SHARED PROMPT CARRIERS WITH BOUNDED TASK RESIDUALS FOR FEDERATED CONTINUAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated continual learning (FCL) requires clients to learn from evolving and heterogeneous tasks while transferring knowledge without sharing data. Prompt-based adaptation is parameter-efficient, but fully shared prompts can cause task interference, whereas isolated prompts limit reuse of common structure. We introduce FedSCR, a prompt parameterization built on a shared carrier with bounded low-rank task residuals. The shared carrier remains trainable across clients, while each task retains a compact offset whose norm is bounded relative to the carrier norm captured at warmup. Experiments on ImageNet-R and DomainNet show competitive accuracy and transfer compared with representative federated continual learners under heterogeneous task arrivals. Analyses indicate that most of the accuracy observed in our experiments is already present in the shared carrier, while residuals provide compact task state for controlled retention. FedSCR offers a simple way to balance cross-client knowledge transfer and controlled task retention in FCL.

1 INTRODUCTION

Federated continual learning (FCL) extends federated learning to clients whose data distributions evolve as new tasks arrive. It therefore couples two learning axes: clients collaborate across spatially heterogeneous private data, while each client must retain knowledge across a temporal task stream. Unlike centralized continual learning, a client cannot inspect the other clients' examples; unlike standard federated learning, its local objective does not remain stationary. Server aggregation can transfer knowledge across clients, but the same update can disrupt representations used by earlier tasks. FCL must reconcile cross-client transfer with temporal retention under the storage and communication constraints of federated systems (Kairouz et al., 2021; McMahan et al., 2017; Shenaj et al., 2023; Yoon et al., 2021).

Prompt-based adaptation offers a rehearsal-free way to address this problem with a frozen pretrained vision transformer. Instead of updating the backbone, clients optimize a small prompt state that guides its representations. Prompt pools and decomposed prompt attention support sequential adaptation (Smith et al., 2023; Wang et al., 2022a,b); federated variants organize prompt exchange through contrastive learning, dual knowledge transfer, or class-aware interaction (Bagwe et al., 2023; Piao et al., 2024; Xu et al., 2025). Once these prompts are federated, however, their allocation determines how spatial transfer and temporal retention interact. A shared prompt lets every aggregation affect every task, whereas an isolated task prompt no longer directly follows the common state learned from later clients and tasks.

The question is therefore not simply whether prompts should be shared or specialized—prior work already combines both (Guo et al., 2026; Yoon et al., 2021; Yu et al., 2024). The unresolved design choice is how far a task may depart from shared knowledge, how long that departure should remain trainable, and who may update it. A low-rank factorization limits the number of adaptation directions but not their amplitude. Freezing a task component preserves its offset but does not specify how that offset should coexist with a shared component that continues to evolve.

FedSCR starts from a continually learned shared carrier and gives each federated task a small, fixed-budget residual. The carrier is the principal cross-client learning path, whereas the residual stores a constrained task offset rather than an independent expert. Each effective residual is initialized to zero, captures a carrier-relative scale after one warmup exposure, adapts during the next two exposures,

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.

and then is frozen during the regular task stream; an armed recovery event can reopen selected old rows; its rank and amplitude are controlled by construction. Explicit row ownership determines which client updates each residual in the evaluated schedule. A frozen residual remains coupled to the evolving carrier, so later shared updates can induce positive or negative transfer while the task-specific offset stays fixed. Prompt composition is based on an image-derived query. In the reported slot-wise evaluation, stored client–task metadata and predefined class masks select the corresponding output block.

Our contributions are:

- We develop a shared-carrier prompt parameterization that gives each task a residual of rank at most two, tied to a carrier scale captured once and bounded in amplitude throughout subsequent training.
- We specify a finite residual lifecycle and explicit row ownership, together with a bounded anchor-recovery update that can reopen frozen task rows after carrier drift while the shared carrier remains the shared learning path.
- We characterize the parameterization on ImageNet-R and DomainNet through seven-method comparisons, a carrier-only control, matched transfer, lifecycle diagnostics, and explicit parameter, communication, and runtime accounting.

We next position FedSCR among continual and federated prompt methods, define its update rules, and evaluate the resulting behavior under fixed federated task streams.

2 Related Work

2.1 Federated Continual Learning

Federated learning (FL) trains a shared model from decentralized data without moving examples to a server (Kairouz et al., 2021; McMahan et al., 2017). Statistical heterogeneity makes local updates both complementary and conflicting; common remedies use client-side regularization, server-side aggregation correction, or synthetic data (Li et al., 2020). Continual learning (CL) adds temporal nonstationarity: rehearsal-based methods retain examples or prototypes, whereas rehearsal-free methods use regularization, distillation, or parameter isolation to preserve earlier tasks (De Lange et al., 2022; Kirkpatrick et al., 2017; Li & Hoiem, 2016; Rebuffi et al., 2017).

Federated continual learning (FCL) combines these spatial and temporal sources of interference: clients receive private task streams, task transitions may be asynchronous, and the server aggregates updates from different histories. Existing FCL methods use sparse parameter transfer, global–local compensation, or asynchronous coordination (Dong et al., 2022; Shenaj et al., 2023; Yoon et al., 2021). These approaches improve transfer or stability, but often couple task-specific state to broader updates or to additional memory and communication mechanisms. A compact state attached to a task while a prompt remains shared is still an open design question.

2.2 Prompt-Based Adaptation for Federated Continual Learning

Prompt tuning adapts a frozen backbone through a small set of trainable tokens (Jia et al., 2022; Lester et al., 2021). L2P selects prompts from a pool, DualPrompt separates general and task-specific components, and CODA-Prompt composes prompts through attention (Smith et al., 2023; Wang et al., 2022a,b). These designs reduce trainable state but primarily address temporal interference in a single stream.

Federated prompt methods extend this idea by deciding which prompt information is exchanged across clients. Fed-CPrompt uses contrastive adaptation, while POWDER and C^2Prompt use prompt-based knowledge transfer and class-aware interaction (Bagwe et al., 2023; Piao et al., 2024; Xu et al., 2025). Shared prompts enable cross-client reuse but can mix incompatible task updates; isolated or class-aware state preserves specialization but introduces additional aggregation and retention choices. The open question is which state should remain shared and which should stay task-specific.

2.3 Shared, Specialized, and Low-Rank Adaptation

A complementary line of work controls the size and structure of client-adaptive state. FedMGP, FedDuet, and Co-LoRA combine shared and specialized prompts, expert pathways, or low-rank modules (Guo et al., 2026; Seo et al., 2026; Yu et al., 2024). LoRA reduces trainable parameters through factorized updates (Hu et al., 2022), and continual variants retain residual or shared/specialized adapters across tasks (He et al., 2025; Menabue et al., 2024). NB-LoRA and Fed-TaLoRA further study rank, magnitude, and post-aggregation correction; factorized aggregation may require alignment between client factors (Bian et al., 2025; Wang et al., 2025; Yu et al., 2025).

These methods emphasize different sources of efficiency: architectural heterogeneity, expert routing, factor capacity, or aggregation error. FedSCR instead captures a carrier-relative scale once, uses a finite number of adaptation exposures, and freezes low-rank factors during the regular task stream while bounding their offset. The resulting parameterization is intended for task-row retention in a fixed prompt backbone. We evaluate the resulting task-row state through slot-level accuracy and transfer, lifecycle measurements, and resource accounting under a fixed slot-aware FCL protocol.

3 FedSCR

3.1 Task streams and prompt state

Let $\mathcal{C}$ be a set of clients and r a global communication round. Client c trains on its current task data $\mathcal{D}_{c,r}$; task transitions need not occur simultaneously across clients. A *slot* is a pair (c, t) consisting of a client and its current local task. Each slot maps to one independent residual row. This row is a schedule index, not a semantic class or a task state shared across clients; overlapping physical classes therefore do not merge rows. We use "task row" and "residual row" for this slot-indexed state below. The server receives model states, not client examples.

We freeze an ImageNet-21K-pretrained ViT-B/16 (Dosovitskiy et al., 2021) and adapt its prompt path and classifier. For each prompt layer l, the path contains prompt values P, matching keys K, and attention vectors A, following decomposed prompt attention (Smith et al., 2023). Write $f \in \{P, K, A\}$ for a parameter family. At communication round r, FedSCR represents each task's effective family as

$$X_{i,l}^{f,r} = S_l^{f,r} + R_{i,l}^{f,r}, \qquad f \in \{P, K, A\}, \tag{1}$$

where $S_l^{f,r}$ is the shared *carrier* and $R_{i,l}^{f,r}$ is the task residual at round r. The carrier remains trainable across the stream. An unseen row has zero residual; a seen row changes only during its adaptation period in the regular task stream, unless an armed recovery event reopens it. Figure 1 summarizes the parameterization and training flow.

3.2 A fixed amplitude budget for each low-rank residual

For a family flattened to $m_f \times d$, define $D_{i,l}^f = U_{i,l}^f (V_{i,l}^f)^\top$, with $U_{i,l}^f \in \mathbb{R}^{m_f \times q}$ and $V_{i,l}^f \in \mathbb{R}^{d \times q}$. Here q is the rank budget and $d = 768$. With prompt-pool size M and prompt length L_p, $m_P = ML_p$ and $m_K = m_A = M$. A factorized family stores $q(m_f + d)$ parameters per task, compared with $m_f d$ for a dense residual.

A low-rank direction can still grow arbitrarily large. We therefore capture the carrier scale once, at the end of the task row's warmup exposure r_i^{w}, and map the direction into a fixed norm budget:

$$s_{i,l}^f = \mathrm{stopgrad}\left(\left\| S_l^f(r_i^{\mathrm{w}}) \right\|_F\right), \qquad R_{i,l}^f = \rho s_{i,l}^f \frac{D_{i,l}^f}{1 + \left\| D_{i,l}^f \right\|_F}. \tag{2}$$

The resulting rank and magnitude invariants are:

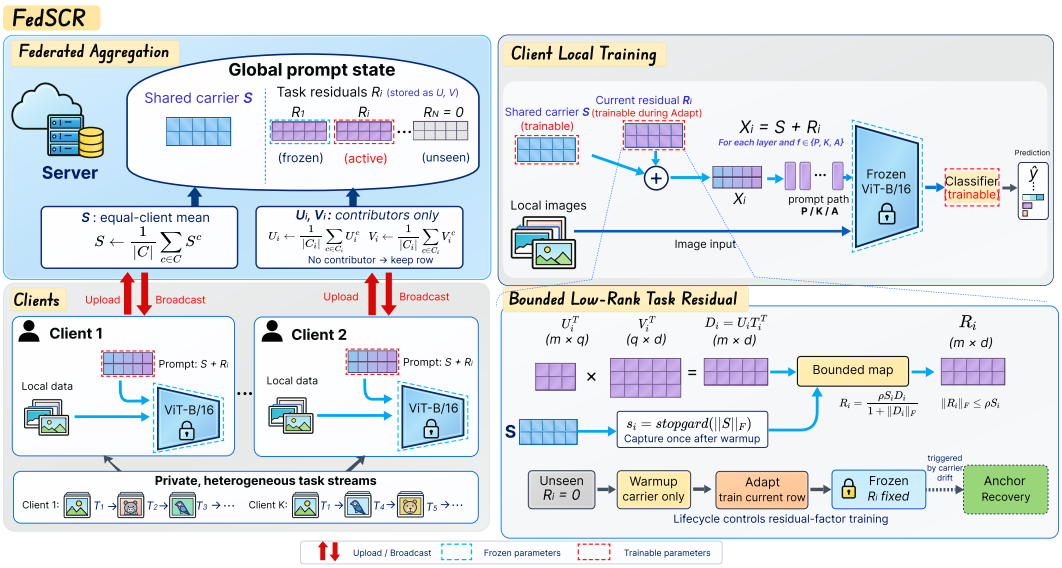


Figure 1: Overview of FedSCR. A trainable shared prompt carrier is combined with bounded low-rank residuals for each client–task slot. Residuals are captured after warmup, adapted for a finite period, and then frozen while the carrier continues to learn across clients.

Theorem 1 (Bounded low-rank residual). *For finite factors $U^f_{i,l}, V^f_{i,l}$ with $q \geq 1$, $s^f_{i,l} \geq 0$, and $\rho \geq 0$, the residual satisfies* $\operatorname{rank}(R^f_{i,l}) \leq q$ *and* $\left\|R^f_{i,l}\right\|_2 \leq \rho s^f_{i,l}$, *while*

$$\left\|R^f_{i,l}\right\|_F = \rho s^f_{i,l} \frac{\left\|D^f_{i,l}\right\|_F}{1 + \left\|D^f_{i,l}\right\|_F} \leq \rho s^f_{i,l}, \tag{3}$$

The proof of Theorem 1 is given in Appendix A. Because Eq. 1 gives $X^{f,r}_{i,l} - S^{f,r}_l = R^{f,r}_{i,l}$, the same bound controls the effective offset. The inequality is strict for positive $s^f_{i,l}$ and finite factors. The scale is held fixed thereafter: the bound is relative to the *captured* carrier, not the carrier at each future round. This prevents later carrier-norm changes from rescaling an otherwise frozen residual.

The parameterization controls residual rank and magnitude, but not earlier-task function stability; a function-level guarantee would require additional carrier-sensitivity assumptions.

The following proposition gives the parameter count for the residual store; its derivation is provided in Appendix A.

Proposition 1 (Residual-store parameter complexity). *Let N be the residual-row capacity and let L_{lay} be the number of adapted prompt layers. The factorized rank-q residual store contains*

$$P_{\text{fact}} = NL_{\text{lay}}q \sum_{f \in \{P,K,A\}} (m_f + d) = NL_{\text{lay}}q\big[M(L_p + 2) + 3d\big] \tag{4}$$

model scalars, whereas a dense residual store contains

$$P_{\text{dense}} = NL_{\text{lay}}d \sum_{f \in \{P,K,A\}} m_f = NL_{\text{lay}}dM(L_p + 2). \tag{5}$$

Their ratio is therefore

$$\frac{P_{\text{fact}}}{P_{\text{dense}}} = \frac{q\big[M(L_p + 2) + 3d\big]}{dM(L_p + 2)}. \tag{6}$$

The count covers residual factors only; the carrier, classifier, backbone, and optimizer states are excluded.

We initialize U randomly with a dedicated generator and V at exact zero. Thus $D = R = 0$ initially and Eq. 1 reduces exactly to the carrier. The zero factor allows gradients to begin changing V while preserving a common initial effective prompt. The dedicated generator avoids perturbing the data-order random stream when allocating residual factors.

3.3 Finite adaptation with continued sharing

Each task row keeps an exposure count; one exposure is a communication round in which the task is active, irrespective of its local epoch count. During the first exposure, the residual remains zero and local training updates the carrier and classifier. The server captures the row's scale after warmup. During its next two active exposures, the current residual factors are trainable at half the base learning rate. After these exposures, the factors are frozen during the regular task stream. Unseen and inactive rows receive no residual updates. If carrier movement satisfies the calibrated condition before a task revisit, the frozen state can be temporarily reopened by the anchor-recovery update described in Section 3.4.

3.4 Anchor recovery under carrier drift

The default lifecycle includes a server-side anchor-recovery branch when the carrier moves substantially before a previously exposed task is revisited. Let $\mathbf{S}^r$ denote the vector obtained by concatenating the shared carrier dictionary tensors (all prompt families and adapted layers) after server aggregation at round r. The drift statistic is

$$\delta_r = \frac{\left\|\mathbf{S}^r - \mathbf{S}^{r-1}\right\|_2}{\left\|\mathbf{S}^{r-1}\right\|_2}. \tag{7}$$

Residual factors, the classifier, and the frozen backbone are excluded from δ_r. The first two rounds are warmup observations. With a fixed threshold τ, recovery is armed only after two consecutive eligible rounds satisfy $\delta_r > \tau$. The trigger depends on parameter movement and is evaluated before a designated revisit boundary.

Let $\mathcal{I}_{\text{old}}$ denote the previously exposed rows, $\mathcal{F} = \{P, K, A\}$ the prompt families, and L_{lay} the adapted layers. Before the revisit, the server stores $\bar{X}^f_{i,l}$ as the effective prompt for each $i \in \mathcal{I}_{\text{old}}$, $f \in \mathcal{F}$, and $l = 1, \ldots, L_{\text{lay}}$. During recovery, $X^{f,\text{rec}}_{i,l}$ denotes the current prompt. The recovery update minimizes their mean squared deviation:

$$\mathcal{L}_{\text{anchor}} = \frac{1}{|\mathcal{I}_{\text{old}}| L_{\text{lay}} |\mathcal{F}|} \sum_{i \in \mathcal{I}_{\text{old}}} \sum_{l=1}^{L_{\text{lay}}} \sum_{f \in \mathcal{F}} \left\| X^{f,\text{rec}}_{i,l} - \bar{X}^f_{i,l} \right\|_F^2. \tag{8}$$

Only the residual factors of previously exposed rows are updated; the carrier, classifier, backbone, and newly arrived rows remain fixed. The same rank and carrier-relative magnitude constraints apply during this update. The server logs the trigger streak, anchor loss, changed rows, carrier delta, and maximum residual-to-captured-scale ratio, then clears the update mask and refreezes the updated rows. Thus the anchor-recovery update is a conditional parameter-space correction rather than an additional training stream.

3.5 Owner-bound updates and image-query readout

In the evaluated schedule, each active residual row has one designated owner. Shared floating-point state, including the carrier and classifier, follows equal-client aggregation. A residual row has at most one contributor in this schedule, so the server copies the owner's factors when the row is updated and leaves the row unchanged otherwise. This owner-bound update is used for ordinary client-side training; the recovery branch uses a separate server-side update. The one-round communication pseudocode is given in Appendix B, Algorithm 1.

Local training minimizes cross-entropy on current client data, with gradients restricted to shared trainable state, the classifier, and eligible residual factors. The evaluated configuration sets the inherited orthogonality penalty to zero and adds no distillation term. At each prompt layer, let $\mathcal{T}$ contain the available task rows and j index their M components. The inherited image-query readout

is

$$\widehat{P}(z) = \frac{1}{|\mathcal{T}|} \sum_{i \in \mathcal{T}} \sum_{j=1}^{M} \cos(z \odot X_{i,j}^{A}, X_{i,j}^{K})\, X_{i,j}^{P}. \tag{9}$$

with layer indices suppressed. Cosines are used directly, without softmax. The resulting tokens split into key and value prefixes for frozen transformer attention. Prompt composition is based on the image-derived query z. Evaluation additionally uses stored client–task metadata and predefined class masks to select the corresponding output block; physical class overlap follows the stored class map. We do not evaluate classification over a single global output space without this task information.

4 Experiments

4.1 Protocol and metrics

Task streams. We evaluate image classification on ImageNet-R (Hendrycks et al., 2021) and DomainNet (Peng et al., 2019). Each protocol has five clients and 15 *synchronous* rounds; two clients switch tasks every three rounds, yielding boundaries with 5, 7, 9, 11, and 13 arrived client–task slots. Arrivals are heterogeneous, but there are no delayed updates, dropout, or partial participation. ImageNet-R uses 20 classes per task and 10 local epochs; DomainNet uses 35 classes, four local epochs, and the painting, real, infograph, sketch, and clipart domains. The main tables use random heterogeneous construction. DomainNet uses a fixed-capacity 1,750-logit virtual classifier head and a 25-row residual store: under the 15-round map, only the first 25 classifier blocks (875 positions) are addressable, while the remaining 25 blocks are reserved and always placed in the complement (inactive) mask; they never contribute targets or predictions. The physical-class, residual-row, and classifier maps are given in Appendix B; a separate construction is reported in Appendix C.4.

Compared methods and implementation. We compare FedSCR with federated versions of L2P, DualPrompt, and CODA-Prompt (Fed-L2P, Fed-Dual, and Fed-CODAP), and with Fed-CPrompt, POWDER, and C^{2}Prompt (Bagwe et al., 2023; Piao et al., 2024; Smith et al., 2023; Wang et al., 2022a,b; Xu et al., 2025). All numbers come from runs under the corresponding shared dataset protocol. Methods use a frozen ImageNet-21K-pretrained ViT-B/16, 224×224 images, batch size 32, and base learning rate 0.005. Data splits, arrival schedules, and evaluation rules are matched within each dataset and seed. All experiments were run on A100 and RTX A6000 GPUs.

Configuration selection. FedSCR uses rank two, $\rho = 0.10$, residual learning-rate multiplier 0.5, one warmup exposure, two adaptations, freezing, and owner-bound updates. The standard main protocols follow the no-trigger path; the recovery branch is assessed in a separate controlled drift–revisit stress test. We selected this configuration on an 80/20 ImageNet-R training/validation split. Nine bounded rank–amplitude settings were screened, the two leading settings received four learning-rate evaluations, and unbounded/dense designs served as diagnostics. Validation AIA was the primary criterion; test data were not used. The selected configuration was then fixed for the three-seed evaluations. Compared methods use their released configurations.

Main efficacy tables report three-seed means with population SD. Design, training-side, and runtime measurements report means with sample SD over their stated repeats.

Evaluation metrics. Metrics are computed on client–task slots at the five observed boundaries $\{2, 5, 8, 11, 14\}$, using zero-based round indices. At each boundary, accuracy is averaged equally over the arrived slots; AIA is the equal average of these five boundary means. FAA is the equally weighted accuracy of the 13 arrived slots at the final boundary. Neither metric pools inactive classifier positions. Each slot is evaluated with its stored client–task block and predefined class mask described in Section 3.5. For each previously observed slot, forgetting is the drop from its best earlier observed boundary; new slots receive zero. FM applies the same slot-then-boundary averaging and can be negative when current accuracy exceeds the earlier maximum. For transfer, every method and seed is paired with a 15-round direct-reset control using the same task and arrival schedule. FT is the mean improvement at a slot's first evaluation boundary over its matched reset control; BT is the mean change from the first to the final boundary; CT is their sum. FT therefore measures transfer after initial task learning rather than zero-shot accuracy. Efficacy tables use population SD over

Table 1: Performance comparison on ImageNet-R and DomainNet under the two fixed heterogeneous task-arrival protocols. AIA summarizes accuracy across the observed task boundaries, while FM measures signed forgetting and FT, BT, and CT quantify forward, backward, and combined transfer.

Dataset	Method	AIA ↑	FM ↓	FT ↑	BT ↑	CT ↑
ImageNet-R	Fed-L2P	80.441±0.785	0.464±0.058	-1.125±0.337	-0.312±0.176	-1.436±0.374
	Fed-Dual	79.935±0.630	0.289±0.163	-1.046±0.124	0.087±0.298	-0.958±0.188
	Fed-CODAP	81.250±1.423	0.041±0.434	0.495±0.563	1.099±1.014	1.594±0.485
	Fed-CPrompt	77.772±1.020	0.301±0.242	-2.308±0.423	-0.203±0.393	-2.511±0.425
	POWDER	85.813±0.224	**-0.414±0.159**	0.233±0.405	**1.861±0.520**	2.094±0.621
	C^2Prompt	85.430±0.736	-0.157±0.153	0.091±0.223	1.038±0.329	1.129±0.511
	FedSCR	**86.841±0.436**	-0.388±0.086	**4.343±0.502**	1.397±0.307	**5.741±0.294**
DomainNet	Fed-L2P	73.559±0.150	0.188±0.010	-1.758±0.149	-0.175±0.121	-1.934±0.214
	Fed-Dual	73.407±0.095	0.109±0.119	-1.217±0.157	0.031±0.134	-1.186±0.112
	Fed-CODAP	74.320±0.111	0.186±0.155	-0.597±0.217	0.004±0.187	-0.593±0.063
	Fed-CPrompt	72.685±0.076	0.176±0.077	-2.127±0.145	-0.229±0.107	-2.357±0.164
	POWDER	77.142±0.056	0.078±0.022	0.029±0.145	0.114±0.223	0.144±0.202
	C^2Prompt	76.901±0.135	0.200±0.044	0.026±0.071	0.048±0.199	0.074±0.157
	FedSCR	**77.921±0.144**	**-0.188±0.078**	**1.744±0.025**	**0.645±0.295**	**2.388±0.271**

seeds 2021–2023, while design and repeat measurements use sample SD. The complete endpoint definitions and seed statistics are in Appendix D; per-seed AIA and FAA values are in Appendix C.1. All comparisons are descriptive three-run estimates.

4.2 Accuracy and Transfer Across the Task Stream

AIA comparison. FedSCR attains the highest mean AIA on both datasets (Table 1). It improves over POWDER by 1.03 percentage points on ImageNet-R and 0.78 points on DomainNet; the corresponding margins over C^2Prompt are 1.41 and 1.02 points. The consistent ordering across the two streams indicates that the shared-carrier design retains accuracy as new client–task slots arrive. Because AIA weights all five boundaries equally, these gains reflect performance across the task stream rather than only the final endpoint.

FM comparison. FedSCR obtains negative signed FM on both ImageNet-R and DomainNet, approximately -0.39 and -0.19, respectively. A negative value means that improvements on previously observed slots outweigh their measured forgetting under the boundary-averaged definition. On ImageNet-R, POWDER is slightly more favorable at approximately -0.41, whereas FedSCR has the lowest FM on DomainNet. FedSCR therefore avoids net degradation across both streams, although it does not dominate the forgetting measure on every dataset.

FT, BT, and CT comparison. FedSCR's clearest transfer advantage is forward transfer. Its FT reaches 4.34 points on ImageNet-R and 1.74 points on DomainNet, compared with 0.23 and 0.03 points for POWDER. BT remains positive at 1.40 and 0.65 points: POWDER is higher on ImageNet-R, while FedSCR is higher on DomainNet. Combining the two directions, FedSCR obtains the largest CT on both datasets, reaching 5.74 and 2.39 points versus 2.09 and 0.14 for POWDER. Since CT is defined as FT + BT, it summarizes rather than independently verifies the two transfer effects. Moreover, FT is measured against matched direct-reset controls and should not be interpreted as zero-shot accuracy. The concentration of the gain in FT is consistent with the shared carrier providing reusable state for later task acquisition, while backward transfer remains dataset dependent.

FAA comparison. FAA provides a complementary endpoint view. On ImageNet-R, FedSCR reaches 87.09% and differs from POWDER's 87.21% by only 0.12 points. On DomainNet, FedSCR reaches 78.01%, approximately 0.55 points above POWDER. Thus the higher AIA reflects stronger accuracy across the observed stream, whereas the final ImageNet-R endpoint is nearly tied. The complete FAA table is in Appendix C.2, and per-seed values are in Appendix C.1.

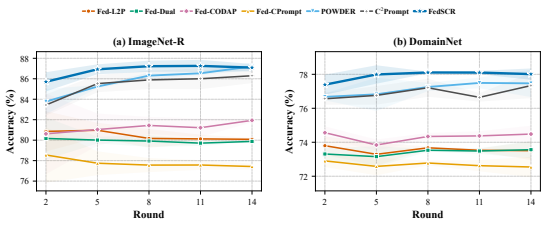


Figure 2: Boundary accuracy at observed checkpoints.

4.3 Ablation studies

Ablation comparison. Table 2 compares four design variants under the same task stream and evaluation protocol. The design comparison is independently executed from the main efficacy runs, so its aggregate can differ slightly. Shared carrier only removes task residuals, diluted aggregation applies inactive-row weighting, and dense residual replaces the bounded rank-two residual store with full-rank offsets. Full FedSCR reaches an AIA of 86.874%, compared with 86.784% for shared carrier only, a difference of 0.09 percentage points. The dense-residual variant uses approximately sixteen times more residual parameters but achieves lower AIA.

Table 2: Ablation of the FedSCR design on ImageNet-R. The full model is compared with shared-carrier-only, diluted-aggregation, and dense-residual variants under the same task construction and evaluation protocol.

Design	AIA ↑	FM ↓	FT ↑	BT ↑	CT ↑
Dense residual	85.455±0.725	**-0.802±0.211**	2.527±0.396	**2.597±0.712**	5.124±0.555
Shared carrier only	86.784±0.563	-0.361±0.122	4.394±0.582	1.357±0.378	5.751±0.508
Diluted aggregation	86.720±0.663	-0.384±0.149	4.284±0.513	1.409±0.313	5.693±0.471
Full FedSCR	**86.874±0.577**	-0.379±0.091	**4.464±0.640**	1.358±0.341	**5.822±0.450**

4.4 Controlled drift–revisit stress test

Recovery comparison. Table 3 reports the primary Frozen-residual versus Anchor-recovery comparison under a prescribed 10% carrier drift on ImageNet-R. Frozen residuals preserve the learned task offset, whereas Anchor recovery applies the bounded carrier-relative update before revisiting the task. Zero residuals are not included in this compact main table; they are the supplementary reference in Appendix C.3.

At 10% drift, Anchor recovery raises earlier-task accuracy from 81.908% to 83.236% (+1.328 points) and drift-task accuracy from 80.596% to 81.796% (+1.200 points). Relative to the Zero-residual, the earlier-task gain is 1.312 points. At 0% drift, the two designs coincide. The bounded update therefore changes the revisit result under the prescribed drift while remaining neutral without drift.

Table 3: Controlled drift–revisit stress test on ImageNet-R. Values are mean ± population SD over three seeds; accuracy is reported in percent.

Drift	Design	Earlier-task acc. ↑	Drift-task acc. ↑
0%	Frozen residuals	**86.826±0.629**	**85.131±0.419**
0%	Anchor recovery	**86.826±0.629**	**85.131±0.419**
10%	Frozen residuals	81.908±2.158	80.596±1.621
10%	Anchor recovery	**83.236±1.771**	**81.796±1.225**

4.5 Training dynamics

Carrier–residual evolution. Across train trajectories, the residual-to-captured-carrier ratio stays near 0.076, below $\rho = 0.10$, and the effective rank remains two (Figure 3). Five rows adapt initially; later only two new rows adapt while earlier rows remain frozen during the regular task stream. These measurements confirm that residual updates follow the specified norm bound and the configured freezing schedule.

The shared carrier continues to move after residual rows freeze. Its relative drift falls from about 0.43 to 0.16, while the gradient-weighted cosine $\cos(S \odot G, R \odot G)$ decreases from roughly 0.27 to 0.07. The update cosine remains positive, changing from about 0.34 to 0.24. Thus the carrier preserves a continuing shared-learning path, while the residual contribution remains a small, controlled offset whose interaction with the carrier changes over the stream.

These measurements describe how the carrier and residuals change during training; their accuracy implications are evaluated separately in the ablation study of Section 4.3. Full per-boundary values are in Appendix C.2, and definitions are in Appendix F.

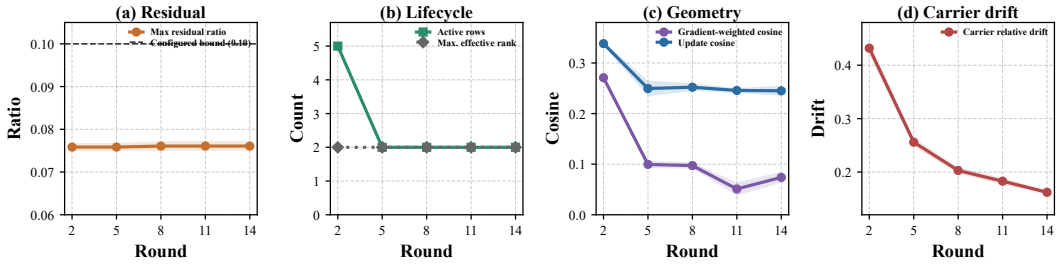


Figure 3: Training dynamics on ImageNet-R: residual ratio, lifecycle, gradient/update cosines, and carrier drift. Curves are three-seed means; bands show sample SD.

Table 4: Parameter, state, and execution overhead on ImageNet-R. Lower is better except throughput; timing is measured under the reported protocol.

Method	Trainable params	Client state total (MiB)	Checkpoint (MiB)	Wall time/run (s)	Throughput (samples/s)	Peak CUDA (MiB)
Fed-L2P	**1,352,180**	**1662.27**	**333.49**	4652.09±507.31	82.40±8.46	11360.08
Fed-Dual	1,609,460	1667.18	334.47	**4324.44±35.82**	**87.98±0.73**	**11303.13**
Fed-CODAP	9,984,500	1826.92	366.42	5281.98±771.75	72.99±9.83	12204.99
Fed-CPrompt	9,984,500	1826.92	366.42	8693.15±15.27	43.77±0.08	13085.01
POWDER	6,144,500	1753.68	351.77	6924.75±32.27	54.94±0.26	12825.02
C^2Prompt	6,144,500	1753.68	354.95	10459.15±126.95	36.38±0.44	12833.06
FedSCR	1,369,525	1662.61	333.58	5007.62±127.84	76.01±1.91	11587.25

4.6 Storage, communication, and runtime

Communication and state overhead. Table 4 reports trainable parameters, client state, checkpoint size, wall time, throughput, and peak CUDA allocation on ImageNet-R, using measurements from three repeats per method. Communication counts tensor payloads only and excludes protocol overhead, retransmission, compression, and network delay. FedSCR stores 0.60M residual scalars; 0.77M–1.37M parameters are eligible across lifecycle stages, and its 333.6 MiB checkpoint includes the frozen backbone. With a cached backbone it exchanges 26.1 MiB per direction per round (0.77 GiB over 15 rounds), close to Fed-L2P/Fed-Dual and 78%/86% below POWDER/C^2Prompt and Fed-CODAP/Fed-CPrompt. The active-trainable payload is 0.27 GiB versus 48.7 GiB for full-state copying, so the comparison concerns method-state exchange rather than a transport-level measurement.

Execution comparison. FedSCR completes 15 rounds in about 5,000 s, processes 76 samples/s, and peaks at 11.6 GiB CUDA memory. Wall time covers the full 15-round run, throughput counts processed training samples, and peak memory is per-process CUDA allocation. Its memory lies near Fed-L2P and Fed-Dual and below the larger prompt-state methods in this measurement. Runs were either isolated or concurrent, so timings are descriptive rather than a controlled speed comparison; Table 4 does not measure transport-level communication or inference latency.

5 Conclusion

We present FedSCR, a shared-carrier prompt parameterization with bounded task residuals for federated continual learning. A carrier-relative norm budget and finite lifecycle with owner-bound updates retain compact task offsets while the carrier shares knowledge across clients. Experiments on slot-aware ImageNet-R and DomainNet streams show competitive endpoint accuracy and transfer relative to representative federated continual learners. The controlled drift–revisit study further shows that anchor recovery can preserve earlier-task performance when carrier movement is present while remaining neutral when no recovery trigger is needed. Overall, FedSCR provides a compact and auditable state design for balancing cross-client knowledge transfer with controlled task retention.

Reproducibility statement

Section 3 specifies the residual map, initialization, lifecycle, and aggregation. Section 4.1 gives the task-stream configuration and evaluation protocol. Appendix B provides the classifier mapping and one-round pseudocode. Appendix C reports per-seed results, endpoint and design tables, the controlled drift–revisit results, the alternative DomainNet construction, and validation selection. Appendix D records metric details, Appendix E gives state and resource accounting, and Appendix F defines the training observations. Resource measurements are distinguished by their scope in Section 4.6.

AI use statement

The authors used generative AI tools to aid or polish the writing and to support retrieval and discovery, including finding related work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

References

Gaurav Bagwe, Xiaoyong Yuan, Miao Pan, and Lan Zhang. Fed-cprompt: Contrastive prompt for rehearsal-free federated continual learning. In *FL@ICML 2023 Workshop*, 2023. Workshop paper; arXiv:2307.04869.

Jieming Bian, Lei Wang, Letian Zhang, and Jie Xu. Lora-fair: Federated lora fine-tuning with aggregation and initialization refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3737–3746, 2025. Preprint arXiv:2411.14961.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3366–3385, 2022. doi: 10.1109/TPAMI.2021.3057446.

Jiahua Dong, Lixu Wang, Zhen Fang, Gan Sun, Shichao Xu, Xiao Wang, and Qi Zhu. Federated class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Mostafa Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. arXiv:2010.11929.

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. URL https://openreview.net/forum?id=Jk8g1OxyZY.

Jiangpeng He, Zhihao Duan, and Fengqing Zhu. CL-LoRA: Continual low-rank adaptation for rehearsal-free class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 30534–30544, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/He_CL-LoRA_Continual_Low-Rank_Adaptation_for_Rehearsal-Free_Class-Incremental_Learning_CVPR_2025_paper.html.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8340–8349, 2021. doi: 10.1109/ICCV48922.2021.00823.

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. arXiv:2106.09685.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *Computer Vision – ECCV 2022*, volume 13693 of *Lecture Notes in Computer Science*, pp. 709–727, 2022. doi: 10.1007/978-3-031-19827-4_41.

Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista A. Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascon, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaïd Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konecný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Ozgur, Rasmus Pagh, Hang Qi, Daniel Ramage, Ramesh Raskar, Mariana Raykova, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramer, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1–2): 1–210, 2021. doi: 10.1561/2200000083.

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.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, 2021. doi: 10.18653/v1/2021.emnlp-main.243.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems*, volume 2, pp. 429–450, 2020. URL https://proceedings.mlsys.org.

Zhizhong Li and Derek Hoiem. Learning without forgetting. In *Computer Vision – ECCV 2016*, volume 9908 of *Lecture Notes in Computer Science*, pp. 614–629, 2016. doi: 10.1007/978-3-319-46493-0_37.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282. PMLR, 2017. URL https://proceedings.mlr.press/v54/mcmahan17a.html.

Martin Menabue, Emanuele Frascaroli, Matteo Boschini, Enver Sangineto, Lorenzo Bonicelli, Angelo Porrello, and Simone Calderara. Semantic residual prompts for continual learning. In *Computer Vision – ECCV 2024*, 2024.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1406–1415, 2019. doi: 10.1109/ICCV.2019.00149.

Hongming Piao, Yichen Wu, Dapeng Wu, and Ying Wei. Federated continual learning via prompt-based dual knowledge transfer. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 40725–40739, 2024. URL https://proceedings.mlr.press/v235/piao24a.html.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2001–2010, 2017. doi: 10.1109/CVPR.2017.587.

Minhyuk Seo, Taeheon Kim, Hankook Lee, Jonghyun Choi, and Tinne Tuytelaars. Colora: Collaborative model personalization on heterogeneous multi-modal clients. In

C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 16450–16505, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/1baeb4973da3e78e523024030ea81cf5-Paper-Conference.pdf.

Donald Shenaj, Marco Toldo, Alberto Rigon, and Pietro Zanuttigh. Asynchronous federated continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 5055–5063, 2023. doi: 10.1109/CVPRW59228.2023.00534.

James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. CODA-Prompt: COntinual decomposed attention-based prompting for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Ruigang Wang, Krishnamurthy Dvijotham, and Ian R. Manchester. Norm-bounded low-rank adaptation. *arXiv preprint arXiv:2501.19050*, 2025. doi: 10.48550/ARXIV.2501.19050. URL https://arxiv.org/abs/2501.19050.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. DualPrompt: Complementary prompting for rehearsal-free continual learning. In *Computer Vision – ECCV 2022*, 2022a.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 139–149, 2022b.

Kunlun Xu, Yibo Feng, Jiangmeng Li, Yongsheng Qi, and Jiahuan Zhou. C^2prompt: Class-aware client knowledge interaction for federated continual learning. In *Advances in Neural Information Processing Systems 38*, 2025. doi: 10.52202/085713-1470.

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 of *Proceedings of Machine Learning Research*, pp. 12073–12086, 2021. Preprint arXiv:2003.03196.

Feng Yu, Jia Hu, and Geyong Min. Task-agnostic low-rank residual adaptation for efficient federated continual fine-tuning. *arXiv preprint arXiv:2505.12318*, 2025. URL https://arxiv.org/abs/2505.12318v2. Version 2, revised April 2026.

Hao Yu, Xin Yang, Xin Gao, Yan Kang, Hao Wang, Junbo Zhang, and Tianrui Li. Personalized federated continual learning via multi-granularity prompt. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024. doi: 10.1145/3637528.3671948. Preprint arXiv:2407.00113.

A Proofs and derivations of the main-text results

Proof of Theorem 1. Because $D^f_{i,l} = U^f_{i,l}(V^f_{i,l})^\top$, its rank is at most q; multiplying by a scalar cannot increase rank. Let $a = \left\|D^f_{i,l}\right\|_F \geq 0$. The residual definition above gives

$$\left\|R^f_{i,l}\right\|_F = \rho s^f_{i,l}\frac{a}{1+a} \leq \rho s^f_{i,l}.$$

The spectral norm is bounded by the Frobenius norm, and $X^{f,r}_{i,l} - S^{f,r}_l = R^f_{i,l}$, which proves the remaining claims.

Derivation of Proposition 1. For one layer and family f, the factorized representation stores $q(m_f + d)$ scalars. Summing over the three families, L_{lay} layers, and N residual rows gives the first expression stated in the proposition. A dense residual stores $m_f d$ scalars per family, which gives the second expression. Substituting $m_P = ML_p$ and $m_K = m_A = M$ yields the displayed closed forms and their ratio.

B MAIN-PROTOCOL TASK AND CLASSIFIER MAPPING

The main DomainNet protocol uses five clients, 15 synchronous rounds, and a task transition cadence of three rounds. It therefore has five potential local task indices per client and a residual-row capacity of 25. Three indices are kept separate throughout the implementation: the client-major classifier block index $u(c,t) = 5c + t$, the task-major residual state row $\mathrm{row}(c,t) = 5t + c$, and the classifier position. The symbol r without arguments continues to denote a global communication round.

$$\begin{aligned} u(c,t) &= 5c + t, & \mathrm{row}(c,t) &= 5t + c, \\ v(c,t,k) &= 35u(c,t) + k, & c &\in \{0,\dots,4\}, \\ & & t &\in \{0,\dots,4\}, \qquad k \in \{0,\dots,34\}. \end{aligned} \tag{10}$$

For completeness, the fixed head capacity has an independent block index $g \in \{0,\dots,49\}$ and head position $v_{\mathrm{head}}(g,k) = 35g + k$. The reported task map uses only $g = u(c,t) \in \{0,\dots,24\}$; hence $g = 25,\dots,49$ are reserved head blocks, not additional task rows or a second residual store. The physical class set $M_{c,t} \subset \{0,\dots,344\}$ has 35 IDs. It is generated with the run seed in classifier-block order u: the first block samples 35 distinct IDs; each later block samples an earlier block uniformly, chooses an overlap $h \in \{0,\dots,35\}$, copies h donor IDs, and samples the remaining $35 - h$ IDs from the donor complement. Thus overlap is random and reproducible, rather than a fixed semantic class partition. Overlap between two physical sets $M_{c,t}$ does not merge their residual rows. For the evaluated protocol, the preallocated-to-runtime row map is one-to-one, $\mathrm{row}_{\mathrm{runtime}}(j) = j$ for every active row; no row-coalescing step is applied. Table 5 summarizes these objects. At the five evaluation boundaries, 5, 7, 9, 11, and 13 client–task slots have arrived, and the one-to-one residual map gives the same number of active state rows. The residual-row store therefore contains $25 \times 35 = 875$ potential row–class positions.

The runtime classifier head is intentionally allocated with 50 blocks, or $50 \times 35 = 1{,}750$ logits, for compatibility with the runtime head and saved checkpoints. The 15-round task map addresses only blocks $u \in \{0,\dots,24\}$, so Eq. 10 produces positions $v \in \{0,\dots,874\}$. Blocks 25–49 (positions 875–1749) are reserved capacity: they are included in every complement mask and are never targets or active predictions in the reported runs. Equivalently, if $\mathcal{I}_{\mathrm{act}}$ is the set of addressed positions, the complement is $\mathcal{I}_{\mathrm{comp}} = \{0,\dots,1749\} \setminus \mathcal{I}_{\mathrm{act}}$. During optimization, logits and labels are restricted to the addressed class list before cross-entropy, so the reserved tail receives no target gradients.

At the final boundary, $13 \times 35 = 455$ positions are active; inactive positions are excluded from AIA and FAA. For slot-wise evaluation, the stored client–task slot selects its classifier block and predefined class mask; overlapping physical classes remain separate virtual outputs. Client domains follow the fixed order painting, real, infograph, sketch, and clipart; quickdraw is excluded. For the main overlap-based construction, classifier block $u(c,t)$ uses the loader interval $[0.02u, 0.02(u+1))$; the separately reported full-domain construction uses each client's full domain interval.

Table 5: Distinct indices and classifier capacity in the main DomainNet protocol. The physical class set may overlap across client–task pairs, but that overlap does not create a shared residual state row.

Object	Definition	Role
Physical class set	$M_{c,t} \subset \{0,\dots,344\}, \lvert M_{c,t}\rvert = 35$	Seeded DomainNet IDs used by client c at local task t; overlap is allowed.
Classifier block	$u(c,t) = 5c + t$	Client-major virtual block used to place the 35 class outputs.
Residual state row	$\mathrm{row}(c,t) = 5t + c$	Task-major client-owned row for the prompt residual; the preallocated-to-runtime map is one-to-one.
Classifier position	$v(c,t,k) = 35u(c,t) + k$	Position of class offset k for an addressed client–task block.
Classifier head capacity	$50 \times 35 = 1{,}750$ logits	Fixed runtime tensor shape; blocks 25–49 are reserved and excluded by the 15-round complement mask.
Head position (reserved included)	$v_{\mathrm{head}}(g,k) = 35g + k,\ g \in \{0,\dots,49\}$	Complete 1,750-logit address space; only $g = u(c,t) \in \{0,\dots,24\}$ is task-addressed.
Addressable classifier range	$u \in \{0,\dots,24\}, v \in \{0,\dots,874\}$	The 25 blocks addressable by the reported task stream; at the final boundary only 13 blocks are active (455 logits).

The same schedule makes the participation convention explicit. All five clients upload after every round. At the first task boundary all clients start the initial task; at each later boundary two clients

Algorithm 1 One communication round of FedSCR

Require: Shared state S^r, factors $\{U_i^r, V_i^r\}$, scales, exposure counts, and anchor metadata
1: Broadcast global state to participating clients $\mathcal{C}_r$
2: **for** each client $c \in \mathcal{C}_r$ **do**
3: Identify its current task row and exposure stage from training metadata
4: Train shared state and classifier; train the current residual only during adaptation
5: Return local state and the set of residual rows updated
6: **end for**
7: Average shared state across $\mathcal{C}_r$ and apply owner updates to changed rows
8: Capture scales for rows completing warmup; advance exposure counts
9: Freeze rows completing their second adaptation; retain all other inactive rows
10: Observe carrier drift and update the consecutive-exceedance streak
11: At the stored anchor point, snapshot effective prompt tensors for previously exposed rows
12: At revisit, if the movement condition is met, optimize the stored-row factors with the anchor objective; otherwise skip
13: Clear the update mask and freeze the updated rows

advance to a new task while the other clients retain their current task. This creates heterogeneous task arrivals without introducing server-side stale updates, delayed messages, dropout, or partial participation. Because the residual row is client-owned by $\mathrm{row}(c, t)$, every nonempty row has one owner in this protocol. The evaluated protocol therefore does not include multi-client updates to the same residual row.

Table 6: Residual-row ownership under the evaluated five-client DomainNet protocol. A row–round group is counted when at least one client updates that residual row. Every nonempty group has one owner. The direct-reset control does not use residual updates.

Mode	Runs	Rounds/run	Nonempty row–round groups	Count $= 1$	Count ≥ 2
Learn	3	15	78	78	0
Direct-reset	3	15	N/A (no residual updates)		

Extension to multiple contributors. For schedules in which more than one client updates the same residual row, let $\mathcal{C}_r$ be the participating clients, $\mathcal{C}_i^r \subseteq \mathcal{C}_r$ the contributors to row i, and $\theta_i = (U_i, V_i)$. Suppressing layer and family indices, the extension uses

$$S^{r+1} = \frac{1}{|\mathcal{C}_r|} \sum_{c \in \mathcal{C}_r} S_c^{r+1}, \qquad \theta_i^{r+1} = \begin{cases} \dfrac{1}{|\mathcal{C}_i^r|} \displaystyle\sum_{c \in \mathcal{C}_i^r} \theta_{c,i}^{r+1}, & |\mathcal{C}_i^r| > 0, \\ \theta_i^r, & |\mathcal{C}_i^r| = 0. \end{cases} \tag{11}$$

It averages U and V separately and therefore generally differs from averaging reconstructed residual matrices; common initialization fixes the operational factor gauge. The bounded map is applied after reconstruction, preserving the residual norm budget. This extension is not evaluated in the reported experiments because every nonempty row–round group has one contributor.

C Additional experimental results

Within each reported table, methods use the same frozen backbone, data split, task map, round schedule, batch size, and seed set. The endpoint table records the main efficacy run family, whereas the design table records a separately executed three-seed design-comparison family under the same construction and evaluation definitions. Baselines use their released prompt dimensions and optimizer settings. Per-seed results are reported below; uncertainty estimates over three seeds are descriptive.

Table 7: Per-seed AIA and FAA for the two main protocols. Each cell reports the boundary-averaged and final-average accuracy for one method and seed.

Dataset	Method	Seed 2021	Seed 2022	Seed 2023
ImageNet-R				
	Fed-L2P	80.488/80.055	79.458/79.947	81.378/80.251
	Fed-Dual	79.805/79.882	79.237/79.806	80.764/79.940
	Fed-CODAP	81.586/81.820	79.363/81.785	82.800/82.197
	Fed-CPrompt	78.115/77.585	76.387/77.035	78.813/77.655
	POWDER	85.773/87.222	85.561/**87.648**	86.105/86.759
	C^2Prompt	85.790/86.751	84.405/85.282	86.096/86.854
	FedSCR	**87.284/87.477**	**86.248**/86.602	**86.991/87.190**
DomainNet				
	Fed-L2P	73.350/73.940	73.629/73.851	73.696/72.721
	Fed-Dual	73.279/73.990	73.505/73.849	73.439/72.847
	Fed-CODAP	74.329/74.914	74.451/74.743	74.179/73.787
	Fed-CPrompt	72.686/73.045	72.777/72.896	72.592/71.691
	POWDER	77.209/**78.453**	77.145/77.476	77.072/76.475
	C^2Prompt	77.048/78.228	76.721/77.365	76.934/76.415
	FedSCR	**77.796**/78.108	**77.844/78.365**	**78.122/77.566**

Table 8: Boundary-averaged and final-average accuracy under the two main protocols. AIA averages the five observed task boundaries, whereas FAA averages the arrived slots at the final boundary. Both metrics are computed from the same efficacy runs.

Dataset	Method	AIA ↑	FAA ↑
ImageNet-R	Fed-L2P	80.441±0.785	80.084±0.126
	Fed-Dual	79.935±0.630	79.876±0.055
	Fed-CODAP	81.250±1.423	81.934±0.187
	Fed-CPrompt	77.772±1.020	77.425±0.278
	POWDER	85.813±0.224	**87.210±0.363**
	C^2Prompt	85.430±0.736	86.295±0.718
	FedSCR	**86.841±0.436**	87.090±0.364
DomainNet	Fed-L2P	73.559±0.150	73.504±0.555
	Fed-Dual	73.407±0.095	73.562±0.509
	Fed-CODAP	74.320±0.111	74.481±0.496
	Fed-CPrompt	72.685±0.076	72.544±0.606
	POWDER	77.142±0.056	77.468±0.807
	C^2Prompt	76.901±0.135	77.336±0.741
	FedSCR	**77.921±0.144**	**78.013±0.333**

Table 9: Extended design comparison on ImageNet-R. The table compares the full design with lifecycle, aggregation, and residual-capacity variants under the same task construction, schedule, and metric definitions as the main efficacy table; the design comparison is a separate three-seed run family.

Design	AIA ↑	FM ↓	FT ↑	BT ↑	CT ↑
Shared carrier only (no task residual)	86.784±0.563	-0.361±0.122	4.394±0.582	1.357±0.378	5.751±0.508
Unbounded rank-one residual	86.744±0.729	-0.347±0.208	4.353±0.625	1.374±0.582	5.727±0.691
Dense residual (full-rank offsets)	85.455±0.725	**-0.802±0.211**	2.527±0.396	**2.597±0.712**	5.124±0.555
No warmup exposure	86.791±0.619	-0.432±0.148	4.154±0.705	1.486±0.366	5.640±0.619
Diluted owner update	86.720±0.663	-0.384±0.149	4.284±0.513	1.409±0.313	5.693±0.471
Full FedSCR (bounded rank-two residuals)	**86.874±0.577**	-0.379±0.091	**4.464±0.640**	1.358±0.341	**5.822±0.450**

C.1 PER-SEED ACCURACY

C.2 ENDPOINT AND DESIGN TABLES

The paired view below uses the same seed and task stream for the full design and each arm. Differences are reported as full FedSCR minus the corresponding arm, so positive values favor the full design. The intervals are descriptive paired t intervals with $n = 3$ and are not significance claims.

Table 10: Paired AIA differences between full FedSCR and the design variants on ImageNet-R. Each difference uses the same seed and task stream, with positive values favoring the full design; the final column gives a descriptive paired interval.

Design	Per-seed difference (2021 / 2022 / 2023)	Mean	SD	95% paired interval
Shared carrier only	$+0.211/+0.110/-0.049$	$+0.091$	0.131	$[-0.236, +0.417]$
Unbounded rank-one	$-0.062/+0.258/+0.195$	$+0.130$	0.170	$[-0.291, +0.552]$
Dense residual	$+1.572/+1.628/+1.059$	$\mathbf{+1.419}$	0.314	$[+0.640, +2.199]$
No warmup	$+0.130/+0.160/-0.040$	$+0.083$	0.108	$[-0.185, +0.352]$
Diluted owner	$+0.092/+0.254/+0.117$	$+0.154$	0.088	$[-0.064, +0.372]$

Table 11: Training dynamics of FedSCR on ImageNet-R across the five observed task boundaries. The measurements track residual scale relative to the captured carrier, update alignment, and carrier drift during the specified residual lifecycle.

Round	Max. ratio	Grad.-weighted cosine	Update cosine	Carrier drift
2	0.075847 ± 0.000934	0.270830 ± 0.002241	0.338217 ± 0.004064	0.431937 ± 0.010956
5	0.075847 ± 0.000934	0.099493 ± 0.007344	0.249515 ± 0.015580	0.255937 ± 0.008140
8	0.076082 ± 0.001165	0.097264 ± 0.006127	0.252051 ± 0.007323	0.202932 ± 0.005815
11	0.076082 ± 0.001165	0.051005 ± 0.012732	0.245738 ± 0.004856	0.182867 ± 0.006395
14	0.076082 ± 0.001165	0.073929 ± 0.011014	0.244979 ± 0.010124	0.162096 ± 0.004148

C.3 CONTROLLED DRIFT–REVISIT ANCHOR-UPDATE RESULTS

This controlled ImageNet-R experiment applies a prescribed earlier-task-gradient-aligned carrier drift before revisiting that task. The Anchor recovery design directly applies the bounded update, while Frozen residuals preserve the task offset and Zero residuals remove it. Table 12 reports per-seed earlier-task accuracy, and Table 13 reports the three-design aggregates. The comparison is scoped to this constructed drift–revisit setting and complements the standard main protocol.

C.4 ALTERNATIVE DOMAINNET TASK CONSTRUCTION

The main manuscript uses random heterogeneous construction. Table 14 reports the alternative task construction as a separate experiment. It uses 35 classes per task, five clients, 15 learning rounds, and a task-row capacity of 25. Training was performed on RTX A6000 GPUs. These results do not replace or combine with the main DomainNet table.

FedSCR reaches 76.175% mean AIA in this construction, compared with 74.098% for POWDER. Its mean FT and BT are both positive, giving CT of 4.862 points. Fed-CODAP has a lower mean AIA and substantially larger seed dispersion, especially in transfer. Those observations belong to this construction and are not used to estimate a pooled method advantage across protocols.

C.5 VALIDATION SELECTION

The validation split contains 20% of the original ImageNet-R training data; the remaining 80% is used for fitting search candidates. Seed 2021 determines this split. The first stage evaluates rank $q \in \{1, 2, 4\}$ and bound $\rho \in \{0.05, 0.10, 0.20\}$ with residual learning-rate multiplier one and the one-warmup, two-adaptation lifecycle. Unbounded rank-one and dense residuals are diagnostics

Table 12: Per-seed earlier-task accuracy for the controlled drift–revisit stress test. Values are percentages; the final column is anchor minus zero residual.

Drift	Seed	Zero	Frozen	Anchor	Anchor–Zero
0%	2021	86.299	86.278	86.278	-0.021
0%	2022	86.491	86.493	86.493	+0.001
0%	2023	87.695	87.706	87.706	+0.011
10%	2021	80.204	80.167	81.881	+1.677
10%	2022	80.619	80.608	82.089	+1.470
10%	2023	84.947	84.949	85.737	+0.790

Table 13: Controlled drift–revisit stress test on ImageNet-R. Values are mean ± population SD over seeds 2021, 2022, and 2023. Accuracy is reported in percent; lower earlier-task loss is better.

Drift	Design	Earlier-task acc. ↑	Drift-task acc. ↑	Earlier-task loss ↓
0%	Zero residuals	86.829±0.618	85.138±0.428	0.5078±0.0263
0%	Frozen residuals	86.826±0.629	85.131±0.419	0.5077±0.0265
0%	Anchor recovery	86.826±0.629	85.131±0.419	0.5077±0.0265
10%	Zero residuals	81.923±2.145	80.579±1.606	0.6769±0.0921
10%	Frozen residuals	81.908±2.158	80.596±1.621	0.6776±0.0918
10%	Anchor recovery	**83.236±1.771**	**81.796±1.225**	**0.6285±0.0761**

rather than eligible main-method candidates. The carrier-movement condition is calibrated separately from this accuracy search and fixed before the controlled stress test; it is not tuned on the test labels.

The two leading eligible structures advance to the learning-rate refinement. Rank two, $\rho = 0.10$, multiplier 0.5 gives validation AIA 85.405210%, compared with 85.402922% for rank four, $\rho = 0.05$, multiplier 2.0. Their difference is 0.002288 points. The ranking is determined by the pre-established AIA-first rule; it does not imply a statistically resolved difference. The selected configuration is then fixed for the main evaluations. The alternative no-warmup lifecycle belongs to the separate three-seed design comparison.

D METRIC DEFINITIONS AND WEIGHTING

Slot and boundary averaging. A slot identifies a client and an arrived task. At each of the five zero-indexed boundaries $b \in \{2, 5, 8, 11, 14\}$, accuracy is averaged equally over the slots observed so far. The slot counts are 5, 7, 9, 11, and 13. AIA averages these five boundary means, giving every boundary the same weight. This differs both from final-boundary accuracy and from pooling all slot observations into one mean, which would give greater weight to later boundaries. The final-average accuracy (FAA) used in the endpoint table is

$$\mathrm{FAA} = \frac{1}{|\mathcal{I}_{14}|} \sum_{i \in \mathcal{I}_{14}} a_{14,i}, \qquad |\mathcal{I}_{14}| = 13. \tag{12}$$

Table 14: DomainNet results under the alternative task construction. Each task contains 35 classes, with 15 learning rounds and a task-row capacity of 25.

Method	AIA ↑	FM ↓	FT ↑	BT ↑	CT ↑
Fed-L2P	71.567±0.782	0.335±0.129	0.040±0.392	-0.212±0.293	-0.172±0.668
Fed-Dual	70.893±0.913	0.668±0.107	-0.047±0.490	-0.684±0.174	-0.731±0.510
Fed-CODAP	56.405±3.778	2.366±1.295	-15.915±3.812	-1.389±10.563	-17.303±12.352
Fed-CPrompt	68.857±0.468	0.375±0.049	-0.690±0.302	-0.435±0.061	-1.125±0.336
POWDER	74.098±0.353	0.769±0.228	0.179±0.516	-1.006±0.351	-0.827±0.210
C^2Prompt	74.002±0.074	0.731±0.242	-0.035±0.228	-0.673±0.561	-0.708±0.467
FedSCR	**76.175±0.362**	**-0.252±0.050**	**3.973±0.151**	**0.889±0.190**	**4.862±0.092**

Table 15: Validation structural screen on the ImageNet-R training/validation split, seed 2021. The residual learning-rate multiplier is one. Only bounded candidates participate in method selection; unbounded and dense rows are diagnostic controls. Parameter counts describe the allocated residual store.

Form	Rank	ρ	Residual params	AIA ↑	FM ↓	BT ↑
Bounded	1	0.05	300,500	85.100719	-0.495244	1.753846
Bounded	1	0.10	300,500	85.348323	-0.657994	2.169231
Bounded	1	0.20	300,500	85.103723	-0.524696	1.847692
Bounded	2	0.05	601,000	85.294177	-0.706946	**2.381538**
Bounded	2	0.10	601,000	**85.394921**	-0.636550	2.151538
Bounded	2	0.20	601,000	85.177026	-0.546542	1.912308
Bounded	4	0.05	1,202,000	85.368706	**-0.722093**	2.305385
Bounded	4	0.10	1,202,000	85.228658	-0.577995	2.155385
Bounded	4	0.20	1,202,000	85.310658	-0.655603	2.123077
Unbounded	1	–	300,500	85.369473	-0.560077	1.898462
Dense	full	–	9,600,000	83.409364	-0.958522	3.293077

Table 16: Validation learning-rate refinement for the two selected bounded structures. The highest validation AIA selects rank two, $\rho = 0.10$, and learning-rate multiplier 0.5. The alternate lifecycle is evaluated separately in the design comparison.

Rank	ρ	LR multiplier	AIA ↑	FM ↓	BT ↑
2	0.10	0.5	**85.405210**	-0.543959	1.961538
2	0.10	2.0	85.266216	-0.555257	1.995385
4	0.05	0.5	85.303368	**-0.734601**	**2.616154**
4	0.05	2.0	85.402922	-0.623230	2.252308

Signed forgetting. Let $\mathcal{H}_{b,i}$ contain earlier observed boundaries for slot i. Its forgetting at b is

$$f_{b,i} = \begin{cases} \max_{b' \in \mathcal{H}_{b,i}} a_{b',i} - a_{b,i}, & \mathcal{H}_{b,i} \neq \varnothing, \\ 0, & \mathcal{H}_{b,i} = \varnothing. \end{cases} \tag{13}$$

FM averages $f_{b,i}$ over slots and then boundaries. Including the current boundary in the maximum would change the metric and force nonnegative values; it is excluded here. A negative FM therefore records improvement beyond earlier observed accuracies. Newly observed slots contribute zero to the boundary mean.

Direct-reset transfer. Each continual-learning method/seed pair has a separate matched reset-control run. For every arriving slot, the first-boundary continual accuracy is paired with that slot's reset accuracy at the same boundary. The schedule and task mapping are shared between the pair. Forward transfer is the mean of these differences after initial task learning; backward transfer compares final accuracy with first-boundary accuracy; combined transfer compares final accuracy with the reset control. The same 13 slots enter all three averages, so CT equals FT plus BT before rounding. This definition is not a zero-shot forward-transfer score.

Seed statistics. Compute each metric within a seed before averaging seeds. Main and alternative DomainNet results use population SD, $\sqrt{\sum_{j=1}^{3}(x_j - \bar{x})^2/3}$. The design comparison and repeat-based measurements use sample SD, $\sqrt{\sum_{j=1}^{3}(x_j - \bar{x})^2/2}$. Displayed CT can differ from the sum of independently rounded FT and BT by the rounding unit. The transfer metrics use the first, last, and reset accuracies at the specified boundaries.

E STATE, LIFECYCLE, AND RESOURCE ACCOUNTING

Prompt dimensions and optimization. The shared prompt contains ten components of length eight at each of the first five transformer layers (zero-indexed layers 0–4). Keys and attention vectors have width 768. Rank two therefore requires 24,040 residual scalars per task row across the five layers; the 25-row residual store contains 601,000 residual scalars. Each U column is initialized from a Gaussian vector and normalized; V starts at zero. Local training uses Adam with $\beta_1 = 0.9$, $\beta_2 = 0.999$, no weight decay, and a cosine schedule with period 50, reinitialized for each local training call. The carrier/classifier base rate is 0.005; adapting residual factors use multiplier 0.5. The orthogonality coefficient is zero, so the evaluated objective is cross-entropy.

Task state. Each layer/family stores shared carrier parameters and per-row residual factors, a captured carrier norm, and lifecycle metadata. Task metadata determines which row can receive a training update. Available effective prompt rows enter the inherited image-query-based prompt composition at inference. No additional task-identification network is trained for the residual mechanism.

Anchor update. The server concatenates the shared carrier tensors after aggregation and computes $\delta_r = \|\mathbf{S}^r - \mathbf{S}^{r-1}\|_2 / \|\mathbf{S}^{r-1}\|_2$. When the movement condition is met, it snapshots the effective P, K, and A tensors for the previously exposed task rows and updates only their U, V factors with the anchor objective from the main text. The carrier, classifier, backbone, and newly arrived rows are unchanged. The snapshot is server-side parameter memory and can be released or refreshed after the update.

Residual size. For a family of shape $m_f \times d$, a dense residual needs $m_f d$ scalars per row, while rank q needs $q(m_f + d)$. Both therefore grow with the number of retained task rows. The implementation preallocates a finite residual-row store; a maximum store size and a maximum number of trainable scalar entries are distinct notions. A row can be present in the store while having zero residual or frozen factors. Aggregation acts on U and V separately, then reconstructs the bounded residual.

Scope of the norm bound. The residual budget is ρ times the carrier norm captured at warmup. This is not a guarantee that $\|R_i\|_F / \|S^r\|_F$ stays below ρ for the current carrier at every future round. After freezing, R_i stays constant but $S^r + R_i$ can move, so output stability is not implied by factor freezing alone. The anchor update reuses the captured scale and reapplies the same bounded map after every factor update, so the rank and captured-scale amplitude invariants continue to hold.

Resource accounting. The local optimizer inventory counts candidate scalars admitted to the optimizer; row masking can restrict the entries that receive updates. All-client state sums stored model tensors over five clients, including backbone copies. It is not the extra state of one task or one client. The checkpoint field measures serialized artifacts and may include auxiliary method files. The reported CUDA peak is allocated memory for the process rather than global device occupancy. Process wall time covers 15 rounds, and samples/s includes repeated training presentations. Logical communication totals count tensor bytes for model-state exchange: the method-state view assumes a cached frozen backbone, whereas the full-state view includes backbone copies. Neither is a packet capture, and synchronized inference latency is not included.

F TRAINING-OBSERVATION DEFINITIONS

For each observed layer, family, and task row, the effective prompt is $X = S + R$ and the shared upstream gradient is $G = \partial \mathcal{L} / \partial X$. The gradient-weighted cosine compares flattened $S \odot G$ with $R \odot G$. It is not the cosine between $\partial \mathcal{L} / \partial S$ and $\partial \mathcal{L} / \partial R$, which are structurally identical for the additive path. Update cosine compares effective S and R before and after local training, using $\cos(\Delta S, \Delta R)$. Measurements with unavailable or zero-norm directions are excluded. Summaries average valid measurements within a run before computing the three-seed mean and sample SD.

The carrier-drift statistic stacks the shared carrier dictionary tensors across layers and families into $\mathbf{s}^r$ and computes $\|\mathbf{s}^{r+1} - \mathbf{s}^r\|_2 / \|\mathbf{s}^r\|_2$ after server aggregation. It excludes residual factors

and the classifier. The residual-ratio statistic divides the effective residual norm by its fixed captured scale, not by the current carrier norm. Observed rank is the numerical rank returned by `torch.linalg.matrix_rank` with its default tolerances, rather than an entropy-based effective-rank score. None of these observations uses task accuracy as an input. The same carrier-movement statistic supplies the update trigger; trigger and recovery fields are logged when an event occurs, together with the anchor loss, changed rows, and post-update bound check.