RGM-SL: RECEIVER-AWARE GRADIENT MATCHING FOR SOCIALIZED GENERALIST CULTIVATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Socialized learning brings together domain experts to cultivate a cross-domain generalist and return its integrated knowledge to the expert community. Under pathological label partitions, this process must transfer complementary local knowledge as the generalist's learning needs evolve. We introduce RGM-SL, a receiver-aware gradient matching approach to socialized generalist cultivation. After an initial bootstrap stage, each expert uses the current generalist as the receiving model and optimizes a compact synthetic dataset to align real- and synthetic-data gradients under its training objective. This makes the transmitted data responsive to both local expert knowledge and the generalist's current state. The framework coordinates model-level receiver states, sample-level synthetic data, and label-level supervision. The server aggregates expert uploads and trains the generalist with hard labels, class-masked soft labels, and expert anchoring. The resulting shared representation supports lightweight feedback heads that extend the experts' cross-domain capabilities. Experiments on CIFAR-10, CIFAR-100, and Tiny-ImageNet show that the complete receiver-aware refinement procedure improves overall generalist accuracy over random real-sample selection at matched images-per-class and round budgets on all three datasets. Generalist feedback further improves community overall accuracy and enables every expert to recognize classes outside its original domain.

1 INTRODUCTION

Independently trained domain experts possess complementary knowledge but have limited coverage beyond their local classes. MASC formalizes socialized learning as an expert–generalist–expert cycle: experts jointly cultivate a cross-domain generalist and subsequently receive its integrated knowledge (Yao et al., 2024). Under pathological (PAT) label partitions, each expert observes a disjoint subset of classes. The challenge is to transfer complementary knowledge through compact messages as the generalist's learning needs change.

Distributed learning offers several communication mechanisms. FedAvg and FedProx exchange model parameters for federated optimization (McMahan et al., 2017; Li et al., 2020); distillation and generation methods transfer predictions or generated examples (Lin et al., 2020; Zhu et al., 2021; Zhang et al., 2022a); and federated condensation communicates synthetic datasets for server-side training (Song et al., 2023; Xiong et al., 2023; Zhang et al., 2024). These approaches motivate compact knowledge exchange, but payload size alone does not determine training utility. A useful message must also reflect the state of the model that will learn from it.

We introduce RGM-SL, Receiver-Aware Gradient Matching for Socialized Learning. After bootstrapping a generalist from condensed local data, each expert receives the current generalist and refines a small synthetic dataset. The refinement aligns synthetic- and real-data gradients under the generalist's actual mixed hard–soft training objective. Thus, the same local data can produce different messages as the receiver evolves. The receiver's knowledge needs enter through its training gradients, without a separate knowledge-gap estimator.

The protocol connects model-level receiver states, sample-level synthetic data, and label-level supervision. Experts upload synthetic images with hard labels and class-masked soft targets; the server aggregates these packets and updates the generalist. After collaboration, the generalist supplies shared

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.

features to lightweight expert heads, completing the socialized learning cycle. Communication-efficient generalist cultivation is the principal focus, with feedback evaluating how its integrated knowledge benefits the community.

We evaluate RGM-SL on CIFAR-10, CIFAR-100, and Tiny-ImageNet under PAT partitions, using validation-selected checkpoints. The experiments compare generalist performance under explicit communication or sample budgets and evaluate expert feedback. A matched-IPC ablation measures the benefit of synthetic-data refinement over random real-sample selection. Our contributions are:

- Receiver-aware gradient matching that adapts each expert's synthetic data to the current generalist's training objective.
- A multi-granularity protocol that couples compact expert uploads with generalist updates and subsequent expert feedback.
- An evaluation of generalist accuracy, budgeted communication, and community feedback, including a controlled synthetic-refinement ablation.

2 Related Work

Federated optimization and knowledge transfer. FedAvg averages local models, FedProx regularizes local optimization, and SCAFFOLD corrects client drift with control variates (McMahan et al., 2017; Li et al., 2020; Karimireddy et al., 2020). FedNova normalizes updates after unequal local training, while FedBN keeps batch-normalization statistics local under feature shift (Wang et al., 2020; Li et al., 2021). FedPAQ combines periodic averaging with quantization; compressed federated optimization has also been studied analytically (Reisizadeh et al., 2020; Haddadpour et al., 2021). Alternative transfer mechanisms include ensemble distillation in FedDF, data-free generation in FedGen and DENSE, and feature exchange in FedFed (Lin et al., 2020; Zhu et al., 2021; Zhang et al., 2022a; Yang et al., 2023). FedBE distills a Bayesian ensemble of client models, whereas FedProto exchanges class prototypes (Chen & Chao, 2021; Tan et al., 2022). Soft-target distillation also supports client collaboration in KT-pFL and global-knowledge retention in FedNTD (Hinton et al., 2015; Zhang et al., 2021; Lee et al., 2022).

Synthetic-data communication. FedD3 uses one-shot local distillation; FedDM builds surrogates through distribution matching; and FedUSD retains principal and residual feature components (Song et al., 2023; Xiong et al., 2023; Xie et al., 2026). FedSD2C transfers client distillates, FedAF couples condensation with soft-label transfer, and 3SFC compresses gradients into synthetic features (Zhang et al., 2024; Wang et al., 2024; Zhou et al., 2023). These methods build on condensation objectives such as gradient, trajectory, and distribution matching (Zhao et al., 2021; Cazenavette et al., 2022; Zhao & Bilen, 2023); SelMatch and RDED further study initialization and synthetic-data quality (Lee & Chung, 2024; Sun et al., 2024).

Receiver relevance and socialized learning. FedFTG uses local–global disagreement, while Take What I Need applies active distillation to global-model training (Zhang et al., 2022b; Li et al., 2025). Our distinction is the matching target: gradients of the current generalist under its mixed expert supervision, recomputed throughout cultivation. MASC supplies the broader bidirectional learning setting (Yao et al., 2024). Shared representations and local prediction mechanisms also appear in personalized FL (Collins et al., 2021; Marfoq et al., 2022); QuPeD combines personalization, quantization, and distillation (Ozkara et al., 2021). We retain expert feedback within this setting while concentrating on receiver-conditioned synthetic messages for generalist training.

3 Preliminaries

Experts and PAT partitions. We consider K domain experts $\{E_k\}_{k=1}^{K}$, with parameters ϕ_k and local datasets $\mathcal{D}_k = \{(x_i, y_i)\}_{i=1}^{n_k}$. Expert k observes labels in $\mathcal{C}_k \subseteq \mathcal{C}$, where $\mathcal{C}$ is the global label space. PAT partitions satisfy

$$\mathcal{C}_k \cap \mathcal{C}_{k'} = \varnothing \ (k \neq k'), \qquad \bigcup_{k=1}^{K} \mathcal{C}_k = \mathcal{C}. \tag{1}$$

Socialized learning cultivates a generalist G_θ from these experts and returns its integrated knowledge to them (Yao et al., 2024). We write $G_\theta^{(r)} := G_{\theta^{(r)}}$ for the receiver at round r. Because the class sets do not overlap, each expert provides direct labels only for its own domain. A generalist must nevertheless predict over the union $\mathcal{C}$, while an expert receiving feedback is evaluated both on $\mathcal{C}_k$ and on $\mathcal{C} \setminus \mathcal{C}_k$. We call these the expert and non-expert classes, respectively. The feedback stage is part of the expert–generalist cycle, but the generalist is trained before feedback heads are fitted.

Condensation and distillation. Gradient matching optimizes synthetic data so that their training gradients approximate those of real data (Zhao et al., 2021). We denote a real batch by R_k and a synthetic dataset by S_k, following Figure 1. For parameters ϑ, training loss ℓ, and a synthetic batch $\widetilde{R}_k \subseteq S_k$, the generic objective is

$$\min_{\widetilde{R}_k} d\Big(\nabla_\vartheta \ell(\vartheta; R_k), \nabla_\vartheta \ell(\vartheta; \widetilde{R}_k)\Big), \tag{2}$$

where d measures gradient discrepancy. The choice of model state, training loss, and discrepancy determines which aspects of the real data the synthetic batch preserves. In the multi-round stage below, both gradients are evaluated at the current generalist, with the same objective used for subsequent generalist training.

Knowledge distillation. Knowledge distillation supplies supervision through softened model outputs (Hinton et al., 2015). For logits $z(x)$ and temperature $T > 0$, the distribution over a class set $\mathcal{A}$ is

$$p_T(c \mid x; \mathcal{A}) = \frac{\exp(z_c(x)/T)}{\sum_{j \in \mathcal{A}} \exp(z_j(x)/T)}, \qquad c \in \mathcal{A}. \tag{3}$$

For a domain expert, taking $\mathcal{A} = \mathcal{C}_k$ prevents its unobserved classes from receiving teacher probability mass. This restriction concerns the teacher target; the generalist still predicts over the full label space $\mathcal{C}$.

Communication. The server and experts collaborate for R rounds. Each expert uploads m images per local class, where m is the images-per-class (IPC) budget. Communication includes both expert uploads and per-recipient model downloads. A one-time bootstrap precedes these rounds. We distinguish the cost of cultivating the generalist from the later broadcast used to train feedback heads; checkpoint files are not network messages and are excluded from the communication count.

4 METHOD

4.1 PROBLEM FORMULATION AND OVERVIEW

Our objective is to cultivate G_θ over $\mathcal{C}$ through compact expert messages. Expert k uploads

$$S_k^{(r)} = \Big\{(\widetilde{x}_{k,j}^{(r)}, y_{k,j}, q_{k,j}^{(r)}, k)\Big\}_{j=1}^{m|\mathcal{C}_k|}, \tag{4}$$

containing synthetic images, hard labels, class-masked soft targets, and source IDs. Figure 1 shows three stages: one-time bootstrap, repeated receiver-aware refinement and generalist training, and generalist-to-expert feedback. The central design is to condition each synthetic packet on the current generalist and the objective used to train it.

4.2 EXPERT KNOWLEDGE BOOTSTRAP

The generalist is initialized from a reference expert E_a. Each expert condenses local data into S_k^{boot} using teacher-gradient matching (Zhao et al., 2021), teacher cross-entropy, and feature-statistic alignment: Let $\ell_k^{\mathrm{loc}}(E_k; B)$ denote cross-entropy on a batch B using only logits in $\mathcal{C}_k$. For a real batch $R_k \subseteq \mathcal{D}_k$ and a synthetic batch $\widetilde{R}_k \subseteq S_k^{\mathrm{boot}}$, the first two terms are

$$\mathcal{L}_{\mathrm{TGM}}^k = d_{\cos}\Big(\nabla_{\phi_k} \ell_k^{\mathrm{loc}}(E_k; R_k), \nabla_{\phi_k} \ell_k^{\mathrm{loc}}(E_k; \widetilde{R}_k)\Big), \tag{5}$$

$$\mathcal{L}_{\mathrm{TCE}}^k = \ell_k^{\mathrm{loc}}(E_k; \widetilde{R}_k), \tag{6}$$

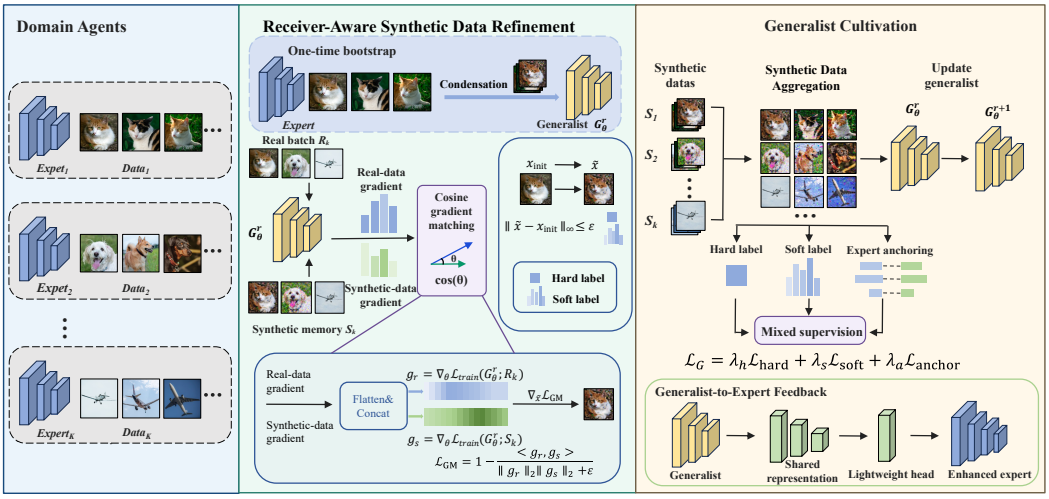

Figure 1: Overview of RGM-SL. Experts bootstrap a generalist and then refine synthetic data S_k by matching real- and synthetic-data gradients on the current receiver $G_\theta^{(r)}$. The server aggregates expert packets and updates the generalist using mixed supervision. Its shared representation subsequently supports lightweight expert feedback heads.

where $d_{\cos}$ is the cosine gradient discrepancy. The optimized variables are synthetic pixels, not expert parameters ϕ_k. Their weighted objective is

$$\mathcal{L}_{\text{boot}}^k = \lambda_{\text{TGM}}\mathcal{L}_{\text{TGM}}^k + \lambda_{\text{TCE}}\mathcal{L}_{\text{TCE}}^k + \lambda_{\text{feat}}\mathcal{L}_{\text{feat}}^k. \tag{7}$$

With $f_k(x)$ the normalized penultimate teacher feature, the feature term is

$$\mathcal{L}_{\text{feat}}^k = \|\mu(f_k(R_k)) - \mu(f_k(\widetilde{R}_k))\|_2^2 + \|\sigma(f_k(R_k)) - \sigma(f_k(\widetilde{R}_k))\|_2^2, \tag{8}$$

where μ and σ are batch mean and population standard deviation. These terms are accumulated over class-specific batches. The server pools the bootstrap packets, $S^{\text{boot}} = \bigcup_k S_k^{\text{boot}}$, and trains the initialized generalist with the mixed supervision below, obtaining $\theta^{(0)}$. Thus every expert contributes an initial class-balanced synthetic packet before receiver-conditioned refinement.

4.3 Receiver-Aware Synthetic Data Refinement

At round r, each expert receives $G_\theta^{(r)}$ and initializes a fresh fixed-IPC packet from local real images. Let $R_{k,c}$ and $S_{k,c}^{(r)}$ be real and synthetic batches of class c. Their receiver gradients are

$$g_{r,k,c} = \text{vec}\left[\nabla_{\theta^{(r)}}\mathcal{L}_{\text{train}}(G_\theta^{(r)}; R_{k,c}, E_k)\right], \tag{9}$$

$$g_{s,k,c} = \text{vec}\left[\nabla_{\theta^{(r)}}\mathcal{L}_{\text{train}}(G_\theta^{(r)}; S_{k,c}^{(r)}, E_k)\right]. \tag{10}$$

The subscripts r and s on g mean real and synthetic, as in the figure; the round dependence is implicit. The operator vec flattens and concatenates the matched parameter gradients in a fixed order. $\mathcal{L}_{\text{train}}$ is the mixed hard–soft loss defined in Section 4.4. Consequently, the target gradient describes the update that local data would induce under the receiver's training objective.

For each class, we compute a single cosine distance between the concatenated vectors and sum over the expert's classes:

$$\mathcal{L}_{\text{GM}}^k = \sum_{c\in\mathcal{C}_k}\left(1 - \frac{\langle g_{r,k,c}, g_{s,k,c}\rangle}{\|g_{r,k,c}\|_2\|g_{s,k,c}\|_2 + \epsilon_g}\right). \tag{11}$$

The figure suppresses the expert and class indices for readability. The real gradient is detached; the synthetic gradient retains its graph for differentiation with respect to pixels. Receiver parameters

remain fixed during this optimization. Matching therefore adapts the uploaded data, rather than updating the receiver locally. The classwise comparison uses the same receiver parameters and parameter ordering for both batches. The cosine stabilizer ϵ_g prevents division by zero when a gradient norm is small. This receiver state changes only after the server combines all expert packets and trains the generalist. The target in round r therefore reflects the update that the current generalist would obtain from expert k's real batch, rather than a gradient at a separate initialization.

The complete refinement objective is

$$\mathcal{L}^k_{\text{pixel}} = \lambda_{\text{GM}}\mathcal{L}^k_{\text{GM}} + \lambda_{\text{TGM}}\mathcal{L}^k_{\text{TGM}} + \lambda_{\text{TCE}}\mathcal{L}^k_{\text{TCE}} + \lambda_{\text{feat}}\mathcal{L}^k_{\text{feat}}. \tag{12}$$

In the main multi-round setting, $\lambda_{\text{TGM}} = 0$; teacher classification and feature alignment regularize the receiver-matching term. For image step t with learning rate η_x, each synthetic image is updated and projected around its real-image initialization:

$$\widetilde{x}^{(t+1)} = \Pi_{[0,1]\cap[x_{\text{init}}-\epsilon_x,\, x_{\text{init}}+\epsilon_x]}\left(\widetilde{x}^{(t)} - \eta_x \nabla_{\widetilde{x}^{(t)}} \mathcal{L}^k_{\text{pixel}}\right). \tag{13}$$

The radius ϵ_x is expressed in raw-pixel coordinates and scaled by channel standard deviations for normalized tensors. It is distinct from the cosine stabilizer ϵ_g. Fixed IPC constrains packet size, while receiver conditioning determines packet content. A fresh packet is refined at each round from its own initialization; the server does not optimize pixels on behalf of an expert.

4.4 Mixed Hard–Soft Supervision

For expert logits $z_k(x) = E_k(x)$, masking classes outside $\mathcal{C}_k$ gives the soft target used for distillation (Hinton et al., 2015):

$$q_{k,c}(x) = \begin{cases} \dfrac{\exp(z_{k,c}(x)/T_s)}{\sum_{j\in\mathcal{C}_k}\exp(z_{k,j}(x)/T_s)}, & c \in \mathcal{C}_k, \\ 0, & c \notin \mathcal{C}_k. \end{cases} \tag{14}$$

Let $p_\theta(x) = \text{softmax}(G_\theta(x))$. On a synthetic batch S, the two supervision terms are

$$\mathcal{L}_{\text{hard}} = -\frac{1}{|S|}\sum_{(x,y,q,k)\in S} \log p_{\theta,y}(x), \qquad \mathcal{L}_{\text{soft}} = -\frac{1}{|S|}\sum_{(x,y,q,k)\in S} q^\top \log p_\theta(x). \tag{15}$$

The figure's weights satisfy $\lambda_h + \lambda_s = 1$, with λ_h equal to the hard-label mixing weight in the implementation:

$$\mathcal{L}_{\text{train}} = \lambda_h\mathcal{L}_{\text{hard}} + \lambda_s\mathcal{L}_{\text{soft}}. \tag{16}$$

The teacher temperature is already incorporated into q; student soft cross-entropy uses neither an additional temperature nor T_s^2 scaling. This same objective supplies the real and synthetic gradients during refinement. Hard labels identify the correct global class, whereas the masked target expresses relative teacher probabilities only among classes the source expert has observed. The source ID identifies which expert provides the target and which domain can be anchored.

An anchor preserves the reference expert's class knowledge. On examples whose labels lie in $\mathcal{C}_a$, it distills restricted reference logits into the corresponding generalist logits:

$$\mathcal{L}_{\text{anchor}} = \tau_a^2\,\text{KL}\left(\text{softmax}\,\frac{z_a(x)|_{\mathcal{C}_a}}{\tau_a}\,\middle\|\,\text{softmax}\,\frac{G_\theta(x)|_{\mathcal{C}_a}}{\tau_a}\right), \tag{17}$$

averaged over the selected examples. The server trains on $S^{(r)} = \bigcup_k S_k^{(r)}$ using

$$\mathcal{L}_G = \lambda_h\mathcal{L}_{\text{hard}} + \lambda_s\mathcal{L}_{\text{soft}} + \lambda_a\mathcal{L}_{\text{anchor}}, \tag{18}$$

and obtains $\theta^{(r+1)}$ after the prescribed training epochs. The condensation-stage anchor weight is zero in the main configuration. Each expert's packet contributes examples to the same server training set; source-specific soft-label masks remain attached after aggregation.

4.5 MULTI-GRANULARITY INTERACTION

Each round exchanges a model state downstream and synthetic images with supervision upstream. For image, hard-label, and soft-target tensors $(X_k^{(r)}, Y_k^{(r)}, Q_k^{(r)})$ and generalist state $\mathcal{S}_\theta^{(r)}$,

$$B_{\mathrm{up}}^{(r)} = \sum_k \left[\mathrm{bytes}(X_k^{(r)}) + \mathrm{bytes}(Y_k^{(r)}) + \mathrm{bytes}(Q_k^{(r)}) \right], \tag{19}$$

$$B_{\mathrm{down}}^{(r)} = K\,\mathrm{bytes}(\mathcal{S}_\theta^{(r)}). \tag{20}$$

The generalist-formation budget adds bootstrap traffic to these round-wise costs:

$$B_{\mathrm{form}} = B_{\mathrm{boot}} + \sum_{r=0}^{R-1} \left(B_{\mathrm{up}}^{(r)} + B_{\mathrm{down}}^{(r)} \right). \tag{21}$$

Here B_{boot} counts the one-time bootstrap upload, while the final broadcast for expert feedback is excluded from B_{form}. This accounting separates message content from learning objectives; transport precision is an experimental configuration.

4.6 GENERALIST-TO-EXPERT FEEDBACK

The server sends the final generalist to the experts and freezes its feature extractor h_θ, completing the feedback direction of socialized learning (Yao et al., 2024). Expert k trains a lightweight head H_k on local features. With head logits $u_k(x) = H_k(h_\theta(x))$, its objective is

$$\begin{aligned} \mathcal{L}_{\mathrm{fb}}^k = \mathcal{L}_{\mathrm{CE}}^k + \lambda_{\mathrm{KD}} \tau_f^2\, \mathrm{KL}\left(\mathrm{softmax}\, \frac{z_k(x)|_{\mathcal{C}_k}}{\tau_f} \,\middle\|\, \mathrm{softmax}\, \frac{u_k(x)|_{\mathcal{C}_k}}{\tau_f} \right) \\ + \lambda_E \left(-\log \sum_{c \in \mathcal{C}_k} e^{u_{k,c}(x)} - a_E \right)^2. \end{aligned} \tag{22}$$

Here $\mathcal{L}_{\mathrm{CE}}^k$ is local-class cross-entropy, τ_f is the distillation temperature, and a_E is the energy target. The head is fitted on expert k's own data while the shared extractor remains fixed. Its local logits are compared only over $\mathcal{C}_k$ in the distillation term; cross-domain predictions remain available from the generalist at inference. A validation-calibrated router selects the generalist when $s_G(x) + b_k \geq s_k(x)$, where $s_G(x) = \log \sum_{c \in \mathcal{C}} e^{[G_\theta(x)]_c}$ and $s_k(x) = \log \sum_{c \in \mathcal{C}_k} e^{u_{k,c}(x)}$; otherwise it selects the local head. The resulting expert combines shared cross-domain prediction with a specialized head.

4.7 OVERALL TRAINING PROCEDURE

The procedure alternates fresh expert packets and generalist updates after one-time bootstrap, then trains the feedback heads with the generalist fixed. Algorithm 1 in the appendix gives the full sequence. Expert data remain local throughout the cycle.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Datasets and partitions. We evaluate ConvNet3 on CIFAR-10 and CIFAR-100 (Krizhevsky, 2009) and Tiny-ImageNet (Stanford CS231n, 2016) with disjoint PAT partitions 5×2, 4×25, and 5×40, respectively. A class-stratified 10% split of each original training set supplies validation data. CIFAR test sets and the official Tiny-ImageNet validation images are held out for test reporting. We select checkpoints by validation accuracy; all reported runs use seed 42.

Baselines and budgets. We compare with FedAvg (McMahan et al., 2017), FedProx (Li et al., 2020), FedUSD (Xie et al., 2026), FedSD2C (Zhang et al., 2024), FedAF (Wang et al., 2024), FedD3 (Song et al., 2023), FedDM (Xiong et al., 2023), and Random Real. Multi-round synthetic-data methods use IPC $= 10$ for 15 rounds. FedAvg and FedProx have no IPC; we select the best validation checkpoint within the training prefix whose total uplink plus downlink does not exceed the

RGM-SL formation budget. One-shot FedD3 and FedSD2C match the cumulative communicated-image budget, including bootstrap packets. We separately compare feedback with MASC (Yao et al., 2024) under the same head-training and routing protocol.

Metrics and implementation. Expert and Non-expert accuracy cover $\mathcal{C}_a$ and $\mathcal{C} \setminus \mathcal{C}_a$, respectively; Overall accuracy covers all test examples. Community results average the corresponding metrics over experts after feedback. ConvNet3 has three width-128 convolutional blocks with InstanceNorm, ReLU, and average pooling. Generalist training uses SGD with momentum 0.9, weight decay 5×10^{-4}, 30 epochs per phase, and batch size 64. The hard-label weight is $\lambda_h = 0.4$, with teacher-soft and anchor temperatures 3 and 2. Dataset-specific settings, condensation batches, and the communication accounting are given in Appendix A and Appendix C.

5.2 Generalist Cultivation

Table 1 reports validation-selected generalist accuracy. RGM-SL has the highest Overall and Expert accuracy in each dataset block. The Non-expert column complements these results by showing how each generalist performs outside the reference domain.

Table 1: Generalist accuracy (%) with ConvNet3 under PAT partitions. Multi-round data-centric methods use IPC $= 10$ for 15 rounds. FedAvg and FedProx use validation-selected checkpoints within the same total communication budget as RGM-SL. One-shot FedD3 and FedSD2C use matched cumulative sample budgets. Bold marks the largest value per column.

	CIFAR-10, PAT 5×2			CIFAR-100, PAT 4×25			Tiny-ImageNet, PAT 5×40		
Method	Expert	Non-expert	Overall	Expert	Non-expert	Overall	Expert	Non-expert	Overall
FedAvg	55.90	**45.96**	48.54	36.52	35.24	35.56	18.60	**20.88**	20.31
FedProx	46.30	35.39	37.57	31.68	29.08	29.73	18.95	16.05	16.78
FedAF	52.85	44.02	45.79	29.84	27.61	28.17	11.55	13.08	12.77
FedUSD	52.7	39.57	42.85	33.64	34.75	34.47	15.95	16.48	16.37
FedDM	58.75	45.03	48.46	29.16	27.67	28.04	11.45	12.62	12.39
FedSD2C	47.30	41.21	42.43	29.56	29.32	29.38	14.35	16.83	16.32
FedD3	39.9	28.33	30.64	35.36	**35.36**	35.36	13.95	11.39	11.90
RGM-SL	**76.3**	42.2	**49.02**	**48.76**	33.91	**37.62**	**33.95**	17.84	**21.06**

Information per upload. The IPC budget fixes how many images an expert can send, while receiver-aware refinement determines what those images convey. After each server update, the next packet is optimized against the updated generalist without increasing the number of image slots. The matched Random Real comparison in Table 4 tests the value of this complete refinement step at unchanged IPC and round count.

Results across domains. The main table compares different ways of using constrained communication. FedAvg and FedProx are read at validation-selected checkpoints within the generalist-formation communication budget, whereas one-shot condensation methods receive the cumulative sample allowance. Multi-round data-centric methods use IPC $= 10$ in each of 15 rounds. Under these stated controls, RGM-SL leads Overall and Expert accuracy in every dataset block. On CIFAR-10 it reaches 49.02% Overall, compared with 48.54% for FedAvg and 48.46% for FedDM. Its 76.3% Expert accuracy shows that training on packets from all five domains retains strong recognition of the reference domain; FedAvg records 55.90% in that column. On CIFAR-100, the 37.62% Overall result exceeds 35.36% for cumulative-sample-matched FedD3 and 35.56% for budget-matched FedAvg. The improvement is accompanied by 48.76% Expert accuracy, the highest in that block.

Tiny-ImageNet assigns 40 classes to each expert while preserving the same IPC per round. RGM-SL obtains 21.06% Overall and 33.95% Expert accuracy, against 20.31% and 18.60% for FedAvg. The corresponding Random Real control reaches 20.66% Overall at the same IPC and round count (Table 4). These comparisons show that repeated compact uploads remain useful as the label space expands; the Non-expert columns separately reveal how much of that performance extends beyond

the reference expert's classes. The next experiment asks a different question: whether the trained generalist can return its representation to each persistent domain expert.

Expert–non-expert balance. Under a disjoint PAT partition, the reference expert contributes labels for only part of the test space. We therefore report Expert and Non-expert accuracy alongside Overall rather than interpreting the latter alone. The three dataset blocks show that RGM-SL preserves particularly strong accuracy on the reference classes while learning a classifier over the union of expert domains. Its Non-expert scores also exceed those of the matched Random Real control on every dataset (Table 4). Thus the gain from refining synthetic packets is not confined to classes already recognized by the reference expert. Overall accuracy remains an example-weighted test metric, not an unweighted average of the Expert and Non-expert columns.

5.3 Generalist-to-Expert Feedback

The cultivated generalist's features support a local head for each expert. Community Overall accuracy increases from 19.12% to 43.82% on CIFAR-10, 14.11% to 34.15% on CIFAR-100, and 8.79% to 18.67% on Tiny-ImageNet. Every expert gains nonzero accuracy outside its local classes, as shown in Table 2. Table 3 compares community averages with MASC under the same feedback protocol. MASC attains higher community accuracy on all three datasets. These community scores evaluate routed expert models, whereas Table 1 evaluates one generalist.

Table 2: Per-expert accuracy before and after generalist feedback (%). "Before" and "After" denote each expert model before and after receiving knowledge from the cultivated generalist.

		Before feedback			After feedback		
Dataset	Agent	Expert	Non-expert	Overall	Expert	Non-expert	Overall
CIFAR-10	Agent 0	97.5	0	19.50	78.15	39.75	47.43
	Agent 1	90.75	0	18.15	69.30	29.75	37.66
	Agent 2	94.75	0	18.95	58.05	43.77	46.63
	Agent 3	97.95	0	19.59	81.45	34.34	43.76
	Agent 4	96.95	0	19.39	67.75	37.60	43.63
	Average	95.58	0	19.12	70.94	37.04	43.82
CIFAR-100	Agent 0	57.08	0	14.27	50.68	30.76	35.74
	Agent 1	53.88	0	13.47	45.88	27.15	31.83
	Agent 2	58.24	0	14.56	40.52	33.83	35.50
	Agent 3	56.60	0	14.15	42.72	30.45	33.52
	Average	56.45	0	14.11	44.95	30.55	34.15
Tiny-ImageNet	Agent 0	44.25	0	8.85	34.15	17.77	21.05
	Agent 1	45.80	0	9.16	32.60	13.61	17.41
	Agent 2	44.90	0	8.98	30.70	16.50	19.34
	Agent 3	39.45	0	7.89	28.80	13.90	16.88
	Agent 4	45.25	0	9.05	34.85	14.60	18.65
	Average	43.93	0	8.79	32.22	15.28	18.67

The gain is shared across experts, rather than being concentrated in a single source domain. After feedback, every CIFAR-10 expert reaches at least 29.75% Non-expert accuracy; every CIFAR-100 expert reaches at least 27.15%; and every Tiny-ImageNet expert reaches at least 13.61%. Their corresponding pre-feedback Non-expert accuracies are zero under the disjoint PAT partition. The community averages therefore capture a change in class coverage as well as an increase in Overall accuracy. Feedback is evaluated separately from generalist cultivation: it trains local heads on a frozen representation and does not alter the generalist scores in Table 1.

The per-agent results also show how feedback changes performance across the local domains. On CIFAR-10, Agent 0 reaches 47.43% Overall after feedback, while Agent 2 reaches 43.77% Non-expert accuracy, the largest such value among the five agents. On CIFAR-100, all four agents exceed 30% Overall; Agent 2 has the highest Non-expert accuracy at 33.83%. On Tiny-ImageNet, the largest Overall value is 21.05% for Agent 0, and every agent exceeds 13% Non-expert accuracy. These results show that the shared representation supports cross-domain predictions for each expert, even though their post-feedback accuracies differ by domain.

Table 3: Feedback-stage community accuracy averages (%).

Dataset	Method	Expert	Non-expert	Overall
CIFAR-10	RGM-SL	70.94	37.04	43.82
	MASC	89.65	85.56	86.38
CIFAR-100	RGM-SL	44.95	30.55	34.15
	MASC	58.93	50.30	52.46
Tiny-ImageNet	RGM-SL	32.22	15.28	18.67
	MASC	37.95	37.60	37.67

5.4 Ablation Studies

Table 4 compares complete refinement with Random Real at the same IPC and communication-round count.

Table 4: Ablation of receiver-aware synthetic-data refinement (accuracy, %). Random Real randomly selects local real samples without synthetic-image optimization; RGM-SL refines synthetic data by matching gradients under the current generalist. Both use IPC $= 10$ and 15 communication rounds.

Dataset	Method	Expert	Non-expert	Overall
CIFAR-10	Random Real	72.6	41.36	47.61
	RGM-SL	**76.3**	**42.2**	**49.02**
CIFAR-100	Random Real	**49.32**	30.81	35.44
	RGM-SL	48.76	**33.91**	**37.62**
Tiny-ImageNet	Random Real	33.05	17.56	20.66
	RGM-SL	**33.95**	**17.84**	**21.06**

Random Real sends selected local images without optimizing them for the receiver. RGM-SL uses the same IPC and round count, but refines each packet under the current generalist. This comparison tests the complete refinement pipeline, including teacher classification and feature regularization, rather than isolating individual loss terms.

Dataset-wise ablation. The matched-packet comparison reveals different accuracy patterns across the three PAT partitions. On CIFAR-10, refinement raises Expert accuracy from 72.6% to 76.3% and Non-expert accuracy from 41.36% to 42.2%. Both subsets therefore contribute to the increase in Overall accuracy. On CIFAR-100, the larger change is outside the reference domain: Non-expert accuracy rises by 3.10 percentage points, from 30.81% to 33.91%. Expert accuracy changes from 49.32% to 48.76%, while Overall accuracy increases from 35.44% to 37.62%. On Tiny-ImageNet, refinement raises Expert accuracy from 33.05% to 33.95% and Non-expert accuracy from 17.56% to 17.84%. These three comparisons use the same IPC and round count within each dataset; they measure the benefit of refining the selected images under the current generalist rather than allocating additional image slots.

6 Conclusion

RGM-SL conditions expert synthetic data on the current generalist's training gradients within a bidirectional socialized learning cycle. Each round aligns synthetic and real-data updates under the receiver's training objective before aggregating compact expert packets. Across three PAT benchmarks, this yields the highest Overall and Expert generalist accuracy among the compared methods under the stated budget protocols; the complete refinement pipeline also improves Overall accuracy over matched Random Real packets. The trained representation gives every expert nonzero accuracy outside its original class set through lightweight feedback heads, completing the expert–generalist cycle.

REFERENCES

George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A. Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10718–10727, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Cazenavette_Dataset_Distillation_by_Matching_Training_Trajectories_CVPR_2022_paper.html.

Hong-You Chen and Wei-Lun Chao. FedBE: Making bayesian model ensemble applicable to federated learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=dgtpE6gKjHn.

Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 2089–2099, 2021. URL https://proceedings.mlr.press/v139/collins21a.html.

Farzin Haddadpour, Mohammad Mahdi Kamani, Aryan Mokhtari, and Mehrdad Mahdavi. Federated learning with compression: Unified analysis and sharp guarantees. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, volume 130, pp. 2350–2358, 2021.

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

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J. Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 5132–5143, 2020.

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.

Gihun Lee, Minchan Jeong, Yongjin Shin, Sangmin Bae, and Se-Young Yun. Preservation of the global knowledge by not-true distillation in federated learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Yongmin Lee and Hye Won Chung. SelMatch: Effectively scaling up dataset distillation via selection-based initialization and partial updates by trajectory matching. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 26546–26567, 2024.

Hongcheng Li, Yucan Zhou, Yibin Wang, Xiaoyan Gu, Bo Li, and Weiping Wang. Take what i need: Active data distillation for federated learning. In *IEEE International Conference on Multimedia and Expo*, pp. 1–6, 2025. doi: 10.1109/ICME59968.2025.11209890.

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

Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. FedBN: Federated learning on non-IID features via local batch normalization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=6YEQUn0QICG.

Tao Lin, Lingjing Kong, Sebastian U. Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Othmane Marfoq, Giovanni Neglia, Richard Vidal, and Laetitia Kameni. Personalized federated learning through local memorization. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 15070–15092, 2022.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pp. 1273–1282, 2017.

Kaan Ozkara, Navjot Singh, Deepesh Data, and Suhas Diggavi. Quped: Quantized personalization via distillation with applications to federated learning. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, Ali Jadbabaie, and Ramtin Pedarsani. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, volume 108, 2020.

Rui Song, Dai Liu, Dave Zhenyu Chen, Andreas Festag, Carsten Trinitis, Martin Schulz, and Alois Knoll. Federated learning via decentralized dataset distillation in resource-constrained edge environments. In *International Joint Conference on Neural Networks*, pp. 1–10, 2023. doi: 10.1109/IJCNN54540.2023.10191879.

Stanford CS231n. Tiny imagenet challenge. CS231n course project, 2016. URL `https://cs231n.stanford.edu/2016/project.html`.

Peng Sun, Bei Shi, Daiwei Yu, and Tao Lin. On the diversity and realism of distilled dataset: An efficient dataset distillation paradigm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9390–9399, 2024.

Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. FedProto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8432–8440, 2022. doi: 10.1609/aaai.v36i8.20819.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H. Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL `https://papers.neurips.cc/paper/2020/hash/564127c03caab942e503ee6f810f54fd-Abstract.html`.

Yuan Wang, Huazhu Fu, Renuga Kanagavelu, Qingsong Wei, Yong Liu, and Rick Siow Mong Goh. An aggregation-free federated learning for tackling data heterogeneity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26233–26242, 2024.

Weiying Xie, Chenhe Hao, Haozhi Shi, Jitao Ma, Daixun Li, Jiazhe Li, Hengyi Wang, Leyuan Fang, and Yunsong Li. Fedusd: Unbiased synthetic data for federated learning. In *Forty-Third International Conference on Machine Learning*, 2026.

Yuanhao Xiong, Ruochen Wang, Minhao Cheng, Felix Yu, and Cho-Jui Hsieh. Feddm: Iterative distribution matching for communication-efficient federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16323–16332, 2023.

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.

Xinjie Yao, Yu Wang, Pengfei Zhu, Wanyu Lin, Jialu Li, Weihao Li, and Qinghua Hu. Socialized learning: Making each other better through multi-agent collaboration. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 56927–56945, 2024.

Jie Zhang, Song Guo, Xiaosong Ma, Haozhao Wang, Wenchao Xu, and Feijie Wu. Parameterized knowledge transfer for personalized federated learning. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Jie Zhang, Chen Chen, Bo Li, Lingjuan Lyu, Shuang Wu, Shouhong Ding, Chunhua Shen, and Chao Wu. DENSE: Data-free one-shot federated learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022a.

Junyuan Zhang, Songhua Liu, and Xinchao Wang. One-shot federated learning via synthetic distiller-distillate communication. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Lin Zhang, Li Shen, Liang Ding, Dacheng Tao, and Ling-Yu Duan. Fine-tuning global model via data-free knowledge distillation for non-iid federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10174–10183, 2022b.

Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 6514–6523, 2023.

Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In *International Conference on Learning Representations*, 2021.

Yuhao Zhou, Mingjia Shi, Yuanxi Li, Yanan Sun, Qing Ye, and Jiancheng Lv. Communication-efficient federated learning with single-step synthetic features compressor for faster convergence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5031–5040, 2023.

Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 12878–12889, 2021.

APPENDIX

A EXPERIMENTAL PROTOCOL AND REPRODUCIBILITY

Data partitions and model selection. We use disjoint, class-based PAT partitions: 5×2 for CIFAR-10, 4×25 for CIFAR-100, and 5×40 for Tiny-ImageNet. Expert k receives training examples only from $\mathcal{C}_k$. A class-stratified 10% split of the original training images supplies validation data; the original CIFAR test sets and the official Tiny-ImageNet validation images are held out for test reporting. Checkpoints and hyperparameters are selected using validation accuracy, not test accuracy. The reported main experiments use ConvNet3 and seed 42; they are single-seed results.

Generalist and condensation configuration. Table 5 collects the dataset-specific settings stated in Section 5. The shared generalist optimizer is SGD with momentum 0.9 and weight decay 5×10^{-4}. Each generalist-training phase lasts 30 epochs, with the learning rate multiplied by 0.1 halfway through and at three quarters of the phase. Its batch size is 64. The hard-label mixing weight is $\lambda_h = 0.4$, the teacher soft-label temperature is $T_s = 3$, and the anchor temperature is $\tau_a = 2$. In each round, the candidate pool contains 50 local examples per class and uses random selection. The synthetic-image learning rate is 1.0; the teacher cross-entropy and feature-statistic weights are 0.1 and 0.01, respectively. The condensation-stage anchor weight is zero. These values describe the main configuration, not a search space.

Table 5: Dataset-specific configuration of the main generalist experiments. The real/synthetic batch sizes refer to the per-class condensation step.

Setting	CIFAR-10	CIFAR-100	Tiny-ImageNet
PAT partition	5×2	4×25	5×40
Generalist learning rate	0.001	0.002	0.005
Expert-anchor weight	0.05	0.0125	0.025
Condensation iterations	50	50	100
Real/synthetic batch size	50/10	50/10	40/10

What is matched in a round. For each expert class c, real and synthetic batches are evaluated on the same received generalist. Their parameter gradients are flattened and concatenated in a fixed order before *one* cosine distance is computed for that class; the distances are then summed over $c \in \mathcal{C}_k$. Thus the matching term in Section 4.3 uses a global concatenated cosine distance per class. The

Algorithm 1 Receiver-Aware Generalist Cultivation and Expert Feedback

Require: Experts $\{E_k, \mathcal{D}_k, \mathcal{C}_k\}_{k=1}^K$, reference expert a, IPC m, rounds R
Ensure: Generalist $G_\theta^{(R)}$ and feedback heads $\{H_k, b_k\}_{k=1}^K$

```
Initialize the generalist from reference expert E_a
for k = 1 to K do
    Condense D_k into a fixed-IPC bootstrap packet S_k^boot using L_boot^k
    Attach hard labels, class-masked expert soft targets, and expert IDs
end for
Aggregate the bootstrap packets and train the generalist to obtain θ^(0)
for r = 0 to R − 1 do
    Broadcast the current generalist state G_θ^(r) to the K experts
    for k = 1 to K in parallel do
        Initialize m synthetic images per class from local real images
        Optimize the pixels using L_pixel^k on receiver G_θ^(r)
        Project the pixels into the valid range and their initialization neighborhood
        Attach hard labels, class-masked soft targets, and expert IDs
        Encode and upload the packet S_k^(r)
    end for
    Decode and aggregate S^(r) ← ⋃_{k=1}^K S_k^(r)
    Train the generalist with L_G to obtain θ^(r+1)
end for
Broadcast the final generalist and freeze its feature extractor h_{θ^(R)}
for k = 1 to K in parallel do
    Train H_k on local features using L_fb^k
    Calibrate router bias b_k on validation data
end for
return G_θ^(R) and {H_k, b_k}_{k=1}^K
```

real-data gradient is detached. The synthetic gradient retains the graph needed to differentiate the matching loss with respect to synthetic pixels, while the received generalist parameters remain fixed during condensation. Both real and synthetic receiver gradients use the mixed hard–soft training objective. In the main multi-round configuration the teacher-gradient-matching weight is zero; the teacher classification and feature-statistic terms still regularize the synthetic images.

Pixel constraint. Each synthetic image starts from a local real image x^{init} and is projected after every image update onto $[0, 1] \cap [x^{\text{init}} - \epsilon_x, x^{\text{init}} + \epsilon_x]$ in raw-pixel coordinates, with $\epsilon_x = 0.05$ in the main configuration. For normalized training tensors, the corresponding bound is scaled by the channel-wise normalization standard deviation. This projection limits displacement from the selected initialization; it is not a privacy mechanism.

B Full Training Procedure

Algorithm 1 expands the procedure described in Section 4.7. We denote the one-time bootstrap packet by S_k^{boot} and the fresh packet in round r by $S_k^{(r)}$.

C Evaluation and Budget Matching

Expert and non-expert accuracy. The generalist's Expert accuracy is computed on the reference expert's classes $\mathcal{C}_a$; Non-expert accuracy is computed on their complement $\mathcal{C} \setminus \mathcal{C}_a$. Overall accuracy uses all held-out test examples. Because these subsets have different sizes, Overall is computed over examples, not as an unweighted mean of the first two columns. For the feedback experiment, the corresponding Expert and Non-expert subsets are defined separately for each expert k using $\mathcal{C}_k$ and $\mathcal{C} \setminus \mathcal{C}_k$.

Comparison budgets. The multi-round synthetic-data methods in the main comparison use IPC $= 10$ and 15 rounds. For FedAvg and FedProx, which have no IPC parameter, we use the longest recorded training prefix whose cumulative uplink plus downlink does not exceed the generalist-formation communication budget of RGM-SL; the best checkpoint inside that prefix is selected by validation accuracy. FedD3 and FedSD2C use one-shot condensation, so their uploaded packet is compared against the cumulative communicated-sample budget, including bootstrap packets. The one-shot IPC is therefore a budget for the complete upload, not an IPC-per-round setting. These comparisons control different resource axes and should not be interpreted as identical training protocols.

Communication accounting. Let $X_k^{(r)}$, $Y_k^{(r)}$, and $Q_k^{(r)}$ denote the encoded images, hard labels, and soft targets uploaded by expert k in round r. The reported uplink includes the serialized payloads for these objects, and the downlink counts a generalist state sent to each expert. Bootstrap traffic is counted once. The total for generalist formation is

$$B_{\text{form}} = B_{\text{boot}} + \sum_{r=0}^{R-1} \left[\sum_{k=1}^{K} \big(\text{bytes}(X_k^{(r)}) + \text{bytes}(Y_k^{(r)}) + \text{bytes}(Q_k^{(r)}) \big) + K \, \text{bytes}(\mathcal{S}_\theta^{(r)}) \right]. \tag{23}$$

The final generalist broadcast for expert feedback is accounted for separately and is excluded from B_{form}. A stored checkpoint's file size is not substituted for a transmitted payload's size. Precision settings, when evaluated, belong to the communication experiments rather than the definition of the training objective.

D Generalist-to-Expert Feedback Protocol

The feedback stage freezes the trained generalist feature extractor and trains one lightweight head H_k for each expert using that expert's local data. At inference, the router compares the generalist score $s_G(x) = \log \sum_{c \in \mathcal{C}} \exp[G_\theta^{(R)}(x)]_c$ with the local-head score $s_k(x) = \log \sum_{c \in \mathcal{C}_k} \exp[H_k(h_{\theta^{(R)}}(x))]_c$. It predicts with the generalist when $s_G(x) + b_k \geq s_k(x)$ and otherwise uses the local head, whose outputs are restricted to $\mathcal{C}_k$. The bias b_k is calibrated on validation data. Feedback does not update the generalist. Each expert's post-feedback accuracy is evaluated with its own router and head; the community value is the average across experts, not the accuracy of a single generalist. The MASC feedback comparison uses the same feedback protocol.

E AI Assistance

Generative AI tools assisted with research brainstorming and experiment planning, code drafting and troubleshooting, literature discovery and reference formatting, figure drafting, and the drafting, editing, and polishing of parts of this manuscript. In particular, AI assistance was used to explore experimental options, locate candidate related work, and prepare preliminary text and visual layouts. The reported experimental measurements come from the described runs, not from AI-generated estimates. The authors reviewed the final methods, code, citations, figures, results, and claims and take responsibility for the content of the submission.