DART: DISTRIBUTION-GUIDED ADAPTIVE ROUTING FOR FEDERATED CONTINUAL LEARNING OF GENERATIVE VISION-LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

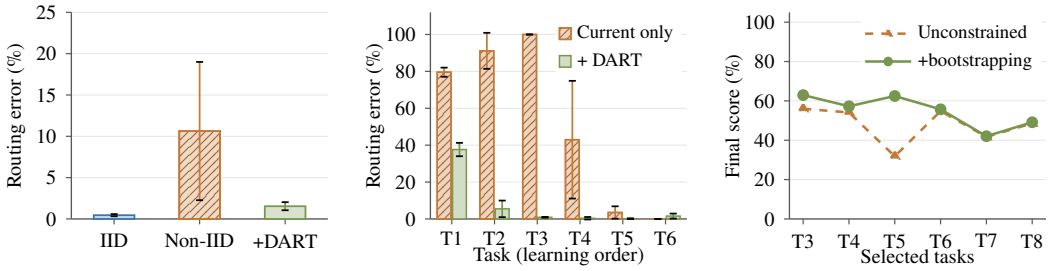
Federated continual learning (FCIL) enables vision-language models (VLMs) to continually learn from decentralized, evolving data while retaining previously learned knowledge. Although mixture-of-experts (MoE) adaptation supports task specialization, client heterogeneity and sequential updates can bias routing toward dominant or recent tasks, while updates to the shared backbone can degrade historical experts even when they are correctly selected. We propose DART, a Distribution-guided Adaptive RouTing framework that jointly addresses routing bias and backbone-induced forgetting. Clients summarize expert-specific routing inputs in a fixed reference space, while the server maintains per-expert historical distributions and integrates them with current client statistics. Balanced synthetic features sampled from these distributions calibrate a global router without replaying historical raw image and text examples. Meanwhile, historical experts are frozen and backbone updates are constrained to a bootstrapped residual subspace, reducing interference with previously learned representations. Extensive experiments show that DART consistently outperforms state-of-the-art baselines, with performance improvements of up to 7.18%.

1 INTRODUCTION

Vision-language models (VLMs) are increasingly adapted to a broad range of generative visual reasoning tasks, including image description, visual question answering, and diagram understanding (Liu et al., 2023; 2024; Chen et al., 2024). In practical deployments, however, the data required to support these capabilities are often distributed across multiple clients rather than centralized in a single repository. At the same time, the learning environment is inherently non-stationary, as clients continually encounter new tasks over time and the global model must acquire new capabilities while retaining previously learned knowledge. Federated continual learning provides a natural framework for this setting by enabling collaborative learning from distributed task streams while mitigating catastrophic forgetting (Yoon et al., 2021; Qi et al., 2023; Zhang et al., 2023). Recent work further extends this paradigm to continual instruction tuning of VLMs, allowing multimodal capabilities to be acquired without centralizing raw training data (Guo et al., 2025). This setting therefore poses a central challenge of continually learning from heterogeneous and evolving client streams while preserving, transferring, and effectively reusing knowledge accumulated across clients and tasks.

A natural response is to structure adaptation around specialized vision experts that retain and reuse task-specific knowledge. DISCO (Guo et al., 2025) stores task knowledge in LoRA subspaces within the language backbone and selectively activates their outputs, while FedDuet (Guo et al., 2026) combines shared semantic prompts with personalized adapters for federated VLM adaptation. Recent studies further identify the vision projectors as a critical interface for preserving multimodal alignment in generative VLMs (Jin et al., 2025; Zhang et al., 2026). However, preserving historical vision projector parameters alone does not guarantee that their learned mappings remain effective over time. As new tasks arrive under uneven client data coverage, the router must continue to reliably distinguish among retained experts. At the same time, even correctly routed historical projector outputs are subsequently processed by shared language backbone parameters that may drift during later adaptation. These two effects create a coupled challenge in maintaining both reliable expert selection and the functional compatibility of historical experts with an evolving backbone.

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.



(a) Federated Routing (b) Temporal Routing (c) Bootstrapping Adaptation

Figure 1: (a) Routing error under increasing client heterogeneity with fixed experts and routing features and after our DART calibration. (b) Routing errors before and after DART calibration following six sequential tasks. (c) Final task scores under unconstrained and bootstrapped updates to each local shared language backbone within DART. Details are provided in Appendix B.5.

These challenges are borne out by our empirical findings shown in Figure 1, which reveal two complementary failure modes in federated continual expert reuse. First, Fig. 1a and 1b show that routing reliability deteriorates under both client heterogeneity and successive task updates. Second, also as shown in Fig. 1c, even after the effective DART router calibration for a reliable routing, unconstrained updates to local language backbones substantially degrade performance on historical tasks compared to the bootstrapped constrained adaptation in DART. Together, these observations show that effective continual expert reuse requires both accurate access to historical experts and adaptive downstream computations through which their outputs are generated. Since retaining or transmitting raw historical samples is undesirable in federated settings, this raises a central question: *how can a local federated continual VLM model maintain reliable routing while being adapted to historical computations without storing or transmitting raw examples at the global central server?*

To address these challenges, we propose DART, which couples global router calibration for reliable routing of local projector experts with bootstrapped residual adaptation of the shared generative language backbone. To mitigate routing drift across heterogeneous clients and successive tasks, clients provide their Gaussian summaries of routing inputs for server calibration. The server then fuses current summaries by expert identity, retains historical distributions separately, and samples synthetic features evenly across these distributions to calibrate the global router without replaying raw examples. As local historical projector experts remain frozen while shared parameters continue to evolve, we constrain subsequent LoRA updates to a bootstrapped residual space to limit interference with historical representations. Together, the mechanism that supports reliable access to retained experts and preservation of their associated historical computations can be characterized by our theoretical analysis. Experiments on two FCIL benchmarks demonstrate the effectiveness of DART, with gains of up to 7.18 points over SOTA methods.

Our main contributions are summarized as follows.

- Globally, we propose a distribution-guided calibration mechanism for vision-projector routing in FCIL. By continuously integrating current and historical seen-task distributions, the server calibrates the router without local router optimization, reducing Non-IID client bias while preserving reliable access to retained experts without replaying historical data.

- Locally, we introduce shared LLM bootstrapping to prevent the continually updated language backbone from becoming incompatible with retained projector experts. This design also reduces the per-round communication of LLM adaptation from full LoRA factors to a single trainable factor.

- We provide a theoretical analysis that decomposes and bounds historical computation drift. The analysis and empirical diagnostics show how global calibration and LLM bootstrapping jointly constrain routing and backbone-induced deviations.

2 Related Work

Federated continual learning and expert routing. Recent federated continual learning methods increasingly organize knowledge through specialized modules and adaptive expert selection. DISCO allocates task-specific LoRA subspaces and activates them through subspace-selective inference, while FedDuet coordinates server-side semantic prompts with client-specific adapters for continual federated VLM adaptation (Guo et al., 2025; 2026). More broadly, FedFMX performs Fisher-guided expert subset selection for federated class-incremental learning, and FCE maintains a historical expert bank with feature-based matching for federated domain-incremental perception (Yuan et al., 2026; Liao et al., 2026). These methods demonstrate the value of expert specialization under heterogeneous federated data, but DART addresses a different routing constraint: the server does not retain historical raw examples, while current routing evidence is fragmented across participating clients. DART therefore formulates routing as a summary-based global calibration problem, fusing client-provided statistics in a fixed projector-routing space and retaining one historical distribution per projector expert for synthetic calibration across clients and tasks.

Continual VLM routing and expert stabilization. Expert routing has also been studied extensively in centralized continual VLM adaptation. MVP and PMA introduce specialized visual projectors to preserve task-dependent cross-modal alignment (Jin et al., 2025; Zhang et al., 2026), while SOCiaL models task distributions with Gaussian mixtures to generate synthetic replay for an adaptive router (Xu et al., 2026). Recent methods further identify routing and expert drift directly: SAME stabilizes routing through orthogonal subspaces and regulates expert updates using historical input statistics, whereas PASs-MoE calibrates routing and stabilizes LoRA directions through pathway activation subspaces (Xie et al., 2026; Hou et al., 2026). DART instead targets recovery of historical task-end computations under federated data constraints. It couples global calibration of the projector router with shared LLM bootstrapping to preserve expert access and downstream compatibility.

3 Problem Setup and Preliminary

Problem setting. We consider federated continual instruction tuning (FCIT) with clients $c \in \{1, \dots, C\}$ and task stages $t \in \{1, \dots, T\}$. At stage t, each participating client c receives a private local dataset $\mathcal{D}_{c,t} = \{(x, q, y)\}$ of images, prompts, and target responses, where data distributions may be heterogeneous across clients. Clients perform local adaptation and exchange model updates and statistical summaries with a central server for global aggregation. As tasks arrive sequentially, historical raw client data are not retained for subsequent training. The objective is to continually acquire knowledge from distributed evolving tasks while preserving performance on previously learned ones, without access to task identities at testing stage.

Projector experts and routing. In our LLaVA realization, projector experts $P_1, \dots, P_t$ connect a frozen vision encoder to a shared decoder. Let $v = V_0(x)$ denote visual patch tokens and $E_0(q)$ the frozen token embeddings of the formatted prompt, excluding the response. The routing descriptor is $h = [\mathrm{Mean}(v); \mathrm{Mean}(E_0(q))]$, where Mean averages valid tokens and $[\,;\,]$ denotes concatenation. A linear router with parameters $\psi_t = (R_t, b_t)$ produces

$$\pi_t(h) = \mathrm{softmax}(R_t h + b_t), \qquad Z_t = \sum_{s=1}^{t} \pi_{t,s}(h) P_s(v). \tag{1}$$

The decoder processes Z_t together with prompt embeddings. Because h is computed from frozen visual and textual features, its coordinates remain fixed throughout continual adaptation.

Accumulated LoRA updates. For a pretrained layer with weight $W_0 \in \mathbb{R}^{d_o \times d_i}$, LoRA (Hu et al., 2022) represents the task-t update as $B_t A_t$, where $B_t \in \mathbb{R}^{d_o \times r}$ and $A_t \in \mathbb{R}^{r \times d_i}$. Sequential adaptation accumulates these low-rank increments in the shared weight. Absorbing the LoRA scale into B_t, the task-end weight is $W_t = W_{t-1} + B_t A_t = W_0 + \sum_{s=1}^{t} B_s A_s$.

4 Theoretical Motivation

Since a task is typically best represented by the model immediately after it has been learned, our goal is to preserve this task-end computation during subsequent continual adaptation. For a historical task

$i < t$, we take the task-i checkpoint as the reference state and ask whether the current model can reproduce its historical computation. In our setting, this computation is determined by two parts: the task-specific projector expert P_i, which provides the multimodal interface, and the shared LLM parameters, which process the resulting representation. After subsequent tasks, either the selected projector or the shared LLM computation may deviate from this historical reference.

Let $\mathcal{D}_j(z, q)$ denote the shared LLM's next-token logits after task j, given projector representation z, prompt q, and the same response prefix. At task t, the router produces $Z_t = \sum_{s=1}^{t} \pi_{t,s}(h) P_s(v)$. We measure historical computation drift as $\Delta_{t,i} = \left(\mathbb{E}_i \|\mathcal{D}_t(Z_t, q) - \mathcal{D}_i(P_i(v), q)\|_2^2\right)^{1/2}$, where the reference is the actual task-i model rather than a jointly trained or optimal model.

Adding and subtracting $\mathcal{D}_t(P_i(v), q)$ separates this drift into two sources. By Minkowski's inequality,

$$\Delta_{t,i} \leq \left(\mathbb{E}_i \left\|\mathcal{D}_t(Z_t, q) - \mathcal{D}_t(P_i(v), q)\right\|_2^2\right)^{1/2} + \left(\mathbb{E}_i \left\|\mathcal{D}_t(P_i(v), q) - \mathcal{D}_i(P_i(v), q)\right\|_2^2\right)^{1/2}. \quad (2)$$

The first term measures whether the current router recovers the historical projector interface, whereas the second term measures whether the continually adapted shared LLM remains compatible with that historical projector. This decomposition directly motivates DART, with global distribution-guided router calibration targets historical expert access, while shared LLM bootstrapping controls downstream compatibility. Section 6 formally shows how these two mechanisms constrain the corresponding terms and thereby bound historical computation drift.

5 METHOD

In this section, we introduce DART, whose overall federated continual learning workflow is illustrated in Figure 2(a). We then describe how the server calibrates the router for projector experts in panel (b), followed by the orthogonal update mechanism for the shared backbone in panel (c).

5.1 EXPERT LEARNING AND GLOBAL ROUTING

For task t, the current projector P_t is initialized from the preceding global expert, or from the pre-trained projector when $t = 1$. Client c trains a local copy $P_{c,t}$ together with the backbone factor $B_{c,t}$, using $W_{c,t} = W_{t-1} + B_{c,t}A_t$. All other parameters, including A_t, historical experts, and the router, remain fixed. During local training, all current-task samples are processed directly by the current projector $P_{c,t}$ rather than by the router-weighted expert mixture in Equation 1. Thus, the instruction loss jointly optimizes only $P_{c,t}$ and $B_{c,t}$:

$$\mathcal{L}_{\text{inst}}(\mathcal{B}) = -\frac{1}{N_{\mathcal{B}}} \sum_{a \in \mathcal{B}} \sum_{j \in \mathcal{Y}_a} \log p_{P_{c,t}, B_{c,t}} \left(y_{a,j} \mid x_a, q_a, y_{a,<j}\right), \quad (3)$$

where $\mathcal{Y}_a$ indexes supervised response tokens and $N_{\mathcal{B}} = \sum_{a \in \mathcal{B}} |\mathcal{Y}_a|$ is their total number.

After model aggregation, the server updates only the router, keeping the backbone and expert bank fixed. At inference, task identity is unavailable, and the calibrated router combines projector outputs according to Equation 1 before the decoder generates a response.

5.2 GLOBAL DISTRIBUTIONAL ROUTER CALIBRATION

To learn selection boundaries among retained experts, the server maintains a distributional memory of their routing inputs. Because the reference descriptor h is fixed, current and historical statistics remain in the same coordinates as the model adapts.

Let $\mathcal{C}_{t,m}$ denote the clients participating in round m of task t. Each client computes routing descriptors h from sampled local training records and estimates their mean $\mu_{c,t}$ and a low-rank diagonal covariance $S_{c,t}$. It uploads these statistics together with the expert identity t and its local training-set size $n_{c,t}$ to the server. The expert identity determines which distribution is updated, while $n_{c,t}$ is used to weight client statistics during aggregation. With $N_{t,m} = \sum_{c \in \mathcal{C}_{t,m}} n_{c,t}$ and $\alpha_{c,t,m} = n_{c,t}/N_{t,m}$,

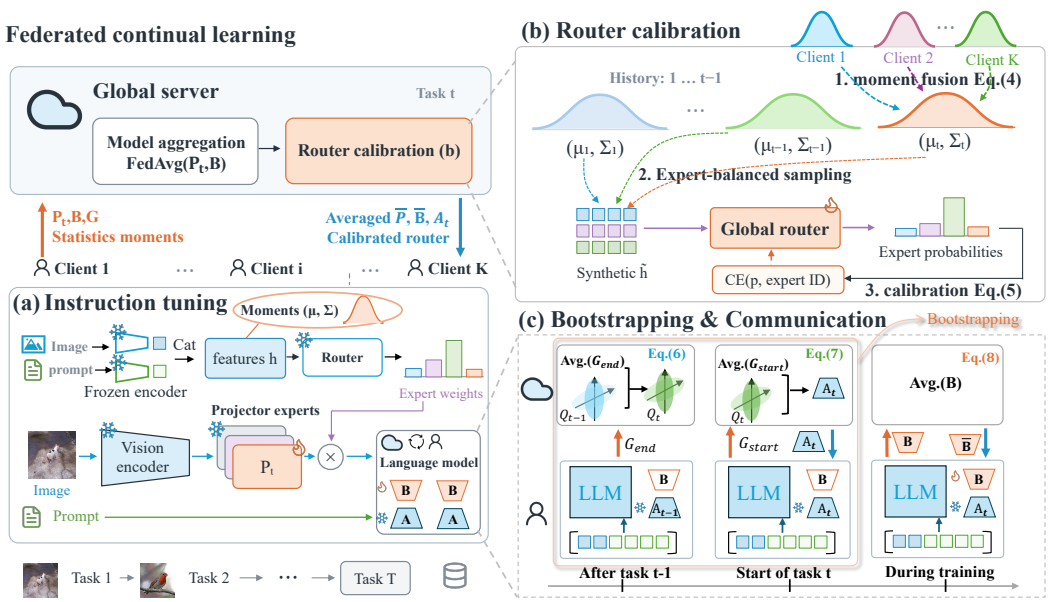


Figure 2: **Framework overview for LLaVA.** (a) **Training overview:** During federated learning, the model maintains a bank of historical projector experts P_i, $i < t$ while sharing a common backbone. Clients upload the current projector, LoRA factor B, and routing statistics; the server first aggregates model updates, then calibrates the global router and broadcasts the updated model and router. (b) **Router calibration:** The server fuses routing input statistics from participating clients while preserving separate distributions for historical experts. It then samples synthetic features evenly from current and historical distributions and calibrates the global router using expert identity supervision with label smoothing. (c) **LLM Bootstrapping and Communication:** The server constructs a shared LLM bootstrap A_t from aggregated gradient information and broadcasts it to participating clients. With A_t fixed, clients optimize and communicate only B during local training, while task-end gradients update the historical state for subsequent bootstrapping.

the server computes

$$\mu_{t,m} = \sum_{c \in \mathcal{C}_{t,m}} \alpha_{c,t,m} \mu_{c,t}, \Sigma_{t,m} = \frac{N_{t,m}}{N_{t,m} - 1} \sum_{c \in \mathcal{C}_{t,m}} \alpha_{c,t,m} \left[S_{c,t} + (\mu_{c,t} - \mu_{t,m})(\mu_{c,t} - \mu_{t,m})^\top \right]. \tag{4}$$

The second term captures distribution differences between clients that would be missed by averaging local covariances alone. Sampling and covariance estimation are detailed in Appendix B.2.

During task t, the server recomputes the current-task global distribution from the clients participating in each communication round, replacing the previous round's estimate. After the final round, this distribution is stored on the server as the historical distribution associated with expert t and remains fixed thereafter. Thus, at any stage t, the server maintains one distribution (μ_s, Σ_s) for each seen expert $s \leq t$. The server calibrates the router by sampling synthetic routing features from these distributions:

$$\mathcal{L}_{\text{cal}}(\psi) = \frac{1}{t} \sum_{s=1}^{t} \mathbb{E}_{\tilde{h} \sim \mathcal{N}(\mu_s, \Sigma_s)} \left[\text{CE}\left(q_s^\epsilon, \pi_\psi(\tilde{h}) \right) \right], \qquad q_s^\epsilon = (1 - \epsilon) e_s + \frac{\epsilon}{t} \mathbf{1}, \tag{5}$$

where $\pi_\psi(h) = \text{softmax}(Rh + b)$ and e_s denotes the label of expert s. Uniform averaging over experts keeps both current and historical distributions represented during calibration.

However, reliable routing alone does not guarantee historical computation preservation. Although historical projectors are retained, subsequent updates to the shared LLM can alter the computation induced by their representations. We therefore introduce shared LLM bootstrapping to control this compatibility mismatch during continual adaptation.

5.3 Bootstrapping Backbone Adaptation

To control the second term in Equation 2, we introduce shared LLM bootstrapping for continual local adaptation. Following the orthogonal-update principle (Luo et al., 2026), the bootstrap constrains subsequent LLM updates to stabilize the computations induced by historical projector representations.

The initial protection basis Q_0 is constructed from pretrained weight directions. For $t \geq 2$, after completing task $t-1$ and merging its backbone update, clients evaluate gradients without taking optimizer steps. The server aggregates these gradients into G_{t-1}^{end}. Let $\Pi_s = I - Q_s Q_s^\top$ denote the projector onto the residual space of Q_s. The protection basis is expanded by

$$Q_{t-1} = \text{orth}\left(\left[Q_{t-2}, \text{SVD}_{k_{t-1}}\left(\Pi_{t-2}(G_{t-1}^{\text{end}})^\top\right)\right]\right). \tag{6}$$

Here $\text{SVD}_j(\cdot)$ returns the top j left singular vectors, and $\text{orth}(\cdot)$ orthonormalizes the concatenated directions. The number of added directions k_{t-1} is determined by the retained gradient energy criterion. Thus, Q_{t-1} combines previously protected directions with additional directions extracted after learning task $t-1$.

At the start of task t, selected clients $c \in \mathcal{S}_t$ estimate local layer gradients $\widehat{G}_{c,t}^{\text{start}}$ from sampled local records at the same inherited model, without taking optimizer steps. The server aggregates these gradient estimates as $G_t^{\text{start}} = \sum_{c\in\mathcal{S}_t} \frac{n_{c,t}}{\sum_{j\in\mathcal{S}_t} n_{j,t}} \widehat{G}_{c,t}^{\text{start}}$, where $\mathcal{S}_t$ is the selected client set and $n_{c,t} = |\mathcal{D}_{c,t}|$ is the local record count. Using the protection basis from previous tasks, the server constructs a shared LoRA input factor:

$$A_t = \text{SVD}_r\left(\Pi_{t-1}(G_t^{\text{start}})^\top\right)^\top. \tag{7}$$

This construction selects the top r singular directions of the projected aggregate gradient. The extracted directions are reorthogonalized in the residual space. The server initializes B_t to zero and broadcasts A_t, which remains fixed throughout the task.

During each communication round, clients optimize the current projector $P_{c,t,m}$ and LoRA output factor $B_{c,t,m}$ using Equation 3, while the shared factor A_t remains fixed. The server aggregates $P_{c,t,m}$ and $B_{c,t,m}$ with FedAvg. Because all clients share the same A_t, averaging B is exactly equivalent to averaging the corresponding LoRA increments:

$$\sum_{c\in\mathcal{C}_{t,m}} \omega_{c,t,m} B_{c,t,m} A_t = \left(\sum_{c\in\mathcal{C}_{t,m}} \omega_{c,t,m} B_{c,t,m}\right) A_t = B_{t,m} A_t. \tag{8}$$

The server then calibrates the router and broadcasts the updated global state.

After the final round, the aggregated LoRA increment is merged into the shared backbone to obtain W_t. The same gradient evaluation is then performed to update the historical protection basis Q_t for bootstrapping the next task, and the completed projector is added to the frozen expert bank. Construction and communication details are provided in Appendix A.4, while Algorithm 1 in Appendix A.5 summarizes the complete workflow.

6 Theoretical Analysis

We now formalize how DART controls the two sources of historical computation drift identified in Section 4. In particular, shared LLM bootstrapping bounds the compatibility term through the historical representation coverage of the protected space.

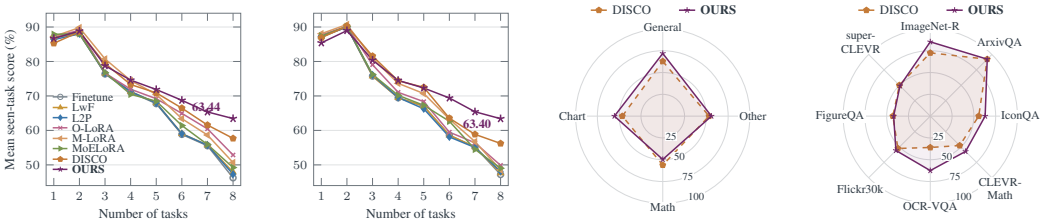
Theorem 1 (Historical computation stability under DART)**.** *Under the setting of Section 4, assume frozen historical projectors, accumulated LoRA updates satisfying $A_{\ell,s}Q_{\ell,s-1} = 0$ for $s > i$, and the finite-moment and computation-graph Lipschitz conditions in Appendix A.2. Then*

$$\left(\mathbb{E}_i \left\|\mathcal{D}_t(Z_t, q) - \mathcal{D}_i(P_i(v), q)\right\|_2^2\right)^{1/2} \leq K_0 R_{t,i} + \sum_{\ell\in\mathcal{L}} K_\ell S_{\ell,t,i} b_{\ell,i}, \tag{9}$$

where $R_{t,i}$ is the routing mismatch, $S_{\ell,t,i} = \sum_{s=i+1}^{t} \|B_{\ell,s}\|_2 \|A_{\ell,s}\|_2$ measures subsequent update magnitude, and $b_{\ell,i}$ bounds the historical layer-input component outside the protected space. The precise definitions and proof are given in Appendices A.1 and A.2. Appendix B.6 empirically examines both effects.

Table 1: Hom-FCIT Last and Avg. †: DISCO-reported results (Guo et al., 2025). Unmarked DART entries use complete tests. Dashes indicate unavailable results.

	Capability-related: four stages						Task-related: eight tasks					
	$\beta = 0.5$		$\beta = 1.0$		$\beta = 5.0$		$\beta = 0.5$		$\beta = 1.0$		$\beta = 5.0$	
Method	Last	Avg	Last	Avg	Last	Avg	Last	Avg	Last	Avg	Last	Avg
Zero-shot†	30.57	–	30.57	–	30.57	–	29.08	–	29.08	–	29.08	–
Individual†	61.75	–	59.62	–	60.27	–	64.21	–	63.85	–	64.07	–
Centralized MTL†	63.83	–	63.83	–	63.83	–	66.60	–	66.60	–	66.60	–
Finetune†	48.68	60.05	49.67	59.01	50.40	58.08	46.24	68.93	47.20	68.79	48.00	69.97
EWC†	49.12	60.13	49.46	59.28	49.89	58.76	47.51	69.04	47.92	69.22	48.15	70.27
LwF†	48.87	60.32	50.02	59.88	50.15	59.06	47.89	69.57	47.62	69.14	48.26	70.30
L2P†	48.22	59.79	49.56	59.34	49.62	59.25	47.30	69.31	48.08	69.65	48.42	70.16
O-LoRA†	51.65	60.19	49.13	57.93	50.29	58.18	52.87	71.54	49.87	70.26	47.76	70.84
M-LoRA†	49.04	60.08	50.39	60.76	50.56	58.06	50.68	71.94	48.53	71.58	48.38	71.21
MoELoRA†	49.69	61.00	50.90	60.18	50.43	59.11	49.23	70.96	49.02	70.65	48.82	71.08
DISCO†	53.73	62.00	55.47	62.07	55.06	60.53	57.69	74.03	56.22	73.03	55.58	72.64
DART (ours)	**60.81**	**63.90**	**59.55**	**64.87**	**59.75**	**64.58**	**63.44**	**74.79**	**63.40**	**74.97**	**63.38**	**75.41**



(a) $\beta = 0.5$ (b) $\beta = 1.0$ (c) Capabilities ($\beta = 1.0$) (d) Task-related ($\beta = 1.0$)

Figure 3: Hom-FCIT performance: eight-task trajectories (a,b), four-stage capability scores (c), and eight-task dataset scores (d).

7 Experiments

7.1 Experimental Setup

Datasets. We evaluate DART on the task-related and capability-related Hom-FCIT benchmarks introduced by DISCO (Guo et al., 2025). The task-related setting sequentially learns eight datasets: ImageNet-R, ArxivQA, IconQA, CLEVR-Math, OCR-VQA, Flickr30k, FigureQA, and super-CLEVR. The capability-related setting groups twelve datasets into four stages corresponding to General, Other, Math, and Chart capabilities. Both settings use the released Dirichlet client partitions, with clients following a common stage sequence under heterogeneous local data. Task identities are unavailable at inference. The complete dataset composition and stage order are provided in Appendix B.1.

Baselines & Evaluation Metrics. We compare DART with Zero-shot, Finetune, EWC, LwF, L2P, O-LoRA, M-LoRA, MoELoRA, and DISCO using published results (Kirkpatrick et al., 2017; Li & Hoiem, 2018; Wang et al., 2022; 2023; Guo et al., 2024; Chen et al., 2024; Guo et al., 2025), and include Individual and Centralized MTL as reference settings. We report Last, the mean score over all datasets after the final stage, and Avg, the average seen-dataset score across stages, using each dataset's native metric with equal dataset weighting. We additionally report routing accuracy and downstream generation performance in mechanism analyses.

Implementation Details. We use LLaVA-1.5-7B (Liu et al., 2024) with rank-32 orthogonal LoRA. Each stage contains ten communication rounds, with five of fifty clients participating per round and one local epoch. Following DISCO, client heterogeneity is controlled by the Dirichlet concentration parameter $\beta \in \{0.5, 1.0, 5.0\}$, where a smaller β indicates stronger non-IID heterogeneity. Each

client estimates routing statistics from at most 128 current-task examples using a rank-32-plus-diagonal covariance, and the server samples 1,024 synthetic descriptors per seen expert for 400 router calibration steps.

7.2 Main Comparisons

Table 1 summarizes the main results on the capability-related and task-related Hom-FCIT benchmarks. At $\beta = 1.0$, DART achieves Last/Avg scores of 59.55/64.87 in the four-stage setting and 63.40/74.97 in the eight-task setting, compared with DISCO's reported 55.47/62.07 and 56.22/73.03, respectively. The improvement is also consistent at the dataset level: in the eight-task benchmark, DART outperforms DISCO on six of eight datasets, while in the four-stage benchmark it achieves higher scores in three of the four reported capability groups. Under stronger client heterogeneity ($\beta = 0.5$), DART obtains eight-task Last/Avg scores of 63.44/74.79, compared with DISCO's reported 57.69/74.03. Figure 3(a,b) shows that DART generally outperforms the comparison methods as tasks accumulate, with a clearer advantage at later stages. Figure 3(d) details the dataset-level performance gaps. Detailed per-dataset and per-capability results are provided in Appendix B.3.

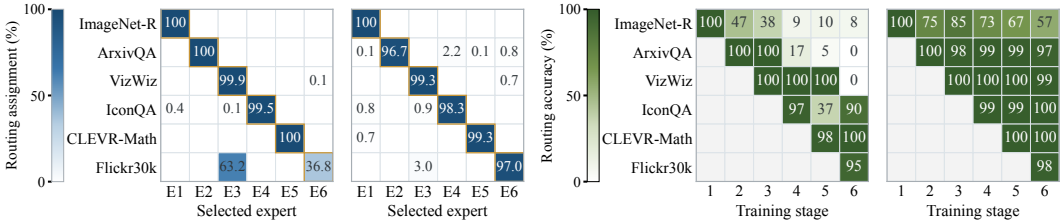
7.3 Ablations and Mechanism Tests

We conduct ablation studies to examine the effectiveness of the two key components of DART: global router calibration and orthogonal adaptation of the shared backbone. For routing, we consider several alternatives to clarify why distribution-guided calibration is needed. We first remove informative routing altogether and assign equal weights to all retained expert outputs (*Uniform Mixing*). To examine the role of historical distributional memory, we calibrate the router using only the current-task distribution while keeping the calibration budget unchanged (*Current Only*). For the shared backbone, we compare unconstrained continual updates with the proposed orthogonal update rule while keeping the routing mechanism and training configuration matched (*Unconstrained Update* and *Orthogonal Update*).

As shown in Table 2, DART reaches 63.40 Last, outperforming Uniform Mixing and Current Only by 6.91 and 8.27 points, respectively. These results show that informative routing and, in particular, historical distributional memory are important for reliable expert reuse. With calibrated routing fixed, orthogonal adaptation improves Last from 58.20 to 63.40, a 5.20-point gain over unconstrained updates, showing that backbone protection provides additional value beyond router calibration. Overall, the ablations support the complementary roles of distribution-guided routing and orthogonal backbone adaptation.

Table 2: Ablations on eight-task Hom-FCIT.

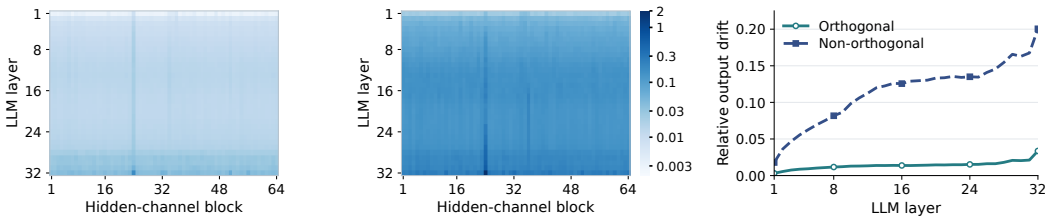
Router strategy	LLM update	Last
Uniform mixing	Orthogonal	56.49
Current-only calibration	Orthogonal	55.13
DART calibration	Unconstrained	58.20
No calibration	Orthogonal	46.57
DART calibration	**Orthogonal**	**63.40**



(a) Non-IID Acc.: 89.37% (b) DART Acc.: 98.46% (c) Continual No calibration (d) Continual With calibration

Figure 4: Router-only diagnostics on the same six tasks, gray cells denote unseen tasks.

Router calibration analysis. Figure 4 examines routing under federated heterogeneity and continual task shifts. In panels (a,b), we fix six experts and partition the same training data using Dirichlet



(a) Orthogonal (b) Non-orthogonal (c) Relative drift distance

Figure 5: Historical LLM output drift. (a,b): output drift on 128 ImageNet-R examples; darker blue denotes larger normalized drift. (c): mean relative drift across seven historical tasks.

$\alpha = 0.1$; each row shows one task's expert assignments. Calibrating the router obtained after 20 FedAvg rounds raises Flickr30k source-expert accuracy from 36.85% to 97.01%. Panels (c,d) evaluate all seen tasks after each learning stage: ImageNet-R routing accuracy falls from 100% after Task 1 to 9% after Task 4 without calibration, versus 73% with calibration. These results show improved access to retained experts across both settings. Appendix B.7 examines soft versus hard routing and sensitivity to statistical-memory coverage.

Historical computation preservation. We test whether orthogonal backbone updates preserve historical computations. For each task, we compare each model's final eight-task layer outputs with its own task-end checkpoint, using the same source expert and evaluation inputs. Figure 5(a,b) visualizes ImageNet-R drift across 32 LLM layers and 64 hidden-channel blocks; darker regions indicate larger deviations. Non-orthogonal updates produce larger drift across nearly all layers and blocks. Panel (c) confirms this trend in the mean layer-wise drift over seven historical tasks. These results support the preservation of historical computations by orthogonal adaptation; detailed results appear in Appendix B.5.

7.4 Communication and Memory Budget

Figure 6 compares cumulative communication over eight tasks at matched LoRA ranks. At rank 32, DART requires 168.95 GiB, versus 149.46 GiB for FedAvg and 191.16 GiB for DISCO, reducing communication by 11.62% relative to DISCO. At rank 8, however, DART is more costly because task-boundary gradient overhead does not decrease with LoRA rank. Routing statistics and protection information therefore introduce fixed overhead, while remaining competitive at the rank used in our main experiments. Detailed accounting assumptions and storage costs are provided in Appendix B.9.

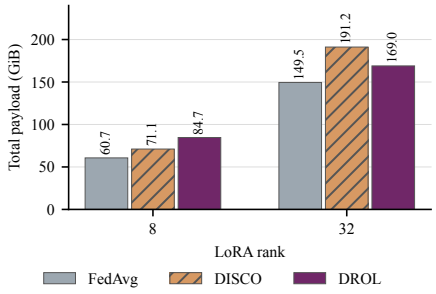


Figure 6: **communication analysis.**

8 Conclusion

We study reliable expert reuse in federated continual learning for vision-language models. DART combines distribution-guided global router calibration with orthogonal shared-backbone adaptation to address routing degradation across clients and tasks as well as interference with historical expert computations. Our experiments demonstrate that distribution-guided calibration effectively restores access to retained experts, while the theoretical and empirical analyses show that orthogonal adaptation helps preserve the computations associated with historical tasks. Together, these results indicate that calibrated routing and orthogonal backbone updates provide complementary mechanisms for recovering historical task computations during continual adaptation. Future work will explore more adaptive distributional memory and routing strategies under stronger heterogeneity.

AI USE STATEMENT

Generative AI tools assisted with research methodology, writing, and checking of theoretical claims and proofs. They also assisted with literature search, manuscript drafting and language editing, and figure preparation. The authors take responsibility for the final content of this work, including all AI-assisted text, claims, and artifacts.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the benchmarks, data partitions, evaluation protocols, and implementation settings in the appendix. Training configurations and additional experimental results are included to support reproducibility. The source code is available in an anonymous repository at https://anonymous.4open.science/r/DART-C63D.

REFERENCES

Cheng Chen, Junchen Zhu, Xu Luo, Heng Tao Shen, Jingkuan Song, and Lianli Gao. CoIN: A benchmark of continual instruction tuning for multimodel large language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Haiyang Guo, Fei Zhu, Wenzhuo Liu, Xu-Yao Zhang, and Cheng-Lin Liu. PILoRA: Prototype guided incremental LoRA for federated class-incremental learning. In *European Conference on Computer Vision*, pp. 141–159, 2024.

Haiyang Guo, Fanhu Zeng, Fei Zhu, Wenzhuo Liu, Da-Han Wang, Jian Xu, Xu-Yao Zhang, and Cheng-Lin Liu. Federated continual instruction tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1325–1335, 2025.

Tao Guo, Junwei Chen, and Laizhong Cui. Fed-Duet: Dual expert-orchestrated framework for continual federated vision-language learning. In *International Conference on Learning Representations*, 2026.

ZhiYan Hou, Haiyun Guo, Haokai Ma, Yandu Sun, Yonghui Yang, and Jinqiao Wang. PASs-MoE: Mitigating misaligned co-drift among router and experts via pathway activation subspaces for continual learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31959–31972, 2026. doi: 10.18653/v1/2026.acl-long.1474.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Hyundong Jin, Hyung Jin Chang, and Eunwoo Kim. Instruction-grounded visual projectors for continual learning of generative vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3466–3476, 2025. doi: 10.1109/ICCV51701.2025.00331.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. doi: 10.1073/pnas.1611835114.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2018. doi: 10.1109/TPAMI.2017.2773081.

Xiuheng Liao, Zhiwei Si, Tao Liu, Di Wu, Buzhen He, Weiyu Wang, and Chunhua Su. Federated continual learning with domain-routed historical expert matching for multimodal perception. *Pattern Recognition*, 180:114532, 2026. doi: 10.1016/j.patcog.2026.114532.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26296–26306, 2024.

Mao-Lin Luo, Zi-Hao Zhou, Yi-Lin Zhang, Yuanyu Wan, Min-Ling Zhang, and Tong Wei. KeepLoRA: Continual learning with residual gradient adaptation. In *International Conference on Learning Representations*, 2026.

Daiqing Qi, Handong Zhao, and Sheng Li. Better generative replay for continual federated learning. In *International Conference on Learning Representations*, 2023.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10658–10671, 2023. doi: 10.18653/v1/2023.findings-emnlp.715.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 139–149, 2022.

Zhen-Hao Xie, Jun-Tao Tang, Yu-Cheng Shi, Han-Jia Ye, De-Chuan Zhan, and Da-Wei Zhou. SAME: Stabilized mixture-of-experts for multimodal continual instruction tuning. In *International Conference on Machine Learning*, 2026.

Xiang Xu, Yiyang Su, Tianchen Zhao, Zheng Zhang, Zhuowen Tu, Anil Jain, and Jonathan Wu. Beyond disjoint tasks: Towards more natural continual learning for vision-language models. In *European Conference on Computer Vision*, 2026.

Jaehong Yoon, Wonyong Jeong, Giwoong Lee, Eunho Yang, and Sung Ju Hwang. Federated continual learning with weighted inter-client transfer. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 12073–12086, 2021.

Wenhao Yuan, Chenchen Lin, Jian Chen, Jinfeng Xu, Zewei Liu, and Edith Cheuk Han Ngai. Fisher-routed mixture of experts for federated class-incremental learning. In *European Conference on Computer Vision*, 2026.

Duzhen Zhang, Yahan Yu, Qiaoyi Su, Jiahua Dong, and Tielin Zhang. Progressive multimodal alignment for continual instruction tuning. In *Proceedings of the 34th ACM International Conference on Multimedia*, 2026. doi: 10.1145/3767308.3836259.

Jie Zhang, Chen Chen, Weiming Zhuang, and Lingjuan Lyu. TARGET: Federated class-continual learning via exemplar-free distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4782–4793, 2023.

APPENDIX CONTENTS

A Theoretical Analysis

A.1 From Gradient Protection to Input Coverage 13
A.2 Historical Computation Preservation 14
A.3 Numerical Error and Scope of the Guarantee 15
A.4 Orthogonal Subspaces and Federated Aggregation 16
A.5 Overall Algorithm 17

B Supplementary Experiments

B.1 Benchmarks and Evaluation Metrics 18
B.2 Routing Statistics and Calibration Settings 18
B.3 Detailed Hom-FCIT Results 19
B.4 Centralized Six-Task UCIT Comparison 19
B.5 Spatial and Temporal Routing Diagnostics 21
B.6 Empirical Diagnostics of the Theoretical Components 23
B.7 Routing Behavior and Memory Coverage 24
B.8 Eleven-Dataset CLIP Benchmark 25
B.9 Communication and Storage Analysis 26
B.10 Qualitative Examples 28

A Theoretical Analysis

We first connect the gradient subspace used by the algorithm to the inputs of the adapted layers, then propagate the resulting perturbations through the decoder. All weights and probe matrices are conditioned on the realized training trajectory; expectations refer only to a specified historical evaluation distribution. The reference is the actual model after task i, not a separately trained or jointly optimized model. Vector norms are Euclidean, matrix norms with subscript 2 are spectral, and $\|\cdot\|_F$ denotes the Frobenius norm. LoRA scales are absorbed into B throughout the analysis.

A.1 From Gradient Protection to Input Coverage

Fix an adapted linear layer and suppress its index. Let $U_i \in \mathbb{R}^{d_i \times n}$ contain its historical input vectors as columns, evaluated using the entire task-i model and expert P_i. The token positions, mask, and response prefix are held fixed in each historical/current comparison. Its uncentered second moment is $C_i = \mathbb{E}_i[U_i U_i^\top]$, with $\mathrm{tr}(C_i) < \infty$; using a centered covariance would omit the mean-input contribution. Let $H_i = (G_i^{\mathrm{end}})^\top \in \mathbb{R}^{d_i \times d_o}$ be the transpose of the actual client-weighted mean gradient used by Equation 19, at this same layer and task boundary. It is not a matrix of individual gradients or a gradient second moment.

Write $H_i^\dagger$ for the Moore–Penrose inverse and $J_i = H_i H_i^\dagger$ for the orthogonal projector onto its column space. For an orthonormal protection basis Q_i, define $\Pi_i = I - Q_i Q_i^\top$, $\kappa_i = \|H_i^\dagger C_i^{1/2}\|_2$, and $\xi_i = \|\Pi_i (I - J_i) C_i^{1/2}\|_F$, where $C_i^{1/2}$ is the positive semidefinite square root. Thus ξ_i is the root mean square contribution of input directions unobserved by the mean gradient and still unprotected by Q_i. These quantities characterize the guarantee; the algorithm need not store C_i or historical activations.

Proposition 1 (Gradient-to-input coverage)**.** *Let $U_i, H_i, C_i, Q_i, \kappa_i, \xi_i$ be as above. Suppose that Q_j has orthonormal columns and $\mathrm{span}(Q_i) \subseteq \mathrm{span}(Q_j)$ for $j \geq i$. Set $\varepsilon_i = \|\Pi_i H_i\|_F$ and $b_i = \kappa_i \varepsilon_i + \xi_i$. Then*

$$\left(\mathbb{E}_i \|\Pi_j U_i\|_F^2\right)^{1/2} = \|\Pi_j C_i^{1/2}\|_F \leq b_i. \tag{10}$$

If additionally $\|Q_i^\top H_i\|_F^2 \geq \eta \|H_i\|_F^2$ for $\eta \in [0,1]$, then $b_i \leq \kappa_i \sqrt{1-\eta}\, \|H_i\|_F + \xi_i$.

Proof. Set $M_i = H_i^\dagger C_i^{1/2}$ and $E_i = (I - J_i) C_i^{1/2}$, so that $C_i^{1/2} = H_i M_i + E_i$. Nested orthogonal projections satisfy $\Pi_j \Pi_i = \Pi_j$ and $\|\Pi_j\|_2 \leq 1$. The second-moment identity, projection contraction, and the Frobenius product inequality give the chain

$$\begin{aligned}
\left(\mathbb{E}_i \|\Pi_j U_i\|_F^2\right)^{1/2} &= \left(\mathrm{tr}(\Pi_j C_i \Pi_j)\right)^{1/2} = \|\Pi_j C_i^{1/2}\|_F \\
&= \|\Pi_j \Pi_i (H_i M_i + E_i)\|_F \\
&\leq \|\Pi_i H_i M_i\|_F + \|\Pi_i E_i\|_F \\
&\leq \|\Pi_i H_i\|_F \|M_i\|_2 + \|\Pi_i E_i\|_F = \kappa_i \varepsilon_i + \xi_i.
\end{aligned} \tag{11}$$

Orthogonality also gives $\varepsilon_i^2 = \|H_i\|_F^2 - \|Q_i^\top H_i\|_F^2 \leq (1-\eta)\|H_i\|_F^2$, proving the last claim. The proof includes $H_i = 0$: then $H_i^\dagger = 0$, $\kappa_i = 0$, and $b_i = \|\Pi_i C_i^{1/2}\|_F$, so a vanishing mean gradient gives no additional input-coverage guarantee. □

The proposition separates a controlled gradient tail from two problem-dependent quantities. Small ε_i is useful when κ_i is moderate and ξ_i is small; neither follows from the energy threshold. A concrete sufficient condition is stable observability, $C_i \preceq c_i^2 H_i H_i^\top$ for a finite $c_i \geq 0$. Indeed, this condition places the range of $C_i^{1/2}$ in the range of H_i, giving $\xi_i = 0$; congruence by $H_i^\dagger$ gives $\kappa_i \leq c_i$. This is an additional property of the gradient/input pair, not an assumption imposed by the algorithm. Weak singular directions can instead make κ_i large, and averaging across clients, examples, or tokens can hide input directions.

For example, let a one-output linear layer receive $u = e_1$ or e_2 with equal probability, with output derivatives $\delta(e_1) = 2$ and $\delta(e_2) = 0$. Then its mean gradient is $G_i = e_1^\top$ and $C_i = I_2/2$. The basis $Q_i = e_1$ captures all gradient energy, yet the allowed update $\Delta W = e_2^\top$ changes the output on

e_2; here $\varepsilon_i = 0$ but $\xi_i = 1/\sqrt{2}$. This explicitly rules out identifying retained gradient energy with retained activation energy.

The input-coverage result concerns a frozen historical forward pass. We next use it to control the effect of later weight increments at every adapted layer, and then account for the changing inputs caused by all preceding operations.

A.2 Historical Computation Preservation

Represent a decoder forward pass at a fixed context by a finite directed acyclic computation graph, in topological order. Its input node 0 contains visual tokens and the fixed text/context embeddings; its output node o returns the next-token logits. Let $X_{k,j}$ denote node k's value at task boundary j. Fixed operations include nonlinearities, attention products, normalization, and residual additions. Each adapted node $\ell \in \mathcal{L}$ is a linear operation $W_{\ell,j}U_{\ell,j} + a_\ell$ with a fixed bias. If a weight is used at several nodes, each occurrence is included with its corresponding input.

For every node, let nonnegative constants $L_{k,p}$ bound its task-t operation separately in each parent p: $\|f_{k,t}(x) - f_{k,t}(y)\|_F \le \sum_{p\in \mathrm{pa}(k)} L_{k,p}\|x_p - y_p\|_F$. These are uniform bounds on a common domain containing the two trajectories and all mixed parent tuples used in the comparison, not merely Jacobian norms at the historical states. For an adapted linear node with parent p, take $L_{\ell,p} = \|W_{\ell,t}\|_2$. Let K_k be the sum, over directed paths from k to o, of the products of the edge constants along each path. The empty path has weight one, and the sum is zero if no path exists. These factors include residual branches; no serial-MLP approximation of the Transformer is made.

For every adapted node, define $C_{\ell,i}, H_{\ell,i}, \varepsilon_{\ell,i}, \kappa_{\ell,i}, \xi_{\ell,i}, b_{\ell,i}$ as in Proposition 1, using its actual input $U_{\ell,i}$ in the historical decoder. In particular, an internal layer's $C_{\ell,i}$ is not the projector-output second moment. All constants refer to the specified historical distribution, which may be an empirical evaluation distribution.

Theorem 1 (restated) (Gradient protection and historical preservation)**.** *Fix $1 \le i \le t$ and the computation graph and constants above. Suppose every adapted weight has the actual accumulated form $W_{\ell,j} = W_{\ell,0} + \sum_{s=1}^{j} B_{\ell,s}A_{\ell,s}$, with all other decoder parameters and historical expert P_i fixed. Assume nested orthonormal bases and $A_{\ell,s}Q_{\ell,s-1} = 0$ for $s > i$. Let $\pi_{t,s}(h) \ge 0$, $\sum_{s=1}^{t} \pi_{t,s}(h) = 1$, and $Z_t = \sum_{s=1}^{t} \pi_{t,s}(h)P_s(v)$. Assume the Lipschitz bounds above hold and the historical inputs and routing discrepancy have finite second moments. Set $S_{\ell,t,i} = \sum_{s=i+1}^{t} \|B_{\ell,s}\|_2\|A_{\ell,s}\|_2$ and $R_{t,i} = (\mathbb{E}_i[(\sum_{s\neq i} \pi_{t,s}(h)\|P_s(v) - P_i(v)\|_F)^2])^{1/2}$. Then*

$$\left(\mathbb{E}_i\|\mathcal{D}_t(Z_t, q) - \mathcal{D}_i(P_i(v), q)\|_2^2\right)^{1/2} \le K_0 R_{t,i} + \sum_{\ell\in\mathcal{L}} K_\ell S_{\ell,t,i} b_{\ell,i}.$$

Here the response prefix is fixed and suppressed in the notation $\mathcal{D}_j(\cdot, q)$. If $R_{t,i} = 0$ and all $b_{\ell,i} = 0$, the historical logits are preserved almost surely.

Proof. First consider a historical input $U_{\ell,i}$ at one adapted node. For $s > i$, $A_{\ell,s}Q_{\ell,s-1} = 0$ implies $A_{\ell,s}U_{\ell,i} = A_{\ell,s}\Pi_{\ell,s-1}U_{\ell,i}$. Using the accumulated update identity, Minkowski's inequality in L^2, and Proposition 1 yields

$$\begin{aligned}\left(\mathbb{E}_i\|(W_{\ell,t} - W_{\ell,i})U_{\ell,i}\|_F^2\right)^{1/2} &= \left(\mathbb{E}_i\left\|\sum_{s=i+1}^{t} B_{\ell,s}A_{\ell,s}\Pi_{\ell,s-1}U_{\ell,i}\right\|_F^2\right)^{1/2} \\ &\le \sum_{s=i+1}^{t} \|B_{\ell,s}\|_2\|A_{\ell,s}\|_2 \left(\mathbb{E}_i\|\Pi_{\ell,s-1}U_{\ell,i}\|_F^2\right)^{1/2} \\ &\le S_{\ell,t,i}b_{\ell,i}.\end{aligned} \tag{12}$$

Only increments after task i appear; all earlier changes are already included in the actual reference $W_{\ell,i}$.

Next write $d_k = (\mathbb{E}_i\|X_{k,t} - X_{k,i}\|_F^2)^{1/2}$. For an adapted node ℓ with parent p, adding and subtracting $W_{\ell,t}U_{\ell,i}$, rather than assuming its input is unchanged, gives

$$\begin{aligned} d_\ell &= \left(\mathbb{E}_i\,\|W_{\ell,t}(U_{\ell,t} - U_{\ell,i}) + (W_{\ell,t} - W_{\ell,i})U_{\ell,i}\|_F^2\right)^{1/2} \\ &\le \|W_{\ell,t}\|_2 d_p + \left(\mathbb{E}_i\|(W_{\ell,t} - W_{\ell,i})U_{\ell,i}\|_F^2\right)^{1/2} \\ &\le L_{\ell,p}d_p + S_{\ell,t,i}b_{\ell,i}. \end{aligned} \tag{13}$$

At an unchanged node the assumed Lipschitz inequality and Minkowski's inequality give $d_k \le \sum_p L_{k,p}d_p$. Thus, with $e_\ell = S_{\ell,t,i}b_{\ell,i}$ at adapted nodes and $e_k = 0$ otherwise, every node satisfies $d_k \le \sum_p L_{k,p}d_p + e_k$. Expanding this recurrence in topological order assigns each source error the sum of products along all of its downstream paths, precisely the coefficient K_k.

At node 0, the text and context embeddings coincide, while $Z_t - P_i(v) = \sum_{s\neq i}\pi_{t,s}(h)(P_s(v) - P_i(v))$. Consequently $d_0 \le R_{t,i}$ by the triangle inequality and nonnegativity of the routing weights. Substituting the node bounds and this input bound into the expanded recurrence gives

$$\begin{aligned} \left(\mathbb{E}_i\|\mathcal{D}_t(Z_t, q) - \mathcal{D}_i(P_i(v), q)\|_2^2\right)^{1/2} = d_o &\le K_0 d_0 + \sum_{k\neq 0} K_k e_k \\ &\le K_0 R_{t,i} + \sum_{\ell\in\mathcal{L}} K_\ell S_{\ell,t,i}(\kappa_{\ell,i}\varepsilon_{\ell,i} + \xi_{\ell,i}), \end{aligned} \tag{14}$$

which proves Equation 9. When all stated error terms vanish, the nonnegative squared logit difference has zero expectation and is therefore zero almost surely. Empty update sums cover $t = i$. □

The same argument applies to any intermediate node chosen as the output, so it also bounds hidden-representation drift. A correct router argmax alone does not set $R_{t,i}$ to zero for a soft mixture; assigning all mass to i is sufficient. The graph bound accounts for upstream drift and downstream amplification, but its constants can be large. It makes no optimization-convergence claim and does not require the current weights to equal the old weights.

The preceding guarantee uses exact subspaces to isolate the connection between gradients and representations. We now state how finite-precision orthogonality and imperfect retention of an earlier basis enter the same argument, rather than treating them as exact properties of stored factors.

A.3 Numerical Error and Scope of the Guarantee

Proposition 2 (Approximate subspace protection)**.** *At one adapted layer, keep C_i, H_i, Q_i, b_i from Proposition 1 and let $W_j = W_0 + \sum_{s=1}^{j} B_s A_s$, but allow later orthonormal bases Q_{s-1} to be nonnested and $A_s Q_{s-1} \neq 0$. Define $\alpha_{s,i} = \|\Pi_{s-1}Q_i\|_2$, $c_i = \|Q_i^\top C_i^{1/2}\|_F$, and $c_{s,i} = \|Q_{s-1}^\top C_i^{1/2}\|_F$. Then*

$$\left(\mathbb{E}_i\|(W_t - W_i)U_i\|_F^2\right)^{1/2} \le \sum_{s=i+1}^{t} \|B_s\|_2\left[\|A_s\|_2(b_i + \alpha_{s,i}c_i) + \|A_s Q_{s-1}\|_2 c_{s,i}\right]. \tag{15}$$

This quantity can replace $S_{\ell,t,i}b_{\ell,i}$ at each adapted node in Theorem 1.

Proof. Decomposing $C_i^{1/2}$ using Q_i and contracting Π_{s-1} gives $\|\Pi_{s-1}C_i^{1/2}\|_F \le \|\Pi_i C_i^{1/2}\|_F + \|\Pi_{s-1}Q_i\|_2\|Q_i^\top C_i^{1/2}\|_F \le b_i + \alpha_{s,i}c_i$, where Proposition 1 is used only at $j = i$. Using $I = \Pi_{s-1} + Q_{s-1}Q_{s-1}^\top$ and the same second-moment identity then gives

$$\begin{aligned} \left(\mathbb{E}_i\|(W_t - W_i)U_i\|_F^2\right)^{1/2} &= \left\|\sum_{s=i+1}^{t} B_s A_s C_i^{1/2}\right\|_F \\ &\le \sum_{s=i+1}^{t} \|B_s\|_2\left(\|A_s\|_2\|\Pi_{s-1}C_i^{1/2}\|_F + \|A_s Q_{s-1}\|_2 c_{s,i}\right) \\ &\le \sum_{s=i+1}^{t} \|B_s\|_2\left[\|A_s\|_2(b_i + \alpha_{s,i}c_i) + \|A_s Q_{s-1}\|_2 c_{s,i}\right]. \end{aligned} \tag{16}$$

The decoder proof uses each local perturbation only through its L^2 upper bound, so replacing that bound proves the last claim. □

Each Q in this result is an orthonormal basis of the represented subspace. For rounded stored matrices, one must use an orthonormalized basis to interpret these projectors; $I - QQ^\top$ is otherwise not an orthogonal projector. Exact nesting gives $\alpha_{s,i} = 0$, and exact orthogonality also removes the final leakage term. The actual measured ε_i remains valid when the numerical energy criterion differs slightly from its nominal target.

The bridge explains what gradient protection establishes and what requires additional evidence. For an exactly protected later increment $\Delta W = B_s A_s$, nesting gives $|\langle G_i^{\mathrm{end}}, \Delta W\rangle_F| \le \varepsilon_i \|\Delta W\|_F$. This concerns the directional derivative of the sampled historical probe objective at its old checkpoint; it alone controls neither finite loss changes nor hidden representations. Proposition 1 supplies the missing representation terms, and Theorem 1 propagates their effect through the computation graph.

The probe clients and examples need not match the evaluation distribution. The identities still hold for the specified C_i, but small κ_i and ξ_i on unseen data do not follow from sampled gradient energy, and no population generalization bound is asserted. These coefficients, update sizes, and downstream sensitivities have not been measured in the reported experiments; the theorem therefore does not provide a numerical certificate of retention. Frozen routing descriptors make h invariant without making decoder activations invariant. The comparison fixes the response prefix and deterministic inference; a guarantee for a freely generated sequence would additionally require bounds along its prefixes and stability of the decoding decisions. Nor does the theorem assert that synthetic router calibration necessarily reduces $R_{t,i}$ on real test inputs.

A.4 Orthogonal Subspaces and Federated Aggregation

For each LLaVA layer, $W_\ell \in \mathbb{R}^{d_o \times d_i}$, $A_{\ell,t} \in \mathbb{R}^{r \times d_i}$, and $B_{\ell,t} \in \mathbb{R}^{d_o \times r}$. The protection basis $Q_{\ell,t} \in \mathbb{R}^{d_i \times k_{\ell,t}}$ has orthonormal columns, transposed relative to code storage. This construction section writes the unscaled code factor $B_{\ell,t}$ explicitly: the increment is $s_\ell B_{\ell,t} A_{\ell,t}$, whereas the main text absorbs s_ℓ into B. The scale is shared across clients for a given layer, not necessarily across layers. Initial directions come from a pretrained-weight protection cache with target squared-singular-value energy 0.6, not router-input statistics. LoRA rank is 32; protection rank grows with the energy criterion.

At the same inherited model, each probe client estimates an average parameter gradient $\widehat{G}^{\mathrm{start}}_{\ell,c,t}$ on 20% of its current training support, without an optimizer update. The server forms

$$G^{\mathrm{start}}_{\ell,t} = \sum_{c \in \mathcal{S}_t} \nu_{c,t} \widehat{G}^{\mathrm{start}}_{\ell,c,t}, \qquad \nu_{c,t} = \frac{|\mathcal{D}_{c,t}|}{\sum_{j \in \mathcal{S}_t} |\mathcal{D}_{j,t}|}. \tag{17}$$

Population counts weight this mean of gradients, not gradient Grams. The probe set $\mathcal{S}_t$ contains the five first-round clients, out of 50, and is reused at task end; it does not cover every client. Randomized SVD of the projected gradient transpose, followed by two residual-projection/QR passes, constructs A. The implementation audits protected-space overlap and fails if fewer than r free directions remain. FP32 construction and lower-precision storage make $AQ = 0$ a numerical approximation. The server retains Q and broadcasts the same A throughout the task.

Each round starts from common B and expert parameters. Clients train backbone B and the current projector on the response loss; A, inherited weights, historical experts, and the router are fixed. The server aggregates updates, calibrates the router, and broadcasts the state. After the final round, it merges

$$W_{\ell,t} = W_{\ell,t-1} + s_\ell B^{\mathrm{final}}_{\ell,t} A_{\ell,t}. \tag{18}$$

It clears B before collecting post-task probes at the merged model. For $H_{\ell,t} = (G^{\mathrm{end}}_{\ell,t})^\top$, residual directions come from $\Pi_{\ell,t-1} H_{\ell,t}$. The SVD search expands until the old plus new directions satisfy

$$\frac{\|Q_{\ell,t}^\top H_{\ell,t}\|_F^2}{\|H_{\ell,t}\|_F^2} \ge \eta, \qquad \eta = 0.99. \tag{19}$$

The criterion covers the complete average gradient, not its residual alone. An already sufficient basis or zero gradient requires no addition. Captured energy and orthogonality are audited after re-projection. This does not certify historical activation coverage.

Equation 8 follows by linearity in B with shared A, scaling, base weights, and normalized client weights. Different client-specific A_c invalidate it: multiplying separately averaged factors introduces cross-client terms. The identity concerns constrained LoRA increments, not unrestricted full fine-tuning. Besides B, uploads include current projector parameters, routing summaries, and task-boundary gradient probes. Downloads include A; protection bases remain server-side. Appendix B.9 distinguishes dense-probe accounting from the compressed communication scenario.

A.5 Overall Algorithm

Algorithm 1 summarizes the workflow. Notation follows Equations 4, 5, and 17; K synthetic descriptors are drawn per expert from $\mathcal{M}_{t,m}$. Backbone operations are layerwise, with LoRA scales absorbed into B.

Algorithm 1 DART for federated continual instruction tuning

Require: Task streams $\{\mathcal{D}_{c,t}\}$, pretrained model, rounds M, LoRA rank r, and protection threshold η.
Ensure: Backbone W_T, frozen projector experts $\{P_s\}_{s=1}^{T}$, and calibrated router ψ_T.

Server side: Initialize Q_0 and distribution memory $\mathcal{M}_0$.
for task $t = 1, \dots, T$ **do**
  Clients side: Compute and upload $\widehat{G}_{c,t}^{\text{start}}$.
  Server side:
  $G_t^{\text{start}} \leftarrow \sum_{c\in\mathcal{S}_t} \nu_{c,t}\widehat{G}_{c,t}^{\text{start}}$.
  $\Pi_{t-1} \leftarrow I - Q_{t-1}Q_{t-1}^{\top}$.
  $A_t \leftarrow \mathrm{SVD}_r\big(\Pi_{t-1}(G_t^{\text{start}})^{\top}\big)^{\top}$.
  Freeze A_t; initialize $B_{t,0}$ and $P_{t,0}$.
  for communication round $m = 1, \dots, M$ **do**
    Clients side: For selected clients $c \in \mathcal{C}_{t,m}$ **in parallel**:
    $(B_{c,t,m}, P_{c,t,m}) \leftarrow \mathrm{LocalTrain}_{\mathcal{L}_{\text{inst}}}(\mathcal{D}_{c,t}; A_t)$.
    Compute and upload routing moments $(\mu_{c,t}, S_{c,t})$ and local updates.
    Server side: Aggregate local updates with FedAvg.
    $\mu_{t,m} \leftarrow \sum_{c\in\mathcal{C}_{t,m}} \alpha_{c,t,m}\mu_{c,t}$.
    $\Sigma_{t,m} \leftarrow \frac{N_{t,m}}{N_{t,m}-1}\sum_{c\in\mathcal{C}_{t,m}} \alpha_{c,t,m}\big[S_{c,t} + (\mu_{c,t}-\mu_{t,m})(\mu_{c,t}-\mu_{t,m})^{\top}\big]$.
    $\mathcal{M}_{t,m} \leftarrow \mathcal{M}_{t-1} \cup \{(t, \mu_{t,m}, \Sigma_{t,m})\}$.
    $\tilde{h}_{s,j} \sim \mathcal{N}(\mu_s, \Sigma_s), \quad s = 1, \dots, t, \quad j = 1, \dots, K$.
    $\psi_{t,m} \leftarrow \mathrm{Optimize}_{\psi}\ \frac{1}{tK}\sum_{s=1}^{t}\sum_{j=1}^{K} \mathrm{CE}\Big(q_s^{\epsilon}, \pi_{\psi}(\tilde{h}_{s,j})\Big)$.
    Broadcast $B_{t,m}$, $P_{t,m}$, and calibrated $\psi_{t,m}$.
  end for
  Server side: Merge $B_{t,M}A_t$; freeze P_t and retain $\mathcal{M}_t$ and ψ_t.
  Clients side: Compute and upload $\widehat{G}_{c,t}^{\text{end}}$ at the merged model.
  Server side:
  $G_t^{\text{end}} \leftarrow \sum_{c\in\mathcal{S}_t} \nu_{c,t}\widehat{G}_{c,t}^{\text{end}}$.
  $U_t \leftarrow \mathrm{SVD}_{k_t}\big(\Pi_{t-1}(G_t^{\text{end}})^{\top}\big)$.
  $Q_t \leftarrow \mathrm{orth}([Q_{t-1}, U_t])$.
end for
return W_T, $\{P_s\}_{s=1}^{T}$, and ψ_T.

B Supplementary Experiments

B.1 Benchmarks and Evaluation Metrics

We evaluate task acquisition and retention under two generative Hom-FCIT sequences. The task-related sequence is ImageNet-R, ArxivQA, IconQA, CLEVR-Math, OCR-VQA, Flickr30k, FigureQA, and super-CLEVR. The capability-related sequence groups twelve datasets into General (AOKVQA, IconQA, Grounding, ImageNet-R), Other (Flickr30k, OCR-VQA), Chart (ArxivQA, FigureQA, DVQA), Math (CLEVR-Math, super-CLEVR, TabMWP). The two models are trained independently, with one projector per stage. Our capability sequence follows the released partitions; DISCO's supplementary Table B6 instead labels Chart before Math (Guo et al., 2025).

Let $\mathcal{A}_{t,j}$ be dataset j's native benchmark score after stage t, and $\mathcal{J}_t$ the datasets seen by that stage. The main metrics equally weight datasets and then stages:

$$\text{Last} = \frac{1}{|\mathcal{J}_T|}\sum_{j\in\mathcal{J}_T}\mathcal{A}_{T,j}, \qquad \text{Avg} = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{|\mathcal{J}_t|}\sum_{j\in\mathcal{J}_t}\mathcal{A}_{t,j}. \tag{20}$$

Here $|\mathcal{J}_t| = t$ for the eight-task sequence and $4, 6, 9, 12$ for the capability sequence. Caption scores are not classification accuracies. Forgetting averages each historical dataset's best pre-final score minus its final score, excluding newly introduced final-stage datasets and allowing negative values. Source-expert top-1 accuracy is a routing diagnostic, not generation quality or utility-optimal expert selection. PMA's MFN corresponds to a final-task mean, whereas its diagonal MFT is not our Avg.

The main baselines are published DISCO references, not matched reruns. Individual and Centralized MTL use different task/data access; M-LoRA denotes DISCO's LoRA-merging component rather than the complete PILoRA method. Models use LLaVA-1.5-7B, rank-32 orthogonal LoRA, ten rounds per stage, five of fifty clients per round, and one local epoch. Calibration settings and orthogonal construction are given in Appendices B.2 and A.4. More detailed settings are in Table 13.

B.2 Routing Statistics and Calibration Settings

The formal LLaVA path collects $m_c = \min(128, n_c)$ training records without replacement, where n_c counts all current-task records for client c, not unique images. We suppress task and round indices here and require $m_c \geq 2$, as enforced by the estimator. Sampling uses a task/client-specific seed, so repeated participation selects the same record indices. The descriptor concatenates mean pre-projector CLIP patch features with mean frozen embedding-lookup vectors over valid formatted prompt tokens. The collector formats the first user instruction with the Vicuna template and no assistant response. It uses neither target answers nor contextual decoder states, and applies no descriptor-level ℓ_2 normalization.

Let H_c contain the centered sampled descriptors as rows, and let d_h be their dimension. Randomized PCA with four power iterations approximates its leading right singular vectors U_c and singular values $\sigma_{c,j}$ at rank $q_c = \min(32, m_c - 1, d_h)$. With elementwise clamping, the stored local fit is

$$\begin{aligned} \Lambda_{c,jj} &= \frac{\sigma_{c,j}^2}{m_c - 1}, \qquad d_c = \frac{\operatorname{diag}(H_c^\top H_c)}{m_c - 1}, \\ r_c &= \max\!\left(d_c - \operatorname{diag}(U_c\Lambda_c U_c^\top), 10^{-6}\right), \\ \widehat{\Sigma}_c &= U_c\Lambda_c U_c^\top + \operatorname{Diag}(r_c), \qquad S_c = \frac{m_c - 1}{m_c}\widehat{\Sigma}_c. \end{aligned} \tag{21}$$

Thus, S_c in Equation 4 uses the population convention of the sampled fit; it is not the exact population covariance of the client's full dataset.

For $N = \sum_c n_c$, $\alpha_c = n_c/N$, $\gamma = N/(N-1)$, and $\mu = \sum_c \alpha_c\mu_c$, the server retains a factor L and diagonal residual r:

$$\begin{aligned} L &= \operatorname{concat}_c\left[\sqrt{\gamma\alpha_c\frac{m_c - 1}{m_c}}\,U_c\Lambda_c^{1/2},\ \sqrt{\gamma\alpha_c}\,(\mu_c - \mu)\right], \\ r &= \gamma\sum_c \alpha_c\frac{m_c - 1}{m_c}r_c, \qquad \Sigma = LL^\top + \operatorname{Diag}(r). \end{aligned} \tag{22}$$

Table 3: Final per-dataset scores on eight-task Hom-FCIT ($\beta = 1.0$). $\Delta = DART - \text{DISCO}$.

Dataset	DISCO	DART	Δ
ImageNet-R	72.56	**85.21**	+12.65
ArxivQA	92.34	**92.39**	+0.05
IconQA	55.59	**62.95**	+7.36
CLEVR-Math	47.72	**57.27**	+9.55
OCR-VQA	35.97	**62.48**	+26.51
Flickr30k	52.49	**55.72**	+3.23
FigureQA	**42.92**	42.05	−0.87
super-CLEVR	**50.16**	49.14	−1.02
Last	56.22	**63.40**	+7.18

There is no second PCA truncation: five clients contribute at most $5(32 + 1) = 165$ factor columns. L need not be orthonormal. These factors exactly implement Equation 4 for the fitted summaries, not an exact full-data covariance. Sampling uses independent standard normal vectors:

$$\tilde{h} = \mu + L\xi + \sqrt{r} \odot \epsilon, \qquad \xi \sim \mathcal{N}(0, I), \quad \epsilon \sim \mathcal{N}(0, I). \tag{23}$$

Each round constructs its current summary from that round's participating clients only. The server records N and $\sum_c m_c$ separately; repeated participation does not accumulate counts across rounds. Only the final-round current summary is committed at a task boundary, and past summaries remain unchanged. Consequently, the memory represents sampled support from those participants, not all clients or every mode encountered during the task. Frozen reference coordinates prevent coordinate drift but cannot recover unobserved support. Sharing moments avoids historical raw-sample replay; it is not a formal differential-privacy guarantee.

For each calibration call, the server generates a fixed pool of 1,024 descriptors per seen expert and standardizes each coordinate using the pool mean and population standard deviation, clamped below at 10^{-3}. It initializes the full linear router from its current parameters in these standardized coordinates, then performs 400 AdamW updates with batch size 256, learning rate 0.03, weight decay 10^{-4}, and label smoothing 0.02. Minibatches are sampled from this pool with replacement; this does not alter represented counts. Finally, standardization is folded into the router weights and biases, retaining the raw-coordinate inference rule in Equation 1.

B.3 Detailed Hom-FCIT Results

We examine whether aggregate gains are shared across datasets and capabilities. All results use complete official test sets: 61,310 records for the eight-task final evaluation and 79,141 for the four-stage final evaluation. Tables 3 and 4 compare our seed-42 results with DISCO's supplementary Tables B8 and B6.

The eight-task gains are largest on OCR-VQA, ImageNet-R, and CLEVR-Math, while FigureQA and super-CLEVR decline slightly. In the capability sequence, General, Other, and Chart improve, but Math declines. Thus the aggregate benefit is not universal, and the capability-order difference prevents attributing every gap solely to the method.

The complete stage-mean trajectories are $[85.40, 88.96, 80.40, 74.52, 72.30, 69.41, 65.41, 63.40]$ for eight tasks at $\beta = 1.0$, $[86.68, 88.84, 78.76, 74.60, 71.87, 68.74, 65.36, 63.44]$ at $\beta = 0.5$, and $[71.98, 67.53, 60.43, 59.55]$ for four stages at $\beta = 1.0$. Avg is computed from unrounded values, giving 74.97, 74.79, and 64.87. Figure 3 uses these measurements; baseline curves are extracted from DISCO's vector artwork without rescaling to its table averages. Its radar uses final dataset scores.

B.4 Centralized Six-Task UCIT Comparison

We test whether distribution-guided routing also benefits centralized continual instruction learning. UCIT follows ImageNet-R, ArxivQA, VizWiz, IconQA, CLEVR-Math, and Flickr30k. We retain KeepLoRA's definitions: Transfer averages each task's pre-learning scores over tasks 2–6; Average

Table 4: Final scores on four-stage Hom-FCIT ($\beta = 1.0$).

		Dataset score	Capability mean		
Capability	Dataset	DART	DISCO	DART	Δ
General	AOKVQA	78.43	62.92	**71.90**	+8.98
	IconQA	58.30			
	Grounding	64.62			
	ImageNet-R	86.26			
Other	Flickr30k	55.46	53.14	**55.81**	+2.67
	OCR-VQA	56.16			
Math	CLEVR-Math	61.67	**56.15**	49.69	−6.46
	super-CLEVR	48.82			
	TabMWP	38.59			
Chart	ArxivQA	93.42	46.43	**55.45**	+9.02
	FigureQA	43.13			
	DVQA	29.80			
Last (equal mean over 12 datasets)			55.47	**59.55**	+4.08

Table 5: Centralized six-task UCIT results with LLaVA-1.5-7B. Unmarked baselines are from KeepLoRA Table 4 (Luo et al., 2026). Reproduced KeepLoRA and centralized post-hoc DART use 3,000 test examples per task and native dataset metrics.

Method	ImageNet-R	ArxivQA	VizWiz	IconQA	CLEVR	Flickr30k	Mean
Transfer							
O-LoRA	–	52.87	19.57	4.42	16.85	41.04	26.95
KeepLoRA	–	52.83	20.39	9.18	18.12	41.50	28.40
KeepLoRA (reproduced)	–	53.77	17.49	13.06	17.82	38.54	28.13
Average							
O-LoRA	82.43	80.06	41.73	35.87	33.94	43.74	52.96
KeepLoRA	86.50	83.63	42.66	40.08	35.24	44.11	55.37
KeepLoRA (reproduced)	85.46	85.61	39.47	44.28	35.77	41.59	55.36
Last							
O-LoRA	74.17	80.93	45.30	62.87	63.83	**57.24**	64.06
KeepLoRA	82.43	86.70	46.54	67.80	66.40	57.18	67.84
KeepLoRA (reproduced)	80.23	93.37	46.15	72.90	66.93	56.81	69.40
DART†	**91.40**	**93.73**	**60.39**	**79.63**	**75.63**	56.76	**76.26**

covers all six datasets at all six checkpoints, including future tasks; Last covers the final checkpoint. Its Average therefore differs from Hom-FCIT Avg.

Our reproduced KeepLoRA uses its native projector at each stage. The centralized DART variant combines the final KeepLoRA decoder with six saved projector experts and fits a router from Gaussian statistics of the six task training sets. Each reported local score uses 3,000 examples per task. VizWiz and Flickr30k use the seven-metric caption mean (BLEU-1–4, METEOR, ROUGE-L, CIDEr); the remaining tasks use accuracy. Table 5 reports the published baselines, our reproduction, and the calibrated endpoint.

KeepLoRA reproduced reaches Last 69.40, compared with the paper's 67.84; centralized calibration reaches 76.26, a 6.86-point difference from our reproduction. Most of this gain is on VizWiz, ImageNet-R, IconQA, and CLEVR-Math; Flickr30k decreases by 0.05. These scores support a useful centralized extension while remaining distinct from federated training.

B.5 Spatial and Temporal Routing Diagnostics

We isolate expert selection from backbone learning to test spatial routing imbalance and temporal forgetting. The six tasks follow Appendix B.4; descriptors concatenate frozen Task1 visual features and question embeddings. Each task supplies 256 training records, and exact training image–prompt matches are excluded from evaluation. Seeds 42–44 vary routing optimization, partitioning, and synthetic draws, not backbone training. Reported uncertainty is sample standard deviation. No answer labels enter descriptor construction.

Spatial competition. Five fully participating clients share identical global task counts under IID or task-label Dirichlet $\alpha = 0.1$ partitions. The baseline receives 20 FedAvg rounds, ten local updates per client, batch 64, and learning rate 0.003. Continued training, inverse-frequency balancing, and Gaussian calibration each receive 400 additional optimizer updates; this matches update count, not communication or FLOPs. Evaluation uses 256 fixed examples per task. Table 6 and Figures 1(a) and 4(a,b) show the results.

Table 6: Six-task spatial routing accuracy (%). Each seed uses the same 1,536 held-out examples; Balance denotes local inverse-frequency sampling.

Seed	IID	Non-IID	Continued	Balance	Diagonal	Low-rank
42	99.54	84.31	91.02	98.37	99.28	97.98
43	99.67	99.02	99.09	99.35	99.15	98.96
44	99.41	84.77	87.37	87.63	98.37	98.44

The mean error is $0.46 \pm 0.13\%$ under IID, $10.63 \pm 8.37\%$ under Non-IID, and $1.54 \pm 0.49\%$ after low-rank calibration. Continued training reaches 92.49% mean accuracy. The main failure is Flickr30k-to-VizWiz confusion: correct Flickr30k assignment rises from 36.85% to 97.01%. Calibration slightly harms one seed, and diagonal calibration exceeds low-rank calibration in two seeds. These results demonstrate setting-dependent routing imbalance, not a universal benefit from covariance correlations or recovery of unobserved clients.

Sequential calibration. To test genuine continual retention, Figure 4(c,d) compares independent never-calibrated and calibrated trajectories, not a final-round intervention. Each branch inherits only its own weights and adds one zero expert row per task; all seen rows remain trainable and Task1 normalization stays fixed. Only current-task real descriptors enter five-client IID training. The never-calibrated branch receives 28 rounds per stage; the calibrated branch receives 20 rounds plus 400 updates on 4,096 expert-balanced synthetic descriptors from seen-task rank-32-plus-diagonal moments. Both receive 1,400 batch-64 updates, using AdamW with learning rate 0.003, weight decay 10^{-4}, label smoothing 0.02, and local optimizer reset each round. Historical raw descriptors are discarded; evaluation uses the first 128 examples of each fixed spatial test set.

Final six-task accuracy rises from $48.87 \pm 1.48\%$ to $91.93 \pm 0.85\%$. Historical-task accuracy rises from $39.69 \pm 1.69\%$ to $90.68 \pm 1.30\%$, while latest-task accuracy changes from $94.79 \pm 4.71\%$ to $98.18 \pm 1.63\%$. Independent checkpoint replay reproduces all 16,128 test sample–run predictions. Thus historical statistical coverage reduces forgetting, although final ImageNet-R accuracy remains only 57.29%.

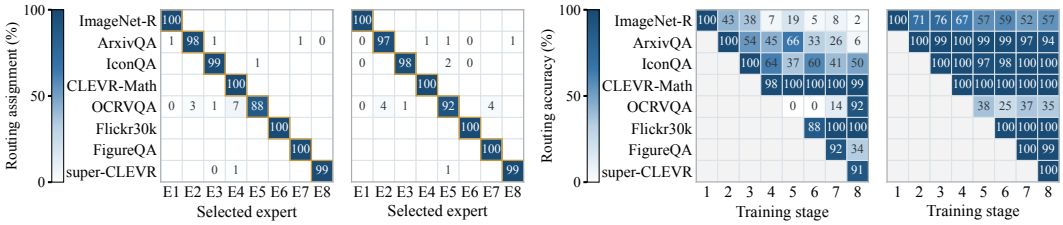
Historical coverage. We separately isolate the value of memory in Figure 1(b). Both branches calibrate for 400 updates on 3,072 synthetic descriptors per stage, using either the current Gaussian alone or an equal allocation over all seen Gaussians. They evolve independently, share fixed Task1 normalization, and use the optimizer above with 256 held-out examples per task. Historical-task error decreases from $63.44 \pm 4.89\%$ to $8.91 \pm 0.94\%$; latest-task error changes from $0.00 \pm 0.00\%$ to $1.56 \pm 1.35\%$. This isolates the benefit of historical distribution coverage, since both conditions use calibration.

Eight-task robustness check. We assess task-pool dependence using the eight-task order in Appendix B.1, with 256 training and 128 fixed test examples per task. Spatial runs expose all experts and use IID or Dirichlet $\alpha = 0.1$ partitions over five clients, rejecting singleton client–task cells and

clients with fewer than 51 examples. Temporal runs expose tasks sequentially. Both use the matched 1,400-update budgets, 4,096 synthetic descriptors, rank-32-plus-diagonal moments, and optimizer settings above; spatial normalization uses all training descriptors, whereas temporal normalization remains fixed at Task1. Table 7 retains all three seeds.

Table 7: Eight-task router-only accuracy (%). Spatial columns use a fixed expert pool; temporal columns report the final eight-task mean. These are not end-to-end benchmark scores.

	Spatial: IID		Spatial: Non-IID		Temporal: final	
Seed	No calib.	Calib.	No calib.	Calib.	No calib.	Calib.
42	96.29	97.75	98.54	96.88	59.38	85.84
43	99.61	99.71	99.12	99.90	57.62	87.79
44	99.22	99.02	96.29	98.14	60.64	83.11
Mean	98.37	98.83	97.98	98.31	59.21	85.58
Sample SD	1.81	0.99	1.49	1.52	1.52	2.35



(a) Spatial No calibration (b) Spatial Calibrated (c) Temporal No calibration (d) Temporal Calibrated

Figure 7: Eight-task router-only diagnostics, averaged over seeds 42–44 with 256 training and 128 test examples per task. (a,b) Row-normalized expert assignments under five-client Dirichlet $\alpha = 0.1$ partitions. (c,d) Per-task accuracy along independently trained continual trajectories with matched update counts. Gray cells denote unseen tasks.

Figure 7 shows the corresponding eight-task spatial assignments and stage-wise temporal accuracies. The spatial gain is only 0.33 points on average and reverses for seed 42. Temporal mean accuracy improves from 59.21% to 85.58%, but OCRVQA falls from 92.19% to 35.16%. These results limit any claim of universal routing recovery. The six- and eight-task diagnostics measure source-expert selection on frozen descriptors, not end-to-end generation or backbone drift.

Orthogonal-backbone ablation. Table 8 reports the complete final evaluation of independently trained orthogonal and unconstrained branches. Last improves by 5.24 points with orthogonal adaptation. Figure 1(c) selects the six tasks with positive final differences, rather than an outcome-independent subset; the two negative differences remain in the table and the full Last.

Table 8: Full eight-task comparison with calibrated routing. Differences are orthogonal minus unconstrained scores.

LLM update	ImageNet-R	ArxivQA	IconQA	CLEVR-Math	OCRVQA	Flickr30k	FigureQA	super-CLEVR	Last
Unconstrained	85.41	93.08	55.98	54.13	31.85	54.83	41.73	48.56	58.20
Orthogonal	85.21	92.39	62.95	57.27	62.48	55.72	42.05	49.14	63.40
Difference	−0.20	−0.69	+6.97	+3.14	+30.63	+0.89	+0.32	+0.58	+5.21

Layer-wise historical computation. We test whether orthogonal updates preserve historical decoder computations after the source expert is correctly selected. Using the independently trained $\beta = 0.5$, seed-42 branches, we compare each final model with its own checkpoint immediately after learning each of Tasks 1–7, on 128 matched test examples per task. Within each pair, the source projector is identical and frozen, and the full ground-truth response is teacher-forced. We measure all 32 decoder-block outputs at response-token positions before subsequent normalization. For each example, relative drift is $\|H_\ell^{\text{final}} - H_\ell^{\text{hist}}\|_F / (\|H_\ell^{\text{hist}}\|_F + \epsilon)$; we average examples and then tasks equally.

Table 9 gives task-level results. Orthogonal updates reduce mean relative drift from 0.1149 to 0.0141; the difference favors orthogonal updates for all 224 task–layer means. These comparisons support preservation of each branch's own historical computation, not equivalence to joint training or a common-start causal intervention.

Table 9: Relative LLM output drift against each branch's own historical model. Mean averages the 32 decoder blocks; Last layer denotes block 32, not the benchmark Last score.

	Orthogonal		Non-orthogonal	
Task	Mean	Last layer	Mean	Last layer
ImageNet-R	0.0157	0.0467	0.1366	0.3583
ArxivQA	0.0153	0.0578	0.1566	0.2237
IconQA	0.0174	0.0307	0.1585	0.2343
CLEVR-Math	0.0122	0.0248	0.1063	0.1606
OCRVQA	0.0163	0.0499	0.1128	0.2400
Flickr30k	0.0098	0.0126	0.0552	0.0779
FigureQA	0.0120	0.0129	0.0780	0.1087
Mean	0.0141	0.0336	0.1149	0.2005

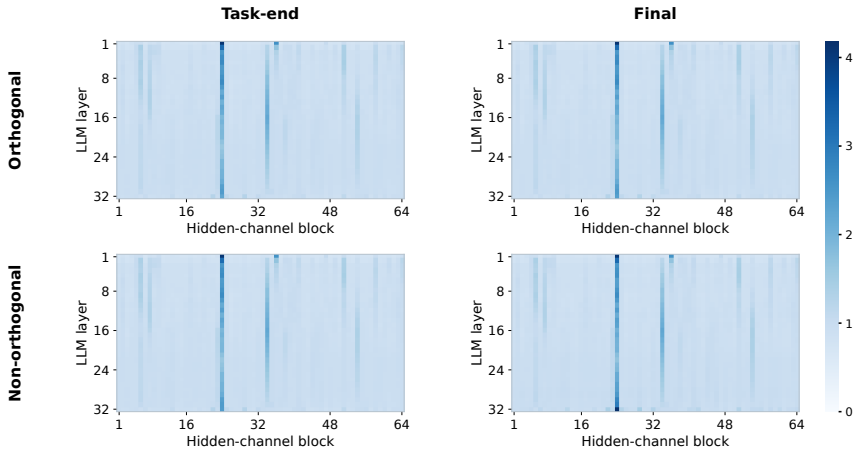


Figure 8: ImageNet-R activation magnitudes: orthogonal (top) and non-orthogonal (bottom), at the task-end checkpoint (left) and final checkpoint (right). Each cell groups 64 contiguous hidden channels in their original order. Block RMS is normalized by the same example's historical whole-layer RMS, then averaged over 128 examples; all four panels share a linear color scale. The main-text drift maps instead measure elementwise activation differences before aggregation, not differences between these magnitude maps.

B.6 Empirical Diagnostics of the Theoretical Components

We connect Theorem 1 to observable quantities using the orthogonal trajectory in Appendix B.5. The diagnostics separately examine routing mismatch and historical-input coverage, without changing the backbone or experts.

Routing mismatch. For historical task i, we estimate $R_{t,i}$ and the actual interface deviation $E_{t,i} = (\mathbb{E}_i\|Z_t - P_i(v)\|_F^2)^{1/2}$. We divide both by $(\mathbb{E}_i\|P_i(v)\|_F^2)^{1/2}$ and average the normalized values equally over historical tasks. Table 10(a) shows that calibrated routing reduces normalized R from 0.4173 to 0.0303 (92.75%) and normalized E from 0.3186 to 0.0303 (90.50%). The pointwise triangle inequality underlying $E_{t,i} \le R_{t,i}$ holds numerically throughout the diagnostic. Uniform mixing is an inference intervention on the same expert pool, not an independently trained never-calibrated model; the source expert defines the historical reference rather than an accuracy oracle.

Historical-input coverage. We directly measure the unprotected component of historical linear-layer inputs, using each task's own task-end model and protection basis:

$$\widehat{\rho}_{\ell,i} = \left(\frac{\sum_x \|(I - Q_{\ell,i}Q_{\ell,i}^\top)U_{\ell,i}(x)\|_F^2}{\sum_x \|U_{\ell,i}(x)\|_F^2} \right)^{1/2}. \tag{24}$$

Inputs are not mean-centered. Table 10(b) averages the module-wise ratios and compares each protection basis with a random orthonormal basis of identical rank. The learned protection space yields a smaller residual in all 45 task–module comparisons, indicating preferential coverage of historical activations beyond the effect of rank alone. Coverage is nevertheless incomplete: appreciable residuals remain, especially in some deeper modules.

Table 10: Empirical routing mismatch and historical-input residuals. Lower is better.

(a) Normalized routing mismatch: historical-task mean		
Metric	Uniform mixing	Calibrated routing
Routing term R	0.4173	**0.0303**
Interface deviation E	0.3186	**0.0303**
(b) Historical-input residual: module mean		
Historical task	Random subspace	Protection subspace
ImageNet-R	0.9244	**0.6433**
CLEVR-Math	0.9076	**0.5790**
FigureQA	0.8113	**0.4543**

Together, the measurements support the two mechanisms identified by the bound: calibrated routing reduces interface mismatch, while the protection space covers historical input directions more effectively than a same-rank random space. They are empirical component diagnostics, not a numerical certificate for the complete logit-error bound: the downstream sensitivity constants and all remaining bound factors are not estimated.

B.7 Routing Behavior and Memory Coverage

We test whether source-expert accuracy reflects soft-mixture inference and whether calibration is sensitive to client coverage. We fix the final eight-task model at $\beta = 1.0$, seed 42, and evaluate the same 128 test records per task. Table 11 compares inference rules and statistical-memory support without updating the backbone or experts.

The soft router assigns at least 0.99 probability to the source expert on 97.17% of examples. Soft and top-1 routing produce identical text on all 1,024 records; forced-source routing changes three responses but leaves the aggregate score unchanged. On the full 61,310-record test set, soft and top-1 Last scores are 63.4016 and 63.3928. These results connect the routing diagnostic to actual soft-mixture behavior at this checkpoint, rather than establishing an advantage of soft mixing or treating source identity as a performance oracle.

For memory coverage, we reconstruct one training-data summary per distinct client participating during each task's ten rounds, covering 29–35 clients instead of the final five, and pool them with population weights. Both memory conditions calibrate from the same Task8 round-9 router for 400 steps. Wider coverage increases source mass but does not improve the subset score. Thus final-round statistics are effective in this setting, not necessarily representative of all clients; this fixed-checkpoint intervention is neither an end-to-end streaming-memory ablation nor a held-out-client evaluation.

Table 11: Routing and memory controls with fixed backbone and experts. Top: 1,024 matched test records; source mass is the mean source-expert probability and the subset score averages native task metrics. Bottom: full-test Last on 61,310 records using the same saved statistics; Δ is DART minus each control.

Configuration	Source mass (%)	Subset score
DART: final-round memory + soft	98.24	64.29
Same router + top-1	98.44	64.29
Forced source expert	100.00	64.29
Within-task pooled memory + soft	99.49	64.10
Routing rule (full test)	**Last**	**Δ**
DART	63.40	—
Hard cosine nearest mean	61.81	+1.59
Soft mean distance	63.02	+0.38
Gaussian likelihood (low-rank + diagonal)	63.51	−0.11

For the full-test controls, soft-rule temperatures are selected using only synthetic draws from the same saved training statistics, with no additional real samples or test-set tuning. DART improves Last by 1.59 and 0.38 points over hard and soft mean routing but trails direct Gaussian likelihood by 0.11 points, so this single-seed endpoint comparison does not establish superiority over all distribution-based rules. Our contribution is federated distribution-based router calibration together with the maintenance of historical computation, rather than the exclusive superiority of a particular routing rule; effective alternatives for exploiting distribution memory are compatible with this framework.

Table 12: Eleven-dataset Order-I CLIP results (%). Dataset columns report final accuracies; Last and Avg are macro aggregates. All entries are local results. KeepLoRA and DART use centralized training; Fed-Duet uses IID federated training. Rows are descriptive references, not a matched-setting ranking.

Method	Aircraft	Caltech101	CIFAR100	DTD	EuroSAT	Flowers	Food	MNIST	OxfordPet	Cars	SUN397	Last	Avg
KeepLoRA	54.19	75.35	68.33	67.87	97.94	84.32	91.62	99.35	93.35	84.83	78.27	81.40	75.68
Fed-Duet	24.99	64.00	59.87	43.14	78.48	67.85	91.04	99.25	93.00	77.44	81.52	70.96	62.97
DART†	54.28	92.57	81.06	68.94	97.13	93.38	87.41	99.27	91.91	82.34	75.63	83.99	80.13

B.8 Eleven-Dataset CLIP Benchmark

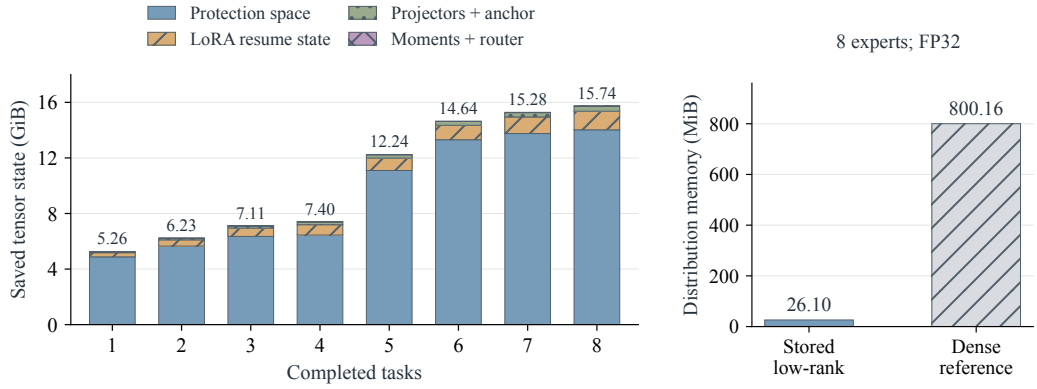
Table 12 reports the final dataset scores and macro aggregates. We examine whether distribution-guided adaptation also benefits CLIP beyond generative VLMs. We compare KeepLoRA, a centralized CLIP extension of DART, and a reproduced Fed-Duet (Guo et al., 2026) on the same eleven-dataset Order-I sequence: Aircraft, Caltech101, CIFAR100, DTD, EuroSAT, Flowers, Food, MNIST, OxfordPet, StanfordCars, and SUN397. All use CLIP ViT-B/16 and predict over all seen classes rather than a task-restricted label set. Last is the equal-weight mean over the eleven final dataset scores; Avg averages the seen-dataset mean at each stage. These definitions differ from KeepLoRA's published MTIL Average and Fed-Duet's sample-weighted log metrics.

Relative to the shared KeepLoRA trajectory, the CLIP extension improves Last by 2.59 points and Avg by 4.45 points, with the largest final gains on Caltech101, CIFAR100, and Flowers. Food, StanfordCars, and SUN397 decline, so the benefit is not uniform. The routed-expert control without head calibration also reaches Last 83.99; adding head calibration mainly lowers forgetting from 3.00 to 2.08 rather than improving that endpoint. Fed-Duet provides a completed federated reference on the same task sequence, but its lower macro score does not isolate an algorithmic advantage.

B.9 Communication and Storage Analysis

We examine whether communication conclusions depend on the accounting assumptions. Figure 6 counts model, projector, router, moments, and task-boundary gradient payloads under shared attention/FFN modules, BF16 model parameters, five participants, ten rounds per task, and version-aware caches. At rank 32, effective upload plus download is 149.46 GiB for FedAvg, 191.16 GiB for the conditional DISCO scenario, and 168.95 GiB for DART. At rank 8, the values are 60.67, 71.09, and 84.66 GiB. Hence relative cost depends on rank rather than uniformly favoring DART.

Persistent storage. We measure which retained components dominate storage as tasks accumulate, separately from transmitted payloads. We inspect tensor shapes and numerical formats in all eight task-boundary checkpoints of the $\beta = 1.0$, seed-42, rank-32 Hom-FCIT run. Figure 9(a) reports one checkpoint's logical tensor bytes at each boundary, excluding preinstalled backbone, vision encoder, and frozen routing-feature assets, rather than summing archived checkpoints. Saved state grows from 5.26 to 15.74 GiB. At task eight, the FP32 protection bases occupy 14.02 GiB (89.1%); saved LoRA factors occupy 1,372.50 MiB, including 1,220 MiB of historical A/B factors used to reconstruct merged weights and 152.50 MiB of current factors. Eight projector experts, the fixed anchor, and router occupy 360.30 MiB, of which the router accounts for only 0.16 MiB. The serialized checkpoint occupies 15.742 GiB, consistent with the 15.741 GiB tensor count up to serialization and metadata overhead. These historical LoRA factors are a resumption representation, not additional adapters all required during inference.



(a) Saved state across task boundaries.

(b) Eight-expert distribution memory.

Figure 9: Storage audit of DART. (a) Logical saved tensor bytes at each task boundary, excluding preinstalled models. (b) Eight expert distributions: actual low-rank storage versus calculated dense means and covariances (FP32). Dense storage is a representation reference, not a trained baseline. Note the different units.

To isolate the cost of distributional memory, each expert stores a mean, residual diagonal, covariance factor, and factor weights. With routing dimension $d = 5120$ and retained factor width q_k, its FP32 tensor budget is $4(2d + dq_k + q_k)$ bytes. The audited merged factors have $q_k = 165$ for all eight experts, giving 26.10 MiB in total. Figure 9(b) compares this with $4T(d^2 + d)$ bytes for dense means and covariances, or 800.16 MiB at $T = 8$: the stored representation is $30.66\times$ smaller. This comparison concerns storage, not equivalence to an empirical full covariance or matched task accuracy. The result supports compact routing memory, but not a claim that the complete method has negligible storage overhead: protection bases dominate server-side state. Neither these bases nor historical distribution statistics are needed by inference clients once the merged backbone, projector bank, and calibrated router are deployed.

Table 13: Reproduction settings for the two Hom-FCIT runs and the centralized UCIT comparison.

Setting	Eight-task Hom-FCIT	Four-stage Hom-FCIT	Six-task UCIT
Training regime	Federated continual SFT	Federated continual SFT	Centralized KeepLoRA SFT
Sequence	8 datasets; App. B.1	General, Other, Math, Chart (12 datasets)	ImageNet-R, ArxivQA, VizWiz, IconQA, CLEVR-Math, Flickr30k
Base model	LLaVA-1.5-7B; CLIP ViT-L/14-336 vision tower; two-layer GELU projector		
LoRA rank	32	32	32
Trainable SFT modules	LoRA B + current projector	LoRA B + current projector	LoRA B + projector
LoRA A during SFT	Fixed; orthogonal construction	Fixed; orthogonal construction	Fixed; KeepLoRA construction
Projector bank at evaluation	8 experts	4 experts	6 saved task projectors
Clients / selected per round	50 / 5	50 / 5	Not applicable
Partition	Released non-IID, $\beta = 1.0$	Released non-IID, $\beta = 1.0$	Centralized task training sets
Rounds per stage	10	10	Not applicable
Epoch budget	1 local epoch / round	1 local epoch / round	1 epoch / task
Effective SFT batch size	64 (microbatch 8, accumulation 8)	64 (microbatch 8, accumulation 8)	64 (configured: $2 \times 8 \times 4$)
Backbone learning rate	2×10^{-4}	2×10^{-4}	2×10^{-4} (T1–5); 10^{-4} (T6)
Projector configured LR[a]	2×10^{-5}	2×10^{-5}	2×10^{-5}
SFT schedule / precision	Cosine; warmup ratio 0.03; weight decay 0; BF16; maximum sequence length 2,048		
Training / calibration seed	42 / 42	42 / 42	42 / 42
Router descriptor	5,120 dimensions: mean visual features (1,024) + mean prompt embeddings (4,096)		
Router architecture	Linear $5120 \to 8$ + softmax	Linear $5120 \to 4$ + softmax	Linear $5120 \to 6$ + softmax
Statistics sample budget	Up to 128 / current client	Up to 128 / current client	Up to 128 / task training set
Local covariance model	Rank-32 + residual diagonal	Rank-32 + residual diagonal	Rank-32 + residual diagonal
Historical memory	Task-end aggregated moments	Task-end aggregated moments	Post-hoc moments from six training sets
Calibration timing	After each aggregation round	After each aggregation round	Once at final checkpoint
Synthetic pool	1,024 / seen expert	1,024 / seen expert	1,024 / expert
Calibration optimizer	AdamW; 400 steps; batch 256; LR 0.03; weight decay 10^{-4}; label smoothing 0.02		
Final test coverage	Complete: 61,310 records	Complete: 79,141 records	3,000 examples / task
Reported aggregates	Last; seen-task Avg	Last; seen-dataset Avg	KeepLoRA: Transfer, Average, Last; DART: Last only
Result table	Table 3	Table 4	Table 5

B.10 Qualitative Examples

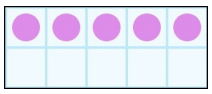
We inspect how calibration changes generated answers with the backbone and eight experts fixed. Figure 10 compares uniform $1/8$ mixing (zero router parameters) with DART. These selected cases show useful corrections and a remaining caption error, not their frequency.

(a) ImageNet-R



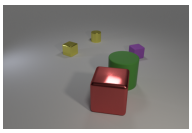
Input. Question: What is the object in the image?
Answer the question using a single word or phrase.
Reference. Hammerhead
Uncalibrated. Great white shark
DART. Hammerhead

(b) IconQA



Input. How many dots are on the frame?
Answer the question using a single word or phrase.
Reference. 5
Uncalibrated. 6
DART. 5

(c) CLEVR-Math



Input. Subtract all red blocks. How many blocks are left?
Answer the question using a single word or phrase.
Reference. 2
Uncalibrated. 3
DART. 2

(d) OCRVQA



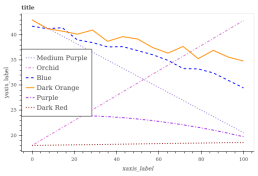
Input. What is the title of this book?
When the provided information is insufficient, respond with 'Unanswerable'.
Answer each question using a single word or phrase.
Reference. The Blacker the Ink: Constructions of Black Identity in Comics and Sequential Art
Uncalibrated. The Blacker the Ink: Constructions of Black Identity in Comics, Graphic Novels, and Reggae Music
DART. The Blacker the Ink: Constructions of Black Identity in Comics and Sequential Art

(e) Flickr30k



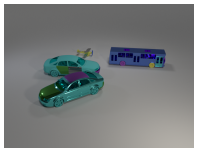
Input. What is happening in the image?
Generate a brief caption for the image.
Reference caption. A person in an orange jacket getting out of the rear of a Toyota SUV , parked in front of a view of the ocean and mountains
Uncalibrated. A man is standing next to a tan SUV with a surfboard on top of it.
DART. A man in an orange shirt is standing next to a tan van

(f) FigureQA



Input. Please answer the question and provide the final answer at the end.
Question: Which color intersects with both Blue and Dark Orange?
Reference. Orchid
Uncalibrated. Purple
DART. Orchid

(g) super-CLEVR



Input. Question: What number of other things are the same shape as the tiny matte thing?
Answer the question using a single word or phrase.
Reference. 0
Uncalibrated. 1
DART. 0

Figure 10: Selected examples with an uncalibrated router and DART on the formal eight-task benchmark. Panel (e) removes the surfboard hallucination but misidentifies the SUV as a van.