TRUSTWORTHY PERSONALIZED FEDERATED GENERALIZATION FOR UNSEEN CLIENTS UNDER OPEN DISTRIBUTION SHIFTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated learning is valuable when data cannot be shared, yet most existing methods implicitly assume a relatively closed set of training clients and largely stationary local distributions, and often interpret deviations from the global direction uniformly as personalization signals. In open environments, the arrival of unseen clients, distribution drift, and fluctuations in update quality may occur simultaneously, coupling shared-knowledge transfer, local personalization, and the suppression of unreliable effects. To address this setting, we propose TriFed, a trustworthy personalized federated generalization framework. Based on cross-client evidence and local utility, TriFed organizes training contributions into three functional roles—shared retention, personalized reinforcement, and harmful suppression. Locally useful knowledge can therefore participate in shared learning while strengthening local discrimination, whereas sample contributions supported by negative-utility evidence are filtered through cross-round confirmation. On the server, global aggregation and cluster-level collaboration form two channels with isolated training and deployment states; at deployment, unseen clients use shared-model initialization and lightweight support-set adaptation. We evaluate seen and tail client performance, unseen-client zero-shot and adapted accuracy, open-perturbation responses, and overhead, with structural ablations of tri-decomposition and dual-channel collaboration. Ablation and gating diagnostics indicate distinct roles: tri-decomposition mainly affects robustness under open shifts and contamination, the private head primarily supports seen-client performance, and the measurable benefit of cluster-level collaboration is concentrated in adaptation initialization for unseen clients. On FEMNIST with 20% label noise, TriFed maintains both the average and bottom-10% performance of seen clients, while ranking within the top two entries of the same table for unseen-client zero-shot prediction and adaptation.

Keywords: federated learning; personalized federated learning; federated generalization; unseen clients; open distribution drift; trustworthy learning; robust aggregation

1 INTRODUCTION

Federated learning (FL) provides an important paradigm for collaborative modeling across organizations, regions, and devices without centralizing raw data McMahan et al. (2017); Li et al. (2020). However, many FL/PFL methods still adopt a relatively closed train–deployment assumption: training clients are expected to represent future deployment environments, client heterogeneity is assumed to remain broadly stable, and local deviations are treated mainly as useful personalization signals. These assumptions often fail in open federated environments. New clients that never participated in training may appear at deployment Zhong et al. (2025); local distributions can drift with time, environment, and task Peng and Tang (2025); and local updates can be affected by label noise, local overfitting, anomalous samples, or corrupted data Gholami et al. (2026); Fang et al. (2025). Open FL must therefore address three coupled questions: what knowledge should be shared across clients, what information should be retained for personalization, and what update effects should be suppressed.

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.

Personalized federated learning (PFL) alleviates the limitations of a single global model through local layers, meta-learning, adaptive aggregation, model decomposition, or personalized regularization Li et al. (2021); Luo et al. (2023); Wang et al. (2025). Most existing methods, however, optimize primarily for participating training clients and devote less attention to initialization and few-shot adaptation for future unseen clients Huang and Shu (2025). Meanwhile, non-shared residuals are often treated directly as personalized knowledge, without explicitly separating locally useful differences from unreliable deviations caused by noise, drift, or anomalies. Although federated generalization, personalization, and robust learning have each progressed substantially, a unified information-flow mechanism spanning training and open deployment remains underexplored.

We propose **TriFed**, a trustworthy personalized federated generalization framework. TriFed performs a sample-level tri-decomposition of training contributions Zeng et al. (2026): the shared component supports normal joint training, personalized ones reinforce the private head by positive local utility, and unreliable ones are filtered by cross-client negative-utility evidence. The server maintains two knowledge channels through shared-model aggregation and cluster-level private-head collaboration. Cluster candidates are isolated from the training state and only initialize unseen-client heads. When an unseen client arrives, the shared initialization supports zero-shot prediction followed by budget-matched lightweight adaptation, unifying shared transfer, personalized collaboration, and reliability control within one train–deployment pipeline. Our contributions are summarized as follows:

1. We formulate an open personalized federated setting with unseen clients, temporal drift, and heterogeneous update quality, and evaluate it through complementary views of average/tail seen-client performance, zero-shot and few-shot deployment to unseen clients, and responses to open perturbations Krouka et al. (2026).
2. We develop an evidence- and utility-driven tri-decomposition mechanism that combines shared retention, private-head reinforcement, and harmful-contribution filtering, together with a dual-channel architecture consisting of shared-model aggregation and isolated cluster-level collaboration.
3. We build a multi-seed evaluation suite covering strictly held-out testing, structural ablations, robustness to label noise, and few-shot adaptation. On FEMNIST with 20% label noise, TriFed achieves the highest average and tail accuracies for seen clients in the corresponding table, while all four unseen-client zero-shot and 15%-support metrics rank within the top two.

2 Problem Setting and Design Objectives

2.1 Open Federated Setting and Model Parameterization

During training, there are K known clients $\mathcal{C}_{\mathrm{tr}} = \{1, \ldots, K\}$. Client k holds data $\mathcal{D}_k$ generated from a local distribution P_k^r, where superscript r indicates that the distribution may vary across communication rounds. In round r, the server samples $\mathcal{P}^r \subseteq \mathcal{C}_{\mathrm{tr}}$, broadcasts shared parameters θ^r, and aggregates the returned client information. To support both shared representation learning and local adaptation, the client model is written as $f_k(x) = f(x; \theta, \phi_k)$, where θ contains a shared backbone and global head, and ϕ_k is the client-specific private head. Specifically, let $\theta = (\psi, W)$ and $z = h_\psi(x)$ denote normalized features. The shared head W and private head ϕ_k classify the same features; both include weights and biases. Data at a known client are partitioned into gradient-training, in-training probe, and final held-out subsets for parameter updates, utility estimation, and evaluation, respectively.

At deployment, unseen clients $\mathcal{C}_{\mathrm{un}} = \{K+1, \ldots, K+M\}$ that did not participate in collaborative training may arrive. The distribution P_u of an unseen client u need not match that of any training client. Its small labeled support set is further split into disjoint $\mathcal{D}_u^{\mathrm{sup,tr}}$ and $\mathcal{D}_u^{\mathrm{sup,val}}$, and the fixed query set is denoted by $\mathcal{D}_u^{\mathrm{qry}}$. Support-train is used for gradient-based head adaptation, support-validation for prespecified candidate/checkpoint selection, and the query set only for final evaluation. The open setting also allows $P_k^{r+1} \neq P_k^r$ and client-update quality to vary because of label noise, sparse data, local overfitting, or anomalous behavior.

2.2 Three Functional Roles

Shareability, personalized value, and harmfulness define three functional roles for training contributions. The shared component retains normal contributions that pass filtering; the personalized component reinforces the private head according to positive utility on local probes and yields offsets that can support collaboration among similar clients; and the unreliable component is identified jointly by cross-client evidence and cross-round confirmation and is removed from training.

2.3 Design Objectives and Evaluation Metrics

TriFed is designed and evaluated along four complementary objectives: shared transfer, personalized adaptation, tail performance, and reliability. To align the formal definitions with the accuracy metrics reported in the experiments, let $\mathcal{C}_{\text{eval}}$ denote the set of seen clients satisfying the held-out sample-count requirement, let Acc_k be the accuracy of client k on its isolated held-out set, and let $\mathcal{S} \subset \mathcal{C}_{\text{eval}}$ contain the bottom 10% of clients by held-out accuracy. For an unseen client u, a budget-constrained local adaptation operator $\mathcal{A}$ produces $W_u^{\mathcal{A}} = \mathcal{A}(W; \mathcal{D}_u^{\text{sup,tr}}, \mathcal{D}_u^{\text{sup,val}})$. The principal evaluation quantities are

$$\begin{aligned} A_{\text{tr}} &= \frac{1}{|\mathcal{C}_{\text{eval}}|} \sum_{k \in \mathcal{C}_{\text{eval}}} \text{Acc}_k, \\ A_{\text{worst}} &= \frac{1}{|\mathcal{S}|} \sum_{k \in \mathcal{S}} \text{Acc}_k, \\ A_u^{\text{adapt}} &= \frac{1}{|\mathcal{D}_u^{\text{qry}}|} \sum_{(x,y) \in \mathcal{D}_u^{\text{qry}}} \mathbb{I}\Big[\arg\max_c f_{u,c}^{\text{adapt}}(x) = y\Big], \\ A_{\text{un}} &= \mathbb{E}_{u \sim \mathcal{Q}}[A_u^{\text{adapt}}]. \end{aligned} \tag{1}$$

where $\mathcal{Q}$ denotes the distribution over unseen clients. Zero-shot accuracy is computed analogously on the same query set using the unadapted shared model. Unless otherwise stated in a table caption, $\mathcal{C}_{\text{eval}}$ contains seen clients with at least 10 held-out samples, yielding 136 clients on FEMNIST and 14 clients in the bottom 10%. In the tables, A_{tr} is abbreviated as R-tr and Worst denotes A_{worst}. Reliability is further characterized through the three routing channels, performance retention under contamination, and responses to open perturbations.

2.4 Trust Model and Privacy Boundary

Here, "trustworthy" refers to reliability of training contributions, not a formal privacy or Byzantine-security guarantee. We assume an honest-but-curious server that observes the protocol uploads, including probe features and labels; hence we do not claim server-side privacy protection. The full trust boundary and compatible privacy enhancements are detailed in Appendix A.1.

3 Trustworthy Personalized Federated Generalization

3.1 Framework Overview

TriFed organizes open federated learning into three stages: evidence-calibrated routing on clients, dual-channel collaboration on the server, and adaptation for unseen clients at deployment. At sample level, clients construct $\alpha/\beta/\gamma$: α retains contributions for normal joint training, β reinforces the private head according to positive local utility, and γ filters contributions supported by cross-client negative-utility evidence. The server aggregates models through the global shared channel and collects private-head-to-shared-head offsets through the cluster collaboration channel to form deployment candidates. An unseen client inherits the shared model and performs lightweight adaptation with a small labeled support set.

The overall architecture is shown in Fig. 1.

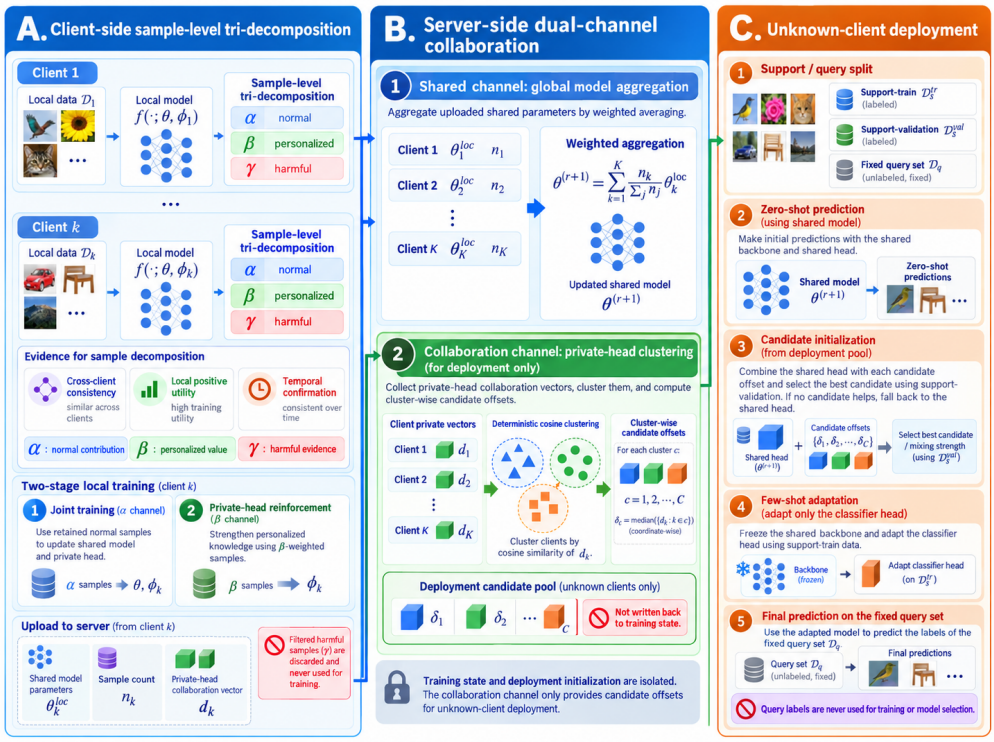

Figure 1: Overview of TriFed. **A** Client-side sample-level tri-decomposition: two mutually exclusive evidence pools provide cross-client candidate agreement, negative utility, and cross-round repetition as harmful evidence, while local probes provide personalization utility. Two-stage local training performs normal joint updates followed by private-head reinforcement. Each client uploads shared parameters θ_k^{loc}, sample count n_k, and the private-head offset d_k relative to the shared head. **B** Server-side dual channels: the shared channel aggregates the global model weighted by gradient-partition sample counts; the cluster channel deterministically clusters valid offsets by cosine similarity and forms deployment candidates using a coordinate-wise median within each cluster. Cluster candidates are isolated from the training state and do not update the next-round shared model or seen-client states. **C** Under a strict support/query split, an unseen client uses support-validation to decide whether to adopt a cluster candidate for initialization, then performs zero-shot prediction and support-set head adaptation, with final evaluation on the fixed query set.

3.2 Evidence-Calibrated Tri-Decomposition

In an open federated environment, deviation of a local update from the global direction may reflect client-specific patterns, label noise, or local overfitting rather than useful personalized knowledge. TriFed routes contributions according to their observed utility, allowing shared and local value to coexist while controlling rejection through an independent harmful-evidence path.

For sample i at client k, let γ_{ki}^r denote its rejection indicator in round r, α_{ki}^r its normal-training coefficient, and β_{ki}^r its personalization score. Omitting the within-round optimization-step index, they satisfy

$$\begin{aligned} \gamma_{ki}^r &\in \{0,1\}, \qquad \alpha_{ki}^r = 1 - \gamma_{ki}^r, \\ \beta_{ki}^r &\in [0,1], \qquad b_{ki}^r = \alpha_{ki}^r \lambda_\beta \beta_{ki}^r, \end{aligned} \tag{2}$$

where b_{ki}^r is the effective private-head reinforcement weight and λ_β controls reinforcement strength. α and γ represent retention and rejection of normal contributions, respectively, whereas β independently measures local reinforcement value.

The three roles are realized through two-stage local optimization. Let B be the original mini-batch size, and let ℓ_i^{g} and ℓ_i^{p} be the cross-entropy losses of the shared and private heads. The normal stage

jointly updates θ and ϕ_k:

$$\mathcal{L}_k^{\text{normal}} = \frac{1}{B} \sum_{i \in \mathcal{B}} \alpha_{ki}^r \big(\ell_i^{\text{g}} + \ell_i^{\text{p}}\big). \tag{3}$$

The final local representation is then frozen, and the private classifier is reinforced in the same feature coordinates:

$$\mathcal{L}_k^{\beta} = \frac{1}{B} \sum_{i \in \mathcal{B}} b_{ki}^r \ell_i^{\text{p}}. \tag{4}$$

Samples with $\gamma = 1$ have zero effective weight in both stages; samples with $\gamma = 0$ and $\beta > 0$ contribute to both normal training and private reinforcement.

3.3 Client-Side Evidence Construction and Selective Routing

Clients use cross-client evidence to estimate harmfulness and local training probes to estimate personalized value. Training probes are isolated from the gradient-training partition and inherit the observed labels of the original training data.

Cross-client harmful evidence. At the beginning of each round, training probes from other clients are split by source client into two mutually exclusive evidence pools. Probes whose shared-head prediction matches the observed label with confidence above a threshold enter the candidate pool. For sample i, feature neighbors are retrieved separately from the two pools. A cross-client candidate is formed when the neighbors in each pool cover enough source clients and consistently support the same class different from the sample's current label.

A single-sample trial update is then applied to the shared classifier according to the sample's current label, and the change in cross-entropy on the two neighbor groups is measured. Negative-utility evidence is obtained when the loss of every matched neighbor increases beyond a numerical tolerance. Cross-round confirmation is completed when the same candidate class recurs in two consecutive participations of the source client:

$$\gamma_{ki}^r = \mathbb{I}\big[E_{ki,\text{peer}}^r \wedge E_{ki,\text{harm}}^r \wedge E_{ki,\text{repeat}}^r\big]. \tag{5}$$

The three terms denote cross-client candidate agreement, negative utility of the classifier trial update, and cross-round repetition, respectively; the confirmation state is reset whenever the evidence chain is interrupted.

Local positive-utility evidence. After normal training, training samples and retained probe features are cached under the fixed local representation. For private head ϕ_k, the single-sample trial update for sample i is

$$\phi'_{k,i} = \phi_k - \eta_\beta^{\text{trial}} \nabla_{\phi_k} \ell_i^{\text{p}}, \tag{6}$$

where $\eta_\beta^{\text{trial}}$ is the private-head learning rate divided by the mini-batch size. Let μ_{ki} be the mean decrease in cross-entropy induced across probes by this trial update and s_{ki} its standard error. The personalization score is defined as

$$\beta_{ki}^r = \frac{[\mu_{ki} - s_{ki}]_+}{|\mu_{ki}| + s_{ki} + \varepsilon}. \tag{7}$$

This score combines mean benefit with variability across probes: samples that consistently improve local probes receive larger weights and reinforce the private head according to Eq. equation 4. If fewer than two probes remain, the normal training path is retained and the reinforcement stage is skipped; private-head reinforcement uses SGD on fixed features.

In implementation, rejected samples are removed before forward computation, while the loss is still normalized by the original mini-batch size B. Batch-normalization statistics are updated using only retained samples, and the optimizer update is skipped when an entire mini-batch is rejected.

3.4 Server-Side Dual-Channel Collaboration

The server accumulates cross-client knowledge through a global shared channel and forms deployment candidates through a cluster channel that summarizes each client's private-head offset relative

to the shared head. The two channels separately maintain training state and deployment-initialization state, isolating continual local learning from cross-client collaboration.

Let $\mathcal{P}^r$ be the set of participating clients in round r. Each client returns the shared parameters $\theta_k^{r,\mathrm{loc}}$ trained after sample filtering, and the server aggregates them according to the size n_k of the gradient-training partition:

$$\theta^{r+1} = \sum_{k \in \mathcal{P}^r} \frac{n_k}{\sum_{j \in \mathcal{P}^r} n_j} \theta_k^{r,\mathrm{loc}}. \tag{8}$$

This rule updates both the shared backbone and global head; the effect of α is already reflected in each locally trained model.

The cluster channel uses the offset between the post-local-training private head and the shared classifier of the current round as its collaboration carrier:

$$d_k^r = \phi_k^r - W^r, \tag{9}$$

where W^r is the shared classifier contained in θ^r at round r, and ϕ_k^r is the private head of client k after completing local training in that round. The offset is captured before the client receives any cross-client collaboration information and enters the valid pool only if it yields positive utility on the retained training probes of its source client. The server deterministically clusters valid d_k^r by cosine similarity, using direction to represent collaborative local knowledge, and retains clusters that satisfy the minimum-size requirement.

The server takes the coordinate-wise median of member offsets in each cluster to obtain a candidate

$$\widehat{d}_g^r = \underset{k \in g}{\mathrm{median}} \{d_k^r\}, \tag{10}$$

where g indexes clusters. A candidate never enters the training state and is used only as an optional initialization component for unseen-client deployment. For unseen client u, the server compares cluster candidate g and a predefined decreasing set of mixing magnitudes $\mathcal{M}$ on support-validation:

$$(g_u^\star, m_u^\star) = \arg \min_{g,\, m \in \mathcal{M} \cup \{0\}} \mathcal{L}_u^{\mathrm{sup,val}} \left(W^\star + m \widehat{d}_g^R \right), \tag{11}$$

The best candidate is accepted only if it strictly decreases the support-validation loss relative to $W^\star$; otherwise $m_u^\star = 0$. Thus,

$$W_u^{(0)} = W^\star + m_u^\star \widehat{d}_{g_u^\star}^R, \tag{12}$$

When $m_u^\star = 0$, initialization falls back to the shared head. Before final evaluation of a seen client, the shared or private head is selected according to cross-entropy on that client's gradient-training partition and then evaluated on its isolated held-out set.

3.5 Stage-Wise Deployment to Unseen Clients

Unseen-client deployment consists of zero-shot prediction followed by support-set adaptation.

The zero-shot stage directly uses the final shared backbone and global classifier:

$$f_u^{\mathrm{zero}}(x) = W^\star h_{\psi^\star}(x). \tag{13}$$

In the post-adaptation stage, $W_u^{(0)}$ from Eq. equation 12 initializes the client classifier (falling back to $W^\star$ when no candidate is selected). The backbone and feature-normalization state are frozen, and support-train is used for budget-constrained head adaptation:

$$W_u^\star = \mathcal{A}(\psi^\star, W_u^{(0)}; \mathcal{D}_u^{\mathrm{sup,tr}}), f_u^{\mathrm{adapt}}(x) \quad = W_u^\star h_{\psi^\star}(x). \tag{14}$$

The complete round-wise execution order, including confirmation-memory updates, shared aggregation, and cluster-pool maintenance, is summarized in Appendix Algorithm A1.

Table 1: Overall results on FEMNIST under clean training. Accuracy (%) is reported as mean±sample standard deviation over four seeds; Worst is the mean accuracy of the bottom 10% clients (136-client evaluation set); Time is end-to-end wall-clock time in hours.

Method	R-tr ↑	Worst ↑	ZS-Std ↑	ZS-Shift ↑	Adapt15-Std ↑	Adapt15-Shift ↑	Time ↓
TriFed	91.74±0.39	75.52±0.59	**82.97±0.30**	77.59±0.67	84.80±0.25	81.04±0.60	1.29±0.10
PGFedMo	**92.92±0.52**	**78.12±0.95**	81.36±0.91	72.30±1.67	84.70±0.65	76.84±0.78	1.04±0.02
PGFed	92.73±0.45	77.68±1.98	81.15±0.73	72.13±1.80	84.67±0.59	76.55±0.69	1.10±0.11
Ditto	92.25±0.13	74.55±0.57	81.22±0.77	**77.76±1.63**	**85.12±0.21**	**81.09±0.98**	2.61±0.05
FedAvg	84.18±0.25	62.21±0.82	81.25±1.11	77.49±1.89	84.97±0.24	80.42±0.95	0.77±0.03
FedProx	84.67±0.44	62.95±1.16	81.15±1.39	77.38±1.89	84.88±0.56	80.42±0.79	1.76±0.04
GAS+Median	82.79±0.77	60.40±1.67	79.92±1.02	77.09±1.67	84.51±0.32	78.31±1.30	0.76±0.03
FedRG	82.84±0.36	54.20±0.94	81.43±0.66	74.59±1.22	82.43±0.45	75.62±1.20	1.68±0.02

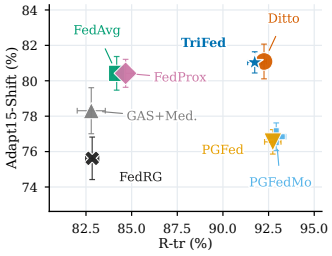

Figure 2: Seen–unseen performance trade-off on FEMNIST. Marker area denotes bottom-10% seen-client accuracy; values are four-seed means.

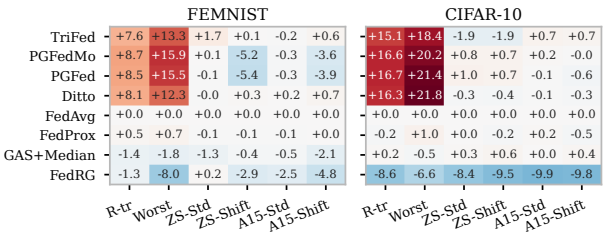

Figure 3: Metric gains over FedAvg on FEMNIST and CIFAR-10 (percentage points; four-seed means). Red/blue denotes positive/negative gain.

4 Experiments

4.1 Experimental Protocol and Overall Performance

FEMNIST is the primary environment, with writers as natural clients. We use four seeds (7/41/42/2026), 200 rounds, and 20% participation per round, with 140 seen writers and 40 disjoint unseen writers (20 Standard and 20 strong-shift); 10% of each seen client's samples form an isolated held-out set. Unseen-client support-train, support-validation, and fixed-query subsets are used only for adaptation, selection, and final evaluation, respectively. Baselines are FedAvg McMahan et al. (2017), FedProx Li et al. (2020), Ditto Li et al. (2021), PGFed/PGFedMo Luo et al. (2023), FedRG Wen et al. (2026), and GAS+Median Liu et al. (2023); all share the same splits, backbone, participation schedules, and evaluation protocol. Full partitioning, tuning, and checkpoint details are in Appendix B.

Table 1 reports clean-training results. TriFed obtains 91.74% R-tr and 75.52% bottom-10% accuracy, while three of the four unseen-client metrics rank within the top two. Figures 2 and 3 summarize the seen–unseen trade-off and gains over FedAvg; full CIFAR-10 results are in Appendix Table A11.

4.2 Tri-Channel Routing and Core Structural Ablations

We ablate tri-decomposition (*w/o tridecomp*), cluster collaboration (*w/o cluster*), and the private-head branch (*w/o personal*) under the same held-out protocol; removing the private head also removes β reinforcement and its derived cluster offsets. Figure 4 summarizes the clean-setting changes, with complete values in Appendix Table A3. Full TriFed improves ZS-Std/ZS-Shift by 1.10/2.24 points over *w/o tridecomp*, while the



Figure 4: Structural-ablation changes relative to full TriFed on clean FEMNIST (percentage points; four-seed means).

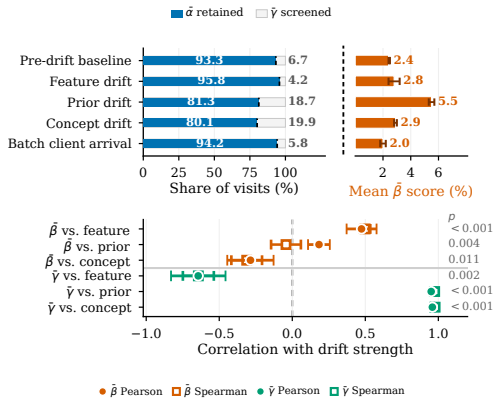

Figure 5: Gating diagnostics under open dynamic scenarios on FEMNIST. Top: retention/rejection and private-head utility; bottom: correlations with drift intensity.

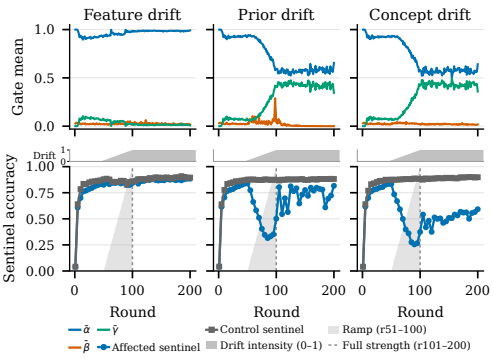

Figure 6: Round-wise gating and sentinel-accuracy responses under open dynamic scenarios on FEMNIST (four-seed means). The dashed line marks full-intensity onset.

private-head removal mainly damages seen-client R-tr/Worst (5.78/11.60 points).

4.3 Gating Behavior and Tri-Channel Diagnostics

Beyond structural ablations, we track the $\alpha/\beta/\gamma$ channels under open dynamics. α and γ are complementary retention/rejection fractions in normal training, and β is the private-head utility score. Figure 5 shows their phase summaries and drift correlations: feature drift aligns with higher $\bar{\beta}$ and lower $\bar{\gamma}$, whereas prior and concept drift primarily increase $\bar{\gamma}$; the prior-drift $\bar{\beta}$ relationship is weak and concept-drift $\bar{\beta}$ does not survive the corrected significance threshold. Figure 6 shows the corresponding round-wise trajectories and sentinel responses. Round aggregation and correlation statistics are reported in Appendix C.2.

4.4 Strict Few-Shot and Strong-Shift Protocol

We further evaluate strictly nested 5%/10%/15% support budgets on a fixed query set for Standard and strong-shift unseen clients; only the classifier head is adapted, and complete adaptation settings/results are in Appendix Table A4. TriFed is highest at 5% and 10% support in both groups (83.19/84.04% for Standard and 78.92/80.18% for Shift), reaching 84.80% and 81.04% at 15%.

4.5 Robustness in Open Environments

We inject 20% symmetric label-flip noise into seen-client training data and probes, leaving held-out and unseen-client data clean. Appendix Table A9 shows that TriFed exceeds the best competing method by 4.77 and 9.00 points in R-tr and bottom-10% accuracy, while all four unseen-client metrics rank within the top two; the 40% comparison is in Appendix Table A10. Structural ablations preserve the same division of roles at 20% noise. Figure 7 shows the support-budget/noise trends; detailed ablations, quantiles, and paired cluster gains are reported in Appendix Tables A5, A13, and A14.

4.6 Open Distribution Drift and Batched Client Arrival

We evaluate feature drift, class-prior drift, concept drift, and batched client arrival. The three drifts ramp during rounds 51–100 and persist thereafter, while previously unseen training clients join at round 101. Sentinels are prespecified and concept-drift targets are evaluated under the post-drift label rule. Full perturbation definitions and seed-level paired statistics are in Appendix Table A8.

Figure 8 summarizes the four event effects. Across all scenarios, measurable event-linked changes are confined to targeted clients; $\bar{\gamma}$ rises during label-side drift, while unaffected controls continue improving without perturbation-synchronous degradation. Detailed effect sizes and gating statistics are reported in Appendix C.2. Runtime in Table 1 is end-to-end; costs and CIFAR-10 comparisons are in the supplement.

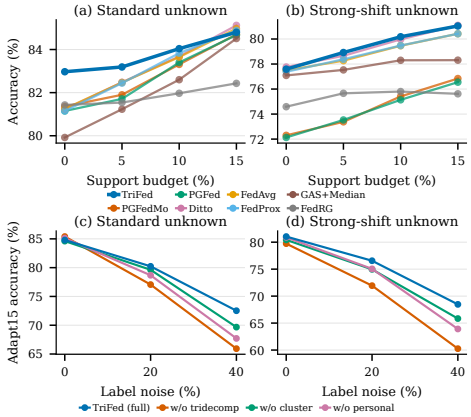

Figure 7: Unseen-client adaptation versus support budget and label noise on FEMNIST (four-seed means).

5 Related Work

Federated generalization and unseen clients. Federated domain/generalization methods address train–deployment mismatch through domain-invariant representations, ensembles, or transferability-aware adaptation Liu et al. (2025); Liao et al. (2025); Kang and Lee (2026); Wang et al. (2026); Che et al. (2025); Niu et al. (2026). FLUX and Feroma use client clustering/descriptors and profile mapping for dynamic shifts and unseen clients Fenoglio et al. (2025); Li et al. (2026), while GAINS integrates knowledge from an expanding client population Zhong et al. (2025). Complementary theory characterizes participation/generalization gaps and risk on unseen networks Wang et al. (2025); Najafi et al. (2025). These approaches emphasize domain/generalization structure rather than sample-level reliability.

Personalized FL. PFL combines shared knowledge with client-specific adaptation through local parameters, gradient control, or adaptive aggregation Liu et al. (2025); Ghoddousi Boroujeni et al. (2025); Huang and Shu (2025); Bikila and Capek (2026); Zhou et al. (2025); Jin et al. (2025); Sun et al. (2025). Foundation-model variants such as DualPFL further improve low-shot personalization through prompt parameterization Chang et al. (2025).

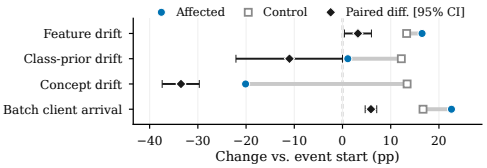

Figure 8: Sentinel-accuracy changes around open dynamic events on FEMNIST (four-seed means). Diamonds show paired differences with 95% confidence intervals.

Robust/trustworthy FL and positioning. Robust FL mitigates noisy labels, contamination, and anomalous updates through filtering, correction, and robust aggregation Gholami et al. (2026); Jiang and Zhang (2025); Pu et al. (2025); Fang et al. (2025); Wu et al. (2025); Kumar et al. (2025). TriFed combines three concerns that are usually treated separately: sample-level harmful-evidence filtering, positive-utility private-head reinforcement, and cluster-level deployment candidates isolated from training state. Its emphasis is therefore the information flow among shared, personalized, and suppressible contributions under unseen clients and drift, rather than client clustering or unseen-client mapping alone.

6 Conclusion

TriFed unifies sample-level shared retention, personalized reinforcement, and harmful suppression with a dual-channel design that isolates training state from deployment initialization. This supports zero-shot and few-shot deployment to unseen clients without writing cluster candidates back into training. Experiments indicate complementary roles: the private head mainly supports seen-client personalization, tri-decomposition improves robustness under open shifts and contamination, and cluster collaboration contributes through unseen-client initialization; CIFAR-10 provides complementary cross-environment evidence.

AI USE STATEMENT

In this work, we used generative AI tools to implement the method and to create the figures. All AI-generated content was reviewed, verified, and revised by the authors. We take responsibility for the final content of this work.

REFERENCES

H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*, PMLR 54, pp. 1273–1282, 2017.

T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated Optimization in Heterogeneous Networks," in *Proc. Machine Learning and Systems (MLSys)*, 2020.

Z. Zhong, W. Jiang, W. Bao, J. Wang, C. Wang, G. Wang, Y. Deng, and J. Ren, "Gains: Fine-grained Federated Domain Adaptation in Open Set," in *Advances in Neural Information Processing Systems*, 2025, arXiv:2510.15967.

F. Peng and M. Tang, "FedDAA: Dynamic Client Clustering for Concept Drift Adaptation in Federated Learning," arXiv:2506.21054, 2025.

S. Gholami, A. Ali, T. Haghighi, A. Arafa, and M. N. Alam, "FedSIR: Spectral Client Identification and Relabeling for Federated Learning with Noisy Labels," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition Workshops (FedVision)*, 2026, arXiv:2604.20825.

X. Fang, M. Ye, and B. Du, "Robust Asymmetric Heterogeneous Federated Learning With Corrupted Clients," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 47, no. 4, pp. 2693–2705, 2025, doi: 10.1109/TPAMI.2025.3527137.

T. Li, S. Hu, A. Beirami, and V. Smith, "Ditto: Fair and Robust Federated Learning Through Personalization," in *Proc. International Conference on Machine Learning (ICML)*, PMLR 139, pp. 6357–6368, 2021.

J. Luo, M. Mendieta, C. Chen, and S. Wu, "PGFed: Personalize Each Client's Global Objective for Federated Learning," in *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3946–3956, 2023, doi: 10.1109/ICCV51070.2023.00365.

S. Wang, J. Li, Z. Liu, Y. Zhang, and M. Gong, "FedFSL-CFRD: Personalized Federated Few-Shot Learning with Collaborative Feature Representation Disentanglement," in *Proc. AAAI Conf. Artificial Intelligence (AAAI)*, vol. 39, no. 20, pp. 21243–21251, 2025, doi: 10.1609/aaai.v39i20.35423.

G. Huang and T. Shu, "Federated Oriented Learning: A Practical One-Shot Personalized Federated Learning Framework," in *Proc. International Conference on Machine Learning (ICML)*, 2025.

D. Zeng, W. Tian, H. Wang, J. Lu, W. Xiao, and Z. Xu, "Ripple Shapley: Data Influence Attribution in One Federated Training Run," in *Proc. AAAI Conf. Artificial Intelligence (AAAI)*, vol. 40, no. 33, pp. 28085–28093, 2026, doi: 10.1609/aaai.v40i33.40034.

M. Krouka, C. B. Issaid, and M. Bennis, "Distributionally Robust Federated Learning With Client Drift Minimization," *IEEE Transactions on Machine Learning in Communications and Networking*, 2026, doi: 10.1109/TMLCN.2026.3658026.

T. Wen, Z. Yang, Y. Zhang, X. Jiang, H. Peng, Y. Wang, and B. Han, "FedRG: Unleashing the Representation Geometry for Federated Learning with Noisy Clients," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24545–24556, 2026, arXiv:2603.19722.

Y. Liu, C. Chen, L. Lyu, F. Wu, S. Wu, and G. Chen, "Byzantine-Robust Learning on Heterogeneous Data via Gradient Splitting," in *Proc. International Conference on Machine Learning (ICML)*, PMLR 202, pp. 21404–21425, 2023.

Y. Liu, Z. Qu, S. Wang, C. Shen, Y. Liang, and J. Wang, "A unified personalized federated learning framework ensuring domain generalization," *Expert Systems with Applications*, vol. 263, Art. 125700, 2025, doi: 10.1016/j.eswa.2024.125700.

T. Liao, B. Xie, L. Fu, S. Huang, B. Deng, C. Chen, and Z. Zheng, "Federated Domain Generalization with Decision Insight Matrix," in *Proc. 34th International Joint Conference on Artificial Intelligence (IJCAI)*, 2025, pp. 5689–5697, doi: 10.24963/ijcai.2025/633.

Y. H. Kang and J. H. Lee, "Federated domain generalization with source knowledge preservation via discriminative ensembles," *Information Sciences*, vol. 727, Art. 122804, 2026, doi: 10.1016/j.ins.2025.122804.

C. Wang, J. Jiang, X. Hu, X. Liu, and X. Ji, "Federated domain generalization via data-centric flatness optimization," *Pattern Recognition*, vol. 174, Art. 113023, 2026, doi: 10.1016/j.patcog.2025.113023.

H. Che, Y. Wu, H. Jin, Y. Xia, and H. Chen, "FedDAG: Federated Domain Adversarial Generation Toward Generalizable Medical Image Analysis," *IEEE Transactions on Medical Imaging*, vol. 44, no. 12, pp. 4931–4944, 2025, doi: 10.1109/TMI.2025.3584967.

K. Niu, J. Cai, W. Tai, Y. Pan, and K. Shi, "TGFed: Transferability-guided federated learning for unseen client adaptation," *Expert Systems with Applications*, vol. 299, part B, Art. 129949, 2026, doi: 10.1016/j.eswa.2025.129949.

D. Fenoglio, M. Li, P. Barbiero, N. D. Lane, M. Langheinrich, and M. Gjoreski, "FLUX: Efficient Descriptor-Driven Clustered Federated Learning under Arbitrary Distribution Shifts," in *Advances in Neural Information Processing Systems*, 2025.

M. Li, D. Fenoglio, M. Gjoreski, and M. Langheinrich, "Federated Learning with Profile Mapping under Distribution Shifts and Drifts," in *Proc. International Conference on Learning Representations (ICLR)*, 2026.

Z. Wang, C. Long, and Y. Mao, "Generalization in Federated Learning: A Conditional Mutual Information Framework," in *Proc. International Conference on Machine Learning (ICML)*, 2025.

A. Najafi, S. M. Sani, and F. Farnia, "Certifiably Robust Model Evaluation in Federated Learning under Meta-Distributional Shifts," in *Proc. 42nd International Conference on Machine Learning*, PMLR 267, pp. 45588–45623, 2025.

Y. Liu, Q. Li, J. Tan, Y. Shi, L. Shen, and X. Cao, "Understanding the Stability-based Generalization of Personalized Federated Learning," in *Proc. International Conference on Learning Representations (ICLR)*, 2025.

M. Ghoddousi Boroujeni, A. Krause, and G. Ferrari-Trecate, "Personalized Federated Learning of Probabilistic Models: A PAC-Bayesian Approach," *Transactions on Machine Learning Research*, 2025.

D. D. Bikila and J. Capek, "FedPGO: A personalized federated learning framework with gradient orthogonalization for IoT intrusion detection," *Computer Networks*, vol. 280, Art. 112210, 2026, doi: 10.1016/j.comnet.2026.112210.

X. Zhou *et al.*, "FedPDA: Personalized federated learning based on attribute similarity migration," *Information Sciences*, vol. 720, Art. 122553, 2025, doi: 10.1016/j.ins.2025.122553.

K. Jin, L. Xu, X. Wang, S.-Y. Hsieh, J. Wu, and L. Lin, "FedCPD: Personalized Federated Learning with Prototype-Enhanced Representation and Memory Distillation," in *Proc. 34th International Joint Conference on Artificial Intelligence (IJCAI)*, 2025, pp. 5498–5507, doi: 10.24963/ijcai.2025/612.

Y. Sun, A. Sun, S. Pan, Z. Fu, and J. Guo, "FedAPA: Server-side Gradient-Based Adaptive Personalized Aggregation for Federated Learning on Heterogeneous Data," in *Proc. 34th International Joint Conference on Artificial Intelligence (IJCAI)*, 2025, pp. 6219–6226, doi: 10.24963/ijcai.2025/692.

Y. Chang, X. Shi, X. Zhao, Z. Chen, and D. Ma, "Dual prompt personalized federated learning in foundation models," *Scientific Reports*, vol. 15, Art. 28026, 2025, doi: 10.1038/s41598-025-11864-4.

X. Jiang and J. Zhang, "FedClean: A General Robust Label Noise Correction for Federated Learning," in *Proc. International Conference on Machine Learning (ICML)*, 2025.

R. Pu, L. Yu, S. Zhan, G. Xu, F. Zhou, C. X. Ling, and B. Wang, "FedELR: When federated learning meets learning with noisy labels," *Neural Networks*, vol. 187, Art. 107275, 2025, doi: 10.1016/j.neunet.2025.107275.

J. Wu, P. Zhao, C. Wang, and Z. Liu, "Robust Federated Learning Under Realistic Corruption: An Iterative Filtering Approach," *IEEE Internet of Things Journal*, vol. 12, no. 9, pp. 12948–12957, 2025.

K. N. Kumar, R. R. Jha, C. K. Mohan, and R. B. Tallamraju, "Fortifying Federated Learning Towards Trustworthiness via Auditable Data Valuation and Verifiable Client Contribution," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4999–5009, 2025, doi: 10.1109/CVPR52734.2025.00471.