WHAT DOES THE GLOBAL MODEL LEAVE BEHIND? CLIENT-CONDITIONED RESIDUAL EXPERTS FOR FEDERATED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

A shared federated model can retain systematic, client-dependent errors after training. We ask what predictive structure remains after shared learning and whether it can be exploited to improve predictions without updating the deployed model. We introduce FedCoREx (Federated Client-Conditioned Residual Experts), which freezes a deployment-selected shared model and federatively learns a bank of shared residual experts together with a client-conditioned router. The router uses frozen sample representations, base prediction probabilities, and train-only client contexts to combine the experts' logit corrections. Each context summarizes the base model's class-conditional prediction behavior, and the combined correction is added to the unchanged base logits. We show theoretically that the remaining log-loss improvement available after freezing the shared predictor has two sources: sample-only model misspecification and additional client-conditioned information. Realizing this improvement requires both an expressive residual family and effective learning. Statistical heterogeneity alone therefore does not imply correctable residual structure. Experiments on four datasets show substantial gains under controlled visual concept shifts, modest gains on FEMNIST, and near-neutral results on Sentiment140. Client-conditioned routing is needed to recover the large visual gains in the tested architectures, and correction improves multiple shared-model bases.

1 INTRODUCTION

Federated learning trains a shared predictor from decentralized data (McMahan et al., 2017). Advances in heterogeneous optimization improve this predictor (Karimireddy et al., 2020; Li et al., 2025), yet its remaining errors can still depend systematically on the client. Personalized models and mixtures of experts exploit client-dependent prediction (Li et al., 2021a; Marfoq et al., 2021; Farhat et al., 2025). We study a complementary deployment problem: *what predictive structure does a selected shared model leave behind, and can clients exploit it through a shared conditional layer while keeping that model fixed?*

Clients with identical input and label marginals can have different input–label correspondences, leaving conflicting targets for a sample-only predictor. Client identity can add no label information even when input supports are disjoint, provided the frozen sample state is sufficient. We therefore ask: *given what the frozen model exposes about a sample, what does client context add?*

We formalize this question through population log-loss. Allowing arbitrary residual functions of the frozen sample state and client context defines *information-level headroom*: the greatest possible loss reduction using that information. It decomposes into model misspecification given the sample state and the conditional mutual information between the label and client context. The gain attained by a learned correction further depends on its function-family approximation gap and learning gap.

We introduce **FedCoREx**, *Federated Client-Conditioned Residual Experts* (Figure 1). Each client summarizes the frozen base's class-conditional prediction behavior in a train-only context c_i. A shared router combines this context with the base representation $h_B(x)$ and probabilities $p_B(x)$ to

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

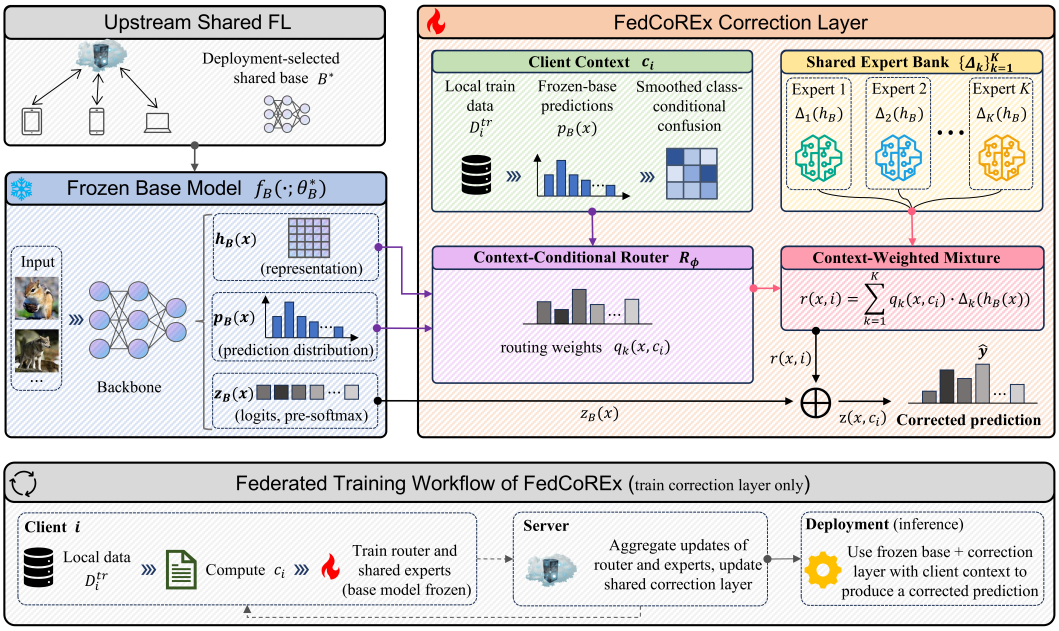

Figure 1: Overview of FedCoREx. A deployment-selected shared base remains frozen and provides $h_B(x)$, $p_B(x)$, and $z_B(x)$. Each client constructs a train-only context c_i; a shared router uses $[h_B(x); p_B(x); c_i]$ to compose residual experts into $r(x,i)$, which is added to $z_B(x)$. Only router and expert parameters are trained federatively; zero-output initialization recovers the base predictor. The lower panel summarizes context construction, correction learning, aggregation, and deployment.

compose shared residual experts:

$$z(x,i) = z_B(x) + \sum_{k=1}^{K} q_k(x,c_i)\Delta_k\big(h_B(x)\big). \tag{1}$$

The experts produce additive logit corrections (Jacobs et al., 1991; He et al., 2016); only they and the router are trained and aggregated. Zero-output initialization recovers the base predictor. All trainable neural parameters are shared, with client-specific information supplied by the locally stored context.

Contributions.

1. **Residual-correction analysis.** We characterize the improvement available after fixing a shared predictor, separating information-level headroom from approximation and learning gaps, and show why marginal heterogeneity alone cannot determine correction opportunities.
2. **Shared conditional residual experts.** FedCoREx learns sample- and client-conditioned corrections on the exact deployed base using train-only contexts and shared neural parameters.
3. **Evidence across tasks and bases.** Experiments on four datasets characterize when conditional correction is useful. Structural controls test the roles of client context and residual composition, and comparisons across six shared-model bases assess portability across upstream training methods.

2 Related Work

Shared federated optimization. Starting from FedAvg (McMahan et al., 2017), federated optimizers address heterogeneous local updates through regularization and drift correction (Li et al., 2020; Karimireddy et al., 2020; Acar et al., 2021), update normalization, server adaptation, and

participation-variance reduction (Wang et al., 2020; Reddi et al., 2021; Jhunjhunwala et al., 2022). Sharpness-aware and symbolic updates further target generalization and communication (Qu et al., 2022; Li et al., 2025; Lu et al., 2025; Sun et al., 2023). Local batch normalization and logit calibration address feature and label skew, respectively (Li et al., 2021b; Zhang et al., 2022). These methods shape the predictor during upstream training; FedCoREx learns a conditional correction after its deployment checkpoint has been fixed.

Personalization and conditional prediction. Client specificity can reside in related private models (Smith et al., 2017; Dinh et al., 2020; Li et al., 2021a), locally adapted initializations (Fallah et al., 2020), or personalized heads and representations (Collins et al., 2021; Oh et al., 2022). Other methods combine local and global predictions for each sample (Panchal et al., 2023; Guo et al., 2021), accommodate heterogeneous client architectures (Yi et al., 2026), or mix collaboratively learned distribution-specific predictors (Marfoq et al., 2021). Personalization can also use private modules, prompt experts, or exchanged prediction heads (Feng et al., 2025; Luo et al., 2025; Xie et al., 2025). FedCoREx conditions a shared correction on a fixed local summary of the base model's training-data predictions.

Expert composition and residual learning. Classical mixtures learn gating over specialized functions (Jacobs et al., 1991); sparse MoEs scale conditional computation (Shazeer et al., 2017; Lepikhin et al., 2021; Fedus et al., 2022), with balanced assignment and expert-choice routing controlling allocation (Lewis et al., 2021; Zhou et al., 2022). Federated experts serve as user-selected predictors or jointly trained specialists (Reisser et al., 2021; Farhat et al., 2025), shared generalists and local specialists (Fan et al., 2025), and server-side or group-balanced ensembles (Jiang et al., 2025; Zeng et al., 2023). In FedCoREx, experts produce additive logit corrections (He et al., 2016), combined by a dense, client-conditioned router to correct a frozen shared predictor.

3 Problem Formulation and Residual-Correction Analysis

Let $J \sim \omega$ index a fixed population of N participating clients, with $(X, Y) \mid J = i \sim \mathbb{P}_i$ and labels in $\{1, \dots, C\}$. A deployment-selected shared base model B^* provides the frozen sample state $S = (H, Q_B)$, where $H = h_B(X)$ and $Q_B = p_B(X) = \mathrm{softmax}(z_B(X))$. Client i also has a fixed context $\xi_i = \Gamma(D_i^{\mathrm{tr}}; B^*)$ built only from its training data. Writing $\Xi = \xi_J$, we seek a shared residual $r_\theta(S, \Xi)$ that improves prediction while leaving B^* unchanged. Section 4 instantiates the abstract context ξ_i as the confusion-derived vector c_i.

Let $\mathcal{R}_B$ be the base's population log-loss and $\mathcal{R}^*_{S,\Xi}$ the infimum of population log-loss over all measurable, finite-valued additive residual functions of (S, Ξ). This unrestricted function class allows any residual architecture using the fixed sample state and context. The difference $\mathcal{R}_B - \mathcal{R}^*_{S,\Xi}$ is the information-level headroom: the greatest possible reduction in population log-loss using that information. For a fixed realized base and context map, finite base risk, and positive base probabilities, Theorem 1 (Appendix A.3) gives

$$\mathcal{R}_B - \mathcal{R}^*_{S,\Xi} = \mathbb{E}_S D_{\mathrm{KL}}(\mathbb{P}(Y = \cdot \mid S) \,\|\, q_B(\cdot \mid S)) + I(Y; \Xi \mid S). \tag{2}$$

Here $q_B(\cdot \mid S) = Q_B$, and all quantities use the same client weights ω. The first term is the base's misspecification given its own sample state; the second is the additional label information retained by the context. The gain realized by a finitely parameterized residual family also depends on its approximation gap and the excess population loss of the learned correction over the family optimum.

The ideal additive correction is a log-probability ratio between the desired conditional predictor and the frozen base. Input and label marginals do not determine how much label information client context adds to the frozen sample state. FedCoREx uses a finitely parameterized shared residual family whose composition depends on S and client context, with the deployed predictor held fixed. Appendices A–B provide the full residual-correction analysis, confusion-context identifiability and estimation results, and proofs.

4 FEDCOREX

FedCoREx learns a shared correction to a frozen, deployment-selected predictor. Figure 1 illustrates the architecture and federated workflow. Freezing the selected base fixes the prediction errors to be corrected. Each client summarizes its class-conditional confusions using local training data; the router combines this summary with the current sample state to weight shared residual experts. Clients then jointly train the router and experts, keeping the base and contexts fixed, and validation selects the complete correction for deployment. Algorithms 1–3 give the corresponding context-construction, training, and inference procedures. We consider participating clients with a common label space and local training data available for context construction.

4.1 FROZEN BASE AND TRAIN-ONLY CLIENT CONTEXT

The base B^* is exactly the validation-selected predictor evaluated in the standalone comparison. It provides frozen representations $h_B(x) \in \mathbb{R}^{d_h}$, logits $z_B(x) \in \mathbb{R}^C$, and probabilities $p_B(x) = \mathrm{softmax}(z_B(x))$. Each client summarizes the frozen base's local class-conditional prediction behavior through confusion counts on its training split:

$$M_i[a,b] = \sum_{(x,y)\in D_i^{\mathrm{tr}}} \mathbf{1}\{y=a\}\mathbf{1}\{\arg\max_j z_{B,j}(x) = b\}. \tag{3}$$

Rows index local labels and columns index predicted labels. With $n_{i,a} = \sum_b M_i[a,b]$, Laplace smoothing and row normalization give

$$P_i[a,b] = \frac{M_i[a,b]+\alpha}{n_{i,a}+C\alpha}, \qquad c_i = \mathrm{vec}_{\mathrm{row}}(P_i) \in \mathbb{R}^{C^2}, \quad \alpha = 1. \tag{4}$$

An unobserved class contributes a uniform row. Row normalization describes class-conditional prediction behavior separately from class frequency: clients with similar label proportions can still exhibit different systematic confusions. Each c_i is constructed once using deterministic preprocessing, kept locally, and reused throughout correction training and inference.

4.2 SHARED EXPERTS AND CONTEXT-CONDITIONED ROUTING

Each shared expert k, parameterized by ψ_k, maps the frozen representation to a signed logit correction:

$$\Delta_k(h) = W_{2k}\,\mathrm{ReLU}(W_{1k}h + b_{1k}) \in \mathbb{R}^C, \qquad k = 1,\dots,K. \tag{5}$$

The hidden width is W. Expert outputs depend only on the sample representation; client dependence enters their composition through the router:

$$u_i(x) = \mathcal{N}\big([\mathcal{N}(h_B(x)); p_B(x); \mathcal{N}(c_i)]\big), \qquad q(x,c_i) = \mathrm{softmax}\big(\bar{W}_R u_i(x)\big). \tag{6}$$

Here $\mathcal{N}$ is non-affine per-vector layer normalization: the representation and context are normalized separately, then the concatenated vector is normalized. The probabilities enter directly. The router parameters are $\phi = W_R$; $\bar{W}_R$ applies row-wise ℓ_2 normalization with a numerical floor (Appendix C.3). The mixture is dense, with $q_k \geq 0$ and $\sum_k q_k = 1$.

The final predictor adds the weighted residual to the base logits:

$$\begin{aligned} r(x,i;\Theta_{\mathrm{corr}}) &= \sum_{k=1}^{K} q_k(x,c_i)\Delta_k(h_B(x)), \\ z(x,i) &= z_B(x) + r(x,i;\Theta_{\mathrm{corr}}), \qquad \widehat{p}(y \mid x,i) = \mathrm{softmax}(z(x,i))_y. \end{aligned} \tag{7}$$

All experts are evaluated, and the base remains an always-on path with coefficient one.

The router combines systematic client-level prediction patterns with the current example's representation and base probabilities. This lets the same expert functions contribute differently across clients and across examples within a client.

4.3 FEDERATED CORRECTION-ONLY LEARNING

Clients jointly learn the router and experts, $\Theta_{\text{corr}} = \{\phi, \psi_1, \ldots, \psi_K\}$, using only cross-entropy on the corrected prediction:

$$\mathcal{L}_i(\Theta_{\text{corr}}; \mathcal{B}_i) = -\frac{1}{|\mathcal{B}_i|} \sum_{(x,y)\in\mathcal{B}_i} \log \widehat{p}(y \mid x, i), \qquad \mathcal{B}_i \subseteq D_i^{\text{tr}}. \tag{8}$$

The base and contexts remain fixed. At round t, selected clients $\mathcal{S}_t$ start from the shared correction $\Theta_{\text{corr}}^{(t-1)}$ and perform E local SGD epochs. They return their updated correction parameters and training-sample counts n_i; data and contexts stay local. The server applies sample-weighted aggregation to the complete correction:

$$\Theta_{\text{corr}}^{(t)} = \sum_{i\in\mathcal{S}_t} \frac{n_i}{\sum_{j\in\mathcal{S}_t} n_j} \Theta_{\text{corr},i}^{(t)}. \tag{9}$$

The shared task loss couples expert learning and routing: routing probabilities weight expert gradients, and expert outputs determine the router's gradients.

4.4 INITIALIZATION AND DEPLOYMENT SELECTION

Random input projections and zero output projections initialize the correction to the unchanged base:

$$W_{2k}^{(-1)} = 0 \quad \text{for all } k \quad \Longrightarrow \quad r(x, i; \Theta_{\text{corr}}^{(-1)}) = 0 \quad \Longrightarrow \quad z^{(-1)}(x, i) = z_B(x). \tag{10}$$

This initialization, indexed by -1, is a deployment candidate alongside the completed correction rounds. Candidates are compared by client-macro validation negative log-likelihood (NLL),

$$\mathcal{J}_{\text{val}}(t) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{|D_i^{\text{val}}|} \sum_{(x,y)\in D_i^{\text{val}}} -\log \widehat{p}_t(y \mid x, i), \tag{11}$$

giving the selected correction

$$t_{\text{corr}}^* \in \operatorname*{arg\,min}_{t\in\{-1,0,\ldots,T_{\text{corr}}-1\}} \mathcal{J}_{\text{val}}(t), \qquad \Theta_{\text{corr}}^* = \Theta_{\text{corr}}^{(t_{\text{corr}}^*)}. \tag{12}$$

Router and experts are restored together from this candidate. Each participating client predicts using the same base and correction weights with its stored c_i. Including zero correction ensures that the selected validation NLL is no worse than the base's; test performance is assessed separately.

Appendix C gives the exact network operations, initialization, training settings, and complete algorithms; Appendix I provides parameter, computation, and communication accounting.

5 EXPERIMENTS

We evaluate correction gains across data distributions and shared-model training methods, ablate the correction architecture, and examine how the gains relate to conventional heterogeneity measures. The central comparison pairs a validation-selected base B^* with that *same* base plus its learned correction, $B^* + C^*(B^*)$. We report both absolute performance and the paired gain from correction.

5.1 EXPERIMENTAL SETUP

Data and models. We evaluate CIFAR-10, Fashion-MNIST, FEMNIST, and Sentiment140 (Krizhevsky, 2009; Xiao et al., 2017; Caldas et al., 2019; Go et al., 2009), each with 50 participating clients and disjoint local training, validation, and test splits. The controlled image settings comprise IID, label skew (Label), client-group label permutations (Concept), and label permutations combined with image rotations (Feature+Concept, FC). FEMNIST and Sentiment140 preserve natural writer and user identities. We use a LeNet-5 variant for images and a compact TextCNN with train-only vocabulary fitting for text (LeCun et al., 1998; Kim, 2014).

Comparisons and budgets. The main baselines are FedAvg, FedProx, FedGMT, Vanilla-MoE, and an adapted DDOME; CIFAR-10 additionally includes SCAFFOLD, FedDyn, and FedSMU. Shared-model and end-to-end MoE baselines train for 500 rounds with 20% participation, three local epochs, and batch size 32. FedCoREx uses $K = 4, W = 16$ and adds 200 correction rounds to the selected base. A 700-round FedGMT continuation controls for the additional round budget. Method-specific optimization rules are retained, and FedCoREx settings are fixed across bases. The evaluated DDOME adaptation uses an anchor-specific schedule and a pretrained FedGMT routing encoder.

Selection and reporting. Base and correction checkpoints minimize client-macro validation NLL, with zero correction included as a candidate. Each correction is paired with the exact base that supplied its representations and contexts. The primary metric is client-macro test accuracy; secondary metrics include Bottom-10% accuracy (the mean of each method's five lowest client accuracies), NLL, pooled macro-F1, and sample-weighted accuracy. Results summarize three training seeds by means and sample standard deviations, with differences computed within matched base–correction pairs. Accuracy and F1 differences are percentage points (pp); negative $\Delta\mathrm{NLL} = \mathrm{Corrected} - \mathrm{Base}$ denotes improvement. Appendices D–E give dataset construction, model and baseline details, hyperparameters, and the complete evaluation protocol.

5.2 Main results: correction gains depend on the setting

Table 1: Client-macro test accuracy (%, mean ± sample SD over three seeds). FedCoREx uses the frozen FedGMT base. Panels separate datasets and client semantics. Shading identifies FedCoREx; bold marks each condition's highest displayed mean.

Method	(a) CIFAR-10			
	IID	Label	Concept	FC
FedAvg	66.00 ± 0.37	59.71 ± 0.49	46.12 ± 0.63	35.89 ± 0.77
FedProx	65.90 ± 0.46	59.46 ± 0.32	45.73 ± 0.70	35.53 ± 0.15
SCAFFOLD	66.67 ± 0.45	58.89 ± 0.70	45.75 ± 0.71	35.45 ± 0.90
FedDyn	67.87 ± 0.51	64.28 ± 1.11	49.13 ± 0.33	39.84 ± 0.31
FedSMU	70.02 ± 0.77	63.20 ± 0.60	49.03 ± 1.11	38.08 ± 1.07
FedGMT	74.86 ± 0.35	**73.56 ± 0.80**	56.47 ± 0.91	45.79 ± 0.86
Vanilla-MoE	67.56 ± 0.13	63.16 ± 0.85	48.38 ± 0.65	37.41 ± 0.45
DDOME (adapted)	50.16 ± 2.96	53.56 ± 3.31	17.56 ± 2.84	13.17 ± 2.69
FedCoREx (FedGMT base)	**74.88 ± 0.34**	73.45 ± 0.80	**67.92 ± 3.34**	**54.94 ± 1.63**
Method	(b) Fashion-MNIST			
	IID	Label	Concept	FC
FedAvg	88.41 ± 0.14	87.29 ± 0.12	64.77 ± 0.07	56.09 ± 0.15
FedProx	88.43 ± 0.08	87.16 ± 0.26	64.25 ± 0.72	55.58 ± 0.51
FedGMT	91.79 ± 0.26	91.17 ± 0.05	68.67 ± 0.15	64.93 ± 0.27
Vanilla-MoE	88.55 ± 0.39	87.50 ± 0.21	64.20 ± 0.32	56.13 ± 0.92
DDOME (adapted)	85.53 ± 0.98	79.28 ± 7.93	39.53 ± 20.13	16.59 ± 2.48
FedCoREx (FedGMT base)	**91.84 ± 0.17**	**92.02 ± 0.21**	**84.33 ± 1.65**	**81.69 ± 1.57**
Method	(c) Natural-client partitions			
	FEMNIST (writers)		Sentiment140 (users)	
FedAvg	61.79 ± 0.60		65.21 ± 0.59	
FedProx	61.43 ± 0.25		65.14 ± 0.49	
FedGMT	71.45 ± 0.51		65.95 ± 1.37	
Vanilla-MoE	62.31 ± 1.17		65.66 ± 1.68	
DDOME (adapted)	49.09 ± 0.41		58.94 ± 2.30	
FedCoREx (FedGMT base)	**72.89 ± 1.80**		**66.20 ± 1.16**	

FC denotes Feature+Concept. Standalone methods use 500 rounds; FedCoREx adds 200 correction rounds to its base. SCAFFOLD, FedDyn, and FedSMU are evaluated on CIFAR-10. DDOME has the adaptation and dependency described in Section 5.1.

Correction under visual concept shifts. Table 1 shows that FedGMT is already a strong shared base in these settings. FedCoREx further improves Concept and FC accuracy on both image datasets: paired gains range from 9.15 to 16.77 pp (Table 2). Mean Bottom-10% accuracy, NLL, and F1 also improve in these conditions.

Table 2: Paired effects of FedCoREx over the *same* deployed FedGMT base. All deltas are Corrected−Base. Accuracy, Bottom-10%, and F1 differences are in pp; lower NLL is better.

Dataset	Setting	Paired test effect (three seeds)			
		$\Delta A \uparrow$ (mean ± SD)	Δ Bottom↑	Δ NLL↓	Δ F1↑
CIFAR-10	IID	$+0.022 \pm 0.073$	$+0.044$	-0.00108	$+0.029$
	Label	-0.107 ± 0.096	$+0.133$	-0.02491	-0.157
	Concept	$+11.444 \pm 2.957$	$+13.378$	-0.29281	$+11.611$
	FC	$+9.151 \pm 1.271$	$+11.600$	-0.26823	$+9.272$
Fashion-MNIST	IID	$+0.052 \pm 0.090$	-0.111	-0.01148	$+0.048$
	Label	$+0.856 \pm 0.206$	$+2.667$	-0.02329	$+0.866$
	Concept	$+15.652 \pm 1.570$	$+12.852$	-0.46911	$+15.639$
	FC	$+16.767 \pm 1.830$	$+15.519$	-0.46147	$+16.875$
FEMNIST	Natural	$+1.440 \pm 1.493$	$+1.987$	-0.06022	$+0.972$
Sentiment140	Natural	$+0.255 \pm 0.244$	-0.672	$+0.00045$	$+0.088$

A: client-macro accuracy. Other metric columns show mean paired deltas. Weighted-accuracy gains on FEMNIST and Sentiment140 are $+1.592$ and $+0.259$ pp, respectively.

Other data settings. IID accuracy remains essentially unchanged on both image datasets. Correction under label skew has metric- and dataset-dependent effects: CIFAR-10 improves mean NLL with a small accuracy decline, while Fashion-MNIST has a smaller positive accuracy effect. FEMNIST provides a modest natural-client gain in both accuracy and NLL. Sentiment140 remains near-neutral: its small mean accuracy increase accompanies worse Bottom-10% accuracy and slightly higher NLL.

The five-seed CIFAR-10 confirmation gives mean gains of 11.752 pp on Concept and 9.267 pp on FC, with descriptive 95% intervals of $[8.812, 14.692]$ and $[7.952, 10.581]$ pp (Table 17). IID and Label remain near zero. These intervals summarize training variability for the fixed client population and data realization.

5.3 Mechanism controls: conditional residual composition

We compare a single width-64 residual (Direct64), four width-16 residual experts averaged uniformly (Uniform), a sample-conditioned router with the client context removed (NoContext), and full FedCoREx. All use the same frozen base, correction budget, and validation selection rule. Direct64 and Uniform match the expert-bank parameter budget; Full additionally includes the router.

Table 3: Structural controls on CIFAR-10. Accuracy (%) and NLL are three-seed means. A learned router without client context, uniform expert averaging, and a single residual fail to reproduce the full correction's Concept/FC accuracy gains.

Correction variant	Parameters	Concept		FC		Client context
		Acc↑	NLL↓	Acc↑	NLL↓	
None (FedGMT base)	0	56.471	1.20639	45.791	1.54431	–
Direct64	6,080	56.231	1.20138	45.524	1.53670	No
Uniform 4×16	6,080	56.031	1.20072	45.662	1.53659	No
Routed 4×16, NoContext	6,856	56.018	1.20315	45.520	1.53450	Masked
Full FedCoREx	6,856	**67.916**	**0.91358**	**54.942**	**1.27608**	Yes

Direct64 and Uniform exclude the 776 router parameters. These controls compare alternative correction architectures.

Table 3 shows that all three controls remain close to the base in accuracy, despite small NLL reductions. Among the tested architectures, only the full client-conditioned construction recovers the large Concept/FC gains.

Table 4: Client-context ablation across settings. Accuracy columns are three-seed means (%). The paired Full−NoContext contrast is in pp.

Dataset	Setting	Client-macro accuracy↑			Full−NoContext (pp)
		Base	NoContext	Full	
CIFAR-10	Concept	56.471	56.018	67.916	+11.898
	FC	45.791	45.520	54.942	+9.422
Fashion-MNIST	Concept	68.674	68.226	84.326	+16.100
	FC	64.926	64.578	81.693	+17.115
FEMNIST	Natural	71.445	71.310	72.885	+1.575
Sentiment140	Natural	65.950	66.139	66.205	+0.066

Removing client context. In Table 4, the full construction exceeds NoContext by 9.422–17.115 pp on average in the visual Concept/FC conditions. The mean advantage is smaller on FEMNIST (1.575 pp) and near zero on Sentiment140 (0.066 pp), whose NLL and tail contrasts also remain unfavorable. These architectural contrasts reflect both the residual family's expressiveness and how effectively it is learned.

Continued base training. Continuing FedGMT from 500 to 700 rounds yields 56.640% accuracy under Concept and 45.609% under FC, compared with 56.471% and 45.791% at the original budget. FedCoREx remains ahead by 11.276 and 9.333 pp, respectively. The continuation matches the correction stage's additional round budget, participation schedule, and local-task randomization. The stages optimize different parameters, so their computational and communication costs differ (full results in Table 23).

5.4 Portability across shared-model bases

For each base method, we independently select its deployment checkpoint and fit FedCoREx while keeping that checkpoint fixed. Table 5 reports six shared-model bases on CIFAR-10. Mean gains range from 5.956 to 12.929 pp under Concept and from 4.871 to 9.151 pp under FC. Mean NLL also improves for the SCAFFOLD, FedDyn, and FedSMU corrections.

Table 5: CIFAR-10 portability across six evaluated bases. Base and +FedCoREx are mean client-macro accuracies (%). Paired deltas are in pp. All entries summarize three seeds.

Shared base	Concept			Feature+Concept		
	Base	+FedCoREx	ΔA	Base	+FedCoREx	ΔA
FedAvg	46.120	55.858	+9.738	35.893	40.764	+4.871
FedProx	45.733	55.560	+9.827	35.533	41.298	+5.764
FedGMT	56.471	67.916	+11.444	45.791	54.942	+9.151
SCAFFOLD	45.751	54.289	+8.538	35.449	41.991	+6.542
FedDyn	49.133	62.062	+12.929	39.840	45.782	+5.942
FedSMU	49.031	54.987	+5.956	38.076	42.978	+4.902

Each correction is paired with its own deployed base.

FedDyn has the largest Concept accuracy gain; FedGMT yields the highest corrected accuracy. The bases thus rank differently by correction gain and final performance, and their rankings can also differ between NLL and accuracy.

Corrections to FedAvg, FedProx, and FedGMT improve mean accuracy and NLL across the Fashion-MNIST Concept/FC settings and yield smaller positive mean accuracy effects on FEMNIST. On Sentiment140, mean accuracy declines for FedAvg and FedProx and improves only slightly for FedGMT (Tables 18–19).

5.5 Additional analyses

Expert granularity. A sensitivity analysis at near-equal parameter budgets shows positive mean gains for two, four, and eight routed experts on CIFAR-10 Concept/FC. Eight experts improve mean accuracy over four by 3.182 pp on Concept and 1.302 pp on FC (Table 25), despite the narrower individual experts. The main configuration remains fixed at four experts. Validation routing diagnostics help interpret how the bank is used (Table 27). With four experts, exponentiating the mean per-example routing entropy gives effective counts of 1.358 on Concept and 1.209 on FC. All experts remain active in this dense mixture; Appendix G.5 reports bank-wide usage.

Deployment and training cost. The additional training stage includes dense expert evaluation, frozen-base forward passes, and all-client validation. Relative to 500-round FedGMT training, the 200-round correction adds approximately 10.6–15.3% logical communication and 41.3–50.6% cumulative round time across datasets. These implementation-specific measurements include separate same-workload executions; Appendix I.4 specifies their sources and measurement scope.

Residual-structure diagnostics. Training-side confusion diagnostics distinguish the visual Concept/FC gains from label skew more clearly than marginal heterogeneity measures. On CIFAR-10, excess confusion dispersion is 0.1180 for Concept and 0.0852 for FC, versus approximately 0.0002 under Label (Table 30). This descriptor measures cross-client variation in class-conditional predictions after subtracting a support-matched sampling null mean. It describes one projection of residual structure; the conditional information and learning gaps in Section 3 remain unestimated (Appendix H.1). FEMNIST still improves with near-zero excess confusion dispersion, illustrating the diagnostic's limited coverage.

6 Discussion

Correction after shared learning. Correction complements upstream optimization by addressing client-dependent errors in the selected predictor. In the tested Concept/FC configurations, extra base-training rounds and context-free residual capacity alone yield little improvement; the shared client-conditioned correction recovers large gains.

From available information to realized gain. The decomposition helps interpret weak as well as strong correction gains. On Sentiment140, a small gain could reflect limited useful conditional information, loss of that information in the chosen context, or approximation and learning gaps. Distinguishing these explanations requires evidence beyond the observed gain itself.

Context design and scale. The train-only confusion profile summarizes client-dependent prediction behavior and admits an identifiability argument for permutation-style concept shifts. Theorem 2 makes this link precise: under its shared-channel and positive class-support assumptions, distinct prediction-channel rows yield distinct population contexts for different client label permutations. Its main practical limitation is the combination of C^2 state and sparse per-class support as the label space grows. FEMNIST illustrates this tradeoff through a larger router and sparsely supported confusion rows. Further work should test more compact contexts, larger backbones, and client populations beyond the 50-client settings studied here.

7 Conclusion

We studied what remains learnable after a shared federated model has been selected for deployment. FedCoREx keeps that predictor fixed and learns shared residual experts with a router conditioned on frozen sample state and train-only client context. Our analysis separates information-level correction headroom from the expressiveness and learning needed to realize it. Experiments reveal substantial gains under structured visual concept shifts, smaller gains on FEMNIST, and near-neutral outcomes on Sentiment140. In the visual Concept/FC controls, only the full client-conditioned correction recovers the large gains, which also extend across the evaluated shared-model bases. Shared federated predictors can thus serve as fixed bases for further collaborative learning of client-conditioned corrections.

AI USE STATEMENT

In this work, we used generative AI tools for the following tasks with recommended disclosure: language polishing, grammar and spelling correction, and formatting optimization. We have not used generative AI tools for any tasks with required disclosure, such as generating synthetic datasets, developing theoretical models, formulating mathematical claims, or implementing methods. The remaining tasks with required disclosure are not applicable to this work.

We reviewed all AI-assisted content. Specifically, all text edited by AI tools was manually verified by the authors for correctness and intended meaning. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide an anonymized implementation of FedCoREx in the supplementary material, including data preparation scripts, experiment configurations, and training and evaluation commands. All experiments use publicly available datasets, with acquisition and preprocessing instructions included in the code package. Appendices C–E detail the algorithms, model architectures, hyperparameters, and evaluation protocol, while Appendices A–B provide the theoretical assumptions and proofs. The source code will also be made publicly available.

REFERENCES

Durmus Alp Emre Acar, Yue Zhao, Ramon Matas, Matthew Mattina, Paul Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *International Conference on Learning Representations*, 2021.

Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečný, H. Brendan McMahan, Virginia Smith, and Ameet Talwalkar. LEAF: A benchmark for federated settings. In *Workshop on Federated Learning for Data Privacy and Confidentiality*, 2019.

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. PMLR, 2021.

Canh T. Dinh, Nguyen Tran, and Josh Nguyen. Personalized federated learning with moreau envelopes. In *Advances in Neural Information Processing Systems*, volume 33, pp. 21394–21405. Curran Associates, Inc., 2020.

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. In *Advances in Neural Information Processing Systems*, volume 33, pp. 3557–3568. Curran Associates, Inc., 2020.

Dongyang Fan, Bettina Messmer, Nikita Doikov, and Martin Jaggi. On-device collaborative language modeling via a mixture of generalists and specialists. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 15833–15861. PMLR, 2025.

Yehya Farhat, Hamza ElMokhtar Shili, Fangshuo Liao, Chen Dun, Mirian Hipolito Garcia, Guoqing Zheng, Ahmed Awadallah, Robert Sim, Dimitrios Dimitriadis, and Anastasios Kyrillidis. Learning to specialize: Joint gating-expert training for adaptive MoEs in decentralized settings. In *Advances in Neural Information Processing Systems*, volume 38, pp. 2515–2549. Curran Associates, Inc., 2025.

William Fedus, Barret Zoph, and Noam Shazeer. Switch Transformers: Scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.*, 23(1), 2022.

Yu Feng, Yangli-ao Geng, Yifan Zhu, Zongfu Han, Xie Yu, Kaiwen Xue, Haoran Luo, Mengyang Sun, Guangwei Zhang, and Meina Song. PM-MOE: Mixture of experts on private model parameters for personalized federated learning. In *Proceedings of the ACM on Web Conference 2025*, pp. 134–146. Association for Computing Machinery, 2025.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. CS224N project report, Stanford University, 2009.

Binbin Guo, Yuan Mei, Danyang Xiao, and Weigang Wu. PFL-MoE: Personalized federated learning based on mixture of experts. In *Web and Big Data*, pp. 480–486. Springer International Publishing, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

Divyansh Jhunjhunwala, Pranay Sharma, Aushim Nagarkatti, and Gauri Joshi. FedVARP: Tackling the variance due to partial client participation in federated learning. In *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180, pp. 906–916. PMLR, 2022.

Jingang Jiang, Yanzhao Chen, Xiangyang Liu, Haiqi Jiang, and Chenyou Fan. Heterogeneous federated learning with scalable server mixture-of-experts. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, 2025.

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

Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1746–1751, Doha, Qatar, 2014. Association for Computational Linguistics.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. GShard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*, 2021.

Mike Lewis, Shruti Bhosale, Tim Dettmers, Naman Goyal, and Luke Zettlemoyer. BASE layers: Simplifying training of large, sparse models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6265–6274. PMLR, 2021.

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.

Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6357–6368. PMLR, 2021a.

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*, 2021b.

Yuhang Li, Tong Liu, Yangguang Cui, Ming Hu, and Xiaoqiang Li. One arrow, two hawks: Sharpness-aware minimization for federated learning via global model trajectory. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 35317–35341. PMLR, 2025.

Xinyi Lu, Hao Zhang, Chenglin Li, Weijia Lu, Zhifei Yang, Wenrui Dai, Xiaodong Zhang, Xiaofeng Ma, Can Zhang, Junni Zou, and Hongkai Xiong. FedSMU: Communication-efficient and generalization-enhanced federated learning through symbolic model updates. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 40903–40929. PMLR, 2025.

Jun Luo, Chen Chen, and Shandong Wu. Mixture of experts made personalized: Federated prompt learning for vision-language models. In *International Conference on Learning Representations*, volume 2025, pp. 58061–58080, 2025.

Othmane Marfoq, Giovanni Neglia, Aurélien Bellet, Laetitia Kameni, and Richard Vidal. Federated multi-task learning under a mixture of distributions. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2021.

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

Jaehoon Oh, SangMook Kim, and Se-Young Yun. FedBABU: Toward enhanced representation for federated image classification. In *International Conference on Learning Representations*, 2022.

Kunjal Panchal, Sunav Choudhary, Nisarg Parikh, Lijun Zhang, and Hui Guan. Flow: Per-instance personalized federated learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Zhe Qu, Xingyu Li, Rui Duan, Yao Liu, Bo Tang, and Zhuo Lu. Generalized federated learning via sharpness aware minimization. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 18250–18280. PMLR, 2022.

Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and Hugh Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations*, 2021.

Matthias Reisser, Christos Louizos, Efstratios Gavves, and Max Welling. Federated mixture of experts. *CoRR*, abs/2107.06724, 2021.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Yan Sun, Li Shen, Tiansheng Huang, Liang Ding, and Dacheng Tao. FedSpeed: Larger local interval, less communication round, and higher generalization accuracy. In *International Conference on Learning Representations*, 2023.

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, pp. 7611–7623. Curran Associates, Inc., 2020.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.

Luyuan Xie, Tianyu Luan, Wenyuan Cai, Guochen Yan, Zhaoyu Chen, Nan Xi, Yuejian Fang, Qingni Shen, Zhonghai Wu, and Junsong Yuan. dFLMoE: Decentralized federated learning via mixture of experts for medical data analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10203–10213, 2025.

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.

Yaopei Zeng, Lei Liu, Li Liu, Li Shen, Shaoguo Liu, and Baoyuan Wu. Global balanced experts for federated long-tailed learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4792–4802, 2023.

Jie Zhang, Zhiqi Li, Bo Li, Jianghe Xu, Shuang Wu, Shouhong Ding, and Chao Wu. Federated learning with label distribution skew via logits calibration. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 26311–26329. PMLR, 2022.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Y. Zhao, Andrew Dai, Zhifeng Chen, Quoc Le, and James Laudon. Mixture-of-experts with expert choice routing. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2022.

APPENDIX CONTENTS

Reading guide. For the correction headroom and its relation to learned gains, see Appendix A, especially Theorem 1 and Appendix A.4. Appendix B explains when confusion contexts distinguish concept shifts and how finite samples affect their estimation. For the exact model and end-to-end procedures, see Appendix C, including Algorithms 1–3. Appendices D–E specify the data, training settings, checkpoint selection, and metrics. Full performance and portability results are in Appendix F; structural controls and expert diagnostics in Appendix G; heterogeneity diagnostics in Appendix H; and resource accounting in Appendix I.

A Residual-Correction Theory **16**

A.1 Setup and assumptions 16

A.2 Levels of attainable improvement 17

A.3 Conditional residual risk decomposition 17

A.4 Consequences for context and learning 18

A.5 Optimal additive logit correction 19

A.6 Why marginal heterogeneity is insufficient 20

A.7 Zero-residual containment 20

B Theory of the Confusion Context **21**

B.1 Permutation-style concept-shift model 21

B.2 Identifiability and repeated-row ambiguity 21

B.3 Finite-sample estimation and smoothing 22

B.4 Missing classes and data-dependent contexts 23

C Method Specification **24**

C.1 Frozen-base interface and deployed state 24

C.2 Train-only context construction 24

C.3 Correction architecture 24

C.4 Correction training and checkpoint selection 25

C.5 Context, training, and inference algorithms 26

D Experimental Setup **26**

D.1 Client populations 26

D.2 Controlled shift definitions 28

D.3 Backbones and preprocessing 29

D.4 Training budgets and hyperparameters 30

D.5 Baseline implementations 30

E Evaluation and Statistical Protocol **31**

E.1 Checkpoint-aligned evaluation and model selection 31

E.2 Metric definitions 32

E.3 Pairing and uncertainty summaries 32

F Full Performance and Base Portability **33**

F.1 Full absolute performance 33

F.2 Same-base paired correction effects 33

F.3 Five-seed confirmation 34

F.4 Portability across shared bases 36

G Mechanism Controls and Expert Granularity **37**

G.1 Structural controls and capacity accounting 37

G.2 Structural and context ablations 38

G.3 Extended-base-training controls 39

G.4 Expert granularity at near-equal budgets 39

G.5 Permutation-invariant routing diagnostics 41

H Heterogeneity and Residual-Structure Diagnostics **41**

H.1 Diagnostic definitions 42

H.2 Controlled-condition census 43

H.3 Descriptive alignment at the condition level 44

H.4 FEMNIST and construction sensitivity 45

I Complexity and Resource Accounting **45**

I.1 Analytic parameter and state counts 45

I.2 Computation and communication 46

I.3 Measured resource costs 46

I.4 Hardware and measurement scope 47

A RESIDUAL-CORRECTION THEORY

We establish the residual-risk decomposition in Section 3 and analyze how much population log-loss can be reduced after selecting a shared model for deployment. For the participating clients, we distinguish the improvement allowed by the available information, the improvement attainable within a fixed residual family, and the gain achieved by a learned member of that family.

A.1 SETUP AND ASSUMPTIONS

Let $\mathcal{I} = \{1, \dots, N\}$ be the client population and $\mathcal{Y} = \{1, \dots, C\}$ the label space. A population example is drawn as

$$J \sim \omega, \qquad (X, Y) \mid J = i \sim \mathbb{P}_i, \qquad \omega_i > 0, \quad \sum_{i=1}^{N} \omega_i = 1. \tag{A.1}$$

Uniform weights $\omega_i = 1/N$ define client-macro risk, whereas $\omega_i = n_i / \sum_j n_j$ define a sample-size-weighted risk for fixed client sample counts n_i. All risks and information quantities below use the same fixed ω. These evaluation weights need not equal the aggregation weights used by the training algorithm.

The shared base B^* has already been trained and its deployment checkpoint selected. We condition on this realized base and the training data used to construct client contexts, and take expectations over fresh population examples. Write

$$H = h_B(X), \qquad Z_B = z_B(X), \qquad Q_B = \mathrm{softmax}(Z_B), \qquad S = (H, Q_B). \tag{A.2}$$

The state S consists of the frozen representation and class probabilities. We write $q_B(y \mid S) = Q_{B,y}$ for the base's predictive distribution, corresponding to $p_B(x)$ in the introduction. Assume finite base logits almost surely, so that $q_B(y \mid S) > 0$, and finite base risk $\mathcal{R}_B = \mathbb{E}[-\log q_B(Y \mid S)]$. Conditional distributions are assumed to exist, and all logarithms are natural.

Contexts and residual predictors. Client i constructs a context $\xi_i = \Gamma(D_i^{\mathrm{tr}}; B^*)$ using only its training data and the frozen base. After conditioning on these realized quantities, $\Xi = \xi_J$ is a deterministic function of client identity. Here ξ_i denotes the train-only confusion context c_i used by the method. The unrestricted analysis allows any such fixed context map. A finite-valued measurable residual $r(S, \Xi) \in \mathbb{R}^C$ induces

$$q_r(y \mid S, \Xi) = \frac{q_B(y \mid S) \exp(r_y(S, \Xi))}{\sum_{b=1}^{C} q_B(b \mid S) \exp(r_b(S, \Xi))}, \qquad \mathcal{R}(r) = \mathbb{E}[-\log q_r(Y \mid S, \Xi)]. \tag{A.3}$$

This is exactly $\mathrm{softmax}(Z_B + r(S, \Xi))$. The common shift of the base logits cancels, so S suffices to define the corrected distribution.

Let $\mathcal{R}_S^*$ be the infimum of $\mathcal{R}(r)$ over all measurable residuals depending only on S, and $\mathcal{R}_{S,\Xi}^*$ the corresponding infimum over residuals depending on (S, Ξ). These unrestricted classes characterize information-level limits. For the actual parameterized family $\mathcal{F} = \{r_\theta : \theta \in \Theta\}$, define

$$\mathcal{R}_{\mathcal{F}}^* = \inf_{\theta \in \Theta} \mathcal{R}(r_\theta), \qquad \mathcal{R}(\theta) = \mathcal{R}(r_\theta). \tag{A.4}$$

We use infima because finite logits cannot attain a conditional distribution with zero probabilities exactly. FedCoREx contains the zero residual through zero-output expert initialization. Consequently, $\mathcal{R}_{S,\Xi}^* \le \mathcal{R}_{\mathcal{F}}^* \le \mathcal{R}_B$.

For FedCoREx specifically, the family has the form

$$r_\theta(S, \Xi) = \sum_{k=1}^{K} q_{\phi,k}(S, \Xi) \Delta_{\psi_k}(H), \qquad \theta = (\phi, \psi_1, \dots, \psi_K). \tag{A.5}$$

The weights $q_{\phi,k}$ specify probabilities over experts. Experts receive H, whereas the router receives the normalized sample state and context specified in the method. The fixed widths, normalization operations, and shared parameters are part of $\mathcal{F}$; the unrestricted classes impose none of these architectural restrictions.

Regularity. The state spaces admit regular conditional distributions, as holds for the finite and Euclidean spaces used here. Unrestricted residuals are measurable and finite-valued almost surely, but need not be uniformly bounded. We additionally assume $\mathcal{R}(\widehat{\theta}) < \infty$ when discussing learned gain. Since $\mathcal{F}$ contains zero, its risk infimum is finite even if some other members have infinite risk. These conditions avoid differences of infinite risks.

Table 6: Notation for the residual-correction analysis. All population quantities use the same fixed client weights ω.

Symbol	Meaning
$J, \omega, \mathbb{P}_i$	Client identity, population weights, and client data law.
B^*, H, Z_B, Q_B	Selected frozen base, representation, logits, and probabilities.
$S = (H, Q_B), \Xi = \xi_J$	Frozen sample state and train-only client context.
$q_B, q_r, \mathcal{F}$	Base and corrected class distributions; actual residual family.
$\mathcal{R}_B, \mathcal{R}(\theta)$	Base and learned-model population log-loss.
$\mathcal{R}^*_S, \mathcal{R}^*_{S,\Xi}$	Unrestricted sample-only and sample–context risk infima.
$\mathcal{R}^*_{\mathcal{F}}$	Risk infimum within the actual family.
$\mathcal{H}_{\text{info}}, \mathcal{H}_{\text{family}}$	Information-level and family-level headroom.
$\mathcal{G}_{\text{learned}}$	Gain achieved by the learned parameters.
$\mathcal{A}_{\mathcal{F}}, \mathcal{E}_{\text{learn}}$	Approximation and learning gaps.

A.2 Levels of attainable improvement

We define information-level headroom, family-level headroom, and learned gain as follows:

$$\mathcal{H}_{\text{info}} = \mathcal{R}_B - \mathcal{R}^*_{S,\Xi}, \qquad \text{information-level headroom,} \tag{A.6}$$

$$\mathcal{H}_{\text{family}} = \mathcal{R}_B - \mathcal{R}^*_{\mathcal{F}}, \qquad \text{family-level headroom,} \tag{A.7}$$

$$\mathcal{G}_{\text{learned}} = \mathcal{R}_B - \mathcal{R}(\widehat{\theta}), \qquad \text{learned gain.} \tag{A.8}$$

For any learned $\widehat{\theta} \in \Theta$ with finite risk,

$$\mathcal{G}_{\text{learned}} \le \mathcal{H}_{\text{family}} \le \mathcal{H}_{\text{info}}, \qquad 0 \le \mathcal{H}_{\text{family}}. \tag{A.9}$$

The first two headrooms are nonnegative, but the learned gain need not be. Held-out log-loss differences provide empirical evidence about learned gain, rather than identifying the unrestricted information-level headroom. The identities below concern log-loss; accuracy gains require separate evaluation.

The ordering follows directly from nesting the residual classes and including $r \equiv 0$ in $\mathcal{F}$.

A.3 Conditional residual risk decomposition

Lemma 1 (Log-loss infimum under an observable state)**.** *Under the assumptions in Appendix A.1, let $V = S$ or $V = (S, \Xi)$. The infimum of population log-loss over all measurable, almost-surely finite additive residuals of V is $H(Y \mid V)$. This remains true when the conditional label distribution has zero entries.*

Proof. Let V denote either S or (S, Ξ) and let $p_V(y) = \mathbb{P}(Y = y \mid V)$. For any strictly positive predictive distribution $q(\cdot \mid V)$,

$$\begin{aligned} \mathbb{E}[-\log q(Y \mid V)] &= \mathbb{E}_V\left[-\sum_y p_V(y) \log q(y \mid V)\right] \\ &= H(Y \mid V) + \mathbb{E}_V D_{\text{KL}}(p_V \,\|\, q(\cdot \mid V)). \end{aligned} \tag{A.10}$$

Here $0 \log 0 = 0$, and terms with $p_V(y) = 0$ contribute zero. This is the conditional form of the logarithmic-score identity discussed by Gneiting & Raftery (2007). Nonnegativity of KL gives the lower bound $H(Y \mid V)$.

To show that finite additive residuals can approach this bound even when p_V has zero entries, fix $\varepsilon \in (0,1)$ and set

$$p_{V,\varepsilon}(y) = (1-\varepsilon)p_V(y) + \frac{\varepsilon}{C}, \qquad r_{\varepsilon,y}(V) = \log p_{V,\varepsilon}(y) - \log q_B(y \mid S). \tag{A.11}$$

These residuals are measurable and finite almost surely. Their induced distribution is exactly $p_{V,\varepsilon}$, since the numerator in Eq. (A.3) equals $p_{V,\varepsilon}(y)$ and sums to one. For every class with $p_V(y) > 0$, $p_{V,\varepsilon}(y) \geq (1-\varepsilon)p_V(y)$. Consequently,

$$0 \leq D_{\mathrm{KL}}(p_V \,\|\, p_{V,\varepsilon}) \leq -\log(1-\varepsilon), \tag{A.12}$$

and Eq. (A.10) yields

$$H(Y \mid V) \leq \mathcal{R}(r_\varepsilon) \leq H(Y \mid V) - \log(1-\varepsilon). \tag{A.13}$$

Taking $\varepsilon \downarrow 0$ proves that the unrestricted infimum equals $H(Y \mid V)$. No attainment or interchange of a limit and an expectation is required. □

Theorem 1 (Conditional residual risk decomposition)**.** *Under the setup above, the unrestricted residual risks satisfy*

$$\mathcal{R}_S^* = H(Y \mid S), \qquad \mathcal{R}_{S,\Xi}^* = H(Y \mid S, \Xi). \tag{A.14}$$

The information-level headroom decomposes as

$$\mathcal{R}_B - \mathcal{R}_{S,\Xi}^* = \underbrace{\mathbb{E}_S D_{\mathrm{KL}}(\mathbb{P}(Y = \cdot \mid S) \,\|\, q_B(\cdot \mid S))}_{\textit{sample-only model misspecification}} + \underbrace{I(Y; \Xi \mid S)}_{\textit{additional context information}}. \tag{A.15}$$

Proof. Lemma 1, applied separately to $V = S$ and $V = (S, \Xi)$, gives Eq. (A.14). Applying Eq. (A.10) to the frozen predictor gives

$$\mathcal{R}_B = H(Y \mid S) + \mathbb{E}_S D_{\mathrm{KL}}(\mathbb{P}(Y = \cdot \mid S) \,\|\, q_B(\cdot \mid S)). \tag{A.16}$$

Subtract $\mathcal{R}_{S,\Xi}^* = H(Y \mid S, \Xi)$ and use $H(Y \mid S) - H(Y \mid S, \Xi) = I(Y; \Xi \mid S)$. This proves Eq. (A.15). All entropies are bounded by $\log C$, and the assumed finite base risk makes the expected KL term finite. □

The first term measures what a sample-only correction could improve: even given its own representation and probabilities, the base may not output the true conditional label distribution. It is broader than a calibration error conditioned only on the predicted probabilities, since S also includes H. The second term measures label information retained by the context beyond that sample state. Thus, useful correction can exist without additional client information, and additional client information can remain even when the base is optimal among sample-only predictors.

A.4 Consequences for context and learning

Corollary 1 (Ideal context value)**.** *Under the assumptions of Theorem 1,*

$$\mathcal{R}_S^* - \mathcal{R}_{S,\Xi}^* = I(Y; \Xi \mid S). \tag{A.17}$$

Context strictly improves the unrestricted optimal log-loss if and only if $I(Y; \Xi \mid S) > 0$.

Proof of Corollary 1. Subtract the two identities in Eq. (A.14). The resulting conditional entropy difference is $I(Y; \Xi \mid S)$. Since conditional mutual information is nonnegative, the unrestricted improvement is strict exactly when this quantity is positive. □

Corollary 2 (Decomposition of learned gain)**.** *Define the approximation gap and learning gap by*

$$\mathcal{A}_{\mathcal{F}} = \mathcal{R}_{\mathcal{F}}^* - \mathcal{R}_{S,\Xi}^*, \qquad \mathcal{E}_{\mathrm{learn}} = \mathcal{R}(\widehat{\theta}) - \mathcal{R}_{\mathcal{F}}^*. \tag{A.18}$$

Both are nonnegative, and

$$\boxed{\mathcal{G}_{\mathrm{learned}} = \mathcal{H}_{\mathrm{info}} - \mathcal{A}_{\mathcal{F}} - \mathcal{E}_{\mathrm{learn}}.} \tag{A.19}$$

Proof of Corollary 2. The actual family is contained in the unrestricted state-dependent family, so $\mathcal{R}^*_{\mathcal{F}} \geq \mathcal{R}^*_{S,\Xi}$. By definition of an infimum, $\mathcal{R}(\widehat{\theta}) \geq \mathcal{R}^*_{\mathcal{F}}$. These give nonnegativity of both gaps. Adding and subtracting the two infima gives

$$\begin{aligned}\mathcal{R}_B - \mathcal{R}(\widehat{\theta}) = (\mathcal{R}_B - \mathcal{R}^*_{S,\Xi}) - (\mathcal{R}^*_{\mathcal{F}} - \mathcal{R}^*_{S,\Xi}) \\ - (\mathcal{R}(\widehat{\theta}) - \mathcal{R}^*_{\mathcal{F}}),\end{aligned} \tag{A.20}$$

which is Eq. (A.19). The inclusion of zero also implies $\mathcal{R}^*_{\mathcal{F}} \leq \mathcal{R}_B$, giving Eq. (A.9). □

The approximation gap reflects the expressiveness of the chosen residual family given its available state. The learning gap includes the effects of finite training data and imperfect optimization relative to the population family optimum. Equation (A.19) allows a small observed gain to arise from limited information, limited expressiveness, or unsuccessful learning.

Corollary 3 (Context as a deterministic statistic of client identity)**.** *For the fixed context map* $\Xi = \xi_J$*,*

$$I(Y; J \mid S) = I(Y; \Xi \mid S) + I(Y; J \mid S, \Xi) \geq I(Y; \Xi \mid S). \tag{A.21}$$

Proof of Corollary 3. Once the realized training data and base are fixed, Ξ is a deterministic function of J. Hence $I(Y; \Xi \mid S, J) = 0$. Expanding the same conditional mutual information in two orders gives

$$\begin{aligned}I(Y; J, \Xi \mid S) &= I(Y; J \mid S) + I(Y; \Xi \mid S, J) = I(Y; J \mid S), \\ I(Y; J, \Xi \mid S) &= I(Y; \Xi \mid S) + I(Y; J \mid S, \Xi).\end{aligned} \tag{A.22}$$

Equating these expressions and using nonnegativity proves the result. □

Information discarded by the context is already excluded from $\mathcal{H}_{\text{info}}$; family approximation error is defined relative to the information that remains. Equality in Eq. (A.21) holds when identity adds no label information beyond (S, Ξ). In particular, a deterministic context can be injective on a finite client population. Its construction alone therefore guarantees neither strict information compression nor protection against client memorization.

To interpret context ablations, consider restricted sample-only and context-dependent families with approximation gaps $\mathcal{A}_S$ and $\mathcal{A}_{S,\Xi}$ relative to their respective unrestricted risks, and learning gaps $\mathcal{E}_S$ and $\mathcal{E}_{S,\Xi}$. Their *population* risk difference satisfies

$$\mathcal{R}(\widehat{\theta}_S) - \mathcal{R}(\widehat{\theta}_{S,\Xi}) = I(Y; \Xi \mid S) + (\mathcal{A}_S - \mathcal{A}_{S,\Xi}) + (\mathcal{E}_S - \mathcal{E}_{S,\Xi}). \tag{A.23}$$

An empirical context ablation combines these information, approximation, and learning effects with test-sample variation, so its measured gain need not equal the information term.

A.5 Optimal additive logit correction

Proposition 1 (Optimal additive logit residual)**.** *At a state* (s, ξ) *where* $p^*_y = \mathbb{P}(Y = y \mid S = s, \Xi = \xi) > 0$ *for every class, the residuals that minimize conditional log-loss are exactly*

$$r^*_y(s, \xi) = \log \frac{p^*_y}{q_B(y \mid s)} + a(s, \xi), \qquad y = 1, \ldots, C, \tag{A.24}$$

where $a(s, \xi)$ *is any finite class-independent shift. If some* $p^*_y = 0$*, the same optimal risk is approached by a sequence of finite residuals, but no finite residual attains it.*

Proof. Fix (s, ξ) with $p^*_y > 0$ for all y. Conditional log-loss is $H(p^*) + D_{\mathrm{KL}}(p^* \| q_r)$, so it is minimized exactly when $q_r = p^*$. Substituting Eq. (A.24) into Eq. (A.3) gives numerator $p^*_y \exp(a)$ and denominator $\exp(a)$, establishing sufficiency. Conversely, if $q_r = p^*$, let $Z_r = \sum_b q_B(b \mid s) \exp(r_b(s, \xi))$. Taking logarithms of $p^*_y = q_B(y \mid s) \exp(r_y(s, \xi))/Z_r$ gives $r_y(s, \xi) = \log(p^*_y / q_B(y \mid s)) + \log Z_r$. This is the stated form with a common finite shift.

If any $p^*_y = 0$, every finite residual still produces strictly positive probabilities, so it cannot equal p^*. The construction in Eq. (A.11) nevertheless approaches p^* in conditional log-loss by Eq. (A.12). Thus the optimal risk is an infimum at such a state. □

The ideal residual is a log-probability ratio up to a class-independent shift, which leaves softmax unchanged. The approximation gap $\mathcal{A}_{\mathcal{F}}$ captures the limitations of representing this target with a finitely parameterized shared expert bank.

A.6 Why marginal heterogeneity is insufficient

Proposition 2 (Marginals do not determine conditional residual information)**.** *There exist federated populations and frozen states for which: (i) client input supports are disjoint and $I(X;J) = H(J)$, yet $I(Y;J \mid S) = 0$; and (ii) all clients have identical input marginals and identical label marginals, yet $I(Y;J \mid S) > 0$.*

Proof. It suffices to give two finite constructions with two labels, written as $\{0,1\}$. In both, set the base logits to zero, so $Q_B = (1/2, 1/2)$ and $\mathcal{R}_B = \log 2$. The base representation specified below is compatible with this constant output, for example through a zero final prediction layer.

Disjoint input supports with no additional identity information. Let J be uniform on $N \geq 2$ clients and let U be an independent uniform binary variable. Set $X = (J, U)$, $Y = U$, and $H = U$. The supports of X across clients are disjoint because the first coordinate identifies J. Therefore $H(J \mid X) = 0$ and

$$I(X;J) = H(J) = \log N. \tag{A.25}$$

Since Q_B is constant, S carries exactly the information in U. The label is already determined by S, so

$$H(Y \mid S) = H(Y \mid S, J) = 0, \qquad I(Y;J \mid S) = 0. \tag{A.26}$$

This example has no additional identity-dependent information, but it still has sample-only headroom $\log 2$ because the base probabilities are uniform.

Identical marginals with positive identity information. For this construction, index the two clients by $J \in \{0,1\}$. Let J and X be independent uniform binary variables, set $Y = X \oplus J$, and take $H = X$. On each client, X and Y individually have the same uniform binary distribution. Thus both the input and label marginals agree across clients. However, given S, which carries exactly X, the unknown J leaves Y uniform. Once J is supplied, Y is determined:

$$H(Y \mid S) = \log 2, \qquad H(Y \mid S, J) = 0, \qquad I(Y;J \mid S) = \log 2. \tag{A.27}$$

The two constructions prove the proposition. □

The second construction changes the conditional label mechanism $\mathbb{P}_i(Y \mid X)$. Preserving this identity information depends on the context: an injective context retains it, whereas a constant context gives $I(Y;\Xi \mid S) = 0$. Indeed, with the constant base prediction used here and a fixed tie-breaking rule, the population class-conditional hard-confusion matrices have identical rows on all clients. This provides a concrete case in which the chosen confusion statistic does not capture the available identity information.

A.7 Zero-residual containment

Remark 1 (Zero-residual containment). Zero-output initialization gives parameters θ^{-1} with $r_{\theta^{-1}} \equiv 0$ and $q_{\theta^{-1}} = q_B$ exactly. This establishes $\mathcal{R}^*_{\mathcal{F}} \leq \mathcal{R}_B$ at the function-family level. It does not establish the same inequality for a learned model.

The unchanged base is also included as the round -1 candidate in validation selection. Let $\widehat{\mathcal{R}}_{\mathrm{val}}(t)$ denote the fixed validation loss used to compare the base and correction checkpoints under identical evaluation semantics. If

$$t^* \in \arg\min_{t \in \{-1,0,\ldots,T_{\mathrm{corr}}-1\}} \widehat{\mathcal{R}}_{\mathrm{val}}(t), \tag{A.28}$$

then candidate containment gives

$$\widehat{\mathcal{R}}_{\mathrm{val}}(t^*) \leq \widehat{\mathcal{R}}_{\mathrm{val}}(-1) = \widehat{\mathcal{R}}_{\mathrm{val}}(B^*). \tag{A.29}$$

Round -1 provides a zero-correction fallback whose guarantee is confined to validation log-loss. Independent test loss and population risk require separate assessment, as do accuracy and worst-client performance: the validation log-loss ordering need not hold for any of them.

B THEORY OF THE CONFUSION CONTEXT

The confusion context in Section 4.1 summarizes the frozen base's class-conditional prediction behavior. We distinguish the population identifiability of this summary from its finite-sample estimation and examine the effects of missing classes and data reuse.

B.1 PERMUTATION-STYLE CONCEPT-SHIFT MODEL

Consider an idealized population model with latent canonical class $T \in \{1, \dots, C\}$ and client-specific bijection $\sigma_i : \{1, \dots, C\} \to \{1, \dots, C\}$ such that $Y = \sigma_i(T)$ on client i. Assume every latent class has positive support on every client: $\Pr(T = t \mid J = i) > 0$ for all i, t. Let $\widehat{Y}_B$ be the hard prediction of the fixed base, with a fixed tie-breaking rule. In a common output coordinate system, assume the shared prediction channel

$$Q[t, b] = \Pr(\widehat{Y}_B = b \mid T = t, J = i) \tag{B.1}$$

does not depend on i. Here Q is a channel matrix, distinct from the per-example probability vector Q_B. Define the population class-conditional confusion matrix

$$A_i[a, b] = \Pr(\widehat{Y}_B = b \mid Y = a, J = i). \tag{B.2}$$

The shared-channel assumption permits different latent class proportions, but excludes shifts that alter the base prediction channel after conditioning on the latent class.

B.2 IDENTIFIABILITY AND REPEATED-ROW AMBIGUITY

Theorem 2 (Identifiability of permutation-style concept shift)**.** *Under the assumptions in Appendix B.1,*

$$A_i[a, b] = Q[\sigma_i^{-1}(a), b], \qquad A_i = \Pi_i Q, \tag{B.3}$$

where the row permutation matrix Π_i satisfies $(\Pi_i Q)[a, :] = Q[\sigma_i^{-1}(a), :]$. If the rows of Q are pairwise distinct, then $\sigma_i \mapsto A_i$ is injective for fixed Q.

Proof. Since σ_i is a bijection, $Y = a$ on client i is equivalent to $T = \sigma_i^{-1}(a)$. Thus

$$\begin{aligned} A_i[a, b] &= \Pr(\widehat{Y}_B = b \mid T = \sigma_i^{-1}(a), J = i) \\ &= Q[\sigma_i^{-1}(a), b]. \end{aligned} \tag{B.4}$$

This gives Eq. (B.3). If $A_i = A_j$, each observed-label row satisfies $Q[\sigma_i^{-1}(a), :] = Q[\sigma_j^{-1}(a), :]$. Pairwise distinctness of the channel rows implies $\sigma_i^{-1}(a) = \sigma_j^{-1}(a)$ for every a, hence $\sigma_i = \sigma_j$. □

If the channel is accurate and aligned with canonical labels, $Q \approx I_C$ and $A_i \approx \Pi_i$. This identifiability result concerns the population mapping for a fixed prediction channel; finite-sample estimation is addressed below. If Q is unknown and no canonical reference is available, a common relabeling of the latent classes remains possible. Nearly identical rows can also be difficult to distinguish in finite samples.

Repeated rows.

Corollary 4 (Indistinguishable latent classes)**.** *Define $t \sim t'$ when $Q[t, :] = Q[t', :]$. Permuting latent classes within any equivalence class leaves the resulting population confusion matrix unchanged after the corresponding change in σ_i. Thus A_i cannot identify label assignments within each row-equivalence class.*

Proof. Let τ permute latent classes only within equivalence classes and set $\sigma_i' = \sigma_i \circ \tau$. Then

$$A_i'[a, :] = Q[\tau^{-1}(\sigma_i^{-1}(a)), :] = Q[\sigma_i^{-1}(a), :] = A_i[a, :]. \tag{B.5}$$

Distinct permutations related in this way therefore yield the same context. Conversely, equal confusion matrices require the corresponding inverse images of each observed label to belong to the same row-equivalence class. □

B.3 FINITE-SAMPLE ESTIMATION AND SMOOTHING

For this subsection only, we allow the context-estimation samples to vary. Fix a base, client i, true class a, and sample count $n = n_{i,a} > 0$. Assume the predicted labels $V_1, \dots, V_n$ used to estimate this row are independent draws from one fixed categorical distribution $p = p_{i,a} \in \Delta^{C-1}$. For a smoothing coefficient $\alpha > 0$, define

$$\bar{p}_b = \frac{1}{n}\sum_{j=1}^{n} \mathbf{1}\{V_j = b\}, \qquad \widehat{p}^{(\alpha)} = \frac{n\bar{p} + \alpha\mathbf{1}}{n + C\alpha} = \lambda\bar{p} + (1-\lambda)u, \tag{B.6}$$

where $u = \mathbf{1}/C$ and $\lambda = n/(n + C\alpha)$. Stacking these row estimates gives $\widehat{A}_i$, denoted by P_i in Eq. (4) when $\alpha = 1$; the implemented context is its row-major vectorization. The population matrix A_i in Appendix B.1 and the finite-data context are therefore distinct objects.

Lemma 2 (Error of a smoothed confusion row)**.** *Under this sampling model,*

$$\mathbb{E}\left\|\widehat{p}^{(\alpha)} - p\right\|_1 \le \frac{n}{n + C\alpha}\sqrt{\frac{C}{n}} + \frac{2C\alpha}{n + C\alpha}. \tag{B.7}$$

For any $\delta \in (0,1)$, with probability at least $1-\delta$,

$$\left\|\widehat{p}^{(\alpha)} - p\right\|_1 \le \frac{n}{n + C\alpha}\left[\sqrt{\frac{C}{n}} + \sqrt{\frac{2\log(1/\delta)}{n}}\right] + \frac{2C\alpha}{n + C\alpha}. \tag{B.8}$$

The right-hand side may be capped at 2.

Proof. Cauchy–Schwarz and Jensen's inequality give

$$\mathbb{E}\|\bar{p} - p\|_1 \le \sqrt{C\,\mathbb{E}\|\bar{p} - p\|_2^2}. \tag{B.9}$$

The empirical frequencies are unbiased and, under the assumed independent categorical sampling,

$$\mathbb{E}\|\bar{p} - p\|_2^2 = \sum_b \mathrm{Var}(\bar{p}_b) = \frac{1}{n}\sum_b p_b(1-p_b) = \frac{1 - \sum_b p_b^2}{n} \le \frac{1}{n}. \tag{B.10}$$

It follows that $\mathbb{E}\|\bar{p} - p\|_1 \le \sqrt{C/n}$. Changing one predicted label changes $\bar{p}$ in at most two coordinates by $1/n$ each. By the reverse triangle inequality, the function $f(V_1, \dots, V_n) = \|\bar{p} - p\|_1$ therefore has bounded differences at most $2/n$ in each argument. The bounded-differences inequality yields

$$\Pr(f - \mathbb{E}f \ge t) \le \exp\left(-\frac{2t^2}{n(2/n)^2}\right) = \exp(-nt^2/2). \tag{B.11}$$

Taking $t = \sqrt{2\log(1/\delta)/n}$ bounds $\|\bar{p} - p\|_1$ by the bracketed term in Eq. (B.8). Finally,

$$\begin{aligned}\|\widehat{p}^{(\alpha)} - p\|_1 &\le \lambda\|\bar{p} - p\|_1 + (1-\lambda)\|u - p\|_1 \\ &\le \lambda\|\bar{p} - p\|_1 + 2(1-\lambda).\end{aligned} \tag{B.12}$$

Taking expectations in the same inequality gives Eq. (B.7). The cap follows because the L_1 distance between two probability distributions is at most 2. □

Corollary 5 (Simultaneous bound for observed rows)**.** *Suppose each observed row of N clients and C classes satisfies the sampling conditions of Lemma 2. With probability at least $1-\delta$, every row with $n_{i,a} > 0$ satisfies*

$$\begin{aligned}\left\|\widehat{p}_{i,a}^{(\alpha)} - p_{i,a}\right\|_1 \le \frac{n_{i,a}}{n_{i,a} + C\alpha}\left[\sqrt{\frac{C}{n_{i,a}}} + \sqrt{\frac{2\log(NC/\delta)}{n_{i,a}}}\right] \\ + \frac{2C\alpha}{n_{i,a} + C\alpha}.\end{aligned} \tag{B.13}$$

Proof. Apply Lemma 2 with failure probability $\delta/(NC)$ to each of at most NC observed rows and take a union bound. Independence between different rows is not required. When the row counts are random, the conclusion holds conditional on counts for which the stated rowwise sampling conditions hold. □

B.4 MISSING CLASSES AND DATA-DEPENDENT CONTEXTS

For fixed C and α, Eq. (B.8) implies $\|\widehat{p}_{i,a}^{(\alpha)} - p_{i,a}\|_1 \to 0$ in probability as $n_{i,a} \to \infty$ under its sampling assumptions. The bound separates sampling error, scaling with $\sqrt{C/n_{i,a}}$, from a smoothing contribution controlled by $C\alpha/(n_{i,a} + C\alpha)$. For $n_{i,a} = 0$, the smoothing rule instead returns u. The uniform row encodes the absence of class-specific observations. If the class itself has zero population probability on that client, the corresponding conditional row is not defined by the population law.

A full confusion context has C^2 entries, so its storage cost and the router input dimension grow quadratically with the number of classes. Per-class sample support may also decrease as the label space grows.

Reusing training data. The estimator model in Appendix B.3 requires independent row samples from a fixed prediction channel. In FedCoREx, the context is constructed from the local training split, which may also have been used to fit the selected base. Conditioning on the fitted base does not generally make those same training examples independent draws from that channel. Applying Lemma 2 to this data-reuse protocol would therefore require an additional dependence argument, such as appropriate stability assumptions. Using independently sampled context data would instead change the estimation protocol. The population identity in Theorem 1 remains valid because it conditions on the realized base and context.

C METHOD SPECIFICATION

This appendix expands Section 4 into exact network operations and complete context-construction, training, and inference algorithms (Appendix C.5).

C.1 FROZEN-BASE INTERFACE AND DEPLOYED STATE

Let B^* denote the upstream method's validation-selected deployment checkpoint. The standalone and corrected predictors use the same base parameters:

$$\theta_B^{\text{standalone}} = \theta_B^* = \theta_B^{\text{correction base}}. \tag{C.1}$$

Its interface returns a representation $h_B(x) \in \mathbb{R}^{d_h}$ and the logits obtained from that representation, $z_B(x) \in \mathbb{R}^C$; write $p_B(x) = \mathrm{softmax}(z_B(x))$. The feature-extraction and classification calls must compose to the same predictor used for standalone evaluation. This identity concerns the checkpoint's complete predictive state, not just its architecture.

During context construction, correction learning, and inference, all base parameters have gradients disabled and the base stays in evaluation mode; dropout and other training-only stochastic layers are inactive. Freezing also includes any running statistics and other state affecting its forward pass. The base representation and logits are detached before they enter the correction network. Local optimization still uses the prescribed data augmentation. Deterministic preprocessing is used for context construction, validation, and test evaluation (Appendix D.3).

The complete deployment is

$$\mathcal{M}^* = (\theta_B^*, \Theta_{\text{corr}}^*, \{c_i\}_{i\in\mathcal{I}}), \qquad c_i = \Gamma(D_i^{\text{tr}}; \theta_B^*). \tag{C.2}$$

Client i deploys $(B^*, \Theta_{\text{corr}}^*, c_i)$: the base and correction weights are shared across clients, while c_i is fixed and stored locally. The contexts and correction are constructed and learned for the selected base; changing that base requires rebuilding the contexts and retraining the correction.

C.2 TRAIN-ONLY CONTEXT CONSTRUCTION

For each client, count the frozen base's predictions on its deterministic training-data view:

$$M_i[a,b] = \sum_{(x,y)\in D_i^{\text{tr}}} \mathbf{1}\{y=a\}\mathbf{1}\{\arg\max_j z_{B,j}(x) = b\}, \tag{C.3}$$

$$P_i[a,b] = \frac{M_i[a,b]+1}{\sum_{j=1}^{C} M_i[a,j] + C}, \qquad c_i = \mathrm{vec}_{\text{row}}(P_i) \in \mathbb{R}^{C^2}. \tag{C.4}$$

Rows index true classes and columns index predicted classes, in the dataset's fixed class order. Argmax ties use the first class in that order. Row-major vectorization concatenates the rows without transposition. Laplace smoothing has coefficient $\alpha = 1$; a class absent from the local training split therefore contributes a uniform row. Normalization preserves the separate rows for each true class.

The context is computed once using the selected base and remains fixed throughout the correction stage. It uses neither validation/test samples nor explicit client IDs, concept-group labels, or rotation-group labels. For transformed datasets, its counts use the client's effective training labels and transformed inputs, just as the local task does. The deterministic view preserves the training split's exact sample membership; it only removes stochastic training augmentation. The context lengths are 100 for CIFAR-10 and Fashion-MNIST, 3,844 for FEMNIST, and 4 for Sentiment140.

C.3 CORRECTION ARCHITECTURE

Each expert is a two-layer residual map with a bias-free output layer:

$$\Delta_k(h) = W_{2k}\,\mathrm{ReLU}(W_{1k}h + b_{1k}), \quad W_{1k} \in \mathbb{R}^{W\times d_h}, \quad b_{1k} \in \mathbb{R}^W, \quad W_{2k} \in \mathbb{R}^{C\times W}. \tag{C.5}$$

Each expert's input weights and biases are initialized uniformly in $[-d_h^{-1/2}, d_h^{-1/2}]$ using an expert-specific seed, and every output matrix is initialized to zero. Thus the initial residual is exactly zero

for every input, regardless of the initial routing probabilities. The expert functions depend only on h, with client context controlling their composition weights through the router.

Define non-affine, per-vector layer normalization by

$$\mathcal{N}(v) = \frac{v - \bar{v}\mathbf{1}}{\sqrt{m^{-1}\sum_{j=1}^{m}(v_j - \bar{v})^2 + 10^{-5}}}, \qquad \bar{v} = m^{-1}\sum_{j=1}^{m} v_j, \quad v \in \mathbb{R}^m. \tag{C.6}$$

The router computes

$$u_i(x) = \mathcal{N}\big([\mathcal{N}(h_B(x)); p_B(x); \mathcal{N}(c_i)]\big), \qquad d_u = d_h + C + C^2, \tag{C.7}$$

$$\bar{W}_{R,k} = \frac{W_{R,k}}{\max\{\|W_{R,k}\|_2, 10^{-12}\}}, \qquad q(x, c_i) = \mathrm{softmax}(\bar{W}_R u_i(x)). \tag{C.8}$$

Here $W_R \in \mathbb{R}^{K \times d_u}$ is the only trainable router tensor. Each normalization acts on a single sample's coordinates, and the probability block enters the concatenation directly. The router is a bias-free linear softmax map with unit temperature and non-affine normalizations. For initialization, a seeded Gaussian matrix of shape $d_u \times K$ is orthogonalized by a reduced QR decomposition with fixed diagonal-sign convention; its transposed orthonormal columns initialize W_R. The row normalization in Eq. (C.8) is also applied at every subsequent forward pass.

The predictive logits and probabilities are

$$z_i(x) = z_B(x) + \sum_{k=1}^{K} q_k(x, c_i)\Delta_k(h_B(x)), \qquad p_i(x) = \mathrm{softmax}(z_i(x)). \tag{C.9}$$

This dense mixture adds the experts' logit residuals to the base logits with base coefficient one, then applies the class softmax. The main configuration is $K = 4, W = 16$. Its correction parameter count is

$$P_{\mathrm{corr}} = KW(d_h + C + 1) + K(d_h + C + C^2). \tag{C.10}$$

The first term counts the expert bank and the second the router; the fixed contexts are not neural parameters.

C.4 Correction training and checkpoint selection

Write $n_i = |D_i^{\mathrm{tr}}|$ and $\Theta = \{W_R, (W_{1k}, b_{1k}, W_{2k})_{k=1}^{K}\}$. For a local mini-batch $\mathcal{B}_i$, the sole training objective is

$$\mathcal{L}_i(\Theta; \mathcal{B}_i) = -\frac{1}{|\mathcal{B}_i|}\sum_{(x,y)\in\mathcal{B}_i} \log p_i(y \mid x; \Theta). \tag{C.11}$$

In the main configuration, each round selects 10 of 50 clients without replacement according to a seeded schedule. Starting from the same global correction, each participant performs three local epochs of SGD with batch size 32, learning rate 0.01, zero momentum, and zero weight decay. The final partial mini-batch is retained. Optimization is restricted to the router and expert parameters under this cross-entropy objective.

If $\Theta_i^{(t)}$ is the returned local state, the server updates

$$\Theta^{(t)} = \sum_{i\in\mathcal{S}_t} \frac{n_i}{\sum_{j\in\mathcal{S}_t} n_j}\Theta_i^{(t)}. \tag{C.12}$$

All expert and router parameters use the same training-sample aggregation weights. The checkpoint-selection criterion below gives each client's validation loss equal weight. Each accepted round includes all scheduled client updates.

The zero-residual initialization is candidate round -1. Before learning, validation predictions from that state are checked against the standalone base. After each of the 200 correction rounds, the full candidate is evaluated on all clients' validation splits:

$$\mathcal{J}_{\mathrm{val}}(t) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|D_i^{\mathrm{val}}|}\sum_{(x,y)\in D_i^{\mathrm{val}}} -\log p_i^{(t)}(y \mid x), \tag{C.13}$$

$$t^* = \arg\min_{t\in\{-1,0,\ldots,199\}} \mathcal{J}_{\mathrm{val}}(t). \tag{C.14}$$

Exact ties retain the earlier candidate. The router and experts are saved and restored together from round t^*, with the frozen base and contexts unchanged. Validation may select round -1; including this candidate does not guarantee improved test performance.

C.5 Context, training, and inference algorithms

Algorithm 1 implements Eq. (C.4), and Algorithm 2 implements the local optimization, aggregation, and complete-state selection above. Algorithm 3 specifies the final prediction path.

In Algorithm 2, $\mathrm{Prefer}_{\mathrm{val}}$ compares the candidate with the incumbent using Eq. (11) and the fixed numerical and tie-breaking rules. The superscript t on a client's parameters denotes its current local state within round t until those parameters are returned.

Algorithm 1 Train-only client context construction

Require: Selected shared base θ_B^*; client training splits $\{D_i^{\mathrm{tr}}\}$;
class count C; smoothing $\alpha = 1$; fixed deterministic preprocessing
Ensure: Fixed local contexts $\{c_i\}$ bound to the selected base and training data
Frozen-base setup
Verify the base identity in Eq. (C.1); freeze its full predictive state
Local context construction
for each client $i \in \mathcal{I}$ **in parallel do**
  $M_i \leftarrow 0_{C\times C}$
  for each $(x, y) \in D_i^{\mathrm{tr}}$ under deterministic preprocessing **do**
    $b \leftarrow \arg\max_{a\in\mathcal{Y}} z_{B,a}(x)$ using the frozen evaluation-mode base
    $M_i[y, b] \leftarrow M_i[y, b] + 1$
  end for
  $P_i[a, b] \leftarrow (M_i[a, b] + \alpha)/(\sum_{\ell=1}^{C} M_i[a, \ell] + C\alpha)$ for all a, b
  $c_i \leftarrow \mathrm{vec}_{\mathrm{row}}(P_i)$; store locally with its construction dependencies
end for
Distributed output
return $\{c_i\}$ as distributed, non-trainable local state

At inference, each participating client combines the stored training-derived context with the selected checkpoint in a fixed forward pass, without using test labels.

D Experimental Setup

The settings below expand Section 5.1, covering data construction, model architectures, and method-specific training rules. Table 7 summarizes the datasets, client semantics, and backbones.

Table 7: Experimental settings. Each row uses 50 clients and one fixed data realization. The class count is C and the frozen representation width is d_h.

Dataset	Conditions / client identity	Dimensions		Backbone
		C	d_h	
CIFAR-10	IID, Label, Concept, FC	10	84	LeNet-5 variant
Fashion-MNIST	IID, Label, Concept, FC	10	84	LeNet-5 variant
FEMNIST	Natural writers	62	84	LeNet-5 variant
Sentiment140	Natural users	2	128	TextCNN variant

D.1 Client populations

All principal settings use a fixed realization of 50 clients. Table 8 gives the exact populations underlying the local test evaluations. Training, validation, and test membership is fixed before model training and is shared across methods in a setting. The partitions target $70/15/15$ locally; their integer-allocation rules are specified below.

Algorithm 2 Federated client-conditioned residual correction

Require: Selected base θ_B^*; fixed contexts from Algorithm 1;
local training and validation splits; K, W, T_{corr}; local epochs E;
batch size; learning rates $\{\eta_t\}$; seeded participation and data-order schedules
Ensure: Deployment $\mathcal{M}^*$ from Eq. (C.2)

Initialization

Keep the base in evaluation mode and contexts fixed; check their dependency identities
Initialize the shared experts and router; set every $W_{2k} \leftarrow 0$ to obtain $\Theta_{\text{corr}}^{(-1)}$
Verify round -1 base parity on validation data; abort if the check fails
$\Theta_{\text{corr}}^* \leftarrow \text{Copy}(\Theta_{\text{corr}}^{(-1)})$; $t^* \leftarrow -1$; evaluate $\mathcal{J}_{\text{val}}(-1)$
for $t = 0, \ldots, T_{\text{corr}} - 1$ **do**
    Obtain $\mathcal{S}_t$ from the seeded schedule; broadcast $\Theta_{\text{corr}}^{(t-1)}$

    Client-side update

    for each client $i \in \mathcal{S}_t$ **in parallel do**
        $\Theta_{\text{corr},i}^{(t)} \leftarrow \text{Copy}(\Theta_{\text{corr}}^{(t-1)})$
        for $e = 1, \ldots, E$ **do**
            for each seeded training mini-batch $\mathcal{B}_i \subseteq D_i^{\text{tr}}$ **do**
                Compute detached $h_B(x), z_B(x), p_B(x)$ for each $(x, y) \in \mathcal{B}_i$
                Compute expert outputs and routes by Eqs. (5)–(6)
                Form corrected logits and $\mathcal{L}_i$ by Eqs. (7)–(8)
                $\Theta_{\text{corr},i}^{(t)} \leftarrow \Theta_{\text{corr},i}^{(t)} - \eta_t \nabla_{\Theta_{\text{corr},i}^{(t)}} \mathcal{L}_i(\Theta_{\text{corr},i}^{(t)}; \mathcal{B}_i)$
            end for
        end for
        Return $\Theta_{\text{corr},i}^{(t)}$ and n_i to the server
    end for

    Server aggregation and validation

    Require all scheduled updates to succeed before accepting the round
    Aggregate $\Theta_{\text{corr}}^{(t)}$ using Eq. (9)
    Evaluate $\mathcal{J}_{\text{val}}(t)$ on all clients using their stored contexts
    if $\text{Prefer}_{\text{val}}(t, t^*)$ **then**
        $\Theta_{\text{corr}}^* \leftarrow \text{Copy}(\Theta_{\text{corr}}^{(t)})$; $t^* \leftarrow t$ ▷ Save router and experts together
    end if
end for

Deployment output

Verify that the base and contexts are unchanged
return $(\theta_B^*, \Theta_{\text{corr}}^*, \{c_i\}_{i \in \mathcal{I}})$

Algorithm 3 Deployed FedCoREx inference on client i

Require: Frozen selected base B^*; selected correction Θ_{corr}^*;
stored training-derived context c_i; input x
Ensure: Predictive distribution $p_i(x)$ and class prediction $\widehat{y}_i(x)$

Frozen-base state and routing

Apply the client's deterministic evaluation preprocessing to x
Compute $h \leftarrow h_B(x)$, $z_B \leftarrow z_B(x)$, $p_B \leftarrow \text{softmax}(z_B)$
Compute $u_i(x)$ and $q(x, c_i)$ using Eq. (C.8)

Residual composition and prediction

for $k = 1, \ldots, K$ **do**
    $\Delta_k \leftarrow W_{2k} \,\text{ReLU}(W_{1k} h + b_{1k})$
end for
$p_i(x) \leftarrow \text{softmax}(z_B + \sum_k q_k(x, c_i) \Delta_k)$
return $p_i(x)$ and $\arg\max_y p_{i,y}(x)$

Table 8: Sample counts in the canonical 50-client realizations. Per-client summaries count all three local splits. CIFAR-10 and Fashion-MNIST counts apply to each of IID, Label, Concept, and FC.

Dataset	Pooled local split size			Samples per client		
	Train	Validation	Test	Min.	Mean	Max.
CIFAR-10	35,000	7,500	7,500	1,000	1,000	1,000
Fashion-MNIST	42,000	9,000	9,000	1,200	1,200	1,200
FEMNIST	7,460	1,605	1,591	166	213.12	304
Sentiment140	4,543	956	1,029	100	130.56	189

Controlled image populations. We partition the 50,000 official training images of CIFAR-10 and the 60,000 official training images of Fashion-MNIST across clients (Krizhevsky, 2009; Xiao et al., 2017). The original official test sets are not the local test sets reported here. In both IID and Label, each client has 1,000 examples for CIFAR-10 and 1,200 examples for Fashion-MNIST; Concept and FC inherit these per-client sample counts from IID. For each client, split sizes use largest-remainder rounding of $0.70n_i^{\text{all}}, 0.15n_i^{\text{all}}, 0.15n_i^{\text{all}}$. Class-stratified integer allocation respects both these split totals and the client's class totals; within-class sample assignment is seeded. This gives exactly 700/150/150 examples per CIFAR-10 client and 840/180/180 per Fashion-MNIST client.

Natural writers. The FEMNIST source preserves 3,597 writer identities and 62 character classes (Caldas et al., 2019). We pool each writer's source training and test examples, then repartition locally. There are 1,831 eligible writers with 166–307 total examples. From the stable ordered candidate list, a fixed PCG64 draw selects 50 writers without replacement. The selected population has 10,656 examples and covers all 62 classes in aggregate. Its local partitions use the same stratified, largest-remainder rule as the image populations above. Writer identity is preserved: no example is reassigned to a different natural client.

Natural text users. Sentiment140 uses only the 1.6-million-example source-training corpus (Go et al., 2009); the separately annotated source-test corpus is excluded. Negative and positive labels are mapped to the two class indices. Among source users with 100–199 training tweets, 142 meet the eligibility rule. A fixed, label- and text-independent hash ranking of user identity and data seed selects the first 50; neither class balance nor eventual performance is used to choose them. Within each selected user, a deterministic ranking of source-row identifiers allocates $\lfloor 0.70n_i^{\text{all}} \rfloor$ rows to training, $\lfloor 0.15n_i^{\text{all}} \rfloor$ to validation, and the remainder to test. Unlike the image splits, this rule is not label-stratified. Source-row identities are disjoint across splits, but texts are not deduplicated: the fixed population contains 12 exact-text groups spanning local training and test splits, with no cross-client exact-duplicate groups.

The natural-dataset evaluations concern these selected writer and user populations. No clients are held out as an unseen-client test population.

D.2 Controlled shift definitions

IID and Label. For IID, integer client-by-class counts approximate the source class proportions while satisfying fixed client and global class totals. For Label, each client's class preference vector is drawn from the symmetric distribution $\text{Dirichlet}(a, \ldots, a)$ over C classes, with each concentration parameter $a = 0.3$. A constrained transport allocation fits these preferences subject to the same fixed client totals and exact source class totals, followed by integer allocation and seeded within-class sampling. The concentration controls label preference under fixed client sample counts. Label is allocated separately from the same source pool and per-client budgets.

Concept. Starting from the IID ownership and split assignments, we assign clients to four near-balanced groups (13, 13, 12, and 12 clients). Each group applies its designated class permutation to every local split. For the canonical construction, the four permutations are the cycles

$$\pi_0 = (0\ 2), \qquad \pi_1 = (1\ 4), \qquad \pi_2 = (3\ 5\ 7), \qquad \pi_3 = (6\ 8\ 9). \tag{D.1}$$

A cycle maps each listed label to its successor and the last back to the first; labels not listed are unchanged. Thus each client observes $y' = \pi_g(y)$ for its group g. The same effective label semantics govern training, validation, and local test evaluation. The original images, client memberships, and split memberships remain fixed.

Feature+Concept (FC). The four concept permutations are crossed with rotations of $0°$, $90°$, $180°$, and $270°$. The 16 joint groups are assigned near-uniformly across the 50 clients, with three or four clients per joint group. For client group (g, r), the example becomes $(\mathrm{Rot}_r(x), \pi_g(y))$ in every split. Rotation is a fixed property of the client's task, applied before the split-specific preprocessing; it precedes any training augmentation. Both Concept and FC reuse the IID sample ownership, local splits, and within-split sample identities. Concept and rotation group labels are used only to construct or describe the benchmark, never as FedCoREx inputs.

Robustness and auxiliary conditions. The CIFAR-10 robustness study compares three data realizations with the training randomization held fixed. Construction robustness replaces the canonical cycles by either $(0\ 1), (2\ 3), (4\ 5\ 6), (7\ 8\ 9)$ or $(0\ 5), (2\ 8), (1\ 3\ 9), (4\ 6\ 7)$, with data realization and training randomization held fixed. Participation robustness uses 10%, 20%, and 50% participation on FC; these robustness axes are varied one at a time. Quantity-only and rotation-only conditions are included only in the auxiliary heterogeneity analysis, as matched base–FedCoREx test results are unavailable.

D.3 Backbones and preprocessing

Images. All visual runs use a LeNet-5 variant (LeCun et al., 1998) on 32×32 inputs. The convolutional blocks have 6, 16, and 120 output channels with 5×5 kernels; the first convolution uses padding 2 and the other two use no padding. ReLU follows each convolution, and 2×2 average pooling with stride 2 follows the first two. The resulting $120 \times 2 \times 2$ tensor is flattened, mapped to 84 coordinates, passed through ReLU, and classified by a linear $84 \to C$ layer. The 84-dimensional post-ReLU vector is h_B. The backbone uses neither batch normalization nor dropout. Grayscale images are resized from 28×28 to 32×32 and retain one channel; CIFAR-10 retains three channels.

Pixels are converted to $[0, 1]$ and standardized using the following channel statistics:

$$\begin{array}{lll} \text{CIFAR-10:} & \mu = (0.4914, 0.4822, 0.4465), & \sigma = (0.2470, 0.2435, 0.2616), \\ \text{Fashion-MNIST:} & \mu = 0.2860, & \sigma = 0.3530, \\ \text{FEMNIST:} & \mu = 0.5, & \sigma = 0.5. \end{array} \tag{D.2}$$

CIFAR-10 training additionally uses random 32×32 crops after padding by four pixels and random horizontal flips. This applies to both upstream and correction local training. Context construction, validation, and test use deterministic conversion and normalization without these augmentations. The grayscale datasets use deterministic resizing and normalization in all splits.

Text. Preprocessing decodes HTML entities, replaces URLs and mentions by separate reserved tokens, case-folds text, and collapses whitespace. Hashtags, punctuation, emoticons, and elongated words are retained. Whitespace tokenization is followed by right truncation or padding to length 32. The vocabulary is fitted only on the union of selected clients' training splits, with a minimum frequency of 2 and a maximum size of 5,000; tokens are ordered by decreasing frequency, with ties broken lexicographically. The realized vocabulary contains 3,300 tokens, including padding, unknown, URL, and user placeholders. Validation and test tokens do not expand it.

The TextCNN variant (Kim, 2014) learns 64-dimensional embeddings from scratch and uses no dropout. Parallel one-dimensional convolutions have widths 3, 4, and 5 with 64 filters each, followed by ReLU and temporal max pooling. Pooling excludes windows extending beyond the nonpadding length; for a text shorter than a kernel, the first partially padded window is retained. The three pooled vectors are concatenated to 192 dimensions, projected to 128, layer-normalized with trainable affine parameters, and passed through ReLU to obtain h_B. A linear $128 \to 2$ classifier produces the logits. The backbone's affine layer normalization is distinct from the router's non-affine normalizations.

Table 9: Model dimensions and inference parameter counts for the main $K = 4, W = 16$ configuration. Base counts refer to a single shared classifier. The counts exclude stored client contexts.

Dataset	d_h	C	d_u	Base	Correction	Total
CIFAR-10	84	10	194	92,246	6,856	99,102
Fashion-MNIST	84	10	194	91,946	6,856	98,802
FEMNIST	84	62	3,990	96,366	25,368	121,734
Sentiment140	128	2	134	285,762	8,920	294,682

D.4 Training budgets and hyperparameters

Shared-model and end-to-end MoE baselines train for 500 communication rounds. The main participation rate is 20%, with three local epochs and batch size 32 for every selected client. Correction stages update only the correction parameters for a separate 200-round budget, using the same local epochs and batch size. Table 10 records the local optimizers; method-specific server rules are described in Appendix D.5. Methods retain their own objectives, optimizer states, and communication payloads under these training budgets.

Table 10: Local SGD settings. Momentum and weight decay describe client optimization; server histories and control variates are specified separately. The round index t starts at zero. SCAFFOLD, FedDyn, and FedSMU are evaluated on CIFAR-10.

Method / component	Rounds	Initial LR	Momentum	Weight decay	Local LR schedule
FedAvg	500	0.01	0	0	Constant
FedProx	500	0.01	0	0	Constant
FedGMT	500	0.01	0.9	10^{-5}	$0.01(0.998)^t$
Vanilla-MoE	500	0.01	0	0	Constant
DDOME experts	500	0.01	0.9	0	Constant
DDOME router	500	0.001	0	0	Constant
SCAFFOLD	500	0.01	0	0	Constant
FedDyn	500	0.01	0	0	Constant
FedSMU	500	0.1	0	10^{-3}	Constant
FedCoREx / controls	200	0.01	0	0	Constant

DDOME also depends on a pretrained FedGMT encoder and uses an anchor-specific participation schedule. FedSMU's server step size, unlike its local LR, decays each round. Correction settings are unchanged across upstream training methods.

The horizon control continues the FedGMT state after 500 communication rounds for 200 additional rounds, preserving its training state and learning-rate schedule. Its additional participation and compatible local-task random streams are paired with the correction stage. It tests whether 200 additional upstream rounds account for the correction gains; computational and communication costs depend on each stage's trainable state. The fixed-base correction starts from the upstream validation-selected checkpoint, as required by the deployment comparison.

The expert-granularity extension uses $(K, W) = (1, 72), (2, 34), (4, 16), (8, 7)$ to keep correction sizes approximately matched. For $K = 1$, the route is identically one and the router has no trainable parameters. The main $K = 4$ configuration is reused and remains the primary model choice. Client sampling and local random streams are defined independently of worker completion order.

D.5 Baseline implementations

Shared-global methods. FedAvg uses local cross-entropy and training-sample-weighted model aggregation (McMahan et al., 2017). FedProx adds $\frac{\mu}{2}\|\theta - \theta^{(t)}\|_2^2$ with $\mu = 0.1$ to that local objective (Li et al., 2020). FedGMT retains trajectory distillation with weight 1 and temperature 3, an exponential-moving-average teacher with coefficient 0.95, and persistent per-client dual accumulators with coefficient 10 (Li et al., 2025). Its server combines the sample-weighted local delta with

the population-mean dual state over all 50 clients. The deployed predictor is its validation-selected shared global model. These auxiliary states are specific to upstream optimization.

Vanilla-MoE. This internal architectural baseline trains a shared backbone, an auxiliary anchor classifier, four shared linear specialist heads, and a feature-only router end to end. Predictions are a soft probability mixture of the specialist predictions. Its task loss is supplemented by anchor-classifier cross-entropy with weight 1 and a load-balancing term with weight 0.01. All trainable components use the same sample-weighted aggregation, with no private client head or context input.

DDOME. We reimplement DDOME from its paper, retaining full-model experts, anchor-based specialization, and joint gating–expert optimization (Farhat et al., 2025). The adaptation uses five full experts and a shared $d_h \to 64 \to 5$ softmax router; the hidden width 64 is an implementation choice. A frozen FedGMT encoder from the end of its 500-round training supplies routing features as DDOME's upstream dependency.

Five anchors are chosen by deterministic farthest-first selection on client-mean training embeddings, without labels or validation/test data. All five anchors and five nonanchor clients participate each round. An anchor trains its assigned expert using task cross-entropy and its router using the assigned expert index as the routing target. A nonanchor chooses its top two experts by summed full-pool router probabilities on its training embeddings, and jointly optimizes those experts and the router. The selected training weights are not renormalized over the selected subset. The router is aggregated over all participants with training-sample weights; each expert is aggregated only over its actual trainers. For participating-client inference, the client's training-derived top-two set is frozen and each sample uses the router's hard argmax within that set. No validation labels select routes. The adaptation retains the anchor-specific participation schedule and pretrained encoder dependency for evaluation on participating clients; unseen-client evaluation is outside this protocol.

Additional shared-global baselines. SCAFFOLD and FedDyn follow the implementations distributed with the FedDyn reference code, adapted to the common models, local epochs, and partial participation (Karimireddy et al., 2020; Acar et al., 2021). SCAFFOLD uses Option II with the actual local mini-batch step count, persistent client controls, a server learning rate of 1, uniform selected-client model-delta aggregation, and a population-scaled server-control update. FedDyn uses dynamic regularization coefficient 0.01, scaled inversely by the normalized client training-sample weight. Its deployment object is the cloud model formed from the uniform selected-client mean and population-mean history.

FedSMU follows its reference implementation's symbolic-update mechanism (Lu et al., 2025). Client momentum coefficients are $\beta_1 = \beta_2 = 0.9$; the server averages selected-client symbols uniformly and applies the split-Lion update with initial $\gamma_1 = 0.015$, $\gamma_2 = 0.01$, and round multiplier 0.9988 on γ_1. The stated one-bit update representation defines its logical payload accounting. All three extensions retain their training-only state for upstream optimization; FedCoREx subsequently freezes their selected shared predictors and uses the same correction configuration as for the original bases.

Direct, uniform-mixture, and no-context residual models serve as internal controls: they vary the correction parameterization while keeping the training and checkpoint-selection protocol fixed.

E Evaluation and Statistical Protocol

This appendix defines the checkpoint pairing, metrics, and uncertainty summaries used throughout Section 5.

E.1 Checkpoint-aligned evaluation and model selection

The paired comparison is

$$B^* \quad \longrightarrow \quad B^* + C^*(B^*), \tag{E.1}$$

$$B^*_{\text{standalone}} = B^*_{\text{context}} = B^*_{\text{correction}} = B^*_{\text{test}}. \tag{E.2}$$

Equality is checked on the complete predictive state. The upstream model is selected by minimum client-macro validation NLL over its evaluated training candidates. The deployed state is the method's shared predictor: for example, FedGMT's global model or FedDyn's cloud model.

Contexts are constructed from that selected base, and the zero-residual candidate must reproduce its validation predictions before correction training begins. Correction candidates are selected by Eq. (C.14); no accuracy-based tie-breaker or per-client checkpoint selection is applied. The final correction state contains the experts and router from one selected round, with the matching fixed contexts and base. The test evaluator restores this full predictive state and uses client-local test data under the same task semantics.

Model configurations, deployment checkpoints, and comparison sets are fixed before each formal test evaluation. Repeated model selection uses validation data; test labels remain excluded from context construction, correction optimization, and checkpoint choice.

E.2 Metric definitions

Let client i have m_i test examples, with accuracy and mean negative log-likelihood

$$a_i = \frac{1}{m_i}\sum_{j=1}^{m_i} \mathbf{1}\{\widehat{y}_{ij} = y_{ij}\}, \qquad \ell_i = -\frac{1}{m_i}\sum_{j=1}^{m_i} \log p_i(y_{ij} \mid x_{ij}). \tag{E.3}$$

All evaluated clients have nonempty local test sets. Client-macro accuracy, sample-weighted accuracy, and client-macro NLL are

$$A_{\text{macro}} = \frac{1}{N}\sum_i a_i, \qquad A_{\text{weighted}} = \frac{\sum_i m_i a_i}{\sum_i m_i}, \qquad L_{\text{macro}} = \frac{1}{N}\sum_i \ell_i. \tag{E.4}$$

The primary accuracy metric gives each client equal weight. The weighted metric differs on the natural datasets because their client sample counts are unequal.

Let $a_{(1)} \le \cdots \le a_{(N)}$ be a method's sorted client accuracies and $b = \lceil 0.1N \rceil$. Bottom-10% accuracy is

$$A_{\text{bottom}} = \frac{1}{b}\sum_{j=1}^{b} a_{(j)}, \qquad b = 5 \text{ for } N = 50. \tag{E.5}$$

Sorting is performed independently for each method and repetition, so each lower tail contains that method's own five lowest accuracies.

For macro-F1, pool the confusion counts over all client test examples. If $\mathrm{TP}_c, \mathrm{FP}_c, \mathrm{FN}_c$ are those pooled counts for class c, then

$$F_{1,\text{macro}} = \frac{1}{C}\sum_{c=1}^{C} \frac{2\mathrm{TP}_c}{2\mathrm{TP}_c + \mathrm{FP}_c + \mathrm{FN}_c}, \tag{E.6}$$

with a zero contribution when the denominator is zero. This sample-pooled macro-F1 averages over all C task classes.

Accuracies and F1 are multiplied by 100 for reporting; their paired differences are percentage points. NLL uses natural logarithms and is reported in its unscaled loss units. Performance tables define $\Delta M = M_{\text{corrected}} - M_{\text{base}}$, so a negative ΔL_{macro} is an improvement. The heterogeneity analysis instead reports NLL improvement as $L_{\text{base}} - L_{\text{corrected}}$, so positive values indicate improvement.

E.3 Pairing and uncertainty summaries

Main comparisons use three training repetitions on a fixed data realization. Data randomization determines client allocation, local splits, and shift assignment; training randomization determines initialization, participation, and local-task order. Compatible streams are paired across methods, with method-intrinsic randomness handled separately and DDOME using its anchor-specific participation schedule. Table 11 summarizes the replication design.

Table 11: Replication design. Data and training randomization are separate factors; robustness axes are varied one at a time.

Evidence group	Data realizations	Training repetitions per setting
Main comparisons (four datasets)	1	3
CIFAR-10 confirmation	1	5
CIFAR-10 data-realization robustness	3	1 (fixed)
CIFAR-10 concept-construction robustness	1 per mapping	1 (fixed)
CIFAR-10 participation / expert count	1	3
CIFAR-10 additional baselines	1	3

A base–correction pair shares the dataset, construction, randomization, participation rate, and exact deployed base. For repetition-level metrics M_s, $\bar{M} \pm s_M$ reports the arithmetic mean and sample standard deviation with denominator $n-1$, where n is the number of training repetitions. Paired effects use $d_s = M_{s,\text{corrected}} - M_{s,\text{base}}$ before computing their mean and standard deviation. Training repetitions are the statistical units for these uncertainty summaries.

The five-repetition confirmation additionally reports the descriptive paired t interval

$$\bar{d} \pm t_{4,0.975}\frac{s_d}{\sqrt{5}}, \qquad t_{4,0.975} \simeq 2.776. \tag{E.7}$$

The heterogeneity correlations use the four condition-level means after averaging over training repetitions.

F Full Performance and Base Portability

The full results below use the checkpoint-aligned comparison defined in Appendix E.1 and expand Section 5. Unless stated otherwise, summaries use the three main algorithm seeds in Table 11, with the dataset realization and participation rate fixed. Uncertainty is the sample standard deviation across training repetitions. Accuracy, Bottom-10% accuracy, and pooled macro-F1 are reported in percent; their differences are percentage points (pp). NLL is in natural-log units. In performance tables, Δ means corrected minus reference, so a negative ΔNLL is an improvement.

F.1 Full absolute performance

Tables 12–14 give the full set of primary and secondary metrics. Light blue shading identifies FedCoREx in these absolute comparisons. The visual conditions cover IID, Label, Concept, and Feature+Concept (FC). SCAFFOLD, FedDyn, and FedSMU are evaluated on CIFAR-10 under the same checkpoint-aligned protocol. DDOME denotes our framework adaptation, whose assumptions and differences from the original setting are stated in Appendix D.5.

On CIFAR-10, FedCoREx improves the FedGMT base primarily on Concept and FC, while remaining close to it on IID and Label. These absolute comparisons evaluate complete predictors, including standalone MoEs; the paired results below isolate the addition of correction to a given deployed predictor.

Fashion-MNIST exhibits the same concentration of gains on the structured concept conditions, with larger accuracy changes than CIFAR-10. Table 14 additionally reports sample-weighted accuracy to account for unequal natural-client sizes.

F.2 Same-base paired correction effects

Table 15 computes every difference within a matched seed before averaging.

The large Concept/FC accuracy gains are accompanied by lower NLL and better tail-client accuracy. CIFAR-10 Label has a mean accuracy change of -0.107 pp. On Sentiment140, accuracy improves by $+0.255$ pp, but Bottom-10% accuracy changes by -0.672 pp and mean test NLL increases slightly.

Table 12: CIFAR-10: complete local-test metrics (three-seed mean ± sample SD). Accuracy, Bottom-10%, and F1 are percentages; NLL uses natural-log units. FC denotes Feature+Concept.

Method	Accuracy	Bottom-10%	NLL	Pooled macro-F1
IID				
FedAvg	66.00 ± 0.37	59.96 ± 1.61	0.9860 ± 0.0097	65.83 ± 0.31
FedProx	65.90 ± 0.46	60.00 ± 1.67	0.9941 ± 0.0110	65.74 ± 0.45
FedGMT	74.86 ± 0.35	68.98 ± 0.56	0.7444 ± 0.0034	74.83 ± 0.37
Vanilla-MoE	67.56 ± 0.13	61.91 ± 0.68	0.9548 ± 0.0048	67.45 ± 0.16
DDOME (adapted)	50.16 ± 2.96	44.27 ± 3.84	1.9151 ± 0.0899	47.00 ± 3.68
SCAFFOLD	66.67 ± 0.45	60.67 ± 0.83	0.9595 ± 0.0099	66.44 ± 0.46
FedDyn	67.87 ± 0.51	61.51 ± 2.16	0.9943 ± 0.0148	67.91 ± 0.49
FedSMU	70.02 ± 0.77	64.04 ± 0.68	0.8750 ± 0.0173	70.00 ± 0.77
FedCoREx (FedGMT base)	74.88 ± 0.34	69.02 ± 0.60	0.7434 ± 0.0044	74.86 ± 0.34
Label				
FedAvg	59.71 ± 0.49	47.47 ± 2.20	1.1328 ± 0.0152	59.59 ± 0.76
FedProx	59.46 ± 0.32	47.20 ± 1.74	1.1378 ± 0.0155	59.36 ± 0.42
FedGMT	73.56 ± 0.80	62.67 ± 1.80	0.7957 ± 0.0133	73.58 ± 0.81
Vanilla-MoE	63.16 ± 0.85	50.71 ± 0.73	1.0489 ± 0.0189	63.03 ± 0.96
DDOME (adapted)	53.56 ± 3.31	30.67 ± 8.63	2.3145 ± 0.1040	52.14 ± 4.00
SCAFFOLD	58.89 ± 0.70	46.04 ± 0.87	1.1429 ± 0.0222	58.56 ± 0.92
FedDyn	64.28 ± 1.11	52.13 ± 1.83	1.0638 ± 0.0157	64.30 ± 0.80
FedSMU	63.20 ± 0.60	46.44 ± 1.47	1.0350 ± 0.0054	62.83 ± 0.32
FedCoREx (FedGMT base)	73.45 ± 0.80	62.80 ± 1.19	0.7708 ± 0.0135	73.42 ± 0.70
Concept				
FedAvg	46.12 ± 0.63	37.11 ± 1.00	1.6137 ± 0.0146	45.91 ± 0.45
FedProx	45.73 ± 0.70	36.53 ± 0.80	1.6189 ± 0.0134	45.54 ± 0.49
FedGMT	56.47 ± 0.91	45.29 ± 1.56	1.2064 ± 0.0143	56.32 ± 0.91
Vanilla-MoE	48.38 ± 0.65	39.87 ± 2.43	1.5574 ± 0.0248	48.15 ± 0.73
DDOME (adapted)	17.56 ± 2.84	11.47 ± 2.37	2.2609 ± 0.0272	12.41 ± 1.79
SCAFFOLD	45.75 ± 0.71	36.00 ± 1.60	1.5560 ± 0.0157	45.36 ± 0.83
FedDyn	49.13 ± 0.33	40.36 ± 2.37	1.5809 ± 0.0221	48.96 ± 0.24
FedSMU	49.03 ± 1.11	38.98 ± 2.40	1.7361 ± 0.0376	48.78 ± 1.08
FedCoREx (FedGMT base)	67.92 ± 3.34	58.67 ± 2.63	0.9136 ± 0.0560	67.93 ± 3.26
FC				
FedAvg	35.89 ± 0.77	25.78 ± 0.68	1.8700 ± 0.0150	35.27 ± 0.95
FedProx	35.53 ± 0.15	25.78 ± 0.47	1.8781 ± 0.0170	34.96 ± 0.41
FedGMT	45.79 ± 0.86	34.80 ± 0.80	1.5443 ± 0.0362	45.27 ± 0.93
Vanilla-MoE	37.41 ± 0.45	27.51 ± 0.15	1.8315 ± 0.0052	36.92 ± 0.47
DDOME (adapted)	13.17 ± 2.69	7.82 ± 0.78	2.2822 ± 0.0189	7.99 ± 2.88
SCAFFOLD	35.45 ± 0.90	24.58 ± 1.35	1.8282 ± 0.0174	34.84 ± 0.91
FedDyn	39.84 ± 0.31	29.38 ± 1.58	1.7877 ± 0.0308	39.62 ± 0.07
FedSMU	38.08 ± 1.07	28.98 ± 0.80	1.8485 ± 0.0176	37.77 ± 1.39
FedCoREx (FedGMT base)	54.94 ± 1.63	46.40 ± 2.72	1.2761 ± 0.0454	54.55 ± 1.39

Note. All methods use the checkpoint-aligned evaluation protocol.

Table 16 reports weighted comparisons for the natural datasets; the visual test partitions have equal client sizes.

F.3 Five-seed confirmation

The CIFAR-10 confirmation adds two algorithm seeds to the original three, retaining the same four conditions and fixed data realization. Table 17 reports the paired accuracy mean, sample SD, and descriptive 95% Student-t interval defined in Eq. (E.7), together with the secondary paired metrics. The mean accuracy gains on Concept and FC are 11.752 and 9.267 pp, with intervals $[8.812, 14.692]$ and $[7.952, 10.581]$ pp. IID and Label remain close to zero, with intervals spanning zero.

The enlarged set includes the three main-table repetitions. Its intervals summarize training variability conditional on the fixed client population and data realization.

Table 13: Fashion-MNIST: complete local-test metrics (three-seed mean ± sample SD). Accuracy, Bottom-10%, and F1 are percentages; NLL uses natural-log units. FC denotes Feature+Concept.

Method	Accuracy	Bottom-10%	NLL	Pooled macro-F1
IID				
FedAvg	88.41 ± 0.14	84.70 ± 0.17	0.3347 ± 0.0003	88.32 ± 0.10
FedProx	88.43 ± 0.08	84.63 ± 0.57	0.3333 ± 0.0013	88.35 ± 0.02
FedGMT	91.79 ± 0.26	87.85 ± 0.34	0.2668 ± 0.0005	91.76 ± 0.26
Vanilla-MoE	88.55 ± 0.39	84.81 ± 0.50	0.3308 ± 0.0007	88.52 ± 0.38
DDOME (adapted)	85.53 ± 0.98	81.48 ± 0.39	1.0416 ± 0.0590	85.63 ± 0.81
FedCoREx (FedGMT base)	91.84 ± 0.17	87.74 ± 0.42	0.2554 ± 0.0043	91.80 ± 0.18
Label				
FedAvg	87.29 ± 0.12	74.26 ± 0.83	0.3558 ± 0.0023	87.26 ± 0.12
FedProx	87.16 ± 0.26	74.48 ± 1.36	0.3573 ± 0.0038	87.13 ± 0.29
FedGMT	91.17 ± 0.05	81.67 ± 0.29	0.2776 ± 0.0020	91.17 ± 0.05
Vanilla-MoE	87.50 ± 0.21	75.37 ± 1.54	0.3535 ± 0.0064	87.46 ± 0.12
DDOME (adapted)	79.28 ± 7.93	56.67 ± 15.08	1.0665 ± 0.0638	78.99 ± 7.97
FedCoREx (FedGMT base)	92.02 ± 0.21	84.33 ± 1.66	0.2543 ± 0.0047	92.03 ± 0.21
Concept				
FedAvg	64.77 ± 0.07	57.11 ± 0.48	1.1835 ± 0.0114	64.68 ± 0.09
FedProx	64.25 ± 0.72	56.74 ± 1.30	1.1901 ± 0.0153	64.17 ± 0.79
FedGMT	68.67 ± 0.15	61.56 ± 0.40	0.8807 ± 0.0052	68.61 ± 0.15
Vanilla-MoE	64.20 ± 0.32	57.33 ± 0.40	1.1486 ± 0.0302	64.07 ± 0.32
DDOME (adapted)	39.53 ± 20.13	32.44 ± 18.49	2.1081 ± 0.1089	34.30 ± 24.36
FedCoREx (FedGMT base)	84.33 ± 1.65	74.41 ± 3.01	0.4116 ± 0.0241	84.25 ± 1.76
FC				
FedAvg	56.09 ± 0.15	47.33 ± 0.38	1.4220 ± 0.0152	55.85 ± 0.32
FedProx	55.58 ± 0.51	46.70 ± 1.30	1.4252 ± 0.0128	55.34 ± 0.38
FedGMT	64.93 ± 0.27	56.41 ± 0.86	1.0121 ± 0.0033	64.80 ± 0.27
Vanilla-MoE	56.13 ± 0.92	45.81 ± 5.18	1.3993 ± 0.0079	55.79 ± 0.85
DDOME (adapted)	16.59 ± 2.48	7.67 ± 1.87	2.5017 ± 0.3584	12.62 ± 3.14
FedCoREx (FedGMT base)	81.69 ± 1.57	71.93 ± 3.36	0.5506 ± 0.0368	81.68 ± 1.56

Table 14: Natural-client datasets: all formal main baselines and FedCoREx (three-seed mean ± sample SD). Sample-weighted accuracy weights clients by their test-set sizes; the other accuracy columns follow Appendix E.2.

Method	Accuracy	Bottom-10%	NLL	Pooled macro-F1	Weighted acc.
FEMNIST (Natural)					
FedAvg	61.79 ± 0.60	38.81 ± 1.07	1.6684 ± 0.0496	34.51 ± 1.03	61.51 ± 0.63
FedProx	61.43 ± 0.25	38.30 ± 1.53	1.6840 ± 0.0561	33.77 ± 0.97	60.95 ± 0.49
FedGMT	71.45 ± 0.51	55.74 ± 1.32	1.1308 ± 0.0429	46.19 ± 1.43	72.13 ± 0.60
Vanilla-MoE	62.31 ± 1.17	42.77 ± 1.39	1.7144 ± 0.0966	33.93 ± 1.64	62.27 ± 1.57
DDOME (adapted)	49.09 ± 0.41	25.94 ± 0.99	3.0866 ± 0.2742	23.08 ± 1.89	48.02 ± 0.91
FedCoREx (FedGMT base)	72.89 ± 1.80	57.73 ± 3.71	1.0706 ± 0.0761	47.17 ± 1.72	73.73 ± 1.76
Sentiment140 (Natural)					
FedAvg	65.21 ± 0.59	35.75 ± 3.95	0.6384 ± 0.0072	62.62 ± 1.45	65.60 ± 0.35
FedProx	65.14 ± 0.49	35.47 ± 3.50	0.6386 ± 0.0073	62.51 ± 1.35	65.50 ± 0.19
FedGMT	65.95 ± 1.37	40.34 ± 2.05	0.6114 ± 0.0058	63.21 ± 1.63	66.18 ± 1.40
Vanilla-MoE	65.66 ± 1.68	37.74 ± 2.62	0.6287 ± 0.0038	63.27 ± 1.33	65.79 ± 1.44
DDOME (adapted)	58.94 ± 2.30	18.31 ± 4.08	0.7131 ± 0.0710	45.31 ± 7.58	59.48 ± 2.48
FedCoREx (FedGMT base)	66.20 ± 1.16	39.67 ± 1.56	0.6119 ± 0.0059	63.30 ± 1.58	66.44 ± 1.19

Table 15: FedCoREx minus its identical deployed FedGMT base: paired mean ± sample SD. Accuracy, tail, and F1 differences are pp; negative ΔNLL is better.

Condition	ΔAcc.	ΔBottom	ΔNLL	ΔF1
CIFAR-10				
IID	$+0.022 \pm 0.073$	$+0.044 \pm 0.468$	-0.00108 ± 0.00097	$+0.029 \pm 0.059$
Label	-0.107 ± 0.096	$+0.133 \pm 1.162$	-0.02491 ± 0.00092	-0.157 ± 0.112
Concept	$+11.444 \pm 2.957$	$+13.378 \pm 2.431$	-0.29281 ± 0.04313	$+11.611 \pm 2.899$
FC	$+9.151 \pm 1.271$	$+11.600 \pm 3.002$	-0.26823 ± 0.01899	$+9.272 \pm 1.303$
Fashion-MNIST				
IID	$+0.052 \pm 0.090$	-0.111 ± 0.111	-0.01148 ± 0.00435	$+0.048 \pm 0.079$
Label	$+0.856 \pm 0.206$	$+2.667 \pm 1.444$	-0.02329 ± 0.00640	$+0.866 \pm 0.206$
Concept	$+15.652 \pm 1.570$	$+12.852 \pm 3.407$	-0.46911 ± 0.02054	$+15.639 \pm 1.676$
FC	$+16.767 \pm 1.830$	$+15.519 \pm 3.780$	-0.46147 ± 0.04007	$+16.875 \pm 1.827$
FEMNIST				
Natural	$+1.440 \pm 1.493$	$+1.987 \pm 3.478$	-0.06022 ± 0.03351	$+0.972 \pm 0.502$
Sentiment140				
Natural	$+0.255 \pm 0.244$	-0.672 ± 1.163	$+0.00045 \pm 0.00040$	$+0.088 \pm 0.083$

Table 16: Sample-weighted accuracy on natural-client datasets. Absolute values are percentages and paired differences are pp; each uses three seeds.

Dataset	Base	Corrected	ΔWeighted acc.
FEMNIST	72.13 ± 0.60	73.73 ± 1.76	$+1.592 \pm 1.276$
Sentiment140	66.18 ± 1.40	66.44 ± 1.19	$+0.259 \pm 0.224$

Table 17: CIFAR-10 five-seed paired confirmation. Means and sample SDs use matched base/correction seeds; the descriptive 95% interval applies to the accuracy difference. The five-seed set includes the three main-table repetitions.

(a) Accuracy gain and interval

Condition	ΔAcc.	95% interval
IID	$+0.016 \pm 0.053$	$[-0.050, +0.082]$
Label	-0.064 ± 0.117	$[-0.209, +0.081]$
Concept	$+11.752 \pm 2.367$	$[+8.812, +14.692]$
FC	$+9.267 \pm 1.059$	$[+7.952, +10.581]$

(b) Secondary paired metrics

Condition	ΔBottom	ΔNLL	ΔF1
IID	$+0.000 \pm 0.340$	-0.00098 ± 0.00088	$+0.016 \pm 0.047$
Label	-0.267 ± 1.162	-0.02526 ± 0.00153	-0.124 ± 0.102
Concept	$+13.653 \pm 1.810$	-0.30043 ± 0.03971	$+11.863 \pm 2.318$
FC	$+11.573 \pm 2.578$	-0.25584 ± 0.02675	$+9.416 \pm 1.040$

F.4 Portability across shared bases

Tables 18 and 19 expand the portability comparison. For each base, we construct context and train the correction from its own deployed checkpoint, using a fixed correction configuration. The comparison includes FedAvg, FedProx, and FedGMT on both visual Concept/FC settings and both natural datasets, plus SCAFFOLD, FedDyn, and FedSMU on CIFAR-10 Concept/FC.

On CIFAR-10, the mean accuracy changes range from 5.956 to 12.929 pp on Concept and from 4.871 to 9.151 pp on FC. FedDyn has the largest Concept increment, while the corrected FedGMT predictor has higher absolute accuracy, illustrating the distinction between correction gain and final

Table 18: Base quality, corrected quality, and paired accuracy gains across all evaluated shared bases (three-seed mean ± sample SD). Absolute accuracy is in percent and differences are pp. The correction configuration is shared across bases.

Base training method	Base acc.	Corrected acc.	ΔAcc.
CIFAR-10 / Concept			
FedAvg	46.12 ± 0.63	55.86 ± 2.24	$+9.738 \pm 1.937$
FedProx	45.73 ± 0.70	55.56 ± 2.22	$+9.827 \pm 2.081$
FedGMT	56.47 ± 0.91	67.92 ± 3.34	$+11.444 \pm 2.957$
SCAFFOLD	45.75 ± 0.71	54.29 ± 2.15	$+8.538 \pm 1.570$
FedDyn	49.13 ± 0.33	62.06 ± 2.62	$+12.929 \pm 2.291$
FedSMU	49.03 ± 1.11	54.99 ± 4.15	$+5.956 \pm 3.362$
CIFAR-10 / FC			
FedAvg	35.89 ± 0.77	40.76 ± 1.88	$+4.871 \pm 2.619$
FedProx	35.53 ± 0.15	41.30 ± 1.75	$+5.764 \pm 1.898$
FedGMT	45.79 ± 0.86	54.94 ± 1.63	$+9.151 \pm 1.271$
SCAFFOLD	35.45 ± 0.90	41.99 ± 1.37	$+6.542 \pm 0.568$
FedDyn	39.84 ± 0.31	45.78 ± 2.72	$+5.942 \pm 2.768$
FedSMU	38.08 ± 1.07	42.98 ± 2.90	$+4.902 \pm 2.637$
Fashion-MNIST / Concept			
FedAvg	64.77 ± 0.07	77.21 ± 3.79	$+12.441 \pm 3.825$
FedProx	64.25 ± 0.72	78.51 ± 5.08	$+14.259 \pm 4.669$
FedGMT	68.67 ± 0.15	84.33 ± 1.65	$+15.652 \pm 1.570$
Fashion-MNIST / FC			
FedAvg	56.09 ± 0.15	64.75 ± 4.33	$+8.656 \pm 4.237$
FedProx	55.58 ± 0.51	63.60 ± 2.37	$+8.019 \pm 2.297$
FedGMT	64.93 ± 0.27	81.69 ± 1.57	$+16.767 \pm 1.830$
FEMNIST / Natural			
FedAvg	61.79 ± 0.60	63.49 ± 0.45	$+1.701 \pm 1.005$
FedProx	61.43 ± 0.25	64.51 ± 1.32	$+3.075 \pm 1.385$
FedGMT	71.45 ± 0.51	72.89 ± 1.80	$+1.440 \pm 1.493$
Sentiment140 / Natural			
FedAvg	65.21 ± 0.59	64.95 ± 0.70	-0.259 ± 0.168
FedProx	65.14 ± 0.49	64.96 ± 0.81	-0.182 ± 0.337
FedGMT	65.95 ± 1.37	66.20 ± 1.16	$+0.255 \pm 0.244$

performance. Corrections to FedAvg, FedProx, and FedGMT also improve mean accuracy and NLL in the Concept/FC settings on both image datasets.

FEMNIST has positive mean accuracy effects for all three bases. On Sentiment140, mean accuracy decreases by 0.259 and 0.182 pp for FedAvg and FedProx, respectively, while the FedGMT change is small and positive.

G Mechanism Controls and Expert Granularity

The controls below examine the roles of residual capacity, client context, training budget, and expert granularity in the gains reported in Section 5.

G.1 Structural controls and capacity accounting

The controls keep the deployed base frozen and use the same 200-round correction protocol, loss, participation schedule, and validation criterion. Table 20 lists their architectures and parameter counts. Direct64 uses one width-64 residual network; Uniform4×16 averages four width-16 residual experts with fixed weights $1/4$; NoContext retains a learned sample-dependent router but replaces the client-context block by zeros; Full is FedCoREx. In these visual controls, NoContext retains the full router input shape and parameter count.

Table 19: Tail, probability, and pooled macro-F1 effects across evaluated shared bases. Each cell is a three-seed paired mean ± sample SD.

Base training method	ΔBottom	ΔNLL	ΔF1
CIFAR-10 / Concept			
FedAvg	$+10.178 \pm 4.677$	-0.36890 ± 0.06151	$+9.626 \pm 2.158$
FedProx	$+9.867 \pm 2.888$	-0.36778 ± 0.06109	$+9.705 \pm 2.256$
FedGMT	$+13.378 \pm 2.431$	-0.29281 ± 0.04313	$+11.611 \pm 2.899$
SCAFFOLD	$+10.044 \pm 4.481$	-0.27656 ± 0.03719	$+8.675 \pm 1.659$
FedDyn	$+12.889 \pm 3.215$	-0.46664 ± 0.04522	$+13.038 \pm 2.463$
FedSMU	$+7.778 \pm 5.243$	-0.48944 ± 0.11304	$+5.900 \pm 3.487$
CIFAR-10 / FC			
FedAvg	$+7.600 \pm 1.923$	-0.20657 ± 0.07112	$+4.991 \pm 3.039$
FedProx	$+8.000 \pm 2.083$	-0.23992 ± 0.05041	$+5.887 \pm 2.244$
FedGMT	$+11.600 \pm 3.002$	-0.26823 ± 0.01899	$+9.272 \pm 1.303$
SCAFFOLD	$+9.867 \pm 0.353$	-0.23886 ± 0.01206	$+6.774 \pm 0.527$
FedDyn	$+6.356 \pm 5.553$	-0.20797 ± 0.03277	$+5.369 \pm 2.873$
FedSMU	$+4.578 \pm 3.774$	-0.26325 ± 0.03906	$+4.759 \pm 2.251$
Fashion-MNIST / Concept			
FedAvg	$+10.407 \pm 4.722$	-0.61280 ± 0.09212	$+12.464 \pm 3.834$
FedProx	$+11.407 \pm 8.095$	-0.62551 ± 0.13079	$+13.995 \pm 4.976$
FedGMT	$+12.852 \pm 3.407$	-0.46911 ± 0.02054	$+15.639 \pm 1.676$
Fashion-MNIST / FC			
FedAvg	$+8.815 \pm 2.686$	-0.39478 ± 0.14563	$+8.614 \pm 3.907$
FedProx	$+5.444 \pm 2.655$	-0.36037 ± 0.11485	$+7.998 \pm 1.713$
FedGMT	$+15.519 \pm 3.780$	-0.46147 ± 0.04007	$+16.875 \pm 1.827$
FEMNIST / Natural			
FedAvg	$+4.286 \pm 2.395$	-0.05192 ± 0.03259	$+1.179 \pm 0.792$
FedProx	$+6.160 \pm 2.863$	-0.08044 ± 0.02814	$+2.592 \pm 2.068$
FedGMT	$+1.987 \pm 3.478$	-0.06022 ± 0.03351	$+0.972 \pm 0.502$
Sentiment140 / Natural			
FedAvg	$+0.424 \pm 0.555$	-0.00093 ± 0.00121	$+0.057 \pm 0.387$
FedProx	$+0.702 \pm 0.432$	-0.00087 ± 0.00130	$+0.158 \pm 0.193$
FedGMT	-0.672 ± 1.163	$+0.00045 \pm 0.00040$	$+0.088 \pm 0.083$

Table 20: Visual structural controls ($d_h = 84$, $C = 10$). All use a 500-round frozen base and a 200-round correction stage. Counts exclude the frozen base and the non-trainable context.

Control	Mixture information	Expert parameters	Router	Total
Direct64	No mixture	6,080	0	6,080
Uniform4×16	Fixed mean	6,080	0	6,080
NoContext	Sample state; zero context	6,080	776	6,856
Full	Sample state and client context	6,080	776	6,856

Direct64 and Uniform have fewer parameters because they omit the router. Their functions and optimization geometry also differ: averaging four outputs changes gradient scaling relative to a single wider network, even when the expert-bank parameter totals agree. All output projections start at zero.

G.2 Structural and context ablations

Table 21 reports all four metrics for the CIFAR-10 structural controls. Direct64, Uniform, and NoContext remain close to the base on both Concept and FC, while Full reaches 67.916% and 54.942% accuracy, respectively. Lower NLL and better tail-client accuracy accompany this separation.

Table 22 extends the Full–NoContext comparison across datasets. The visual concept conditions show large context-associated differences. FEMNIST has a smaller mean accuracy gain of 1.575

Table 21: CIFAR-10 structural controls: three-seed mean ± sample SD of complete test metrics. The upstream base is the identical deployed FedGMT checkpoint for all residual variants.

Variant	Accuracy	Bottom-10%	NLL	Pooled macro-F1
Concept				
FedGMT	56.47 ± 0.91	45.29 ± 1.56	1.2064 ± 0.0143	56.32 ± 0.91
Direct64	56.23 ± 0.94	45.29 ± 1.30	1.2014 ± 0.0154	56.07 ± 0.91
Uniform4×16	56.03 ± 0.83	45.56 ± 1.61	1.2007 ± 0.0138	55.86 ± 0.85
NoContext	56.02 ± 0.67	45.47 ± 1.67	1.2031 ± 0.0163	55.84 ± 0.70
FedCoREx (FedGMT base)	67.92 ± 3.34	58.67 ± 2.63	0.9136 ± 0.0560	67.93 ± 3.26
FC				
FedGMT	45.79 ± 0.86	34.80 ± 0.80	1.5443 ± 0.0362	45.27 ± 0.93
Direct64	45.52 ± 0.75	34.89 ± 0.56	1.5367 ± 0.0375	45.01 ± 0.70
Uniform4×16	45.66 ± 0.89	34.67 ± 0.74	1.5366 ± 0.0366	45.22 ± 0.82
NoContext	45.52 ± 1.07	34.76 ± 0.20	1.5345 ± 0.0365	45.06 ± 1.03
FedCoREx (FedGMT base)	54.94 ± 1.63	46.40 ± 2.72	1.2761 ± 0.0454	54.55 ± 1.39

pp, accompanied by a mean NLL decrease of 0.05586, while the Sentiment140 difference is near neutral (0.066 pp accuracy).

Table 22: Full minus NoContext across datasets (three-seed paired mean ± sample SD). Accuracy, tail, and F1 differences are pp.

Condition	ΔAcc.	ΔBottom	ΔNLL	ΔF1
CIFAR-10				
Concept	$+11.898 \pm 3.058$	$+13.200 \pm 2.623$	-0.28957 ± 0.04077	$+12.083 \pm 2.979$
FC	$+9.422 \pm 1.065$	$+11.644 \pm 2.648$	-0.25842 ± 0.01670	$+9.482 \pm 1.018$
Fashion-MNIST				
Concept	$+16.100 \pm 1.466$	$+13.185 \pm 3.251$	-0.45199 ± 0.02039	$+16.113 \pm 1.562$
FC	$+17.115 \pm 1.948$	$+16.074 \pm 3.635$	-0.44326 ± 0.04099	$+17.196 \pm 1.907$
FEMNIST				
Natural	$+1.575 \pm 1.204$	$+2.659 \pm 1.801$	-0.05586 ± 0.04086	$+0.926 \pm 0.540$
Sentiment140				
Natural	$+0.066 \pm 0.138$	-0.417 ± 0.722	$+0.00010 \pm 0.00010$	-0.012 ± 0.124

For the visual NoContext control, the router retains its input dimensionality while the context block is masked. The Sentiment140 adaptation omits its four context inputs, reducing the router from 134 to 130 inputs; its Full–NoContext comparison includes this small parameter difference. The gap between these learned model families reflects both expressiveness and optimization.

G.3 Extended-base-training controls

Table 23 compares the 500-round FedGMT base, a 700-round FedGMT control, and the 500+200-round FedCoREx pipeline. The shared-model controls use their own base deployment rule from Appendix E.1; correction selects its complete residual state by client-macro validation NLL while keeping the 500-round deployed base fixed.

Extending base training changes mean accuracy from 56.471% to 56.640% on Concept and from 45.791% to 45.609% on FC. The corrected predictors exceed the extended bases by 11.276 and 9.333 pp, respectively. The comparison holds additional communication rounds fixed; base training and correction have different trainable states and costs.

G.4 Expert granularity at near-equal budgets

The granularity study changes the number of experts while keeping the correction parameter budget nearly constant. For $K \in \{1, 2, 4, 8\}$, widths are $72, 34, 16, 7$, respectively. The $K = 1$ control has

Table 23: CIFAR-10 extended-base-training control. FedGMT uses 500 rounds, FedGMT-700 continues it for 200 rounds, and FedCoREx uses a 500-round upstream run plus 200 correction rounds. Metrics are three-seed means ± sample SD.

Variant	Accuracy	Bottom-10%	NLL	Pooled macro-F1
Concept				
FedGMT	56.47 ± 0.91	45.29 ± 1.56	1.2064 ± 0.0143	56.32 ± 0.91
FedGMT-700	56.64 ± 0.52	45.56 ± 1.30	1.2040 ± 0.0113	56.50 ± 0.53
FedCoREx (FedGMT base)	67.92 ± 3.34	58.67 ± 2.63	0.9136 ± 0.0560	67.93 ± 3.26
FC				
FedGMT	45.79 ± 0.86	34.80 ± 0.80	1.5443 ± 0.0362	45.27 ± 0.93
FedGMT-700	45.61 ± 0.12	36.18 ± 0.62	1.5434 ± 0.0268	45.37 ± 0.28
FedCoREx (FedGMT base)	54.94 ± 1.63	46.40 ± 2.72	1.2761 ± 0.0454	54.55 ± 1.39

no router. Table 24 shows that the total counts differ from the fixed main configuration with $K = 4$ by less than 0.25%. Expert count and per-expert width vary together.

Table 24: Exact CIFAR-10 correction capacities in the expert-granularity study. The main instance remains $K = 4, W = 16$.

K	Width W	Expert bank	Router	Total
1	72	6,840	0	6,840
2	34	6,460	388	6,848
4	16	6,080	776	6,856
8	7	5,320	1,552	6,872

Table 25: Near-equal-budget expert granularity: three-seed accuracy, lower-tail accuracy, and paired effects. Differences are pp; summaries report means and sample SDs.

Condition	K	Accuracy	Δ vs. base	Δ vs. $K = 4$	Bottom-10%
Concept	1	56.22 ± 0.85	-0.253 ± 0.232	-11.698 ± 3.189	45.64 ± 1.37
Concept	2	65.37 ± 0.19	$+8.902 \pm 0.826$	-2.542 ± 3.430	53.73 ± 1.64
Concept	4	67.92 ± 3.34	$+11.444 \pm 2.957$	Reference	58.67 ± 2.63
Concept	8	71.10 ± 2.13	$+14.627 \pm 2.077$	$+3.182 \pm 1.416$	62.76 ± 5.76
FC	1	45.54 ± 0.87	-0.253 ± 0.129	-9.404 ± 1.143	35.02 ± 0.43
FC	2	52.56 ± 0.81	$+6.764 \pm 0.547$	-2.387 ± 0.856	43.24 ± 0.67
FC	4	54.94 ± 1.63	$+9.151 \pm 1.271$	Reference	46.40 ± 2.72
FC	8	56.24 ± 1.59	$+10.453 \pm 1.662$	$+1.302 \pm 0.812$	48.49 ± 3.66

Table 26: Expert-granularity probability quality, F1, and checkpoint selection. Metrics, paired NLL differences, and selected rounds are means ± sample SD over three repetitions.

Condition	K	NLL	ΔNLL vs. base	Pooled macro-F1	Selected round
Concept	1	1.2016 ± 0.0162	-0.00477 ± 0.00192	56.04 ± 0.86	36.3 ± 9.3
Concept	2	0.9638 ± 0.0067	-0.24262 ± 0.01646	65.27 ± 0.25	162.7 ± 5.0
Concept	4	0.9136 ± 0.0560	-0.29281 ± 0.04313	67.93 ± 3.26	194.0 ± 4.6
Concept	8	0.8528 ± 0.0392	-0.35355 ± 0.02679	71.09 ± 2.13	199.0 ± 0.0
FC	1	1.5370 ± 0.0369	-0.00734 ± 0.00452	45.05 ± 0.84	37.7 ± 12.7
FC	2	1.3578 ± 0.0296	-0.18656 ± 0.00850	52.07 ± 0.76	184.7 ± 10.5
FC	4	1.2761 ± 0.0454	-0.26823 ± 0.01899	54.55 ± 1.39	195.0 ± 6.9
FC	8	1.2653 ± 0.0458	-0.27897 ± 0.03512	55.89 ± 1.56	198.3 ± 1.2

Tables 25 and 26 report absolute results, paired differences, and selection information. The one-expert control yields little improvement; two experts recover a substantial fraction of the multi-expert gain, and four further improve the mean result. Eight experts exceed four by 3.182 pp on

Concept and 1.302 pp on FC. Within this scan, distributing the budget across routed residual functions yields larger mean gains than widening a single residual function. The scan covers two conditions, with $K = 4$ retained as the main configuration. For $K = 8$, the selected-round summaries lie near the end of the 200-round budget; longer runs could test whether the gains continue.

G.5 Permutation-invariant routing diagnostics

Routing statistics summarize the selected correction checkpoint on validation samples. Within a run, let q_{jk} be the soft routing weight for expert k on validation example j, pooling the validation examples of all 50 clients. These CIFAR-10 partitions have equal validation sizes, so sample pooling also gives equal client weight. Define

$$\overline{H}_q = -\frac{1}{m}\sum_{j=1}^{m}\sum_{k=1}^{K} q_{jk}\log q_{jk}, \qquad H_q^{\mathrm{norm}} = \frac{\overline{H}_q}{\log K}, \qquad K > 1, \tag{G.1}$$

$$N_{\mathrm{eff}} = \exp(\overline{H}_q), \qquad Q_{\max} = \frac{1}{m}\sum_{j=1}^{m}\max_k q_{jk}, \tag{G.2}$$

$$L_{\max} = \max_k \frac{1}{m}\sum_{j=1}^{m} q_{jk}, \qquad T_{\max} = \max_k \frac{1}{m}\sum_{j=1}^{m}\mathbf{1}\{k = \arg\max_\ell q_{j\ell}\}. \tag{G.3}$$

For $K = 1$, normalized entropy is reported as zero by convention and $N_{\mathrm{eff}} = 1$. The effective count exponentiates the *mean entropy*; equivalently, it is the geometric mean of the per-example effective counts, which emphasizes typical per-example concentration. The residual magnitude is

$$R_\Delta = \frac{1}{Km}\sum_{k=1}^{K}\sum_{j=1}^{m}\|\Delta_k(h_j)\|_2, \tag{G.4}$$

which averages the individual expert norms. All quantities are computed within a seed first and then averaged across the three seeds in Table 27.

Table 27: Permutation-invariant validation routing summaries, averaged over three seeds. $\overline{H}_q$ is raw mean entropy, $H_q^{\mathrm{norm}} = \overline{H}_q/\log K$, $Q_{\max}$ is mean sample-wise maximum routing probability, $L_{\max}$ is the largest soft load, and $T_{\max}$ the largest hard argmax share. R_Δ is mean expert-residual norm.

Condition	K	$\overline{H}_q$	H_q^{norm}	N_{eff}	N_{eff}/K	$Q_{\max}$	$L_{\max}$	$T_{\max}$	R_Δ
Concept	1	0.000	0.000	1.000	1.000	1.000	1.000	1.000	0.885
Concept	2	0.149	0.215	1.161	0.580	0.942	0.659	0.665	4.328
Concept	4	0.304	0.219	1.358	0.340	0.875	0.451	0.453	2.925
Concept	8	0.228	0.110	1.256	0.157	0.932	0.387	0.387	1.869
FC	1	0.000	0.000	1.000	1.000	1.000	1.000	1.000	0.709
FC	2	0.236	0.340	1.266	0.633	0.894	0.588	0.615	3.425
FC	4	0.186	0.134	1.209	0.302	0.941	0.448	0.452	2.581
FC	8	0.323	0.155	1.384	0.173	0.898	0.339	0.337	1.340

On Concept, N_{eff} is approximately 1.161, 1.358, and 1.256 for $K = 2, 4, 8$; on FC, it is 1.266, 1.209, and 1.384. The distributions can therefore be sharp for individual examples while different experts are used across examples. For $K = 8$, the largest hard-assignment share is about 0.387 on Concept and 0.337 on FC, indicating that different examples favor different experts. The summaries are invariant to expert-index permutations across repetitions. Hard assignments are diagnostic summaries of the soft weights used for prediction.

H Heterogeneity and Residual-Structure Diagnostics

These training-side diagnostics expand Section 5.5's comparison between marginal heterogeneity, base-prediction structure, and observed correction gains.

H.1 DIAGNOSTIC DEFINITIONS

The census computes all descriptors from local training data with deterministic preprocessing and the deployment-selected FedGMT base for the corresponding condition and seed. Let n_i be client i's training size and $\pi_i \in \Delta^{C-1}$ its empirical label distribution. For probability vectors with C coordinates, define the normalized equal-client Jensen–Shannon divergence

$$\mathcal{J}_C(v_1, \dots, v_N) = \frac{H(N^{-1}\sum_i v_i) - N^{-1}\sum_i H(v_i)}{\log C}. \tag{H.1}$$

The quantity and label descriptors are

$$H_N = \frac{\sum_{i=1}^N \sum_{j=1}^N |n_i - n_j|}{2N\sum_i n_i}, \qquad H_Y = \mathcal{J}_C(\pi_1, \dots, \pi_N). \tag{H.2}$$

Label distributions use the effective labels after the specified concept mapping.

For feature heterogeneity, extract the frozen representation $h(x)$ and let

$$\mu_i = \frac{1}{n_i}\sum_{x \in D_i^{\mathrm{tr}}} h(x), \qquad v_i = \frac{1}{n_i}\sum_{x \in D_i^{\mathrm{tr}}} \|h(x) - \mu_i\|_2^2, \qquad \bar{\mu} = \frac{1}{N}\sum_i \mu_i. \tag{H.3}$$

Then

$$H_X = \frac{N^{-1}\sum_i \|\mu_i - \bar{\mu}\|_2^2}{N^{-1}\sum_i \|\mu_i - \bar{\mu}\|_2^2 + N^{-1}\sum_i v_i + 10^{-12}}. \tag{H.4}$$

This is an equal-client between/total feature-dispersion ratio. Because the encoder is trained separately for each condition and seed, H_X measures dispersion in that encoder's representation space, conditional on its training and data distribution.

Confusion projection. Let $M_i[a, b]$ count training examples with effective label a and frozen-base prediction b. With the same unit Laplace smoothing as the method context,

$$P_i[a, b] = \frac{M_i[a, b] + 1}{n_{ia} + C}, \qquad n_{ia} = \sum_b M_i[a, b]. \tag{H.5}$$

An unsupported class row is uniform. We denote cross-client dispersion in class-conditional confusion by SRH and define

$$\mathrm{ErrorMass} = 1 - \frac{1}{NC}\sum_{i=1}^N \sum_{a=1}^C P_i[a, a], \tag{H.6}$$

$$\mathrm{SRH}_{\mathrm{raw}} = \frac{1}{C}\sum_{a=1}^C \mathcal{J}_C\big(P_1[a, :], \dots, P_N[a, :]\big). \tag{H.7}$$

Error Mass is a smoothed, equal-client, equal-class error summary, distinct from the sample-weighted training error rate. SRH summarizes this projection of residual heterogeneity.

Support-matched null. Finite and unequal class support can itself create cross-client variation. For each true class a, pool the unsmoothed counts to obtain

$$\rho_a[b] = \frac{\sum_i M_i[a, b]}{\sum_i n_{ia}}, \tag{H.8}$$

using a uniform vector if the pooled support is zero. In each of $B = 500$ null replications, independently draw

$$\widetilde{M}_i^{(b)}[a, :] \sim \mathrm{Multinomial}(n_{ia}, \rho_a), \tag{H.9}$$

preserving every client–class support n_{ia}, then apply the same smoothing and Eq. (H.7). The summaries are

$$\mathrm{SRH}_{\mathrm{null}} = \frac{1}{B}\sum_{b=1}^B \widetilde{\mathrm{SRH}}_{\mathrm{raw}}^{(b)}, \qquad \mathrm{SRH}_{\mathrm{excess}} = \max\{0, \mathrm{SRH}_{\mathrm{raw}} - \mathrm{SRH}_{\mathrm{null}}\}. \tag{H.10}$$

Truncation is applied within each trained-base seed before seed averaging. Subtracting the specified null mean adjusts for support-induced variation; other finite-sample biases can remain. These empirical descriptors summarize projected residual structure, rather than estimating $I(Y; \Xi \mid S)$ or the approximation and learning gaps in the theoretical decomposition.

H.2 CONTROLLED-CONDITION CENSUS

Table 28 compares marginal heterogeneity and confusion-based diagnostics with correction effects using matched condition means. The remaining tables give the full descriptor census.

Table 28: Heterogeneity and residual utility (three-seed means). NLL reduction is Base−Corrected, so a positive value is an improvement; ΔA remains Corrected−Base. All controlled conditions shown have $H_N = 0$.

Setting	Marginal descriptors		Base confusion descriptors		Correction effect	
	H_Y	H_X	Error mass	SRH excess	ΔA (pp)	NLL reduction↑ (Base−Corrected)
CIFAR-10: controlled settings						
IID	0.0000	0.0009	0.2573	0.0001	+0.022	+0.00108
Label	0.3699	0.1056	0.4916	0.0002	−0.107	+0.02491
Concept	0.0000	0.0007	0.4433	0.1180	+11.444	+0.29281
FC	0.0000	0.0062	0.5315	0.0852	+9.151	+0.26823
Fashion-MNIST: controlled settings						
IID	0.0000	0.0003	0.1076	0.000078	+0.052	+0.01148
Label	0.3858	0.2147	0.3737	0.000571	+0.856	+0.02329
Concept	0.0000	0.0003	0.3186	0.177523	+15.652	+0.46911
FC	0.0000	0.0360	0.3236	0.154656	+16.767	+0.46147
FEMNIST: natural-client boundary						
Natural	0.0412	0.0434	0.9518	0.000072	+1.440	+0.06022

FEMNIST has $H_N = 0.1272$ and 62 classes; absolute confusion descriptors are not directly comparable across label spaces. Quantity and Feature have diagnostic statistics but no aligned correction effects and are excluded from this matched-effect table.

Tables 29 and 30 cover six CIFAR-10 constructions, and Tables 31 and 32 cover four Fashion-MNIST constructions. For this diagnostic section only, *NLL reduction* is Base minus Corrected, so both accuracy gain and NLL reduction have a positive-improvement direction. Quantity and Feature have descriptors but no checkpoint-aligned correction comparison in this evidence set; their effect cells remain unavailable.

Table 29: CIFAR-10: equal-client marginal heterogeneity and smoothed Error Mass. Values are condition means over the three main algorithm seeds; H_N and H_Y depend only on the fixed data.

Condition	H_N	H_Y	H_X	Error Mass
IID	0.0000	0.0000	0.0009	0.2573
Label	0.0000	0.3699	0.1056	0.4916
Quantity	0.3596	0.0000	0.0015	0.2933
Feature	0.0000	0.0000	0.0097	0.3626
Concept	0.0000	0.0000	0.0007	0.4433
FC	0.0000	0.0000	0.0062	0.5315

Table 30: CIFAR-10: confusion-projected residual heterogeneity and aligned FedCoREx effects. Null subtraction and clipping occur within each seed before averaging. NLL reduction is Base minus Corrected; N/A indicates absent aligned performance evidence.

Condition	SRH raw	SRH null	SRH excess	ΔAcc. (pp)	NLL reduction
IID	0.010091	0.010035	0.000070	+0.022	+0.00108
Label	0.078683	0.078582	0.000169	-0.107	+0.02491
Quantity	0.012530	0.017254	0.000000	N/A	–
Feature	0.013693	0.013617	0.000077	N/A	–
Concept	0.130925	0.012975	0.117950	+11.444	+0.29281
FC	0.101696	0.016503	0.085192	+9.151	+0.26823

Table 31: Fashion-MNIST: equal-client marginal heterogeneity and smoothed Error Mass. Values are condition means over the three main algorithm seeds; H_N and H_Y depend only on the fixed data.

Condition	H_N	H_Y	H_X	Error Mass
IID	0.0000	0.0000	0.0003	0.1076
Label	0.0000	0.3858	0.2147	0.3737
Concept	0.0000	0.0000	0.0003	0.3186
FC	0.0000	0.0000	0.0360	0.3236

Table 32: Fashion-MNIST: confusion-projected residual heterogeneity and aligned FedCoREx effects. Null subtraction and clipping occur within each seed before averaging. NLL reduction is Base minus Corrected; N/A indicates absent aligned performance evidence.

Condition	SRH raw	SRH null	SRH excess	ΔAcc. (pp)	NLL reduction
IID	0.001454	0.001376	0.000078	+0.052	+0.01148
Label	0.104639	0.104068	0.000571	+0.856	+0.02329
Concept	0.181442	0.003918	0.177523	+15.652	+0.46911
FC	0.160338	0.005683	0.154656	+16.767	+0.46147

The census separates marginal imbalance from cross-client variation in class-conditional predictions. For CIFAR-10 Label, $H_Y = 0.3699$ and Error Mass is 0.4916, yet SRH excess is approximately 0.0002 and the accuracy gain is near zero. Concept and FC have similar label marginals across clients but retain much larger excess confusion dispersion (0.1180 and 0.0852), together with large correction gains. Fashion-MNIST shows the same pattern: Label's raw SRH of 0.104639 is almost entirely explained by its matched-support null mean of 0.104068, while Concept and FC retain excess values of 0.177523 and 0.154656. Across these conditions, similar label marginals coexist with different class-conditional base errors and correction gains.

H.3 Descriptive alignment at the condition level

Table 33 correlates each descriptor with the mean paired correction effect over the four conditions with formal effects in each visual dataset. Each dataset contributes four condition-level observations to this descriptive calculation; the 50 clients and individual seed records are aggregated within conditions. Spearman's ρ and Kendall's τ_b are computed separately for each dataset and outcome. Quantity heterogeneity is constant across these four conditions, making its correlations undefined.

Table 33: Descriptive condition-level rank alignment ($n = 4$ per dataset). The two outcomes are accuracy gain and NLL reduction; both positive directions denote improvement. A dash denotes an undefined correlation for constant H_N. No p-values are computed.

Descriptor	Acc.: ρ	Acc.: τ_b	NLL reduction: ρ	NLL reduction: τ_b
CIFAR-10				
H_N	–	–	–	–
H_Y	-0.775	-0.707	-0.258	-0.236
H_X	-0.800	-0.667	-0.400	-0.333
Error Mass	+0.000	+0.000	+0.400	+0.333
SRH excess	+0.800	+0.667	+1.000	+1.000
Fashion-MNIST				
H_N	–	–	–	–
H_Y	-0.258	-0.236	-0.258	-0.236
H_X	+0.400	+0.333	+0.200	+0.000
Error Mass	+0.400	+0.333	+0.200	+0.000
SRH excess	+0.800	+0.667	+1.000	+1.000

SRH excess has $\rho = 0.8$ and $\tau_b = 2/3$ with accuracy gain in each visual dataset, and both rank statistics equal one for NLL reduction. The correlations describe this four-condition ordering, to

which several properties of the controlled constructions may contribute. Statistical significance, a causal role for confusion dispersion, and predictive validity on new tasks remain untested.

H.4 FEMNIST AND CONSTRUCTION SENSITIVITY

On FEMNIST, SRH excess is approximately 0.000072, yet the same-base correction improves mean accuracy by 1.440 pp and mean NLL by 0.06022 (Table 34). Near-zero excess under this confusion projection therefore does not rule out useful correction.

Table 34: FEMNIST natural-client boundary: three-seed mean $\pm$ sample SD. Coverage is the fraction of 62 classes observed in a writer's training split, averaged over writers.

Descriptor	Value
H_N	0.1272 ± 0.0000
H_Y	0.0412 ± 0.0000
H_X	0.0434 ± 0.0036
Error Mass	0.9518 ± 0.0005
SRH raw	0.001204 ± 0.000033
SRH null	0.001131 ± 0.000025
SRH excess	0.000072 ± 0.000010
Mean writer class coverage	0.8065 ± 0.0000

The 62-class natural-writer setting also changes the behavior of the descriptor. On average, only about 80.6% of classes occur in a client's training split, and observed classes can have few examples. Unit smoothing contributes 62 pseudo-counts to each row, with uniform rows for absent classes. Class support and smoothing therefore affect Error Mass and support-corrected confusion variation differently from the balanced ten-class constructions, limiting direct comparisons of their absolute values.

Table 35: CIFAR-10 single-factor SRH checks with one fixed training randomization. Mapping variants use the same data realization; the data-realization row summarizes three allocations by mean $\pm$ sample SD. The canonical allocation is shared by the two checks.

Mapping	n	SRH raw	SRH null	SRH excess
Concept mapping				
01/23/456/789	1	0.127510	0.012435	0.115075
02/14/357/689	1	0.132590	0.012887	0.119703
05/28/139/467	1	0.141138	0.013425	0.127713
Data realization				
02/14/357/689	3	0.129571 ± 0.002932	0.013071 ± 0.000263	0.116500 ± 0.003171

Table 35 changes one construction factor at a time: the concept-label cycle mapping or the dataset realization, keeping the algorithm seed fixed for this check. The mapping alternatives and the aggregate over three data realizations retain substantial SRH excess. These checks assess how SRH changes with the concept mapping and data realization; prediction gains were not evaluated for these variants.

I COMPLEXITY AND RESOURCE ACCOUNTING

This appendix derives the correction's parameter and communication costs and details the measured overheads summarized in Section 5.5.

I.1 ANALYTIC PARAMETER AND STATE COUNTS

For a representation of size d_h, C classes, K experts, and hidden width W, each expert has an input matrix, hidden bias, and bias-free output matrix. The router has one weight row per expert, no bias,

and no trainable affine normalization parameters. Consequently,

$$P_{\text{expert}} = KW(d_h+C+1), \qquad P_{\text{router}} = K(d_h+C+C^2), \qquad P_{\text{corr}} = P_{\text{expert}}+P_{\text{router}}. \tag{I.1}$$

The $K = 1$ granularity control omits the router; Eq. (I.1) describes the routed architecture. Table 36 uses the fixed main configuration $K = 4, W = 16$. The deployed neural state contains $P_{\text{base}} + P_{\text{corr}}$ parameters, of which P_{corr} are trained in the second stage. Each client also stores its C^2-entry context locally as non-trainable state, separate from the neural parameter count and transmitted updates.

Table 36: Exact parameter and local-context counts for the main instance. Deployed totals count one predictor, not optimizer state, server teachers, or copies held by multiple workers.

Dataset	Frozen base	Expert bank	Router	Correction	Deployed	Context
CIFAR-10	92,246	6,080	776	6,856	99,102	100
Fashion-MNIST	91,946	6,080	776	6,856	98,802	100
FEMNIST	96,366	9,408	15,960	25,368	121,734	3,844
Sentiment140	285,762	8,384	536	8,920	294,682	4

Correction adds about 7.43%, 7.46%, 26.33%, and 3.12% to the respective base neural counts. FEMNIST's larger increase comes mainly from the router's C^2 context input, with expert width held fixed. For Sentiment140, the deployed base count includes the $3300 \times 64 = 211{,}200$ embedding parameters once. They are trainable during base learning and frozen during correction. Training-time copies and teacher states are excluded from the deployed embedding count.

I.2 Computation and communication

FedCoREx uses dense soft routing. Beyond the base forward pass, the leading per-example matrix computation is of order

$$O\big(KW(d_h + C) + K(d_h + C + C^2)\big), \tag{I.2}$$

with additional normalization and mixture operations. All K residual experts are evaluated even when the routing distribution is sharp; the accounting therefore uses dense execution.

Ideal parameter-only payload. If a client caches the frozen base, a simple FP32 parameter exchange sends $4P_{\text{corr}}$ bytes down and the same number up for a dense correction update, or $8P_{\text{corr}}$ bytes per participating client per round. This ideal count excludes serialization and metadata, optimizer or transport state, and an initial base distribution of $4P_{\text{base}}$ bytes per recipient when needed. The train-only context remains local and requires no server aggregation.

Recorded logical communication. The resource tables count the framework's serialized *public training payloads* and public client updates. Tensor payloads are represented by dtype, shape, and raw tensor bytes, together with serialized public metadata. Each correction training package contains both the fixed base and the public correction state, so the executed protocol retransmits the base instead of caching it. Public updates include their transmitted tensor deltas and metadata, but exclude private client state such as the locally stored context. The resulting totals exceed the correction-only parameter formula and measure serialized public-message sizes, not network traffic. Transport framing, retransmissions, and hypothetical network compression are outside this accounting.

This distinction also matters for bases such as FedGMT whose training payload includes more than one deployed parameter vector. Communication is summed directly from the recorded messages. Reported GiB use 2^{30} bytes. One-time context construction and an external initial model distribution are not added as extra round messages in these summaries.

I.3 Measured resource costs

Table 37 isolates the additional 200-round correction stage. Table 38 compares the 500-round FedGMT base with the complete base-plus-correction workload, summing logical communication and recorded round times across the two stages. The pipeline GPU statistic is the larger of the two stage observations.

Table 37: Additional 200-round correction stage only. Cost summaries pool four conditions and three seeds on CIFAR-10/Fashion-MNIST, and three seeds on each natural dataset. Time and GPU observations are mean ± sample SD.

Dataset	n	Logical GiB	Cumulative round time (min)	GPU obs. (MiB)
CIFAR-10	12	0.794	12.98 ± 0.45	65.4 ± 0.2
Fashion-MNIST	12	0.792	13.11 ± 0.62	65.3 ± 0.2
FEMNIST	3	1.101	7.33 ± 0.08	65.5 ± 0.1
Sentiment140	3	2.268	6.50 ± 0.02	65.8 ± 0.0

Note. CIFAR-10, Fashion-MNIST, and FEMNIST costs use separate same-workload executions; Sentiment140 costs use its performance runs. Measurement sources and client-process GPU sampling are specified in Appendix I.4.

Table 38: End-to-end resource accounting: 500-round FedGMT versus its 500+200-round FedCoREx pipeline. Logical communication and recorded round wall time are summed across stages. The GPU observation is the maximum of the two stage observations within each run; run-level values are then summarized across runs.

Dataset	n	Pipeline	Logical GiB	Cumulative round time (min)	GPU obs. (MiB)
CIFAR-10	12	Base	6.884	28.61 ± 0.95	67.2 ± 0.1
CIFAR-10	12	Base + correction	7.678	41.59 ± 1.24	67.2 ± 0.1
Fashion-MNIST	12	Base	6.861	29.14 ± 0.50	67.5 ± 0.2
Fashion-MNIST	12	Base + correction	7.654	42.25 ± 1.04	67.5 ± 0.2
FEMNIST	3	Base	7.191	14.48 ± 0.20	67.5 ± 0.0
FEMNIST	3	Base + correction	8.292	21.81 ± 0.27	67.5 ± 0.0
Sentiment140	3	Base	21.306	15.74 ± 0.39	69.2 ± 0.1
Sentiment140	3	Base + correction	23.573	22.24 ± 0.41	69.2 ± 0.1

Note. CIFAR-10, Fashion-MNIST, and FEMNIST costs use separate same-workload executions; Sentiment140 costs use its performance runs. Measurement sources and client-process GPU sampling are specified in Appendix I.4.

Under this implementation, correction adds approximately 10.6–15.3% logical communication and 41.3–50.6% recorded round time relative to base training, depending on the dataset. The time increase is larger than the trainable-parameter increase because the frozen base is still evaluated and the protocol evaluates clients every round. The communication increase follows the executed payload policy described above.

I.4 Hardware and measurement scope

Experiments use one NVIDIA GeForce RTX 4070 GPU with 12 GB of device memory and 32 GB of system RAM, running Ubuntu with four client workers. The software environment is Python 3.12.8, PyTorch 2.6.0+cu126, and CUDA 12.6. Resource summaries use 50 clients and 20% participation per round.

Time and memory scope. Recorded round time covers client execution, server aggregation, and the round's validation work. Initialization, one-time context construction, report generation, and post-timer operations such as checkpoint output fall outside this sum, which therefore does not represent full command elapsed time. The GPU observation is the maximum, over client tasks, of their sampled time-averaged `torch.cuda.memory_allocated` values. Sampling uses the resource monitor's approximately 0.1-second interval and time weighting within a task. It covers monitored client-process allocations; device-wide peaks and concurrent-worker totals are outside this measurement.

Aggregation and measurement source. CIFAR-10 and Fashion-MNIST pool 12 runs each (four conditions and three seeds); each natural dataset pools three runs. Thus the visual-dataset SD also includes differences between conditions, not only seed variation within one condition. The CIFAR-10,

Fashion-MNIST, and FEMNIST cost values come from separate executions with the same architectures, round budgets, participation, and per-round validation workload. They serve as workload-level resource references; time and memory were not remeasured alongside the reported predictive results. Sentiment140 costs come from the executions underlying its reported performance.