DECOUPLED IN TIME, ALIGNED IN SPACE: GEOMETRIC ALIGNMENT FOR TEMPORALLY HETEROGENEOUS FEDERATED CONTINUAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated Continual Learning (FCL) extends federated learning to evolving local data streams, but existing methods largely assume comparable task progression across clients. In realistic edge environments, heterogeneous task workloads cause clients to advance through continual learning at different rates, producing persistent temporal misalignment across learning states. This misalignment biases global aggregation and progressively distorts previously accumulated knowledge. We formalize this problem as *temporal heterogeneity* and propose **TARDIS** (**T**emporal-**A**nchored **R**epresentation Alignment for **D**rift-**I**nhibiting **S**tabilization). TARDIS progressively constructs a global Equiangular Tight Frame (ETF) as continual knowledge expands, providing shared geometric anchors that maintain representational consistency across heterogeneous client states. Extensive experiments on four continual-learning benchmarks show that TARDIS consistently outperforms representative FCL baselines under both Task-IL and Class-IL settings, with particularly strong gains in global-class prediction and knowledge retention.

1 INTRODUCTION

Centralized machine learning rests on an increasingly fragile premise that data can be assembled into a centralized corpus before optimization begins. Yet the experience from which models must learn is generated overwhelmingly at the network edge, across personal devices, autonomous platforms, sensors, and organizational silos (Deng et al., 2020; Wang et al., 2025a). Federated Learning (FL) responds to this decentralized reality by coordinating distributed clients to optimize a shared model through locally computed updates, while keeping raw data at their sources (Konečnỳ et al., 2016; McMahan et al., 2017; Li et al., 2020c).

The decentralized nature of edge learning extends beyond where data are stored to how they evolve over time. FL addresses the spatial decentralization of data, but typically optimizes over stationary local data distributions (Li et al., 2020b). Therefore, conventional FL can be viewed as a special case in which temporal evolution collapses to a stationary learning process. This assumption is increasingly restrictive in real-world edge systems, where data continue to emerge after training begins and may evolve as devices encounter new users, contexts, classes, or operating conditions (Shaheen et al., 2022; Soltoggio et al., 2024). Federated Continual Learning (FCL) relaxes this assumption by allowing each client to learn from a sequence of incrementally arriving tasks or data distributions while collaboratively retaining and transferring knowledge across the federation (Yoon et al., 2021; Huang et al., 2022; Zhang et al., 2023; Babakniya et al., 2023; Qi et al., 2023; Wang et al., 2024; Wuerkaixi et al., 2024; Qi et al., 2026b).

The continual and federated nature of FCL poses two fundamental challenges: preserving knowledge over time and learning across heterogeneous clients. Existing FCL research has primarily focused on catastrophic forgetting (Kirkpatrick et al., 2017), aiming to retain global capabilities acquired from earlier tasks as clients continually adapt to newly arriving data (Yoon et al., 2021; Qi et al., 2023; Zhang et al., 2023; Babakniya et al., 2023). Notably, the dynamics of forgetting are inherently shaped by the heterogeneous environments where the continual adaptation takes place. In

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

standard FL, heterogeneity is commonly characterized by non-IID local data distributions, disparities in computational and communication resources, and varying participation patterns across clients (Li et al., 2020c; Karimireddy et al., 2020; Wang et al., 2020; Li et al., 2020b). FCL inherits these discrepancies while embedding them within an evolving learning process, where client distributions themselves change as new tasks or data arrive (Li et al., 2024). Consequently, heterogeneity in FCL extends beyond cross-client misalignment at a given stage of training to discrepancies associated with continual progression itself. It spans two complementary axes: a cross-client axis capturing disparities among participants, and a temporal axis capturing distributional variation across their evolving learning states.

This temporal dimension becomes particularly pronounced when task workloads are heterogeneous across clients. Variations in local data volume and task complexity impose different training and communication demands, causing clients to progress through task streams at different rates. This gives rise to **task-stream misalignment**, where updates in the same global aggregation may originate from different learning states and encode distinct task-specific knowledge. Such misalignment can induce **aggregation bias**, as particular clients or learning states become disproportionately represented. Over continual progression, this persistent bias can lead to **global knowledge drift**, progressively skewing the shared model toward dominant learning states while weakening knowledge acquired elsewhere.

Despite its practical relevance, the temporal dimension of heterogeneity remains comparatively underexplored in FCL. Existing methods primarily address catastrophic forgetting and knowledge transfer under heterogeneous local data, without explicitly modeling how heterogeneous task workloads induce divergent client progression. To address this challenge, we propose **TARDIS** (**T**emporal-**A**nchored **R**epresentation Alignment for **D**rift-**I**nhibiting **S**tabilization), which progressively constructs a global Equiangular Tight Frame (ETF) as the knowledge space expands. By anchoring local representations to this evolving geometry, TARDIS maintains cross-client representational consistency across learning states and mitigates the propagation of temporal misalignment into global knowledge drift. We summarize the main contributions as follows:

- We investigate **temporal heterogeneity** in FCL induced by heterogeneous task workloads, and analyze its impact on continual client progression and global aggregation.
- We propose **TARDIS**, which progressively constructs a global ETF to provide shared geometric anchors for stabilizing representations across heterogeneous learning states.
- Extensive experiments on multiple benchmarks show that TARDIS consistently outperforms representative FCL baselines under both memory-buffer and data-free settings.

2 Related Work

2.1 Federated Continual Learning

Continual learning enables models to learn from sequentially evolving tasks while retaining prior knowledge, with common approaches including regularization, distillation, parameter isolation, and replay (Kirkpatrick et al., 2017; Rebuffi et al., 2017; Parisi et al., 2019; Li & Hoiem, 2017; Shin et al., 2017; Mirzadeh et al., 2020; De Lange et al., 2021). FCL extends this setting to decentralized clients with evolving local data streams, requiring knowledge preservation and transfer across the federation. Existing methods address this through parameter decomposition and selective transfer (Yoon et al., 2021), distillation (Ma et al., 2022; Huang et al., 2022), generative or data-free replay (Qi et al., 2023; Zhang et al., 2023; Babakniya et al., 2023), and task tracing (Wang et al., 2024). Recent approaches further consider heterogeneous client knowledge through selective forgetting, class-aware interaction, multi-scale expertise, and balanced replay (Wuerkaixi et al., 2024; Xu et al., 2026; Yi et al., 2026; Qi et al., 2026b). However, these methods largely presume comparable task progression across clients, whereas heterogeneous task demands can place clients at different stages of continual learning.

2.2 Heterogeneous Federated Learning

The distributed nature of FL induces substantial heterogeneity across clients, spanning statistical, system, participation, and model dimensions (Li et al., 2020a; Zhu et al., 2021; Ye et al., 2023).

Existing work addresses non-IID optimization through proximal regularization, drift correction, normalized aggregation, dynamic regularization, representation alignment, and client-specific normalization (Li et al., 2020c; Karimireddy et al., 2020; Wang et al., 2020; Acar et al., 2021; Li et al., 2021a;b), primarily under stationary local learning objectives. FCL further couples such cross-client heterogeneity with continually evolving local knowledge, with recent studies considering conflicting knowledge, class imbalance, and dynamic distribution shifts (Wuerkaixi et al., 2024; Li et al., 2024; Xu et al., 2026; Qi et al., 2026b). Nevertheless, these formulations still primarily characterize heterogeneity through differences in client data, systems, or acquired knowledge, while assuming broadly aligned temporal progression. Heterogeneous task workloads can break this alignment, introducing persistent progression disparities and a distinct temporal dimension of heterogeneity.

2.3 Neural Collapse

Papyan et al. (2020) observed that deep classifiers on balanced data converge toward a regular last-layer geometry in which within-class variability collapses, class means form a simplex ETF, and classifier directions align with them. Building on this, Yang et al. (2022) fix the classifier as a simplex ETF to preserve this geometry under class imbalance. In FL, the same principle has been used to mitigate classifier bias under non-IID data (Li et al., 2023), with extensions to long-tailed (Xiao et al., 2024; Yan et al., 2025), personalized (Du & Li, 2025), and multi-label (Peng et al., 2025) settings. CL methods further use ETF geometry as stable and well-separated class targets across sequential tasks, typically assume a known label space, reserving vertices for future classes (Yang et al., 2023a), maintaining a fixed terminus (Yang et al., 2023b), or incorporating fixed prototypes into contrastive learning (Dang et al., 2025); Wang et al. (2025b) instead expand the ETF as new classes appear while limiting displacement of existing targets. However, FCL places a different demand on ETF geometry, requiring class anchors to remain mutually consistent across distributed clients as the knowledge space evolves throughout training.

3 Preliminaries

3.1 Problem Formulation: Federated Temporal Heterogeneity

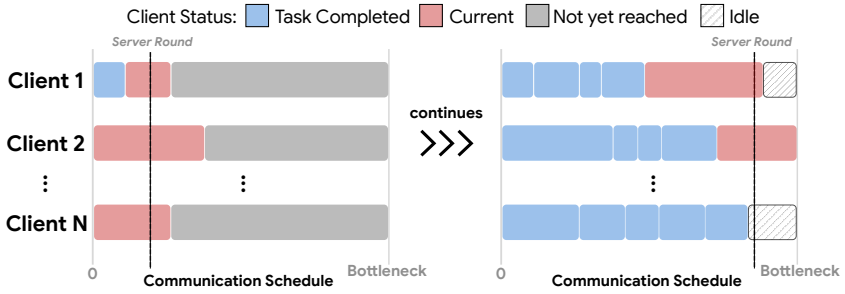


Figure 1: Example of temporal heterogeneity in FCL at early (left) and later (right) training stages. Clients follow heterogeneous per-task round budgets, causing different progression rates. A client that completes its sequence goes idle, awaiting the remaining clients, while the slowest client determines the overall communication horizon.

We consider an FCL system of N clients learning a sequence of T tasks under a central server. Unlike conventional FCL with synchronized task transitions, clients may hold heterogeneous per-task workloads and therefore progress through continual learning at different rates. We term the resulting disparity in learning states *temporal heterogeneity*, which introduces persistent misalignment into local adaptation and global aggregation.

Setup. Clients are indexed by $i \in \{1, \dots, N\}$ and tasks by $\mathcal{T} = (\tau_0, \dots, \tau_{T-1})$, where task τ_t has label space $\mathcal{C}_t$ with $\mathcal{C}_t \cap \mathcal{C}_{t'} = \varnothing$ for $t \neq t'$. For each class $c \in \mathcal{C}_t$, client allocation proportions are sampled from a symmetric Dirichlet distribution $\mathrm{Dir}(\alpha \mathbf{1}_N)$, and class-$c$ samples are distributed across the N clients accordingly. This produces client partition $\mathcal{D}_{i,t}$ with size $n_{i,t} = |\mathcal{D}_{i,t}|$. Smaller α yields more imbalanced allocations and hence greater disparities in local task volumes. Learning proceeds over communication rounds $r = 1, \dots, R$, with K local epochs and batch size B.

Per-task client workload. We define the *round budget* $R_{i,t}$ as the number of communication rounds client i requires to complete task τ_t. As one round processes $\lceil n_{i,t}/B \rceil$ mini-batches per epoch, the workload is governed by the local volume $n_{i,t}$, giving

$$R_{i,t} \;=\; \max\!\left(1,\; \left\lceil \frac{K \cdot n_{i,t}}{B} \right\rceil\right). \tag{1}$$

Since $R_{i,t}$ scales with $n_{i,t}$, the Dirichlet-induced imbalance in partition sizes translate directly into disparate budgets, and hence disparate task-transition schedules, across clients.

Temporal heterogeneity. Let $\phi_i(r) \in \{0, \ldots, T-1\}$ denote the task client i is engaged on at round r,

$$\phi_i(r) \;=\; \max\Big\{\, t \;:\; \textstyle\sum_{s=0}^{t-1} R_{i,s} \;<\; r \Big\}, \qquad \textstyle\sum_{s=0}^{-1} R_{i,s} \triangleq 0, \tag{2}$$

i.e., the highest task whose starting round precedes r. A system is *temporally heterogeneous* when clients do not all share the same task index at some round,

$$\exists\, r \;\text{ s.t. }\; \big|\{\phi_i(r) : i = 1, \ldots, N\}\big| > 1. \tag{3}$$

Because the budgets $R_{i,t}$ differ whenever the volumes $n_{i,t}$ differ, this condition holds generically under $\mathrm{Dir}(\alpha)$ partitioning. The total horizon is set by the slowest client, $R = \max_i \sum_{t=0}^{T-1} R_{i,t}$.

Observation I: Cross-state aggregation under temporal misalignment. As visualized in Figure 1, temporal heterogeneity extends conventional cross-client heterogeneity to the learning-state dimension. At round r, clients with $\phi_i(r) \neq \phi_j(r)$ optimize distinct task objectives and, in the task-incremental setting, potentially disjoint label spaces:

$$P_i(X, Y \mid \phi_i(r)) \neq P_j(X, Y \mid \phi_j(r)), \qquad \mathcal{C}_{\phi_i(r)} \cap \mathcal{C}_{\phi_j(r)} = \varnothing. \tag{4}$$

Global aggregation therefore combines updates representing different stages of knowledge acquisition, introducing conflicting optimization directions and bias toward task states with greater aggregate influence.

Observation II: Cumulative global knowledge drift. Temporal misalignment makes such interference persistent across continual progression. Client-specific task transitions cause global updates learned from one set of states to interact with local models occupying others, while subsequent local adaptation feeds these discrepancies back into aggregation. Repeated across rounds, this interaction compounds local forgetting with cross-state interference, progressively distorting the globally accumulated knowledge.

3.2 Continual-Learning Objective and Evaluation Protocols

At round r, client i trains local parameters $\theta_i^{(r)}$ on its current partition $\mathcal{D}_{i,\phi_i(r)}$ alone, without access to completed tasks, while the resulting model is still expected to retain predictive performance on every task that client has encountered. For candidate parameters θ, we express this requirement as the task-balanced historical risk:

$$\mathcal{R}_i^{(r)}(\theta) \;=\; \frac{1}{\phi_i(r)+1} \sum_{s=0}^{\phi_i(r)} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_{i,s}}\big[\ell(g_\theta(\mathbf{x}), y)\big], \tag{5}$$

where $g_\theta(\mathbf{x})$ denotes the vector of class scores. The protocols below differ only in which classes participate in prediction. Because past partitions cannot be revisited, minimizing the current-task term of Eq. 5 can increase its past-task terms, giving rise to continual forgetting.

We report the same learned model under two protocols. For a test sample from task τ_s ($s \leq t$, with $\tau_0, \ldots, \tau_t$ the evaluation history), Task-IL (i.e., incremental learning) is given the task identity and restricts prediction to that task's label space,

$$\hat{y}_{\mathrm{TIL}} \;=\; \arg\max_{c \in \mathcal{C}_s} g_\theta(\mathbf{x})_c, \tag{6}$$

while Class-IL withholds task identity and predicts over every class observed so far,

$$\hat{y}_{\mathrm{CIL}} \;=\; \arg\max_{c \in \mathcal{C}_{0:t}} g_\theta(\mathbf{x})_c, \qquad \mathcal{C}_{0:t} \;=\; \bigcup_{u=0}^{t} \mathcal{C}_u. \tag{7}$$

Task-IL therefore measures within-task discrimination alone, whereas Class-IL additionally requires scores learned by clients at different task states to remain comparable on a common scale – a requirement that motivates the shared class geometry introduced next.

3.3 Neural Collapse

Let $\boldsymbol{\mu}_c$ denote the mean representation of class c, $\boldsymbol{\mu}_G$ the global mean, and $\mathbf{w}_c$ its classifier vector. In the terminal phase of balanced classification, neural collapse (NC) describes a highly structured representation geometry characterized by four properties (Papyan et al., 2020):

(NC1) *Variability collapse.* Within-class features concentrate around their class mean $\boldsymbol{\mu}_c$.

(NC2) *Simplex ETF structure.* The centered class means $\boldsymbol{\mu}_c - \boldsymbol{\mu}_G$ become equinorm and equiangular, converging to a simplex Equiangular Tight Frame (ETF).

(NC3) *Self-duality.* Classifier directions align with the centered class means, $\mathbf{w}_c \propto \boldsymbol{\mu}_c - \boldsymbol{\mu}_G$.

(NC4) *Nearest-class-center prediction.* Classification reduces to selecting the class center nearest to $f_\theta(\mathbf{x})$.

Writing $\mathbf{E} = [\mathbf{e}_1, \ldots, \mathbf{e}_K] \in \mathbb{R}^{d\times K}$ for the ETF of K classes, we use this construction for $d \geq K$:

$$\mathbf{E} = \sqrt{\tfrac{K}{K-1}}\mathbf{U}\Big(\mathbf{I}_K - \tfrac{1}{K}\mathbf{1}_K\mathbf{1}_K^\top\Big), \qquad \mathbf{U}^\top\mathbf{U} = \mathbf{I}_K, \tag{8}$$

whose columns are unit-norm and equiangular, with the maximal pairwise separation attainable under the simplex constraint: $\mathbf{e}_c^\top \mathbf{e}_{c'} = -\frac{1}{K-1}$ for every $c \neq c'$. The geometry is therefore fixed by K alone up to the choice of orthonormal basis $\mathbf{U}$, which sets its orientation in $\mathbb{R}^d$.

These properties motivate complementary components of TARDIS. NC2 provides the simplex ETF as the target class geometry, while NC3 supports aligning classifier directions with the same frame. NC1 further justifies representing each class by a single vertex as within-class features concentrate, and NC4 supports using the resulting geometry for classification.

Fixed ETF classifier. Following the fixed-ETF principle of Yang et al. (2022), TARDIS uses the current ETF as the classifier during each local optimization round, $g_\theta(\mathbf{x})_c = \mathbf{e}_c^\top f_\theta(\mathbf{x})$. The classifier directions remain fixed within that round, while the ETF itself may be updated between rounds as the global class space expands and the server broadcasts a new frame. Local representations are aligned to the current class anchor through dot regression. For unit-norm vectors $\mathbf{a}$ and $\mathbf{b}$,

$$\mathcal{D}_{\mathrm{DR}}(\mathbf{a}, \mathbf{b}) \;=\; \tfrac{1}{2}\big(\mathbf{a}^\top\mathbf{b} - 1\big)^2, \qquad \|\mathbf{a}\|_2 = \|\mathbf{b}\|_2 = 1, \tag{9}$$

and we use $\mathcal{D}_{\mathrm{DR}}(\hat{f}_\theta(\mathbf{x}), \mathbf{e}_y)$, where $\hat{f}_\theta(\mathbf{x}) = f_\theta(\mathbf{x})/\|f_\theta(\mathbf{x})\|_2$. The shared class-to-vertex mapping therefore provides a common geometric reference across heterogeneous client states, forming the basis of the progressive anchoring mechanism in Section 4.

4 Method

4.1 Temporal-Anchored Alignment via ETF Vertex Anchoring

Temporal heterogeneity causes clients to reach task boundaries at different communication rounds, making globally synchronized interventions ineffective. TARDIS instead maintains a shared geometric reference in which each class is assigned a consistent vertex independent of when or where it is encountered. As illustrated in Figure 2, the server first recovers a global ETF from the learned representation space and then progressively expands this geometry as new classes emerge.

Recovering the base orientation. Equation 8 leaves the orientation matrix $\mathbf{U}$ unspecified. Rather than using a random orientation, which would require unnecessary representation rotation, TARDIS estimates $\mathbf{U}$ from the federation's learned features. Before constructing the base frame, clients train with the standard federated objective. Upon completing a base task, client i uploads the sum of its normalized penultimate features and sample count for each $c \in \mathcal{C}_t$. The server aggregates these statistics into sample-weighted class means $\bar{\boldsymbol{\mu}}_c$ and collects their centered, normalized forms in $\tilde{\mathbf{M}} \in \mathbb{R}^{K_b\times d}$. The base orientation is obtained by

$$\mathbf{A} \;=\; \sqrt{\tfrac{K_b-1}{K_b}}\,\tilde{\mathbf{M}}^\top\Big(\mathbf{I}_{K_b} - \tfrac{1}{K_b}\mathbf{1}\mathbf{1}^\top\Big)^+, \qquad \mathbf{U} \;=\; \arg\min_{\mathbf{V}^\top\mathbf{V}=\mathbf{I}} \|\mathbf{V} - \mathbf{A}\|_F \;=\; \mathbf{P}\mathbf{Q}^\top, \tag{10}$$

where $\mathbf{A} = \mathbf{P}\boldsymbol{\Sigma}\mathbf{Q}^\top$ is a thin SVD. The resulting $\mathbf{U}$ is the orthonormal orientation closest to the empirical class geometry, from which the base ETF is constructed via equation 8. Construction

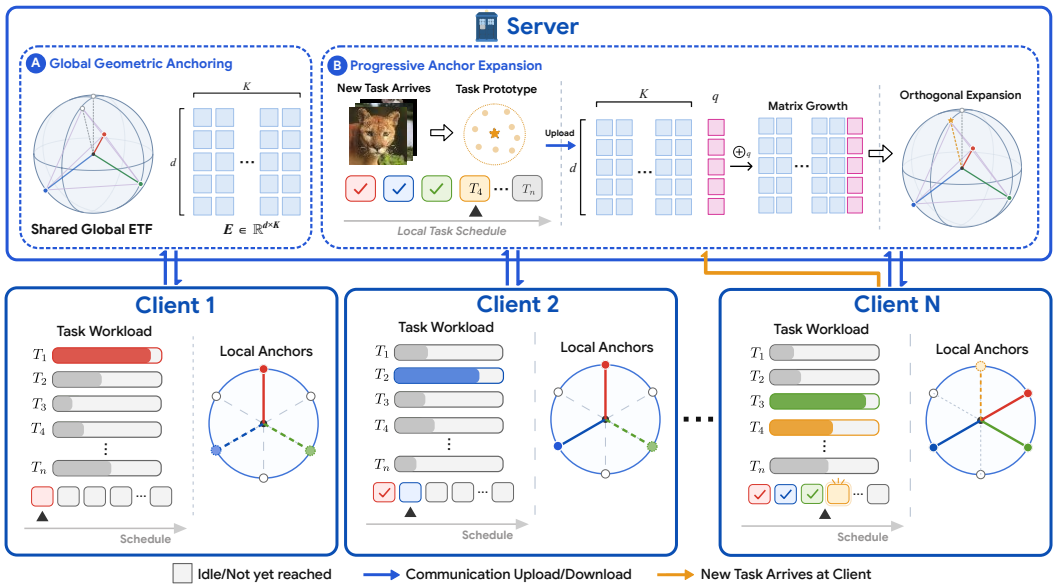


Figure 2: **Overview of TARDIS. (A)** A globally shared ETF $\mathbf{E} \in \mathbb{R}^{d\times K}$ provides coordinated class anchors across heterogeneous client states. **(B)** As continual learning introduces new classes, the server progressively expands the ETF over the evolving knowledge space and broadcasts the updated geometry to clients, which align their local representations to the anchors available at their current learning states.

begins once a fraction ρ_{base} of clients has deposited the required statistics; clients completing the base phase earlier remain idle until the frame becomes available.

Progressive expansion. When new classes appear, the server augments the existing basis with $|\mathcal{C}_t|$ orthonormal directions using Gram-Schmidt, $\mathbf{U} \leftarrow [\mathbf{U}, \mathbf{U}_\perp]$, and recomputes $\mathbf{E}$ through equation 8. The previously established basis directions are retained, while re-centering and re-scaling update the complete ETF to accommodate the enlarged class space. The updated frame is broadcast to all clients, each of which uses the subset of anchors corresponding to the knowledge it has reached. Additionally, an ablation of the base orientation and expansion strategy is provided in Appendix A.1.3, showing the benefit of empirical initialization and progressive ETF growth over simpler random or pre-allocated alternatives.

Anchoring the client objective. For a client learning task τ_t, TARDIS combines within-task discrimination, ETF alignment, and representation distillation. For minibatch $\mathcal{B} \subset \mathcal{D}_{i,t}$, let $\hat{f}_\theta \equiv \hat{f}_\theta(\mathbf{x})$ and $\hat{f}^{\,\text{old}} \equiv \hat{f}_{\theta_i^{(t-1)}}(\mathbf{x})$. The local objective is

$$\mathcal{L} \;=\; \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{B}}\Big[\lambda_{\text{ce}}\,\ell_{\text{CE}}\big(h_\theta(\mathbf{x})\big|_{\mathcal{C}_t},\, y\big) \;+\; \lambda_{\text{al}}\,\mathcal{D}_{\text{DR}}\big(\hat{f}_\theta,\, \mathbf{e}_y\big) \;+\; \lambda_{\text{ds}}\,\mathcal{D}_{\text{DR}}\big(\hat{f}_\theta,\, \hat{f}^{\,\text{old}}\big)\Big], \tag{11}$$

where $h_\theta(\mathbf{x})\big|_{\mathcal{C}_t}$ denotes the logits restricted to the current task's classes, and λ_{ce}, λ_{al}, λ_{ds} weight the cross-entropy, alignment, and distillation terms respectively. The distillation term is omitted for $t = 0$. The masked cross-entropy term provides within-task discrimination through an auxiliary head h_θ, while the alignment term attracts each normalized representation toward its globally assigned ETF vertex. Since class separation is already encoded by the simplex geometry, no explicit repulsion between vertices is required. The distillation term further constrains representation changes relative to the previous-task model.

Crucially, $\mathbf{e}_y$ depends only on class identity. Clients occupying different learning states therefore optimize within the same global coordinate system, reducing representation inconsistency caused by temporal misalignment. The auxiliary head is used only during training and for the inversion objective below. At inference, predictions use the ETF classifier $g_\theta(\mathbf{x})_c = \mathbf{e}_c^\top f_\theta(\mathbf{x})$, with the candidate classes restricted according to the Task-IL or Class-IL protocol.

4.2 Data-Free Replay for Old-Class Anchors

Equation 11 directly anchors only classes represented in the current batch. For previously learned classes, current-task distillation alone may not sufficiently preserve their feature-to-anchor mappings. TARDIS therefore provides a data-free variant that synthesizes old-class inputs without retaining raw historical samples.

Inverting the frozen model. After task τ_t, client i trains a lightweight conditional generator $G_\psi(\mathbf{z}, y)$ for $y \in \mathcal{C}_{0:t}$ against its frozen local model. Following DeepInversion (Yin et al., 2020), we optimize a batch of B_G samples $\mathbf{x}_j = G_\psi(\mathbf{z}_j, y_j)$, with labels y_j drawn uniformly from $\mathcal{C}_{0:t}$ and noise latents $\mathbf{z}_j$ sampled from the generator's prior, to match their target class and the teacher's stored BatchNorm statistics; an entropy term (Chen et al., 2019) encourages diversity across the batch:

$$\mathcal{L}_G \;=\; \frac{1}{B_G}\sum_{j=1}^{B_G} \ell_{\mathrm{CE}}\big(h_\theta(\mathbf{x}_j),\, y_j\big) \;+\; \beta \sum_{\ell} \Big(\big\|\boldsymbol{\mu}_\ell - \tilde{\boldsymbol{\mu}}_\ell\big\|^2 + \big\|\boldsymbol{\sigma}^2_\ell - \tilde{\boldsymbol{\sigma}}^2_\ell\big\|^2 \Big) \;-\; \eta\, H\big(\bar{p}\big), \tag{12}$$

where $\boldsymbol{\mu}_\ell, \boldsymbol{\sigma}^2_\ell$ are the batch statistics of $\{\mathbf{x}_j\}$ at layer ℓ, $\tilde{\boldsymbol{\mu}}_\ell, \tilde{\boldsymbol{\sigma}}^2_\ell$ the teacher's corresponding running statistics, and $\bar{p} = \frac{1}{B_G}\sum_{j=1}^{B_G} \mathrm{softmax}(h_\theta(\mathbf{x}_j))$ the batch-averaged prediction. No real samples are used; the generator is retrained at each task boundary and remains entirely client-local.

Data-free replay. During subsequent optimization, labels are sampled uniformly from completed classes and synthetic inputs $\mathbf{x}^f = G_\psi(\mathbf{z}, y^f)$ are generated on demand. They participate only in the alignment and distillation terms of equation 11, preserving old feature-to-anchor relations without affecting the current-task auxiliary classifier.

Decoupling replay coverage from optimization balance. Synthetic samples contribute only to the alignment and distillation losses, together with the real current-task samples $\mathbf{x}^r \in \mathcal{B}$. Let $B_r = |\mathcal{B}|$ and B_f denote the real and synthetic batch sizes, and let $\mathcal{L}_q(\mathbf{x}^r)$ and $\mathcal{L}_q(\mathbf{x}^f)$ denote the corresponding batch-averaged losses for $q \in \{\mathrm{al}, \mathrm{ds}\}$. Direct concatenation couples replay coverage with optimization weight: increasing B_f to cover more old classes also increases the synthetic contribution to the loss. We instead combine the two contributions separately,

$$\tilde{\mathcal{L}}_q = \frac{\mathcal{L}_q(\mathbf{x}^r) + \omega_q \mathcal{L}_q(\mathbf{x}^f)}{1+\omega_q}, \qquad q \in \{\mathrm{al}, \mathrm{ds}\}, \tag{13}$$

where, in Eq. 11, $\lambda_{\mathrm{al}}\tilde{\mathcal{L}}_{\mathrm{al}}$ and $\lambda_{\mathrm{ds}}\tilde{\mathcal{L}}_{\mathrm{ds}}$ replace the original alignment and distillation contributions, respectively. The synthetic batch size B_f therefore controls replay coverage, while ω_{al} and ω_{ds} independently control its optimization strength; setting $\omega_q = B_f/B_r$ recovers the batch-size-weighted average obtained by direct concatenation. We denote the resulting data-free variant TARDIS$_{\mathrm{DF}}$ and evaluate it alongside buffer-based TARDIS.

5 Experiments

5.1 Experimental Setup

Baselines. We compare TARDIS with two standard FL methods, FedAvg (McMahan et al., 2017) and FedProx (Li et al., 2020c), and six representative FCL methods: FedWeIT (Yoon et al., 2021), TARGET (Zhang et al., 2023), MFCL (Babakniya et al., 2023), AF-FCL (Wuerkaixi et al., 2024), FedCBDR (Qi et al., 2026b), and FEAT$_{\mathrm{F}}$ (Qi et al., 2026a). All baselines follow the same temporally heterogeneous training schedule, with task-boundary operations triggered according to each client's local progression.

Datasets. We evaluate on Split-CIFAR10 (5×2 classes) and Split-CIFAR100 (10×10) (Krizhevsky et al., 2009), Split-EMNIST (9×5 classes from the 47-class `balanced` split) (Cohen et al., 2017), and Split-TinyImageNet (10×20) (Le et al., 2015). Classes are first divided into disjoint tasks in index order. Within each task, samples from every class are then distributed across clients independently according to $\mathrm{Dir}(\alpha)$ with $\alpha = 0.5$, producing heterogeneous client-specific task volumes and hence different temporal progression rates. We report Task-IL and Class-IL accuracy (equations 6–7) and average forgetting.

Implementation details. TARDIS employs composite-weighted aggregation instead of simple average, details provided in Appendix A.2.1. We use the ResNet-18 model as the backbone. Local

Table 1: Main results across four benchmarks under the **Task-IL** and **Class-IL** settings, reported as mean ± std over three random seeds. Each cell reports accuracy (Acc, %, top) and **forgetting** (F, %, bottom); higher Acc and lower F indicate better performance. **Bold** and underline indicate the best and second-best accuracy per column.

Method	Split-CIFAR-10		Split-CIFAR-100		Split-EMNIST		Split-TinyImageNet	
	Task-IL	Class-IL	Task-IL	Class-IL	Task-IL	Class-IL	Task-IL	Class-IL
FedAvg (AISTATS, 2017)	84.40±2.04 *8.75±1.07*	31.51±2.52 *24.57±1.71*	64.19±0.23 *21.88±0.20*	20.65±0.19 *37.40±0.59*	85.77±0.92 *11.97±0.90*	28.54±0.81 *51.91±0.57*	27.35±0.82 *39.97±0.73*	9.45±0.37 *40.65±0.73*
FedProx (MLSys, 2020)	84.95±0.91 *9.66±0.47*	29.34±1.01 *32.18±2.53*	65.27±0.26 *20.84±0.36*	20.84±0.38 *36.98±0.22*	83.91±0.67 *13.99±0.58*	27.49±1.33 *53.11±1.60*	26.69±1.47 *38.16±2.22*	8.49±1.12 *39.92±2.52*
FedWeIT (ICML, 2021)	52.87±1.13 *12.25±1.22*	45.85±0.84 *10.46±0.83*	25.06±0.35 *16.49±0.41*	22.56±1.32 *16.64±0.96*	32.25±1.34 *20.55±1.93*	29.44±1.24 *20.46±1.14*	8.87±0.81 *4.33±0.36*	6.77±0.49 *4.06±0.57*
TARGET (ICCV, 2023)	85.01±0.09 *7.70±0.15*	24.75±1.15 *19.97±1.41*	48.37±1.50 *28.01±1.82*	6.39±1.14 *22.69±0.48*	80.50±1.76 *16.86±2.07*	22.03±2.69 *31.27±1.71*	16.11±0.36 *27.39±2.23*	4.12±0.51 *10.95±0.52*
MFCL (NeurIPS, 2023)	82.74±0.39 *10.97±0.63*	29.72±1.94 *43.79±0.61*	71.97±0.92 *11.64±0.86*	26.20±0.44 *30.79±0.74*	89.58±1.10 *8.86±1.03*	48.13±3.68 *45.61±4.11*	40.67±0.88 *23.78±1.67*	12.90±1.23 *42.71±2.40*
AF-FCL (ICLR, 2024)	78.55±1.84 *12.69±1.27*	19.25±0.16 *45.50±4.96*	44.78±1.62 *37.36±1.54*	9.99±0.26 *70.11±0.27*	89.79±2.05 *8.33±2.20*	20.80±1.32 *74.51±0.95*	19.56±0.43 *41.84±0.67*	6.71±0.26 *54.63±0.32*
FedCBDR (NeurIPS, 2025)	91.87±0.52 *3.08±0.64*	25.51±3.02 *84.90±2.61*	71.55±1.56 *16.46±1.84*	13.48±0.38 *79.35±0.55*	96.35±0.44 *1.93±0.22*	45.16±0.15 *59.65±0.52*	33.33±0.47 *38.52±0.96*	9.56±0.14 *64.76±0.44*
FEAT$_F$ (CVPR, 2026)	91.08±0.68 *5.12±0.96*	28.46±4.49 *73.57±3.20*	40.86±0.47 *29.14±1.21*	10.62±0.18 *51.70±1.23*	90.69±0.89 *8.08±0.98*	45.45±1.85 *52.30±1.38*	20.60±0.61 *42.19±0.51*	7.52±0.24 *51.22±0.65*
TARDIS (DF)	92.54±0.22 *1.20±0.71*	49.81±0.32 *8.70±1.40*	**78.02**±0.10 *5.90±0.12*	33.44±0.18 *22.44±0.84*	97.75±0.03 *0.38±0.01*	72.30±1.20 *10.48±1.12*	**49.51**±0.63 *13.21±0.94*	**19.60**±0.57 *17.80±0.57*
TARDIS	**93.79**±0.28 *0.46±0.20*	**65.61**±0.56 *13.52±0.87*	76.06±0.16 *4.00±0.30*	**38.83**±0.54 *10.91±0.48*	**97.96**±0.01 *0.32±0.05*	**85.62**±0.63 *5.61±1.22*	42.68±0.70 *6.41±0.55*	18.76±1.10 *5.13±1.24*

epochs K and batch size B are fixed per dataset across all methods, full configurations provided in Appendix A.2.2. Exemplar-based methods (FedCBDR, FEAT$_F$, and buffer-based TARDIS) use a memory budget of 200 samples per client, while generative methods draw 128 synthetic samples per optimization step. Experiments are conducted with Flower (Beutel et al., 2020) on four NVIDIA H100 GPUs, using Ray-based simulation backend to manage 10 virtual clients communicating over gRPC. The horizon is set by the slowest client, and results are averaged over three independent runs.

5.2 Main Result

Overall performance. Table 1 shows that TARDIS performs consistently well across all four benchmarks and both evaluation protocols. Buffer-based TARDIS achieves the best accuracy on five of eight settings and ranks second on the remaining three, while TARDIS$_{DF}$ attains the best results on Split-CIFAR100 Task-IL and both Split-TinyImageNet settings. The gains are particularly pronounced under Class-IL, where TARDIS improves over the strongest baseline by 19.76%, 12.63%, 37.49%, and 5.86% on Split-CIFAR10, Split-CIFAR100, Split-EMNIST, and Split-TinyImageNet, respectively. These results indicate that the shared geometric anchoring is especially effective when knowledge acquired at different learning states must remain globally comparable.

Knowledge retention under temporal heterogeneity. TARDIS also maintains low forgetting across benchmarks, particularly on Split-CIFAR100 and Split-EMNIST, where it improves both accuracy and retention over existing FCL approaches. Notably, several baselines achieve strong Task-IL accuracy but degrade substantially under Class-IL, suggesting that preserving task-specific knowledge alone is insufficient when temporally misaligned clients contribute to a shared classifier. In contrast, TARDIS maintains a globally coordinated representation space across learning states, yielding substantially stronger Class-IL performance.

Data-free replay. TARDIS$_{DF}$ remains competitive without storing historical samples and achieves the strongest results on the more challenging Split-TinyImageNet benchmark. Although buffer-based TARDIS generally provides better retention, the data-free variant preserves much of the benefit of temporal anchoring, demonstrating that the proposed geometric stabilization is not dependent on access to real replay data.We further analyze the communication and runtime overhead of TARDIS$_{DF}$ in Appendix 6.

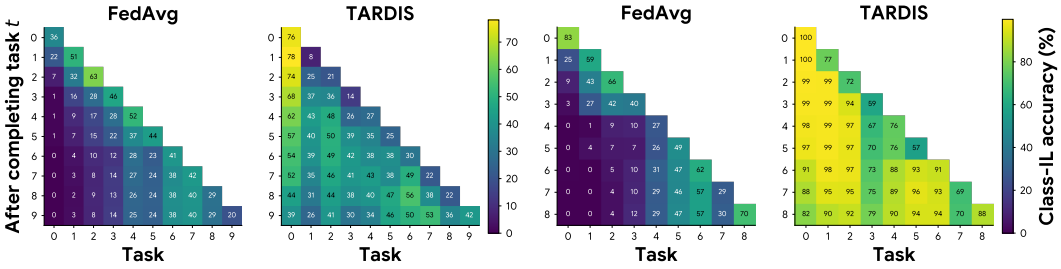
5.3 Ablation Study

Robustness to Temporal Heterogeneity. Table 2 evaluates TARDIS across increasing levels of temporal heterogeneity. Performance remains consistently stable as α decreases: accuracy varies by at most 2.95 points on CIFAR-100 and less than one point on EMNIST. The consistent Class-IL performance under $\alpha = 0.3$ further demonstrates robustness to substantial variation in client progression.

Table 2: **Ablation for temporal heterogeneity level.** Smaller α induces stronger workload imbalance and temporal misalignment.

Dataset	Protocol	Dirichlet α		
		0.3	0.5	0.7
CIFAR-100	Task-IL	73.66±0.48	76.06±0.16	76.61±0.61
		4.90±0.23	*4.00±0.30*	*3.23±0.06*
	Class-IL	37.71±0.42	38.83±0.54	38.45±0.31
		10.84±0.23	*10.91±0.48*	*11.92±0.54*
EMNIST	Task-IL	97.80±0.15	97.96±0.01	97.41±0.16
		0.35±0.02	*0.32±0.05*	*0.94±0.07*
	Class-IL	84.84±0.15	85.62±0.63	85.70±0.22
		4.92±0.17	*5.61±1.22*	*9.51±0.40*

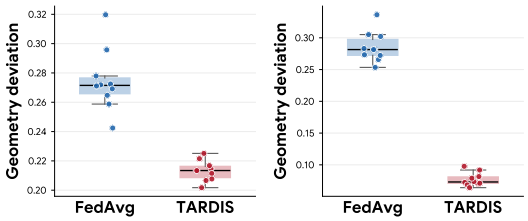
Knowledge retention across tasks. Figure 3 compares TARDIS with FedAvg, which lacks an explicit geometric anchoring mechanism, by tracking task-wise Class-IL accuracy throughout continual progression. FedAvg progressively loses performance on earlier tasks as new knowledge is acquired, whereas TARDIS retains substantially stronger performance across both benchmarks. The contrast becomes increasingly pronounced for older tasks, indicating that the shared ETF geometry provides a persistent reference that stabilizes previously acquired representations over time.



(a) Split-CIFAR100 (b) Split-EMNIST

Figure 3: Task-wise Class-IL accuracy throughout continual progression. Row t reports performance on tasks $0, \ldots, t$ once all clients have completed task t, with cells restricted to $s \leq t$ for matched learning states.

Cross-client geometric consistency. We evaluate whether the shared ETF induces a consistent representation geometry across clients. For each client, geometric deviation is measured as the mean absolute difference between pairwise class-mean cosine similarities and the ideal simplex value $-1/(K-1)$. As shown in Figure 4, TARDIS substantially reduces both the mean deviation and cross-client dispersion on Split-CIFAR100 and Split-EMNIST, indicating consistent alignment across heterogeneous learning states. Figure 5 further decomposes this deviation by task pair.



(a) Split-CIFAR100 (b) Split-EMNIST

Figure 4: Cross-client deviation from simplex ETF geometry. Lower values indicate stronger geometric consistency. Points denote clients and boxes show the interquartile range.

6 Conclusion

In this work, we study temporal heterogeneity in FCL, where heterogeneous task workloads cause clients to progress through tasks at different rates and induce persistent temporal misalignment. We propose TARDIS, which progressively constructs a shared ETF geometry to stabilize representations across evolving client states. Experiments across multiple benchmarks show that TARDIS consistently improves knowledge retention, geometric consistency, and overall performance under heterogeneous progression. Our study targets identifiable task boundaries and disjoint task label spaces, which may not hold in more open-ended continual environments. Extending TARDIS to task-free streams and broader forms of system heterogeneity remains an important direction for future work.

AI USE STATEMENT

In this work, we used generative AI tools to assist with code debugging during the implementation of our method and to suggest additional ablation topics to investigate. We did not use generative AI tools for other required-disclosure tasks, and the remaining required categories were not applicable to this work. Additionally, we used generative AI tools to suggest experimental parameters, identify relevant literature, and refine several subsection titles to improve readability. All experimental parameters were independently evaluated and selected through our own experiments, while the selection, organization, and formatting of the literature were conducted by the authors. All AI-assisted outputs were reviewed by the authors, who take full responsibility for the final content, claims, code, and experimental results presented in this work.

REFERENCES

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

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.

Daniel J Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Javier Fernandez-Marques, Yan Gao, Lorenzo Sani, Kwing Hei Li, Titouan Parcollet, Pedro Porto Buarque de GusmÃÇo, et al. Flower: A friendly federated learning research framework. *arXiv preprint arXiv:2007.14390*, 2020.

Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. Data-free learning of student networks. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3514–3522, 2019.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pp. 2921–2926. IEEE, 2017.

Trung-Anh Dang, Vincent Nguyen, Ngoc-Son Vu, and Christel Vrain. Memory-efficient continual learning with neural collapse contrastive. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385, 2021.

Shuiguang Deng, Hailiang Zhao, Weijia Fang, Jianwei Yin, Schahram Dustdar, and Albert Y Zomaya. Edge intelligence: The confluence of edge computing and artificial intelligence. *IEEE Internet of Things Journal*, 7(8):7457–7469, 2020.

Haizhou Du and Pengfei Li. Decoupling feature entanglement for personalized federated learning via neural collapse. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pp. 615–624, 2025.

Wenke Huang, Mang Ye, and Bo Du. Learn from others and be yourself in heterogeneous federated learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10133–10143. IEEE, 2022.

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

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.

Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

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.

Baoxue Li, Pengyu Song, Chunhui Zhao, and Min Xie. Facing spatiotemporal heterogeneity: A unified federated continual learning framework with self-challenge rehearsal for industrial monitoring tasks. *Knowledge-Based Systems*, 289:111491, 2024.

Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10708–10717. IEEE, 2021a.

Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37(3):50–60, 2020a.

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, 2020b.

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, 2020c.

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

Zexi Li, Xinyi Shang, Rui He, Tao Lin, and Chao Wu. No fear of classifier biases: Neural collapse inspired federated learning with synthetic and fixed classifier. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5296–5306. IEEE, 2023.

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

Yuhang Ma, Zhongle Xie, Jue Wang, Ke Chen, and Lidan Shou. Continual federated learning based on knowledge distillation. In *Ijcai*, pp. 2182–2188, 2022.

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

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Razvan Pascanu, and Hassan Ghasemzadeh. Understanding the role of training regimes in continual learning. *Advances in neural information processing systems*, 33:7308–7320, 2020.

Vardan Papyan, X. Y. Han, and David L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40): 24652–24663, 2020.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural networks*, 113:54–71, 2019.

Can Peng, Yuyuan Liu, Yingyu Yang, Pramit Saha, Qianye Yang, and J Alison Noble. Neural collapse-inspired multi-label federated learning under label-distribution skew. *arXiv preprint arXiv:2509.12544*, 2025.

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

Zhuang Qi, Ying-Peng Tang, Lei Meng, Guoqing Chao, Lei Wu, Han Yu, and Xiangxu Meng. From selection to scheduling: Federated geometry-aware correction makes exemplar replay work better under continual dynamic heterogeneity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17494–17504, June 2026a.

Zhuang Qi, Ying-Peng Tang, Lei Meng, Han Yu, Xiaoxiao Li, and Xiangxu Meng. Class-wise balancing data replay for federated class-incremental learning. *Advances in Neural Information Processing Systems*, 38:145309–145334, 2026b.

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)*, pp. 5533–5542. IEEE, 2017.

Khadija Shaheen, Muhammad Abdullah Hanif, Osman Hasan, and Muhammad Shafique. Continual learning for real-world autonomous systems: Algorithms, challenges and frameworks. *Journal of Intelligent & Robotic Systems*, 105(1):9, 2022.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017.

Andrea Soltoggio, Eseoghene Ben-Iwhiwhu, Vladimir Braverman, Eric Eaton, Benjamin Epstein, Yunhao Ge, Lucy Halperin, Jonathan How, Laurent Itti, Michael A Jacobs, et al. A collective ai via lifelong learning and sharing at the edge. *Nature Machine Intelligence*, 6(3):251–264, 2024.

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

Qiang Wang, Bingyan Liu, and Yawen Li. Traceable federated continual learning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12872–12881. IEEE, 2024.

Xubin Wang, Zhiqing Tang, Jianxiong Guo, Tianhui Meng, Chenhao Wang, Tian Wang, and Weijia Jia. Empowering edge intelligence: A comprehensive survey on on-device ai models. *ACM Computing Surveys*, 57(9):1–39, 2025a.

Zheng Wang, Wanhao Yu, Li Yang, and Sen Lin. Rethinking continual learning with progressive neural collapse. *arXiv preprint arXiv:2505.24254*, 2025b.

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, pp. 50778–50796, 2024.

Zikai Xiao, Zihan Chen, Liyinglan Liu, Yang Feng, Joey Tianyi Zhou, Jian Wu, Wanlu Liu, Howard Yang, and Zuozhu Liu. Fedloge: Joint local and generic federated learning under long-tailed data. In *International Conference on Learning Representations*, volume 2024, pp. 50357–50376, 2024.

Kunlun Xu, Yibo Feng, Jiangmeng Li, Yongsheng Qi, and Jiahuan Zhou. C^2 Prompt: Class-aware client knowledge interaction for federated continual learning. *Advances in Neural Information Processing Systems*, 38:44109–44139, 2026.

Shanshan Yan, Zexi Li, Chao Wu, Meng Pang, Yang Lu, Yan Yan, and Hanzi Wang. You are your own best teacher: Achieving centralized-level performance in federated learning under heterogeneous and long-tailed data. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2750–2759. IEEE, 2025.

Yibo Yang, Shixiang Chen, Xiangtai Li, Liang Xie, Zhouchen Lin, and Dacheng Tao. Inducing neural collapse in imbalanced learning: Do we really need a learnable classifier at the end of deep neural network? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Yibo Yang, Haobo Yuan, Xiangtai Li, Zhouchen Lin, Philip Torr, and Dacheng Tao. Neural collapse inspired feature-classifier alignment for few-shot class-incremental learning. In *International Conference on Learning Representations (ICLR)*, 2023a.

Yibo Yang, Haobo Yuan, Xiangtai Li, Jianlong Wu, Lefei Zhang, Zhouchen Lin, Philip H. S. Torr, Bernard Ghanem, and Dacheng Tao. Neural collapse terminus: A unified solution for class incremental learning and its variants. *arXiv preprint arXiv:2308.01746*, 2023b.

Mang Ye, Xiuwen Fang, Bo Du, Pong C Yuen, and Dacheng Tao. Heterogeneous federated learning: State-of-the-art and research challenges. *ACM Computing Surveys*, 56(3):1–44, 2023.

Xiaoyang Yi, Yang Liu, Binhan Yang, and Jian Zhang. Federated continual learning via orchestrating multi-scale expertise. *Advances in Neural Information Processing Systems*, 38:22785–22814, 2026.

Hongxu Yin, Pavlo Molchanov, Jose M. Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K. Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8715–8724, 2020.

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*, pp. 12073–12086. PMLR, 2021.

Jie Zhang, Chen Chen, Weiming Zhuang, and Lingjuan Lyu. Target: Federated class-continual learning via exemplar-free distillation. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4759–4770. IEEE, 2023.

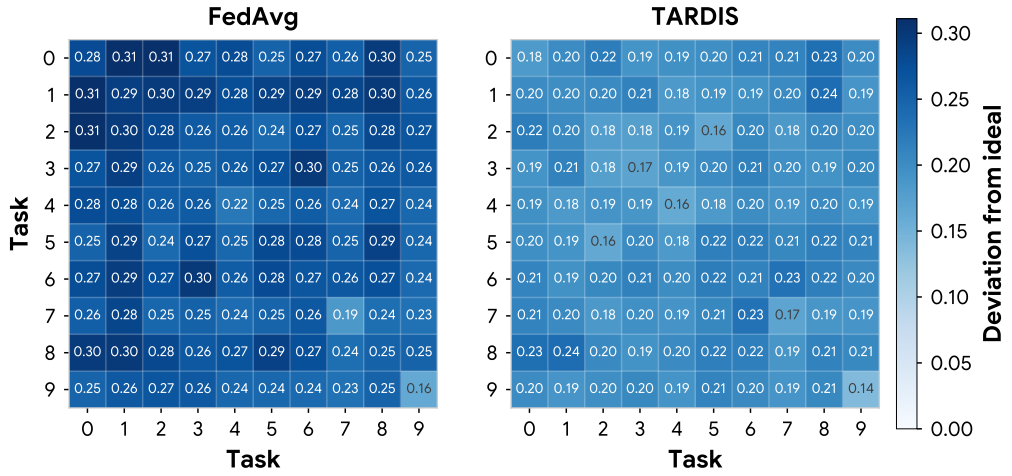
Hangyu Zhu, Jinjin Xu, Shiqing Liu, and Yaochu Jin. Federated learning on non-iid data: A survey. *Neurocomputing*, 465:371–390, 2021.

A Appendix

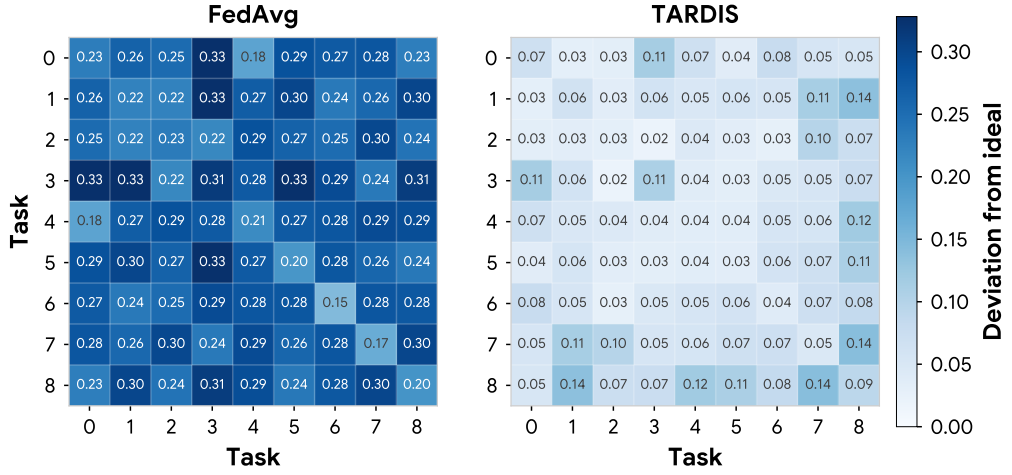
A.1 Additional Ablation Studies

A.1.1 Task-pair geometric consistency.

Figure 5 localizes the aggregate deviation in Figure 4 according to the tasks from which each class pair originates. FedAvg exhibits substantial geometric distortion across both within-task and cross-task pairs, whereas TARDIS remains consistently close to the target ETF geometry throughout the task matrix. This shows that the geometric consistency of TARDIS is maintained across the continual task sequence rather than being concentrated on particular tasks.



(a) Split-CIFAR100



(b) Split-EMNIST

Figure 5: Task-pair decomposition of the deviation from the simplex ETF. Cell (i, j) averages $|\cos(\boldsymbol{\mu}_c, \boldsymbol{\mu}_{c'}) + 1/(K-1)|$ for class pairs from tasks i and j; lower is better. Diagonal cells measure within-task geometry and off-diagonal cells measure cross-task geometry.

A.1.2 Ablation on Composite-Weighted Aggregation

Table 3 evaluates the contribution of the composite-weighted aggregation described in Appendix A.2.1. Replacing it with uniform averaging leads to only modest changes in accuracy on most benchmarks, indicating that the shared ETF geometry remains the primary source of performance gains. Nevertheless, composite weighting consistently reduces forgetting across all datasets

and both evaluation protocols, with larger improvements under the more challenging settings. This suggests that weighting clients by task proximity, local progress, and data support mainly improves the stability of global knowledge aggregation under temporal heterogeneity.

Table 3: Effect of composite-weighted aggregation. *Uniform* replaces $w_i \propto p_i s_i c_i$ with an unweighted mean over contributing clients, leaving the shared ETF and the task-partitioned head unchanged. Each cell reports accuracy (Acc, %, top) and forgetting (F, %, bottom) over three runs.

Protocol	Split-CIFAR-10		Split-CIFAR-100		Split-EMNIST		Split-TinyImageNet	
	Uniform	Composite	Uniform	Composite	Uniform	Composite	Uniform	Composite
Task-IL	92.91±0.15	**93.79**±0.28	75.24±0.37	**76.06**±0.16	97.65±0.02	**97.96**±0.01	38.54±0.40	**42.68**±0.70
	0.99±0.17	*0.46±0.20*	*4.90±0.57*	*4.00±0.30*	*0.52±0.02*	*0.32±0.05*	*14.38±0.10*	*6.41±0.55*
Class-IL	64.66±0.28	**65.61**±0.56	37.85±0.81	**38.83**±0.54	84.98±0.23	**85.62**±0.63	16.85±0.21	**18.76**±1.10
	13.80±0.44	*13.52±0.87*	*13.14±0.60*	*10.91±0.48*	*9.12±0.27*	*5.61±1.22*	*12.28±0.15*	*5.13±1.24*

A.1.3 Ablation on ETF Initialization and Expansion

Table 4 examines both the initialization and expansion strategy of the global ETF. A randomly oriented ETF remains competitive, showing that TARDIS does not critically depend on empirical orientation recovery. However, recovering the base orientation from federated class means generally yields stronger accuracy, particularly under Class-IL and on the more challenging benchmarks. Progressive expansion also performs better overall than pre-allocating all class vertices, supporting the design of adapting the ETF as the knowledge space grows. These results support empirical orientation recovery and progressive expansion as the default design. Although random initialization removes the initial aggregation overhead, Appendix A.1.4 further shows that the empirical orientation itself can be recovered reliably from only a small fraction of clients.

Table 4: ETF initialization and expansion. *Base orientation* is either a random orthogonal basis or estimated from the base task's federated class means (Eq. 10); *frame* is either pre-allocated over all classes before training or progressively extended as each task arrives. The last row is TARDIS's default configuration. Each cell reports accuracy (Acc, %, top) and forgetting (F, %, bottom) over three runs.

Base orientation	Frame	Split-CIFAR-10		Split-CIFAR-100		Split-EMNIST		Split-TinyImageNet	
		Task-IL	Class-IL	Task-IL	Class-IL	Task-IL	Class-IL	Task-IL	Class-IL
Random	Pre-allocated	92.35±0.50	53.20±1.41	74.13±0.77	35.38±0.20	97.19±0.39	85.02±0.75	38.92±0.55	15.36±0.37
		0.44±0.14	*4.18±0.87*	*4.80±0.24*	*13.76±0.42*	*0.45±0.15*	*5.92±0.96*	*14.15±0.45*	*17.50±0.27*
Random	Progressive	93.10±0.34	57.50±0.35	**76.56**±0.35	37.75±0.34	97.72±0.14	85.46±0.37	39.16±0.97	15.96±0.64
		0.44±0.14	*6.21±0.44*	*3.63±0.22*	*12.62±0.26*	*0.32±0.03*	*5.99±0.43*	*12.77±0.67*	*15.91±0.54*
Empirical	Progressive	**93.79**±0.28	**65.61**±0.56	76.06±0.16	**38.83**±0.54	**97.96**±0.01	**85.62**±0.63	**42.68**±0.70	**18.76**±1.10
		0.46±0.20	*13.52±0.87*	*4.00±0.30*	*10.91±0.48*	*0.32±0.05*	*5.61±1.22*	*6.41±0.55*	*5.13±1.24*

A.1.4 Ablation on Client Participation for Base Orientation Recovery

Table 5 studies how many clients are required to recover the empirical base orientation. Performance remains broadly stable across participation levels, and even a single-client estimate is competitive with the default 50% setting on both benchmarks. Greater participation provides no consistent performance gain, suggesting that a useful base orientation can be recovered from relatively few clients. The default 50% therefore provides a practical balance between robustness and initial synchronization overhead.

A.1.5 Communication and Runtime Overhead of Data-Free Replay

Table 6 shows that data-free replay introduces no additional communication cost, since the generator is trained and retained entirely on each client. Its additional computation increases walltime due to generator inversion at task boundaries, representing the main cost of avoiding stored exemplars. In

Table 5: Effect of client participation in base-orientation recovery. The server constructs the base ETF after collecting statistics from the specified fraction of clients; "1 client" requires no cross-client aggregation. Each cell reports accuracy (Acc, %, top) and forgetting (F, %, bottom) over three runs. 50% is the default setting of TARDIS.

Dataset	Protocol	Base-vertex participation				
		1 client	30%	50% (default)	70%	100%
CIFAR-100	Task-IL	75.47±0.41	75.98±0.41	76.06±0.16	75.66±0.24	**76.18**±0.70
		4.48±0.16	*3.79±0.11*	*4.00±0.30*	*3.79±0.34*	*3.75±0.25*
	Class-IL	37.75±0.62	**38.84**±0.34	38.83±0.54	38.11±0.29	38.40±0.25
		10.85±0.22	*10.27±0.32*	*10.91±0.48*	*11.75±1.03*	*12.25±0.32*
EMNIST	Task-IL	97.52±0.15	97.85±0.10	**97.96**±0.01	97.64±0.11	96.28±0.42
		0.81±0.15	*0.39±0.10*	*0.32±0.05*	*0.58±0.20*	*1.32±0.17*
	Class-IL	85.46±0.26	**85.72**±0.69	85.62±0.63	84.82±0.25	73.62±1.67
		6.91±0.44	*6.66±0.69*	*5.61±1.22*	*5.72±0.75*	*3.56±0.59*

return, TARDIS$_{\mathrm{DF}}$ removes the need to continually retain raw historical samples while requiring only a lightweight local generator of 6.24 MiB on average. This preserves the data-locality of FCL and provides a practical alternative for privacy-sensitive settings where persistent storage of past samples is undesirable.

Table 6: Communication and runtime cost of TARDIS$_{\mathrm{DF}}$. Communication is measured per client per round and is unchanged from TARDIS, as the generator remains local and is never transmitted. Runtime overhead arises from local generator inversion at task boundaries.

Dataset	Communication (MB/round)			Walltime		
	Uplink	Downlink	Total	TARDIS	TARDIS$_{\mathrm{DF}}$	Overhead
Split-CIFAR-10	42.66	42.67	85.34	01:02:03	01:21:22	+31.1%
Split-CIFAR-100	42.84	42.95	85.78	02:22:49	02:46:50	+16.8%
Split-EMNIST	42.73	42.78	85.51	04:07:23	04:38:21	+12.5%
Split-TinyImageNet	43.03	43.25	86.28	05:53:17	06:58:54	+18.6%

A.2 Additonal Implementation Details

A.2.1 Composite-Weighted Aggregation under Temporal Heterogeneity

In Section 3.1, observation I shows that, under temporal heterogeneity, global aggregation may combine updates optimized for different task states. TARDIS addresses this mismatch through *composite-weighted aggregation*, which adjusts each client's contribution according to task proximity, local progress, and data support.

At round r, the server aggregates participating client models as

$$\bar{\theta} = \sum_i w_i \theta_i, \quad w_i = \frac{p_i s_i c_i}{\sum_j p_j s_j c_j}, \tag{14}$$

where p_i, s_i, and c_i denote the data, task-proximity, and completion weights, respectively.

Task proximity. Let $\tau_{\mathrm{maj}}(r) = \min(\arg\max_t |i : \phi_i(r) = t|)$denote the prevailing task at round r, with ties resolved toward the earlier task. We set $s_i = \gamma^{|\phi_i(r) - \tau_{\mathrm{maj}}(r)|}$, where $\gamma \in (0, 1]$, such that clients farther from the prevailing task receive progressively smaller weights. The tie-breaking rule prevents aggregation from advancing prematurely when comparable client groups remain at different task states.

Local progress. The completion ratio of client i on its current task is $\rho_i = \big(r - \sum_{s<\phi_i(r)} R_{i,s}\big)/R_{i,\phi_i(r)}$. We define $c_i = \varepsilon + (1-\varepsilon)\rho_i$, with $\varepsilon \in (0, 1)$, assigning greater in-

fluence to updates obtained later in local task progression while retaining a minimum contribution from newly entered clients.

Data support. Following standard federated aggregation, we set $p_i = n_{i,\phi_i(r)}$, weighting each update by the amount of local data supporting its current task. Together, the three factors balance statistical support, optimization maturity, and temporal alignment when aggregating clients at heterogeneous learning states.

A.2.2 Examples of Local Client Task Schedule under Temporal Heterogeneity

Tables 7–10 illustrate the client-specific task schedules induced by temporal heterogeneity across the four benchmarks. Under the same Dirichlet partitioning protocol used in the main experiments, variations in local task volumes translate into heterogeneous round budgets $R_{i,t}$, causing clients to progress through tasks at different rates. The resulting schedules exhibit substantial variation both across clients and across tasks, while the overall training horizon is determined by the slowest client, $R = \max_i \sum_t R_{i,t}$.

Table 7: Per-client round budgets on Split-CIFAR10 (5 tasks $\times$ 2 classes, N=10 clients, heterogeneity Dir(α=0.5), K=5, B=128). Each entry is the number of rounds client i needs for task τ_t. The schedule horizon is the largest client total, $R = 275$ rounds.

	Task					
Client	τ_0	τ_1	τ_2	τ_3	τ_4	**Total**
C00	39	76	5	7	25	152
C01	68	34	10	44	39	195
C02	12	19	22	64	8	125
C03	25	29	40	1	49	144
C04	18	39	86	118	14	275
C05	117	61	57	20	8	263
C06	30	27	40	34	52	183
C07	17	19	69	26	35	166
C08	13	51	41	22	81	208
C09	57	39	24	58	86	264
max	117	76	86	118	86	**275**

Table 8: Per-client round budgets on Split-CIFAR100 (10 tasks $\times$ 10 classes, N=10 clients, heterogeneity Dir(α=0.5), K=3, B=32). Each entry is the number of rounds client i needs for task τ_t. The schedule horizon is the largest client total, $R = 570$ rounds.

	Task										
Client	τ_0	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	τ_7	τ_8	τ_9	**Total**
C00	49	46	37	63	57	43	22	24	38	51	430
C01	91	28	57	59	39	35	44	59	28	47	487
C02	26	46	58	66	39	75	39	40	47	49	485
C03	71	63	18	38	30	39	51	58	34	65	467
C04	36	86	22	73	63	32	50	94	67	47	570
C05	48	69	51	20	37	55	52	60	31	36	459
C06	39	45	67	56	51	57	73	28	31	56	503
C07	43	41	55	40	57	60	66	40	26	22	450
C08	35	36	77	29	65	45	46	48	72	45	498
C09	36	13	33	30	35	33	32	24	99	56	391
max	91	86	77	73	65	75	73	94	99	65	**570**

Table 9: Per-client round budgets on Split-EMNIST (9 tasks $\times$ 5 classes, N=10 clients, heterogeneity $\mathrm{Dir}(\alpha$=0.5), K=5, B=128). Each entry is the number of rounds client i needs for task τ_t. The schedule horizon is the largest client total, $R = 525$ rounds.

	Task									
Client	τ_0	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	τ_7	τ_8	**Total**
C00	21	14	27	24	57	19	79	57	68	366
C01	69	86	27	30	21	85	22	52	18	410
C02	33	38	13	48	21	16	58	83	49	359
C03	52	28	24	77	51	45	42	63	49	431
C04	103	98	32	46	74	30	27	33	19	462
C05	27	7	47	68	82	74	62	39	48	454
C06	12	53	46	77	57	41	24	9	120	439
C07	35	43	107	60	40	82	80	56	22	525
C08	41	76	99	20	41	69	20	32	38	436
C09	82	32	52	24	29	14	58	51	42	384
max	103	98	107	77	82	85	80	83	120	**525**

Table 10: Per-client round budgets on Split-TinyImageNet (10 tasks $\times$ 20 classes, N=10 clients, heterogeneity $\mathrm{Dir}(\alpha$=0.5), K=4, B=64). Each entry is the number of rounds client i needs for task τ_t. The schedule horizon is the largest client total, $R = 695$ rounds.

	Task										
Client	τ_0	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	τ_7	τ_8	τ_9	**Total**
C00	63	66	67	31	59	56	54	97	50	42	585
C01	80	77	49	68	50	57	78	97	60	79	695
C02	48	82	76	52	64	65	68	78	61	62	656
C03	89	37	46	72	65	69	60	69	94	85	686
C04	81	63	63	96	76	66	37	37	42	67	628
C05	78	47	62	74	44	46	60	54	61	35	561
C06	56	81	72	67	58	64	55	46	80	72	651
C07	56	63	78	70	32	53	53	37	68	77	587
C08	47	70	73	62	78	77	83	47	56	57	650
C09	33	42	45	37	103	76	83	67	58	55	599
max	89	82	78	96	103	77	83	97	94	85	**695**