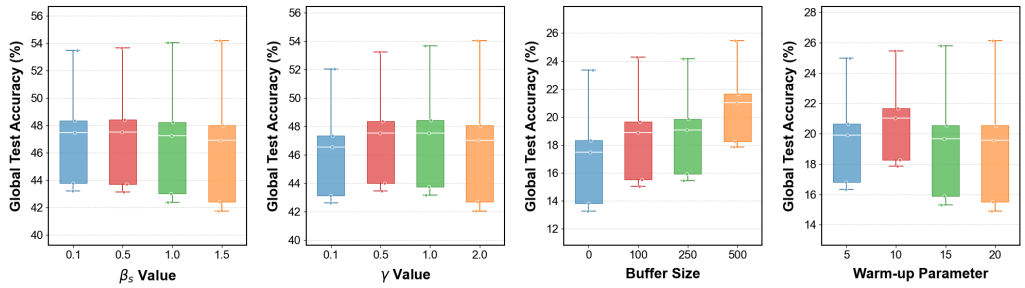


Figure 4: Sensitivity analysis of Ha-FCL with respect to key design factors. (a) Spatial distillation weight β_s. (b) Historical-logit regularization coefficient γ. (c) Public buffer size. (d) Warm-up schedule parameter η for representation co-distillation.

Sensitivity Analysis. Figure 4 summarizes the sensitivity of Ha-FCL to four key design factors. For the spatial distillation weight β_s, performance remains stable from 0.1 to 1.0, with a slight drop at 1.5, suggesting that overly strong feature alignment may hinder adaptation. For the temporal regularization coefficient γ, moderate values around 0.5–1.0 perform best, balancing historical knowledge retention and current-task plasticity. For the public buffer size, accuracy generally improves as more anchors are used, confirming the benefit of richer public-anchor coverage for server-side distillation. For the warm-up parameter η, a moderate value yields the best result, indicating that gradually strengthening representation co-distillation is preferable to either weak or overly aggressive scheduling. Overall, Ha-FCL remains stable across a practical range of hyperparameter and buffer settings.

Communication and Privacy. Ha-FCL incurs additional public-anchor response uploads without increasing communication rounds. These responses are computed on public inputs; formal privacy protection would require additional safeguards. Appendix G reports the upload costs and discusses privacy considerations.

5 Conclusion

We propose Ha-FCL, a heterogeneity-aware distillation framework for federated continual learning in heterogeneous edge environments. Ha-FCL combines width-aware federated training with server-side public-anchor distillation, addressing spatial static heterogeneity and temporal dynamic heterogeneity through Multi-Layer Feature Distillation, Historical-Logit Regularization, and Representation Co-Distillation. Extensive experiments across multiple datasets, non-IID levels, and model-width configurations show that Ha-FCL consistently outperforms strong reference baselines, highlighting the benefit of explicitly disentangling heterogeneous factors in FCL. Future work will explore theoretical convergence analysis, broader architecture families, adaptive public-buffer construction, and stronger privacy/fairness safeguards for real-world deployment.

Acknowledgments and Disclosure of Funding

This work is supported by the National Natural Science Foundation of China under grants 625B2073.

References

Sara Babakniya, Zalan Fabian, Chaoyang He, Mahdi Soltanolkotabi, and Salman Avestimehr. A data-free approach to mitigate catastrophic forgetting in federated class incremental learning for vision tasks. *Advances in Neural Information Processing Systems*, 36:66408–66425, 2023.

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191, 2017.

Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *European conference on computer vision*, pages 556–572. Springer, 2018.

Vishnu Pandi Chellapandi, Liangqi Yuan, Stanislaw H Żak, and Ziran Wang. A survey of federated learning for connected and automated vehicles. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pages 2485–2492. IEEE, 2023.

Chuan Chen, Tianchi Liao, Xiaojun Deng, Zihou Wu, Sheng Huang, and Zibin Zheng. Advances in robust federated learning: A survey with heterogeneity considerations. *IEEE Transactions on Big Data*, 11(3):1548–1567, 2025.

Zekai Chen, Chentao Jia, Ming Hu, Xiaofei Xie, Anran Li, and Mingsong Chen. Flexfl: Heterogeneous federated learning via apoz-guided flexible pruning in uncertain scenarios. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 43(11):4069–4080, 2024.

Enmao Diao, Jie Ding, and Vahid Tarokh. Heterofl: Computation and communication efficient federated learning for heterogeneous clients. *arXiv preprint arXiv:2010.01264*, 2020.

Jiahua Dong, Lixu Wang, Zhen Fang, Gan Sun, Shichao Xu, Xiao Wang, and Qi Zhu. Federated class-incremental learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10154–10163. IEEE, 2022.

Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

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

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Samuel Horvath, Stefanos Laskaridis, Mario Almeida, Ilias Leontiadis, Stylianos Venieris, and Nicholas Lane. Fjord: Fair and accurate federated learning under heterogeneous targets with ordered dropout. *Advances in Neural Information Processing Systems*, 34:12876–12889, 2021.

Chentao Jia, Ming Hu, Zekai Chen, Yanxin Yang, Xiaofei Xie, Yang Liu, and Mingsong Chen. Adaptivefl: Adaptive heterogeneous federated learning for resource-constrained aiot systems. In *Proceedings of the 61st ACM/IEEE Design Automation Conference*, pages 1–6, 2024.

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

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

Zhiqiang Kou, Junyang Chen, Xin-Qiang Cai, Xiaobo Xia, Ming-Kun Xie, Dong-Dong Wu, Biao Liu, Yuheng Jia, Xin Geng, Masashi Sugiyama, et al. Positive-unlabeled reinforcement learning distillation for on-premise small models. *arXiv preprint arXiv:2601.20687*, 2026a.

Zhiqiang Kou, Junxiang Wu, Wenke Huang, Wenwen He, Ming-Kun Xie, Changwei Wang, Yuheng Jia, Di Jiang, Yang Liu, Xin Geng, et al. Fedharmony: Harmonizing heterogeneous label correlations in federated multi-label learning. *arXiv preprint arXiv:2604.28024*, 2026b.

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

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

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

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.

Yichen Li, Qunwei Li, Haozhao Wang, Ruixuan Li, Wenliang Zhong, and Guannan Zhang. Towards efficient replay in federated incremental learning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12820–12829. IEEE, 2024a.

Yichen Li, Yuying Wang, Haozhao Wang, Yining Qi, Tianzhe Xiao, and Ruixuan Li. Rehearsal-free continual federated learning with synergistic synaptic intelligence. *arXiv preprint arXiv:2412.13779*, 2024b.

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

Yichen Li, Hang Su, Huifa Li, Haolin Yang, Xinlin Zhuang, Haochen Xue, Haozhao Wang, and Imran Razzak. Fedcd: Towards consolidated distillation for heterogeneous federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 23256–23264, 2026a.

Yichen Li, Haozhao Wang, Hang Su, Yulong Li, Xiaoquan Yi, Yankai Jiang, Chuang Zhao, Imran Razzak, and Ruixuan Li. Cross-task calibration for asynchronous federated continual learning. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 70938–70953. PMLR, 2026b. URL `https://proceedings.mlr.press/v306/li26gx.html`.

Yichen Li, Xiuying Wang, Wenchao Xu, Haozhao Wang, Yining Qi, Jiahua Dong, and Ruixuan Li. Feature distillation is the better choice for model-heterogeneous federated learning. *Advances in Neural Information Processing Systems*, 38:104726–104744, 2026c.

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

Yan-Shuo Liang and Wu-Jun Li. Inflora: Interference-free low-rank adaptation for continual learning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23638–23647. IEEE, 2024.

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.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

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.

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

Zhuang Qi, Lei Meng, Zhaochuan Li, Han Hu, and Xiangxu Meng. Cross-silo feature space alignment for federated learning on clients with imbalanced data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 19986–19994, 2025.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pages 5533–5542. IEEE, 2017.

Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger R Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N Galtier, Bennett A Landman, Klaus Maier-Hein, et al. The future of digital health with federated learning. *NPJ digital medicine*, 3(1):119, 2020.

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

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

Stacey Truex, Ling Liu, Ka-Ho Chow, Mehmet Emre Gursoy, and Wenqi Wei. Ldp-fed: Federated learning with local differential privacy. In *Proceedings of the third ACM international workshop on edge systems, analytics and networking*, pages 61–66, 2020.

Haozhao Wang, Yichen Li, Wenchao Xu, Ruixuan Li, Yufeng Zhan, and Zhigang Zeng. Dafkd: Domain-aware federated knowledge distillation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20412–20421. IEEE, 2023.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European conference on computer vision*, pages 631–648. Springer, 2022a.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 139–149. IEEE, 2022b.

Junxiang Wu, Zhiqiang Kou, Hongwei Zeng, Wenke Huang, Biao Liu, Hanlin Gu, Yuheng Jia, Di Jiang, Yang Liu, and Xin Geng. Trustworthy federated label distribution learning under annotation quality disparity. *arXiv preprint arXiv:2605.04827*, 2026.

Abudukelimu Wuerkaixi, Sen Cui, Jingfeng Zhang, Kunda Yan, Bo Han, Gang Niu, Lei Fang, Changshui Zhang, and Masashi Sugiyama. Accurate forgetting for heterogeneous federated continual learning. In *International Conference on Learning Representations*, volume 2024, pages 50778–50796, 2024.

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

Hao Yu, Xin Yang, Xin Gao, Yan Kang, Hao Wang, Junbo Zhang, and Tianrui Li. Personalized federated continual learning via multi-granularity prompt. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4023–4034, 2024.

A Training Procedure

Algorithm 1 summarizes the overall training procedure of Ha-FCL. The procedure consists of four major stages in each communication round: width-aware local training on private client streams, public-anchor response collection, server-side width-aware aggregation, and public-anchor distillation with historical-logit caching.

Algorithm 1 Training Procedure of Ha-FCL

Input: Initial full-width global model Θ^0, width set $\mathcal{R}$, task sequence $\{1, \dots, T\}$, communication rounds $\{Q_t\}_{t=1}^{T}$, local epochs E, local learning rate ρ.
Initialize: Public distillation buffer $\mathcal{D}_{\text{buf}}^0 = \emptyset$; projectors for MFD and RCD; historical logit targets $\{\tilde{y}_r^0\}_{r\in\mathcal{R}} = \emptyset$.
for task $t = 1$ **to** T **do**
  Maintain $\mathcal{D}_{\text{buf}}^t$ under the fixed memory budget using public anchors associated with classes observed up to task t.
  for communication round $q = 1$ **to** Q_t **do**
    Server: Sample active clients $\mathcal{S}_q$ and dispatch width-specific sub-networks to clients.
    for client $k \in \mathcal{S}_q$ **in parallel do**
      Extract the local sub-network $\theta_k = \mathcal{S}_{r_k}(\Theta)$ according to width configuration r_k.
      for local epoch $e = 1$ **to** E **do**
        Update θ_k on private task data $\mathcal{T}_k^t$ by minimizing $\mathcal{L}_{\text{local}}^k$.
      end for
      Evaluate the trained client model on $\mathcal{D}_{\text{buf}}^t$ to obtain public-anchor feature responses $\{z_k^{(\ell)}(x)\}$ and logits $\{g_k(x)\}$.
      Upload local model updates and public-anchor responses to the server.
    end for
    Server: Perform width-aware aggregation to update the global model Θ.
    Construct width-specific feature centroids $\bar{z}^{(\ell)}(r; x)$ and aggregated logits $\bar{g}^{(r)}(x)$ from public-anchor responses.
    Refine Θ and the projectors using the server-side objective:

$$\mathcal{L}_{\text{server}} = \beta_s \mathcal{L}_{\text{spatial}} + \gamma \mathcal{L}_{\text{temporal}} + \lambda(q)\beta_{\text{repr}} \mathcal{L}_{\text{repr}}.$$

  end for
  Cache historical logits $\tilde{y}_r^t(x) = f_{\mathcal{S}_r(\Theta)}(x)$ for $x \in \mathcal{D}_{\text{buf}}^t$ and $r \in \mathcal{R}$ for future historical-logit regularization.
end for
Output: Final global model Θ.

B Additional Method Details

B.1 Width-Aware Aggregation

Since clients may train sub-networks with different width configurations, directly averaging all parameters is not well-defined. Following width-scalable heterogeneous FL protocols such as HeteroFL and FjORD [Diao et al., 2020, Horvath et al., 2021], Ha-FCL aggregates each parameter coordinate only over the clients whose sub-network contains that coordinate.

Let $\mathcal{I}_k$ denote the set of parameter indices included in the sub-network $\mathcal{S}_{r_k}(\Theta)$ of client k. For a parameter coordinate u in the full-width global model, define the set of active clients contributing to this coordinate as:

$$\mathcal{S}_q(u) = \{k \in \mathcal{S}_q : u \in \mathcal{I}_k\}. \tag{11}$$

Given local updates $\{\theta_k\}_{k\in\mathcal{S}_q}$, the server updates the corresponding global parameter by:

$$\Theta_u \leftarrow \begin{cases} \dfrac{\sum_{k\in\mathcal{S}_q(u)} n_k \theta_{k,u}}{\sum_{k\in\mathcal{S}_q(u)} n_k}, & \text{if } |\mathcal{S}_q(u)| > 0, \\ \Theta_u, & \text{otherwise}, \end{cases} \tag{12}$$

where n_k denotes the number of local training samples on client k. This aggregation rule reduces to standard sample-weighted federated averaging when all clients share the same architecture, while remaining compatible with nested width-scalable sub-networks.

B.2 Public-Anchor Response Aggregation

After local training, each active client evaluates its trained sub-network on the public distillation buffer $\mathcal{D}_{\text{buf}}^t$. This step does not require uploading private data. For each public anchor x, the server collects intermediate features, final-layer features, and logits from participating clients. For a width group r, the server forms width-specific teacher responses as:

$$\bar{z}^{(\ell)}(r;x) = \sum_{k \in \mathcal{S}_q^r} \omega_k z_k^{(\ell)}(x), \quad \bar{g}^{(r)}(x) = \sum_{k \in \mathcal{S}_q^r} \omega_k g_k(x), \tag{13}$$

where

$$\omega_k = \frac{n_k}{\sum_{j \in \mathcal{S}_q^r} n_j}. \tag{14}$$

These aggregated responses provide the supervision signals for Multi-Layer Feature Distillation and Representation Co-Distillation.

The sample-size weighting should be interpreted as a standard practical estimator rather than a universal robustness guarantee. Suppose each client response statistic is an unbiased estimate of a width-group semantic center and its variance approximately scales as $1/n_k$. Under this approximation, the minimum-variance unbiased linear estimator uses weights proportional to n_k. This is consistent with the sample-size weighting principle commonly used in federated aggregation and prototype-based federated learning [McMahan et al., 2017, Li et al., 2020, Tan et al., 2022]. However, if some clients are systematically noisier or lower-quality, sample size alone may be insufficient, and uncertainty-aware weighting would be a useful future extension.

B.3 Historical-Logit Caching

At the end of each task stage, Ha-FCL caches historical logits on the current public buffer. For each width group $r \in \mathcal{R}$ and each public anchor $x \in \mathcal{D}_{\text{buf}}^t$, the cached target is given by:

$$\tilde{y}_r^t(x) = f_{\mathcal{S}_r(\Theta)}(x). \tag{15}$$

During the subsequent task, these cached logits remain fixed and serve as output-level anchors. In class-incremental learning, when new classes are introduced, historical-logit regularization is applied only on the output dimensions corresponding to previously observed classes. This avoids imposing constraints on newly introduced classes while preserving the decision behavior learned from earlier tasks.

HLR follows the general intuition of output-space knowledge distillation for retaining prior functional behavior [Hinton et al., 2015, Li and Hoiem, 2017]. It can be interpreted as a controlled stability–adaptability trade-off. The cross-entropy loss on current private task data promotes adaptation to new classes, while the KL regularization term preserves historical behavior on the fixed public-anchor set. The strength of this temporal constraint is controlled by γ.

B.4 Server-Side Projectors

The projection modules used in MFD and RCD are maintained on the server and optimized during the public-anchor distillation stage. They are not deployed to clients during inference. Therefore, Ha-FCL introduces no additional inference-time computation on edge devices. The role of these projectors is to map heterogeneous feature or logit responses into compatible embedding spaces, thereby enabling stable feature-level and representation-level distillation across different width groups.

C Public Buffer and Evaluation Protocol

C.1 Public Distillation Buffer

Following common practice in federated distillation and incremental learning [Li and Wang, 2019, Lin et al., 2020, Rebuffi et al., 2017], Ha-FCL uses a compact public distillation buffer $\mathcal{D}_{\text{buf}}^t$ with a fixed memory budget $B = 500$, drawn from a reserved public candidate pool. This pool is reserved before client partitioning, excluded from private local training streams, and disjoint from the test set.

Thus, the buffer does not contain private client samples and is not used for centralized supervised training.

When new classes are observed, the buffer is updated under the same fixed budget. Let $\mathcal{Y}^{\leq t}$ denote the set of classes observed up to task t. The buffer maintains public anchors associated with these classes, and the available budget is allocated across observed classes to avoid excessive bias toward recently introduced tasks. If the number of available public anchors for a class is smaller than the allocated quota, all available anchors for that class are retained. In this buffer-management protocol, class identity metadata is used for quota allocation, class-coverage tracking, and identifying historical output dimensions for HLR; public labels are not used as supervised optimization targets.

The buffer is used for three purposes only:

- collecting feature and logit responses from trained client sub-networks;
- constructing width-specific teacher centroids and aggregated logits;
- caching historical logits for future temporal regularization.

No private client data are stored or reconstructed on the server.

C.2 Evaluation Protocol

We follow the class-incremental evaluation protocol. After each task, the server global model is evaluated on all classes observed so far. Let $\mathcal{Y}^{\leq t}$ denote the cumulative label space after task t. The Top-1 accuracy at task t is computed over the centralized test samples whose labels belong to $\mathcal{Y}^{\leq t}$. For each run, we report the Top-1 accuracy averaged over the last ten communication rounds. The final reported numbers are the mean and standard deviation over three independent random seeds.

Following standard continual learning evaluation protocols [Lopez-Paz and Ranzato, 2017, Chaudhry et al., 2018], we also report forgetting and backward transfer (BWT). Let $A_{t,i}$ denote the accuracy on task i after learning task t. The average forgetting is defined as:

$$\text{Forgetting} = \frac{1}{T-1} \sum_{i=1}^{T-1} \left(\max_{t \in \{i,\dots,T-1\}} A_{t,i} - A_{T,i} \right), \tag{16}$$

and BWT is defined as:

$$\text{BWT} = \frac{1}{T-1} \sum_{i=1}^{T-1} \left(A_{T,i} - A_{i,i} \right). \tag{17}$$

Lower forgetting and less negative BWT indicate better retention of previously learned knowledge.

D Implementation Details

All experiments use ResNet-18 [He et al., 2016] as the backbone with width multipliers $\{1, 1/2, 1/4, 1/8, 1/16\}$, corresponding to width configurations $\{a, b, c, d, e\}$. The federation contains $K = 20$ clients with a participation ratio of 0.4, i.e., 8 clients are sampled per communication round. Each incremental task is trained for 80 communication rounds. Clients perform $E = 10$ local epochs with batch size 64 using SGD. The learning rate is selected from $\{0.005, 0.01, 0.05\}$ according to the dataset.

In our ResNet-18 implementation, the MFD projectors are implemented as 1×1 convolutions. RCD applies global average pooling to the final-layer student features and teacher centroids, followed by parameter-free layer normalization of the teacher centroids.

Unless otherwise specified, server-side distillation hyperparameters and the warm-up schedule are kept fixed across datasets and heterogeneity settings. The sensitivity to key design factors, including β_s, γ, public buffer size, and the warm-up schedule parameter η, is summarized in Fig. 4. Detailed numerical results are provided in Sec. F. This protocol avoids per-setting hyperparameter retuning and provides a consistent basis for comparing different methods under heterogeneous FCL scenarios.

E Complete Experimental Results

This section reports the complete Top-1 accuracy results across all datasets, all evaluated Dirichlet coefficients $\alpha \in \{1, 5, 10\}$, and both model-width configurations. These tables complement the compact main-text comparison.

Table 3: Complete results on CIFAR-10. Results are reported as mean±std.

Method	$\alpha = 1$		$\alpha = 5$		$\alpha = 10$	
	a-b-c-d-e	*a-e*	*a-b-c-d-e*	*a-e*	*a-b-c-d-e*	*a-e*
HeteroFL	30.91±0.15	42.23±0.19	35.73±0.13	47.25±0.57	37.24±0.37	44.65±0.19
FedCIL	34.84±0.17	37.41±0.21	36.30±0.11	38.71±0.10	37.30±0.24	38.08±0.15
AF-FCL	34.59±0.21	35.47±0.19	37.29±0.18	37.64±0.14	38.46±0.11	39.71±0.11
Re-Fed	33.12±0.05	36.36±0.05	38.38±0.31	41.22±0.03	42.63±0.06	48.34±0.21
Target	24.73±0.04	26.03±0.06	24.97±0.03	31.65±0.04	25.79±0.03	33.01±0.04
MFCL	22.69±0.07	30.09±0.18	24.19±0.03	31.02±0.39	25.41±0.29	32.95±0.05
FedWeIT	28.38±0.14	28.48±0.11	31.40±0.02	26.22±0.11	33.37±0.04	35.87±0.03
Ha-FCL	**47.21**±0.15	**49.31**±0.38	**47.45**±0.32	**50.98**±0.35	**48.24**±0.35	**49.43**±0.42

Table 4: Complete results on CIFAR-100. Results are reported as mean±std.

Method	$\alpha = 1$		$\alpha = 5$		$\alpha = 10$	
	a-b-c-d-e	*a-e*	*a-b-c-d-e*	*a-e*	*a-b-c-d-e*	*a-e*
HeteroFL	12.24±0.05	13.00±0.05	13.80±0.11	15.99±0.09	14.49±0.06	16.08±0.19
FedCIL	12.40±0.17	14.87±0.03	12.71±0.23	15.76±0.21	12.66±0.09	16.28±0.07
AF-FCL	13.85±0.13	13.68±0.16	15.94±0.21	15.34±0.13	16.12±0.22	15.53±0.14
Re-Fed	14.76±0.03	17.54±0.04	14.89±0.42	17.80±0.03	15.24±0.01	19.44±0.02
Target	11.24±0.01	10.66±0.03	13.40±0.02	13.25±0.04	13.48±0.03	14.64±0.03
MFCL	12.87±0.24	15.58±0.01	12.99±0.14	16.93±0.23	13.29±0.17	17.00±0.12
FedWeIT	11.88±0.04	13.89±0.06	11.93±0.03	14.46±0.09	12.99±0.05	14.70±0.08
Ha-FCL	**20.85**±0.07	**24.12**±0.29	**21.29**±0.10	**25.69**±0.07	**22.19**±0.09	**26.25**±0.12

Table 5: Complete results on Tiny-ImageNet. Results are reported as mean±std.

Method	$\alpha = 1$		$\alpha = 5$		$\alpha = 10$	
	a-b-c-d-e	*a-e*	*a-b-c-d-e*	*a-e*	*a-b-c-d-e*	*a-e*
HeteroFL	5.49±0.03	6.96±0.04	7.03±0.05	7.89±0.04	7.26±0.05	7.95±0.13
FedCIL	5.44±0.06	7.52±0.07	6.40±0.03	7.58±0.03	6.79±0.04	8.25±0.03
AF-FCL	2.27±0.17	3.07±0.23	2.31±0.11	3.49±0.16	2.66±0.24	4.06±0.24
Re-Fed	4.19±0.20	4.66±0.07	6.00±0.22	7.51±0.05	6.02±0.02	7.49±0.03
Target	4.59±0.02	5.48±0.07	4.97±0.02	5.63±0.05	5.28±0.02	6.23±0.06
MFCL	2.30±0.21	3.25±0.03	3.15±0.10	3.37±0.07	3.41±0.06	4.17±0.13
FedWeIT	4.01±0.02	5.23±0.07	4.08±0.03	5.89±0.07	4.86±0.11	6.15±0.07
Ha-FCL	**8.01**±0.03	**9.36**±0.07	**8.50**±0.07	**9.55**±0.04	**9.03**±0.07	**9.60**±0.08

F Additional Empirical Analyses

This section provides additional empirical analyses, including FCL-specific forgetting metrics, buffer-size sensitivity, client-number scalability, robustness to the absence of full-width clients, RCD design choices, hyperparameter sensitivity, and computation/communication overhead. Unless otherwise specified, these analyses are conducted on CIFAR-100 under the severe non-IID setting ($\alpha = 1$), where both the class space and the task sequence are more challenging than CIFAR-10.

F.1 Forgetting and backward transfer

We further report average forgetting and backward transfer (BWT) on CIFAR-10 and CIFAR-100. Values are reported in decimal scale. Lower forgetting and less negative BWT indicate better retention of previously learned knowledge. It is worth noting that the apparently favorable retention metrics of HeteroFL should be interpreted cautiously: its lower forgetting and less negative BWT largely stem from weak initial acquisition under heterogeneous continual learning, leaving less previously learned knowledge to forget. Therefore, forgetting and BWT should be interpreted together with the final cumulative accuracy reported in the main text.

Table 6: Forgetting and BWT on CIFAR-10 and CIFAR-100. Lower forgetting and less negative BWT are preferred.

Method	CIFAR-10		CIFAR-100	
	Forgetting ↓	BWT ↑	Forgetting ↓	BWT ↑
HeteroFL	0.3207	-0.3012	0.1180	-0.1000
FedCIL	0.6726	-0.6510	0.3240	-0.3216
AF-FCL	0.6302	-0.6041	0.3124	-0.3029
Re-Fed	0.6430	-0.6338	0.3219	-0.3128
Target	0.7421	-0.7165	0.5173	-0.4938
MFCL	0.5648	-0.5382	0.4215	-0.3984
FedWeIT	0.4896	-0.4630	0.3464	-0.3291
Ha-FCL	0.4797	-0.4488	0.2702	-0.2613

F.2 Sensitivity to Public Buffer Size

Table 7: Sensitivity to the public distillation buffer size on CIFAR-100 under $\alpha = 1$ and the a-b-c-d-e configuration.

Buffer size	0	50	100	250	500
Accuracy (%)	17.25	17.33	18.68	18.90	20.85

As shown in Table 7, performance improves as the number of public anchors increases, indicating that richer anchor coverage benefits both spatial and temporal distillation. Nevertheless, Ha-FCL remains functional even with a small buffer: using 100 anchors retains approximately 90% of the performance obtained with the full 500-anchor budget.

F.3 Sensitivity to the Number of Clients

Table 8: Sensitivity to the number of clients on CIFAR-100 under $\alpha = 1$ and the a-b-c-d-e configuration.

Number of clients	10	20	50
Accuracy (%)	16.57	20.85	18.18

The performance improves when increasing the client number from 10 to 20, suggesting that a moderate number of clients provides more diverse collaborative knowledge. When the number of clients further increases to 50, local data become more fragmented under the fixed data budget and non-IID partition, which makes local optimization and response aggregation more challenging.

F.4 Robustness without Full-Width Clients

Table 9: Performance when full-width clients are removed on CIFAR-100 under $\alpha = 1$.

Configuration	HeteroFL	Re-Fed	Ha-FCL
b-c-d-e	13.25	16.10	**17.08**
b-e	14.56	16.62	**17.29**

Table 9 shows that Ha-FCL still outperforms the compared methods even when the largest-width clients are absent. This indicates that the improvement is not solely due to a few full-width clients acting as dominant teachers.

F.5 Analysis of Representation Co-Distillation

Table 10: Effect of different RCD supervision sources on CIFAR-100 under $\alpha = 1$.

Method	*a-b-c-d-e*	*a-e*
Only logits	18.01	21.13
Only features	19.90	22.05
Ha-FCL	**20.85**	**24.12**

The full RCD design achieves the best performance in both configurations, suggesting that feature-level and output-level supervision provide complementary information.

Table 11: Comparison of RCD loss functions on CIFAR-100 under $\alpha = 1$ and the *a-b-c-d-e* configuration.

RCD loss	MSE	Smooth L1	Cosine
Accuracy (%)	20.19	**20.85**	20.12

Smooth L1 [Girshick, 2015] achieves the best result. This is consistent with our motivation: in early heterogeneous training, teacher-student discrepancies can be large, making MSE overly sensitive to outliers. Cosine similarity mainly constrains directional consistency and may overlook magnitude mismatch.

F.6 Additional Hyperparameter Sensitivity

Table 12: Sensitivity to the spatial distillation coefficient β_s and historical-logit coefficient γ in a representative CIFAR-10 mixed-width setting.

β_s	0.1	0.3	0.5	0.8	1.0	1.5
Accuracy (%)	47.25	47.31	47.22	47.15	46.99	46.66
γ	0.1	0.5	0.8	1.0	1.5	2.0
Accuracy (%)	46.32	47.30	47.33	47.31	47.09	46.76

Table 13: Sensitivity to the warm-up schedule parameter η on CIFAR-100 under $\alpha = 1$ and the a-b-c-d-e configuration.

Warm-up parameter η	5	10	15	20
Accuracy (%)	19.73	**20.85**	19.44	19.33

A moderate η performs best. If the transition is too gradual, representation coordination may be insufficient; if it is too sharp, the sudden increase in alignment strength may interfere with current-task adaptation.

F.7 Computation and Projector Overhead

Table 14: Parameter and FLOP statistics under the ResNet-18 width-scalable setting. FLOPs are measured for a single forward pass.

Component	Params (M)	FLOPs (M)
Full-width ResNet-18	11.22	557.94
Width-b	2.82	140.56
Width-c	0.71	35.67
Width-d	0.18	9.19
Width-e	0.05	2.43
Total projector overhead	1.15	–

The total number of projector parameters is approximately 10.22% of the full-width ResNet-18 backbone. These projectors are used only during server-side public-anchor distillation and are not deployed during client inference. On CIFAR-100, average training time per round increases from 9.230 s for HeteroFL to 13.121 s for Ha-FCL (+3.891 s). Measurements average ten rounds on a single RTX 4090 with batch size 64 and sequential client simulation, excluding network communication.

G Communication Payload and Privacy Scope

Ha-FCL follows the standard federated training schedule and does not introduce additional communication rounds. In each round, the server broadcasts the corresponding width-specific sub-network to each active client, and each client uploads its local model update after private training. In addition, Ha-FCL requires public-anchor responses for server-side distillation.

For an active client k, the additional uploaded payload is:

$$\mathcal{U}_k^q = \left\{ z_k^{(\mathrm{mid})}(x), z_k^{(\mathrm{fin})}(x), g_k(x) \ : \ x \in \mathcal{D}_{\mathrm{buf}}^t \right\}, \tag{18}$$

where all responses are computed by forwarding the shared public distillation buffer through the trained local sub-network. Therefore, the uploaded feature/logit responses are tied to public anchors rather than private client samples.

The granularity of the additional payload is per public anchor and per communication round. Let $B = |\mathcal{D}_{\mathrm{buf}}^t|$ denote the public buffer size, d_{mid} and d_{fin} denote the per-anchor element counts of the transmitted intermediate and final feature tensors, and C_t denote the transmitted logit dimension. The additional communication cost per participating client is:

$$\mathcal{O}\big(B(d_{\mathrm{mid}} + d_{\mathrm{fin}} + C_t)\big). \tag{19}$$

If transmitted in 32-bit floating point format, this corresponds to approximately:

$$4B(d_{\mathrm{mid}} + d_{\mathrm{fin}} + C_t) \tag{20}$$

bytes before compression. This cost is independent of the size of the private local dataset and can be reduced by response compression, lower-precision transmission, or using a smaller public-anchor subset.

Table 15: Additional public-response upload per active client per round on CIFAR-100 ($B = 500$, float32, decimal MB), excluding one-time initialization.

Width	Response upload (MB)
a	164.240
b	82.320
c	41.360
d	20.880
e	10.640

In the current implementation, the server observes these public-anchor responses in the clear. Thus, Ha-FCL is not a parameter-only communication protocol. However, the additional signals are model responses on a shared public/proxy set, not features or logits computed from private client samples. The residual privacy risk is therefore similar to the indirect leakage that may arise from model updates or proxy-set outputs visible to the server, rather than direct exposure of private-sample statistics.

When a width group contains only one active client in a round, the group-wise aggregated response degenerates to that client's response on the public distillation buffer. This reduces the smoothing benefit of aggregation and may make the teacher signal less stable. Nevertheless, the source of the uploaded information remains the shared public-anchor set. In practical deployments, Ha-FCL can be combined with secure aggregation [Bonawitz et al., 2017], differential privacy [Truex et al., 2020], response compression, or minimum-group-size filtering to further reduce the privacy risk of server-visible statistics.

Importantly, Ha-FCL does not increase inference-time complexity on clients, because all distillation projectors and server-side objectives are discarded after training. Each client only deploys its assigned width-specific sub-network during inference.

H Broader impacts

Ha-FCL is a methodological contribution for federated continual learning under heterogeneous edge environments. Its potential positive impact lies in enabling distributed models to adapt over time without centralizing private client data, which may benefit privacy-sensitive applications such as mobile intelligence and medical or industrial edge systems. Potential negative impacts include residual privacy leakage through model updates or public-anchor responses, as well as fairness concerns if heterogeneous clients receive systematically different model capacities. We discuss these risks and possible safeguards, including secure aggregation, differential privacy, response compression, and minimum-group-size filtering, in Appendix G and Appendix I.

I Limitations and Future Directions

Ha-FCL assumes access to a compact public distillation buffer that is disjoint from private client data and the test set. Although this setting is common in federated distillation, the representativeness of the public anchors may affect the quality of feature/logit supervision, especially when the public buffer is highly mismatched with private client distributions. Developing adaptive and privacy-preserving public-anchor construction strategies is therefore an important future direction.

Ha-FCL also introduces additional server-visible public-anchor responses beyond standard parameter exchange. These responses are computed on public anchors rather than private client samples, but they may still carry indirect information through the trained local models. In privacy-sensitive deployments, additional safeguards such as secure aggregation [Bonawitz et al., 2017], differential privacy [Truex et al., 2020], response compression, or minimum-group-size filtering may be required.

The current implementation focuses on task-aware class-incremental learning, where task boundaries are available for updating historical anchors and cached logits. Extending Ha-FCL to task-agnostic FCL would require additional mechanisms such as drift detection or change-point estimation. Moreover, our experiments mainly study width-scalable ResNet architectures. A possible extension to non-nested architectures is to aggregate compatible models within architecture families and transfer knowledge across families through public-anchor distillation, using architecture-specific adapters to

align corresponding feature representations. Evaluation across these broader architecture families remains future work.

Finally, the absolute performance on Tiny-ImageNet remains low due to the combined difficulty of large class space, non-IID partitioning, continual task shifts, and model heterogeneity under lightweight backbones. Improving performance in more challenging large-scale FCL settings and establishing formal convergence guarantees under coupled spatial and temporal heterogeneity are important directions for future work.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction state the main contributions of Ha-FCL, including the decomposition of heterogeneity into spatial static and temporal dynamic components and the proposed distillation modules for addressing them. These claims are supported by the methodology in Sec. 3 and the empirical results in Sec. 4.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The limitations are discussed in Appendix I, including the reliance on a public distillation buffer, residual privacy risks of public-anchor responses, the task-aware class-incremental setting, and the focus on width-scalable ResNet architectures.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

   Justification: The paper does not present formal theoretical results or theorem-proof claims. The mathematical expressions are used to define the optimization objectives and algorithmic components of Ha-FCL.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Sections 3–4 and Appendices A–D describe the method, training procedure, public-buffer construction, and evaluation protocol. The supplementary code package provides the implementation, experimental configurations, and reproduction instructions.

   Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: The experiments use publicly available benchmark datasets, including CIFAR-10, CIFAR-100, and Tiny-ImageNet, which are cited in Sec. 4.1. A code package with implementation details, configuration files, and reproduction instructions is provided in the supplemental material.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Sec. 4.1 specifies the datasets, class-incremental splits, non-IID Dirichlet partitions, client number, participation ratio, communication rounds, local epochs, optimizer, batch size, learning-rate search space, width configurations, and memory budget. Additional implementation details are provided in Appendix D, and the experimental configurations and reproduction instructions are included in the supplementary code package.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: The main results are reported as mean and standard deviation over three independent random seeds, as stated in Sec. 4.1. The corresponding values are shown in Table 1 and the complete appendix tables.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [No]

   Justification: The paper reports model sizes, FLOPs, and representative per-round training times on a single RTX 4090 in Appendix F.7. However, memory requirements, per-experiment runtimes, and total compute across the full study are not fully documented.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research uses standard public image classification benchmarks and does not involve human subjects, sensitive personal data, or high-risk data collection. The privacy scope and residual risks of public-anchor responses are discussed in Appendix G.

Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Appendix H discusses the potential benefits of continual adaptation on heterogeneous devices while keeping private training data local, together with potential privacy leakage and fairness risks. Possible safeguards and limitations are discussed in Appendices G and I.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper does not release high-risk assets such as large pretrained language models, image generators, scraped datasets, or models intended for direct deployment in sensitive applications. The method is evaluated on standard public image classification benchmarks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: The existing datasets used in the experiments are standard public benchmarks and are properly cited in Sec. 4.1. Dataset access instructions and license or usage information are documented in the supplemental code package where available.

    Guidelines:

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

13. **New assets**

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

    Answer: [Yes]

    Justification: The paper provides a code package as a new research artifact. The package includes documentation, configuration files, and instructions for reproducing the main experimental results.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing, user studies, or research with human subjects.

Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve human subjects, crowdsourcing experiments, or collection of participant data, so IRB approval or equivalent review is not applicable.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: The core method development, experiments, and analysis do not use LLMs as an important, original, or non-standard component of the research.

    Guidelines:

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