JAM: JOINT ALLOCATION AND MATCHING FOR SAMPLE SHARING UNDER EMERGING ENVIRONMENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

In edge deployments without cloud connectivity, clients encounter different new environments at different times, such as distribution shifts caused by lighting or terrain. Directing information about environments already encountered by some clients to those that have not yet encountered them can support early adaptation. With limited communication, however, each sender must decide whom to send to, how much to send, and what to send. Two challenges arise: class labels do not readily reveal environmental differences within a class, and the same sample has different value to different clients. We propose JAM (Joint Allocation and Matching), which measures differences using features from the common initial model rather than class labels and jointly decides whom to send to, how much to send, and what to send within the budget. Multi-seed experiments on three datasets show that, compared with random sharing, separate allocation and selection, sample selection, and prototype exchange, JAM can significantly improve deployment performance in new environments on datasets with identifiable within-class environmental deviations, without sacrificing average old-environment loss. Quantizing feature averages in JAM control metadata to 4-bit reduces communication while retaining the main adaptation gains.

1 INTRODUCTION

Learning models are increasingly deployed on edge devices that cannot maintain cloud connectivity, such as disaster-response equipment (Wang et al., 2024; Wagle et al., 2025). We call a device capable of independent execution, training, and communication an **edge client** (hereafter, client). Before deployment clients receive a common model trained on **old-environment** data, i.e., data conditions encountered before deployment, but typically do not retain the original training data (Wang et al., 2021; Niu et al., 2022). In the field, they adapt to new environments, including changes in lighting, terrain, and weather, using continuously arriving labeled data (Niu et al., 2023). Such changes can create substantial within-class differences that class-label distributions cannot reveal. Clients operate at different locations: a new environment not yet encountered by one client may already have been observed by others that have accumulated relevant data (Figure 1(a)). For that client, we call such an environment a **Deficit environment**. It may later appear locally, at which point adaptation requires multiple local-model updates unless the client has learned in advance. Whether the client can obtain complementary data from earlier observers before encountering the environment therefore determines whether this adaptation lag can be avoided.

Point-to-point links in this setting are often intermittent and bandwidth-limited because of energy constraints and terrain obstructions. They support only small exchanges of samples and statistics, making frequent full-model transmission difficult (McMahan et al., 2017; Lian et al., 2017; Tan et al., 2022). A single **sender budget** also limits the total samples each sender transmits to all receivers in a round. Our central question is therefore: under a limited sender budget, which receivers should receive budget, how much should each receive, and which samples should be sent to each? The utility of each budget unit depends on the specific receiver–sample pair. Public labels do not reveal a receiver's deficit, and a sample that helps client j may not benefit client k. Separating quota allocation from sample selection can discard this pair-specific information and spend scarce budget on samples that are valuable overall but redundant for a particular receiver.

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.

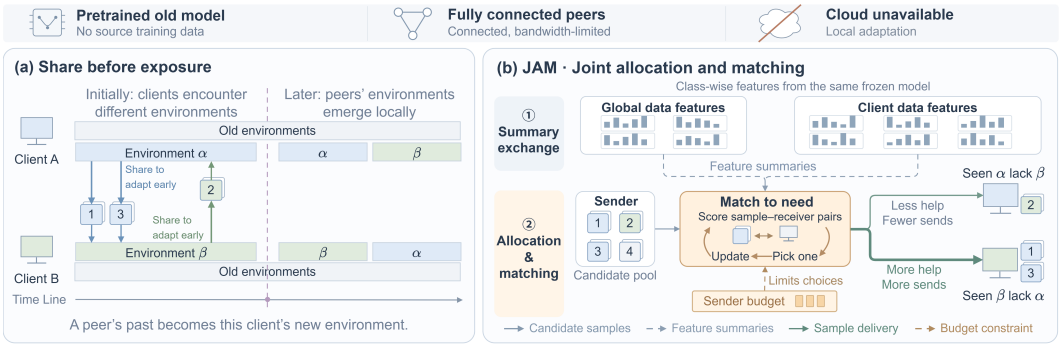

Figure 1: Overview of the deployment setting and JAM. Clients start from a common old-environment model, observe new environments asynchronously, and exchange only a few samples and per-class statistics. JAM jointly selects receiver–sample pairs under each sender's total budget.

To address these challenges, we propose **JAM (Joint Allocation and Matching)**, which uses the old-environment model as a feature extractor to assess the effect of sending each sample to each client. As Figure 1(b) shows, JAM prioritizes samples that best address receiver deficits, jointly deciding whom to send to, how much to send, and what to send.

Our main contributions are:

- We formalize limited point-to-point data sharing during cloud outages: new environments arrive at different times, and each sender jointly decides whom to send to, how much to send, and what to send within its sender budget.
- We propose JAM, characterize receiver–sample coupling in the single-pair cost and its implications for independent additive scoring, and develop a dynamic marginal-gain greedy algorithm for joint selection.
- Experiments show that JAM significantly improves pre-arrival and whole-process adaptation to new environments over most baselines on datasets with identifiable within-class environmental deviations. Mechanism ablations, the offline summary signal-to-noise ratio (Summary SNR), and communication-cost analysis characterize its applicability and limitations.

2 Related Work

A direct response to a new environment is to update the client model using local observations while seeking to preserve prior capabilities. Test-time adaptation updates models without ground-truth class labels, using post-deployment data and the model's own predictions. For example, TENT reduces predictive uncertainty, EATA selects update samples and constrains changes to important parameters, and SAR selects reliable samples while reducing sensitivity to noise (Wang et al., 2021; Niu et al., 2022; 2023). In continual learning, EWC mitigates forgetting by constraining changes to parameters important to previous tasks (Kirkpatrick et al., 2017). These methods focus on local updates; when a new environment first appears elsewhere, local observations alone cannot provide the corresponding training information in advance.

One collaborative approach to exploiting environments already observed by other clients is to share raw samples for peer-assisted training. CF-CL selects samples for exchange using representative points retained by receivers and aligns embeddings through unsupervised contrastive training; DACA organizes sharing by data distributions and trusted links (Wagle et al., 2022; Hu et al., 2023). Resource-constrained collaborative learning also studies communication-link selection and data-transfer volumes between devices (Wang et al., 2024; Wagle et al., 2025). Choosing what to share is related to data selection: ODE retains streaming samples by training value (Gong et al., 2023), while EL2N, Moderate Coreset, CCS, FedCS, and SES select training subsets using prediction error, feature representativeness, or structural entropy (Paul et al., 2021; Xia et al., 2023; Zheng et al., 2023; Hao et al., 2025; Xie et al., 2025). ToV, DataMIL, and TADS further select data based on target-task

performance or target distributions (Jain et al., 2026; Dass et al., 2026; Cheng et al., 2026). These studies address how to organize sharing and which samples are valuable. We further ask how to jointly decide whom to send to, how much to send, and what to send under a sender budget when new environments arrive at different times and environment identities are inaccessible.

Clients can also transfer knowledge by exchanging models or feature representations rather than raw samples. Federated averaging and FedProx aggregate local-model updates at a server (McMahan et al., 2017; Li et al., 2020), while decentralized optimization exchanges model parameters among neighbors (Lian et al., 2017). Recent work further explores communication timing, personalized collaboration, and lightweight adapter sharing (Zhu et al., 2026; Zhang & Azizan, 2026; Seo et al., 2026). In our setting, cloud outages interrupt server aggregation, and full-model exchange remains costly under limited bandwidth. To reduce parameter transmission, FedProto and FedTGP exchange class prototypes, i.e., feature vectors representing a class (Tan et al., 2022; Zhang et al., 2024); FedFed shares performance-sensitive features extracted from local data to mitigate distribution differences across clients (Yang et al., 2023). FedProto's per-class mean prototypes have a statistical form similar to our summaries, making it a representative baseline for comparing direct prototype exchange with summary-guided sample sharing. We adapt the original server-aggregated FedProto to point-to-point FedProto.

3 Problem Setting

Let $\mathcal{K} = \{1, \dots, K\}$ be the client set and $\mathcal{C} = \{1, \dots, C\}$ the public class set. All clients start from the common model G_0 trained on old-environment data, with parameters θ_0. During the cloud outage $t = 0, \dots, T-1$, each client updates and deploys its local model θ_i^t. Let $\mathcal{R}_i^t$ denote the labeled data naturally arriving in round t. An environment $h \in \mathcal{H}$ is a latent condition affecting data generation: even for a fixed public class c, the conditional distribution $P(X \mid Y = c, H = h)$ can vary with h. Differences in the environments encountered by clients and their mixture proportions induce within-class non-IID data across clients. Clients observe only samples, public class labels, and arrival times, not environment identities. Appendix B.1 details the within-class environment mixtures.

Let $\mathcal{R}_i^{\leq t} = \bigcup_{\tau=0}^{t} \mathcal{R}_i^{\tau}$, and let $\mathcal{S}_i^{<t}$ denote previously received data available for training. In each round, all methods perform the same number of updates L on the currently available cumulative data:

$$\mathcal{D}_i^t = \mathcal{R}_i^{\leq t} \cup \mathcal{S}_i^{<t}, \qquad \theta_i^{t+1} = \mathrm{LocalUpdate}(\theta_i^t, \mathcal{D}_i^t; L) \tag{1}$$

Samples sent in round t enter training from round $t+1$. Clients form a fully connected point-to-point network, with $\mathcal{N}_i = \mathcal{K} \setminus \{i\}$ denoting sender i's receivers. Below, j denotes a receiver, x a candidate sample, and y_x its public class. Sender i's set of feasible receiver–sample pairs in round t is

$$\Omega_i^t = \big\{(j, x) : j \in \mathcal{N}_i,\ (x, y_x) \in \mathcal{R}_i^t, x \text{ has not been sent along } i \to j \text{ yet}\big\}. \tag{2}$$

Let $A_i^t \subseteq \Omega_i^t$ be the set of selected receiver–sample pairs, with each (j, x) called a pair. The set of samples sent to receiver j is $A_{i\to j}^t = \{x : (j, x) \in A_i^t\}$. The sender budget upper bound imposes

$$|A_i^t| = \sum_{j \in \mathcal{N}_i} |A_{i\to j}^t| \leq U_i^t. \tag{3}$$

Transmitting one pair consumes one budget unit; sending the same sample to two receivers counts twice. Candidate deduplication, data forwarding, and rules for fixed sending doses are described in Appendix B.1.

4 JAM: Joint Allocation and Matching

4.1 Motivation

Public labels and class counts describe only class proportions, not the full environmental differences within a class. We therefore move from discrete labels to feature representations: samples of the same class may differ substantially in feature space, revealing distribution shifts invisible to label counts. The model G_0, already available at deployment, provides these representations; freezing

its feature extractor gives clients a common basis for comparison. JAM thus describes receiver differences through per-class feature averages. Comparing classes separately also prevents shifts in different classes from canceling in the overall average (Appendix A.1).

The same sample may reduce one receiver's deviation but increase another's. Its value therefore depends on the receiver, whereas per-class differences describe only the receiver side. JAM directly evaluates receiver–sample pairs under the sender budget to jointly determine receiver-budget allocation and receiver–sample matching.

4.2 Joint Receiver–Sample Formulation

We freeze the feature extractor ϕ_0 preceding the classifier in the old-environment model G_0, providing a common feature space for deviation calculations. A sample's reference feature is $z_x = \phi_0(x) \in \mathbb{R}^d$. Clients' deployed models continue to update according to equation 1.

The network reference is constructed from cumulative naturally arriving data across all clients through round t. Let $\mathcal{R}_{k,c}^{\le t} = \{(x,y) \in \mathcal{R}_k^{\le t} : y = c\}$. Client k computes local per-class statistics, which are aggregated by sample count to form the network reference:

$$\begin{aligned} N_{k,c}^t &= |\mathcal{R}_{k,c}^{\le t}|, \qquad \bar{\mu}_{k,c}^t = \frac{1}{N_{k,c}^t} \sum_{(x,y)\in\mathcal{R}_{k,c}^{\le t}} z_x, \\ N_c^t &= \sum_{k\in\mathcal{K}} N_{k,c}^t, \qquad p_c^t = \frac{N_c^t}{\sum_{c'} N_{c'}^t}, \qquad\qquad \mu_c^t = \frac{\sum_{k\in\mathcal{K}} N_{k,c}^t \bar{\mu}_{k,c}^t}{N_c^t}. \end{aligned} \tag{4}$$

Here, $\{(N_{k,c}^t, \bar{\mu}_{k,c}^t)\}_c$ are the local per-class statistics, and their aggregation gives the network reference $\{(p_c^t, \mu_c^t)\}_c$; shared copies are not counted again. Receiver summaries must also reflect previously received shared data. Let $\mathcal{D}_{j,c}^t = \{(x,y) \in \mathcal{R}_j^{\le t} \cup \mathcal{S}_j^{<t} : y = c\}$, and define

$$n_{j,c}^t = |\mathcal{D}_{j,c}^t|, \qquad \mu_{j,c}^t = \frac{1}{n_{j,c}^t} \sum_{(x,y)\in\mathcal{D}_{j,c}^t} z_x. \tag{5}$$

The set $\{(n_{j,c}^t, \mu_{j,c}^t)\}_c$ is client j's receiver summary in round t. Including received samples allows subsequent selection to reflect differences already reduced by sharing. For any feasible transmission set A_i^t, let $A_{i\to j,c}^t = \{x \in A_{i\to j}^t : y_x = c\}$ contain the class-c samples in $A_{i\to j}^t$ sent to receiver j. The temporary per-class feature average after adding these samples is

$$\mu_{j,c}^{t,+}(A_i^t) = \frac{n_{j,c}^t \mu_{j,c}^t + \sum_{x\in A_{i\to j,c}^t} z_x}{n_{j,c}^t + |A_{i\to j,c}^t|}. \tag{6}$$

For sender i, JAM solves

$$\begin{aligned} \underset{A_i^t \subseteq \Omega_i^t}{\text{minimize}} \quad & \Phi_i^t(A_i^t) = \sum_{j\in\mathcal{N}_i} \sum_{c\in\mathcal{C}} p_c^t \left\| \mu_{j,c}^{t,+}(A_i^t) - \mu_c^t \right\|_2^2, \\ \text{subject to} \quad & |A_i^t| \le U_i^t. \end{aligned} \tag{7}$$

Here, p_c^t aggregates per-class differences using network public-class proportions, and all receivers have equal weight. The resulting set jointly determines receiver budgets, class budgets, and specific samples. The constraint in equation 7 follows the deployment setting: the round budget is an upper bound, allowing senders to choose how much to send. Our formal comparisons fix this degree of freedom at a sending dose $b_i^t = U_i^t$ and minimize the same objective subject to $|A_i^t| = b_i^t$.

To formalize the dependence of sample value on the receiver, consider transmitting one pair in equation 7 and omit round superscripts. Let $e_{q,c} = \mu_{q,c} - \mu_c$ be receiver q's pre-sharing within-class deviation, and $r_s = z_s - \mu_c$ the candidate-sample offset of s from the same-class network reference. In the single-sample case of equation 6, subtracting the network reference μ_c from the updated per-class feature average gives the post-sharing within-class deviation $e_{q,c}^+(s) = \mu_{q,c}^+(s) - \mu_c = (n_{q,c} e_{q,c} + r_s)/(n_{q,c} + 1)$. Sending sample s to receiver q changes only one receiver–class term in equation 7; we call this term the single-pair cost, $C_{q,s} = p_c \|e_{q,c}^+(s)\|_2^2$.

Algorithm 1 Dynamic marginal-gain selection in JAM

Require: Feasible receiver–sample pairs Ω_i^t; sending dose b_i^t; network reference $\{p_c^t, \mu_c^t\}_c$; receiver summaries $\{n_{j,c}^t, \mu_{j,c}^t\}_{j,c}$

Ensure: Set of selected receiver–sample pairs A_i^t

$r \leftarrow 0$; $A_i^{t,(0)} \leftarrow \varnothing$; $\widetilde{\Omega}_i^t \leftarrow \Omega_i^t$

$\widetilde{n}_{j,c} \leftarrow n_{j,c}^t$; $\widetilde{\mu}_{j,c} \leftarrow \mu_{j,c}^t$ for all $j \in \mathcal{N}_i$, $c \in \mathcal{C}$

while $r < b_i^t$ **and** $\widetilde{\Omega}_i^t \neq \varnothing$ **do**

  for all $(j,x) \in \widetilde{\Omega}_i^t$ **do**

    Set $c \leftarrow y_x$ and compute $\Delta_{i,j,x}^{t,(r)}$ by equation 9 and equation 10

  end for

  $(j^*, x^*) \leftarrow \arg\max_{(j,x)\in\widetilde{\Omega}_i^t} \Delta_{i,j,x}^{t,(r)}$

  Set $c^* \leftarrow y_{x^*}$ and $A_i^{t,(r+1)} \leftarrow A_i^{t,(r)} \cup \{(j^*, x^*)\}$

  $\widetilde{\mu}_{j^*,c^*} \leftarrow \frac{\widetilde{n}_{j^*,c^*}\widetilde{\mu}_{j^*,c^*} + z_{x^*}}{\widetilde{n}_{j^*,c^*}+1}$; $\widetilde{n}_{j^*,c^*} \leftarrow \widetilde{n}_{j^*,c^*} + 1$

  $\widetilde{\Omega}_i^t \leftarrow \widetilde{\Omega}_i^t \setminus \{(j^*, x^*)\}$; $r \leftarrow r + 1$

end while

return $A_i^t \leftarrow A_i^{t,(r)}$

Proposition 1 (non-separability of the single-pair cost). Fix a public class c with $p_c > 0$. Suppose receivers j, k have equal positive class counts $n_{j,c} = n_{k,c} > 0$, and candidate samples x, y and all four receiver–sample pairs are feasible. Then

$$C_{j,x} + C_{k,y} - C_{j,y} - C_{k,x} = \frac{2n_{j,c}p_c}{(n_{j,c}+1)^2} \langle e_{j,c} - e_{k,c}, r_x - r_y \rangle . \tag{8}$$

Thus, whenever the inner product is nonzero, the single-pair cost cannot be written as $C_{q,s} = a_q + b_s$, where a_q depends only on the receiver and b_s only on the sample. Appendix A.2 provides the full proof, its equivalence to the marginal-gain formulation, and the degenerate cases.

4.3 Marginal-Gain Greedy Optimization

A transmission changes the receiver state and subsequent sample values, so JAM uses a dynamic greedy heuristic for this problem with a fixed sending dose, without an optimality guarantee. Fix sender i and round t, and let $A_i^{t,(r)}$ be the set after r selections. Algorithm 1 maintains temporary copies $\widetilde{n}_{j,c}$ and $\widetilde{\mu}_{j,c}$ of receiver summaries, initialized from the current round's $n_{j,c}^t$ and $\mu_{j,c}^t$. Each selected pair updates the corresponding class count and feature average. For a candidate pair (j,x), let $c = y_x$. The pre-sharing within-class deviation, candidate-sample offset, and post-sharing within-class deviation satisfy

$$e = \widetilde{\mu}_{j,c} - \mu_c^t, \quad r_x = z_x - \mu_c^t, \quad e^+ = \frac{\widetilde{n}_{j,c}e + r_x}{\widetilde{n}_{j,c}+1}. \tag{9}$$

We define the marginal gain as the objective difference before and after adding (j,x):

$$\Delta_{i,j,x}^{t,(r)} = \Phi_i^t\Big(A_i^{t,(r)}\Big) - \Phi_i^t\Big(A_i^{t,(r)} \cup \{(j,x)\}\Big) = p_c^t \left(\|e\|_2^2 - \|e^+\|_2^2\right). \tag{10}$$

Maximizing equation 10 selects the feasible pair with the largest current marginal gain for the objective in equation 7. The algorithm fills the sending dose b_i^t. A fixed sending dose permits comparing selection quality at the same transmission count. Moreover, increasing the dose improves both random sharing and JAM within the tested budget range (Section 5.4), making full-budget transmission a reasonable operating strategy. Filling does not stop at nonpositive gains: even when all candidates have negative marginal gains, the best candidate is sent. Equivalent expansions and derivations appear in Appendix A.2, and Algorithm 1 gives the complete procedure.

Table 1: Main results on early adaptation and whole-process deployment performance. Pre A denotes Deficit accuracy before natural arrival; Nov L and Dep L denote whole-process Novel and Deploy Loss averaged over rounds 1 to 9. Best results are in bold.

	CIFAR-100, $U = 16$			PACS, $U = 5$			Office-Home, $U = 5$		
Method	Pre A	Nov L	Dep L	Pre A	Nov L	Dep L	Pre A	Nov L	Dep L
Frozen G_0	19.18	2.8118	2.5100	**71.94**	0.9230	0.8175	18.35	3.5382	3.4862
Local-U0	20.75	2.5590	2.3601	68.66	0.8031	0.7601	20.69	3.4143	3.3958
Neutral Random	21.00	2.5492	2.3541	69.85	0.7957	0.7566	**20.87**	3.4134	**3.3950**
SES	20.89	2.5475	2.3533	70.06	0.7942	0.7549	20.84	3.4138	3.3952
Need-SES	21.23	2.5459	2.3522	69.95	0.7940	0.7550	20.71	3.4134	3.3951
Need-CF-CL	21.12	2.5488	2.3532	69.84	0.7969	0.7574	20.64	3.4148	3.3961
FedProto	20.73	2.5434	2.3492	68.64	0.7996	0.7593	20.67	3.4135	3.3953
JAM	**21.69**	**2.5303**	**2.3434**	70.95	**0.7761**	**0.7485**	20.83	**3.4132**	3.3952

5 Experiments

5.1 Experimental Setup

All clients start from the old-environment model G_0 and continue training during a ten-round cloud outage. We simulate environment shifts by changing data composition within each public class. Old denotes environments used to train G_0; Seen denotes the new environment a client first encounters after deployment; Deficit denotes the new environment already observed by the other client group but not naturally encountered by that client until round 4. Links between the two groups are complementary links; links within a group are same-group links. CIFAR-100 uses 20 superclasses as public classes and fine classes to simulate environments; PACS and Office-Home use real visual domains (Krizhevsky, 2009; Li et al., 2017; Venkateswara et al., 2017). Except for Office-Home's round-0 full-class-coverage allocation of 16 Old and 49 Seen samples, naturally arriving data contain equal Old/Seen shares in rounds 0–3 and Old/Seen/Deficit shares of $50/25/25\%$ in rounds 4–8. The final round only observes the effect of sharing in the preceding round. The sender budget upper bound is $U = 16$ for CIFAR-100 and $U = 5$ for PACS and Office-Home (arrival counts and schedules are in Appendix B.1). Evaluation distributions use these three environments: Novel combines Seen and Deficit, while Deploy combines all three with round-dependent weights (Appendix B.2). Metrics use client macro-averages. Whole-process metrics average rounds 1–9 equally, excluding round 0 because sharing has not yet taken effect; Pre Accuracy is evaluated in round 3.

Main experiments use ResNet-18 (He et al., 2016). We train G_0 from random initialization on CIFAR-100 and initialize it with ImageNet-pretrained weights on PACS and Office-Home (Deng et al., 2009). Robustness experiments also use MobileNetV2 (Sandler et al., 2018). Main results use five random seeds with paired significance tests; mechanism, robustness, and communication analyses use three common seeds from this set and report means and sample standard deviations. Seed values and statistical procedures are given in Appendix B.3.

The main table reports Frozen G_0, Local-U0, Neutral Random, SES (Xie et al., 2025), Need-SES, Need-CF-CL, FedProto (Tan et al., 2022), and JAM; additional baselines appear in Appendix Table 14. Appendix B.3 details implementations and hyperparameters.

5.2 Main Results

On all three datasets, Local-U0 lowers whole-process Novel Loss relative to Frozen G_0; sharing only U samples per round with Neutral Random lowers it further at the same local-update count. Appendix Table 14 reports other local-update and sharing methods.

The next question is how to use the limited sharing budget. Need-SES and Need-CF-CL first allocate the sender budget according to each receiver's overall need, aggregated from per-class deviations (allocation rules are in Appendix B.3), then select samples within the quotas using their respective rules. Their allocation and selection stages are separate, and their whole-process Novel Loss differs little from random sharing. JAM jointly determines allocation and matching and achieves the lowest Novel Loss on all three datasets. On CIFAR-100 and PACS, paired improvements over all six base-

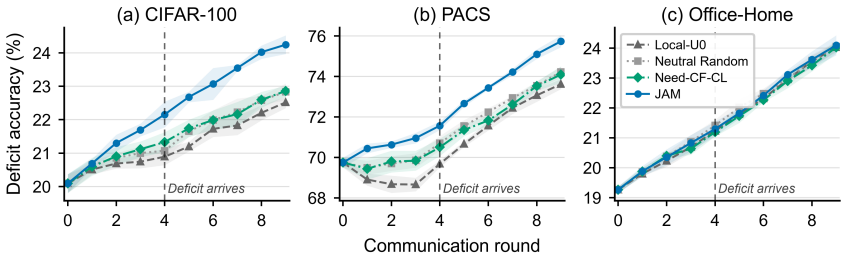

Figure 2: Deficit adaptation before and after natural arrival. Curves report five-seed mean accuracy. The vertical line marks the first natural arrival of the Deficit environment; samples selected in round t become available from round $t + 1$.

Table 2: Mechanism analysis relative to Neutral Random. ΔNovel is the whole-process Novel Loss reduction over rounds 1–9; R_{alloc} is the ratio of average per-link budgets on complementary versus same-group links (1 denotes uniform), and Def. (%) is the percentage of transmitted samples from the receiver's Deficit environment, both computed over rounds 0–3 before natural arrival. By construction, JAM-Matching (JAM-Allocation) shares the R_{alloc} of Neutral Random (JAM); these two columns describe decision flow rather than performance. Per-seed values with standard deviations are in Appendix Table 16.

	CIFAR-100, SNR = 5.37			PACS, SNR = 14.98			Office-Home, SNR = 0.55		
Method	ΔNovel	R_{alloc}	Def. (%)	ΔNovel	R_{alloc}	Def. (%)	ΔNovel	R_{alloc}	Def. (%)
Neutral Random	0	1.016	28.2	0	0.917	21.7	0	0.917	31.9
JAM-Matching	0.0110	1.016	31.7	0.0063	0.917	37.3	−0.0000	0.917	31.9
JAM-Allocation	0.0035	1.443	31.8	0.0106	4.194	51.5	−0.0000	1.161	31.0
JAM-Permuted	0.0132	1.140	31.2	0.0197	2.616	74.3	0.0002	1.155	31.4
Need-JAM	0.0103	1.019	33.4	0.0145	0.902	52.7	−0.0003	0.943	31.5
Static-JAM	0.0123	1.554	39.4	0.0085	11.319	83.8	−0.0005	1.195	32.5
JAM	0.0203	1.443	38.0	0.0205	4.194	81.9	0.0003	1.161	31.9

lines remain significant after Holm correction. On Office-Home, improvements in whole-process Novel Loss over these six baselines are all below 0.002 and are not significant. JAM's whole-process Old Loss is lower than that of the pre-deployment model G_0 on all three datasets, although local trade-offs remain: whole-process Seen Loss on CIFAR-100 is slightly higher than Neutral Random's, and whole-process and final-round Old Accuracy on Office-Home are 0.17 and 0.29 pp lower than Need-SES's, respectively (Table 13). Differences, confidence intervals, and complete tests are in Appendix D.1.

Figure 2 tracks Deficit accuracy across rounds. At the last pre-arrival evaluation (round 3), JAM has higher Pre Accuracy than most baselines on CIFAR-100 and PACS, with improvements significant after Holm correction. The only main-table exception to its higher Pre Accuracy is Frozen G_0 on PACS. JAM still leads Need-SES and Need-CF-CL by 0.5–1.1 pp, but these leads are not significant. Its improvements over both baselines in whole-process Novel Loss and Final Accuracy remain significant after Holm correction (Appendix D.1). Accuracy generally rises after the first natural arrival (round 4); on PACS, JAM's earlier deficit relative to Frozen G_0 becomes a clear lead by the final round (Appendix Table 14). Office-Home trajectories remain close at all evaluation times; this dataset also has the lowest Summary SNR (Section 5.3).

5.3 Mechanism Analysis

We use four variants and one replay control to examine receiver-budget allocation, receiver–sample matching, and dynamic updates to temporary receiver states. For component analysis, JAM-Matching retains Neutral Random's uniform round-robin budget and selects samples only with JAM's matching rule. JAM-Allocation replays, round by round, frozen per-receiver, per-class transmission counts from the same-seed JAM run; its quotas do not change with its own state, and samples

Table 3: Robustness under changes in architecture, sender budget, initialization quality, and data-stream structure. $\Delta L = L_{\mathrm{Random}} - L_{\mathrm{JAM}}$; larger values favor JAM. MNV2 denotes MobileNetV2. Nov L and Dep L follow Table 1; Pre$_\tau$ L is the Deficit loss at the last evaluation round before each client's own Deficit arrival in the staggered schedule. All dimensions use the three common seeds; values are means, with per-seed values and standard deviations in Appendix Table 16.

Variation	Setting	Metric	Neutral Random	JAM	ΔL
Architecture	PACS, MNV2	Nov L	0.9049	0.8752	+0.0297
Architecture	CIFAR-100, MNV2	Nov L	2.6627	2.6512	+0.0115
Sender budget	CIFAR-100, U=8	Nov L	2.5506	2.5355	+0.0151
Sender budget	CIFAR-100, U=32	Nov L	2.5430	2.5172	+0.0258
Initialization	Old-strong G_0	Nov L	3.0262	2.9757	+0.0505
Initialization	Old-strong G_0	Dep L	2.4379	2.4058	+0.0321
Data stream	Staggered 2:8	Nov L	2.5408	2.5220	+0.0188
Data stream	Staggered 2:8	Pre$_\tau$ L	2.6662	2.6325	+0.0337

are selected randomly within each receiver–class cell. For decision-process analysis, Need-JAM recomputes and fixes receiver quotas each round from its own state using the Need-SES rule, then dynamically selects samples with JAM scores within those quotas. Static-JAM scores pairs only once at the start of each round, without updating scores as samples are selected. The fifth control, JAM-Permuted, replays JAM's transmission logs, preserving the selected samples and per-class received counts while permuting their receivers. Unlike JAM-Allocation's random sampling, JAM-Permuted retains the exact samples selected by JAM to test how permuting receiver–sample pairs under these constraints affects training. In the ΔNovel columns of Table 2, JAM-Matching improves more than JAM-Allocation on CIFAR-100, whereas the reverse holds on PACS; complete improvements for other metrics appear in Appendix Table 15. All four variants yield smaller mean Novel Loss reductions than full JAM on these two datasets, supporting joint selection and dynamic re-evaluation in the tested settings. Permuting pairs removes about 1/3 of the improvement on CIFAR-100 but has little effect on PACS. Thus, CIFAR-100 gains partly arise from specific pairs, whereas PACS gains arise mainly from allocation.

The R_{alloc} and Def. (%) columns of Table 2 show where budgets and samples flow. JAM increases both on CIFAR-100 and PACS: more budget goes to complementary links, and more transmitted samples come from receivers' Deficit environments, with larger changes on PACS. On Office-Home, R_{alloc} increases while Def. (%) is unchanged. Receiver-budget matrices are shown in Appendix Figure 4.

Both allocation and matching require within-class environmental deviations to be identifiable from finite samples. We characterize this identifiability for each dataset using Summary SNR. It treats within-class environmental shifts as signal and estimation noise as the denominator, scaled to a common sample count based on arrivals (definition and derivation are in Appendix A.3). As the header of Table 2 shows, Office-Home has the lowest SNR, and all variants and full JAM have ΔNovel within ± 0.001. This indicates that receiver-aware selection cannot benefit when estimation noise obscures within-class deviations. Cross-dataset performance differences follow the SNR ordering, suggesting that finite-sample mean estimation may limit the gains from receiver-aware selection.

With fixed model and data snapshots, replacing a round's received samples using permuted receiver–sample pairs and applying the same brief training produces near-zero Deficit loss differences relative to JAM's original receiver–sample pairs, with mixed directions across seeds (Appendix D.2). JAM's pair-specific value is realized over the full sharing-and-training process rather than within a single round.

5.4 Robustness

We test robustness to backbone architecture, sender budget, initialization quality, and data-arrival structure (Table 3). JAM's Novel Loss gains over Neutral Random persist with MobileNetV2 (MNV2) on both datasets and with smaller (U=8) and larger (U=32) budgets, increasing with budget. Thus, the gains are not specific to one backbone or budget.

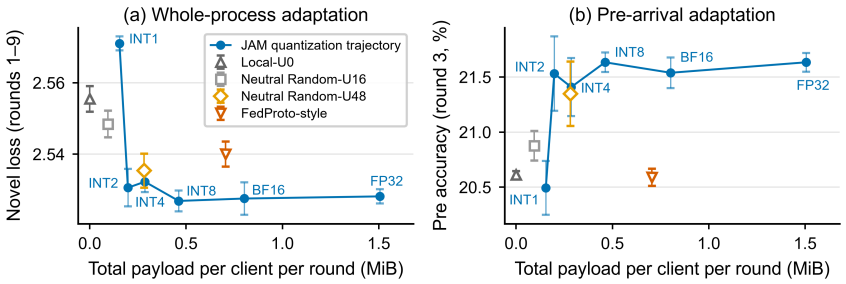

Figure 3: Communication cost versus adaptation performance on CIFAR-100. Panel (a) reports whole-process Novel Loss and panel (b) reports pre-arrival Deficit accuracy. The JAM quantization variants form the compression trajectory; other methods are shown as independent operating points.

JAM also retains its advantage with the thoroughly trained Old-strong G_0. Its Novel Loss and Pre_τ L advantages over Neutral Random persist in Staggered 2:8, which staggers each client's Deficit arrival. Appendices B.1 and D.3 give the schedule and complete results, respectively.

5.5 Communication Cost and Resource Trade-offs

We further evaluate JAM's communication cost and the trade-off between control metadata and raw samples. Control metadata and FedProto class prototypes are charged for all-to-all exchange, whereas raw samples are charged for actual directed transmissions. All costs are reported as bidirectional payload per client per effective communication round (Appendix C.1).

Figure 3 compares total communication cost and adaptation performance in the tested CIFAR-100 setting. JAM's quantization variants form a compression trajectory. BF16 and INT8 barely change transmission decisions. INT4 begins to change decisions but retains the main adaptation gains at about one-fifth of FP32's communication cost. INT2 substantially changes sample selections, yet task performance remains close to FP32. INT1 markedly reduces score-ranking agreement with FP32 and degrades performance. Decision-overlap rates and numerical results appear in Appendix C.3.

Communication can instead be spent directly on raw samples. A CIFAR-100 sample costs only 3 KiB in uncompressed pixel bytes, so JAM's INT4 control metadata is equivalent to 32 additional transmissions. Random-U48 in the figure sends 48 random samples at approximately the same total communication cost as JAM-INT4 (288 KiB versus 293 KiB). JAM's main-table advantage is established at the same sending dose. With approximately matched total communication, 48 random samples perform similarly to 16 targeted samples plus control metadata, offsetting most of the equal-dose advantage. This trade-off depends on raw-sample payload: with PACS and Office-Home images, the corresponding control metadata converts to only 0 and 1 additional samples, respectively (cross-dataset ledgers are in Appendix C.3).

6 Conclusion

We proposed JAM to jointly optimize receiver-budget allocation and receiver–sample matching under sender budgets. On CIFAR-100 and PACS, where within-class environmental deviations are identifiable, JAM supports early adaptation to complementary environments and significantly improves whole-process new-environment and deployment performance. Sharing methods perform similarly on Office-Home, which has the weakest identifiability signal. JAM achieves lower average old-environment loss than the pre-deployment model G_0 on all three datasets. Mechanism controls and Summary SNR analysis characterize the role and limits of joint selection. On CIFAR-100, 4-bit feature-average quantization retains the main adaptation gains while reducing control-metadata cost; comparisons with approximately matched total communication show that targeted-selection gains must be weighed against summary costs.

AI USE STATEMENT

We used generative AI tools to assist with discussing and checking research plans, writing and debugging portions of the experimental code and analysis scripts, organizing experimental results and checking their consistency, and drafting portions of mathematical derivations and proofs. These tools also assisted with language editing, literature searches, and scripts for generating figures and tables. The authors made all final decisions on the research questions, method design, experimental settings, result interpretations, and conclusions. All AI-assisted code, mathematical derivations, citations, and manuscript text were manually reviewed by the authors. Mathematical derivations were checked step by step, and the authors independently ran and verified the formal experiments under the reported settings. Generative AI tools were not used to generate, alter, or replace the experimental data or results reported in this paper. The authors take full responsibility for all content, conclusions, and research artifacts.

REPRODUCIBILITY STATEMENT

All formal experiments used fixed random seeds and deterministic training settings. Seed values and statistical procedures are provided in Appendix B.3; data-arrival streams and deployment evaluation distributions in Appendices B.1 and B.2; and models, initialization, all hyperparameters, and exact baseline implementations in Appendix B.3. Per-seed values and sample standard deviations for the main table and the mechanism, robustness, and communication analyses are provided in Appendix D.1 and Tables 16 and 15.

REFERENCES

Hao Cheng, Rui Zhang, Ling Li, Na Di, Jiaheng Wei, Zhaowei Zhu, and Bo Han. Task-aware data selection via proxy-label enhanced distribution matching for LLM finetuning. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=R40WoYbYab.

Shivin Dass, Alaa Khaddaj, Logan Engstrom, Aleksander Madry, Andrew Ilyas, and Roberto Martín-Martín. DataMIL: Selecting data for robot imitation learning with datamodels. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=AcTsKglDdh.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848. URL https://www.image-net.org/static_files/papers/imagenet_cvpr09.pdf.

Chen Gong, Zhenzhe Zheng, Yunfeng Shao, Bingshuai Li, Fan Wu, and Guihai Chen. To store or not? online data selection for federated learning with limited storage. In *Proceedings of the ACM Web Conference 2023*, pp. 3044–3055, 2023. doi: 10.1145/3543507.3583426. URL https://doi.org/10.1145/3543507.3583426.

Chenhe Hao, Weiying Xie, Daixun Li, Haonan Qin, Hangyu Ye, Leyuan Fang, and Yunsong Li. FedCS: Coreset selection for federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15434–15443, 2025. doi: 10.1109/CVPR52734.2025.01438. URL https://openaccess.thecvf.com/content/CVPR2025/html/Hao_FedCS_Coreset_Selection_for_Federated_Learning_CVPR_2025_paper.html.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90. URL https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html.

Gang Hu, Yinglei Teng, Nan Wang, and F. Richard Yu. Clustered data sharing for non-iid federated learning over wireless networks. In *IEEE International Conference on Communications*, pp.

1175–1180, 2023. doi: 10.1109/ICC45041.2023.10279434. URL https://doi.org/10.1109/ICC45041.2023.10279434.

Ayush Jain, Andrea Montanari, and Eren Sasoglu. Train on validation (ToV): Fast data selection with applications to fine-tuning. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=fWHd3yYicX.

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. URL https://www.pnas.org/doi/10.1073/pnas.1611835114.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5542–5550, 2017. doi: 10.1109/ICCV.2017.591. URL https://openaccess.thecvf.com/content_iccv_2017/html/Li_Deeper_Broader_and_ICCV_2017_paper.html.

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, 2020. URL https://proceedings.mlsys.org/paper_files/paper/2020/hash/1f5fe83998a09396ebe6477d9475ba0c-Abstract.html.

Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://papers.nips.cc/paper_files/paper/2017/hash/f75526659f31040afeb61cb7133e4e6d-Abstract.html.

Xiangru Lian, Wei Zhang, Ce Zhang, and Ji Liu. Asynchronous decentralized parallel stochastic gradient descent. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3043–3052. PMLR, 2018. URL https://proceedings.mlr.press/v80/lian18a.html.

H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera 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.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16888–16905. PMLR, 2022. URL https://proceedings.mlr.press/v162/niu22a.html.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=g2YraF75Tj.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/ac56f8fe9eea3e4a365f29f0f1957c55-Abstract.html.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018. doi: 10.1109/CVPR.2018.00474. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Sandler_MobileNetV2_Inverted_Residuals_CVPR_2018_paper.html.

Minhyuk Seo, Taeheon Kim, Hankook Lee, Jonghyun Choi, and Tinne Tuytelaars. Co-LoRA: Collaborative model personalization on heterogeneous multi-modal clients. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=0g5Dk4Qfh0.

Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009. URL https://research.cs.wisc.edu/techreports/2009/TR1648.pdf.

Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. FedProto: Federated prototype learning across heterogeneous clients. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(8):8432–8440, 2022. doi: 10.1609/aaai.v36i8.20819. URL https://ojs.aaai.org/index.php/AAAI/article/view/20819.

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5385–5394, 2017. doi: 10.1109/CVPR.2017.572. URL https://openaccess.thecvf.com/content_cvpr_2017/html/Venkateswara_Deep_Hashing_Network_CVPR_2017_paper.html.

Satyavrat Wagle, Seyyedali Hosseinalipour, Naji Khosravan, Mung Chiang, and Christopher G. Brinton. Embedding alignment for unsupervised federated learning via smart data exchange. In *IEEE Global Communications Conference*, pp. 492–497, 2022. doi: 10.1109/GLOBECOM48099.2022.10000962. URL https://doi.org/10.1109/GLOBECOM48099.2022.10000962.

Satyavrat Wagle, Anindya Bijoy Das, David J. Love, and Christopher G. Brinton. Multi-agent reinforcement learning for graph discovery in d2d-enabled federated learning, 2025. URL https://arxiv.org/abs/2503.23218.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=uXl3bZLkr3c.

Su Wang, Roberto Morabito, Seyyedali Hosseinalipour, Mung Chiang, and Christopher G. Brinton. Device sampling and resource optimization for federated learning in cooperative edge networks. *IEEE/ACM Transactions on Networking*, 32(5):4365–4381, 2024. doi: 10.1109/TNET.2024.3423673. URL https://doi.org/10.1109/TNET.2024.3423673.

Xiaobo Xia, Jiale Liu, Jun Yu, Xu Shen, Bo Han, and Tongliang Liu. Moderate coreset: A universal method of data selection for real-world data-efficient deep learning. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=7D5EECbOaf9.

Tianchi Xie, Jiangning Zhu, Guozu Ma, Minzhi Lin, Wei Chen, Weikai Yang, and Shixia Liu. Structural-entropy-based sample selection for efficient and effective learning. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/92ce40962b4098f7bf6eed33128fc606-Abstract-Conference.html.

Zhiqin Yang, Yonggang Zhang, Yu Zheng, Xinmei Tian, Hao Peng, Tongliang Liu, and Bo Han. FedFed: Feature distillation against data heterogeneity in federated learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/bdcdf38389d7fcefc73c4c3720217155-Abstract-Conference.html.

Chenyu Zhang and Navid Azizan. Personalized collaborative learning with affinity-based variance reduction. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=OvCyDyW9kU.

Jianqing Zhang, Yang Liu, Yang Hua, and Jian Cao. FedTGP: Trainable global prototypes with adaptive-margin-enhanced contrastive learning for data and model heterogeneity in federated learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(15):16768–16776, 2024. doi: 10.1609/aaai.v38i15.29617. URL https://ojs.aaai.org/index.php/AAAI/article/view/29617.

Haizhong Zheng, Rui Liu, Fan Lai, and Atul Prakash. Coverage-centric coreset selection for high pruning rates. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=QwKvL6wC8Yi.

Tongtian Zhu, Tianyu Zhang, Mingze Wang, Zhanpeng Zhou, and Can Wang. On the surprising effectiveness of a single global merging in decentralized learning. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=zrFnwRHuQo. Oral Presentation.

A THEORETICAL DETAILS

This appendix presents the per-class deviation decomposition, the proof of non-separability of the single-pair cost, a stable marginal-gain expansion, and the connection between finite-sample summaries and the SNR used in the main text.

A.1 PER-CLASS FEATURE AVERAGE DECOMPOSITION

We show how per-class comparison distinguishes changes in public-class proportions from within-class feature changes. Fix receiver j and a round, and omit time superscripts. We use the receiver's per-class sample counts and feature averages $n_{j,c}, \mu_{j,c}$ and the network reference p_c, μ_c defined in Section 4.2. Define the receiver's public-class proportions as $p_{j,c} = n_{j,c} / \sum_{c'} n_{j,c'}$. The overall feature averages of the receiver and the network reference are

$$\mu_j = \sum_{c \in \mathcal{C}} p_{j,c} \mu_{j,c}, \qquad \mu = \sum_{c \in \mathcal{C}} p_c \mu_c. \tag{11}$$

Adding and subtracting $\sum_{c \in \mathcal{C}} p_{j,c} \mu_c$ in their difference gives

$$\begin{aligned} \mu_j - \mu &= \sum_{c \in \mathcal{C}} p_{j,c} \mu_{j,c} - \sum_{c \in \mathcal{C}} p_c \mu_c \\ &= \left(\sum_{c \in \mathcal{C}} p_{j,c} \mu_c - \sum_{c \in \mathcal{C}} p_c \mu_c \right) + \left(\sum_{c \in \mathcal{C}} p_{j,c} \mu_{j,c} - \sum_{c \in \mathcal{C}} p_{j,c} \mu_c \right) \\ &= \underbrace{\sum_{c \in \mathcal{C}} (p_{j,c} - p_c) \mu_c}_{\text{class-proportion shift}} + \underbrace{\sum_{c \in \mathcal{C}} p_{j,c} (\mu_{j,c} - \mu_c)}_{\text{within-class environment shift}} . \end{aligned} \tag{12}$$

The first term arises from differences in public-class proportions between the receiver and network reference; the second arises from feature-average differences within the same public class. Even when class proportions coincide and the first term vanishes, the second may remain nonzero. Within-class shifts from different classes may also cancel in the overall sum. Per-class feature averages therefore capture receiver differences invisible to public-label counts and retain information that an overall average may obscure, motivating the per-class comparison in Section 4.1.

A.2 NON-SEPARABILITY AND MARGINAL-GAIN DERIVATION

For a fixed public class c, consider receivers j, k and candidate samples x, y. Under Proposition 1, the receivers have equal positive class counts $n_{j,c} = n_{k,c} > 0$, all four receiver–sample pairs are feasible, and the public-class weight satisfies $p_c > 0$. Use $e_{u,c} = \mu_{u,c} - \mu_c$ and $r_s = z_s - \mu_c$ from Section 4.2, where $u \in \{j, k\}$ and $s \in \{x, y\}$. Because receiver counts are equal, define their common coefficient as

$$\gamma_c = \frac{p_c}{(n_{j,c} + 1)^2}. \tag{13}$$

After adding sample s, the deviation is $(n_{u,c} e_{u,c} + r_s)/(n_{u,c} + 1)$, and the corresponding single-pair cost expands as

$$\begin{aligned} C_{u,s} &= p_c \left\| \frac{n_{u,c} e_{u,c} + r_s}{n_{u,c} + 1} \right\|_2^2 \\ &= \gamma_c \left[n_{u,c}^2 \|e_{u,c}\|_2^2 + 2 n_{u,c} \langle e_{u,c}, r_s \rangle + \|r_s\|_2^2 \right]. \end{aligned} \tag{14}$$

If the single-pair cost could be written as $C_{u,s} = a_u + b_s$, with a_u depending only on the receiver and b_s only on the sample, its four-term cross-difference would necessarily satisfy

$$\begin{aligned} &C_{j,x} + C_{k,y} - C_{j,y} - C_{k,x} \\ &= (a_j + b_x) + (a_k + b_y) - (a_j + b_y) - (a_k + b_x) \\ &= 0. \end{aligned} \tag{15}$$

Substituting equation 14 into the same cross-difference cancels the receiver-only terms $\|e_{u,c}\|_2^2$ and sample-only terms $\|r_s\|_2^2$, leaving

$$\begin{aligned} &C_{j,x} + C_{k,y} - C_{j,y} - C_{k,x} \\ &= 2n_{j,c}\gamma_c \left(\langle e_{j,c}, r_x\rangle + \langle e_{k,c}, r_y\rangle - \langle e_{j,c}, r_y\rangle - \langle e_{k,c}, r_x\rangle\right) \\ &= \frac{2n_{j,c}p_c}{(n_{j,c}+1)^2} \langle e_{j,c} - e_{k,c}, r_x - r_y\rangle . \end{aligned} \tag{16}$$

Thus, whenever there exist j, k, x, y such that

$$\langle e_{j,c} - e_{k,c}, r_x - r_y\rangle \neq 0, \tag{17}$$

the necessary condition in equation 15 fails, and the single-pair cost cannot be decomposed into a receiver term plus a sample term. This proves Proposition 1.

When $n_{j,c} = n_{k,c} > 0$ and $p_c > 0$, the separable form holds if and only if all feasible choices of receivers and candidate samples satisfy

$$\langle e_{j,c} - e_{k,c}, r_x - r_y\rangle = 0. \tag{18}$$

Necessity of equation 18 follows from the cross-difference. For sufficiency, fix an arbitrary reference receiver j_0 and reference sample x_0. If equation 18 holds for all feasible choices of receivers and candidate samples, then

$$\langle e_{j,c}, r_x\rangle = \langle e_{j,c}, r_{x_0}\rangle + \langle e_{j_0,c}, r_x\rangle - \langle e_{j_0,c}, r_{x_0}\rangle. \tag{19}$$

The first term on the right varies only with the receiver; the remaining terms vary only with the sample or are constant. Substitution into equation 14 expresses the single-pair cost as $a_j + b_x$.

Condition equation 18 includes several degenerate cases: all receivers have the same within-class deviation; all candidates have the same candidate-sample offset; or differences between receiver deviations are always orthogonal to differences between candidate-sample offsets. In these cases, the receiver–sample inner product no longer induces pair-specific differences, and separate receiver and sample decisions may exactly represent the same single-pair cost. If $p_c = 0$, the weighted class cost is identically zero, giving another trivial separable case. Equal receiver counts in the proposition serve only to isolate and establish receiver–sample coupling; JAM itself does not require equal counts across receivers.

JAM's dynamic marginal-gain greedy algorithm compares changes in the overall objective, not isolated single-pair costs. For the current receiver state, the two are related by $\Delta_{q,s} = p_c\|e_{q,c}\|_2^2 - C_{q,s}$. Since the first term depends only on the receiver, $C_{q,s}$ admits a receiver-plus-sample decomposition if and only if $\Delta_{q,s}$ does; their cross-differences differ only in sign. Proposition 1 for the single-pair cost is therefore fully equivalent to the corresponding statement for the marginal gain in equation 10.

The inner product on the proposition's right-hand side captures the coupling between receiver deviations and candidate-sample offsets. Appendix A.3 further examines how fluctuations in a single-environment sample mean affect greedy candidate-gain differences.

Stable marginal-gain expansion. Fix sender i, round t, and greedy step $r+1$, with current selected set $A_i^{t,(r)}$. Use the temporary receiver state $\widetilde{n}_{j,c}, \widetilde{\mu}_{j,c}$ from Algorithm 1 and e, r_x, e^+ from equation 9. Sending candidate x of public class c to receiver j changes only the term for receiver j and class c in equation 7; all other receiver and class terms cancel. The overall objective difference before and after adding this pair is therefore

$$\begin{aligned} \Delta_{i,j,x}^{t,(r)} &= \Phi_i^t\Big(A_i^{t,(r)}\Big) - \Phi_i^t\Big(A_i^{t,(r)} \cup \{(j,x)\}\Big) \\ &= p_c^t \left(\|e\|_2^2 - \|e^+\|_2^2\right), \end{aligned} \tag{20}$$

This is equation 10 in the main text. By equation 9, the post-sharing within-class deviation can be written as

$$e^+ = e + \delta, \qquad \delta = \frac{r_x - e}{\widetilde{n}_{j,c} + 1}. \tag{21}$$

Expanding its squared norm gives

$$\begin{aligned}\|e^+\|_2^2 &= \|e+\delta\|_2^2 \\ &= \|e\|_2^2 + 2\langle e,\delta\rangle + \|\delta\|_2^2 \\ &= \|e\|_2^2 + \frac{2\langle e, r_x - e\rangle}{\widetilde{n}_{j,c}+1} + \frac{\|r_x-e\|_2^2}{(\widetilde{n}_{j,c}+1)^2}.\end{aligned} \tag{22}$$

Substituting equation 22 into equation 20 yields

$$\Delta_{i,j,x}^{t,(r)} = p_c^t \left[-\frac{2\langle e, r_x - e\rangle}{\widetilde{n}_{j,c}+1} - \frac{\|r_x-e\|_2^2}{(\widetilde{n}_{j,c}+1)^2} \right], \tag{23}$$

Further expanding the inner product and squared norm gives the alternative form

$$\Delta_{i,j,x}^{t,(r)} = \frac{p_c^t}{(\widetilde{n}_{j,c}+1)^2} \left[(2\widetilde{n}_{j,c}+1)\|e\|_2^2 - 2\widetilde{n}_{j,c}\langle e, r_x\rangle - \|r_x\|_2^2 \right]. \tag{24}$$

Equation equation 20 shows that a positive marginal gain corresponds to a decrease in equation 7 at the current step. Equations equation 23 and equation 24 are equivalent expressions for the same marginal gain. The implementation uses the former to avoid subtracting nearly equal squared norms; the latter facilitates analyzing the contribution of each term to the score.

A.3 Derivation and Interpretation of Summary SNR

Starting from the weighted squared deviation in the JAM objective, we decompose it into an environmental mean shift and finite-sample mean estimation error to define class–environment SNR. We then relate this ratio to greedy candidate rankings and aggregate it at the average naturally arriving sample count to obtain dataset-level Summary SNR. Fix receiver j, class c, and environment h, and consider $n_{j,c,h} > 0$ samples $u_1, \ldots, u_{n_{j,c,h}}$ drawn independently and identically from this class–environment distribution. With the frozen feature extractor ϕ_0 of G_0, their features are $z_\ell = \phi_0(u_\ell) \in \mathbb{R}^d$, where d is the output dimension of ϕ_0. These random vectors have finite second moments. Define their conditional mean and covariance as $\mu_{c,h} = \mathbb{E}[z_\ell \mid c,h]$ and $\Sigma_{c,h} = \mathrm{Cov}(z_\ell \mid c,h)$, both fixed population parameters of the conditional distribution. The corresponding sample mean is

$$\mu_{j,c,h} := \frac{1}{n_{j,c,h}} \sum_{\ell=1}^{n_{j,c,h}} z_\ell. \tag{25}$$

Both $\mu_{j,c,h}$ and individual features lie in $\mathbb{R}^d$, so $\mathrm{Cov}(\mu_{j,c,h})$ and $\Sigma_{c,h}$ are $d \times d$ matrices. In the main text, $\mu_{j,c}$ and $n_{j,c}$ still denote the feature average and count over all class-c data at the receiver.

Treat the class-c feature average μ_c in the network reference as fixed, and let $\delta_{c,h} = \mu_{c,h} - \mu_c$ denote the environmental mean shift. Write the sample-minus-population mean difference $\mu_{j,c,h} - \mu_{c,h}$ as $\varepsilon_{j,c,h}$, the estimation error of the within-environment sample mean. Following the within-class deviation definition in equation 9, denote this single-environment sample mean's deviation from the network reference by $e_{j,c,h}$:

$$\begin{aligned}\varepsilon_{j,c,h} &:= \mu_{j,c,h} - \mu_{c,h}, \\ e_{j,c,h} &:= \mu_{j,c,h} - \mu_c \\ &= (\mu_{c,h} - \mu_c) + (\mu_{j,c,h} - \mu_{c,h}) = \delta_{c,h} + \varepsilon_{j,c,h}.\end{aligned} \tag{26}$$

Sample independence makes cross-covariances between distinct samples zero. Since $\varepsilon_{j,c,h}$ subtracts a fixed population mean from the sample mean, the two have the same covariance. Hence,

$$\begin{aligned}\mathbb{E}[\varepsilon_{j,c,h}] &= 0, \\ \mathrm{Cov}(\mu_{j,c,h}) &= \frac{1}{n_{j,c,h}^2} \sum_{\ell=1}^{n_{j,c,h}} \mathrm{Cov}(z_\ell \mid c,h) = \frac{\Sigma_{c,h}}{n_{j,c,h}}, \\ \mathrm{Cov}(\varepsilon_{j,c,h}) &= \mathrm{Cov}(\mu_{j,c,h} - \mu_{c,h}) = \mathrm{Cov}(\mu_{j,c,h}), \\ \mathbb{E}\|\varepsilon_{j,c,h}\|_2^2 &= \mathrm{tr}(\mathrm{Cov}(\varepsilon_{j,c,h})) = \frac{\mathrm{tr}(\Sigma_{c,h})}{n_{j,c,h}}.\end{aligned} \tag{27}$$

When receiver j's class-c summary in equation 7 is the mean and count $(\mu_{j,c,h}, n_{j,c,h})$ from environment h, the weighted squared deviation before adding a candidate is $p_c\|e_{j,c,h}\|_2^2$. For a fixed class weight $p_c > 0$, substituting equation 26 and using equation 27 gives

$$\begin{aligned}\mathbb{E}\big[p_c\|e_{j,c,h}\|_2^2\big] &= p_c\Big[\|\delta_{c,h}\|_2^2 + 2\langle\delta_{c,h}, \mathbb{E}[\varepsilon_{j,c,h}]\rangle + \mathbb{E}\|\varepsilon_{j,c,h}\|_2^2\Big] \\ &= p_c\left[\|\delta_{c,h}\|_2^2 + \frac{\mathrm{tr}(\Sigma_{c,h})}{n_{j,c,h}}\right].\end{aligned} \tag{28}$$

This decomposition shows that the squared deviation used by JAM contains both the signal energy of the environmental mean shift and finite-sample mean estimation noise. Taking their ratio to measure environmental identifiability and canceling the common weight p_c, we define the class–environment SNR as

$$\mathrm{SNR}_{c,h}(n_{j,c,h}) := \frac{\|\delta_{c,h}\|_2^2}{\mathbb{E}\|\varepsilon_{j,c,h}\|_2^2} = \frac{n_{j,c,h}\|\delta_{c,h}\|_2^2}{\mathrm{tr}(\Sigma_{c,h})}, \tag{29}$$

where $\mathrm{tr}(\Sigma_{c,h}) > 0$ is required. This ratio measures the environmental mean shift relative to fluctuations in the estimated sample mean.

This SNR characterizes mean estimation quality. We next examine how that quality affects JAM's ranking of two candidates for the same receiver in one greedy step. Fix sender i, receiver j, class weight p_c, and feasible class-c candidates x, y. Treat the network reference and candidate features as given quantities independent of the sampling above, with $r_x = z_x - \mu_c$ and $r_y = z_y - \mu_c$; the candidates may come from different environments. Equation equation 10 measures the decrease in the same objective after adding a candidate. Substituting the single-environment summary $(n_{j,c,h}, \mu_{j,c,h})$ into equation 9 and equation 10 gives the marginal gain $\Delta_{i,j,x}^{(h)}$:

$$\begin{aligned}\Delta_{i,j,x}^{(h)} &:= p_c\left[\|e_{j,c,h}\|_2^2 - \left\|\frac{n_{j,c,h}e_{j,c,h} + r_x}{n_{j,c,h}+1}\right\|_2^2\right] \\ &= \frac{p_c}{(n_{j,c,h}+1)^2}\Big[(2n_{j,c,h}+1)\|e_{j,c,h}\|_2^2 \\ &\qquad\qquad - 2n_{j,c,h}\langle e_{j,c,h}, r_x\rangle - \|r_x\|_2^2\Big].\end{aligned} \tag{30}$$

The superscript (h) identifies the pre-sharing single-environment analysis state; the updated vector in brackets is its within-class deviation after adding a candidate. Actual JAM still uses the summary $(n_{j,c}, \mu_{j,c})$ over all data of the class. Here, we examine estimation fluctuations of the same marginal-gain expression under a single-environment summary, omitting time and greedy-step indices.

Define the gain difference between these candidates as

$$\begin{aligned}M_{j,c,h}(x,y) &:= \Delta_{i,j,x}^{(h)} - \Delta_{i,j,y}^{(h)} \\ &= -\frac{p_c}{(n_{j,c,h}+1)^2}\Big[2n_{j,c,h}\langle e_{j,c,h}, r_x - r_y\rangle \\ &\qquad\qquad + \|r_x\|_2^2 - \|r_y\|_2^2\Big].\end{aligned} \tag{31}$$

The condition $M_{j,c,h}(x,y) > 0$ means that x has a larger marginal gain than y in this analysis state, so JAM ranks x ahead of y. The ranking depends on both the inner product and the difference in candidate-offset norms. With the network reference, candidate features, and sample count fixed, the norm difference does not vary with the receiver's sample mean. Receiver mean estimation error affects candidate rankings through $\langle e_{j,c,h}, r_x - r_y\rangle$. Substituting the full decomposition in equation

26 gives

$$\begin{aligned} M_{j,c,h}(x,y) = & -\frac{p_c}{(n_{j,c,h}+1)^2}\Big[2n_{j,c,h}\langle \delta_{c,h}+\varepsilon_{j,c,h}, r_x - r_y\rangle \\ & \qquad + \|r_x\|_2^2 - \|r_y\|_2^2\Big] \\ = & -\frac{p_c}{(n_{j,c,h}+1)^2}\Big[2n_{j,c,h}\langle \delta_{c,h}, r_x - r_y\rangle \\ & \qquad + \|r_x\|_2^2 - \|r_y\|_2^2\Big] \\ & -\frac{2n_{j,c,h}p_c}{(n_{j,c,h}+1)^2}\langle \varepsilon_{j,c,h}, r_x - r_y\rangle. \end{aligned} \tag{32}$$

The first complete term depends on the environmental mean shift and candidate features and remains fixed during sampling; the second is a random perturbation caused by sample mean estimation error. The gain-difference variance therefore measures how mean estimation error changes the comparison score for the same two candidates. By equation 27, the second term has zero expectation, and

$$\begin{aligned} \mathrm{Var}[M_{j,c,h}(x,y)] &= \left(\frac{2n_{j,c,h}p_c}{(n_{j,c,h}+1)^2}\right)^2 \mathbb{E}\big[\langle \varepsilon_{j,c,h}, r_x - r_y\rangle^2\big] \\ &= \left(\frac{2n_{j,c,h}p_c}{(n_{j,c,h}+1)^2}\right)^2 \frac{(r_x-r_y)^\top \Sigma_{c,h}(r_x-r_y)}{n_{j,c,h}} \\ &\le \left(\frac{2n_{j,c,h}p_c}{(n_{j,c,h}+1)^2}\right)^2 \|r_x-r_y\|_2^2 \frac{\mathrm{tr}(\Sigma_{c,h})}{n_{j,c,h}}. \end{aligned} \tag{33}$$

The final step uses positive semidefiniteness of the covariance matrix: for any vector v, $v^\top \Sigma_{c,h} v \le \|v\|_2^2 \,\mathrm{tr}(\Sigma_{c,h})$. When $p_c > 0$, $\delta_{c,h} \neq 0$, and $r_x \neq r_y$, normalizing by the norm scales of the environmental shift and candidate difference yields

$$\frac{\mathrm{Var}[M_{j,c,h}(x,y)]}{\left(\frac{2n_{j,c,h}p_c}{(n_{j,c,h}+1)^2}\right)^2 \|\delta_{c,h}\|_2^2 \|r_x-r_y\|_2^2} \le \frac{\mathrm{tr}(\Sigma_{c,h})}{n_{j,c,h}\|\delta_{c,h}\|_2^2} = \frac{1}{\mathrm{SNR}_{c,h}(n_{j,c,h})}. \tag{34}$$

This connects local SNR to JAM's candidate-gain comparisons: a larger SNR gives a smaller upper bound on the normalized gain-difference variance, indicating a clearer environmental mean shift relative to finite-sample estimation noise. These fluctuations and the candidates' underlying gain margin jointly affect ranking stability.

Equation equation 29 describes local SNR when receiver j estimates a mean from $n_{j,c,h}$ samples of class c and environment h. To obtain dataset-level Summary SNR, we first set a common sample scale n for all class–environment combinations and then aggregate their signal and noise. Experiments use the average cumulative number of naturally arriving samples per client and public class over the deployment protocol's full arrival phase:

$$n := \frac{1}{KC}\sum_{j\in\mathcal{K}}\sum_{t=0}^{T-1} |\mathcal{R}_j^t|. \tag{35}$$

Here, K and C are the numbers of clients and public classes, respectively, and $\mathcal{R}_j^t$ denotes client j's naturally arriving data in round t; rounds without natural arrivals contribute zero. This scale excludes shared samples, is common to all methods on the same dataset, and is not rounded. Replacing $n_{j,c,h}$ in equation 29 by the common scale n leaves each combination's signal energy $\|\delta_{c,h}\|_2^2$ unchanged and rescales its mean estimation noise to $\mathrm{tr}(\Sigma_{c,h})/n$. Averaging signal and noise separately with equal weights over valid class–environment combinations, then taking their ratio, gives dataset-level Summary SNR:

$$\mathrm{SNR}(n) := \frac{\mathbb{E}_{c,h}\|\delta_{c,h}\|_2^2}{\mathbb{E}_{c,h}[\mathrm{tr}(\Sigma_{c,h})/n]} = \frac{n\,\mathbb{E}_{c,h}\|\delta_{c,h}\|_2^2}{\mathbb{E}_{c,h}\,\mathrm{tr}(\Sigma_{c,h})}. \tag{36}$$

This is the Summary SNR used in the main text. Here, $\mathbb{E}_{c,h}$ denotes the uniform average over valid combinations. The standardized sample size n, determined by the average naturally arriving sample

count above, differs from the receiver-specific within-environment count $n_{j,c,h}$ and the total class count $n_{j,c}$ that includes shared data. By rescaling each environment's mean estimation noise to a common sample size within the dataset, Summary SNR reflects both environmental identifiability in feature space and the average number of naturally arriving samples. It is the ratio of average signal energy to average noise energy, not the arithmetic mean of local SNRs, and does not directly measure estimation error in actual mixed-environment summaries.

For the empirical computation, we use each dataset's experimental model G_0 and frozen feature extractor ϕ_0 to extract features from the full original training pool. In this computation, the class center μ_c is the average of all training features of that class, while $\mu_{c,h}$ and $\Sigma_{c,h}$ are estimated by the sample mean and unbiased sample covariance of the corresponding class–environment subset. Only subsets with at least 3 samples are retained and aggregated with equal weights. Means and covariances are estimated using every sample in each retained subset; n rescales the noise in equation 36 and does not require drawing another n samples. Under the arrival protocol in Appendix B.1, cumulative natural arrivals per client are $9\times32 = 288$ for PACS, $65+8\times32 = 321$ for Office-Home, and $9\times100 = 900$ for CIFAR-100. As all clients have equal arrival counts, K cancels in equation 35, giving n values of $288/7 \approx 41.1429$ for PACS, $321/65 \approx 4.9385$ for Office-Home, and $900/20 = 45$ for CIFAR-100. Office-Home has 33 more cumulative arrivals than PACS because of its first-round full-class-coverage allocation. Environment identities are used in this mechanism analysis; JAM's runtime selection still uses receiver summaries, the network reference, and candidate features. Cross-dataset comparisons of SNR and matching gains are given in Appendix D.2.

B Experimental Setup Details

B.1 Data Streams and Evaluation Distributions

We treat an environment as a latent data-generating condition that changes $P(X \mid Y)$, not specifically as a visual attribute. Hidden fine classes in CIFAR-100 and visual domains in PACS and Office-Home provide controlled instances of this condition. Each client receives naturally arriving data $\mathcal{R}_i^t$ in round t. CIFAR-100 provides 100 samples per client per round; PACS provides 32, split equally between Old and Seen in rounds 0–3 and in Old/Seen/Deficit proportions of $50/25/25\%$ in rounds 4–8. Office-Home follows the PACS schedule in rounds 1–8, but receives 65 rather than 32 samples in round 0: one Seen sample for each of 49 of the 65 public classes and one Old sample for each of the remaining 16 classes. These 16 Old slots are assigned to classes with the fewest Seen candidates. With only 32 arrivals per round, many of the 65 public classes would be absent from many clients' data in the early rounds. Because JAM compares per-class statistics in receiver summaries with the network reference, an unobserved class has no receiver summary, degrading selection and confounding method performance with missing-class effects. Allowing one sample per class in round 0 ensures that every client observes every public class from the start. JAM always compares fully covered classes, with positive per-class sample counts $n_{j,c}^t$ in receiver summaries. Per-class quotas likewise ensure full class coverage in round 0 for CIFAR-100 and PACS. No natural data arrive in round 9. Clients train on cumulative naturally arriving data and previously received shared data available for training, but transmission candidates come only from the sender's current-round natural arrivals. Samples selected in a round become available before training starts in the next round. Environment identities are used only to generate streams, construct evaluation distributions, and compute Summary SNR for mechanism analysis. All deployable methods access only samples, public classes, arrival rounds, and model-generated representations.

To form fixed, non-overlapping evaluation distributions, we partition the held-out test set by public class c and environment h into class–environment test slices:

$$\mathcal{T}_{c,h} = \{(x, y) : y = c,\ H(x) = h\}, \tag{37}$$

Here, $H(x)$ is the environment identity used in the experimental construction. All test slices exclude training and shared samples. Table 4 gives the mappings for the three datasets. For each client, Seen is the new environment first encountered naturally after deployment, and Deficit is the new environment already observed by the other group but not naturally encountered by that client until round 4. After round 4, the Deficit label continues to identify the same fixed test slice to track performance before and after its first natural arrival.

Table 4: Old, Seen, and Deficit environments in each dataset.

Dataset and clients	Old	Seen	Deficit
CIFAR-100, clients 0–4	hidden groups 0–2	hidden group 3	hidden group 4
CIFAR-100, clients 5–9	hidden groups 0–2	hidden group 4	hidden group 3
PACS, clients 0–3	art painting, cartoon	photo	sketch
PACS, clients 4–7	art painting, cartoon	sketch	photo
Office-Home, clients 0–3	Art, Clipart	Product	Real-World
Office-Home, clients 4–7	Art, Clipart	Real-World	Product

The *Old distribution* comprises the empirical distributions of all old-environment test slices. The *Seen distribution* includes only the new environment first encountered naturally by the client after deployment. The *Novel distribution* includes both Seen and Deficit: CIFAR-100 and Office-Home use the full test distribution obtained by combining the two new environments, while PACS evaluates photo and sketch separately and averages the two domains equally. Deficit Pre, First, and Final metrics are evaluated in rounds 3, 4, and the final round, respectively. Round 3 is the last evaluation before natural arrival; its relevant supervision comes from shared samples received in rounds 0–2 and already available for training. Round 4 follows the first natural arrival and the corresponding local training. The Novel, Deploy, Seen, and Old columns in the main table report average loss over rounds 1–9 on the respective distributions.

Beyond the main data stream, the robustness analysis in Section 5.4 uses a staggered CIFAR-100 variant, Staggered 2:8. The 10 clients form complementary groups in a 2:8 ratio: group A has 2 clients whose Deficit environment is hidden group 4; group B has 8 whose Deficit environment is group 3. Rather than fixing every Deficit arrival at round 4, each client's arrival round τ is assigned from $\{2, 2, 3, 3, 4, 4, 5, 5, 6, 6\}$. Before round τ, each round brings 50 Old and 50 Seen samples; from that round onward, it brings 50 Old, 25 Seen, and 25 Deficit samples. Total arrivals per round match the main protocol. Round-wise Deploy distribution weights retain the main schedule and are not reset according to τ.

B.2 Dynamic Deployment Distribution

The Deploy distribution does not alter the training stream; it specifies only how fixed test slices are combined each round. Let $a^t_{j,h}$ be client j's target weight for environment h in round t, and let $m_c = \sum_{h'} |\mathcal{T}_{c,h'}| / \sum_{c',h'} |\mathcal{T}_{c',h'}|$ be the fraction of samples in the full held-out test set belonging to public class c. The final class–environment test-slice weight is

$$w^t_{j,c,h} = m_c \frac{a^t_{j,h}}{\sum_{h':\, \mathcal{T}_{c,h'} \neq \varnothing} a^t_{j,h'}}, \tag{38}$$

with weights assigned only to nonempty slices. Environment proportions thus vary over time, while each public class retains its sample proportion in the full held-out test set as its total weight. Deploy Loss and Accuracy in each round use the same weights $w^t_{j,c,h}$.

Table 5 gives the complete environment-level weight schedule. Old probability mass is split equally among hidden groups 0, 1, and 2 in CIFAR-100, and between the two old domains in PACS and Office-Home. Each client's Seen and Deficit environments follow Table 4.

B.3 Training and Baseline Protocols

CIFAR-100 uses 10 clients, 20 superclasses, and a randomly initialized ResNet-18. PACS and Office-Home use 8 clients and ImageNet-initialized ResNet-18 models predicting 7 and 65 classes, respectively. Each dataset's G_0 is then trained for five rounds using only Old data. Adaptation lasts ten rounds, with each client performing 4 supervised updates per round at batch size 64. In the MobileNetV2 robustness experiments, 5 pretraining rounds from random initialization still represent an early training stage, so training of CIFAR-100's G_0 is extended to 15 rounds. ImageNet-initialized PACS retains the 5 rounds used for ResNet. Learning rates are 0.01 for CIFAR-100 and 0.001 for PACS and Office-Home; the optimizer is SGD with momentum 0.9 and weight decay 10^{-5}. Sharing

Table 5: Environment weights (%) used to construct the round-wise Deploy distribution. O, S, and D denote Old, Seen, and Deficit, respectively.

	CIFAR-100			PACS / Office-Home		
Round	O	S	D	O	S	D
0	85	15	0	50	50	0
1	80	20	0	50	50	0
2	75	25	0	50	50	0
3	70	25	5	50	50	0
4	65	20	15	50	25	25
5	60	20	20	50	25	25
6	55	22.5	22.5	50	25	25
7	50	25	25	50	25	25
8	45	27.5	27.5	50	25	25
9	45	27.5	27.5	50	25	25

occurs in rounds 0–8, and round 9 observes the effect of data received from round 8. All comparisons share the same G_0, natural-arrival sequence, current-round candidate pool, update count, and timing of shared-data availability. JAM summaries use penultimate-layer features from frozen G_0 with identical fixed input processing across clients and no random data augmentation. The arrival protocol in Appendix B.1 covers all public classes from round 0, so per-class sample counts in receiver summaries remain positive. Received samples are not forwarded. When feasible pairs are sufficient, all methods fill the same sending dose.

Neutral Random distributes each sender's budget as uniformly as possible among other receivers using a round-dependent round-robin schedule; any two receivers' quotas differ by at most one sample within a round. Public-class slots also cycle across rounds, and samples are drawn randomly from the corresponding current-round candidates. Entropy (Settles, 2009), FedCS (Hao et al., 2025), and SES (Xie et al., 2025) retain this uniform round-robin budget and replace only sender-side sample ranking. JAM instead computes marginal gains for specific receiver–sample pairs under a common objective. FedProto shares no samples. After local training in each round, each client uses its current model and currently available data to compute within-class mean prototypes at the classification head's input and sends them to every other client. These prototypes come from each client's continually updated local model, not the reference features z_x of frozen G_0. For each class, a global prototype is obtained by uniformly averaging over all clients holding a prototype for that class, including the client itself. In the next round, local training adds prototype-distance regularization with weight λ=1 to cross-entropy, following the original FedProto formulation. The regularizer copies a batch's features, replaces each copied feature with its class's global prototype when available, and minimizes the mean squared difference between the copy and original features. Samples without a global class prototype remain unchanged and contribute nothing to this term. With no prototypes in the first round, training uses cross-entropy alone. Inference still uses the local classification head; prototypes are not used for prediction. Frozen G_0 retains the initial model, Local-U0 uses only local naturally arriving data, TENT-style-U0 updates only normalization affine parameters, and EWC-style-U0 constrains parameters relative to the old-environment model during supervised updates. These methods do not access environment identities.

SES and CF-CL each have a demand-based outer-allocation variant, Need-SES and Need-CF-CL. Within each baseline pair, only the source of the outer quotas differs; sample-selection rules inside the quotas are identical. For SES versus Need-SES, SES passes uniform round-robin quotas to its structural-entropy selector. Need-SES changes only receiver quotas. It first aggregates each receiver's per-class deviations into scalar need $d_j = \big(\sum_c p_c \|\mu_{j,c} - \mu_c\|_2^2\big)^{1/2}$, summing only over observed classes, with p_c denoting network-reference class proportions. It then allocates the sender budget greedily, one unit at a time, to the receiver with the largest decrease in the proxy $V_j(s) = \big(n_j/(n_j{+}s)\big)^2 d_j^2$, where n_j is the receiver's current sample count and s its allocation so far in the round. Quotas are thus integers by construction. Ties or all-zero needs are resolved by a seed-derived round-robin order; the dose is always filled when feasible capacity permits. Class composition follows Neutral Random's cyclic class schedule. SES determines only which samples to send, and seed-derived random matching assigns them to receivers. For CF-CL versus Need-CF-CL, selection uses receiver representative points as references. Sender current-round candidates and

receiver representative points are clustered jointly. The Macro stage allocates sampling budget by each cluster's candidate-to-representative count ratio; the Micro stage scores candidates by triplet distance to representatives and samples probabilistically within clusters. Both variants use the same receiver–sample pair probability table. CF-CL retains uniform round-robin outer quotas, whereas Need-CF-CL uses the same demand-based allocation function as Need-SES, recomputed each round from its own current state. Each baseline pair thus isolates whether receiver-aware budget allocation itself provides gains.

CF-CL and Need-CF-CL select representative points from receivers' cumulative naturally arriving data, excluding received shared samples. Representatives are rebuilt before sharing each round and then held fixed within that round. Scoring uses the frozen G_0 features common to all clients, whereas the original method uses embeddings progressively aligned through contrastive training. Candidates come only from the sender's current-round natural arrivals, and receivers train through ordinary supervised updates. This adaptation retains the original two-stage Macro/Micro receiver-conditioned sampling and its parameter settings: a representative-set cap of 500, 4 clusters, and sampling temperature λ increasing linearly from 4 to 10. The triplet margin is set to 0; the original paper does not specify this value; server aggregation is omitted. Representative information is charged as control metadata rather than treated as free information.

Statistical analysis. We analyze statistical significance using the existing five-seed main experiments, with seeds 40, 42, 44, 45, and 46. Each complete seeded run is one statistical observation, yielding five observations; JAM–baseline differences are computed within each seed. For each dataset, the five runs share a fixed G_0, initial partition, and environment-arrival slots. Seeds vary sample order within slots and training randomness. Whole-process metrics use equal averaging over rounds 1–9 and client macro-averages; Pre, First, and Final use rounds 3, 4, and 9, respectively. All five seeds are retained, using unrounded per-seed metrics.

For a fixed dataset, metric, and baseline, let d_s be the improvement for seed s ($s = 1, \ldots, 5$): baseline minus JAM for loss, and JAM minus baseline for accuracy, so positive values indicate improvement. Accuracy differences are reported in percentage points (pp), not relative percentage improvements. Let $\bar{d} = \frac{1}{5}\sum_s d_s$ and $s_d^2 = \frac{1}{4}\sum_s (d_s - \bar{d})^2$. Assuming independent, approximately normally distributed differences across seeds, the two-sided paired t-test compares $H_0 : \mathbb{E}[d_s] = 0$ against $H_1 : \mathbb{E}[d_s] \neq 0$, with statistic $\bar{d}/(s_d/\sqrt{5})$ and 4 degrees of freedom. The 95% confidence interval is $\bar{d} \pm t_{0.975,4} s_d/\sqrt{5}$, where $t_{0.975,4}$ is the 97.5% quantile of the t distribution with 4 degrees of freedom. All intervals are individual, unadjusted for multiple comparisons. Numerical calculations use SciPy 1.18.0.

New-environment analysis focuses on whole-process Novel Loss, Pre Accuracy, and Final Accuracy. For each metric, comparisons of JAM with Neutral Random, FedCS, SES, Need-SES, Need-CF-CL, and FedProto across three datasets form a family of $3 \times 6 = 18$ tests. We apply Holm correction and declare significance at adjusted $p < 0.05$; significant improvement also requires $\bar{d} > 0$. Main comparisons use all five seeds with these tests. Mechanism, robustness, and communication analyses use three common seeds (40, 42, 44) and report means and sample standard deviations.

To examine whether new-environment adaptation accompanies deterioration on old environments, we compare JAM with the pre-deployment model G_0 and Need-SES, reporting whole-process and final-round Old Loss and Accuracy differences with 95% confidence intervals. The former measures changes from initial old-environment capabilities; the latter measures trade-offs against a strong adaptation baseline.

C Communication Protocol and Accounting

C.1 Exchange Convention

The main setting uses all-to-all exchange. Each client sends its local per-class statistics to the other $K - 1$ clients so that all clients construct the same network reference; each receiver summary is also sent to potential senders. Each of the two statistic types thus produces $K(K - 1)$ directed copies. Communication cost is the sum of sent and received payloads across the network divided by K, i.e., average bidirectional payload per client per effective communication round. Directed

Table 6: Complete per-client, per-round communication ledger on CIFAR-100.

Method	Representation/model (B)	Raw samples (B)	Total (B)	Total (MiB)
JAM-FP32	1,480,320	98,304	1,578,624	1.505493
JAM-BF16	743,040	98,304	841,344	0.802368
JAM-INT8	385,920	98,304	484,224	0.461792
JAM-INT4	201,600	98,304	299,904	0.286011
JAM-INT2	109,440	98,304	207,744	0.198120
FedProto-style	738,720	0	738,720	0.704498
Full-model exchange (all peers)	806,141,664	0	806,141,664	768.796600
Full-model exchange (one neighbor)	89,571,296	0	89,571,296	85.421844

sample-receipt counts may differ across clients, so sample payloads are likewise summed across the network before division by the number of clients.

Let $B_{\mathrm{sum}}(q)$ denote the encoded payload of one client's local per-class statistics or one receiver summary at precision q; both have the same per-class feature-average/count structure. Let B_{sample}, B_{proto}, and B_{model} be the payloads of one raw sample, one set of class prototypes, and one full model, respectively, and let b_i^t be the sending dose actually transmitted by sender i. Then

$$\begin{aligned} B_{\mathrm{JAM}}^t(q) &= 4(K-1)B_{\mathrm{sum}}(q) + \frac{2}{K}\sum_i b_i^t B_{\mathrm{sample}}, \\ B_{\mathrm{Proto}} &= 2(K-1)B_{\mathrm{proto}}, \\ B_{\mathrm{Model,all}} &= 2(K-1)B_{\mathrm{model}}, \qquad B_{\mathrm{Model,one}} = 2B_{\mathrm{model}}. \end{aligned} \tag{39}$$

The formal comparisons use $b_i^t = U$. FedProto and model-exchange costs are also computed per effective communication round. The ledger counts payload only, excluding protocol headers, routing, retransmission, encryption, and medium-access overhead. A temporary coordinator could aggregate local per-class statistics and broadcast the network reference to reduce its copy count; all tables use the more direct all-to-all main setting.

C.2 Payload Encoding and Complete Ledger

Local per-class statistics and receiver summaries each contain $C \times d$ feature-average elements and C 64-bit counts. FP32 and BF16 use direct encoding. INT8, INT4, INT2, and INT1 use symmetric block quantization with 64 elements per block, one additional BF16 scale per block, and bit-packed low-bit values. With block size $s = 64$, the payload is

$$B_{\mathrm{sum}}(q) = \begin{cases} Cdq/8 + 8C, & q \in \{32, 16\}, \\ \lceil Cdq/8 \rceil + 2C\lceil d/s \rceil + 8C, & q \in \{8, 4, 2, 1\}. \end{cases} \tag{40}$$

Quantization applies only to per-class feature averages in exchanged control metadata; local candidate features and training computations remain FP32. CIFAR-100 uses $K = 10, C = 20, d = 512, U = 16$, with payloads of 3,072 B per raw image, 41,040 B per set of class prototypes, and 44,785,648 B for ResNet-18 state tensors. Table 6 gives the unrounded ledger automatically generated from equation 39–equation 40; its totals define the horizontal axis in Figure 3. Even full-model exchange with only one neighbor requires 85.42 MiB (Lian et al., 2018), two orders of magnitude above any summary or sample payload.

C.3 Quantization Stability and Cross-Dataset Budget Equivalence

The preceding ledgers quantify communication at each encoding precision; we now examine how much of JAM's decisions and performance is retained after compression. Using JAM-FP32 as the reference, quantization experiments compare complete trajectories on CIFAR-100 at $U = 16$ over three common seeds. The first three metrics in Table 7 measure agreement with FP32 at progressively finer levels. *Receiver budget* measures retained sender-budget allocation, i.e., whom to send to and how much: for each sender–receiver link, it counts the overlapping transmission count under the two encodings, then expresses the sum as a fraction of all FP32 transmissions. *Receiver–class* additionally compares per-class transmission counts within the same sending dose, capturing changes

in class composition even when receiver totals coincide. *Samples* measures the finest overlap: identical round–sender–receiver–sample transmissions as a fraction of FP32 transmissions. *Spearman* ρ helps explain these changes. For all candidates in each sender–receiver–class group, FP32 and the quantized encoding produce JAM pair scores; Spearman rank correlation compares only score order, not magnitude (1 means identical order, 0 means uncorrelated order), and group correlations are averaged by candidate count. The effects increase as precision decreases. Under BF16 and INT8, about 94%–96% of receiver-budget allocations agree with FP32, and rankings barely change ($\rho \approx 0.99$). INT4 starts to change decisions (overlap rates about 64%–81%, $\rho = 0.87$), but largely retains new-environment adaptation. Under INT2, overlap in specific sample selections falls to about 24% ($\rho = 0.59$), yet performance remains close to FP32. Quantized scores still direct transmissions toward the same receivers and high-quality candidates; they mainly change choices among similarly scored candidates rather than reduce selection to random sampling. INT1 nearly scrambles score rankings ($\rho = 0.20$), causing JAM's selection mechanism and performance to break down.

Table 7: Decision stability relative to JAM-FP32 on CIFAR-100.

Encoding	Receiver budget (%)	Receiver–class (%)	Samples (%)	Spearman ρ
BF16	96.1 ± 0.5	95.3 ± 0.9	95.1 ± 0.9	0.991 ± 0.003
INT8	95.5 ± 1.6	94.4 ± 1.8	94.2 ± 1.7	0.986 ± 0.005
INT4	81.1 ± 1.6	67.5 ± 2.3	64.4 ± 2.1	0.867 ± 0.002
INT2	65.4 ± 1.5	30.0 ± 3.4	24.4 ± 2.9	0.589 ± 0.008
INT1	46.3 ± 1.6	9.3 ± 2.3	4.5 ± 1.8	0.202 ± 0.013

While Table 7 measures decision agreement, Table 8 reports task performance for every point in Figure 3. Both metrics follow the main-text definitions, with each value reported as mean $\pm$ sample standard deviation over the three common seeds (40/42/44). INT4 increases Novel Loss by only 0.0041 and reduces Pre Accuracy by 0.22 pp relative to FP32. INT1 performs worse than Neutral Random ($U = 16$), which sends no control metadata, consistent with the collapse in ranking agreement in Table 7.

Table 8: Task performance of every point in Figure 3 on CIFAR-100 over the three common seeds (40/42/44), reported as mean $\pm$ sample standard deviation. Novel Loss is the rounds-1–9 mean of the per-round client macro-average on the Novel distribution; Pre A is the round-3 client macro-average accuracy on the Deficit test slice (percent). Payload follows the convention of Table 6; the JAM-INT1 row is computed from the same formula and is absent from the ledger table.

Method	Total payload (B)	Novel Loss	Pre A (%)
JAM-FP32	1,578,624	2.5282 ± 0.0020	21.63 ± 0.09
JAM-BF16	841,344	2.5276 ± 0.0046	21.54 ± 0.14
JAM-INT8	484,224	2.5269 ± 0.0029	21.63 ± 0.09
JAM-INT4	299,904	2.5323 ± 0.0029	21.41 ± 0.26
JAM-INT2	207,744	2.5306 ± 0.0053	21.53 ± 0.34
JAM-INT1	161,664	2.5710 ± 0.0019	20.49 ± 0.24
Neutral Random ($U = 16$)	98,304	2.5485 ± 0.0037	20.88 ± 0.14
Random-U48	294,912	2.5354 ± 0.0048	21.35 ± 0.29
Local-U0	0	2.5555 ± 0.0036	20.61 ± 0.03
FedProto	738,720	2.5400 ± 0.0035	20.59 ± 0.08

The same encoding and accounting formulas give cross-dataset upper bounds on the extra sample budget equivalent to control metadata. Define $\Delta U = \left\lfloor B_{\mathrm{ctrl}}/(2B_{\mathrm{sample}}) \right\rfloor$, where B_{ctrl} is total control metadata per client per round at a given precision, and B_{sample} is the per-sample payload under the raw-pixel serialization convention of Table 6. The main text evaluates ΔU at INT4, the recommended operating point on the compression trajectory. The definition tests whether spending the same communication budget on raw samples rather than control metadata is preferable, without requiring a particular encoding precision. Table 9 gives the results: a CIFAR-100 sample costs only 3,072 B, so INT4 control metadata is equivalent to 32 additional transmissions. Random-U48 in Figure 3 provides this approximately matched total-communication comparison. PACS and Office-Home use 224×224 images with payloads of 150,528 B per sample; the corresponding control

metadata converts to only 0 and 1 extra samples, respectively, making such a comparison unavailable for these two datasets.

Table 9: INT4 control-metadata cost in equivalent extra samples, computed from equation 39–equation 40 under the unified serialization convention. For Office-Home, the control budget converts to only one extra sample, and no control experiment is run there.

Dataset	B_{ctrl} (B)	B_{sample} (B)	Base U	ΔU	Approx. byte-matched comparison
CIFAR-100	201,600	3,072	16	32	Random-U48 (run)
PACS	54,880	150,528	5	0	none (subsample infeasible)
Office-Home	509,600	150,528	5	1	at most one sample; not run

D Additional Results and Diagnostics

D.1 Seed-Level Statistical Analysis

We report five-seed results following Appendix B.3. Table 10 gives means and sample standard deviations for the main table's Novel Loss. Tables 11 and 12 report JAM's paired improvements over each baseline, individual 95% confidence intervals, two-sided paired t-test p values, and per-metric Holm-adjusted p_{Holm}. Here, ΔLoss is baseline minus JAM, and ΔA is JAM minus baseline; positive values indicate improvement. Accuracy differences use percentage points (pp).

Table 10: Novel Loss over five paired random seeds.

Method	CIFAR-100	PACS	Office-Home
Frozen G_0	2.8118 ± 0.0000	0.9230 ± 0.0000	3.5382 ± 0.0000
Local-U0	2.5590 ± 0.0054	0.8031 ± 0.0020	3.4143 ± 0.0013
Neutral Random	2.5492 ± 0.0028	0.7957 ± 0.0028	3.4134 ± 0.0010
SES	2.5475 ± 0.0051	0.7942 ± 0.0020	3.4138 ± 0.0010
Need-SES	2.5459 ± 0.0064	0.7940 ± 0.0021	3.4134 ± 0.0009
Need-CF-CL	2.5488 ± 0.0035	0.7969 ± 0.0019	3.4148 ± 0.0020
FedProto	2.5434 ± 0.0053	0.7996 ± 0.0021	3.4135 ± 0.0014
JAM	$\mathbf{2.5303 \pm 0.0034}$	$\mathbf{0.7761 \pm 0.0013}$	$\mathbf{3.4132 \pm 0.0015}$

Whole-process Novel Loss improvements over all six baselines on CIFAR-100 and PACS remain significant after per-metric Holm correction. All six improvements on Office-Home are on the order of 10^{-3} and are not significant. The largest mean improvement is over Need-CF-CL (0.00164), with adjusted $p = 0.8617$. Whole-process baseline performance on this dataset is essentially indistinguishable, consistent with its weakest within-class deviation signal (Appendix D.2).

For Pre Accuracy before the Deficit environment's natural arrival, improvements over Neutral Random, FedCS, SES, and FedProto on CIFAR-100 and PACS remain significant after per-metric Holm correction. Improvements over the receiver-aware baselines Need-SES and Need-CF-CL are smaller and not significant, indicating that demand-based allocation itself narrows the early-adaptation gap. Final Accuracy improvements remain significant over all six baselines on CIFAR-100 and PACS, whereas every Office-Home interval crosses zero. All intervals and tests in Table 12 concern paired absolute accuracy differences.

Beyond new-environment gains, Table 13 examines changes in Old performance. Old data continue to arrive during deployment, accounting for about half of each round's natural arrivals. These results therefore measure old-environment retention under the streaming protocol, not forgetting control without re-observing old data. Whole-process averages rounds 1–9; Final denotes the last round. Each period compares JAM with the pre-deployment model G_0 and Need-SES, using the same difference directions as above and individual 95% confidence intervals.

Relative to the pre-deployment model, whole-process Old Loss improvement intervals lie entirely above zero on all three datasets, and final-round results retain this direction. Whole-process Old Accuracy improves on CIFAR-100 and Office-Home, while its interval crosses zero on PACS. Relative to Need-SES, whole-process Old Loss is lower on CIFAR-100 and PACS. On Office-Home, the

Table 11: Whole-process Novel Loss improvements of JAM over sharing baselines.

Dataset	Baseline	ΔLoss	95% CI	p	p_{Holm}
CIFAR-100	Neutral Random	0.01890	[0.01328, 0.02451]	7.30×10^{-4}	0.0080
	FedCS	0.02532	[0.01757, 0.03306]	8.16×10^{-4}	0.0082
	SES	0.01718	[0.00944, 0.02492]	0.0035	0.0283
	Need-SES	0.01562	[0.00832, 0.02293]	0.0040	0.0283
	FedProto	0.01313	[0.00739, 0.01886]	0.0031	0.0283
	Need-CF-CL	0.01851	[0.01525, 0.02177]	9.46×10^{-5}	0.0011
PACS	Neutral Random	0.01964	[0.01767, 0.02162]	1.02×10^{-5}	0.0002
	FedCS	0.01723	[0.01549, 0.01897]	1.04×10^{-5}	0.0002
	SES	0.01818	[0.01571, 0.02066]	3.43×10^{-5}	0.0004
	Need-SES	0.01790	[0.01645, 0.01936]	4.34×10^{-6}	0.0001
	FedProto	0.02349	[0.02173, 0.02526]	3.22×10^{-6}	0.0001
	Need-CF-CL	0.02080	[0.01889, 0.02271]	7.14×10^{-6}	0.0001
Office-Home	Neutral Random	0.00029	[-0.00089, 0.00147]	0.5347	1.0000
	FedCS	0.00057	[-0.00060, 0.00174]	0.2454	1.0000
	SES	0.00061	[-0.00055, 0.00177]	0.2153	1.0000
	Need-SES	0.00026	[-0.00076, 0.00128]	0.5148	1.0000
	FedProto	0.00033	[-0.00039, 0.00106]	0.2684	1.0000
	Need-CF-CL	0.00164	[-0.00087, 0.00416]	0.1436	0.8617

whole-process Old Loss difference is near zero with an interval crossing zero, and final-round Old Accuracy is lower than Need-SES's but still higher than G_0's. Thus, new-environment gains accompany lower average old-environment loss than the initial model, alongside local accuracy trade-offs relative to other adaptation methods.

The main table reports only three endpoints directly tied to early adaptation and deployment performance. Table 14 gives all remaining endpoints and methods: Deficit Pre, First, and Final use rounds 3, 4, and 9, respectively, and Whole-process columns average rounds 1–9. Evaluation times and distributions are defined in Appendix B.1. Baselines such as TENT-style-U0, EWC-style-U0, and FedProto are adapted to our setting. They share the same frozen G_0 and supervised-update protocol as JAM. Sample-sharing methods also use the same current-round candidates, retaining their original designs only for sample selection, exchanged content, or local-parameter updates. These comparisons thus evaluate selection mechanisms or update rules rather than the original complete training pipelines; adaptations are detailed in Appendix B.3.

D.2 Allocation–Matching and SNR Diagnostics

Here, Summary SNR is a mechanism-analysis measure: it uses experimental environment identities to characterize finite-sample identifiability of receiver within-class deviations and does not participate in JAM selection. Table 2 summarizes whole-process improvements and pre-arrival transmission behavior of the mechanism variants. This appendix supplies the full evidence: receiver-budget matrices at cell level (Figure 4), complete endpoint improvements over Neutral Random (Table 15), and full-process and single-round pair-permutation replay results for JAM-Permuted (Table 17). The behavioral metric R_{alloc} compares complementary links between the two groups with same-group links; this grouping is used only for post hoc statistics. Let B_{cross} and B_{same} be actual transmission counts on these link types, and N_{cross} and N_{same} their respective numbers of directed links. Then

$$R_{\text{alloc}} = \frac{B_{\text{cross}}/N_{\text{cross}}}{B_{\text{same}}/N_{\text{same}}}, \tag{41}$$

A ratio $R_{\text{alloc}} = 1$ means that complementary and same-group links receive the same average budget per link.

The budget matrices in Figure 4 are derived from individual transmission records. Each cell gives the percentage of a sender's budget directed to that receiver, averaged over rounds 0–3 and three seeds; orange boundaries separate the two client groups. Neutral Random is close to a uniform $1/9$ allocation per neighbor, whereas JAM allocates a larger share to receivers across complementary links. O, S, and D denote the fixed Old, Seen, and Deficit environment roles, respectively.

Table 12: Deficit-environment accuracy improvements before natural arrival and at the final round.

Dataset	Baseline	ΔA (pp)	95% CI	p	p_{Holm}
Pre A					
CIFAR-100	Neutral Random	0.693	[0.390, 0.996]	0.0032	0.0410
	FedCS	0.721	[0.401, 1.041]	0.0033	0.0410
	SES	0.804	[0.423, 1.185]	0.0042	0.0464
	Need-SES	0.459	[0.021, 0.897]	0.0437	0.2620
	FedProto	0.965	[0.762, 1.168]	1.92×10^{-4}	0.0033
	Need-CF-CL	0.576	[0.132, 1.020]	0.0227	0.1589
PACS	Neutral Random	1.105	[0.814, 1.397]	4.62×10^{-4}	0.0069
	FedCS	0.954	[0.610, 1.297]	0.0015	0.0214
	SES	0.898	[0.712, 1.083]	1.79×10^{-4}	0.0032
	Need-SES	1.008	[0.489, 1.528]	0.0057	0.0574
	FedProto	2.315	[1.819, 2.811]	2.04×10^{-4}	0.0033
	Need-CF-CL	1.119	[0.447, 1.792]	0.0099	0.0890
Office-Home	Neutral Random	-0.042	[-0.324, 0.240]	0.6999	1.0000
	FedCS	-0.019	[-0.291, 0.252]	0.8534	1.0000
	SES	-0.013	[-0.307, 0.280]	0.9046	1.0000
	Need-SES	0.116	[-0.128, 0.361]	0.2566	1.0000
	FedProto	0.164	[-0.016, 0.344]	0.0644	0.3219
	Need-CF-CL	0.190	[0.073, 0.308]	0.0109	0.0890
Final A					
CIFAR-100	Neutral Random	1.413	[0.841, 1.985]	0.0024	0.0234
	FedCS	1.000	[0.511, 1.489]	0.0048	0.0333
	SES	1.358	[0.844, 1.872]	0.0018	0.0203
	Need-SES	1.396	[0.798, 1.994]	0.0029	0.0234
	FedProto	1.785	[1.577, 1.993]	1.86×10^{-5}	0.0003
	Need-CF-CL	1.386	[0.896, 1.876]	0.0014	0.0171
PACS	Neutral Random	1.514	[1.267, 1.761]	7.00×10^{-5}	0.0011
	FedCS	1.159	[0.792, 1.527]	9.38×10^{-4}	0.0131
	SES	1.358	[0.810, 1.906]	0.0023	0.0234
	Need-SES	1.294	[0.923, 1.665]	6.38×10^{-4}	0.0096
	FedProto	1.995	[1.703, 2.288]	4.57×10^{-5}	0.0008
	Need-CF-CL	1.647	[1.079, 2.215]	0.0013	0.0168
Office-Home	Neutral Random	0.039	[-0.345, 0.423]	0.7911	1.0000
	FedCS	0.064	[-0.281, 0.409]	0.6322	1.0000
	SES	0.016	[-0.385, 0.417]	0.9176	1.0000
	Need-SES	0.175	[-0.201, 0.552]	0.2650	1.0000
	FedProto	0.087	[-0.281, 0.454]	0.5488	1.0000
	Need-CF-CL	0.070	[-0.503, 0.643]	0.7508	1.0000

Using the same evaluation times and distributions as the main table, Table 15 gives complete improvements of the mechanism variants and full JAM over Neutral Random. Loss columns subtract each method's loss from Neutral Random's; A columns subtract Neutral Random's accuracy from each method's accuracy, in pp. Positive values indicate improvement. Neutral Random is the zero reference and is omitted.

The complete endpoint improvements agree with the component findings in Table 2: the matching variant improves more than the allocation variant on CIFAR-100, with the reverse on PACS, and full JAM improves more than any individual variant on both datasets. On Office-Home, which has the lowest Summary SNR, all variants and full JAM have Novel Loss improvements within ± 0.001, consistent with the weakest within-class deviation signal.

Table 16 provides per-seed paired differences and sample standard deviations over the three common seeds for every row in Tables 2 and 3.

Pair-permutation replay (JAM-Permuted). The preceding controls replace JAM components, changing the transmitted samples or sending dose and preventing isolation of pair-specific contri-

Table 13: Old-environment improvements of JAM over the deployment model and Need-SES.

Period	Reference	ΔLoss	95% CI	ΔA (pp)	95% CI
CIFAR-100					
Whole-process	G_0	0.0722	[0.0683, 0.0762]	2.137	[2.098, 2.176]
Whole-process	Need-SES	0.0062	[0.0037, 0.0087]	0.090	[0.032, 0.147]
Final	G_0	0.0708	[0.0653, 0.0763]	2.135	[1.994, 2.277]
Final	Need-SES	0.0103	[0.0031, 0.0175]	0.199	[-0.005, 0.403]
PACS					
Whole-process	G_0	0.0058	[0.0035, 0.0081]	0.047	[-0.060, 0.154]
Whole-process	Need-SES	0.0028	[0.0022, 0.0034]	0.051	[-0.070, 0.172]
Final	G_0	0.0266	[0.0254, 0.0278]	0.941	[0.757, 1.126]
Final	Need-SES	-0.0012	[-0.0033, 0.0009]	0.006	[-0.154, 0.166]
Office-Home					
Whole-process	G_0	0.0554	[0.0542, 0.0566]	1.437	[1.340, 1.535]
Whole-process	Need-SES	-0.0002	[-0.0009, 0.0006]	-0.167	[-0.308, -0.025]
Final	G_0	0.0952	[0.0944, 0.0959]	2.579	[2.418, 2.740]
Final	Need-SES	0.0007	[-0.0006, 0.0020]	-0.287	[-0.426, -0.147]

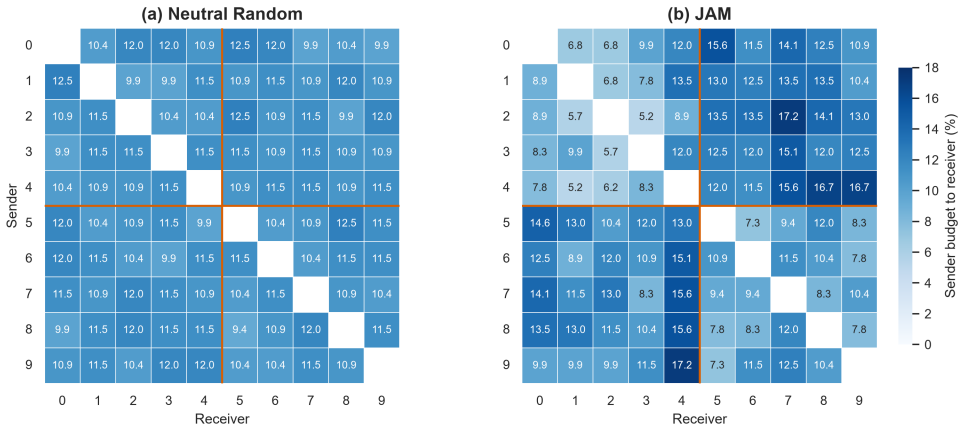

Figure 4: Receiver-allocation matrices before the Deficit environment naturally arrives. Each cell is the percentage of a sender's budget delivered to one receiver, averaged over rounds 0–3 and three seeds. Orange lines separate the two complementary client groups.

butions. JAM-Permuted instead replays transmission logs, keeping JAM's transmitted samples and sending dose unchanged while reassigning their receivers. We call JAM's selected receiver–sample assignments the original receiver–sample pairs, and the reassigned ones the permuted receiver–sample pairs. Permutation is strictly constrained: within each round and public class, every receiver's received count and every sender's sample multiset and total transmission count remain unchanged. Self-transmission, resending the same sample along a directed link across rounds, and sample-ownership violations are prohibited. Each seed produces two independent permutations, changing 43–52% of transmission events on CIFAR-100 and PACS, and 16–22% on Office-Home. Fewer Office-Home events change because, with 65 classes, most rounds send each class to only one receiver, leaving few events that can be reassigned. To verify that replay itself does not alter training, we retrain by replaying the original logs exactly; all evaluations match the formal JAM run bit for bit.

Table 17 reports full-process retraining differences, computed as JAM-Permuted minus JAM. Novel Loss and Final Loss worsen in 6/6 permutations on all three datasets; Old Loss worsens in 6/6 on CIFAR-100 and PACS. On CIFAR-100, mean Pre Loss increases by $+0.0180$ and Pre Accuracy drops by 0.34 pp. PACS differences are small, and Office-Home differences are near zero. Seen Loss improves in 6/6 permutations on CIFAR-100 and PACS: the original pairs trade a small Seen cost for Deficit gains. The statistical unit is the seed, and all three seeds agree in direction; 6/6

Table 14: Complete main-table results for all thirteen methods.

	Deficit environment						Whole-process Loss			
Method	Pre L	Pre A	First L	First A	Final L	Final A	Novel	Deploy	Seen	Old
CIFAR-100										
Frozen G_0	2.8118	19.18	2.8118	19.18	2.8118	19.18	2.8118	2.5100	2.8118	2.3134
Local-U0	2.6978	20.75	2.6907	20.89	2.6123	22.52	2.5590	2.3601	2.4509	2.2521
TENT-style-U0	2.7068	20.72	2.7048	20.88	2.6973	21.17	2.7022	2.3985	2.6994	2.2014
EWC-style-U0	2.7097	20.75	2.7083	20.86	2.7021	21.15	2.7048	2.4022	2.7010	2.2059
Neutral Random	2.6829	21.00	2.6714	21.07	2.5838	22.83	2.5492	2.3541	2.4517	2.2485
Entropy	2.6938	20.81	2.6825	20.92	2.5907	22.68	2.5557	2.3578	2.4570	2.2502
FedCS	2.6902	20.97	2.6800	21.17	2.5904	23.25	2.5556	2.3581	2.4582	2.2510
SES	2.6875	20.89	2.6733	21.01	2.5805	22.89	2.5475	2.3533	2.4495	2.2487
FedProto	2.6832	20.73	2.6716	20.84	2.5830	22.46	2.5434	2.3492	2.4398	2.2446
Need-SES	2.6771	21.23	2.6663	21.12	2.5731	22.85	2.5459	2.3522	2.4500	2.2474
Need-CF-CL	2.6774	21.12	2.6607	21.33	2.5847	22.86	2.5488	2.3532	2.4535	2.2463
CF-CL	2.6855	20.76	2.6744	21.00	2.5821	22.92	2.5483	2.3523	2.4502	2.2463
JAM	2.6423	21.69	2.6142	22.16	2.5179	24.25	2.5303	2.3434	2.4639	2.2412
PACS										
Frozen G_0	0.9230	71.94	0.9230	71.94	0.9230	71.94	0.9230	0.8175	0.9230	0.5483
Local-U0	0.9521	68.66	0.9257	69.68	0.8049	73.62	0.8031	0.7601	0.7142	0.5508
TENT-style-U0	0.9747	67.91	0.9676	68.02	0.9194	69.81	0.8790	0.8028	0.8104	0.5910
EWC-style-U0	0.9628	68.45	0.9554	68.69	0.9072	70.40	0.8687	0.7958	0.8015	0.5875
Neutral Random	0.9254	69.85	0.9005	70.71	0.7923	74.23	0.7957	0.7566	0.7179	0.5462
Entropy	0.9175	70.14	0.8932	70.78	0.7765	74.71	0.7890	0.7530	0.7138	0.5443
FedCS	0.9250	70.00	0.8998	70.86	0.7846	74.58	0.7933	0.7542	0.7159	0.5442
SES	0.9233	70.06	0.8993	70.78	0.7880	74.38	0.7942	0.7549	0.7171	0.5448
FedProto	0.9537	68.64	0.9266	69.66	0.8002	73.74	0.7996	0.7593	0.7081	0.5515
Need-SES	0.9250	69.95	0.8992	70.72	0.7877	74.44	0.7940	0.7550	0.7167	0.5453
Need-CF-CL	0.9247	69.84	0.9002	70.50	0.7935	74.09	0.7969	0.7574	0.7195	0.5464
CF-CL	0.9295	69.85	0.9042	70.66	0.7919	74.35	0.7975	0.7572	0.7195	0.5458
JAM	0.8864	70.95	0.8589	71.57	0.7445	75.74	0.7761	0.7485	0.7183	0.5425
Office-Home										
Frozen G_0	3.5376	18.35	3.5376	18.35	3.5376	18.35	3.5382	3.4862	3.5376	3.3893
Local-U0	3.4606	20.69	3.4456	21.24	3.3676	24.03	3.4143	3.3958	3.3981	3.3345
TENT-style-U0	3.5251	18.55	3.5242	18.55	3.5193	18.47	3.5224	3.4807	3.5206	3.3948
EWC-style-U0	3.5188	18.68	3.5177	18.77	3.5149	18.79	3.5155	3.4753	3.5128	3.3912
Neutral Random	3.4580	20.87	3.4427	21.41	3.3654	24.06	3.4134	3.3950	3.3985	3.3334
Entropy	3.4581	20.85	3.4428	21.41	3.3657	24.03	3.4136	3.3950	3.3986	3.3333
FedCS	3.4582	20.85	3.4429	21.43	3.3660	24.03	3.4137	3.3951	3.3987	3.3334
SES	3.4582	20.84	3.4430	21.40	3.3659	24.08	3.4138	3.3952	3.3988	3.3335
FedProto	3.4601	20.67	3.4450	21.23	3.3663	24.01	3.4135	3.3953	3.3972	3.3348
Need-SES	3.4582	20.71	3.4429	21.30	3.3650	23.92	3.4134	3.3951	3.3984	3.3337
Need-CF-CL	3.4590	20.64	3.4442	21.19	3.3663	24.03	3.4148	3.3961	3.4001	3.3345
CF-CL	3.4929	19.59	3.4782	19.92	3.3946	22.79	3.4455	3.4184	3.4307	3.3439
JAM	3.4573	20.83	3.4421	21.30	3.3612	24.10	3.4132	3.3952	3.3996	3.3339

describes direction only, without a significance test. This control measures pair contributions with transmitted content fixed and is not equivalent to rerunning JAM in closed loop.

To test whether pair-specific value is realized within one training round, we conduct single-round comparisons at round-start snapshots along JAM trajectories. Snapshots cover nine rounds for three seeds, retaining only receivers whose incoming samples can change under permutation, for 267 snapshots. Each snapshot receives four independent permutations, giving 1068 comparisons; 3 permutations leave that receiver's incoming samples unchanged, yielding 1065 actual comparisons. Each comparison substitutes the permuted incoming samples, performs the same 4 updates, and records ΔL and ΔG. Here, ΔL is the post-update loss difference on the receiver's Deficit test slice, permuted minus original pairs. ΔG is the increase in the JAM objective (equation 7) caused by the

Table 15: Allocation and matching effects relative to Neutral Random. The JAM-Permuted rows average six runs (three seeds × two permutation logs); all other rows are three-seed means.

	Deficit environment						Whole-process Loss			
Method	Pre L	Pre A	First L	First A	Final L	Final A	Novel	Deploy	Seen	Old
CIFAR-100										
JAM-Matching	+0.0118	+0.222	+0.0227	+0.500	+0.0469	+1.075	+0.0110	+0.0108	-0.0049	+0.0110
JAM-Allocation	+0.0079	+0.192	+0.0145	+0.435	+0.0149	+0.270	+0.0035	+0.0008	-0.0043	-0.0001
JAM-Permuted	+0.0230	+0.42	+0.0346	+0.67	+0.0379	+0.71	+0.0132	+0.0067	-0.0019	+0.0034
Need-JAM	+0.0232	+0.515	+0.0314	+0.580	+0.0500	+1.277	+0.0103	+0.0068	-0.0130	+0.0062
Static-JAM	+0.0296	+0.575	+0.0438	+0.980	+0.0579	+1.325	+0.0123	+0.0044	-0.0145	+0.0011
JAM	+0.0409	+0.757	+0.0590	+1.275	+0.0682	+1.502	+0.0203	+0.0118	-0.0097	+0.0079
PACS										
JAM-Matching	+0.0116	+0.498	+0.0137	+0.453	+0.0247	+0.924	+0.0063	+0.0033	-0.0028	+0.0022
JAM-Allocation	+0.0225	+0.813	+0.0241	+0.301	+0.0224	+0.569	+0.0106	+0.0048	+0.0010	+0.0035
JAM-Permuted	+0.0394	+1.19	+0.0418	+0.70	+0.0434	+1.30	+0.0197	+0.0078	+0.0015	+0.0033
Need-JAM	+0.0254	+0.831	+0.0235	+0.161	+0.0397	+1.170	+0.0145	+0.0060	+0.0007	+0.0031
Static-JAM	+0.0047	-1.444	+0.0072	-1.938	+0.0378	+0.566	+0.0085	+0.0033	-0.0013	+0.0021
JAM	+0.0403	+1.134	+0.0442	+0.749	+0.0501	+1.460	+0.0205	+0.0082	-0.0003	+0.0039
Office-Home										
JAM-Matching	+0.0001	-0.014	+0.0001	+0.033	+0.0001	+0.091	-0.0000	+0.0000	-0.0001	+0.0000
JAM-Allocation	+0.0005	-0.136	+0.0006	-0.102	+0.0032	+0.189	-0.0000	-0.0005	-0.0013	-0.0009
JAM-Permuted	+0.0005	-0.13	+0.0006	-0.09	+0.0037	+0.19	+0.0002	-0.0003	-0.0010	-0.0006
Need-JAM	+0.0008	-0.222	+0.0002	-0.265	+0.0036	+0.140	-0.0003	-0.0009	-0.0020	-0.0011
Static-JAM	+0.0001	-0.112	-0.0005	-0.213	+0.0028	-0.140	-0.0005	-0.0007	-0.0015	-0.0008
JAM	+0.0005	-0.117	+0.0007	-0.144	+0.0044	+0.260	+0.0003	-0.0003	-0.0010	-0.0006

same substitution. Per-seed mean ΔL values are $+0.0004$, -0.0011, and $+0.0017$: mixed in direction and near zero. Per-seed mean ΔG values are $+0.037$, $+0.046$, and $+0.080$, all positive, so the JAM objective consistently favors the original pairs. Yet the two differences share a sign in only 526 comparisons, close to chance. Task loss does not track objective changes, and pair-specific value is not realized within one round. Together with full-process retraining, where permutation consistently increases loss, these results place the value of pairs in the full sharing-and-training process.

D.3 Robustness Diagnostics

Table 3 compares only Neutral Random and JAM; here we provide complete multi-method results and per-seed details. With MobileNetV2, JAM retains its Novel Loss advantage over Neutral Random, SES, Need-SES, and FedProto on PACS, consistently across all three seeds. On CIFAR-100, its advantage over SES narrows to 0.0007; the advantage over Need-SES is $+0.0068$, positive in 2/3 seeds. Two additional seeds were run for the architecture comparisons and gave consistent mean conclusions; the main text and appendix report the three common seeds for consistency with the other robustness dimensions. The budget experiments use $U \in \{8, 16, 32\}$ on CIFAR-100 with all other settings fixed. JAM has lower Novel Loss than Neutral Random and Need-SES at all three budgets, with $U{=}16$ the main setting. Old-strong G_0 uses the same Old training data and batch structure as the main G_0, but extends pretraining to at most 50 rounds. Mean cross-entropy is evaluated on a held-out Old validation set after each round, and the round with the lowest validation loss is selected. Relative to the original G_0, Old Loss decreases by 0.51 and accuracy increases by 20.8 pp. The Staggered 2:8 stream is defined in Appendix B.1. For Old-strong, the absolute Novel Loss values are Local-U0 3.0502, Neutral Random 3.0262, Need-CF-CL 3.0136, and JAM 2.9757; corresponding Deploy Loss values are 2.4530, 2.4379, 2.4253, and 2.4058. For the staggered stream, Novel Loss values in the same order are 2.5454, 2.5408, 2.5350, and 2.5220, and $\text{Pre}_\tau\ L$ values are 2.6799, 2.6662, 2.6560, and 2.6325. After adaptation, each method further reduces Old Loss by about 0.15 and increases Old Accuracy by about 14 pp relative to its initial model; Novel Accuracy increases by about 7 pp while Novel Loss rises by about 0.45. The stronger old-environment model makes more confident but more frequently incorrect predictions in new environments, so loss and accuracy can move in opposite directions. Under the staggered schedule, the four methods' mean

Table 16: Per-seed paired differences relative to Neutral Random behind Tables 2 and 3. Panel A lists whole-process Novel Loss improvements of the mechanism variants. Panel B lists the robustness dimensions. All rows use the three common seeds; for JAM-Permuted, each per-seed value averages its two permutation logs. Δ is the baseline loss minus the JAM loss, so positive values indicate improvement. The last column reports the mean and the sample standard deviation.

		Seed			
Dimension	Setting	40	42	44	Mean $\pm$ SD
Panel A: mechanism variants (Table 2, ΔNovel)					
CIFAR-100	JAM-Matching	$+0.0116$	$+0.0100$	$+0.0115$	$+0.0110 \pm 0.0009$
CIFAR-100	JAM-Allocation	-0.0026	$+0.0055$	$+0.0077$	$+0.0035 \pm 0.0054$
CIFAR-100	Need-JAM	$+0.0054$	$+0.0078$	$+0.0175$	$+0.0103 \pm 0.0064$
CIFAR-100	Static-JAM	$+0.0070$	$+0.0116$	$+0.0182$	$+0.0123 \pm 0.0056$
CIFAR-100	JAM	$+0.0157$	$+0.0187$	$+0.0265$	$+0.0203 \pm 0.0056$
CIFAR-100	JAM-Permuted	$+0.0095$	$+0.0122$	$+0.0178$	$+0.0132 \pm 0.0042$
PACS	JAM-Matching	$+0.0083$	$+0.0073$	$+0.0034$	$+0.0063 \pm 0.0026$
PACS	JAM-Allocation	$+0.0086$	$+0.0121$	$+0.0111$	$+0.0106 \pm 0.0018$
PACS	Need-JAM	$+0.0127$	$+0.0160$	$+0.0149$	$+0.0145 \pm 0.0017$
PACS	Static-JAM	$+0.0063$	$+0.0132$	$+0.0060$	$+0.0085 \pm 0.0041$
PACS	JAM	$+0.0200$	$+0.0209$	$+0.0205$	$+0.0205 \pm 0.0004$
PACS	JAM-Permuted	$+0.0190$	$+0.0202$	$+0.0198$	$+0.0197 \pm 0.0007$
Office-Home	JAM-Matching	$+0.0004$	-0.0001	-0.0003	-0.0000 ± 0.0004
Office-Home	JAM-Allocation	-0.0009	$+0.0010$	-0.0003	-0.0000 ± 0.0010
Office-Home	Need-JAM	-0.0004	$+0.0008$	-0.0014	-0.0003 ± 0.0011
Office-Home	Static-JAM	-0.0016	$+0.0010$	-0.0008	-0.0005 ± 0.0013
Office-Home	JAM	-0.0005	$+0.0016$	-0.0001	$+0.0003 \pm 0.0011$
Office-Home	JAM-Permuted	-0.0007	$+0.0013$	-0.0001	$+0.0002 \pm 0.0011$
Panel B: robustness dimensions (Table 3, ΔL)					
Architecture	PACS, MNV2	$+0.0275$	$+0.0264$	$+0.0352$	$+0.0297 \pm 0.0048$
Architecture	CIFAR-100, MNV2	$+0.0092$	$+0.0077$	$+0.0176$	$+0.0115 \pm 0.0054$
Sender budget	CIFAR-100, U=8	$+0.0169$	$+0.0159$	$+0.0126$	$+0.0151 \pm 0.0023$
Sender budget	CIFAR-100, U=32	$+0.0227$	$+0.0249$	$+0.0299$	$+0.0258 \pm 0.0037$
Initialization	Old-strong G_0, Nov L	$+0.0441$	$+0.0620$	$+0.0453$	$+0.0505 \pm 0.0100$
Initialization	Old-strong G_0, Dep L	$+0.0279$	$+0.0411$	$+0.0273$	$+0.0321 \pm 0.0078$
Data stream	Staggered 2:8, Nov L	$+0.0132$	$+0.0216$	$+0.0216$	$+0.0188 \pm 0.0049$
Data stream	Staggered 2:8, Pre$_\tau$ L	$+0.0288$	$+0.0366$	$+0.0357$	$+0.0337 \pm 0.0043$

Table 17: Differences between permutation retraining and JAM, averaged over six paired runs (three seeds $\times$ two permutation logs). Positive loss differences mean permutation is worse. Pre metrics are Deficit loss/accuracy at round 3; Novel/Deploy/Seen/Old are round-1–9 average losses; Final metrics are at round 9.

Dataset	Pre L	Pre A (pp)	Nov L	Dep L	Seen L	Old L	Final L	Final A (pp)
CIFAR-100	$+0.0180$	-0.34	$+0.0071$	$+0.0051$	-0.0078	$+0.0045$	$+0.0303$	-0.79
PACS	$+0.0009$	$+0.06$	$+0.0008$	$+0.0004$	-0.0018	$+0.0006$	$+0.0067$	-0.16
Office-Home	$+0.0000$	-0.01	$+0.0002$	$+0.0001$	$+0.0000$	-0.0000	$+0.0007$	-0.07

losses range from 2.6076–2.6722 at first arrival and from 2.5858–2.6519 one round later, with JAM lowest at both times.