LEARNING FROM DISAGREEMENT: CONFLICT-COMPILED ROUTING NETWORKS FOR HETEROGENEOUS FEDERATED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated learning usually treats disagreement between client updates as optimization noise to average, shrink, or project away. We ask whether structured disagreement can instead supervise where model capacity should be allocated. We introduce FEDCRN, a conflict-compiled routing network that converts consensus-centered update directions into addresses for low-rank residual experts inside one shared model. Each expert is aggregated only within its conflict group. To extend these compiled addresses beyond client identity, a detached router learns per-example relative expert competence. This supports address-free unseen-client routing, while a confidence gate attenuates uncertain residuals and exactly recovers the consensus predictor under uniform routing. Our analysis shows how routing overlap changes cross-client interference, gives a separation condition for persistent compiler addresses, and derives a nonconvex bound governed by post-routing heterogeneity; it explains the mechanism rather than asserting universal superiority. Across five image federations, five paired seeds, and 12 methods, FEDCRN attains the best mean rank and leads accuracy on two benchmarks. On 50-client CIFAR-100, it improves over a parameter-matched raw-update ClusterMoE by 0.24 points; after transferring clean-selected hyperparameters to CIFAR-100-C without retuning, it retains the best mean but the 0.20-point gap is not significant. Multi-mode clients, address-free inference, paired tests, and compiler/routing ablations probe the claimed round-to-sample routing path. These results support a different view of heterogeneity: incompatible updates can supervise conditional capacity allocation.

1 INTRODUCTION

Federated learning (FL) trains a model from decentralized client data without collecting raw examples (McMahan et al., 2017). Its central algorithmic difficulty is not decentralization alone, but that each client optimizes a different empirical objective. Under label, covariate, and sample-size shifts, local gradients can point in opposing directions; averaging then cancels useful specialization, while longer local training amplifies drift. Dominant optimizer-oriented responses constrain, normalize, or project heterogeneous updates before aggregation (Li et al., 2020; Karimireddy et al., 2020; Wang et al., 2020; Acar et al., 2021; Reddi et al., 2021). Recent FedLAG is an important exception: it uses gradient conflict to identify layers that should remain personalized rather than globally aggregated (Nguyen et al., 2026). Thus conflict can already guide exclusion; our question is whether its residual geometry can instead construct reusable conditional capacity.

This premise is unnecessarily restrictive. When two reliable clients disagree because they observe different modes of the task, removing the conflicting component may discard exactly the capacity needed by minority clients. Clustered and personalized FL retain more local structure, but typically produce multiple client- or cluster-level models (Guo et al., 2025; Collins et al., 2021). Modular approaches store personalized plug-ins (Shen et al., 2025), align local heads or prototypes (Zhang et al., 2025; Meng et al., 2025), or share architecture-invariant adapters (Seo et al., 2026). Their modules are useful, yet the architecture is not constructed from the geometry of conflict itself. This leaves a basic question unanswered: *can client disagreement supervise a single conditional network, rather than merely select among separately personalized models?*

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.

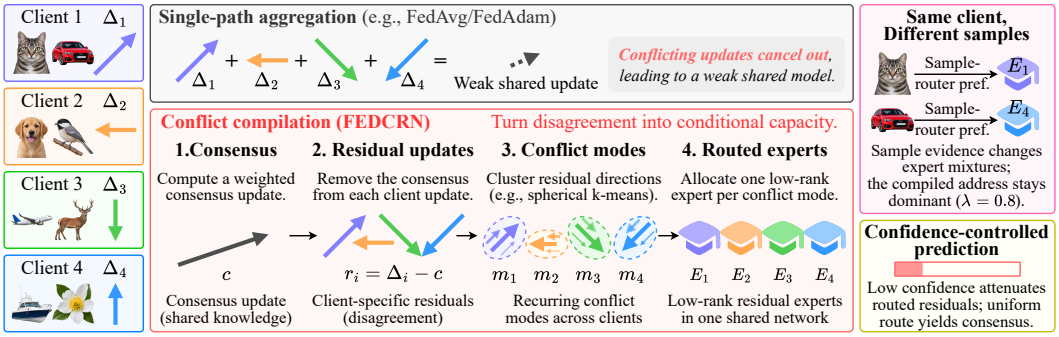


Figure 1: Problem view. Averaging can cancel reliable but opposing residual updates. FEDCRN compiles consensus-centered directions into internal experts. For a known client, sample evidence changes the soft expert mixture while its compiled address remains dominant at $\lambda = .8$; the gate attenuates uncertain residuals, and a uniform address-free route exactly recovers the consensus predictor.

We answer affirmatively with **Conflict-Compiled Routing Networks** (FEDCRN; Figure 1). Each round forms a weighted consensus update, subtracts it from compact client signatures, and spherical-clusters the remaining directions. Cluster identities address low-rank feature experts and zero-initialized decision residuals, each aggregated only within its conflict group. The slow address prior stabilizes expert specialization, while a router learns per-sample competence; hence two modes inside one client can induce different expert mixtures. A confidence gate attenuates uncertain residuals; exact recovery of the shared network occurs for a uniform route.

This design is intentionally orthogonal to optimizer improvements. The shared path uses adaptive server preconditioning, matching a strong FedAdam-style baseline (Reddi et al., 2021); the empirical question is therefore whether compiled residual capacity helps beyond a strong consensus optimizer. Our analysis formalizes the mechanism: cross-client interference in expert parameters is weighted by routing overlap, and the nonconvex convergence rate depends on a post-routing heterogeneity term. The theory predicts the largest benefit when raw conflict is high but admits a small number of recurring modes, and a smaller benefit on nearly homogeneous clients.

Our contributions are:

- To our knowledge, FEDCRN is distinct in using consensus-centered update-residual directions to construct persistent internal expert addresses, then coupling those round-level addresses to competency-trained sample routing inside one deployable network.
- We couple group-wise low-rank residual aggregation with competency-trained sample routing and confidence attenuation, so two modes at one client can induce different mixtures and unseen clients can route without an address.
- We show how routing overlap changes expert interference, give a separation condition controlling compiler error, and derive a convergence bound governed by post-routing heterogeneity; the analysis explains the mechanism rather than claiming universal superiority.
- Our main comparison evaluates 12 methods over five paired seeds and adds 50-client CIFAR-100/C, a parameter-matched ClusterMoE control, within-client multi-mode and unseen-client tests, paired confidence intervals, full compiler/routing ablations, and compute/storage accounting.

2 Related Work

Correcting heterogeneous optimization. FedAvg forms a sample-weighted mean of local models (McMahan et al., 2017). FedProx constrains local drift (Li et al., 2020); SCAFFOLD estimates client and server control variates (Karimireddy et al., 2020); FedNova normalizes unequal local progress (Wang et al., 2020); FedDyn introduces dynamic regularization (Acar et al., 2021); and adaptive federated optimization preconditions the server step (Reddi et al., 2021). MOON regularizes representations (Li et al., 2021), while FedLWS learns layer-wise shrinking factors (Shi et al., 2025). FedLAG explicitly treats conflicting gradients as a signal for layer-wise personalization and excludes

Table 1: Novelty boundary against closest clustered and federated-MoE families. "Conflict-built" means disagreement directly constructs parameter isolation, not merely a client similarity or routing score.

Method	Compiler / selection signal	Expert form	Expert aggregation	Routing	Conflict-built	One network
HCFL+	model similarity	full cluster model	cluster-wise	client	no	no
FedLAG	gradient conflict	private layers	exclusion	client	yes	no
FedJETs	pretrained features	full model	expert-wise	sample	no	no
pFedMoE	local/global features	feature extractors	global/local	sample	no	no
pFedMoAP	prompt distance	prompt	pool retrieval	client + sample	no	no
FLoRAL	learned client mixture	low-rank adapter	utilization-wise	client	no	yes
FC-MoE	expert importance/conflict	LoRA expert	projected average	client	no	yes
FedWeave	local prototypes/trajectory	LoRA expert	prototype-wise	sample	no	yes
ClusterMoE (ours)	raw update, Euclidean	internal residual	group-wise	client	no	yes
FEDCRN (ours)	**centered conflict direction**	**low-rank residual**	**group-wise**	**client + sample**	**yes**	**yes**

the identified layers from global aggregation (Nguyen et al., 2026). FEDCRN differs by centering update signatures, turning recurring residual directions into persistent internal expert addresses, and retaining the addressed residuals in one deployable network.

Personalization, clustering, and modularity. Body/head separation in FedRep and FedBABU distinguishes shared representation from local decisions (Collins et al., 2021; Oh et al., 2022); FedALA adaptively mixes global and local parameters (Zhang et al., 2023). Model-heterogeneous injection/distillation and separated communication/learning architectures likewise allocate distinct client-side components (Xie et al., 2024; Li & Huang, 2023). HCFL+ systematizes and improves clustered FL (Guo et al., 2025). Hot-Pluggable FL trains personalized modules stored at the server and retrieves them at inference (Shen et al., 2025); Co-LoRA shares dimension-invariant low-rank modules across heterogeneous multimodal clients (Seo et al., 2026). Unlike these methods, FEDCRN neither returns one model per cluster nor retrieves complete personalized modules. Conflict groups supervise parameter isolation inside a single network, and inference remains sample-conditioned.

Federated mixtures of experts. FedGMM models client distributions with a federated mixture (Wu et al., 2023); FedJETs routes examples to specialized full-model experts (Dun et al., 2023); pFedMoE mixes local and shared feature experts per sample (Yi et al., 2026); pFedMoAP gates prompt experts (Luo et al., 2025); and FLoRAL learns client-specific mixtures of low-rank adapters (Almansoori et al., 2025). Concurrent FC-MoE suppresses same-expert conflict using importance weighting and consensus projection (Lu et al., 2026); FedWeave aggregates LoRA experts at a prototype level and learns a global router from client trajectories (Duan et al., 2026). These works establish that expert routing and conflict awareness are not individually novel. Our narrower boundary is the complete link from *consensus-centered update-residual directions* to persistent internal addresses, group-wise residual aggregation, and competency-trained sample routing with uniform-route consensus fallback.

Representations and reliable predictions. FedGLCL uses a pretrained text encoder to anchor heterogeneous visual representations (Yan et al., 2025); CrossKT-FRA transfers federated represen-

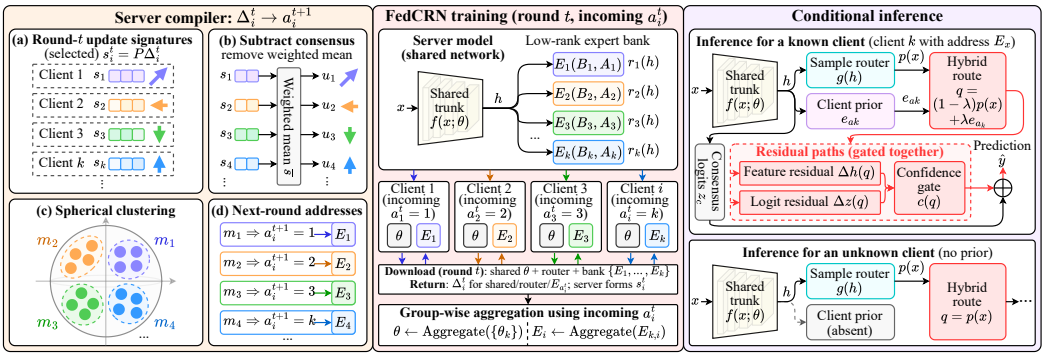


Figure 2: Complete FEDCRN pipeline and chronology. In round t, clients train with incoming a_i^t, and the server aggregates residuals using the same incoming groups. Clients return Δ_i^t (the server forms s_i^t), and the compiler emits a_i^{t+1} for activation only in the next round. Clients download the full bank for competence scoring but upload only the addressed residual; inference combines client and sample evidence with confidence-controlled residual attenuation.

tations across healthcare institutions under semi-supervised vertical FL (Yin et al., 2024); FedTED coordinates heterogeneous models with twin branches and data-free distillation (Zhang et al., 2025); pFedCSPC aligns cross-silo prototypes (Meng et al., 2025); and NUCFL adapts calibration objectives to client–global similarity (Chu et al., 2025). Such supervision can be powerful but may require text, synthesized data, prototypes, or explicit calibration losses. FEDCRN requires only model updates and learns a confidence-gated residual path; calibration is evaluated as an outcome rather than optimized directly.

3 CONFLICT-COMPILED ROUTING NETWORKS

3.1 PROBLEM SETTING AND ARCHITECTURE

There are N clients with weights $p_i \geq 0$, $\sum_i p_i = 1$, and local objectives $F_i(w) = \mathbb{E}_{(x,y)\sim\mathcal{D}_i}\ell(f_w(x), y)$. Standard FL minimizes $F(w) = \sum_i p_i F_i(w)$. We decompose parameters as $w = (\theta, \{\phi_k, \psi_k\}_{k=1}^K, r)$: θ denotes a consensus trunk and classifier, ϕ_k is a rank-d feature residual, ψ_k a decision residual, and r a router. For a trunk feature $h_\theta(x) \in \mathbb{R}^D$,

$$q(x,i) = (1-\lambda)\,\text{softmax}(r(\text{sg}[h_\theta(x)])) + \lambda e_{a_i}, \tag{1}$$

$$c(q) = \left[\frac{\max_k q_k - 1/K}{1 - 1/K}\right]_+, \qquad z = h_\theta(x) + c(q)\sum_k q_k U_k \sigma(V_k h_\theta(x)), \tag{2}$$

$$f(x,i) = C_\theta(z) + c(q)\sum_k q_k \psi_k(h_\theta(x)). \tag{3}$$

Here $\phi_k = (U_k, V_k)$ with $V_k \in \mathbb{R}^{d\times D}$ and $U_k \in \mathbb{R}^{D\times d}$, e_{a_i} is client i's compiled address, sg stops routing gradients from altering the trunk, and $\lambda = 0.8$. The gate is zero for uniform routing and one for a one-hot decision. Residual branches initially output zero: U_k and the logit residual are zero-initialized while V_k uses Kaiming initialization, preserving trainable gradients and starting exactly from the consensus model.

3.2 COMPILING DISAGREEMENT INTO EXPERT ADDRESSES

At the start of round t, client i receives persistent address a_i^t and updates only that residual expert. Let Δ_i^t be its resulting local update. Transmitting and clustering the full vector is unnecessary; we form a compact signature $s_i^t = P\Delta_i^t$ from the final projector and classifier, where P is a fixed coordinate selection. The server removes common motion,

$$\bar{s}^t = \sum_{i\in S_t} \tilde{p}_i s_i^t, \qquad u_i^t = \frac{s_i^t - \bar{s}^t}{\|s_i^t - \bar{s}^t\|_2 + \epsilon}, \tag{4}$$

Algorithm 1 One communication round of FEDCRN

Require: global parameters (θ, Φ, Ψ, r), incoming addresses a_i^t, selected clients S_t
1: **for** $i \in S_t$ in parallel **do**
2: download (θ, r, Φ, Ψ); keep non-addressed experts read-only
3: evaluate all expert predictions to form $\pi(x, y)$ by Equation 6
4: optimize Equation 7, updating $(\theta, r, \phi_{a_i^t}, \psi_{a_i^t})$
5: return updated shared path, router, and addressed residual; server forms $s_i = P\Delta_i$
6: **end for**
7: adaptively aggregate the shared path and router over S_t
8: aggregate residuals over incoming groups $\{i : a_i^t = k\}$ by Equation 5
9: compute u_i^t by Equation 4; compile next-round addresses a_i^{t+1}

and compiles the next-round address by spherical K-means: $a_i^{t+1} = \arg\max_k \langle u_i^t, \mu_k^t \rangle$. Warm-started centers preserve identities across rounds. Centering is the intended compiler signal: raw spherical signatures can be dominated by shared progress, whereas u_i^t isolates the direction discarded by a single mean. This distinction is specific to directional clustering; subtracting one common vector from every point leaves ordinary Euclidean K-means assignments and objective unchanged up to translation.

The consensus parameters are aggregated over all selected clients and updated with server moments. Because local training used the incoming address, expert k is aggregated over the same set $S_{t,k} = \{i \in S_t : a_i^t = k\}$; the newly compiled a_i^{t+1} is activated only in the next round:

$$\bar{\phi}_k^{t+1} = \frac{\sum_{i \in S_{t,k}} n_i \phi_{i,k}^{t+1}}{\sum_{i \in S_{t,k}} n_i}, \quad \bar{\psi}_k^{t+1} = \frac{\sum_{i \in S_{t,k}} n_i \psi_{i,k}^{t+1}}{\sum_{i \in S_{t,k}} n_i}. \tag{5}$$

An empty group retains its previous state. To evaluate all $f_k(x)$ in Equation 6, the measured reference protocol downloads the shared path, router, and complete expert bank at the start of every selected-client round. Non-addressed experts are read-only: the client uploads the shared path, router, and only its addressed expert. Hence uplink grows with one residual, whereas downlink and server storage grow with K; Appendix G reports both directions separately.

3.3 ROUTER LEARNING AND CONFIDENCE ATTENUATION

Addresses summarize round-level behavior but cannot identify multiple modes inside one client. We therefore train the sample branch from the relative competence of the current expert bank. Let $f_k(x)$ denote the prediction using expert k alone and define a detached soft target

$$\pi_k(x, y) = \frac{\exp[-\ell(f_k(x), y)/\tau]}{\sum_j \exp[-\ell(f_j(x), y)/\tau]}. \tag{6}$$

The router receives both the slow address target and the fast per-example competence target:

$$\mathcal{L}_i = \mathcal{L}_{\text{task}} + \frac{\beta}{2}\{\text{CE}(r(\text{sg}[h]), a_i^t) + \text{KL}(\pi \| p(x))\} + \gamma(K\|\bar{q}_i\|_2^2 - 1) + \xi\|G - I\|_F^2, \tag{7}$$

where $p(x) = \text{softmax}(r(\text{sg}[h]))$, $\bar{q}_i$ is batch-average usage, and G is the Gram matrix of normalized expert outputs. The address stabilizes early training; the competency target lets samples from one client induce different sample-dependent expert mixtures, although the address remains the largest component when $\lambda = .8$. Stop-gradient prevents either target from rewriting the trunk. At deployment, a known client uses Equation 1; an unseen client sets $\lambda = 0$. The gate attenuates uncertain residuals, and a uniform route—possible in the address-free setting—sets both residual paths to zero and recovers the consensus network.

3.4 WHY ROUTING CHANGES THE CONFLICT TERM

Write a client's stochastic gradient as $g_i = g_i^0 + \sum_k q_{ik} g_i^k$, where g_i^0 acts on shared parameters and g_i^k on disjoint expert block k. For two clients, orthogonality of parameter blocks gives

$$\langle g_i, g_j \rangle = \langle g_i^0, g_j^0 \rangle + \sum_k q_{ik} q_{jk} \langle g_i^k, g_j^k \rangle. \tag{8}$$

Table 2: Five-seed controlled-image performance (mean ± 95% confidence interval). All 12 algorithms are shown. Best is **bold**; second best is underlined.

Method	MNIST		RotMNIST		Digits	
	Acc. ↑	Worst-10% ↑	Acc. ↑	Worst-10% ↑	Acc. ↑	Worst-10% ↑
FedAvg	92.9 ± 3.7	85.5 ± 8.6	82.9 ± 14.8	63.8 ± 36.1	80.2 ± 2.7	52.7 ± 17.3
FedProx	93.6 ± 2.4	86.3 ± 6.6	86.4 ± 9.4	72.1 ± 23.1	80.2 ± 2.6	52.7 ± 17.3
SCAFFOLD	95.2 ± 0.8	**92.7** ± 0.4	90.9 ± 4.2	83.8 ± 8.5	86.3 ± 4.7	**76.8** ± 6.9
FedDyn	93.3 ± 2.5	85.4 ± 9.1	82.5 ± 15.1	55.8 ± 41.6	80.2 ± 2.8	53.4 ± 15.7
MOON	93.4 ± 2.7	86.0 ± 7.3	83.7 ± 15.3	64.4 ± 39.6	80.6 ± 2.8	54.6 ± 15.7
FedAdam	**95.5** ± 1.3	91.2 ± 4.0	90.4 ± 10.8	80.2 ± 19.6	87.3 ± 5.9	69.2 ± 24.9
FedNova	94.5 ± 2.1	88.6 ± 7.1	**93.4** ± 2.9	**85.1** ± 9.3	81.5 ± 4.4	58.0 ± 17.1
FedBABU	92.9 ± 3.5	82.8 ± 15.1	89.8 ± 5.1	77.1 ± 21.5	81.1 ± 3.3	48.5 ± 19.8
FedALA	93.4 ± 2.5	85.1 ± 8.0	89.6 ± 3.3	72.2 ± 15.8	79.8 ± 3.6	54.4 ± 18.5
FedLWS	88.5 ± 9.6	72.6 ± 32.3	80.5 ± 17.6	61.3 ± 35.1	74.2 ± 8.6	31.8 ± 28.9
ClusterMoE	95.0 ± 2.2	86.5 ± 8.8	87.8 ± 12.9	66.8 ± 37.0	**89.7** ± 5.4	75.1 ± 8.5
FedCRN	95.3 ± 2.1	91.0 ± 4.2	92.0 ± 11.4	78.5 ± 29.1	88.6 ± 7.9	67.6 ± 29.1

Thus expert interference is multiplied by routing overlap $\rho_{ij} = q_i^\top q_j$. Hard distinct routes make it zero, while uncertain routes smoothly recover sharing. Appendix B develops this observation into a negative-conflict bound and a nonconvex convergence result. Importantly, the result does not claim that clustering is always correct; clustering and router errors appear explicitly and motivate our ablations.

4 Experiments

4.1 Protocol

We evaluate five federations with five paired seeds. MNIST ($\alpha = .1$), RotMNIST ($\alpha = .05$; translation, blur, and noise), and low-data Digits use 12 clients and 50% participation. CIFAR-100 and CIFAR-100-C use frozen ResNet-18 features, 50 clients, $\alpha = .3$, and 20% participation; the latter evaluates six corruption episodes. In the five main benchmark comparisons, every method uses the same partitions, backbone, client samples, one local epoch, and 30 rounds. We report accuracy, macro-F1, expected calibration error (ECE), negative update-pair rate, cumulative communication, convergence round, and 10th-percentile client accuracy (Worst-10%). Unless stated otherwise, intervals are two-sided Student-t 95% CIs over independent seeds; paired comparisons use two-sided paired t-tests on per-seed differences. The multi-mode test gives each client an 80% dominant and 20% recurring minority transformation/concept mode, then separately evaluates known-client and address-free unseen-client inference. Full configurations are in Appendix C.

The eleven baselines cover averaging (FedAvg), drift correction (FedProx, SCAFFOLD, FedDyn), representation regularization (MOON), adaptive/normalized aggregation (FedAdam, FedNova, FedLWS), and personalization (FedBABU, FedALA). Most importantly, **ClusterMoE** is architecture-matched: it uses the same expert bank, group aggregation, optimizer, and parameter budget, but clusters raw updates with Euclidean distance and uses client-only routing. It is a controlled baseline, not a claimed published method. Existing federated MoE methods make incompatible assumptions for this controlled protocol: FedJETs starts from a pretrained expert pool, pFedMoE retains a private local extractor, and FLoRAL learns client-specific mixtures. We therefore compare their mechanisms in Table 1 rather than presenting a non-equivalent reimplementation as a direct baseline. FedLWS is a recent ICLR baseline (Shi et al., 2025). FedCRN uses $K = 4$, rank $d = 8$, $\lambda = .8$, $\tau = .5$, $\beta = .03$, $\gamma = .005$, and $\xi = .01$. Training ran in PyTorch 2.0 on one RTX 4080; we make no physical-network or hardware-systems claim.

Table 3: Natural-image performance with 50 clients and 20% participation (five seeds; mean ± 95% CI). CIFAR-100-C reuses clean-selected hyperparameters without corruption-specific tuning.

Method	CIFAR-100		CIFAR-100-C	
	Acc. ↑	Worst-10% ↑	Acc. ↑	Worst-10% ↑
FedAvg	16.68 ± 1.74	9.97 ± 1.00	10.18 ± 1.31	2.98 ± 0.55
FedProx	16.67 ± 1.72	9.97 ± 1.00	10.16 ± 1.31	2.98 ± 0.55
SCAFFOLD	21.27 ± 1.88	13.29 ± 2.03	13.51 ± 0.53	2.76 ± 1.13
FedDyn	16.69 ± 1.75	9.97 ± 1.00	10.19 ± 1.33	2.98 ± 0.55
MOON	17.04 ± 1.98	9.85 ± 1.17	10.71 ± 1.30	2.86 ± 0.66
FedAdam	41.49 ± 1.68	31.91 ± 3.54	25.69 ± 1.43	4.31 ± 2.22
FedNova	19.89 ± 2.78	12.04 ± 3.17	11.76 ± 1.16	2.59 ± 1.15
FedBABU	11.14 ± 1.25	4.75 ± 1.48	6.61 ± 1.34	1.37 ± 0.46
FedALA	8.58 ± 0.54	3.15 ± 0.17	5.62 ± 0.79	1.13 ± 0.43
FedLWS	15.69 ± 2.66	8.03 ± 1.34	9.97 ± 1.72	2.95 ± 1.04
ClusterMoE	42.97 ± 1.57	32.54 ± 2.59	26.38 ± 1.24	4.10 ± 2.00
FedCRN	**43.21** ± 1.10	**32.88** ± 1.79	**26.57** ± 1.05	**5.07** ± 2.54

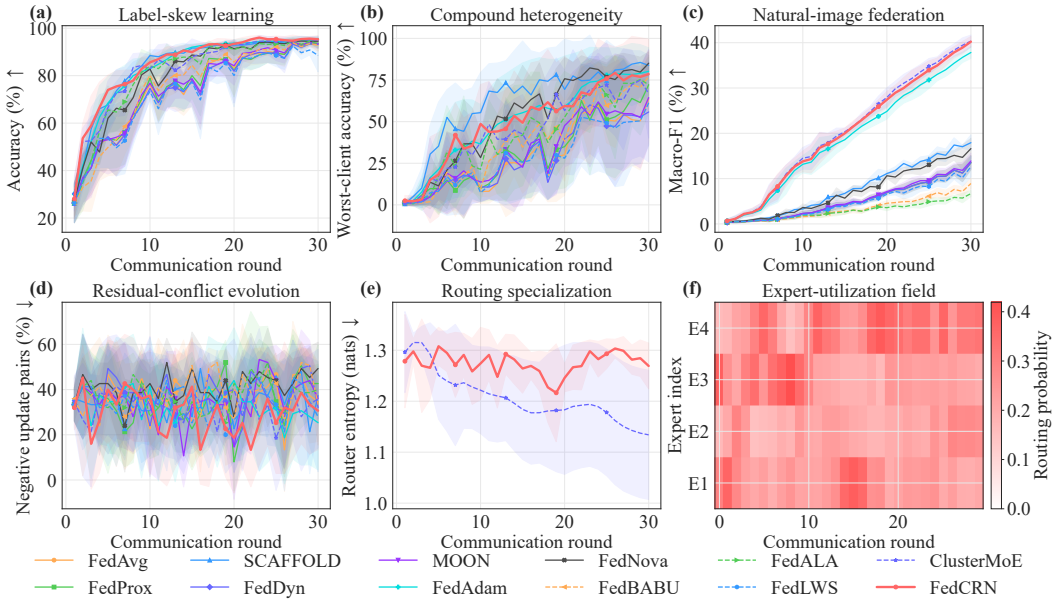


Figure 3: Learning and mechanism dynamics. The top row reports all 12 algorithms with five-seed standard-deviation bands. Bottom panels separately measure residual conflict for all algorithms, router entropy for the two routed architectures, and the FedCRN expert-utilization field.

4.2 Main comparison

Tables 2–3 compare every algorithm under paired partitions. FedCRN has the best mean rank (1.6) and leads mean accuracy on two of five benchmarks. On CIFAR-100/C, FedCRN reaches 43.21/26.57%, versus 42.97/26.38% for the parameter-matched ClusterMoE and 41.49/25.69% for FedAdam. Its gains over ClusterMoE are positive but not significant ($p = .485/.466$), while its gains over FedAdam are significant ($p = .014/.012$). FedAdam is 0.19 points better on MNIST, FedNova is 1.44 points better on RotMNIST, and ClusterMoE is 1.14 points better on Digits; these counterexamples rule out universal dominance. Full paired tests appear in Appendix D.

Figure 3 shows the full optimization trajectories rather than endpoint-only comparisons. The lower row avoids single-curve diagnostics: conflict is shown for all 12 methods, routing entropy contrasts

Table 4: Within-client multi-mode and unseen-client routing after 25 training rounds (five seeds; mean ± 95% CI). NMI is normalized mutual information between sample routes and latent modes.

Variant	Known acc. (%) ↑	Unseen acc. (%) ↑	NMI ↑	Purity ↑
Address only	54.95 ± 10.73	50.98 ± 8.37	0.08 ± 0.05	0.41 ± 0.06
Hybrid	**60.96** ± 11.95	51.90 ± 12.39	0.05 ± 0.03	0.40 ± 0.02
Hybrid, no gate	60.85 ± 12.48	**55.33** ± 9.22	0.08 ± 0.11	0.41 ± 0.04
Sample only	54.66 ± 8.80	54.66 ± 8.80	**0.12** ± 0.09	**0.45** ± 0.11

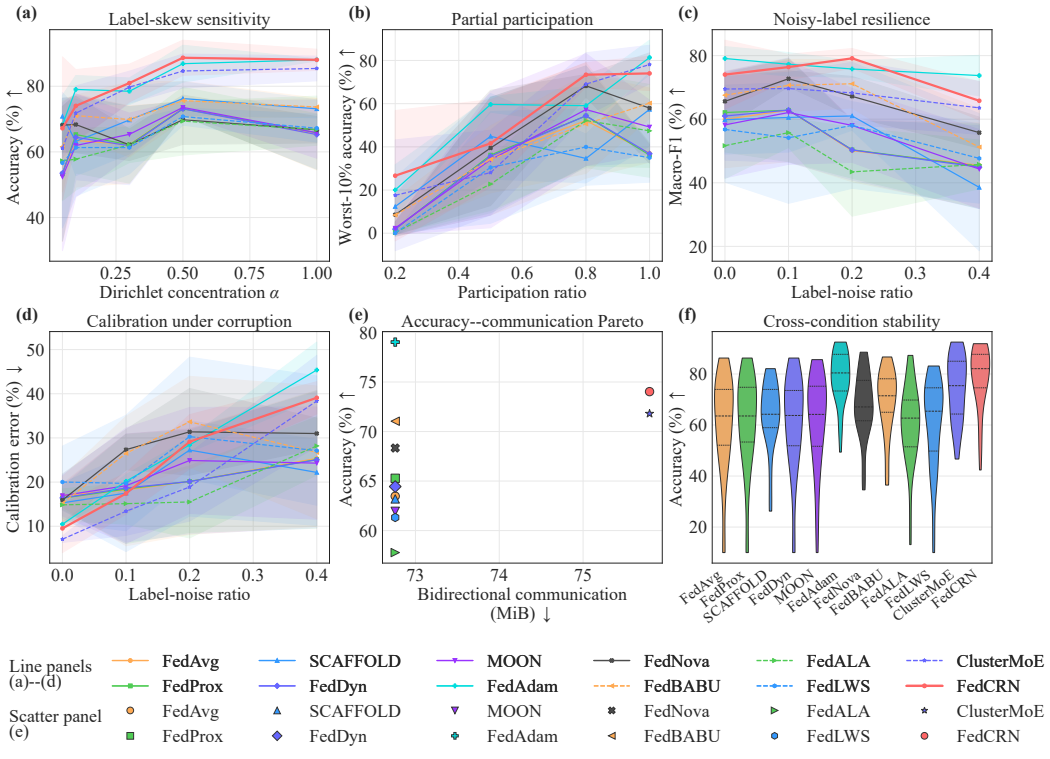


Figure 4: Stress tests for all 12 algorithms on Digits across label skew, participation, and symmetric label noise (three seeds). The lower row reports calibration, accuracy–communication trade-offs, and cross-condition stability.

FEDCRN with ClusterMoE, and only the expert-utilization field is intrinsically method-specific. Expert mass can reallocate while warm starts preserve address identities.

Table 4 makes sample adaptation identifiable: every client contains two modes, so one address cannot determine the latent mode. The hybrid gives the best known-client accuracy (60.96%), removing the confidence gate gives the best address-free accuracy (55.33%), and sample-only routing gives the strongest mode association (NMI 0.12; purity 0.45). The two routing signals and confidence attenuation therefore trade off known-client accuracy, unseen-client transfer, and mode recovery rather than producing one uniformly best variant. These are 25-round endpoints; the paired ablations in Table 6 use the same seeds and partition configuration but stop at 20 rounds, so their accuracy columns are not directly interchangeable.

4.3 STRESS TESTS, CALIBRATION, AND MECHANISM EVIDENCE

Figure 4 varies Dirichlet concentration, participation, and label corruption for every method. It tests whether conclusions persist beyond one operating point and exposes failure cases: structured

Table 5: Compiler, routing, and residual ablation on within-client multi-mode data after 20 training rounds (three seeds; mean ± 95% CI).

Variant	Acc. (%) ↑	Unseen acc. (%) ↑	NMI ↑	Purity ↑
Address only	49.98 ± 27.77	50.00 ± 19.75	0.04 ± 0.09	0.42 ± 0.11
Centered + Euclidean	55.70 ± 30.09	49.70 ± 23.16	0.05 ± 0.04	0.40 ± 0.09
Full FedCRN	55.50 ± 24.20	51.23 ± 23.94	0.08 ± 0.21	0.43 ± 0.17
No confidence gate	55.98 ± 30.76	49.44 ± 25.50	0.05 ± 0.05	0.40 ± 0.05
No residual head	50.20 ± 20.72	50.01 ± 21.44	0.05 ± 0.04	0.41 ± 0.09
No warm start	52.32 ± 21.13	49.48 ± 21.16	0.06 ± 0.06	0.44 ± 0.12
Raw + Euclidean	55.70 ± 30.09	49.70 ± 23.16	0.05 ± 0.04	0.40 ± 0.09
Raw + spherical	**57.95** ± 31.48	50.16 ± 25.71	0.09 ± 0.14	**0.46** ± 0.10
Sample only	49.86 ± 24.04	49.86 ± 24.04	**0.12** ± 0.21	0.45 ± 0.15
Single expert	52.73 ± 22.41	**52.73** ± 22.41	0.00 ± 0.00	0.36 ± 0.08

Table 6: Five-seed, 20-round paired audit of the four ablations closest to the core claims. The difference is computed per seed before its 95% CI and two-sided paired test; Table 4 reports 25-round endpoints.

Variant	Accuracy (%) ↑	Full − variant (pp)	Paired p
Full FedCRN	57.35 ± 11.04	–	–
Raw + spherical	**57.70 ± 15.48**	-0.34 ± 6.53	0.891
No warm start	50.44 ± 10.75	+6.92 ± 6.99	0.052
Address only	52.36 ± 12.22	+4.99 ± 7.99	0.157
Sample only	49.99 ± 8.90	+7.37 ± 5.66	0.023

distribution conflict can be routed, whereas random label corruption need not form persistent modes. We therefore do not interpret the compiler as a replacement for robust-label objectives.

Table 5 decomposes the compiler instead of only switching it off. Raw-versus-centered and Euclidean-versus-spherical rows test whether residual direction, rather than update magnitude, is responsible; reinitializing centers tests address persistence; address-only and sample-only rows test the hybrid; and gate/head/single-expert rows test conditional capacity. Centered+Euclidean and Raw+Euclidean coincide exactly because a common translation leaves Euclidean K-means geometry invariant; their equality is a mathematical control, not a duplicated record. Parameter sweeps for K, rank, signature dimension, and λ appear in Appendix E.

Table 6 preserves uncertainty. Centering ties its raw-spherical control in both five seeds (-0.34 ± 6.53, $p = .891$) and ten seeds (-1.51 ± 2.90, $p = .269$; Appendix F), so we make no end-to-end centering claim. Warm starts have a positive but marginal effect ($+6.92 \pm 6.99$, $p = .052$); removing the address branch causes a significant 7.37 ± 5.66-point loss ($p = .023$), whereas address-only is inconclusive ($+4.99 \pm 7.99$, $p = .157$). In the controlled 50-seed diagnostic at $\|c\| = 8$, centering improves purity by 0.102 ($p < .001$) and persistence by 0.037 ($p = .022$).

5 Discussion, Limitations, and Conclusion

FedCRN targets *structured recurring* disagreement, not non-IID data broadly. Persistent centered-signature clusters indicate reusable modes, whereas random-label corruption destabilizes addresses. Routed low-rank blocks separate opposing residuals and scale their interaction by routing overlap (Equation 8) without eliminating shared-path or within-group conflict. Multi-mode tests distinguish client addresses from transferable sample evidence; Figure 5 and Appendix A audit these claims.

Evidence is simulation-based, and CIFAR-100/C use frozen ResNet-18 features. We do not establish end-to-end representation learning, physical-network latency, secure aggregation, adversarial robustness, fairness, or signature privacy; soft inference also evaluates all K experts (Appendix G). Natural-image gains are modest: we claim a repeatable mechanism and favorable mean rank, not universal dominance.

AI USE STATEMENT

Generative AI tools were used to assist with generating and refining portions of the experimental code and with checking and revising intermediate steps in the mathematical proofs. All AI-assisted code was manually reviewed and tested, and all mathematical derivations and proofs were independently verified by the authors. Generative AI was also used for language polishing and improving the clarity of the manuscript; the research problem, proposed method, experimental design, and interpretation of results were developed by the authors. The authors take full responsibility for the final content of the paper, including its code, theoretical claims, experimental results, and text.

REFERENCES

Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N. Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *ICLR*, 2021. URL https://openreview.net/forum?id=B7v4QMR6Z9w.

Abdulla Jasem Almansoori, Samuel Horváth, and Martin Takáč. Collaborative and efficient personalization with mixtures of adaptors. In *Conference on Parsimony and Learning*, pp. 1328–1364, 2025. URL https://proceedings.mlr.press/v280/almansoori25a.html.

Yun-Wei Chu, Dong-Jun Han, Seyyedali Hosseinalipour, and Christopher Brinton. Unlocking the potential of model calibration in federated learning. In *ICLR*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/f66340d6f28dae6aab0176892c9065e7-Abstract-Conference.html.

Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *ICML*, 2021.

Donghang Duan, Xu Zheng, Lizong Zhang, Chong Mu, and Meng Han. FedWeave: Rethinking the unit of specialization in heterogeneous federated MoE-LoRA. *arXiv preprint arXiv:2607.26618*, 2026. URL https://arxiv.org/abs/2607.26618.

Chen Dun, Mirian Hipolito Garcia, Guoqing Zheng, Ahmed Hassan Awadallah, Robert Sim, Anastasios Kyrillidis, and Dimitrios Dimitriadis. FedJETs: Efficient just-in-time personalization with federated mixture of experts. *arXiv preprint arXiv:2306.08586*, 2023. URL https://arxiv.org/abs/2306.08586.

Yongxin Guo, Xiaoying Tang, and Tao Lin. Enhancing clustered federated learning: Integration of strategies and improved methodologies. In *ICLR*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/d3248f63ad76392608963b97c095ca33-Abstract-Conference.html.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In *ICML*, 2020.

Junyi Li and Heng Huang. Resolving the tug-of-war: A separation of communication and learning in federated learning. In *NeurIPS*, 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/0aa800df4298539770b57824afc77a89-Abstract-Conference.html.

Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *CVPR*, 2021.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *MLSys*, 2020.

Yijun Lu, Zihan Fang, Pengpeng Qiao, Zheng Lin, Jing Yang, Yuxin Zhang, Por Lip Yee, Zhe Chen, and Jun Luo. Conflict-aware federated fine-tuning of large language models with mixture-of-experts. *arXiv preprint arXiv:2606.15625*, 2026. URL https://arxiv.org/abs/2606.15625.

Jun Luo, Chen Chen, and Shandong Wu. Mixture of experts made personalized: Federated prompt learning for vision-language models. In *ICLR*, 2025. URL https://openreview.net/pdf?id=xiDJaTim3P.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, 2017.

Lei Meng, Zhuang Qi, Lei Wu, Xiaoyu Du, Zhaochuan Li, Lizhen Cui, and Xiangxu Meng. Improving global generalization and local personalization for federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 36(1):76–87, 2025. doi: 10.1109/TNNLS.2024.3417452.

Minh-Duong Nguyen, Hoang Khoi Do, Nam-Khanh Le, Nguyen Hoang Tran, Zhaohui Yang, Van-Duc Nguyen, and Trinh Van Chien. Toward layer-wise personalized federated learning: Adaptive layer disentanglement via conflicting gradients. *IEEE Transactions on Networking*, 34:3210–3225, 2026. doi: 10.1109/TON.2026.3653812.

Jaehoon Oh, Sangmook Kim, and Se-Young Yun. FedBABU: Toward enhanced representation for federated image classification. In *ICLR*, 2022.

Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konecny, Sanjiv Kumar, and H. Brendan McMahan. Adaptive federated optimization. In *ICLR*, 2021.

Minhyuk Seo, Taeheon Kim, Hankook Lee, Jonghyun Choi, and Tinne Tuytelaars. Co-LoRA: Collaborative model personalization on heterogeneous multi-modal clients. In *ICLR*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/1baeb4973da3e78e523024030ea81cf5-Abstract-Conference.html.

Lei Shen, Zhenheng Tang, Lijun Wu, Yonggang Zhang, Xiaowen Chu, Tao Qin, and Bo Han. Hot-pluggable federated learning: Bridging general and personalized FL via dynamic selection. In *ICLR*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/fed9263d7f1086e735904ff690527a53-Abstract-Conference.html.

Changlong Shi, Jinmeng Li, He Zhao, Dandan Guo, and Yi Chang. FedLWS: Federated learning with adaptive layer-wise weight shrinking. In *ICLR*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/401de6a666d7672757bdadfc53c3c123-Abstract-Conference.html.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H. Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. In *NeurIPS*, 2020.

Yue Wu, Shuaicheng Zhang, Wenchao Yu, Yanchi Liu, Quanquan Gu, Dawei Zhou, Haifeng Chen, and Wei Cheng. Personalized federated learning under mixture of distributions. In *ICML*, pp. 37860–37879, 2023. URL https://proceedings.mlr.press/v202/wu23z.html.

Luyuan Xie, Manqing Lin, Tianyu Luan, Cong Li, Yuejian Fang, Qingni Shen, and Zhonghai Wu. MH-pFLID: Model heterogeneous personalized federated learning via injection and distillation for medical data analysis. In *ICML*, pp. 54561–54575, 2024. URL https://proceedings.mlr.press/v235/xie24h.html.

Yunlu Yan, Chun-Mei Feng, Wangmeng Zuo, Salman Khan, Lei Zhu, and Yong Liu. On the importance of language-driven representation learning for heterogeneous federated learning. In *ICLR*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/a063ec362f0f0e9cfebd82236e7d6ee5-Abstract-Conference.html.

Liping Yi, Han Yu, Gang Wang, Xiaoguang Liu, and Qinghua Hu. pFedMoE: Data-level personalization with mixture of experts in model-heterogeneous personalized federated learning. *IEEE Transactions on Knowledge and Data Engineering*, 38(3):1905–1918, 2026. doi: 10.1109/TKDE.2026.3656194.

Zilong Yin, Haoyu Wang, Bin Chen, Xin Zhang, Xiaogang Lin, Hangling Sun, Anji Li, and Chenyu Zhou. Federated semi-supervised representation augmentation with cross-institutional knowledge transfer for healthcare collaboration. *Knowledge-Based Systems*, 300:112208, 2024. doi: 10.1016/j.knosys.2024.112208. URL https://www.sciencedirect.com/science/article/pii/S0950705124008426.

Jianqing Zhang, Yang Hua, Hao Wang, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. FedALA: Adaptive local aggregation for personalized federated learning. In *AAAI*, 2023.

Xiongtao Zhang, Ji Wang, Weidong Bao, Yaohong Zhang, Xiaomin Zhu, Hao Peng, and Xiang Zhao. Improving generalization and personalization in model-heterogeneous federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 36(1):88–101, 2025. doi: 10.1109/TNNLS.2024.3405190.

A Expanded Discussion and Claim Audit

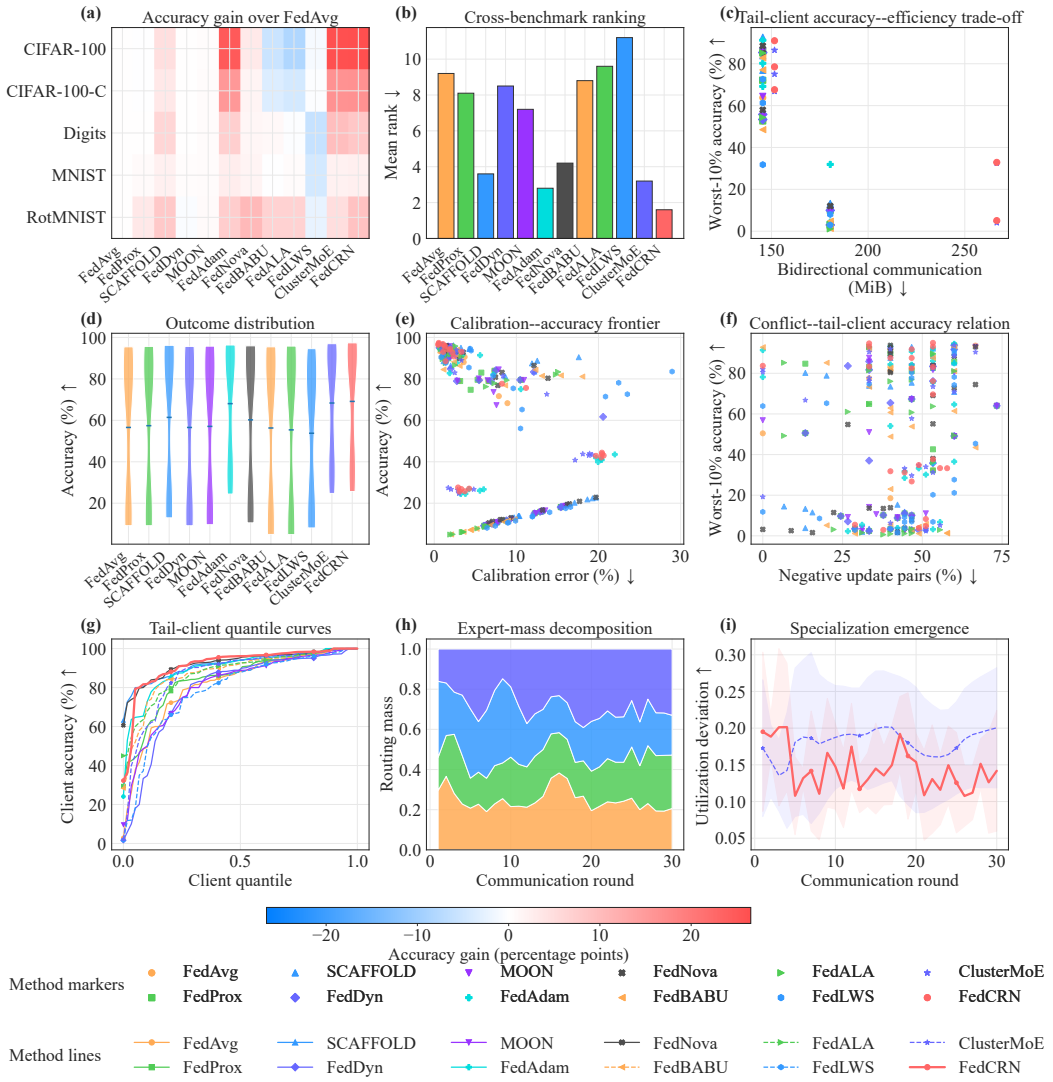


Figure 5: Cross-benchmark mechanism audit. Panels (a)–(b) place algorithms on the horizontal axis and use slanted names; (g) compares all 12 tail-client curves; (i) contrasts both routed models rather than showing an isolated line. Colors are fixed across every figure.

Reading the mechanism figure. Figure 5 separates nine information roles rather than repeating one endpoint metric. The gain field and average ranks summarize benchmark coverage; Pareto, calibration, and conflict panels expose trade-offs; all 12 client-quantile curves audit tail behavior; and two routed architectures are compared wherever a specialization trajectory is meaningful. The FedCRN-only stacked field is a decomposition of its four internal experts, not an algorithm comparison.

Claim-to-evidence audit and privacy. Additional parameters alone cannot explain the intended mechanism because ClusterMoE matches the expert bank and optimizer. Centered/raw, spherical/Euclidean, warm-start, address/sample, gate, head, and single-expert interventions isolate successive links in the design. Signatures are nevertheless not private: temporal group membership may reveal distribution similarity, and secure aggregation ordinarily hides the individual updates needed for clustering. Computing noisy similarities or group sums inside a secure protocol is possible future work but changes the threat model.

Closest federated-MoE protocol audit. FedJETs specializes a pool of full models from pretrained features; pFedMoE combines a shared feature expert with a client-local expert; and FLoRAL learns client-specific mixtures of low-rank adapters. FC-MoE and FedWeave instead target LoRA fine-tuning, using respectively projected expert updates and prototype-level aggregation. Inserting any of these into our shared CNN/feature-MLP protocol would change its private-state, pretrained-expert, model-size, or specialization-unit assumptions rather than provide a drop-in matched baseline. We therefore use ClusterMoE as the parameter-, optimizer-, and training-budget-matched causal control, and do not claim empirical superiority over those systems; Table 1 records the mechanism differences explicitly.

Efficiency. Each selected client uploads the shared path, router, and one rank-d residual; server storage grows with K, but upload does not. Soft inference in the reference implementation costs $O(KDd)$ for residuals, whereas top-1 execution would cost $O(Dd)$. Increasing K can also reduce updates per group and increase compiler error, so expert count is a statistical and computational choice rather than free capacity.

B Detailed Theory

B.1 Definitions and assumptions

Let $w = (\theta, \phi_1, \ldots, \phi_K)$ absorb feature and logit residuals into ϕ_k. Fix exogenous route vectors $q = (q_1, \ldots, q_N)$, with $q_i \in \Delta^{K-1}$, throughout the analyzed optimization horizon, and define the fixed routed objective $F_q(w) = \sum_i p_i F_i(w; q_i)$. We use $g_i^0 \in \mathbb{R}^{d_0}$ for the shared gradient and $g_i^k \in \mathbb{R}^{d_k}$ for the gradient that would be applied to expert k. The embedded gradient is $g_i = (g_i^0, q_{i1} g_i^1, \ldots, q_{iK} g_i^K)$. Define routing overlap $\rho_{ij} = q_i^\top q_j$, raw heterogeneity

$$B_{\mathrm{raw}}^2(w) = \sum_i p_i \|\nabla F_i^{\mathrm{shared}}(w; q_i) - \nabla F_q^{\mathrm{shared}}(w)\|^2, \tag{9}$$

and post-routing heterogeneity

$$B_{\mathrm{route}}^2(w) = \sum_i p_i \|g_i - \bar{g}\|^2, \qquad \bar{g} = \sum_i p_i g_i. \tag{10}$$

Assumption 1 (Smoothness and lower bound)**.** *For the fixed routes q, every $F_i(\cdot; q_i)$ is L-smooth in w, and $F_q(w) \geq F_{q,*}$.*

Assumption 2 (Stochastic gradients)**.** *For an unbiased mini-batch gradient $\hat{g}_i$, $\mathbb{E}\|\hat{g}_i - g_i\|^2 \leq \sigma^2$.*

Assumption 3 (Client sampling)**.** *The round sampler is conditionally unbiased and satisfies*

$$\mathbb{E}_t \left\| \frac{1}{m} \sum_{j=1}^{m} g_{I_j} - \bar{g} \right\|^2 \leq \chi_m B_{\mathrm{route}}^2(w). \tag{11}$$

For i.i.d. draws $I_j \sim p$, $\chi_m = 1/m$; for uniform sampling without replacement with $p_i = 1/N$, $\chi_m = (N-m)/(m(N-1))$.

Assumption 4 (Compiler and router error)**.** *There are ideal routes q_i^* for which recurring conflict modes occupy disjoint or weakly overlapping experts. The learned route satisfies $\mathbb{E}\|q_i - q_i^*\|_1 \leq \varepsilon_q$, and the probability of an incorrect compiled address is at most ε_c.*

The final assumption is measurable: address stability, routing ambiguity, and the raw/centered, clustering, and warm-start ablations diagnose its terms. It is intentionally not hidden inside an unspecified bounded-dissimilarity constant.

B.2 Centered separation and address persistence

The compiler is correct when recurring residual modes are separated by more than their estimation error. This elementary condition makes the clustering term ε_c concrete rather than treating it as an arbitrary oracle error.

Lemma 1 (Separation implies persistent addresses). *Let $v_1, \dots, v_K$ be unit residual-mode directions with $\min_{k \neq l} \|v_k - v_l\|_2 > 2(\delta + \delta_\mu)$. Suppose a centered normalized signature from group k obeys $\|u_i - v_k\|_2 \leq \delta$, and its warm-started center obeys $\|\mu_k - v_k\|_2 \leq \delta_\mu$. Then spherical nearest-center assignment recovers k. If every nonempty group center is updated by a normalized within-group mean whose error remains at most δ_μ, the address identities persist by induction across rounds. If these events fail with probability at most ζ, then $\varepsilon_c \leq \zeta$.*

Proof. For the correct center, $\|u_i - \mu_k\|_2 \leq \delta + \delta_\mu$. For any $l \neq k$, the reverse triangle inequality gives $\|u_i - \mu_l\|_2 \geq \|v_k - v_l\|_2 - (\delta + \delta_\mu) > \delta + \delta_\mu$. Since minimizing Euclidean distance between unit vectors is equivalent to maximizing cosine similarity, spherical assignment selects k. The center-error premise re-establishes the same condition after every update; induction proves persistence. The final statement follows by the union event defining an incorrect address. □

Centering matters because a common update component cancels exactly before normalization. Without subtraction, $s_i = c + r_i$ can be dominated by c, shrinking angular separation between the normalized group directions even when the residual means are well separated. The raw-versus-centered and warm-start ablations test the two premises of Lemma 1.

B.3 Conflict decomposition

Proposition 1 (Routing-overlap conflict bound). *Suppose $\|g_i^k\| \leq G_e$ for every i, k. For any clients i, j,*

$$\langle g_i, g_j \rangle = \langle g_i^0, g_j^0 \rangle + \sum_{k=1}^{K} q_{ik} q_{jk} \langle g_i^k, g_j^k \rangle, \tag{12}$$

$$[-\langle g_i, g_j \rangle]_+ \leq [-\langle g_i^0, g_j^0 \rangle]_+ + G_e^2 \rho_{ij}. \tag{13}$$

In particular, expert conflict vanishes for distinct hard routes and is continuously bounded for soft routes.

Proof. Shared and expert parameters occupy orthogonal coordinate blocks. Expanding the Euclidean inner product therefore eliminates all $k \neq l$ expert cross-terms and gives Equation 12. Since $[-(a + b)]_+ \leq [-a]_+ + [-b]_+$ and Cauchy–Schwarz gives $-\langle g_i^k, g_j^k \rangle \leq G_e^2$, summing with nonnegative $q_{ik} q_{jk}$ yields Equation 13. □

Proposition 1 explains why deleting a conflicting gradient and routing it are not equivalent. Projection methods set part of g_i to zero; FedCRN retains its norm but places it in a block where overlap controls interference. The shared term remains and can still conflict, which is why the consensus optimizer is necessary.

Proposition 2 (Exact hard-routing variance and conflict condition). *Assume ideal hard groups a_i^* under client distribution p_i. Let b_i be the shared-coordinate gradient and r_i the residual-coordinate gradient. Define group mass $\pi_k = \Pr(a_i^* = k)$, group mean $\mu_k = \mathbb{E}[r_i \mid a_i^* = k]$, global residual mean $\bar{\mu} = \sum_k \pi_k \mu_k$, shared heterogeneity $B_b^2 = \mathbb{E}\|b_i - \mathbb{E} b_i\|^2$, and within-group residual variance*

$$W = \sum_k \pi_k \mathbb{E}[\|r_i - \mu_k\|^2 \mid a_i^* = k]. \tag{14}$$

For the disjoint-block embedding $g_i^{\text{route}} = (b_i, e_{a_i^*} \otimes r_i)$,

$$B_{\text{route},*}^2 = B_b^2 + W + \sum_k \pi_k (1 - \pi_k) \|\mu_k\|^2. \tag{15}$$

For a single shared residual block $g_i^{\text{single}} = (b_i, r_i)$,

$$B_{\text{single},*}^2 = B_b^2 + W + \sum_k \pi_k \|\mu_k - \bar{\mu}\|^2. \tag{16}$$

Therefore $B_{\text{route},*}^2 \leq B_{\text{single},*}^2$ exactly when

$$\sum_k \pi_k (1 - \pi_k) \|\mu_k\|^2 \leq \sum_k \pi_k \|\mu_k - \bar{\mu}\|^2. \tag{17}$$

Proof. The shared and residual coordinates are orthogonal, so their variances add. Let $x_i = e_{a_i^*} \otimes r_i$. Because $\mathbb{E}x_i = (\pi_1\mu_1, \ldots, \pi_K\mu_K)$,

$$\mathbb{E}\|x_i - \mathbb{E}x_i\|^2 = \sum_k \pi_k \mathbb{E}[\|r_i\|^2 \mid a_i^* = k] - \sum_k \pi_k^2 \|\mu_k\|^2 \tag{18}$$

$$= W + \sum_k \pi_k(1-\pi_k)\|\mu_k\|^2. \tag{19}$$

Adding B_b^2 proves Equation 15. For a single residual block, the law of total variance gives $\mathbb{E}\|r_i - \bar{\mu}\|^2 = W + \sum_k \pi_k \|\mu_k - \bar{\mu}\|^2$, which proves Equation 16. Comparing the two identities yields Equation 17. □

Thus disjoint coordinates do not universally eliminate group-mean variance. They reduce global residual heterogeneity relative to one shared block only under Equation 17, which is favored by strongly opposing group means and can fail when the means align. This condition makes precise the paper's scope: recurring conflict, rather than arbitrary heterogeneity, is the supervision signal.

With imperfect assignments and bounded $\|r_i\| \le G_e$, moving at most ε_c probability mass between blocks adds at most $4G_e^2\varepsilon_c$. For the block embedding $T(q, r) = (q_1 r, \ldots, q_K r)$, $\|T(q,r)\| \le G_e$ and $\|T(q,r) - T(q^*, r)\| \le G_e\|q - q^*\|_1$. Hence the variance functional $V(X) = \mathbb{E}\|X\|^2 - \|\mathbb{E}X\|^2$ satisfies $|V(T(q,r)) - V(T(q^*,r))| \le 4G_e^2\mathbb{E}\|q - q^*\|_1 \le 4G_e^2\varepsilon_q$. Consequently,

$$B_{\text{route}}^2 \le B_{\text{route},*}^2 + 4G_e^2(\varepsilon_c + \varepsilon_q). \tag{20}$$

B.4 Confidence-gate stability

Lemma 2 (Safe fallback)**.** *Let the residual logit map $R(x, q)$ be L_R-Lipschitz in q and bounded by M_R. For $c(q) = [(\|q\|_\infty - 1/K)/(1 - 1/K)]_+$, the routed perturbation $H(q) = c(q)R(x, q)$ obeys $H(\mathbf{1}/K) = 0$ and*

$$\|H(q) - H(q')\| \le \left(M_R \frac{K}{K-1} + L_R\right)\|q - q'\|_1. \tag{21}$$

Proof. The infinity norm is 1-Lipschitz under ℓ_1, hence $|c(q) - c(q')| \le K\|q - q'\|_1/(K-1)$. Add and subtract $c(q')R(x, q)$, use $0 \le c \le 1$, and apply the triangle inequality. Uniform routing gives $c = 0$. □

This lemma links routing uncertainty to a bounded prediction change. It does not guarantee calibration by itself; the ECE experiment and gate ablation test whether attenuation is useful in practice.

B.5 Nonconvex convergence

We analyze a simplified synchronous local-SGD process for the fixed route vectors $q_1, \ldots, q_N$ defined above. The routes remain exogenous throughout the optimization horizon, so F_q is fixed and the descent terms telescope. The result is therefore a mechanism bound for a route-frozen state, not a convergence proof for the evolving compiler–router dynamics of Algorithm 1. We sample m clients per round, take E local steps of size η, and perform one server aggregation step. Adaptive preconditioning can be incorporated by bounding its diagonal entries above and below; for clarity we state the unpreconditioned form.

Theorem 1 (Convergence under fixed routed heterogeneity)**.** *Under Assumptions 1–4, let $\eta LE \le 1/8$. For T rounds of local SGD,*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F_q(w_t)\|^2 \le \frac{4(F_q(w_0) - F_{q,*})}{\eta ET} + \frac{4L\eta\sigma^2}{m} + 8L\eta E\chi_m \bar{B}_{\text{route}}^2 + 32L^2\eta^2E^2(\bar{B}_{\text{route}}^2 + \sigma^2), \tag{22}$$

where $\bar{B}_{\text{route}}^2 = T^{-1}\sum_t \mathbb{E}B_{\text{route}}^2(w_t)$. Combining with Equation 20 makes compiler and router error explicit. Relative to one shared residual block, routing lowers the heterogeneity terms only when Equation 17 holds. Choosing $\eta = \Theta((LE\sqrt{T})^{-1})$ gives $O(T^{-1/2})$ optimization, stochastic, and client-sampling terms plus $O(T^{-1})$ local drift.

Proof. Let $\hat{g}_t$ be the averaged accumulated local gradient divided by E. Smoothness of the fixed objective gives

$$\mathbb{E}_t F_q(w_{t+1}) \le F_q(w_t) - \eta E\langle \nabla F_q(w_t), \mathbb{E}_t \hat{g}_t\rangle + \frac{L\eta^2 E^2}{2}\mathbb{E}_t\|\hat{g}_t\|^2. \tag{23}$$

Write $\mathbb{E}_t \hat{g}_t = \nabla F_q(w_t) + d_t$, where d_t is local drift. Young's inequality yields

$$-\langle \nabla F_q, \nabla F_q + d_t\rangle \le -\tfrac{3}{4}\|\nabla F_q\|^2 + \|d_t\|^2. \tag{24}$$

For local iterate $w_{t,e}^i$, smoothness and the telescoping local update imply

$$\|d_t\|^2 \le \frac{L^2}{E}\sum_{e=0}^{E-1}\sum_i p_i \mathbb{E}_t\|w_{t,e}^i - w_t\|^2 \tag{25}$$

$$\le 4L^2\eta^2E^2\left(\|\nabla F_q(w_t)\|^2 + B^2_{\text{route}}(w_t) + \sigma^2\right), \tag{26}$$

where the second line follows by expanding the local recursion, applying Jensen's inequality, and separating the client gradient into its global mean and routed deviation. Assumptions 2–3 separate mini-batch noise from partial-participation variance:

$$\mathbb{E}_t\|\hat{g}_t\|^2 \le 2\|\nabla F_q(w_t)\|^2 + 4\|d_t\|^2 + 2\chi_m B^2_{\text{route}}(w_t) + \frac{2\sigma^2}{mE}. \tag{27}$$

Substitute these bounds into Equation 23. Under $\eta LE \le 1/8$, absorb all gradient-norm terms into $-\eta E\|\nabla F_q(w_t)\|^2/4$. Sum over t, telescope the fixed objective $F_q(w_t)$, use $F_q(w_T) \ge F_{q,*}$, and loosen constants to obtain Equation 22. The term $8L\eta E\chi_m \bar{B}^2_{\text{route}}$ is the client-sampling contribution and vanishes under full participation. The final term retains both heterogeneity- and stochastic-drift contributions from Equation 26. Substituting Equation 20 exposes $4G_e^2(\varepsilon_c + \varepsilon_q)$ inside $\bar{B}^2_{\text{route}}$. □

The theorem predicts three testable behaviors, not guaranteed gains. First, strongly opposing residual modes can reduce routed drift only when Equation 17 holds and compiler error remains controlled: low-data Digits benefits, whereas the RotMNIST loss shows that high conflict alone is insufficient. Second, raw clustering and center reinitialization increase the empirical failure modes associated with ε_c. Third, confidence attenuation can reduce effective routing error without maximizing every accuracy metric, matching Table 5.

B.6 Communication and computation

Let P_0 be the shared-path and router parameter count and one rank-d expert contain $P_e = 2Dd + CD$ parameters for a C-class decision residual. In the measured protocol, a selected client downloads $P_0 + KP_e$ values so every f_k is available for the competence target, but uploads only $P_0 + P_e$ values. Server storage is $P_0 + KP_e$. Router computation is $O(DK)$ and expert computation $O(DdK)$ for soft routing (or $O(Dd)$ for hard top-1 routing). Appendix G separately reports per-round uplink, downlink, and their bidirectional total; no cache compression is assumed.

C Reproducibility Details

Data and partitions. MNIST uses 9,000 training and 2,500 test examples. RotMNIST uses the same limits with deterministic translations, blur, and noise. Digits uses 1,250/480 examples resized to 28×28. MultiMNIST places an 80/20 dominant/minority pair of recurring transformation and cyclic-label concept modes in every client. The dominant mode shapes round-level geometry, while the minority mode makes client identity insufficient for prediction. CIFAR-100 uses 12,000/3,000 frozen ResNet-18 embeddings; CIFAR-100-C uses 18,000 source-train and 5,000 corrupted-test embeddings across six episodes. Class proportions are seeded Dirichlet draws. No test example is used for training or hyperparameter selection.

For MultiMNIST, the routing/deployment audit (Table 4 and the top row of Figure 6) uses 25 rounds and seeds 0–4. The component and paired ablation audits (Tables 5–6) use 20 rounds; the paired audit also uses seeds 0–4, with the same partition and evaluation configuration. Thus the routing and ablation accuracy endpoints answer different training-budget questions rather than forming a single matched-round comparison.

Network and optimization. The image trunk has 16- and 32-channel 3×3 convolutions with GroupNorm, GELU, max pooling, and a 64-dimensional projector. Feature benchmarks use a 512–128 MLP with LayerNorm. Each rank-8 GELU residual uses a Kaiming-initialized down projection and zero-initialized up projection; decision residual weights are also zero-initialized. Prespecified local SGD uses learning rate 0.04–0.05, momentum 0.9 (zero for SCAFFOLD), weight decay 5×10^{-4}, batches 128/256, and clipping at 5. Adaptive methods use moments $(.9, .99)$ and stabilizer 10^{-3}; the equal-budget CIFAR-100 selection chooses the values in Table 7.

For the strongest natural-image comparison, FedAdam, SCAFFOLD, FedNova, ClusterMoE, and FEDCRN receive the same three-candidate, three-seed, 20-round selection budget on a fixed stratified 10k/2k split of the CIFAR-100 training pool. The official test pool is never loaded during selection; the chosen setting is rerun for 30 rounds on five paired seeds. CIFAR-100-C reuses the tuple selected on this clean split, with no corruption-specific tuning. Other datasets and methods retain one prespecified configuration. Table 8 and the complete CSV make this limited search auditable; it should not be read as exhaustive tuning.

Table 7: Auditable optimization configuration. Strong CIFAR-100 methods use equal-budget, training-only validation selection; all remaining settings are prespecified.

Method	Local LR	Momentum	Server rule / step	Local epochs	Method coefficient
FedAdam	.04–.05	.9	Adam / .008; C100/C: .016	1	$(\beta_1, \beta_2, \epsilon) = (.9, .99, 10^{-3})$
SCAFFOLD	.04–.05; C100/C: .08	0	mean / 1	1	control variate, coefficient 1
FedNova	.04–.05; C100/C: .08	.9	normalized mean / 1	1	normalized by local steps
ClusterMoE	.04–.05	.9	Adam / .008; C100/C: .016	1	$K = 4$, rank 8, client route
FEDCRN	.04–.05	.9	Adam / .008; C100/C: .016	1	$K = 4$, rank 8, $\lambda = .8$

Table 8: Equal-budget CIFAR-100 validation selection for the five strongest or architecture-matched methods. Best and second-best official-test accuracies within this audited subset are bold and underlined.

Method	Equal-budget candidate grid	Selected	Validation acc. (%) ↑	Test acc. (%) ↑
FedAdam	server step $\{.004, .008, .016\}$	server=.016	26.35	41.49
SCAFFOLD	local LR $\{.02, .05, .08\}$	lr=.08	10.70	21.27
FedNova	local LR $\{.02, .05, .08\}$	lr=.08	9.65	19.89
ClusterMoE	server step $\{.004, .008, .016\}$	server=.016	**28.07**	<u>42.97</u>
FedCRN	server step $\{.004, .008, .016\}$	server=.016	<u>28.05</u>	**43.21**

Execution environment. Experiments ran with Python 3.9.16, PyTorch 2.0.0+cu118, NumPy 1.24.3, scikit-learn 1.3.0, and Matplotlib 3.9.4 on one NVIDIA RTX 4080. Fixed seeds determine data construction, client sampling, transformations, initialization, and local shuffles. Timings cover federated optimization and evaluation after data construction and exclude frozen CIFAR feature extraction; no method component requires specialized hardware.

Result provenance. Run records retain configurations, round-wise aggregate and per-client metrics, routing and conflict diagnostics, communication and parameter counts, and runtime. Paired comparisons reuse identical seeds and partitions, making every reported aggregate, confidence interval, and paired test traceable to its constituent runs.

D PAIRED STATISTICAL TESTS

Table 9: Paired two-sided tests against the strongest optimizer, control-variate baseline, and architecture-matched MoE. Differences and confidence intervals are percentage points over five identical seeds.

Dataset	Paired baseline	Δ acc. (pp)	95% CI	p
MNIST	FedAdam	-0.19	$\pm$0.91	0.589
MNIST	SCAFFOLD	+0.14	$\pm$2.72	0.890
MNIST	ClusterMoE	+0.35	$\pm$1.74	0.605
RotMNIST	FedAdam	+1.65	$\pm$2.30	0.117
RotMNIST	SCAFFOLD	+1.11	$\pm$8.93	0.747
RotMNIST	ClusterMoE	+4.25	$\pm$14.04	0.448
Digits	FedAdam	+1.32	$\pm$5.84	0.565
Digits	SCAFFOLD	+2.28	$\pm$11.34	0.607
Digits	ClusterMoE	-1.14	$\pm$4.24	0.497
CIFAR-100	FedAdam	+1.73	$\pm$1.14	0.014
CIFAR-100	SCAFFOLD	+21.95	$\pm$1.04	$< .001$
CIFAR-100	ClusterMoE	+0.24	$\pm$0.87	0.485
CIFAR-100-C	FedAdam	+0.88	$\pm$0.57	0.012
CIFAR-100-C	SCAFFOLD	+13.06	$\pm$1.29	$< .001$
CIFAR-100-C	ClusterMoE	+0.20	$\pm$0.68	0.466

Table 10: Fifty-client CIFAR-100 scalability at 10%, 20%, and 50% participation using the same 30-round budget and five paired seeds as Table 3 (accuracy in percent with 95% CI). Thus the 20% column is the same operating point, not a shorter-run experiment.

Method	10% clients	20% clients	50% clients
FedAvg	12.77 $\pm$ 3.01	16.68 $\pm$ 1.74	18.43 $\pm$ 1.62
FedProx	12.77 $\pm$ 3.00	16.67 $\pm$ 1.72	18.43 $\pm$ 1.60
SCAFFOLD	16.41 $\pm$ 3.84	21.27 $\pm$ 1.88	24.29 $\pm$ 0.56
FedDyn	12.78 $\pm$ 3.00	16.69 $\pm$ 1.75	18.46 $\pm$ 1.61
MOON	13.05 $\pm$ 3.00	17.04 $\pm$ 1.98	18.63 $\pm$ 1.79
FedAdam	27.09 $\pm$ 1.04	41.49 $\pm$ 1.68	<u>48.56</u> $\pm$ 1.34
FedNova	15.28 $\pm$ 3.06	19.89 $\pm$ 2.78	21.56 $\pm$ 2.49
FedBABU	8.01 $\pm$ 2.60	11.14 $\pm$ 1.25	12.87 $\pm$ 1.06
FedALA	5.66 $\pm$ 1.32	8.58 $\pm$ 0.54	15.01 $\pm$ 1.32
FedLWS	11.79 $\pm$ 3.17	15.69 $\pm$ 2.66	17.59 $\pm$ 2.25
ClusterMoE	<u>28.41</u> $\pm$ 1.40	<u>42.97</u> $\pm$ 1.57	48.44 $\pm$ 1.56
FedCRN	**28.98** $\pm$ 1.02	**43.21** $\pm$ 1.10	**49.65** $\pm$ 1.16

E ROUTING AND SENSITIVITY

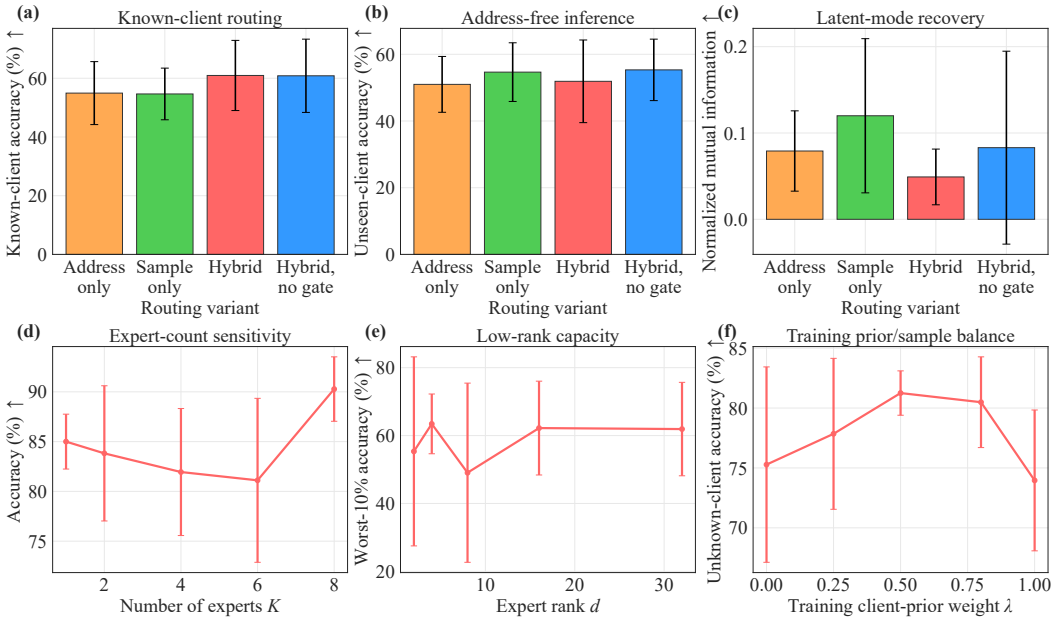


Figure 6: Routing validation and sensitivity. Top: 25-round known-client accuracy, address-free unseen-client accuracy, and latent-mode NMI for four routing variants. Bottom: 20-round sensitivity to expert count, rank, and the client-prior weight used during training; every unseen-client evaluation sets $\lambda = 0$ at inference. Error bars denote two-sided Student-t 95% confidence intervals across independent seeds.

The complete CSV additionally sweeps signature dimension. The intended conclusion is a stable operating region, not that $K = 4$ or $\lambda = .8$ is universally optimal. In particular, $\lambda = 1$ removes within-client adaptation and $\lambda = 0$ removes the stabilizing address prior.

F Controlled Centering Diagnostic

We first increase power on the real MultiMNIST federation while holding every non-centering choice fixed. The negative but nonsignificant estimate scopes our claim: centering is the designed directional signal, but the current end-to-end benchmark does not establish an accuracy benefit from centering alone.

Table 11: End-to-end centered-versus-raw spherical compiler audit over 10 paired seeds (20 rounds). Intervals are 95% CIs; the paired difference has $p = .269$.

Variant	Accuracy ↑ (%)	Paired Δ vs. raw (pp)
Raw + spherical	**56.43 ± 6.61**	–
Centered (full)	54.92 ± 5.19	-1.51 ± 2.90

For each of 50 seeds, we sample one 64-dimensional isotropic Gaussian common direction and normalize it. Three Gaussian residual directions are projected orthogonally to that common direction and orthonormalized; the fourth is their normalized negative sum, giving four separated modes. Each mode contains 30 clients (120 total). A client scales its mode by an independent $\mathrm{LogNormal}(0, 0.55)$ draw and receives isotropic Gaussian signature noise with standard deviation .18. A second observation adds independent Gaussian noise with standard deviation .12 and measures warm-start address persistence. Each seed regenerates directions, scales, and noise, while raw and centered compilers receive the exact same signatures and initialization seed.

Table 12: Isolated common-component intervention over 50 paired seeds. $\|c\|$ scales the shared nuisance component; all intervals are 95% CIs and p is from a two-sided paired test. Best and second within each raw/centered pair are bold and underlined.

$\|c\|$	Metric	Raw	Centered	Δ [95% CI]	Paired p
0.0	Purity	0.967 ± 0.017	**0.971 ± 0.012**	$+0.004$ $[-0.015, +0.023]$	0.665
0.0	Persistence (ARI)	**0.965 ± 0.010**	0.962 ± 0.009	-0.003 $[-0.013, +0.007]$	0.517
0.0	Silhouette	0.319 ± 0.007	**0.325 ± 0.007**	$+0.007$ $[+0.006, +0.007]$	$< .001$
1.0	Purity	**0.975 ± 0.010**	0.971 ± 0.012	-0.004 $[-0.019, +0.011]$	0.596
1.0	Persistence (ARI)	**0.973 ± 0.009**	0.962 ± 0.009	-0.011 $[-0.021, -0.0003]$	0.044
1.0	Silhouette	0.320 ± 0.007	**0.325 ± 0.007**	$+0.005$ $[+0.004, +0.006]$	$< .001$
2.0	Purity	**0.978 ± 0.005**	0.971 ± 0.012	-0.007 $[-0.018, +0.005]$	0.261
2.0	Persistence (ARI)	**0.976 ± 0.007**	0.962 ± 0.009	-0.014 $[-0.024, -0.004]$	0.008
2.0	Silhouette	0.320 ± 0.006	**0.325 ± 0.007**	$+0.006$ $[+0.004, +0.007]$	$< .001$
4.0	Purity	0.940 ± 0.021	**0.971 ± 0.012**	$+0.031$ $[+0.007, +0.056]$	0.012
4.0	Persistence (ARI)	0.948 ± 0.017	**0.962 ± 0.009**	$+0.014$ $[-0.005, +0.033]$	0.135
4.0	Silhouette	0.316 ± 0.006	**0.325 ± 0.007**	$+0.010$ $[+0.008, +0.012]$	$< .001$
8.0	Purity	0.869 ± 0.034	**0.971 ± 0.012**	$+0.102$ $[+0.066, +0.138]$	$< .001$
8.0	Persistence (ARI)	0.925 ± 0.031	**0.962 ± 0.009**	$+0.037$ $[+0.006, +0.069]$	0.022
8.0	Silhouette	0.309 ± 0.007	**0.325 ± 0.007**	$+0.016$ $[+0.013, +0.020]$	$< .001$

As $\|c\|$ grows, centered statistics are invariant by construction, whereas raw spherical signatures eventually collapse toward the common direction. The low-nuisance rows are negative controls rather than uniformly favorable cases. At $\|c\| = 8$, centering improves purity by .102 (95% CI reported in Table 12) and preserves both address persistence and angular separation. This controlled result supports Lemma 1's geometric mechanism without converting the nonsignificant end-to-end result in Table 11 into a stronger claim.

G Efficiency Accounting

Let P_0 be the shared-path/router size and $P_e = 2Dd + CD$ one residual expert. The measured downlink is $P_0 + KP_e$, uplink is $P_0 + P_e$, and server storage is $P_0 + KP_e$. Soft routing costs

Table 13: Bidirectional software-simulation accounting on Digits (12 clients, 50% participation, 30 rounds). All non-MiB columns count parameters; shared, server, client-active, per-round transfer, and cumulative counts are separated. The complete expert bank is downloaded for Equation 6; only the addressed expert is uploaded.

Method	Shared	Server	Active	Up./round	Down./round	Cum. up. (MiB)	Cum. bidir. (MiB)
FedAvg	**105,962**	**105,962**	**105,962**	**105,962**	**105,962**	**72.76**	**145.52**
FedProx	**105,962**	**105,962**	**105,962**	**105,962**	**105,962**	**72.76**	**145.52**
SCAFFOLD	**105,962**	**105,962**	**105,962**	**105,962**	**105,962**	**72.76**	**145.52**
FedDyn	**105,962**	**105,962**	**105,962**	**105,962**	**105,962**	**72.76**	**145.52**
MOON	**105,962**	**105,962**	**105,962**	**105,962**	**105,962**	**72.76**	**145.52**
FedAdam	**105,962**	**105,962**	**105,962**	**105,962**	**105,962**	**72.76**	**145.52**
FedNova	**105,962**	**105,962**	**105,962**	**105,962**	**105,962**	**72.76**	**145.52**
FedBABU	**105,962**	**105,962**	**105,962**	**105,962**	**105,962**	**72.76**	**145.52**
FedALA	**105,962**	**105,962**	**105,962**	**105,962**	**105,962**	**72.76**	**145.52**
FedLWS	**105,962**	**105,962**	**105,962**	**105,962**	**105,962**	**72.76**	**145.52**
ClusterMoE	106,222	112,878	107,886	107,886	112,878	74.08	151.59
FedCRN	106,222	112,878	107,886	107,886	112,878	74.08	151.59

$O(DdK)$ residual operations, while top-1 routing costs $O(Dd)$. The measured time is included for reproducibility but is not a physical-device or network benchmark.

Table 14: Compute scaling on one RTX 4080 with batch size 256. MAC counts cover the residual and competence paths whose cost changes with K; latency is the median of 200 synchronized forward passes.

Backbone	K	Server params	Upload params	Soft MACs	Top-1 MACs	Latency (ms)
CNN (28x28)	1	**107,691**	**107,691**	**2,304**	**2,304**	**0.293**
CNN (28x28)	2	109,420	107,756	4,608	**2,304**	0.326
CNN (28x28)	4	112,878	107,886	9,216	**2,304**	0.377
CNN (28x28)	8	119,794	108,146	18,432	**2,304**	0.527
ResNet-18 feature MLP	1	**95,845**	**95,845**	**29,696**	**29,696**	**0.108**
ResNet-18 feature MLP	2	112,870	95,974	59,392	**29,696**	0.169
ResNet-18 feature MLP	4	146,920	96,232	118,784	**29,696**	0.239
ResNet-18 feature MLP	8	215,020	96,748	237,568	**29,696**	0.394

The parameter bank and soft residual MACs grow linearly with K, whereas the communicated path and hypothetical top-1 MACs remain nearly constant. The measured batch latency rises from 0.293 to 0.527 ms for the CNN and from 0.108 to 0.394 ms for the feature MLP between $K = 1$ and $K = 8$; these are local software measurements, not network or device-system claims.