pFedAER: Adaptive Expert Retrieval for Personalized Federated Prompt Learning

Anonymous authors
Paper under double-blind review

Abstract

Personalized federated prompt learning (pFedPL) exploits clients' lightweight prompts as experts to exchange knowledge for performance improvement while keeping the local vision-language model (VLM) frozen. In general, the effectiveness of pFedPL hinges on how well the server routes which remote prompts are worth communicating to each client. Existing approaches rely on fixed-size retrieval of nearest-neighbor prompts based on a symmetric and time-invariant metric of prompt proximity. However, performance gain in reality is directed, evolves over the course of training, and could be negative. To bridge this gap, we formulate prompt retrieval as a problem of communication-constrained online transfer and propose an adaptive expert retrieval method (pFedAER) for personalized prompt mixture. Specifically, on the client side, each participant measures the signed loss improvement of the gated mixture over its local expert and distributes this credit across selected experts via class-conditional gate attention. On the server side, this feedback is used to estimate directed client-expert utility, to balance exploitation against uncertainty and freshness, and to adapt the number of downloaded experts under a hard communication budget. Furthermore, a persistent negative-transfer guard filters client-expert pairs that are repeatedly harmful. Evaluated on nine datasets and two backbones, pFedAER outperforms competing baselines in several settings, especially under combined feature and label shift. In addition, in a controlled study with observable transfer utilities, pFedAER reduces mean regret from 0.144 to 0.032 relative to Euclidean retrieval, lowers the rate of harmful selections from 39.2% to 8.7%, and consumes only 74.2% of the communication budget. Our implementation code will be available publicly.

1 Introduction

Federated learning (FL) coordinates model adaptation across private data silos without centralizing raw examples (Fan et al., 2025; Liu et al., 2025; Kou et al., 2025; McMahan et al., 2017; Tang et al., 2026; Kou et al., 2026). For vision-language foundation models (VLMs) (Radford et al., 2021; Jia et al., 2021), prompt learning (PL) renders this FL coordination particularly economical: clients exchange only a small set of continuous tokens while the pretrained encoders remain frozen (Zhou et al., 2022; Guo et al., 2023b; Wang et al., 2025; Feng et al., 2023; Guo et al., 2023a; Li et al., 2024). However, communication efficiency alone does not resolve statistical heterogeneity. Clients may differ in label support, image statistics, annotation practice, and sample size, so that a learned prompt by one client may benefit a second client while harming a third.

Personalized prompt-mixture methods (Luo et al., 2025; Ghiasvand et al., 2026) address this heterogeneity by treating client prompts as reusable experts. For instance, in pFedMoAP (Luo et al., 2025), each client downloads several non-local prompts and applies an image-conditioned gate to combine their class representations with those of its own prompt. This pipeline necessitates a *client-side gate* that routes each input among those downloaded prompts from the server, and a *server-side retriever* that determines which prompts become available to that client-side gate. Retrieval is therefore both a prerequisite for effective routing and a direct source of communication cost.

Learning the server-side retrieval policy is challenging because the expert-to-client transfer is *directed*, *nonstationary*, and *observable only after communication*. A remote prompt may benefit client i yet without client i reciprocating the same benefit; the utility of a prompt can evolve as both

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.

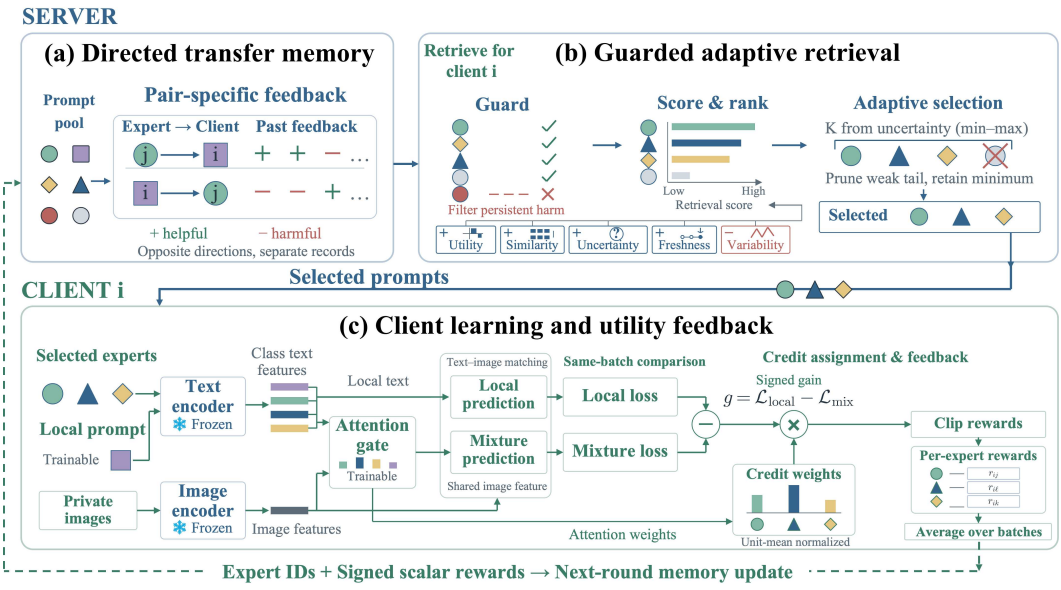

Figure 1: Illustration of pFedAER. The server learns directed client-expert utility, filters persistent negative transfer, and adapts the number of retrieved prompts. Client i gates the selected prompts and returns signed, attention-weighted utility that informs the next retrieval round.

prompts and gates are optimized; and the server can not evaluate a client's predictive loss before committing to an expert. Prior fixed-size nearest-neighbor retrieval captures none of these properties based on the measurable symmetric prompt similarity, which conflates parameter proximity with predictive benefit and saturates performance gain by the bandwidth regardless of how much evidence the server has already accumulated.

To resolve these challenges, we cast prompt retrieval as a communication-constrained online transfer loop and further propose a personalized federated adaptive expert retrieval method (pFedAER). In general, this transfer loop iterates for multiple rounds. In each round, on the one hand, a client can compare the gated mixture of experts against its local prompt on the same private mini-batch, thereby producing a *signed* measure of whether the downloaded set of experts helped. pFedAER allocates this signal among the selected experts via class-conditional gate attention and returns only aggregate scalar feedback to the server. On the other hand, the server uses this feedback to learn a directed utility memory to encode the transfer graph, combines the measured utility with contextual and uncertainty information, and adjusts the download count under a tight budget. In addition, the server embeds a persistent negative-transfer guard to filter repeatedly harmful client-expert pairs at the beginning of each round. Figure 1 summarizes the resulting retrieval and feedback loop.

To isolate whether the retrieval policy recovers the intended directed structure, we evaluate in a controlled environment with observable transfer utilities. Across 20 clients and 10 random seeds, pFedAER reduces mean regret from 0.144 to 0.032 relative to Euclidean retrieval and lowers the harmful-selection rate from 39.2% to 8.7%, while consuming 74.2% of the allowed download budget. This design tests the retrieval mechanism in isolation, free of confounding changes to the prompt architecture or the vision-language backbone.

Our contributions are threefold:

- We formulate server-side expert retrieval in personalized federated prompt learning as a directed, nonstationary online-transfer problem under per-client communication constraints, and show that symmetric prompt similarity can not represent asymmetric transfer utility.
- We propose pFedAER, which uses clipped, attention-weighted client feedback to learn ordered client-expert utilities, adapt the number of retrieval experts, and suppress persistently harmful transfers without exposing private examples and predictions.
- Extensive experiments demonstrate that pFedAER improves over baselines on six of nine datasets with the clearest gains (up to 3.57 accuracy points) under combined feature and label shift.

2 PROBLEM FORMULATION

2.1 PERSONALIZED FEDERATED PROMPT MIXTURES

Consider N clients and communication rounds $t = 1, \ldots, T$. Client i owns a private distribution $\mathcal{D}_i$ and a prompt $P_i^t \in \mathbb{R}^{m \times d}$. The server maintains the prompt pool $\mathcal{P}^t = \{P_i^t\}_{i=1}^N$ and samples an active client set $\mathcal{A}_t$ in each round. In a specific round t, before client $i \in \mathcal{A}_t$ trains, the server selects a non-local expert set $S_i^t \subseteq [N] \setminus \{i\}$ with $K_{\min} \leq |S_i^t| \leq K_{\max}$. The client i downloads $\{P_j^t : j \in S_i^t\}$ and applies a local gate that mixes the class text features of the downloaded prompts with those produced by its own prompt. A returning client restores its personalized prompt prior to retrieval, so the prompt serving as retrieval context coincides with the local expert used during training. Notably, clients without prompt history are initialized from a global aggregate.

Let $z_i^{\mathrm{loc}}(x)$ denote the logits produced by the local prompt and let $z_i^{\mathrm{mix}}(x; S)$ denote the logits after gated mixing. The set-level transfer value is

$$\Delta_i^t(S_i^t) = \mathbb{E}_{(x,y)\sim\mathcal{D}_i}\left[\ell\big(z_i^{\mathrm{loc}}(x), y\big) - \ell\big(z_i^{\mathrm{mix}}(x; S_i^t), y\big)\right], \tag{1}$$

where positive $\Delta_i^t(S_i^t)$ indicates beneficial transfer relative to local-only prediction, and in general this transfer is directed. An ideal retrieval policy π would trade this value against communication cost, and it can be formulated as

$$\max_{\pi} \sum_{t=1}^{T} \sum_{i\in\mathcal{A}_t} \mathbb{E}_{S_i^t\sim\pi}\left[\Delta_i^t(S_i^t) - \lambda_{\mathrm{c}}|S_i^t|\right] \quad \text{s.t.} \quad |S_i^t| \leq K_{\max}. \tag{2}$$

The server can not optimize Eq. 2 directly for three reasons: (i) $\mathcal{D}_i$ remains private so that Eq. 1 can not be evaluated on the server side; (ii) the directed transfer is observed only after selection; and (iii) interactions among experts render exact subset optimization combinatorial. We therefore seek an online policy that estimates directed *pairwise* utility from post-training feedback and employs it as a tractable surrogate for Eq. 2.

2.2 DIRECTED TRANSFER

Prior symmetric retrieval (e.g., Euclidean retrieval) ranks expert $j \in S_i^t$ for client i by the score $-\|P_i^t - P_j^t\|_2$. It not only mismatches parameter distance with predictive benefit, but is also structurally incapable of encoding transfer direction.

Proposition 1 (Representational mismatch of symmetric retrieval)**.** *Let $h(P_i, P_j) = h(P_j, P_i)$ be any symmetric pairwise retrieval score. Suppose there exist transfer values in a case where expert j is beneficial to client i while expert i is harmful to client j, yet h assigns identical values to both directions. Consequently, no symmetric score can, in general, encode directed transfer safety.*

Proof. Fix any two prompts and define their directed expected transfer values as $\mu_{ij} = a > 0$ and $\mu_{ji} = b < 0$. Symmetry requires $h(P_i, P_j) = h(P_j, P_i)$, whereas a direction-aware transfer policy should accept one direction and reject the other. A symmetric policy observing only h therefore can not distinguish these directed pairs without additional direction-specific information. □

The proposition does not render geometry useless. Instead, it implies that geometry can serve as a cold-start prior, while observed client-expert utility requires an explicitly directed representation.

3 PFEDAER: ADAPTIVE EXPERT RETRIEVAL

pFedAER learns which non-local prompts should be made available to each client's local gate. As illustrated in Figure 1, in each round, pFedAER alternates between server-side retrieval and client-side feedback. The server selects a bounded expert set based on prompt context and directed transfer estimates; the client then optimizes the prompt mixture and returns a single signed utility value per selected expert. This feedback informs the next round's retrieval decision without exposing local examples or predictions. The pipeline contains four components presented in Section 3.1, 3.2, 3.3, and 3.4, respectively.

3.1 SIGNED UTILITY FEEDBACK

Because the server can not observe local predictive benefit, retrieval requires a compact feedback signal computed on the client. For a mini-batch $B = \{(x_b, y_b)\}_{b=1}^{|B|}$, client i evaluates both its local prompt and the gated mixture of retrieved experts on the same examples and computes the signed loss improvement $g_i^t(B)$ using

$$g_i^t(B) = \mathcal{L}_{\mathrm{CE}}(z_i^{\mathrm{loc}}(B), y_B) - \mathcal{L}_{\mathrm{CE}}(z_i^{\mathrm{mix}}(B; S_i^t), y_B). \tag{3}$$

A positive value of $g_i^t(B)$ indicates that the downloaded mixture of retrieved experts improves over the local expert, whereas a negative value reveals harmful set-level transfer.

For client i, to allocate this set-level signal among individual experts, let $a_{ij}(x_b, y_b)$ denote the gate attention assigned to expert j for the example b. We average this attention over B and further normalize these non-local credits to unit mean. By denoting the normalized credit as $\widetilde{a}_{ij}^t(B)$, client i calculates the mini-batch-wise signed utility feedback $r_{ij}^t(B)$ using the following equation as

$$r_{ij}^t(B) = \mathrm{clip}\big(g_i^t(B)\, \widetilde{a}_{ij}^t(B), -C, C\big)\,. \tag{4}$$

The clipping operation bounds the influence of any single update and establishes a common reward scale. Finally, at the end of local training, client i averages $r_{ij}^t(B)$ across local mini-batches and releases a single scalar per selected expert.

We note that the attention-based allocation is a *policy-dependent credit estimator* rather than a causal marginal contribution, since $g_i^t(B)$ evaluates the selected set of experts as a whole. Nevertheless, it still yields signed, expert-specific feedback without the additional forward passes demanded by leave-one-out or Shapley-style attribution. This trade-off is well suited to repeated federated retrieval, where feedback must remain cheaper than the required communication decision.

3.2 DIRECTED TRANSFER ESTIMATION

Transfer from expert j to client i need not provide equal transfer in the reverse direction. We therefore associate each *ordered* pair (i, j) with a selection count n_{ij}, a reward sum q_{ij}, a squared-reward sum v_{ij}, and the last observation round ℓ_{ij}. Upon receiving reward r_{ij}, the server updates

$$n_{ij} \leftarrow n_{ij} + 1, \qquad q_{ij} \leftarrow q_{ij} + r_{ij}, \qquad v_{ij} \leftarrow v_{ij} + r_{ij}^2, \tag{5}$$

from which the empirical mean and standard deviation follow as

$$\widehat{\mu}_{ij} = q_{ij}/n_{ij}, \qquad \widehat{\sigma}_{ij} = \sqrt{\max\big(0, v_{ij}/n_{ij} - \widehat{\mu}_{ij}^2\big)}, \tag{6}$$

with zero moments for unseen pairs. The resulting $N \times N$ state is directed: statistics for $i \leftarrow j$ are not shared with $j \leftarrow i$. Because both prompts and gates evolve during training, these moments estimate utility *under the current training and retrieval policy* rather than an intrinsic, stationary expert quality.

Observed utility is sparse at the early stage of training, so expert selection additionally requires a cold-start signal and explicit exploration. To this end, we use prompt cosine similarity $c_{ij}^t = \cos(P_i^t, P_j^t)$ as context and the defined pairwise uncertainty

$$u_{ij}^t = \sqrt{\log\Big(2 + \textstyle\sum_k n_{ik}\Big)/(1 + n_{ij})}. \tag{7}$$

Following the count-based method (Auer et al., 2002), the denominator reduces the value as pair (i, j) is observed more often, while the logarithmic numerator grows the value slowly with the total feedback collected for client i, so that rarely selected pairs receive a larger bonus. Furthermore, to revisit potentially outdated transfers, we define the freshness term $f_{ij}^t = 1$ for unseen pairs and $f_{ij}^t = \exp[-\rho(t - \ell_{ij})]$ otherwise.

To facilitate the negative-transfer guard, for every transfer candidate retained by the directed utility memory, the server integrates the above considerations and computes the selection index by

$$s_{ij}^t = w_u \widehat{\mu}_{ij} + w_c c_{ij}^t + \beta u_{ij}^t + w_f f_{ij}^t - w_v \widehat{\sigma}_{ij}. \tag{8}$$

Each term plays a distinct role: $\widehat{\mu}_{ij}$ exploits measured transfer, c_{ij}^t provides a cold-start prior, u_{ij}^t encourages exploration of poorly observed pairs, and f_{ij}^t revisits stale relations, while the variance penalty favors reliable experts when mean utilities are comparable. We emphasize that Eq. 8 is an optimistic selection index for a nonstationary environment, not a classical confidence bound.

3.3 ADAPTIVE COMMUNICATION BUDGET

A fixed retrieval count can not distinguish a client whose transfer relations remain uncertain from one whose useful experts are already well established. pFedAER instead treats communication as *evidence acquisition*. Let R_i^t denote the top $B_i^t = \min(K_{\max}, |\mathcal{C}_i^t|)$ candidates under Eq. 8, where $\mathcal{C}_i^t$ is the guarded candidate set, and let $\bar{u}_i^t$ be their mean uncertainty. The request size is

$$\widetilde{K}_i^t = \left\lceil K_{\min} + (B_i^t - K_{\min}) \min\left(1, \frac{\bar{u}_i^t}{\tau_u}\right) \right\rceil. \tag{9}$$

where τ_u is an uncertainty normalization threshold that determines whether $\bar{u}_i^t$ is considered "high enough" to request the maximum number of experts. High uncertainty thus expands the request to acquire evidence, whereas accumulated observations contract it toward $K_{\min}$.

The budget $\widetilde{K}_i^t$ caps the number of downloaded experts, but does not account for the distribution of scores within the top candidates. We therefore apply a score-gap filter: from the provisional top-$\widetilde{K}_i^t$ set $\mathcal{T}_i^t$, we retain expert j only if

$$s_{i,\max}^t - s_{ij}^t \le \gamma \left(1 + \bar{u}_{i,\text{sel}}^t\right), \tag{10}$$

where $s_{i,\max}^t$ is the highest selected score and $\bar{u}_{i,\text{sel}}^t$ is the mean uncertainty over the provisional set. The base margin γ discards experts whose estimated utility is dominated by the top candidate, while the uncertainty-dependent inflation keeps pruning conservative when scores are unreliable, avoiding premature elimination of under-explored experts. Formally, the final download set is

$$\mathcal{S}_i^t = \left\{ j \in \mathcal{T}_i^t : s_{i,\max}^t - s_{ij}^t \le \gamma(1 + \bar{u}_{i,\text{sel}}^t) \right\}. \tag{11}$$

The margin prunes weak tail candidates without violating $K_{\min}$, and the final request also never exceeds $K_{\max}$.

3.4 PERSISTENT NEGATIVE-TRANSFER CONTROL

Because the expert selection policy must keep exploring under-observed pairs, it can filter out an expert whose transfer to a particular client is consistently harmful. Uncertainty-driven exploration alone does not prevent this: as long as the pair remains under-observed, its inflated score can keep the expert in the candidate pool despite accumulating negative evidence. pFedAER therefore maintains a per-pair guard that removes such experts from consideration.

Filtering rule. For each pair (i, j), the server tracks a counter c_{ij}^t of *consecutive* rounds in which the observed reward fell below a threshold τ_μ. Any reward at or above τ_μ resets the counter to zero. Expert j is excluded from client i's candidate pool at round t if both of the following hold:

$$c_{ij}^t \ge p \qquad \text{and} \qquad \widehat{\mu}_{ij}^t < \tau_\mu, \tag{12}$$

where $\widehat{\mu}_{ij}^t$ is the running mean reward. The conjunction is deliberate: the count condition guards against exclusion based on a single noisy mini-batch, while the mean condition prevents a transient outlier in training from filtering a pair whose long-run utility is positive. Neither criterion alone is sufficient, i.e., the counter can be triggered by a short burst of noise, and the mean can remain negative for pairs that have recently become useful.

Re-eligibility and Fallback. Client data distributions may drift as prompts evolve, so an exclusion that is warranted early in training may become obsolete later. A filtered pair therefore re-enters the candidate pool after a cooldown of H rounds, with its counter reset. Setting $H = 0$ disables re-eligibility and makes the exclusion persistent, which is appropriate when client distributions are stationary. In the extreme case where the guard excludes every candidate for a client, the server temporarily restores the full pool and ranks it by empirical utility $\widehat{\mu}_{ij}^t$, so that the client always receives at least $K_{\min}$ experts and optimization proceeds.

3.5 COMPLEXITY OF PFEDAER

The directed transfer requires $O(N^2)$ scalar memory, and expert retrieval costs $O(N \log K_{\max})$ per active client. Downloading a single prompt with m tokens and width d costs md parameters, so the non-local download volume is proportional to $\sum_i |S_i^t|\, md$.

Table 1: Few-shot classification accuracy (%) under disjoint-class label shift. Values are mean ± standard deviation. Bold indicates the best mean across each dataset and backbone.

Method	Flowers102	OxfordPets	Food101	Caltech101	DTD
RN50 Backbone					
ZS-CLIP	62.17±0.12	84.47±0.01	75.27±0.05	85.14±0.24	40.21±0.12
CoOp	70.14±0.76	83.21±1.30	70.43±2.42	87.37±0.44	44.23±0.63
PromptFL	72.80±1.14	90.79±0.61	77.31±1.64	89.70±1.99	54.11±0.22
PromptFL+FT	72.31±0.91	91.23±0.50	77.16±1.56	89.70±0.25	53.74±1.36
Prompt+FedPer	72.11±1.35	89.50±1.62	71.29±1.87	86.72±1.45	50.23±0.82
Prompt+FedProx	66.40±0.29	89.24±0.41	76.24±1.94	89.41±0.55	44.26±1.11
Prompt+FedAMP	69.10±0.13	80.21±0.44	74.48±1.71	87.31±1.60	47.16±0.92
pFedPrompt	86.46±0.15	91.84±0.41	92.26±1.34	96.54±1.31	77.14±0.09
FedOTP	96.23±0.44	**98.82±0.11**	92.73±0.15	97.02±0.36	87.64±0.70
pFedMoAP	98.33±0.08	98.75±0.19	92.94±0.03	97.73±0.08	90.57±0.30
pFedMMA	**99.16±0.04**	98.59±0.08	92.37±0.11	**97.94±0.20**	90.79±0.21
pFedAER (Ours)	98.86±0.05	98.50±0.23	**93.08±0.05**	97.74±0.11	**91.27±0.37**
ViT-B/16 Backbone					
pFedMoAP	99.24±0.10	**99.18±0.05**	96.31±0.03	99.02±0.04	92.49±0.33
pFedMMA	**99.53±0.10**	98.81±0.15	96.31±0.06	**99.17±0.04**	**94.02±0.29**
pFedAER (Ours)	99.51±0.00	99.10±0.04	**96.50±0.06**	99.06±0.08	93.04±0.54

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets and heterogeneity. Following pFedMoAP (Luo et al., 2025), we evaluate three settings: few-shot label shift on Flowers102 (Nilsback & Zisserman, 2008), OxfordPets (Parkhi et al., 2012), Food101 (Bossard et al., 2014), Caltech101 (Fei-Fei, 2004), and DTD (Cimpoi et al., 2014) (10 clients per dataset with disjoint classes); Dirichlet label shift on CIFAR10/100 (Krizhevsky, 2009; Krizhevsky et al., 2010) (100 clients, $\alpha = 0.5$); and combined feature and label shift on DomainNet (Peng et al., 2019) and Office-Caltech10 (Gong et al., 2012) (five clients per domain, $\alpha = 0.3$), yielding 30 and 20 clients, respectively.

Training configuration. All datasets use CLIP RN50 and ViT-B/16. Few-shot experiments use 16 examples per class. For pFedAER and pFedMoAP, the encoders are frozen and local training uses five SGD epochs. The default retrieval parameters are $K_{\max} = 9$ and $\tau_u = 0.5$. Communication schedules, optimization parameters, and baseline implementations are detailed in Appendix C.

Comparisons and metrics. We compare with pFedMoAP and pFedMMA (Luo et al., 2025; Ghiasvand et al., 2026) under both backbones, alongside the additional baselines in Tables 1–3. We report mean client accuracy, macro-F1, and 10th-percentile client accuracy (P10), averaging each run's final five rounds before computing the mean and sample standard deviation over three seeds. Communication measures are the mean delivered non-local prompts per training retrieval and aggregate prompt downloads per round (MiB). Additional diagnostics appear in Appendix D.

4.2 PERFORMANCE COMPARISON

Tables 1–3 reveal a consistent pattern: pFedAER improves over pFedMoAP on six of nine datasets with both backbones, with its clearest advantage over both methods under combined feature and label shift. We examine this pattern by heterogeneity setting to identify where feedback-guided retrieval is most effective, then assess whether it persists across both backbones and three seeds.

Few-shot label shift. With disjoint client label sets, pFedAER improves on pFedMoAP in four of five tasks under both backbones (Table 1). DTD shows the largest gains at 0.70 and 0.55 percentage points, while Food101 is the only task on which pFedAER exceeds both methods with either backbone. These results show that improvements remain possible beyond an existing personalized prompt-mixture baseline. Their magnitude depends on the task: several personalized methods already approach ceiling accuracy on Flowers102, OxfordPets, and Caltech101, and pFedAER does not dominate all comparisons. OxfordPets remains below pFedMoAP.

Table 2: Accuracy (%) under Dirichlet label shift on CIFAR10/100.

Method	CIFAR10	CIFAR100
RN50 Backbone		
ZS-CLIP	53.46±0.21	32.68±0.00
CoOp	80.84±0.39	48.74±0.17
PromptFL	73.29±0.37	45.00±0.62
Prompt+FedProx	73.32±0.34	45.63±0.75
pFedMoAP	82.30±0.69	53.53±0.38
pFedMMA	**88.48±0.38**	**61.69±0.19**
pFedAER (Ours)	80.74±0.47	49.44±0.10
ViT-B/16 Backbone		
pFedMoAP	94.53±0.20	72.70±0.18
pFedMMA	**95.93±0.14**	**78.34±0.24**
pFedAER (Ours)	94.18±0.18	70.49±0.17

Table 3: Accuracy (%) under fused feature and label shift on DomainNet and Office-Caltech10.

Method	DomainNet	Office-Caltech10
ViT-B/16 Backbone		
ZS-CLIP	9.81±0.30	9.40±0.77
CoOp	36.99±0.79	32.37±1.81
PromptFL	24.48±12.52	23.41±9.06
Prompt+FedProx	20.27±11.83	20.51±8.48
pFedMoAP	36.88±0.42	33.60±2.59
pFedMMA	36.85±0.99	27.96±1.14
pFedAER (Ours)	**38.17±0.79**	**37.17±2.53**
RN50 Backbone		
pFedMoAP	37.01±0.73	33.76±2.24
pFedMMA	37.11±0.19	29.87±2.40
pFedAER (Ours)	**37.64±0.52**	**34.82±2.05**

Tables 2 and 3 follow the reporting and highlighting conventions of Table 1. The Dirichlet concentration is 0.5 for CIFAR10/100 and 0.3 for DomainNet/Office-Caltech10. Per-domain results are provided in Appendix D.1.

Better transfer utility at higher combined heterogeneity. Viewed together, Tables 2 and 3 reveal a clear reversal. Under label shift alone, pFedAER trails pFedMoAP by 0.35–4.09 accuracy points. In this setting, prompt similarity and existing personalization mechanisms may already provide sufficiently informative expert selection. The pattern changes when feature and label shifts coexist. On DomainNet and Office-Caltech10, pFedAER ranks first in all four dataset-backbone settings, outperforming the strongest competing method by 0.53 and 1.06 points with RN50 and by 1.29 and 3.57 points with ViT-B/16, respectively. This consistent superiority across both datasets and backbones identifies the setting where adaptive expert retrieval is most effective. When feature and label shift coexist simultaneously, prompt proximity becomes a less reliable indicator of transfer quality, whereas recipient-specific feedback allows pFedAER to identify beneficial experts and suppress mismatched transfers. The comparison therefore supports a more precise conclusion than universal superiority: (i) pFedAER is particularly effective under compound client heterogeneity, where learning whom to retrieve becomes central to personalization; (ii) learning recipient-specific transfer utility is most beneficial when higher cross-client heterogeneity by combined feature and label shifts, for which fixed similarity-based retrieval is less informative.

Robustness across backbones and seeds. Backbone changes alter absolute accuracy more than the main task-level pattern. ViT-B/16 improves pFedAER on all nine datasets, yet pFedAER remains above pFedMoAP on the same six datasets, and both combined-shift tasks remain above both competitors. Across three seeds, pFedAER's few-shot accuracy standard deviation is at most 0.54 points. Office-Caltech10 is more variable, at 2.05 and 2.53 points for RN50 and ViT-B/16. The evidence is therefore strongest for a consistent direction of the mean improvements across backbones; their precise magnitude, especially on Office-Caltech10, remains sensitive to the training run.

4.3 Retrieval behavior analysis

We examine what the retrieval policy learns and how it acts on feedback. The diagnostics use 27 runs: three seeds on each dataset, with RN50 for the seven label-shift datasets and ViT-B/16 for DomainNet and Office-Caltech10. We first test if utility contains direction information beyond prompt similarity, then relate guard activity and download counts to the policy's decisions. A controlled simulation supplies complementary evidence where the underlying transfer utilities are observable.

Directed utility and prompt similarity. The utility memory captures a property that symmetric prompt proximity cannot represent: the value of transfer depends on its recipient. In DomainNet, estimates vary more across recipient domains than expert domains, and opposite directions differ by up to approximately 0.08 (Figure 2(a)–(b)). Moreover, across 81,651 selected pair observations, prompt similarity has a pooled correlation of -0.294 with released reward; 25 of 27 run-level correlations are negative. Similar prompts can therefore receive either positive or negative feedback under the learned policy. These observations motivate using geometry as context while learning recipient-specific utility from experience: the memory can distinguish observed transfers that prompt proximity treats alike.

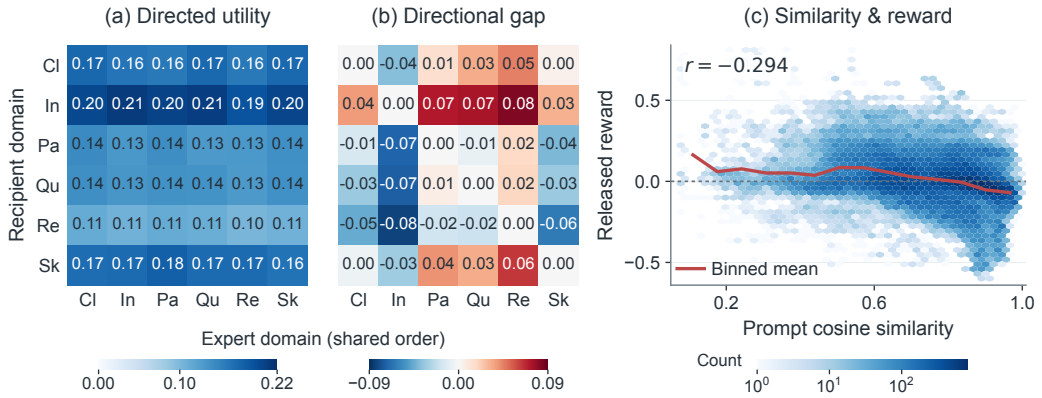

Figure 2: Observed directed utility and prompt similarity. (a) DomainNet utility estimates averaged within recipient and expert domains over three seeds. (b) Directional differences $U_{a\leftarrow b} - U_{b\leftarrow a}$, where U denotes the domain-averaged estimates in (a). Cl/In/Pa/Qu/Re/Sk denote Clipart/Infograph/Painting/Quickdraw/Real/Sketch. (c) All 81,651 selected client-expert observations; color indicates count on a log scale, and the red curve shows the binned mean released reward.

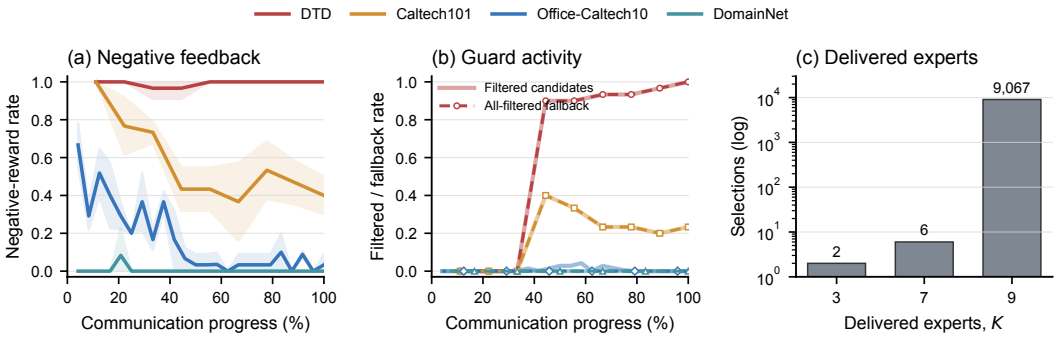

Figure 3: Feedback, guard activity, and delivered experts. (a) Negative-reward rates, where curves and shaded bands denote the mean and standard deviation over three seeds. (b) Lighter solid curves show the fraction of candidates filtered by the guard; dashed curves with open markers show all-filtered fallback rates. The redundant marker encoding keeps fallback visible when curves coincide. (c) Delivered expert counts across all 9,075 training selections, shown on a log count scale.

Feedback and guard activity. Guard activity distinguishes settings with different feedback profiles. DTD combines persistent negative rewards with increasing filtering, whereas DomainNet has predominantly positive rewards and little filtering (Figure 3(a)–(b)). The 3,279 filtered candidate incidences and 115 cooldown re-entries show that exclusion and reconsideration both occur. Fallback exposes the complementary need to maintain an available expert pool: it is invoked in 3.9% of selections overall, but in 62.6% on DTD (Table 14). Thus, filtering repeatedly harmful pairs and maintaining retrieval availability are competing requirements in the most negative-feedback setting. The guard adapts the candidate pool to observed feedback, while fallback preserves access when exclusions exhaust that pool.

Delivered retrieval budget. The default operating point emphasizes expert access: 9,067 of 9,075 training selections deliver nine experts, with only six selections delivering seven and two delivering three (Figure 3(c)). The cap is reached in 99.91% of selections. Adaptive cardinality therefore provides a control mechanism, but its activation depends on the uncertainty scale relative to τ_u. This motivates the study in Section 4.4, which tests how τ_u affects realized downloads at a fixed cap.

Controlled retrieval study. With observable transfer utilities, the controlled simulation directly evaluates whether retrieval learns to choose useful experts (Table 4). Under the same maximum budget of five, pFedAER increases selected utility from 0.059 to 0.209 and reduces harmful selections from 39.2% to 8.7% relative to Euclidean KNN. It uses only 3.710 experts on average, or 74.2% of the budget. The improvement therefore cannot be explained by downloading more experts. This synthetic experiment supports the retrieval principle in an environment where transfer quality can

Table 4: Controlled retrieval over 10 seeds. Utility is the selected true utility; harmful rate measures selections with negative utility. Budget is the fraction of the maximum download budget consumed.

Method	Utility ↑	Mean regret ↓	Harmful rate ↓	Mean K ↓	Budget used ↓
Random	-0.067 ± 0.019	0.271 ± 0.014	0.605 ± 0.041	5.000 ± 0.000	1.000 ± 0.000
Euclidean KNN	0.059 ± 0.028	0.144 ± 0.016	0.392 ± 0.070	5.000 ± 0.000	1.000 ± 0.000
pFedAER	$\mathbf{0.209} \pm 0.028$	$\mathbf{0.032} \pm 0.002$	$\mathbf{0.087} \pm 0.024$	$\mathbf{3.710} \pm 0.059$	$\mathbf{0.742} \pm 0.012$

be measured directly, complementing the task-level benefits and operating behavior observed during prompt learning. More details are in Appendix G.

Component ablations yield small, task-dependent changes at approximately matched reported download traffic. Appendix E.1 provides the full results and analysis.

4.4 Communication and sensitivity

We examine the performance–communication trade-off through two distinct controls. The hard cap $K_{\max}$ limits how many remote prompts may be downloaded; the threshold τ_u determines how much uncertainty warrants a larger request within that cap. On CIFAR10/RN50, we change one control at a time and jointly examine mean accuracy, P10, and prompt traffic (Table 5). This distinguishes an economical operating point on average from one that also preserves lower-tail performance.

Table 5: Sensitivity and communication on CIFAR10 with RN50. Each variant changes one parameter from the default ($K_{\max} = 9$, $\tau_u = 0.5$). Values are mean ± standard deviation over three seeds.

	Predictive performance (%)		Prompt downloads	
Setting	Acc. ↑	P10 ↑	Mean K ↓	MiB/round ↓
Default	80.74 ± 0.47	73.01 ± 0.19	9.00 ± 0.00	1.289 ± 0.000
Maximum retrieval budget				
$K_{\max} = 3$	80.66 ± 0.45	72.51 ± 0.65	3.00 ± 0.00	0.430 ± 0.000
$K_{\max} = 5$	80.67 ± 0.46	72.67 ± 0.35	5.00 ± 0.00	0.716 ± 0.000
$K_{\max} = 10$	81.00 ± 0.47	72.91 ± 0.30	10.00 ± 0.00	1.432 ± 0.000
$K_{\max} = 20$	80.96 ± 0.55	73.03 ± 0.49	19.99 ± 0.00	2.863 ± 0.001
Uncertainty target				
$\tau_u = 0.25$	80.74 ± 0.47	73.01 ± 0.19	9.00 ± 0.00	1.289 ± 0.000
$\tau_u = 1.0$	80.89 ± 0.54	72.93 ± 0.37	8.91 ± 0.00	1.276 ± 0.000
$\tau_u = 2.0$	80.93 ± 0.57	72.77 ± 0.50	8.28 ± 0.01	1.185 ± 0.001

Maximum retrieval budget. The default setting leaves large room to reduce downloads at similar accuracy. Reducing $K_{\max}$ from 9 to 3 cuts prompt traffic by 66.6%, while accuracy changes from 80.74% to 80.66%. $K_{\max}$ of 5 reduces traffic by 44.5% with a 0.07-point accuracy decrease. P10 falls 0.50 and 0.34 points, respectively, so these choices trade a small accuracy change for a larger communication change. Furthermore, increasing $K_{\max}$ from 10 to 20 doubles traffic without accuracy gain. In this setting, the value of additional downloaded experts depends on what the budget is intended to serve.

Uncertainty threshold. The threshold changes realized communication without changing the hard cap. At $K_{\max} = 9$, raising τ_u to 2.0 lowers mean delivered K to 8.28 and traffic by 8.1%. Mean accuracy rises by 0.19 points, while P10 declines by 0.24 points. The gain in mean accuracy therefore does not imply a uniform improvement for all clients. Together with the cap sweep, this result separates two decisions: how much communication is allowed, and how much the current uncertainty justifies using. Both controls should be evaluated against average and lower-tail performance.

Appendix F.3 presents participation, feedback-noise, delay, expert-dropout, and sign-flipping tests. Per-dataset download and runtime measurements are provided in Appendix D.

5 Conclusion

We presented pFedAER, which learns directed expert retrieval from signed client feedback for personalized federated prompt learning. Its strongest empirical gains occur when feature and label shifts coexist, with improvements across both backbones and in lower-tail performance relative to pFedMoAP. The retrieval diagnostics show why prompt similarity is an incomplete proxy for observed transfer, while controlled experiments demonstrate better expert selection under a common maximum budget. Communication studies further identify settings with substantially fewer downloads and similar mean accuracy. Together, these results establish an empirical case for learning which prompts to communicate, alongside learning how to combine them locally, and for treating expert availability as part of personalization.

AI USE STATEMENT

In this work, we used generative AI tools for language editing and writing assistance during manuscript preparation, including enhancing the phrasing, clarity, and organization of author-written text. We have not used generative AI tools for the intellectual and technical content, including the problem formulation, methods, theoretical results, experiments, and analyses. Other tasks with required disclosure are not applicable to this work. Additionally, we used generative AI tools for improving the readability and organization of author-written text. We have reviewed all AI-assisted work. For example, "all AI-assisted text was subsequently reviewed, verified, and revised by the authors to ensure accuracy and appropriateness". We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

The experiments use public image-classification datasets and involve no human subjects or newly collected personal data. Raw images remain local, but prompts, expert identities, and utility feedback may reveal client properties, and the optional noise mechanism does not provide differential privacy. The guard may also distribute access to shared expertise unevenly, so deployment should review dataset licenses and audit privacy and client-level fairness under an explicit threat model.

REPRODUCIBILITY STATEMENT

Section 4 specifies the datasets, client partitions, training protocol, baselines, metrics, and seed aggregation. Appendix B and Appendices C–C.3 provide the algorithm, implementation details, hyperparameters, and metric definitions. The remaining appendices report detailed results and reproduction commands. The code includes the implementations, experiment launchers, summarization and figure scripts, and focused tests. Each run emits manifests, round metrics, retrieval audits, and resumable state for traceability.

REFERENCES

Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256, 2002.

Omar Besbes, Yonatan Gur, and Assaf Zeevi. Stochastic multi-armed-bandit problem with non-stationary rewards. In *Advances in Neural Information Processing Systems*, volume 27, 2014.

Weihao Bo, Yanpeng Sun, Yu Wang, Xinyu Zhang, and Zechao Li. FedMGP: Personalized federated learning with multi-group text-visual prompts. In *Advances in Neural Information Processing Systems*, 2025.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101: Mining discriminative components with random forests. In *European Conference on Computer Vision*, pp. 446–461, 2014.

Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit: General framework and applications. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 151–159. PMLR, 2013.

Gary Cheng, Karan Chadha, and John Duchi. Fine-tuning is fine in federated learning. *arXiv preprint arXiv:2108.07313*, 2021.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3606–3613, 2014.

Tianyu Cui, Hongxia Li, Jingya Wang, and Ye Shi. Harmonizing generalization and personalization in federated prompt learning. In *International Conference on Machine Learning*, 2024.

Mingyuan Fan, Zhanyi Hu, Fuyi Wang, and Cen Chen. Bad-pfl: Exploiting backdoor attacks against personalized federated learning. In *International Conference on Learning Representations*, volume 2025, pp. 39314–39341, 2025.

Li Fei-Fei. Learning generative visual models from few training examples. In *Workshop on Generative-Model Based Vision, IEEE Proc. CVPR, 2004*, 2004.

Chun-Mei Feng, Bangjun Li, Xinxing Xu, Yong Liu, Huazhu Fu, and Wangmeng Zuo. Learning federated visual prompt in null space for mri reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8064–8073, June 2023.

Sajjad Ghiasvand, Mahnoosh Alizadeh, and Ramtin Pedarsani. pfedmma: Personalized federated fine-tuning with multi-modal adapter for vision-language models. In *International Conference on Learning Representations*, volume 2026, pp. 76213–76240, 2026.

Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2066–2073, 2012.

Shuai Gong, Chaoran Cui, Xiaolin Dong, Xiushan Nie, Lei Zhu, and Xiaojun Chang. Token-level prompt mixture with parameter-free routing for federated domain generalization. *IEEE Transactions on Image Processing*, 35:656–669, 2026.

Tao Guo, Song Guo, and Junxiao Wang. pFedPrompt: Learning personalized prompt for vision-language models in federated learning. In *The Web Conference*, pp. 1364–1374, 2023a.

Tao Guo, Song Guo, Junxiao Wang, Xueyang Tang, and Wenchao Xu. PromptFL: Let federated participants cooperatively learn prompts instead of models—federated learning in age of foundation model. *IEEE Transactions on Mobile Computing*, 2023b.

Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang. Personalized cross-silo federated learning on non-iid data. In *AAAI Conference on Artificial Intelligence*, volume 35, pp. 7865–7873, 2021.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pp. 4904–4916. PMLR, 2021.

Wei-Bin Kou, Qingfeng Lin, Ming Tang, Rongguang Ye, Shuai Wang, Guangxu Zhu, and Yik-Chung Wu. Fast-convergent and communication-alleviated heterogeneous hierarchical federated learning in autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 26(7): 10496–10511, 2025.

Wei-Bin Kou, Guangxu Zhu, Ming Tang, Chen Zhang, Lisheng Wu, Lei Zhou, and Yujiu Yang. Omniisr: A unified framework for centralized and federated learning via intermediate supervision and regularization. *arXiv preprint arXiv:2605.20276*, 2026.

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

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. CIFAR-10 and CIFAR-100 Datasets, 2010.

Fan Lai, Xiangfeng Zhu, Harsha V Madhyastha, and Mosharaf Chowdhury. Oort: Efficient federated learning via guided participant selection. In *15th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 21)*, pp. 19–35, 2021.

Hongxia Li, Wei Huang, Jingya Wang, and Ye Shi. Global and local prompts cooperation via optimal transport for federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12151–12161, June 2024.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pp. 661–670, 2010.

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

Yingqi Liu, Qinglun Li, Jie Tan, Yifan Shi, Li Shen, and Xiaochun Cao. Understanding the stability-based generalization of personalized federated learning. In *International Conference on Learning Representations*, volume 2025, pp. 27418–27449, 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*, 2025.

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

H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *International Conference on Learning Representations*, 2018.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, pp. 722–729, 2008.

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3498–3505, 2012.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *IEEE International Conference on Computer Vision*, pp. 1406–1415, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PmLR, 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. *arXiv preprint arXiv:1701.06538*, 2017.

Mainak Singha, Subhankar Roy, Sarthak Mehrotra, Ankit Jha, Moloud Abdar, Biplab Banerjee, and Elisa Ricci. FedMVP: Federated multimodal visual prompt tuning for vision-language models. In *IEEE/CVF International Conference on Computer Vision*, pp. 17869–17878, 2025.

Jianheng Tang, Yajiang Huang, Kejia Fan, Feijiang Han, Jiaxu Li, Jinfeng Xu, Run He, Anfeng Liu, Houbing Song, Huiping Zhuang, and Yunhuai Liu. Deepafl: Deep analytic federated learning. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 60290–60335, 2026.

Linh Tran, Wei Sun, Stacy Patterson, and Ana Milanova. Privacy-preserving personalized federated prompt learning for multimodal large language models. In *International Conference on Learning Representations*, volume 2025, pp. 27362–27380, 2025.

Zhihao Wang, Wenke Huang, Tian Chen, Zekun Shi, Guancheng Wan, Yu Qiao, Bin Yang, Jian Wang, Bing Li, and Mang Ye. An empirical study of federated prompt learning for vision language model. *arXiv preprint arXiv:2505.23024*, 2025.

Zirui Wang, Zihang Dai, Barnabás Póczos, and Jaime Carbonell. Characterizing and avoiding negative transfer. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11285–11294. IEEE, 2019.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130:2337–2348, 2022.

APPENDIX CONTENTS

A Related Work . . . 13
B Method Details and Algorithm . . . 14
C Experimental Setup and Implementation . . . 15
D Additional Predictive Results . . . 17
E Retrieval Diagnostics and Component Ablations . . . 20
F Communication, System Cost, and Robustness . . . 21
G Controlled Synthetic Study . . . 24

A RELATED WORK

Personalized federated prompt learning. Federated prompt learning reduces communication by optimizing compact prompts rather than full foundation models. PromptFL established this parameter-efficient adaptation paradigm (Guo et al., 2023b), and FedOTP and FedPGP subsequently introduced explicit mechanisms for balancing shared and personalized prompt knowledge (Li et al., 2024; Cui et al., 2024). More recent methods enrich the adapted representation through image- and text-conditioned visual prompts, paired multimodal prompt groups, or modality-specific adapters (Singha et al., 2025; Bo et al., 2025; Ghiasvand et al., 2026), with the effectiveness of each choice depending on the form of client heterogeneity (Wang et al., 2025). This work primarily concerns *what* prompt representation to learn and *how* to aggregate it; our focus is instead the preceding communication decision of *which* client prompts to expose to each personalized learner.

Prompt mixtures, routing, and retrieval. Sparse mixture-of-experts models route each input among a subset of available experts (Shazeer et al., 2017). The same principle underlies federated prompt learning: pFedMoAP applies image-conditioned attention over a local prompt and retrieved client prompts (Luo et al., 2025), and TRIP performs parameter-free token routing for federated domain generalization (Gong et al., 2026). These routers presuppose that the candidate experts are already locally accessible. Server-side retrieval operates under a fundamentally different information set and cost structure: it must select prompts *before* observing the private minibatch, and every selected expert incurs communication. pFedAER complements local routing by learning this communication-constrained availability layer.

Online expert selection and private feedback. Classical upper-confidence methods balance exploration and exploitation under stationary rewards, while contextual bandits incorporate side information into sequential decisions (Li et al., 2010). Nonstationary extensions use discounting, sliding windows, or variation budgets to track changing arm values (Besbes et al., 2014). Related work on combinatorial bandits and bandits with knapsacks studies set-valued actions under structured rewards or resource constraints (Chen et al., 2013), while systems such as Oort use utility-aware participant selection to improve federated training efficiency (Lai et al., 2021). pFedAER shares sequential partial feedback and budgeted selection with these lines of work, but each action exposes a recipient-specific set of co-adapting prompt experts whose rewards are observed only through the selected mixture. This setting also raises negative-transfer concerns: prior work formalizes when source knowledge can harm target performance and filters harmful sources (Wang et al., 2019). Finally, private federated optimization provides client- or user-level differential privacy through explicit noise calibration and accounting (McMahan et al., 2018), and DP-FPL extends formal privacy to personalized federated prompt learning (Tran et al., 2025). By contrast, pFedAER communicates clipped aggregate utilities and permits optional perturbation, but does not claim differential privacy without a corresponding privacy accountant.

B METHOD DETAILS AND ALGORITHM

Algorithm 1 summarizes the interaction between server-side retrieval and client-side prompt learning. A returning client restores its personalized prompt and gate before retrieval, so the retrieval context and the local expert used to measure transfer refer to the same client state. Clients without prompt history first train from the global prompt without remote experts. The retrieval bounds apply once trained candidates are available; this initialization step uses an empty expert set.

Algorithm 1 pFedAER training and adaptive expert retrieval

Require: rounds T; prompt pool $\mathcal{P}$; bounds $K_{\min}, K_{\max}$; retrieval and guard parameters
Initialize global prompt P_{g} and ordered-pair statistics; mark client prompts and gates as unseen
for $t = 0, \ldots, T-1$ **do**
  Sample active clients $\mathcal{A}_t$
  for all $i \in \mathcal{A}_t$ **do**
    Clear cached non-local context and reset trainable state
    if $t = 0$ or client i has no prompt history **then**
      Initialize $P_i^t \leftarrow P_{\mathrm{g}}$ and $S_i^t \leftarrow \varnothing$
    else
      Restore client i's personalized prompt and gate
      Form the pool of trained non-local prompts and apply Eq. 12 with cooldown H
      Compute pairwise moments and uncertainties by Eqs. 6–7
      if all available candidates are guarded **then**
        Restore the candidate pool and rank by empirical utility $\widehat{\mu}_{ij}^t$
      else
        Rank eligible candidates by s_{ij}^t in Eq. 8
      end if
      Form S_i^t using the adaptive count and score margin in Eqs. 9–10
      Download selected prompts and cache their text features
    end if
    for all private minibatches B **do**
      Compute local and mixture logits on the same examples
      Compute $g_i^t(B)$ and accumulate clipped, attention-weighted rewards by Eqs. 3–4
      Update the local prompt and gate using the mixture loss
    end for
    Average rewards across minibatches and clip the released values
    Update ordered state (i, j) and its negative-reward streak for each released r_{ij}^t
    Store client i's prompt and, when experts were used, its gate
  end for
  Aggregate active prompts into P_{g} and evaluate all clients
end for

Credit assignment. For each minibatch, the client averages the gate attention assigned to each selected non-local expert at the ground-truth class and normalizes these credits to a unit mean across the selected experts. The resulting credit multiplies the shared loss improvement in Eq. 4. Clipping is applied per minibatch before the rewards are averaged for release. This construction supplies expert-specific feedback using the logits already computed during training. It measures each expert's attention-weighted share of the selected mixture's gain; it does not identify an expert's causal marginal contribution.

Guard and evaluation behavior. The guard combines a negative-reward streak with the running mean condition in Eq. 12. Cooldown permits reconsideration, while the all-filtered fallback restores candidates and uses empirical utility for ranking. Evaluation follows the retrieval rule without updating transfer memory. Algorithm 1 describes the default setting with immediate feedback and available experts. The perturbation, delay, and delivery variants are evaluated separately in Appendix F.3.

The server receives one aggregate scalar per selected expert. It does not receive private examples, predictions, or minibatch losses, and it maintains separate statistics for $i \leftarrow j$ and $j \leftarrow i$.

C Experimental Setup and Implementation

This section specifies the data partitions, training configuration, and reporting conventions used in Section 4. The comparison covers three forms of heterogeneity, with both CLIP RN50 and ViT-B/16 evaluated on every dataset.

C.1 Data partitions and training configuration

Table 6 summarizes the federated schedules. The five few-shot datasets allocate disjoint classes to 10 clients and use 16 training examples per class. CIFAR10/100 use 100 clients with $\mathrm{Dir}(0.5)$ label partitions. DomainNet and Office-Caltech10 combine visual domains with $\mathrm{Dir}(0.3)$ label partitions, assigning five clients to each domain. These settings follow Luo et al. (2025).

Table 6: Federated data and training schedules. Participation is the fraction of clients selected per round. All settings use five local epochs and are evaluated with both RN50 and ViT-B/16.

Datasets	Heterogeneity	Clients	Participation	Rounds
Five few-shot datasets	Disjoint classes	10	1.0	10
CIFAR10/100	$\mathrm{Dir}(0.5)$	100	0.1	120
DomainNet	Domain + $\mathrm{Dir}(0.3)$	30	0.25	25
Office-Caltech10	Domain + $\mathrm{Dir}(0.3)$	20	0.5	25

For pFedAER and pFedMoAP, both CLIP encoders remain frozen. Local optimization uses SGD with prompt learning rate 0.002; the attention gate uses eight heads and learning rate 0.01. Both methods use local-logit coefficient $\lambda = 0.5$. For each method, the two backbone evaluations retain the same data splits, client schedules, optimization settings, and random seeds.

The retrieval diagnostics and per-dataset system measurements use RN50 on the five few-shot datasets and CIFAR10/100, and ViT-B/16 on DomainNet and Office-Caltech10. We refer to these as the reference-backbone configurations; they comprise 27 runs across nine datasets and three seeds. Component ablations use CIFAR10/RN50 and DomainNet/ViT-B/16, while budget sensitivity and stress tests use CIFAR10/RN50.

C.2 Retrieval hyperparameters

Table 7 lists the retriever defaults. $K_{\max}$ counts non-local prompts, so the local expert is additional to this budget. The main comparison uses $K_{\max} = 9$ and $\tau_u = 0.5$. The sensitivity and ablation studies identify their changes relative to these defaults.

Table 7: Default retrieval parameters, using the notation of the main text.

Parameter	Symbol	Default
Mean-utility weight	w_u	1.0
Prompt-context weight	w_c	0.25
Exploration coefficient	β	0.5
Freshness weight	w_f	0.05
Variance penalty	w_v	0.1
Freshness decay	ρ	0.05
Uncertainty threshold	τ_u	0.5
Score margin	γ	0.5
Non-local retrieval bounds	$(K_{\min}, K_{\max})$	$(1, 9)$
Negative-reward threshold	τ_μ	-0.01
Guard patience	p	3
Guard cooldown (rounds)	H	5
Reward clipping bound	C	1.0
Feedback noise standard deviation	–	0

C.3 METRICS AND AGGREGATION

Predictive metrics are mean client accuracy, macro-F1, and 10th-percentile client accuracy (P10). Each run averages its round-level metrics over the final five communication rounds. We then report the mean and sample standard deviation across three seeds. Dataset-macro summaries first average the datasets within each seed, giving every dataset equal weight.

The client-distribution figure uses a related but distinct aggregation: each client's accuracy is averaged over the final five rounds, and these client endpoints are pooled across seeds.

Communication is measured as delivered non-local prompts per training retrieval and aggregate downloaded prompt data per round in MiB (2^{20} bytes). The latter measures prompt downloads; runtime and persistent retrieval-state size are reported separately. The negative-reward rate is the fraction of released rewards below zero, and the fallback rate is the fraction of training selections that restore an all-filtered pool. These diagnostics describe retrieval behavior; P10 describes the lower end of the client accuracy distribution.

C.4 BASELINE IMPLEMENTATIONS

Result sources. The pFedMoAP, pFedMMA, and pFedAER entries in Tables 1–3 are our evaluations. The other available entries, including zero-shot CLIP (Radford et al., 2021), CoOp (Zhou et al., 2022), PromptFL (Guo et al., 2023b), PromptFL+FT (Cheng et al., 2021), Prompt+FedPer (Arivazhagan et al., 2019), Prompt+FedProx (Li et al., 2020), Prompt+FedAMP (Huang et al., 2021), pFedPrompt (Guo et al., 2023a), and FedOTP (Li et al., 2024), are published reference values from Luo et al. (2025). Those references use RN50 for the few-shot and CIFAR tasks and ViT-B/16 for the combined-shift tasks. The per-domain tables mark this distinction with a separator.

pFedMoAP. Our implementation uses nearest-prompt retrieval and a local attention gate over the prompt pool, with $\lambda = 0.5$ under both backbones. The comparison evaluates the complete methods under the specified data and training schedules, and it does not isolate retrieval from every client-state and implementation difference.

pFedMMA. We implement pFedMMA from its published description (Ghiasvand et al., 2026). For ViT-B/16, parallel adapters are inserted into blocks 5–12 of both CLIP transformers. For RN50, the text-transformer indices remain 5–12, while the visual indices refer to the flattened bottleneck sequence. With $(3, 4, 6, 3)$ blocks across the four RN50 stages, this covers the last three blocks of stage 2 and the first five of stage 3. Each visual adapter consumes the bottleneck input; at a stage transition, it uses the frozen downsampled identity to remain parallel to the backbone branch.

The adapters use modality-specific down- and up-projections, a shared middle projection of width 32, and residual scale 10^{-3}. The down/up projections remain private, while shared projections are aggregated in proportion to active clients' sample counts. Only the shared projections are communicated. We use the class template "a photo of a [class]", reset the optimizer and scheduler state between clients, and retain the schedules in Table 6 with learning rate 0.002. Training uses automatic mixed precision with dynamic loss scaling, which skips optimizer updates with non-finite gradients. There were 32 non-finite minibatch losses across three runs; all round-level evaluation endpoints were finite, and all three prespecified seeds were retained for each setting.

C.5 LOCAL-LOGIT COEFFICIENT IN PFEDMOAP

The local-logit coefficient materially affects the strength of the pFedMoAP baseline. Table 8 compares $\lambda = 0$ and $\lambda = 0.5$ on the five few-shot datasets with RN50. The latter improves all three metrics on every dataset, with average gains of 4.22 accuracy points, 4.48 macro-F1 points, and 5.70 P10 points. Food101 has the largest accuracy and macro-F1 gains, while OxfordPets has the largest P10 gain. The main dual-backbone comparison uses $\lambda = 0.5$ for pFedMoAP; these results show why the coefficient matters when interpreting differences between prompt-mixture implementations.

Table 8: RN50 pFedMoAP sensitivity to the local-logit coefficient. All values use the final-five-round, three-seed reporting convention.

	$\lambda = 0$			$\lambda = 0.5$		
Dataset	Acc. ↑	F1 ↑	P10 ↑	Acc. ↑	F1 ↑	P10 ↑
Flowers102	97.00±0.14	96.74±0.17	94.95±0.30	98.33±0.08	98.16±0.14	96.89±0.36
OxfordPets	93.97±0.56	93.85±0.62	87.14±1.14	98.75±0.19	98.75±0.20	96.22±0.78
Food101	86.24±0.12	86.19±0.13	82.24±0.40	92.94±0.03	92.97±0.04	89.97±0.20
Caltech101	94.74±0.19	93.09±0.16	91.40±0.13	97.73±0.08	97.01±0.14	95.24±0.13
DTD	85.26±0.27	85.14±0.26	76.53±0.31	90.57±0.30	90.53±0.30	82.44±0.06

D Additional Predictive Results

We examine where the accuracy gains in Section 4.2 occur by resolving performance across domains, classes, and clients. The results use the final-five-round, three-seed reporting convention in Appendix C.3. Per-domain means explain the combined-shift comparison, while macro-F1, P10, and client distributions assess how broadly the improvements are shared.

D.1 Per-domain accuracy under combined shift

Tables 9 and 10 locate the combined-shift gains within individual domains. They report both backbones for our runs and retain the published reference values from Luo et al. (2025). Their domain averages match Table 3.

On DomainNet with ViT-B/16, pFedAER improves over pFedMoAP in five of six domains, with the largest gains on Quickdraw (2.75 points), Clipart (2.21), and Painting (2.07). On Office-Caltech10, the ViT-B/16 gains span all four domains and are largest on Webcam (7.22) and DSLR (4.69). RN50 also improves the Office-Caltech10 average, but its gains are concentrated in DSLR and Webcam. The aggregate advantage therefore reflects different domain-level responses, with broader improvements under ViT-B/16.

Table 9: DomainNet accuracy under feature and $\mathrm{Dir}(0.3)$ label shift. Rows above the separator are published. The remaining rows are our evaluations; ViT-B/16 follows the original protocol, while RN50 measures backbone sensitivity. Both pFedMoAP rows use $\lambda = 0.5$. Bold marks the best mean among pFedMoAP, pFedMMA, and pFedAER within each backbone; shaded rows denote pFedAER.

Method	Clipart	Infograph	Painting	Quickdraw	Real	Sketch	Avg.
ZS-CLIP	9.18±0.62	10.03±0.16	9.93±0.51	10.25±0.40	9.90±1.30	9.54±1.13	9.81±0.30
CoOp	43.84±3.51	45.72±0.85	29.94±0.46	36.83±1.17	31.64±0.49	33.97±0.78	36.99±0.79
PromptFL	27.63±16.41	27.69±18.07	21.62±8.34	23.45±13.49	20.62±11.03	25.90±8.10	24.48±12.52
Prompt+FedProx	22.23±15.42	21.75±17.00	18.58±8.15	19.40±12.59	17.17±10.25	22.49±8.44	20.27±11.83
pFedMoAP (RN50)	43.16±1.78	46.74±0.80	**30.28**±2.86	37.17±0.60	30.93±0.92	33.78±4.83	37.01±0.73
pFedMoAP (ViT-B/16)	43.70±0.15	**46.54**±0.60	29.61±0.58	34.50±1.88	30.74±1.34	36.20±0.68	36.88±0.42
pFedMMA (RN50)	40.17±0.59	**46.89**±0.83	**30.28**±1.34	**37.50**±1.64	31.63±1.12	36.20±0.36	37.11±0.19
pFedMMA (ViT-B/16)	43.80±1.82	45.72±1.74	29.69±0.80	**37.57**±0.99	28.61±1.59	35.69±1.77	36.85±0.99
pFedAER (RN50, ours)	**45.83**±0.67	46.18±0.50	30.17±1.19	35.31±0.20	**32.09**±0.29	**36.27**±0.89	**37.64**±0.52
pFedAER (ViT-B/16, ours)	**45.91**±0.74	46.50±2.08	**31.68**±0.34	37.25±1.79	**30.83**±1.37	**36.84**±0.31	**38.17**±0.79

Table 10: Office-Caltech10 accuracy under feature and $\mathrm{Dir}(0.3)$ label shift. Rows above the separator are published. The remaining rows are our evaluations; ViT-B/16 follows the original protocol, while RN50 measures backbone sensitivity. Both pFedMoAP rows use $\lambda = 0.5$. Bold marks the best mean among pFedMoAP, pFedMMA, and pFedAER within each backbone; shaded rows denote pFedAER.

Method	Amazon	Caltech	DSLR	Webcam	Avg.
ZS-CLIP	9.83±1.63	10.67±0.89	10.89±1.40	6.20±3.84	9.40±0.77
CoOp	30.29±3.64	35.88±1.30	29.89±5.15	33.43±2.25	32.37±1.81
PromptFL	21.08±9.60	23.72±12.21	22.94±7.96	25.88±7.72	23.41±9.06
Prompt+FedProx	18.64±8.58	19.56±11.59	20.89±7.38	22.96±7.56	20.51±8.48
pFedMoAP (RN50)	**33.34**±1.79	**37.52**±1.46	37.44±4.93	26.75±2.47	33.76±2.24
pFedMoAP (ViT-B/16)	31.18±1.95	35.27±1.43	38.18±5.34	29.76±4.63	33.60±2.59
pFedMMA (RN50)	29.96±3.33	35.14±1.12	28.00±8.55	26.36±1.38	29.87±2.40
pFedMMA (ViT-B/16)	25.98±1.20	31.89±2.75	28.36±1.67	25.59±1.10	27.96±1.14
pFedAER (RN50, ours)	30.95±1.97	36.88±3.68	**38.62**±5.11	**32.83**±4.37	**34.82**±2.05
pFedAER (ViT-B/16, ours)	**32.04**±1.20	**36.78**±1.57	**42.87**±8.89	**36.98**±2.31	**37.17**±2.53

D.2 Class-wise and lower-tail performance

The combined-shift advantage extends beyond mean accuracy. Tables 11 and 12 show that pFedAER improves both macro-F1 and P10 over pFedMoAP on DomainNet and Office-Caltech10 with either backbone. On Office-Caltech10, the macro-F1 gains are 2.78 points with RN50 and 4.81 with ViT-B/16, while P10 improves by 4.95 and 4.74 points. These results connect the main accuracy gains to more balanced class-wise performance and stronger performance among clients near the lower end of the accuracy distribution.

The pattern remains task-dependent. pFedAER trails pFedMoAP on both CIFAR datasets under either backbone, and pFedMMA has the highest nine-dataset macro-F1 for both backbones. Across all nine datasets, pFedAER and pFedMoAP have similar macro P10 with RN50 (68.69 versus 68.53); ViT-B/16 yields a 0.45-point advantage for pFedAER. The clearest secondary-metric evidence therefore comes from combined feature and label shift, consistent with Section 4.2.

Table 11: Secondary endpoints with RN50. F1 is macro-F1 and P10 is the 10th-percentile client accuracy. Values average the final five rounds and are reported as mean ± sample standard deviation over three seeds. Bold marks the best mean for each dataset and metric.

	pFedAER		pFedMoAP		pFedMMA	
Dataset	F1 ↑	P10 ↑	F1 ↑	P10 ↑	F1 ↑	P10 ↑
Flowers102	98.76±0.07	97.59±0.34	98.16±0.14	96.89±0.36	**99.07**±0.04	**98.07**±0.10
OxfordPets	98.49±0.25	95.51±1.55	**98.75**±0.20	96.22±0.78	98.60±0.08	**96.95**±0.90
Food101	**93.09**±0.05	**90.36**±0.06	92.97±0.04	89.97±0.20	92.66±0.16	89.63±0.31
Caltech101	96.98±0.12	94.87±0.15	97.01±0.14	95.24±0.13	**97.26**±0.20	**95.69**±0.15
DTD	**91.23**±0.35	**83.93**±0.85	90.53±0.30	82.44±0.06	91.03±0.22	82.59±0.57
CIFAR10	61.28±0.69	73.01±0.19	66.12±1.57	73.92±1.37	**74.72**±0.28	**82.36**±0.24
CIFAR100	38.53±0.20	40.92±0.36	44.53±0.19	45.55±0.81	**53.13**±0.41	**52.87**±0.45
DomainNet	11.74±0.40	**24.16**±0.32	10.42±0.50	23.62±0.13	**12.02**±0.24	24.01±1.24
Office-Caltech10	**17.45**±1.75	**17.88**±2.81	14.67±1.69	12.93±4.12	16.63±2.94	12.55±2.96

Table 12: Secondary endpoints with ViT-B/16, using the reporting convention of Table 11.

	pFedAER		pFedMoAP		pFedMMA	
Dataset	F1 ↑	P10 ↑	F1 ↑	P10 ↑	F1 ↑	P10 ↑
Flowers102	99.47±0.03	98.77±0.17	99.15±0.12	98.53±0.09	**99.53**±0.12	**98.89**±0.19
OxfordPets	99.10±0.05	**98.61**±0.36	**99.19**±0.05	98.46±0.29	98.82±0.15	95.86±0.76
Food101	**96.56**±0.05	94.92±0.11	96.37±0.03	94.64±0.13	96.54±0.06	**94.99**±0.07
Caltech101	98.74±0.11	**97.74**±0.46	98.74±0.08	97.55±0.17	**98.95**±0.05	97.68±0.10
DTD	93.00±0.55	88.26±1.04	92.46±0.34	87.77±0.93	**94.11**±0.30	**88.47**±0.24
CIFAR10	87.38±1.06	89.69±0.98	89.36±0.61	90.08±1.04	**90.53**±1.25	**92.93**±0.49
CIFAR100	63.47±0.33	62.83±0.21	67.27±0.21	65.89±0.24	**73.69**±0.16	**72.12**±0.60
DomainNet	**11.92**±0.34	**24.96**±0.52	10.33±0.27	23.55±0.75	11.49±0.54	24.00±1.23
Office-Caltech10	**19.21**±0.71	**17.32**±1.75	14.40±2.29	12.58±4.64	17.25±0.40	11.66±2.44

D.3 Client-level distributions under combined shift

Figure 4 pools the client endpoints from the three seeds while preserving each client's final-five-round average. The DomainNet curves largely overlap, consistent with the small P10 differences in Tables 11 and 12. The separation is clearer on Office-Caltech10. With RN50, the pooled 10th percentile is 19.10 for pFedAER, compared with 16.60 for pFedMoAP and 11.66 for pFedMMA. With ViT-B/16, the corresponding values are 17.86, 12.63, and 13.16. The figure therefore locates the Office-Caltech10 gain in the lower and middle parts of the client distribution rather than in a small number of high-performing clients. Pooled quantiles and table P10 use the distinct aggregation orders specified in Appendix C.3.

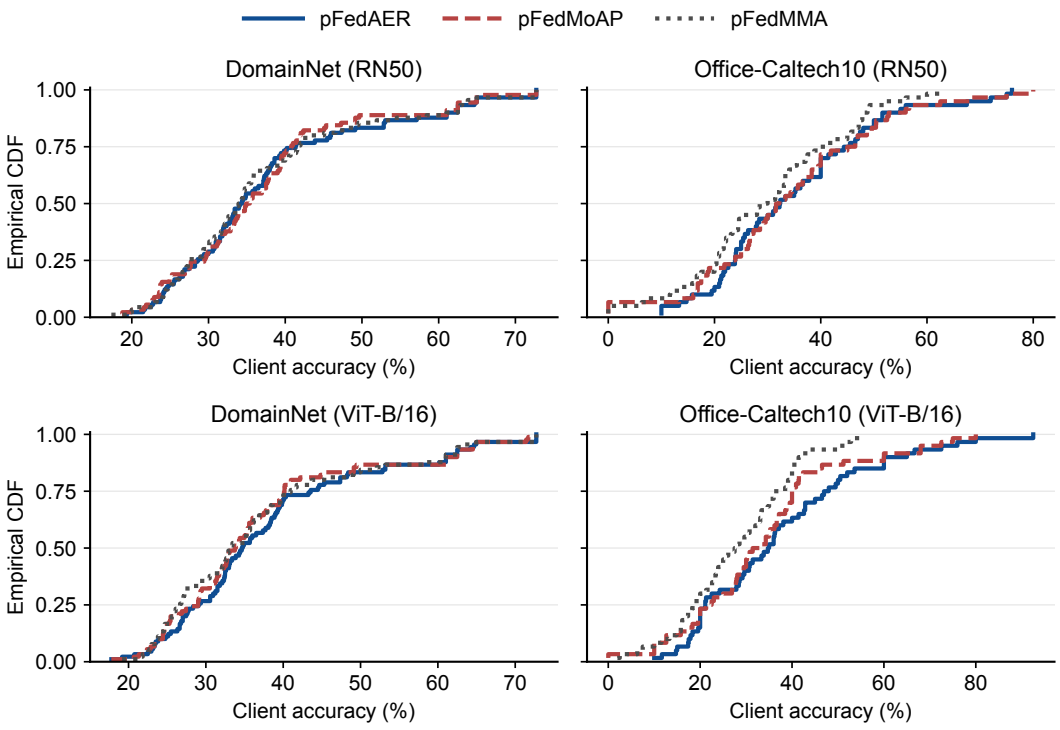

Figure 4: Client-level accuracy distributions on DomainNet and Office-Caltech10. Each curve pools final-five-round client averages over three seeds, giving 90 DomainNet observations and 60 Office-Caltech10 observations per method and backbone. A curve shifted toward higher accuracy indicates better client-level performance.

E Retrieval Diagnostics and Component Ablations

This section complements Section 4.3 by examining how individual retrieval components affect prediction and how the learned policy responds to feedback. Component ablations measure changes within pFedAER, while per-dataset diagnostics describe the operating conditions of the reference-backbone runs.

E.1 Component ablations

Table 13 compares the full method with removal of each scoring term, a fixed retrieval count, and removal of the negative guard on CIFAR10/RN50 and DomainNet/ViT-B/16. All configurations use $K_{\max} = 9$. The $w_u = 0$ variant removes the mean-utility contribution to ranking; reward statistics remain available to the variance penalty and guard. It therefore tests one use of feedback within the retriever.

Table 13: Component ablations on CIFAR10 (RN50) and DomainNet (ViT-B/16). Mean P10 equally averages the two datasets within each seed. Values are mean ± sample standard deviation over three seeds; the shaded row is the full-method reference.

	Accuracy (%) ↑		Mean P10 (%) ↑
Variant	CIFAR10	DomainNet	
Full pFedAER	80.74±0.47	38.17±0.79	48.99±0.33
Scoring terms			
w/o mean-utility score ($w_u = 0$)	80.90±0.46	38.00±0.57	48.57±0.42
w/o context ($w_c = 0$)	81.00±0.45	38.12±0.16	48.91±0.69
w/o exploration ($\beta = 0$)	80.96±0.43	38.07±1.04	48.61±0.61
w/o freshness ($w_f = 0$)	80.87±0.48	38.09±0.68	49.08±0.56
w/o variance penalty ($w_v = 0$)	80.75±0.48	38.19±0.25	49.23±0.59
Budget and guard			
Fixed $K = K_{\max}$	80.74±0.47	37.92±0.73	48.87±0.61
w/o negative guard	80.74±0.47	37.87±0.10	49.04±0.40

All variants report mean $K = 8.99 \pm 0.00$ and prompt downloads of 1.051 ± 0.000 MiB per round at the displayed precision. Both quantities equally average the two datasets within each seed.

Scoring components. Removing mean utility or exploration produces the largest decreases in two-dataset mean P10, at 0.42 and 0.38 points. On DomainNet, removing the mean-utility score lowers accuracy from 38.17% to 38.00%, while removing exploration yields 38.07%. These changes are consistent with a role for observed utility and exploration in serving lower-performing clients. The effects are modest relative to seed variability, and CIFAR10 does not share the mean-accuracy gains. Context, freshness, and the variance penalty have no consistent advantage across the reported metrics.

Budget and guard variants. The fixed-count and no-guard variants reach 37.92% and 37.87% accuracy on DomainNet, compared with 38.17% for the full method; their CIFAR10 results coincide at the displayed precision. All variants report a two-dataset mean $K = 8.99$ and traffic of 1.051 MiB per round. The observed differences thus occur at approximately matched reported download volume. Their interpretation also depends on whether the removed mechanism is active: the reference DomainNet runs record no filtered candidates at the displayed precision (Table 14). These endpoint differences alone do not establish a benefit from guard filtering.

E.2 Feedback and guard activity across datasets

Table 14 resolves the feedback behavior in Figure 3 by dataset. DTD has the most negative feedback: 99.3% of released rewards are below zero, an average of 5.63 candidates are filtered per selection, and 62.6% of selections invoke the all-filtered fallback. DomainNet lies at the other end, with

0.2% negative rewards and no reported filtering. The guard's operating regime therefore varies substantially with the recipient task.

CIFAR10 and CIFAR100 have negative-reward rates above 50% but no reported filtering. A negative reward alone does not satisfy the persistence and running-mean conditions in Eq. 12. On DTD, frequent fallback further shows why guard activity and prevention of harmful transfer should be distinguished: restoring the pool maintains expert availability when exclusions exhaust it. These measurements characterize the response to feedback without treating every exclusion as a verified predictive improvement.

Similarity and observed utility. The mean run-level similarity–reward correlation is negative for every dataset and is strongest on CIFAR100. This extends the pooled evidence in Figure 2 across tasks: prompt similarity alone does not reliably order the released utility observed by the retriever. The correlations describe selected pairs under the learned policy, which is the feedback available to subsequent retrieval decisions.

Table 14: Retrieval diagnostics for the reference-backbone pFedAER runs. Neg. is the percentage of released rewards below zero, Filt. is the number of candidates filtered per training selection, and FB is the percentage of selections invoking the all-filtered fallback. r_{sim} is the run-level similarity–reward correlation. Values are mean $\pm$ sample standard deviation over three seeds.

Dataset	Neg. (%)	Filt.	FB (%)	r_{sim}
Flowers102	56.7 ± 4.0	2.13 ± 0.67	23.7 ± 7.4	-0.177 ± 0.040
OxfordPets	67.4 ± 7.2	2.10 ± 0.10	23.3 ± 1.1	-0.119 ± 0.069
Food101	19.6 ± 6.3	0.27 ± 0.31	3.0 ± 3.4	-0.061 ± 0.150
Caltech101	57.0 ± 7.4	1.63 ± 1.21	18.1 ± 13.4	-0.221 ± 0.133
DTD	99.3 ± 1.3	5.63 ± 0.15	62.6 ± 1.7	-0.067 ± 0.072
CIFAR10	51.5 ± 4.5	0.00 ± 0.00	0.0 ± 0.0	-0.583 ± 0.092
CIFAR100	57.1 ± 3.4	0.00 ± 0.00	0.0 ± 0.0	-0.754 ± 0.007
DomainNet	0.2 ± 0.4	0.00 ± 0.00	0.0 ± 0.0	-0.213 ± 0.069
Office-Caltech10	15.1 ± 1.0	0.15 ± 0.07	0.0 ± 0.0	-0.149 ± 0.031

F Communication, System Cost, and Robustness

We complement the budget study in Section 4.4 with per-dataset download and runtime measurements, followed by changes in client participation, feedback quality, and expert availability. This separates the resource use of the default configuration from its behavior under altered operating conditions.

F.1 Per-dataset prompt downloads

Table 15 reports delivered expert counts and prompt traffic for the reference-backbone configurations. Mean counts remain close to the cap of nine on every dataset, while traffic also depends on the number of active clients and the training schedule. The dataset-macro mean is 1.223 MiB per round. The default threshold consequently provides broad expert access with little contraction of the retrieval count.

The main sensitivity study evaluates more economical settings directly: on CIFAR10/RN50, reducing $K_{\max}$ from nine to three lowers prompt traffic by 66.6% with a 0.08-point mean-accuracy decrease. At a fixed cap of nine, increasing τ_u to 2.0 instead reduces traffic by 8.1%. These are distinct controls over the performance–communication trade-off, and both should be assessed together with the P10 changes in Table 5.

Table 15: Prompt downloads for the reference-backbone pFedAER runs. K counts delivered non-local prompts per training retrieval; traffic is aggregate downloaded prompt data per round. The macro row weights datasets equally within each seed. Values are mean $\pm$ sample standard deviation over three seeds.

Dataset	Backbone	Mean K	MiB/round
Flowers102	RN50	9.00 ± 0.00	1.266 ± 0.000
OxfordPets	RN50	9.00 ± 0.00	1.266 ± 0.000
Food101	RN50	9.00 ± 0.00	1.266 ± 0.000
Caltech101	RN50	9.00 ± 0.00	1.266 ± 0.000
DTD	RN50	9.00 ± 0.00	1.266 ± 0.000
CIFAR10	RN50	9.00 ± 0.00	1.289 ± 0.000
CIFAR100	RN50	9.00 ± 0.00	1.289 ± 0.000
DomainNet	ViT-B/16	8.97 ± 0.00	0.813 ± 0.000
Office-Caltech10	ViT-B/16	8.98 ± 0.03	1.291 ± 0.004
Macro average	Mixed	9.00 ± 0.00	1.223 ± 0.000

F.2 Runtime and retrieval-state size

The server-side decision requires little measured processing time relative to client training. Table 16 reports server times from 0.067 to 0.283 seconds per round, below 0.3% of client training time on each dataset. The dataset-macro averages are 0.119 seconds for server processing, 66.58 seconds for client training, and 47.99 seconds for evaluation. These measurements describe the evaluated configurations; they do not measure the incremental overhead against a different retriever.

Persistent retrieval state increases from approximately 9 KiB on the 10-client datasets to 784–786 KiB on the 100-client CIFAR datasets. This growth is consistent with the directed pairwise state described in the main text. Its measured size is small at the evaluated scales, while the $O(N^2)$ dependence remains relevant for larger client populations.

Table 16: System measurements for the reference-backbone pFedAER runs. Train, Eval., and Server are measured in seconds per round. State is the final persistent retrieval-state size. Values are mean $\pm$ sample standard deviation over three seeds.

Dataset	Train	Eval.	Server	State (KiB)
Flowers102	62.7 ± 0.1	24.6 ± 0.2	0.069 ± 0.003	9.2 ± 0.0
OxfordPets	34.8 ± 0.2	25.0 ± 0.1	0.067 ± 0.001	9.3 ± 0.0
Food101	64.6 ± 0.7	139.6 ± 1.2	0.071 ± 0.001	9.1 ± 0.0
Caltech101	59.7 ± 0.0	17.4 ± 0.0	0.068 ± 0.001	9.3 ± 0.0
DTD	39.8 ± 0.3	22.9 ± 0.1	0.067 ± 0.001	9.2 ± 0.0
CIFAR10	103.4 ± 0.3	65.2 ± 2.8	0.280 ± 0.004	786.4 ± 3.4
CIFAR100	116.4 ± 1.2	71.4 ± 1.5	0.283 ± 0.008	783.9 ± 0.5
DomainNet	80.9 ± 3.8	54.8 ± 2.5	0.077 ± 0.000	71.3 ± 0.2
Office-Caltech10	36.9 ± 0.6	10.9 ± 0.2	0.094 ± 0.002	33.3 ± 0.0

F.3 Participation, feedback, and delivery robustness

Table 17 complements the budget and threshold analysis in Section 4.4. All experiments use CIFAR10 with RN50 and the same 120-round schedule as the default, varying participation or one feedback/delivery condition at a time.

Table 17: Additional sensitivity and stress tests on CIFAR10 with RN50. The default uses $K_{\max} = 9$, $\tau_u = 0.5$, participation 0.1, and no feedback perturbation, delay, expert dropout, or sign-flipping clients. Each variant changes one factor. Values are mean $\pm$ standard deviation over three seeds.

Setting	Acc. ↑	P10 ↑	Mean K ↓	MiB/round ↓
Default	80.74 ± 0.47	73.01 ± 0.19	9.00 ± 0.00	1.289 ± 0.000
Client participation				
$r = 0.3$	79.39 ± 0.52	70.39 ± 0.42	9.00 ± 0.00	4.102 ± 0.000
$r = 0.5$	78.72 ± 0.61	69.97 ± 0.83	9.00 ± 0.00	6.911 ± 0.001
$r = 1.0$	77.80 ± 0.52	68.59 ± 0.32	8.90 ± 0.02	13.788 ± 0.033
Feedback noise standard deviation				
$\sigma = 0.01$	80.82 ± 0.38	73.06 ± 0.26	9.00 ± 0.00	1.289 ± 0.000
$\sigma = 0.05$	80.98 ± 0.57	73.29 ± 0.62	9.00 ± 0.00	1.289 ± 0.000
$\sigma = 0.10$	80.94 ± 0.47	73.37 ± 0.18	9.00 ± 0.00	1.289 ± 0.000
Feedback delay (rounds)				
$d = 1$	80.78 ± 0.52	73.02 ± 0.24	9.00 ± 0.00	1.289 ± 0.000
$d = 3$	80.93 ± 0.52	72.93 ± 0.63	9.00 ± 0.00	1.289 ± 0.000
$d = 5$	80.98 ± 0.52	72.97 ± 0.32	9.00 ± 0.00	1.289 ± 0.000
Expert dropout probability				
$p_{\text{drop}} = 0.1$	80.89 ± 0.50	72.48 ± 0.33	8.10 ± 0.03	1.160 ± 0.004
$p_{\text{drop}} = 0.3$	80.85 ± 0.44	72.80 ± 0.30	6.31 ± 0.05	0.904 ± 0.007
$p_{\text{drop}} = 0.5$	80.67 ± 0.26	72.42 ± 0.66	4.50 ± 0.02	0.645 ± 0.003
Sign-flipping client fraction				
$p_{\text{mal}} = 0.1$	80.83 ± 0.49	73.12 ± 0.12	9.00 ± 0.00	1.289 ± 0.000
$p_{\text{mal}} = 0.3$	80.78 ± 0.51	72.88 ± 0.19	9.00 ± 0.00	1.289 ± 0.000
$p_{\text{mal}} = 0.5$	80.77 ± 0.51	73.00 ± 0.56	9.00 ± 0.00	1.289 ± 0.000

Client participation. Increasing participation from 0.1 to 1.0 raises aggregate prompt download traffic from 1.289 to 13.788 MiB per round, while mean accuracy falls from 80.74% to 77.80% and P10 falls from 73.01% to 68.59%. Because the round count is fixed, higher participation also changes the number of local training updates. The comparison therefore captures the combined effect of communication and optimization changes.

Feedback perturbations. Across the tested noise levels, delays of up to five rounds, and sign-flipping fractions up to 0.5, mean accuracy remains within 0.24 points of the default and P10 within 0.36 points. All of these settings report mean $K = 9.00$ and 1.289 MiB per round at the displayed precision. The small predictive changes should be interpreted in this budget-saturated setting; they do not establish that expert rankings or utility estimates are insensitive to corrupted feedback.

Expert dropout. At dropout probability 0.5, mean delivered K falls to 4.50 and traffic to 0.645 MiB per round, approximately half of the default. Mean accuracy decreases by 0.07 points and P10 by 0.59 points. Here, the reduction in delivered traffic results from unavailable experts, whereas Section 4.4 varies the retrieval policy's budget parameters.

G Controlled Synthetic Study

G.1 Environment and comparison

The controlled study isolates retrieval from vision optimization so that the directed utility of every pair is observable. We generate 20 clients in four prompt clusters. Each prompt is a normalized 12-dimensional vector sampled around its cluster center. True directed utility combines prompt similarity and an unobserved asymmetric interaction:

$$\mu_{ij} = 0.30\langle P_i, P_j\rangle + 0.70\langle h_i, e_j\rangle - 0.12, \tag{13}$$

where client preferences h_i and expert traits e_j are independently sampled as normalized vectors. We simulate 100 rounds, allow at most five experts, and add Gaussian feedback noise with standard deviation 0.08. Random and KNN always select five experts. Results are the mean $\pm$ sample standard deviation over 10 seeds.

G.2 Retrieval outcomes

Table 4 reports the simulation endpoints. The selected-utility mean increases from 0.059 for Euclidean KNN to 0.209 for pFedAER, while the reported harmful-selection rate falls from 39.2% to 8.7%. Mean downloads decrease from five to 3.710, corresponding to 74.2% of the maximum budget. The comparison thus uses the same upper bound on communication, with different realized download counts. These measurements demonstrate the behavior of the retrieval policy in the specified synthetic environment; the end-to-end experiments in Section 4 assess its behavior when prompts and gates also evolve during training.

The utility in this environment is specified directly by Eq. 13. Harmful selection can therefore be assessed against the underlying utility, whereas the real-data diagnostics observe the attention-weighted mixture feedback in Eq. 4. The synthetic results support the retrieval mechanism under observable directed transfer; the predictive results and component ablations assess its behavior within the full prompt-learning system.