FREEING CLIENTS WHILE ENHANCING LARGE MODELS VIA FEDERATED WEAK-TO-STRONG LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated fine-tuning enables large models (LMs) to adapt to decentralized private data without raw data sharing. Existing parameter-efficient methods reduce the number of trainable parameters but still require clients to execute large model backbones, imposing substantial memory and computational overhead on resource-constrained devices. To overcome this limitation, we propose **FedW2S**, a federated weak-to-strong framework that frees up client computational bottlenecks and enhances large models. Instead of deploying a strong model locally, FedW2S transfers the knowledge learned from private data by lightweight weak models on the client side to a strong model on the server side. To mitigate unreliable supervision from weak models, FedW2S combines reliability-based filtering with discrepancy-aware target mixing, selectively absorbing informative weak knowledge while preserving useful knowledge encoded in the strong model. Compatibility-aware strong-to-weak feedback further enables bidirectional knowledge transfer, allowing server-side improvements to propagate back to weak clients and enhance their subsequent supervision. We also conduct a theoretical analysis of this. Experiments across visual and language tasks show that FedW2S achieves competitive strong model performance while significantly reducing the memory and computational requirements for client-side training. The code is available at: `https://anonymous.4open.science/r/FedW2S`.

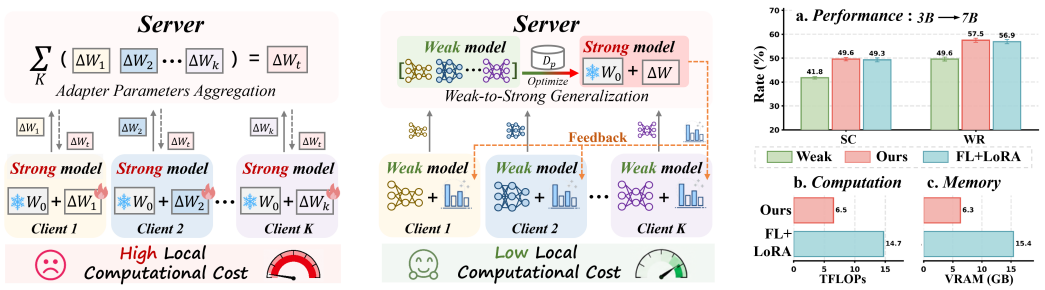
1 INTRODUCTION

Large models (LMs) provide a useful initialization for downstream tasks, but adapting them to specialized applications often requires domain-specific data (Chen et al., 2026; Szep et al., 2026; Lin et al., 2026a). Such data may be distributed across organizations or devices and unavailable for centralized training. Federated Learning (FL) enables collaborative model training while keeping private training data on clients (McMahan et al., 2017). However, applying FL to large models requires that participating clients meet the computational and memory requirements for local training.

Recent work on federated fine-tuning of LMs has primarily focused on parameter-efficient fine-tuning (PEFT), including LoRA and adapters (Zhang et al., 2024; Xiong et al., 2025; Zhang et al., 2026a). As illustrated in Figure 1a, these methods update and communicate only a small set of trainable parameters, reducing optimizer memory and communication costs. The frozen backbone still accounts for the bulk of the local computation: **clients still execute the large model and backpropagate through the relevant layers to update the trainable parameters** (Ma et al., 2026). Thus, simply reducing the number of trainable parameters alone does not truly alleviate the constraints imposed by limited client-side computational resources.

Federated distillation enables knowledge exchange among heterogeneous models through predictions on shared inputs, and prior work (Fan et al., 2025; Lin et al., 2020; Li & Wang, 2019) has explored collaboration between clients and a server. Building on this foundation, we focus on the weak-to-strong (W2S) setting (Jeon et al., 2026; Burns et al., 2024; Lang et al., 2026) in FL, where task-specific knowledge aggregated from lightweight client weak models is used to improve a pre-trained server strong model. When labels are unavailable for the shared public inputs, the reliability of client predictions cannot be directly assessed against ground-truth targets. This may propagate incorrect knowledge to the server strong model during the knowledge transfer process. The central

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.



(a) Federated PEFT (b) FedW2S (c) Performance and Cost

Figure 1: Comparison with federated PEFT. FedW2S keeps lightweight weak models on clients, enhances a strong server model through bidirectional knowledge transfer, and reduces local computation and memory costs while maintaining strong performance.

question is therefore: *How can a strong server model learn from heterogeneous weak clients while limiting the propagation of their errors?*

This setting gives rise to two fundamental challenges. ***First, weak client supervision is informative but inherently unreliable.*** Weak models can capture valuable task-specific knowledge from private data, yet their predictions may be unreliable on public inputs. Without ground-truth labels, disagreement between weak supervision and the strong model may indicate that the strong model lacks useful task knowledge, or that the teachers' predictions are incorrect. Effective weak-to-strong transfer must therefore identify trustworthy client knowledge and control how much of it should be absorbed without compromising the strong model's existing knowledge. ***Second, strong-model knowledge is not universally beneficial to every client.*** Feeding enhanced server knowledge back to weak clients can improve their subsequent supervision, but the strong model may not provide useful guidance for clients under heterogeneous local distributions. Indiscriminate feedback may conflict with local optimization and override valuable client-specific knowledge. Effective strong-to-weak transfer must therefore determine when server knowledge is beneficial and regulate its influence.

To address these challenges, we propose **FedW2S**, a federated weak-to-strong framework for enhancing a strong server model using knowledge from lightweight weak clients. As illustrated in Figure 1b, FedW2S exchanges predictive distributions over unlabeled public data, allowing clients to contribute task-specific knowledge without deploying the strong backbone. On the server side, we introduce a reliability-based filtering strategy to effectively select weakly supervised public data and dynamically adjust the supervision budget as client knowledge develops. For the retained samples, a discrepancy-aware target mixing strategy is used to balance weakly supervised knowledge and strong predictive generalization. While learning from weak supervision, it retains the useful knowledge already encoded in the strong model. After the server update, FedW2S enables bidirectional knowledge transfer through the compatibility-aware strong-to-weak feedback. Only when the server demonstrates a confidence advantage does each client selectively absorb feedback knowledge, relying on absorption weights to limit interference with client-specific knowledge. Our theoretical analysis shows that the proposed mechanisms provide controlled updates for both the server and the client, supporting the feasibility of FedW2S.

Our main contributions are summarized as follows:

- We propose FedW2S, a federated weak-to-strong framework that transfers knowledge learned by weak client models from private data to a strong server model, eliminating the need to deploy the strong model on clients.
- We develop a bidirectional knowledge transfer mechanism that enables weak clients and the strong server model to collaboratively improve each other, while selectively regulating knowledge transfer in both directions to mitigate unreliable or incompatible supervision. The impact of this mechanism is further characterized from a theoretical perspective.
- Extensive experiments across visual and language tasks have shown that FedW2S achieves competitive performance while reducing client memory and computational cost under different data distributions and model configurations.

2 RELATED WORK

Weak-to-Strong Generalization. W2S (Burns et al., 2024) studies whether a strong pretrained model can learn from weaker supervision while generalizing beyond its teacher. Subsequent studies examine its mechanisms and limitations (Medvedev et al., 2025; Bai et al., 2026; Ye et al., 2026), including the reliability of weak supervision (Jeon et al., 2026; Lang et al., 2026). Other works have investigated W2S under distribution shifts and multi-capability settings, where the quality of weak supervision can vary substantially across inputs and tasks (Zhou et al., 2025; Dong et al., 2026). FedW2S studies this problem under federated data heterogeneity, where weak clients provide complementary but unreliable task knowledge. It couples server weak-to-strong with strong-to-weak feedback to improve the client predictions used in subsequent rounds. Ultimately, this enhances the performance of pretrained strong models on the server side.

Parameter-Efficient Fine-Tuning in FL. PEFT provides a practical approach for adapting LMs in FL by optimizing only a small fraction of model parameters (Babakniya et al., 2023). Recent works (Zhang et al., 2024; Guo et al., 2025; Bai et al., 2024; Meng et al., 2026; Zhang et al., 2026a) have focused primarily on mitigating optimization drift caused by non-IID data, correcting aggregation bias in low-rank factors, and supporting client-side personalization. Nevertheless, these methods typically still require each client to store and execute the full pretrained backbone. Another line of work (Lin et al., 2026b; Zhang et al., 2026b; Nair et al., 2025) build upon split learning, where only a small portion of the model is deployed on clients while most computation is offloaded to the server. However, it typically requires frequent transmission of intermediate features, which introduces additional communication overhead and potential privacy risks (Liu et al., 2025). More recently, FedSLM (Zhu et al., 2026) deploys heterogeneous compressed client models and transfers aggregated knowledge to a full-scale server model. FedMKT (Fan et al., 2025) adopts a setting similar to ours but requires labeled public data. In contrast, FedW2S transfers knowledge through predictions on unlabeled public data, avoiding full-model deployment and intermediate feature exchange on clients. We further discuss federated distillation in Appendix C.

3 METHOD

3.1 FEDERATED WEAK-TO-STRONG LEARNING

We consider a federated weak-to-strong setting consisting of K resource-constrained clients and a computationally capable server. Each client $k \in [K]$ has a private dataset $\mathcal{D}_k = (x_i, y_i)_{i=1}^{n_k}$ and maintains a lightweight weak model $f_k^w(\cdot;\boldsymbol{\theta}_k)$. The server maintains a pretrained strong model $f^s(\cdot;\boldsymbol{\Theta})$ and accesses an unlabeled public dataset $\mathcal{D}_p = \{x_i\}_{i=1}^{n_p}$. The strong model is never deployed to clients. In each round t, each client updates the weak model $f_k^w(\cdot;\boldsymbol{\theta})$ using private data, and sends model parameters or predictions on $\mathcal{D}_p$ to the server. The server filters out unreliable weak supervised samples and optimizes the strong model $f^s(\cdot;\boldsymbol{\Theta})$ via weak-to-strong, thereby establishing a strong-to-weak knowledge feedback loop. The objective of FedW2S is to improve the server-side strong model while reducing client memory and computation.

Weak-to-Strong Generalization on the Server Side. First, each client generates a prediction distribution $\boldsymbol{p}_k^w = \sigma\left(f_k^w(x;\boldsymbol{\theta}_k)\right)$ based on the public dataset, which is then transmitted to the server as client knowledge. The server aggregates these heterogeneous predictions into a weak knowledge signal $\boldsymbol{p}^w = \mathcal{A}\left(\{\boldsymbol{p}_k^w\}_{k\in[K]}\right)$, where $\mathcal{A}(\cdot)$ denotes the knowledge aggregation operator. Based on this, the client and server can use heterogeneous models with different structures. Finally, the optimization objective for the server side are as follows:

$$\mathcal{L}_{\mathrm{W2S}} = \mathbb{E}_{x\sim\mathcal{D}_p}\left[\mathcal{G}(x)\cdot D_{\mathrm{KL}}\left(\boldsymbol{p}^w \parallel \boldsymbol{p}^s(x;\boldsymbol{\Theta})\right)\right],\ where\ \ \boldsymbol{p}^s(x;\boldsymbol{\Theta}) = \sigma\left(f^s(x;\boldsymbol{\Theta})\right), \tag{1}$$

where σ is the softmax function and D_{KL} denotes the Kullback–Leibler divergence. In addition, $\mathcal{G}(x)$ denotes a reliability selection indicator that adaptively controls the contribution of weak knowledge according to the confidence of the weak model.

Knowledge Feedback on the Client Side. FedW2S can further utilize the knowledge signals from strong models in public data to enhance weak models through feedback. After the strong model is updated, it generates feedback knowledge $\boldsymbol{q}^s = \sigma\left(f^s(x;\boldsymbol{\Theta}^{t+1})\right)$ on the public dataset. The

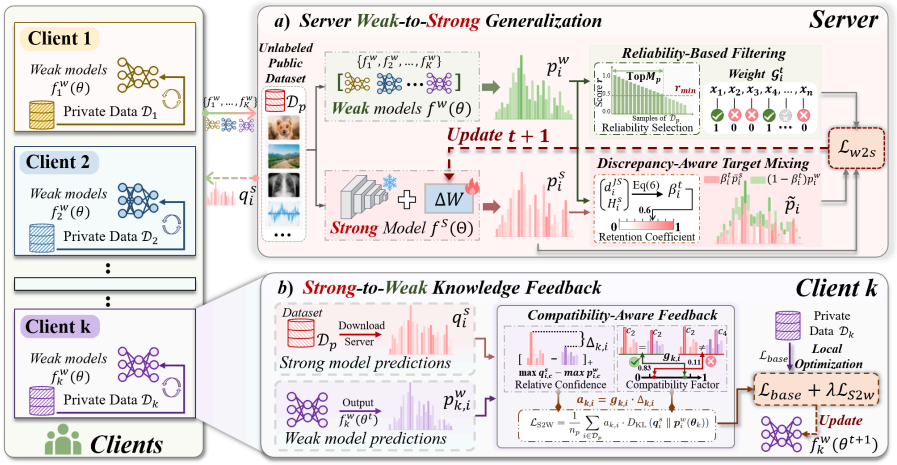


Figure 2: The structure of our FedW2S framework. The framework consists of *a) Server Weak-to-Strong Generalization* and *b) Strong-to-Weak Knowledge Feedback*.

optimization objective for the client side are as follows:

$$\mathcal{L}_k = \underbrace{\mathbb{E}_{(x,y)\sim\mathcal{D}_k}\left[\ell\left(f_k^w(x;\boldsymbol{\theta}_k), y\right)\right]}_{\text{Base Task}} + \lambda \underbrace{\mathbb{E}_{x\sim\mathcal{D}_p}\left[a(x)\cdot D_{\mathrm{KL}}\left(\boldsymbol{q}^s \parallel \boldsymbol{p}_k^w(x;\boldsymbol{\theta}_k)\right)\right]}_{\text{Knowledge Feedback}}, \tag{2}$$

where $a(x)$ represents the client's selective absorption weight for the server's knowledge. Strong model knowledge is used to update the client only when it demonstrates a positive advantage. This prevents global knowledge from overshadowing the specific knowledge learned by the client. Throughout the FedW2S process, weak models continuously contribute domain-specific knowledge to strong models, while strong models provide generalized knowledge to improve the weak clients. The overall framework is shown in Figure 2, and the algorithm is described in Algorithm 1.

3.2 Server Weak-to-Strong Generalization

In this subsection, we focus on describing how FedW2S can better utilize distributed knowledge extracted from weak clients to enhance server strong models.

Reliability-Based Filtering. Before initiating the W2S, multiple rounds of federated warm-up are first performed to mitigate the instability of the initial weak model. Next, the server aggregates the weak model predictions to obtain a global weak knowledge prediction $\boldsymbol{p}_i^w = \sum_{k\in[K]} \frac{n_k}{\sum n_k}\sigma(f_k^w(x_i;\boldsymbol{\theta}_k^t))$. We define a sample-level reliability metric for the outputs of weak models, as follows:

$$r_i = \sqrt{\frac{C\pi_{i,(1)} - 1}{C-1}\cdot\left(\pi_{i,(1)} - \pi_{i,(2)}\right)}, \tag{3}$$

where C is the number of classes. Let $\pi_{i,(1)}$ and $\pi_{i,(2)}$ denote the largest and second-largest entries of $\boldsymbol{p}_i^w$. This score combines normalized prediction confidence with the top two probability margin, serving as a confidence-based proxy for the reliability of weak supervision. To adapt to changes in weak models across rounds, we determine the sample selection budget M_p using the average normalized predictive entropy $\mathcal{H}_w = -\frac{1}{n_p \log C}\sum_{i=1}^{n_p}\sum_{c=1}^{C} p_{i,c}^w \log p_{i,c}^w$. Then the reliability-based filtering mechanism is defined as:

$$\mathcal{G}_i = \mathbb{I}\left(r_i \geq r_{\min} \; and \; i \in \mathrm{Top}\, M_p[r]\right), where \; M_p = \lfloor n_p(1-\mathcal{H}_w)\rfloor \tag{4}$$

where $\mathrm{Top}\, M_p[r]$ contains the indices of the M_p highest-scoring public samples, with ties resolved by sample index. $r_{\min} \in (0,1)$ is a fixed reliability floor, used to filter out samples with extremely low reliability. High predictive uncertainty reduces the selection budget, whereas lower uncertainty permits broader coverage. We further provide an empirical validation of the reliability score r_i in Figure 8. As FL training stabilizes, this mechanism allows a broader range of reliable samples to participate in W2S knowledge transfer.

Discrepancy-Aware Target Mixing. Directly fitting weak supervision may propagate client errors to the strong model. We therefore mix weak predictions with the server's pre-update predictions to balance knowledge acquisition and prediction retention. First, we cache the server predictions $\bar{\boldsymbol{p}}_i^s = \sigma(f^s(x_i; \boldsymbol{\Theta}^t))$. Specifically, we quantify prediction discrepancy d_i^{JS} via Jensen–Shannon divergence and the strong model's uncertainty H_i^s:

$$d_i^{\mathrm{JS}} = \frac{1}{2} D_{\mathrm{KL}}(\bar{\boldsymbol{p}}_i^s \parallel \boldsymbol{m}_i) + \frac{1}{2} D_{\mathrm{KL}}(\boldsymbol{p}_i^w \parallel \boldsymbol{m}_i), \quad H_i^s = -\sum_{c=1}^{C} \bar{p}_{i,c}^s \log \bar{p}_{i,c}^s. \tag{5}$$

where $\boldsymbol{m}_i = (\bar{\boldsymbol{p}}_i^s + \boldsymbol{p}_i^w)/2$ is midpoint distribution. Then we combine normalized discrepancy and uncertainty into a retention score s_i^t, and balance it against the weak-supervision reliability r_i:

$$\beta_i = \frac{s_i}{s_i + r_i} \ , \ \ where \ \ s_i = \frac{d_i^{\mathrm{JS}}}{\log 2} \left(1 - \frac{H_i^s}{\log C}\right), \tag{6}$$

This coefficient β_i increases retention when the server is confident and disagrees with weak supervision, while assigning greater weight to more reliable weak predictions. Finally, the discrepancy-aware mixed supervision target is $\widetilde{\boldsymbol{p}}_i = \mathrm{stopgrad}[\beta_i \bar{\boldsymbol{p}}_i^s + (1-\beta_i)\boldsymbol{p}_i^w]$. The weak-to-strong optimization loss $\mathcal{L}_{\mathrm{W2S}}$ is:

$$\mathcal{L}_{\mathrm{W2S}} = \frac{1}{\sum \mathcal{G}_i + \epsilon} \sum_{i \in \mathcal{D}_p} D_{\mathrm{KL}}\left(\widetilde{\boldsymbol{p}}_i \parallel \boldsymbol{p}_i^s(\boldsymbol{\Theta})\right) \cdot \mathcal{G}_i, \tag{7}$$

where ϵ is a small positive constant and $\boldsymbol{p}_i^s(\boldsymbol{\Theta}) = \sigma(f^s(x_i; \boldsymbol{\Theta}))$ denotes the current server prediction. Reliability filtering $\mathcal{G}_i$ determines which samples provide supervision, while target mixing $\widetilde{\boldsymbol{p}}_i$ balances fitting weak predictions against retaining the server's predictions.

3.3 Strong-to-Weak Knowledge Feedback

Although server weak-to-strong generalization enriches the strong model, it cannot propagate the knowledge gains from the server back to the weak models. To address this, we introduce a client-side strong-to-weak knowledge feedback mechanism.

Compatibility-Aware Feedback. After the t round, the updated strong model evaluates the public dataset to construct strong predictions $\boldsymbol{q}_i^s = \sigma(f^s(x_i; \boldsymbol{\Theta}^{t+1}))$. These strong predictions inherently encode knowledge about the generalization capabilities of the server strong model. When performing knowledge feedback, the client weak model f_k^w also needs to selectively adopt the generalization knowledge from the strong model f^s. Specifically, for client k and public sample x_i, let $q_{i,c}^s \in \boldsymbol{q}_i^s$ and $p_{k,i,c}^w \in \boldsymbol{p}_{k,i}^w$ denote the predicted distributions from the strong model and weak model, respectively. We define the relative confidence margin of the server as

$$\Delta_{k,i} = \left[\max_c q_{i,c}^s - \max_c p_{k,i,c}^w\right]_+, \tag{8}$$

where $[\cdot]_+ = \max(0, \cdot)$. This score measures a difference in predictive confidence rather than an accuracy advantage. When $\Delta_{k,i} = 0$, the server has no relative advantage over the client, and its feedback is therefore suppressed. A positive confidence margin alone does not account for disagreement between the predictive distributions in Equation 8. We further use the Jensen-Shannon divergence to construct the distributional compatibility factor $g_{k,i}$:

$$g_{k,i} = 1 - \frac{d_{k,i}^{\mathrm{JS}}(q_i^s, p_{k,i}^w)}{\log 2} \in [0, 1]. \tag{9}$$

This factor continuously attenuates feedback as distributional disagreement increases, without imposing a hard requirement that the predicted classes match. Finally, the selective absorption weight is $a_{k,i} = \mathrm{stopgrad}[g_{k,i} \cdot \Delta_{k,i}]$. Thus, $\Delta_{k,i}$ gates feedback by the server's confidence advantage, while $g_{k,i}$ modulates its strength according to predictive compatibility. Then the strong-to-weak feedback regularization loss is:

$$\mathcal{L}_{\mathrm{S2W}} = \frac{1}{n_p} \sum_{i \in \mathcal{D}_p} a_{k,i} \cdot D_{\mathrm{KL}}\left(\boldsymbol{q}_i^s \parallel \boldsymbol{p}_i^w(\boldsymbol{\theta}_k)\right). \tag{10}$$

Integration with Federated Training. The final training objectives for each client's weak model are as follows:

$$\mathcal{L}_k = \mathcal{L}_{\mathrm{base}}^k(\mathcal{D}_k; \boldsymbol{\theta}_k) + \lambda \mathcal{L}_{\mathrm{S2W}}(\mathcal{D}_p; \boldsymbol{\theta}_k), \tag{11}$$

where $\mathcal{L}^k_{\text{base}}$ includes the private task loss and any regularization terms prescribed by the base FL method. This formula retains the local training component of the base FL method and adapts to the knowledge transfer structures of different FL methods. Furthermore, the feedback module does not require matching model parameters and can adapt to heterogeneous client architectures.

4 THEORETICAL ANALYSIS

In this section, we analyze the two transfer directions of FedW2S within one communication round. Let u_i, w_i, and v_i denote the cached server prediction, the aggregated weak prediction, and the final server prediction on public input x_i.

Server-side transfer. For a selected sample $i \in S$, the mixed target is $z_i = \beta_i u_i + (1-\beta_i)w_i$, where $\beta_i = s_i/(r_i+s_i)$. We measure prediction quality by the conditional Brier risk $b_i(p) = \mathbb{E}[\|p - e_Y\|_2^2 \mid X = x_i]$, where e_Y is the one-hot label. The mixed target satisfies

$$b_i(u_i) - b_i(z_i) = (1-\beta_i)\big[b_i(u_i) - b_i(w_i) + \beta_i\|u_i - w_i\|_2^2\big]. \tag{12}$$

This identity separates the weak teacher's relative risk from the benefit of mixing. In particular, a locally worse weak teacher can still improve the target when $b_i(w_i) - b_i(u_i) < \beta_i\|u_i - w_i\|_2^2$.

Let $n_S = |S|$ and $\rho = n_S/n_p$. For a nonempty S, define the average target gain A_S, target-fitting error δ_S, and prediction drift on unselected samples $D_{\bar{S}}$:

$$A_S = \frac{1}{n_S}\sum_{i\in S}[b_i(u_i) - b_i(z_i)], \;\; = \frac{1}{n_S}\sum_{i\in S} D_{\mathrm{KL}}(z_i\|v_i), \;\; D_{\bar{S}} = \frac{1}{n_p}\sum_{i\notin S}\|v_i - u_i\|_2. \tag{13}$$

Theorem 1 (Server weak-to-strong risk bound)**.** *Fix the public inputs and a nonempty selected set S. Suppose that $r_i \geq r_{\min} > 0$ and $s_i \geq 0$ for $i \in S$, and that the cached predictions, selection, and mixing coefficients remain fixed throughout the server update. Let $\widehat{R}_B(p) = n_p^{-1}\sum_i b_i(p_i)$ and use natural logarithms in δ_S. For any final probability predictions v_i with finite δ_S,*

$$\widehat{R}_B(v) - \widehat{R}_B(u) \leq -\rho A_S + 2\sqrt{2}\big(\rho\sqrt{\delta_S} + D_{\bar{S}}\big). \tag{14}$$

In particular, $\widehat{R}_B(v) < \widehat{R}_B(u)$ whenever $\rho A_S > 2\sqrt{2}(\rho\sqrt{\delta_S} + D_{\bar{S}})$.

Client-side feedback. The compatibility gate regulates the influence of each server prediction on the client's distillation gradient.

Proposition 1 (Strong-to-weak feedback influence)**.** *At a client gradient evaluation, let $p_i = \mathrm{softmax}(\ell_i)$ use unit temperature and let q_i be a fixed server probability vector. Using natural logarithms, define $d_i = D_{\mathrm{JS}}(p_i, q_i)$ and compute $a_i = [\max_c q_{i,c} - \max_c p_{i,c}]_+(1 - d_i/\log 2)$ from these same predictions. If a_i is detached during differentiation, then*

$$\begin{aligned}\|\nabla_{\ell_i}[a_i D_{\mathrm{KL}}(q_i\|p_i)]\|_2 &= a_i\|p_i - q_i\|_2 \\ &\leq 2\sqrt{2}\, d_i(1 - d_i/\log 2) \leq \frac{\log 2}{\sqrt{2}}.\end{aligned} \tag{15}$$

Theorem 1 identifies when the benefit of complementary weak predictions is preserved after distillation: the mixed-target gain must outweigh fitting error and prediction drift. Proposition 1 establishes a uniform bound on each feedback contribution in logit space, with an upper envelope that vanishes as disagreement approaches its maximum. These properties support target mixing as a balance between knowledge acquisition and prediction retention, and compatibility gating as a means of regulating feedback strength. Proofs are provided in Appendix E.

5 EXPERIMENTS

5.1 EXPERIMENT SETUP

Base Models and Datasets: We evaluate FedW2S on both vision and language tasks to assess its effectiveness across different settings. For *image classification* tasks, we consider three diverse real-world benchmarks: ImageNet (Deng et al., 2009), iWildCam (Beery et al., 2021) and FMoW

Table 1: Performance of baseline FL methods and their FedW2S versions under two non-IID settings ($\vartheta = 0.1$ and $\vartheta = 1.0$). Weak and Strong denote the client and server models, respectively. The downward arrow $\downarrow$ indicates the relative gap between the baseline weak and strong models.

Method	ImageNet				iWildCam				FMoW			
	$\vartheta = 0.1$		$\vartheta = 1.0$		$\vartheta = 0.1$		$\vartheta = 1.0$		$\vartheta = 0.1$		$\vartheta = 1.0$	
	Weak	*Strong*	*Weak*	*Strong*	*Weak*	*Strong*	*Weak*	*Strong*	*Weak*	*Strong*	*Weak*	*Strong*
Local-Weak	46.56		57.12		42.11		53.47		23.55		31.76	
FedAvg	58.68 $\downarrow_{23.8\%}$		67.62 $\downarrow_{18.8\%}$		54.26 $\downarrow_{11.7\%}$		65.50 $\downarrow_{8.7\%}$		30.87 $\downarrow_{14.7\%}$		39.14 $\downarrow_{14.6\%}$	
+ FedW2S	**60.46**	**77.04**	**68.60**	**83.32**	**54.85**	**61.43**	**66.72**	**71.78**	**32.63**	**36.21**	**40.55**	**45.81**
FedProx	58.66 $\downarrow_{23.6\%}$		67.58 $\downarrow_{18.9\%}$		55.69 $\downarrow_{10.4\%}$		66.34 $\downarrow_{8.7\%}$		30.55 $\downarrow_{16.3\%}$		39.19 $\downarrow_{14.8\%}$	
+ FedW2S	**60.26**	**76.76**	**68.58**	**83.34**	**56.68**	**62.18**	**67.54**	**72.69**	**32.82**	**36.52**	**41.05**	**45.98**
SCAFFOLD	54.18 $\downarrow_{25.3\%}$		61.47 $\downarrow_{20.4\%}$		49.31 $\downarrow_{16.4\%}$		58.73 $\downarrow_{12.9\%}$		28.35 $\downarrow_{17.8\%}$		37.46 $\downarrow_{12.3\%}$	
+ FedW2S	**56.08**	**72.50**	**63.31**	**77.19**	**52.72**	**58.96**	**60.21**	**67.43**	**29.63**	**34.51**	**38.27**	**42.71**
FedDyn	53.18 $\downarrow_{25.3\%}$		62.04 $\downarrow_{22.0\%}$		51.38 $\downarrow_{14.7\%}$		61.76 $\downarrow_{9.9\%}$		29.40 $\downarrow_{17.1\%}$		36.63 $\downarrow_{15.8\%}$	
+ FedW2S	**55.20**	**71.16**	**63.74**	**79.52**	**52.83**	**60.21**	**62.27**	**68.54**	**31.28**	**35.46**	**38.25**	**43.48**
FedDF	51.56 $\downarrow_{25.4\%}$		59.21 $\downarrow_{22.9\%}$		52.94 $\downarrow_{12.8\%}$		63.48 $\downarrow_{9.9\%}$		28.93 $\downarrow_{16.3\%}$		36.95 $\downarrow_{16.3\%}$	
+ FedW2S	**53.75**	**69.14**	**61.83**	**76.81**	**53.56**	**60.74**	**65.24**	**70.43**	**30.19**	**34.57**	**37.92**	**44.17**
FedMD	61.16 $\downarrow_{22.1\%}$		69.58 $\downarrow_{18.0\%}$		54.71 $\downarrow_{12.4\%}$		65.52 $\downarrow_{9.7\%}$		31.37 $\downarrow_{17.1\%}$		40.68 $\downarrow_{14.5\%}$	
+ FedW2S	**62.20**	**78.51**	**70.94**	**84.87**	**56.25**	**62.48**	**66.85**	**72.50**	**32.96**	**37.85**	**42.16**	**47.56**

(Koh et al., 2021). Following prior work (Jeon et al., 2026), we use pretrained DINO family ViT-B/8 ($7.6\times$) and ViT-L/14 ($27\times$) (Oquab et al., 2023) as strong models, and ResNet-18 ($1\times$) and ResNet-50 ($2.1\times$) as weak models. Classification accuracy is reported as the primary evaluation metric. For *language tasks*, we study preference alignment with Direct Preference Optimization (DPO) and use the widely adopted UltraFeedback dataset (Cui et al., 2024) for training. We select models with different parameter sizes from the same model family, using Qwen2.5-3B-Instruct as the weak model and the larger-parameter Qwen2.5-7B-Instruct as the strong model. We evaluate instruction-following performance on two widely used benchmarks, AlpacaEval 2 (Dubois et al., 2024) and Arena-Hard (Li et al., 2025). For AlpacaEval, we report both the length-controlled win rate (LC) and the raw win rate (WR); for Arena-Hard, we report the win rate (WR) and the style-controlled win rate (SC). To simulate statistical heterogeneity, we partition the training data using the commonly adopted Dirichlet ϑ distribution.

Baselines: We compare FedW2S against 13 representative baseline methods, covering both traditional FL methods and federated fine-tuning methods. For *traditional FL* methods, we consider FedAvg (McMahan et al., 2017), FedProx (Li et al., 2020), SCAFFOLD (Karimireddy et al., 2020), FedDyn (Acar et al., 2021), FedDF (Lin et al., 2020), and FedMD (Li & Wang, 2019). To further examine the generality of our FedW2S learning framework, we additionally instantiate the FedW2S strategy on top of these FL methods. For *federated fine-tuning* methods, we compare against FedIT (Zhang et al., 2024), FedSA-LoRA (Guo et al., 2025), FlexLoRA (Bai et al., 2024), FLoRG (Meng et al., 2026), Fed-PLoRA (Zhang et al., 2026a), HSplitLoRA (Lin et al., 2026b) and FedSLM (Zhu et al., 2026). All main results are averaged over three independent runs. Further details and additional experiments are provided in Appendix A and Appendix B.

5.2 Main Results

Comparison with Traditional FL. Table 1 evaluates FedW2S on top of six federated baselines under two levels of data heterogeneity. Among all the results, incorporating FedW2S consistently improves the weak client models by approximately 1.5 percentage points. More importantly, the resulting strong server models substantially outperform the corresponding weak baselines. The gains are observed for both parameter aggregation methods, such as FedAvg, FedProx, SCAFFOLD, and FedDyn, and distillation methods, including FedDF and FedMD. For example, the strong models reach up to 84.87%, 72.69%, and 47.56% on ImageNet, iWildCam, and FMoW, respectively. These consistent improvements indicate that FedW2S is highly compatible with other methods. In particular, the benefits obtained when FedW2S is combined with FedDF and FedMD show that our reliability filtering, target mixing, and knowledge feedback provide additional value beyond simple knowledge distillation. In addition, Figure 3 shows the training dynamics under different

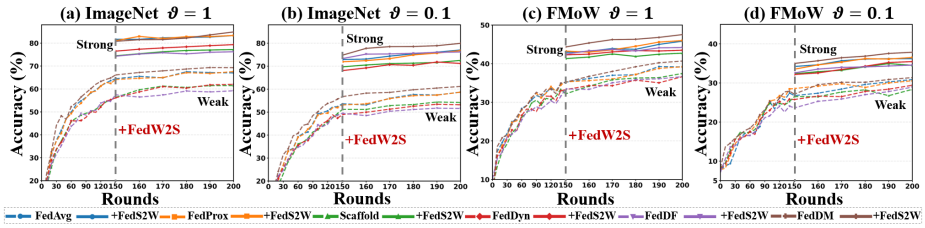


Figure 3: Performance over Communication Rounds under Heterogeneous Data Distributions.

Table 2: Performance comparison among federated fine-tuning methods. Strong model performance is reported for all baselines, whereas FedW2S trains weak models on clients.

Setting	Method	ImageNet			iWildCam			FMoW		
		$\vartheta = 0.1$	$\vartheta = 1.0$	*Avg*	$\vartheta = 0.1$	$\vartheta = 1.0$	*Avg*	$\vartheta = 0.1$	$\vartheta = 1.0$	*Avg*
Strong Model	Local-Strong	53.41	61.16	57.29	39.47	53.81	46.64	23.09	32.46	27.78
	FedIT	76.26	85.82	81.04	53.99	70.67	62.33	36.51	43.37	39.94
	FedSA-LoRA	72.88	79.62	76.25	54.76	68.27	61.52	34.39	42.58	38.49
	FlexLoRA	75.12	**86.12**	80.62	57.89	71.02	64.46	37.95	46.44	42.20
	FLoRG	78.74	84.84	**81.79**	61.89	71.23	66.56	**38.21**	45.28	41.75
	Fed-PLoRA	74.35	84.60	79.48	59.62	70.78	65.20	37.24	46.12	41.68
Partial Model	HSplitLoRA	78.32	82.94	80.63	61.62	72.31	66.97	37.13	47.02	42.08
	FedSLM	77.46	84.25	80.86	61.47	71.12	66.30	36.96	47.13	42.05
Weak Model	FedW2S	**78.51**	84.87	81.69	**62.48**	**72.50**	**67.49**	37.85	**47.56**	**42.71**

data heterogeneity levels. After the Server Weak-to-Strong Generalization is introduced, the server-side strong model rapidly attains substantially higher accuracy than the weak clients and converges within only a few additional rounds. These results suggest that FedW2S can exploit distributed weak knowledge while maintaining stable strong-model adaptation under non-IID client data.

Comparison with Federated Fine-Tuning Methods. Table 2 compares FedW2S with federated fine-tuning methods that deploy the strong backbone on clients. This is a conservative comparison for FedW2S, since our clients train only lightweight weak models. Despite this difference, FedW2S achieves competitive strong-model performance across all three datasets. On ImageNet, it achieves an average accuracy of 81.69%, which is close to the best result of 81.79%. It also achieves the highest average performance on iWildCam (67.49%) and FMoW (42.71%). Compared with FedSLM, which also transfers knowledge from compressed client models to a full-scale server model, FedW2S provides consistently higher results across the three datasets. This improvement shows that our bidirectional transfer mechanism outperforms naive unidirectional transfer.

Generalization to Preference Optimization. Table 3 evaluates FedW2S on AlpacaEval 2 and Arena-Hard after preference optimization. The strong server model achieves an average score of 48.5, matching the strongest federated fine-tuning baseline (HSplitLoRA 48.4). Our method also maintains competitive performance on AlpacaEval 2. These results show that FedW2S can effectively transfer decentralized preference knowledge to a high-capacity server model while preserving its lightweight client-side training setting.

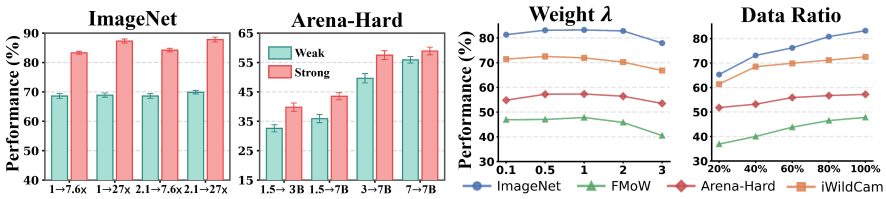
Table 3: Preference alignment results on AlpacaEval 2 and Arena-Hard.

Method	AlpacaEval 2		Arena-Hard		Avg.
	LC	WR	SC	WR	
Base(Strong)	$31.5_{\pm 0.4}$	$30.6_{\pm 1.3}$	$37.3_{\pm 1.8}$	$39.4_{\pm 2.1}$	34.7
FedIT	$36.8_{\pm 0.3}$	$44.4_{\pm 1.8}$	$46.5_{\pm 2.3}$	$54.9_{\pm 2.9}$	45.7
FedSA-LoRA	$35.7_{\pm 0.4}$	$45.9_{\pm 1.3}$	$47.4_{\pm 2.1}$	$55.2_{\pm 2.4}$	46.1
FlexLoRA	$38.9_{\pm 0.5}$	$45.3_{\pm 1.5}$	$47.1_{\pm 2.2}$	$55.0_{\pm 2.2}$	46.6
FLoRG	$38.2_{\pm 0.4}$	$46.6_{\pm 1.6}$	$48.8_{\pm 2.7}$	$55.6_{\pm 2.5}$	47.3
Fed-PLoRA	$39.1_{\pm 0.2}$	$47.8_{\pm 1.7}$	$49.3_{\pm 2.1}$	$56.9_{\pm 2.3}$	48.3
HSplitLoRA	$\mathbf{40.1}_{\pm 0.1}$	$\mathbf{47.7}_{\pm 1.1}$	$49.1_{\pm 1.8}$	$56.8_{\pm 2.0}$	48.4
FedSLM	$39.2_{\pm 0.4}$	$46.5_{\pm 1.7}$	$48.6_{\pm 2.9}$	$56.4_{\pm 2.8}$	47.7
FedW2S	$39.5_{\pm 0.3}$	$47.3_{\pm 1.6}$	$\mathbf{49.6}_{\pm 1.8}$	$\mathbf{57.5}_{\pm 2.1}$	**48.5**

Client Resource Efficiency. Table 4 reports the client-side resource requirements based on parameter estimation. Although PEFT methods train only adapter parameters, they still execute the full strong backbone locally. For vision tasks, FedW2S requires approximately 0.19 GB of memory and 0.01 TFLOPs per input, while the PEFT methods require 0.65–0.84 GB and 0.33–0.43 TFLOPs. The benefits are also clear for language tasks: FedW2S reduces memory usage from 15.4 to 6.31 GB and computational requirements from 14.71 to 6.49 TFLOPs. HSplitLoRA has low resource requirements but requires frequent communication of intermediate features. FedW2S avoids executing strong backbone and feature transfers while reducing the computational load on the client.

Table 4: Client resource requirement comparison under different federated learning methods.

Metric (*One input*)	FedAvg	FedMD	FedIT	FedSA-LoRA	FLoRG	Fed-PLoRA	HSplitLoRA	FedSLM	Ours
Vision Tasks: ResNet-18 → DINO ViT-L /14									
Trainable Params. (M)	11.23	11.23	1.68	1.68	0.79	1.68	0.79	18.98	11.23
Client Memory (GB) ↓	≈0.19	≈0.19	≈0.65	≈0.65	≈0.84	≈0.65	≈0.17	≈0.41	≈0.19
Client FLOPs (TFLOPs) ↓	≈0.01	≈0.01	≈0.33	≈0.33	≈0.43	≈0.33	≈0.08	≈0.21	≈0.01
Language Tasks: Qwen2.5-3B → Qwen2.5-7B									
Trainable Params. (M)	—	—	5.05	5.05	1.84	5.05	2.37	343.35	3.69
Client Memory (GB) ↓	—	—	≈15.4	≈15.4	≈16.9	≈15.4	≈7.86	≈9.56	≈6.31
Client FLOPs (TFLOPs) ↓	—	—	≈14.71	≈14.71	≈16.3	≈14.71	≈6.27	≈8.83	≈6.49



(a) Model Scale (b) Effect of Weight λ and Public Data

Figure 4: Robustness analyses of FedW2S. (a) Performance under different FedW2S model scale combinations. (b) Sensitivity to the feedback weight λ and effect of the public data ratio.

5.3 Ablation and Analysis

Component Ablations. Table 5 shows that removing any component reduces the performance of FedW2S across all four tasks. *Reliability filtering* has the largest impact, with performance drops of 4.2–6.3 points, confirming the importance of suppressing unreliable weak supervision. Removing *target mixing* causes a substantial degradation on ImageNet and iWildCam, indicating that directly following weak predictions can interfere with the strong model's prior knowledge. Meanwhile, *Knowledge Feedback* further improves performance by enabling bidirectional knowledge transfer between the server and the client. Further analysis of reliability scores r_i and other variants in Appendix B.1 and B.2.

Table 5: Ablation study of the key components in FedW2S.

Variant	Imag.	iWil.	Fmow	Alpa.
w/o *Reliability filtering*	78.6	67.3	43.4	38.7
w/o *Target Mixing*	79.4	69.1	47.1	42.3
w/o *Knowledge Feedback*	81.2	68.9	46.3	42.6
FedW2S	**84.9**	**72.5**	**47.6**	**43.4**

Scaling across Model Capacities. In Figure 4a, we evaluate the performance of FedW2S under four combinations of weak-to-strong models. FedW2S consistently improves performance across different model combinations. In general, increasing the capacity of weak models or strong models can improve the performance of FedW2S, but there is a certain upper limit. This indicates that FedW2S is robust to variations in model capacity.

Hyperparameter Sensitivity. In Figure 4b, we investigate the sensitivity of FedW2S to the feedback weight λ and the amount of public data. FedW2S performs consistently for $\lambda \in [0.1, 2]$, with the best results obtained around $\lambda = 0.5$-1. A larger feedback weight results in a significant drop, indicating that excessive strong supervision interferes with local task optimization. Increasing the public data ratio improves performance on all tasks, while the gains become smaller at higher ratios for public datasets. FedW2S still achieves some effectiveness even with limited public data.

6 Conclusion and Discussion

In this work, we introduce FedW2S, a federated weak-to-strong learning framework that enables resource-constrained clients to collaborate with a stronger server model without deploying the latter on devices. FedW2S transfers client knowledge through public predictions, improves the server model via reliability weak-to-strong, and further feeds generalized server knowledge back to clients. We also provide a theoretical analysis of the proposed method. One limitation of FedW2S is its reliance on a public dataset. When there is a significant mismatch between the public data and the client distribution, the effectiveness of this method may decline. Moreover, effective weak-to-strong transfer requires the strong model to possess sufficient prior knowledge to absorb and generalize from weak client supervision. Future work will further investigate these issues.

AI USE STATEMENT

Generative AI tools are used in a limited capacity to assist with language editing, grammar correction, and improving the clarity and readability of the manuscript. They are also occasionally used to assist with coding and debugging during implementation. All research ideas, methodological design, experimental setup, data analysis, interpretation of results, and conclusions are developed and determined by the authors. All AI-assisted content is carefully reviewed and verified by the authors. The authors take full responsibility for the final content of this work.

ETHICS STATEMENT

This work adheres to the ICLR Code of Ethics. We do not recruit human participants, collect new personal data, or conduct experiments involving human or animal subjects. All datasets used in our experiments, including ImageNet, iWildCam, FMoW, AlpacaEval 2, UltraFeedback, and Arena-Hard, are obtained from publicly available sources and used according to their respective licenses and usage guidelines. Our federated experiments use simulated non-IID client partitions, and no personally identifiable information is intentionally accessed or processed. We do not conduct experiments that involve personally identifiable information.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide the implementation of FedW2S in an anonymous repository: `https://anonymous.4open.science/r/FedW2S`. The experimental setup in the main paper and appendix A specifies the weak and strong model configurations, dataset preparation, non-IID client partitions, training schedules, hyperparameters, the public data construction, and evaluation protocols. These details support replication of our experiments on visual recognition and language preference optimization, including the baseline comparisons and ablation studies.

REFERENCES

Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. *arXiv preprint arXiv:2111.04263*, 2021.

Sara Babakniya, Ahmed Roushdy Elkordy, Yahya H Ezzeldin, Qingfeng Liu, Kee-Bong Song, Mostafa El-Khamy, and Salman Avestimehr. Slora: Federated parameter efficient fine-tuning of language models. *arXiv preprint arXiv:2308.06522*, 2023.

Jiamu Bai, Daoyuan Chen, Bingchen Qian, Liuyi Yao, and Yaliang Li. Federated fine-tuning of large language models under heterogeneous tasks and client resources. *Advances in Neural Information Processing Systems*, 37:14457–14483, 2024.

Lichen Bai, Masashi Sugiyama, and Zeke Xie. Weak-to-strong diffusion with reflection. In *International Conference on Learning Representations*, volume 2026, pp. 58921–58957, 2026.

Sara Beery, Arushi Agarwal, Elijah Cole, and Vighnesh Birodkar. The iwildcam 2021 competition dataset. *arXiv preprint arXiv:2105.03494*, 2021.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, Ilya Sutskever, and Jeffrey Wu. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4971–5012. PMLR, 21–27 Jul 2024. URL `https://proceedings.mlr.press/v235/burns24b.html`.

Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *Advances in Neural Information Processing Systems*, 38:57654–57689, 2026.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Ultrafeedback: boosting language models with scaled ai feedback. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Junhao Dong, Cong Zhang, Xinghua Qu, Zejun Ma, Piotr Koniusz, and Yew Soon Ong. Robust superalignment: Weak-to-strong robustness generalization for vision-language models. *Advances in Neural Information Processing Systems*, 38:18345–18377, 2026.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024.

Tao Fan, Guoqiang Ma, Yan Kang, Hanlin Gu, Yuanfeng Song, Lixin Fan, Kai Chen, and Qiang Yang. Fedmkt: Federated mutual knowledge transfer for large and small language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 243–255, 2025.

Pengxin Guo, Shuang Zeng, Yanran Wang, Huijie Fan, Feifei Wang, and Liangqiong Qu. Selective aggregation for low-rank adaptation in federated learning. In *International Conference on Learning Representations*, volume 2025, pp. 99003–99027, 2025.

Chaoyang He, Murali Annavaram, and Salman Avestimehr. Group knowledge transfer: Federated learning of large cnns at the edge. *Advances in neural information processing systems*, 33:14068–14080, 2020.

Myeongho Jeon, Jan Sobotka, Suhwan Choi, and Maria Brbic. Weak-to-strong generalization under distribution shifts. *Advances in Neural Information Processing Systems*, 38:153420–153458, 2026.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pp. 5132–5143. PMLR, 2020.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International conference on machine learning*, pp. 5637–5664. PMLR, 2021.

Hao Lang, Fei Huang, and Yongbin Li. Selective weak-to-strong generalization. In *Proceedings of the Fortieth AAAI Conference on Artificial Intelligence and Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence and Sixteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'26/IAAI'26/EAAI'26. AAAI Press, 2026. ISBN 978-1-57735-906-7. doi: 10.1609/aaai.v40i44.41089. URL https://doi.org/10.1609/aaai.v40i44.41089.

Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019.

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.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. From crowdsourced data to high-quality benchmarks: arena-hard and benchbuilder pipeline. In *Proceedings of the 42nd International Conference on Machine Learning*, ICML'25. JMLR.org, 2025.

Jiacheng Lin, Zhongruo Wang, Kun Qian, Tian Wang, Arvind Srinivasan, Hansi Zeng, Ruochen Jiao, Xie Zhou, Jiri Gesi, Dakuo Wang, et al. Sft doesn't always hurt general capabilities: Revisiting domain-specific fine-tuning in llms. In *International Conference on Learning Representations*, volume 2026, pp. 46954–46992, 2026a.

Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. *Advances in neural information processing systems*, 33:2351–2363, 2020.

Zheng Lin, Yuxin Zhang, Zhe Chen, Zihan Fang, Xianhao Chen, Praneeth Vepakomma, Wei Ni, Jun Luo, and Yue Gao. HSplitLoRA: A Heterogeneous Split Parameter-Efficient Fine-Tuning Framework for Large Language Models . *IEEE Transactions on Mobile Computing*, 25(10): 15543–15559, October 2026b. ISSN 1558-0660. doi: 10.1109/TMC.2026.3680521. URL https://doi.ieeecomputersociety.org/10.1109/TMC.2026.3680521.

Zihan Liu, Yizhen Wang, Rui Wang, and Sai Wu. Dualguard: A parameter space transformation approach for bidirectional defense in split-based llm fine-tuning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 17065–17080, 2025.

Jian Ma, Xinchen Lyu, Jun Jiang, Qimei Cui, Haipeng Yao, and Xiaofeng Tao. Splitfrozen: Split learning with device-side model frozen for fine-tuning llm on heterogeneous resource-constrained devices. *IEEE Communications Magazine*, 64(3):172–179, 2026. doi: 10.1109/MCOM.001.2500165.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. Pmlr, 2017.

Marko Medvedev, Kaifeng Lyu, Dingli Yu, Sanjeev Arora, Zhiyuan Li, and Nathan Srebro. Weak-to-strong generalization even in random feature networks, provably. *arXiv preprint arXiv:2503.02877*, 2025.

Chuiyang Meng, Ming Tang, and Vincent Wong. Florg: Federated fine-tuning with low-rank gram matrices and procrustes alignment. In *International Conference on Learning Representations*, volume 2026, pp. 82804–82824, 2026.

Srijith Nair, Michael Lin, Peizhong Ju, Amirreza Talebi, Elizabeth Serena Bentley, and Jia Liu. Fsl-sage: accelerating federated split learning via smashed activation gradient estimation. In *Proceedings of the 42nd International Conference on Machine Learning*, ICML'25. JMLR.org, 2025.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Marton Szep, Daniel Rueckert, Rüdiger von Eisenhart-Rothe, and Florian Hinterwimmer. Fine-tuning large language models with limited data: a survey and practical guide. *Transactions of the Association for Computational Linguistics*, 14:341–377, 2026.

Baochen Xiong, Xiaoshan Yang, Yaguang Song, Yaowei Wang, and Changsheng Xu. Pilot: Building the federated multimodal instruction tuning framework. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 21716–21724, 2025.

Ruimeng Ye, Zihan Wang, Yang Xiao, Zinan Ling, Manling Li, and Bo Hui. Weak-to-strong generalization with failure trajectories. In *International Conference on Learning Representations*, volume 2026, pp. 150477–150496, 2026.

Jianyi Zhang, Saeed Vahidian, Martin Kuo, Chunyuan Li, Ruiyi Zhang, Tong Yu, Guoyin Wang, and Yiran Chen. Towards building the federatedgpt: Federated instruction tuning. In *ICASSP 2024-2024 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 6915–6919. IEEE, 2024.

Zikai Zhang, Rui Hu, and Jiahao Xu. Heterogeneous federated fine-tuning with parallel one-rank adaptation. *arXiv preprint arXiv:2602.16936*, 2026a.

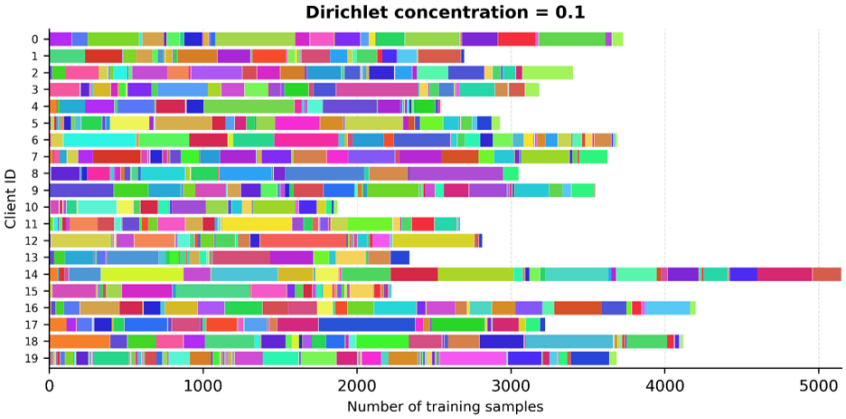
Zishuai Zhang, Hainan Zhang, Weihua Li, Qinnan Zhang, Jin Dong, Yongxin Tong, and Zhiming Zheng. Fedsea-llama: A secure, efficient and adaptive federated splitting framework for large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 28680–28688, 2026b.

Yucheng Zhou, Jianbing Shen, and Yu Cheng. Weak to strong generalization for large language models with multi-capabilities. In *International Conference on Learning Representations*, volume 2025, pp. 11583–11612, 2025.

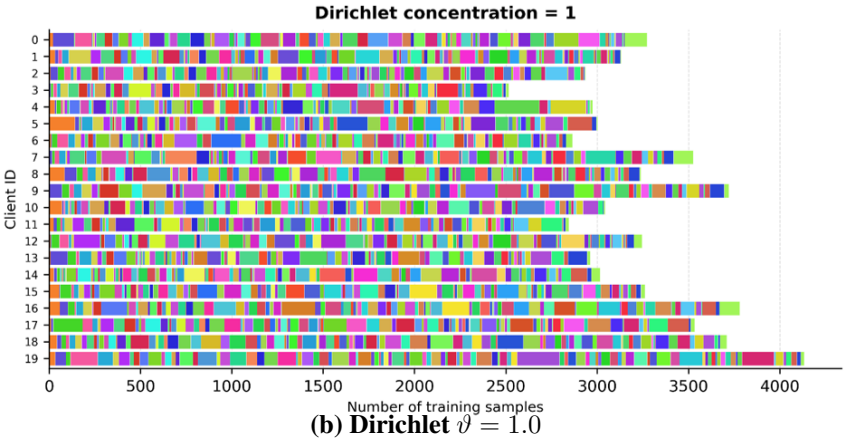
Shengkun Zhu, Jinshan Zeng, Zhihua Allen-Zhao, Mayi Xu, Quanqing Xu, Wei Ren, Qiang Yang, and Yang Liu. Federated foundation models fine-tuning with heterogeneous compressed clients. *arXiv preprint arXiv:2607.29071*, 2026.

A ADDITIONAL EXPERIMENTAL DETAILS

Unless otherwise specified, all experiments follow the same configuration. Client data are partitioned according to a Dirichlet distribution to simulate statistical heterogeneity in federated learning. As illustrated in Figures 5, 6, and 7, a smaller Dirichlet concentration parameter ($\vartheta = 0.1$) leads to substantially more skewed client data distributions than $\vartheta = 1.0$, reflecting a higher degree of statistical heterogeneity. By default, the system consists of 20 clients, and 40% of clients are uniformly sampled for participation in each communication round.



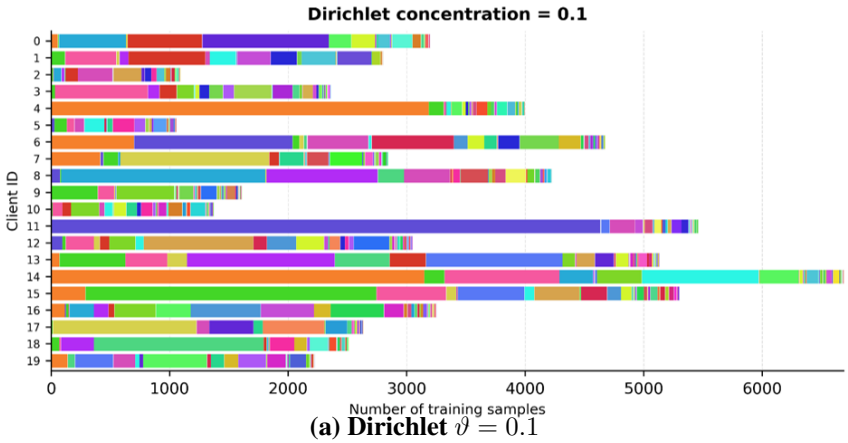
(a) Dirichlet $\vartheta = 0.1$



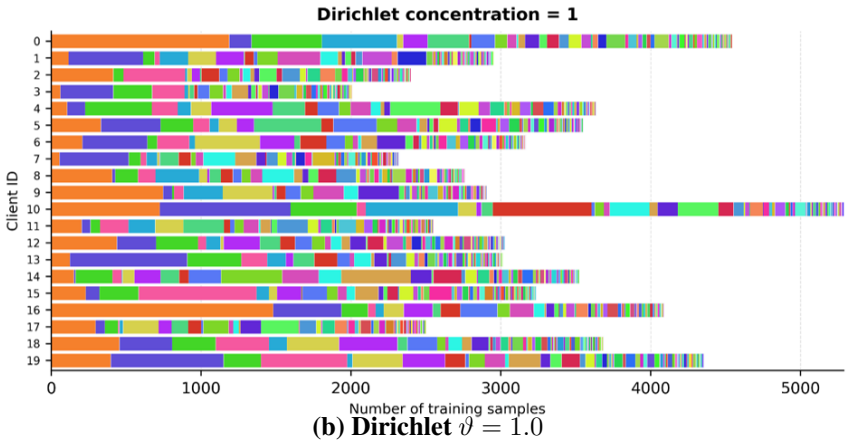
(b) Dirichlet $\vartheta = 1.0$

Figure 5: Visualization of client data distributions on ImageNet dataset.

Vision Experiments. For vision tasks, we train all methods for 200 communication rounds, with the first 150 rounds used as a warm-up stage. Clients optimize their local models using SGD with a learning rate of 0.01, momentum of 0.9, and weight decay of 5×10^{-4}. The dataset is split into source, public, and test subsets with the default ratio: $0.5 : 0.4 : 0.1$. For the ImageNet experiments, we use a subset of the dataset for training and official validation to evaluate the model. For FMoW, we preserve the original capture sequence structure during partitioning to prevent images from the same sequence being randomly distributed across client, public, and test splits. For ImageNet, the DINO backbone of the server model is frozen and only the task-specific classification head is optimized. For other vision datasets, we adopt LoRA-based parameter-efficient fine-tuning, where LoRA adapters and classification heads are optimized while keeping the pretrained backbone frozen. The server-side distillation temperature is set to $T = 2.0$. The strong model optimizer is AdamW with a learning rate of 5×10^{-5} and weight decay of 0.05. r_{min} is 0.2 and the batch size is 64.The default value of λ is 0.7. The W2S procedure for the server is 1 per round. To simulate statistical heterogeneity, we partition the training data across clients using a Dirichlet distribution.



(a) Dirichlet $\vartheta = 0.1$



(b) Dirichlet $\vartheta = 1.0$

Figure 6: Visualization of client data distributions on iWildCam dataset.

Language Experiments. For language tasks, we train all methods for 30 communication rounds. The warm-up is 20 rounds, the learning rate is 5e-05, and the number of local epochs is 1. Both client and server models are fine-tuned using LoRA with rank 16 and scaling factor $\alpha = 32$. Mixed-precision training is enabled by default (FP16). We further construct a Weak-to-Strong DPO setting following work (Jeon et al., 2026). The detailed design is as follows. We use Qwen2.5-3B-Instruct with a scalar reward head on clients and Qwen2.5-7B-Instruct as the server policy. UltraFeedback is partitioned into disjoint private and public subsets at the prompt level. Candidate responses are taken from the dataset rather than generated by clients. Private annotations supervise the standard pairwise reward-model loss, whereas public pairs $(x_i, y_{i,a}, y_{i,b})$ are presented without preference labels or a label-dependent ordering. Client LoRA parameters and reward heads are aggregated using sample-size weights. After warm-up rounds, federated reward learning alternates with server adaptation and client feedback.

Let $m_i(\theta) = f^w(x_i, y_{i,a}; \theta) - f^w(x_i, y_{i,b}; \theta)$ denote the weak reward margin. Let $\Delta\bar{\ell}_{\Theta,i}$ denote the difference between the two responses' mean token log-probabilities, excluding prompt and padding tokens. We define the strong preference logit and the corresponding Bernoulli distributions as

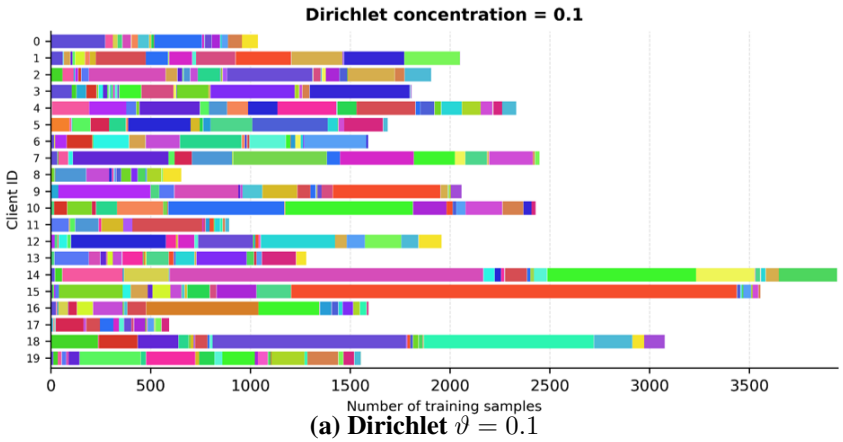
$$z_i^s(\Theta) = \kappa_{\mathrm{DPO}} \left(\Delta\bar{\ell}_{\Theta,i} - \Delta\bar{\ell}_{\mathrm{ref},i} \right), \tag{16}$$

$$p_i^w(\theta) = (\sigma(m_i(\theta)/T), 1 - \sigma(m_i(\theta)/T)), \tag{17}$$

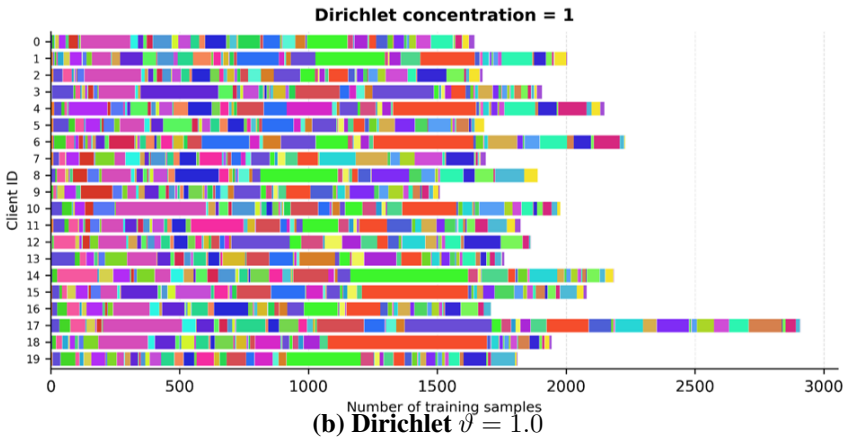
$$p_i^s(\Theta) = (\sigma(z_i^s(\Theta)), 1 - \sigma(z_i^s(\Theta))). \tag{18}$$

The first component represents the preference for $y_{i,a}$ over $y_{i,b}$. The reference policy is the frozen server backbone with its trainable adapters disabled.

At transfer round t, $\bar{\theta}^t$ denotes the aggregated weak reward-model parameters available before the server update. We apply the main-text reliability gate and discrepancy-aware mixing rule to these distributions with $C = 2$, denoting the mixing coefficient by β_i^t to distinguish it from κ_{DPO}. At



(a) Dirichlet $\vartheta = 0.1$



(b) Dirichlet $\vartheta = 1.0$

Figure 7: Visualization of client data distributions on FMoW dataset.

round t, weak predictions and pre-update server predictions are cached, and the complete mixed targets are detached:

$$\widetilde{p}_i^t = \mathrm{sg}\left[\beta_i^t p_i^s(\Theta^t) + (1-\beta_i^t) p_i^w(\bar{\theta}^t)\right], \tag{19}$$

$$\mathcal{L}_{\mathrm{W2S}}^t(\Theta) = \frac{1}{\sum_i \mathcal{G}_i^t + \epsilon} \sum_i \mathcal{G}_i^t \cdot b(z_i^s(\Theta), \widetilde{p}_{i,1}^t), \tag{20}$$

where $b(z,u) = -u\log\sigma(z) - (1-u)\log(1-\sigma(z))$. Thus, filtering selects public pairs, while target mixing specifies their soft preference supervision.

After the server update, clients receive $q_i^{t+1} = \mathrm{sg}[p_i^s(\Theta^{t+1})]$. Client k applies the main-text absorption weight to this feedback and its local preference predictions:

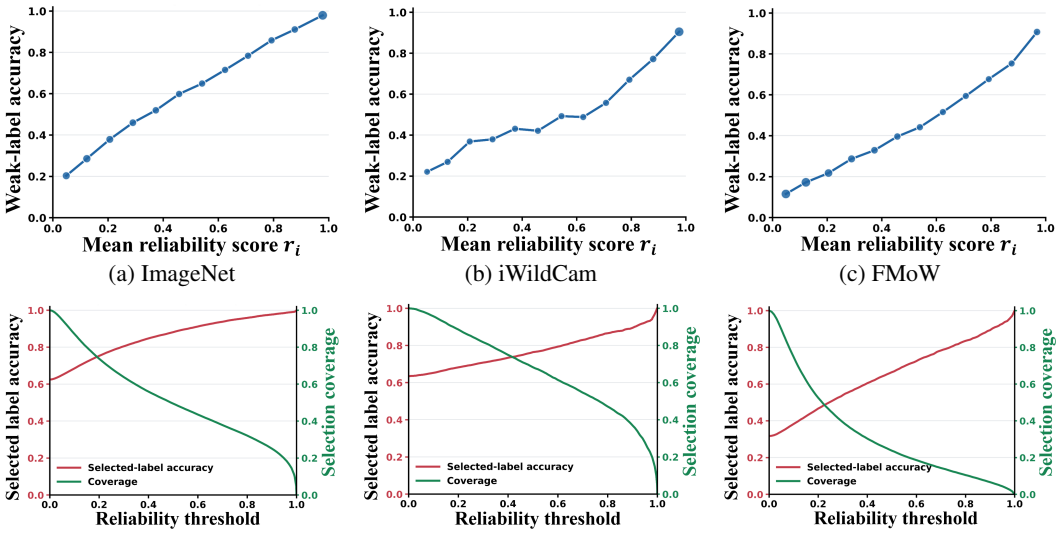
$$\mathcal{L}_k(\theta_k) = \mathcal{L}_{\mathrm{RM}}^k(\theta_k) + \frac{\lambda}{n_p} \sum_i \mathrm{sg}[a_{k,i}^{t+1}]\, b\big(m_i(\theta_k)/T, q_{i,1}^{t+1}\big). \tag{21}$$

This feedback uses the same preference definition as server training and does not include an additional ungated logit-matching loss. Server policies are evaluated on generation benchmarks.

B Additional Experiments

B.1 Empirical validation of the reliability score

We further investigate whether the proposed reliability score r_i provides an informative proxy for the correctness of weak supervision. As shown in Figure 8 (top), weak-label accuracy exhibits a clear positive relationship with the reliability score across all three datasets. In particular, samples in low-reliability bins are substantially more error-prone, whereas the empirical correctness of weak labels



(a) ImageNet (b) iWildCam (c) FMoW

(d) ImageNet (e) iWildCam (f) FMoW

Figure 8: Empirical validation of the proposed reliability score. Top: Weak-label correctness within bins of the reliability score r_i. Higher reliability is consistently associated with more accurate weak supervision. Bottom: Selected weak-label accuracy and sample coverage when sweeping a standalone reliability threshold. Increasing the threshold improves the quality of selected supervision while reducing coverage, revealing an accuracy–coverage trade-off.

progressively increases toward high-reliability regions. This trend is consistently observed on ImageNet, iWildCam, and FMoW despite their different data characteristics, suggesting that r_i provides a dataset-consistent ranking signal for distinguishing reliable from unreliable weak supervision.

We further examine the practical effect of reliability-based selection by sweeping a standalone threshold over r_i. Figure 8 (bottom) shows a consistent accuracy–coverage trade-off: increasing the reliability threshold steadily improves the accuracy of the retained weak labels while reducing the fraction of selected samples. This observation indicates that r_i is not only correlated with weak-label correctness, but also provides an effective criterion for controlling the quality and quantity of supervision transferred to the strong model. Together with the ablation results, these findings empirically support the use of r_i as the reliability signal in our filtering mechanism, which suppresses potentially erroneous weak supervision while retaining informative samples for weak-to-strong transfer. Ground-truth labels are used only for this post-hoc diagnostic analysis and are never involved in FedW2S training.

B.2 Additional variation experiments

Table 6: Comparing the effectiveness of different variants in our setting.

	ImageNet		iWildCam		FMoW	
Variants	$\vartheta = 0.1$	$\vartheta = 1$	$\vartheta = 0.1$	$\vartheta = 1$	$\vartheta = 0.1$	$\vartheta = 1$
Vanilla KD	64.58	74.31	54.92	65.23	31.76	38.75
Vanilla bidirectional KD	67.36	76.94	55.39	66.83	32.74	41.68
Fixed-threshold filtering $\mathcal{G}(\max p_i^w > 0.2)$	72.37	81.48	59.84	69.55	34.46	45.38
Fixed-threshold filtering $\mathcal{G}(\max p_i^w > 0.4)$	76.87	81.61	60.37	69.89	35.74	45.26
FedW2S	78.51	84.87	62.48	72.50	37.85	47.56

Table 6 evaluates FedW2S against vanilla KD, vanilla bidirectional KD, and fixed-threshold filtering across three datasets under two levels of data heterogeneity. FedW2S achieves the highest accuracy in all six settings, outperforming vanilla KD and vanilla bidirectional KD by an average of 9.04 and 7.14 percentage points, respectively. Introducing bidirectional feedback alone improves

vanilla KD by 1.90 percentage points on average, indicating that server-to-client knowledge transfer is beneficial. Fixed-threshold filtering further improves performance over both standard distillation baselines, supporting the value of selectively using weak supervision. Nevertheless, FedW2S consistently outperforms both threshold choices. Even when compared against the better of the two thresholds in each setting, FedW2S retains an average advantage of 2.32 percentage points, with gains ranging from 1.64 to 3.26 points. For example, on ImageNet, it achieves accuracies of 78.51% and 84.87% under $\vartheta = 0.1$ and $\vartheta = 1$, exceeding the corresponding best fixed-threshold results by 1.64 and 3.26 points. Similar improvements are observed on iWildCam and FMoW. These results demonstrate the effectiveness of the complete FedW2S design beyond standard knowledge distillation, unfiltered feedback, and simple confidence-based selection, with consistent benefits across the evaluated datasets and heterogeneity levels.

B.3 Experimental Comparison of Different Client Architectures

Table 7: Strong-model performance $\%$ of FedW2S under homogeneous and heterogeneous client model architectures. ResNet-18/50 denotes a heterogeneous federation in which clients use different weak-model architectures.

	Client Models	**ImageNet**		**iWildCam**		**FMoW**	
Setting	**Weak Architectures**	$\vartheta = 0.1$	$\vartheta = 1.0$	$\vartheta = 0.1$	$\vartheta = 1.0$	$\vartheta = 0.1$	$\vartheta = 1.0$
Homogeneous	ResNet-18	78.51	84.87	62.48	72.50	37.85	47.56
	ResNet-50	79.87	85.97	63.74	74.23	39.47	49.72
Heterogeneous	ResNet-18/50=1:1	79.36	84.96	63.79	74.12	37.29	48.54

We further evaluate FedW2S under model heterogeneity, where ResNet-18 and ResNet-50 clients coexist within the same federation. The basic federated method is FedMD. As shown in Table 7, FedW2S remains effective when client architectures are heterogeneous. Compared with the homogeneous ResNet-18 setting, the heterogeneous ResNet-18/50 configuration improves the strong-model performance in five out of six settings, with gains of up to 1.62 points on iWildCam under $\vartheta = 1$. More importantly, despite only a subset of clients using the larger ResNet-50 model, the heterogeneous configuration achieves performance close to the homogeneous ResNet-50 setting. For example, it obtains 63.79 vs. 63.74 on iWildCam under $\vartheta = 0.1$ and 74.12 vs. 74.23 under $\vartheta = 1.0$. Averaged across all datasets and heterogeneity levels, the heterogeneous setting achieves 64.68, compared with 63.96 for homogeneous ResNet-18 and 65.50 for homogeneous ResNet-50. These results suggest that FedW2S does not require clients to share an identical weak-model architecture and can effectively aggregate knowledge from clients with different model capacities, making it suitable for federated environments with heterogeneous computational resources.

B.4 Comparison of Experimental Results with the FedMKT

FedMKT (Fan et al., 2025) is more closely related to FedW2S as it also performs selective knowledge transfer between small client models and a large server model. However, FedMKT relies on *labeled* public data and uses ground-truth cross-entropy losses to select reliable knowledge. In contrast, FedW2S assumes an entirely *unlabeled* public dataset and determines transfer reliability from model predictions, weak-strong discrepancy, and uncertainty.

Table 8: Comparison with FedMKT on preference alignment benchmarks.

Method	**AlpacaEval 2**		**Arena-Hard**		**Avg.**
	LC	**WR**	**SC**	**WR**	
Base(Strong)	$31.5_{\pm 0.4}$	$30.6_{\pm 1.3}$	$37.3_{\pm 1.8}$	$39.4_{\pm 2.1}$	34.7
FedMKT	$40.5_{\pm 0.8}$	$47.1_{\pm 1.3}$	$50.2_{\pm 2.4}$	$58.7_{\pm 2.1}$	49.1
FedW2S	$39.5_{\pm 0.3}$	$47.3_{\pm 1.6}$	$49.6_{\pm 1.8}$	$57.5_{\pm 2.1}$	48.5

Table 8 compares FedW2S with FedMKT on preference alignment tasks. FedMKT achieves an average score of 49.1, while FedW2S obtains a comparable average score of 48.5. The two meth-

ods exhibit similar performance across all four evaluation metrics: FedW2S achieves 39.5/47.3 on AlpacaEval 2 (LC/WR) and 49.6/57.5 on Arena-Hard (SC/WR), compared with 40.5/47.1 and 50.2/58.7 for FedMKT, respectively. Notably, this comparison favors FedMKT in terms of supervision, as FedMKT relies on labeled public data to identify reliable knowledge, whereas FedW2S performs knowledge transfer using an entirely unlabeled public set. These results suggest that FedW2S can achieve competitive preference alignment without requiring labeled public data, supporting its applicability to federated settings where obtaining annotations for shared data is costly or impractical.

B.5 Communication cost

Table 9: Per-client communication cost per round under different federated learning methods.

Setting	FedAvg	FedMD	FedIT	FedSA-LoRA	FLoRG	Fed-PLoRA	HSplitLoRA	FedSLM	Ours
Vision Tasks: ResNet-18 → DINO ViT-L/14									
Comm. (MB/client/round) ↓	44.92	7.63	6.72	6.72	3.16	6.72	31.83	75.92	**44.92/7.63**
Language Tasks: Qwen2.5-3B → Qwen2.5-7B									
Comm. (MB/client/round) ↓	–	–	20.2	20.2	7.36	20.2	109.83	1373.4	**20.2**

Table 9 compares the per-client communication cost of different federated learning methods.This is the dataset configured for FMoW and UltraFeedback. In vision tasks, FedW2S can transmit prediction results via model parameters or public datasets, so each client requires 44.92/7.63 MB per round. Since the model on the device side is not a LoRA fine-tuned model, this incurs a higher parameter communication cost. For language tasks, FedW2S incurs a parameter update transmission overhead of 20.2 MB per round, which is comparable to standard LoRA-based federated fine-tuning methods such as FedIT, FedSA-LoRA, and Fed-PLoRA, but significantly lower than FedSLM. When transmitting updated parameters, no additional computational overhead is required on the client side. Overall, these results show that FedW2S does not introduce excessive communication overhead despite transferring knowledge between weak clients and the strong server model, and can further benefit from prediction-based communication when the prediction payload is smaller than the model update.

C Differences from Federated Distillation

FedW2S is related to federated distillation methods such as FedMD (Li & Wang, 2019), FedDF (Lin et al., 2020), and FedGKT (He et al., 2020), in that model outputs are used as a model-agnostic interface for transferring knowledge across distributed models. However, FedW2S targets a different learning regime. FedMD enables heterogeneous clients to learn from a consensus prediction on shared public data, while FedDF performs server-side ensemble distillation to fuse knowledge from client models. FedGKT trains a server-side large model through alternating knowledge transfer. These methods primarily address model fusion, heterogeneous collaboration, or resource-efficient distributed training. In contrast, FedW2S formulates federated learning as a *weak-to-strong* problem: the server already possesses a pretrained strong model, whereas clients only train lightweight weak models. Since weak predictions may provide unreliable supervision to an already strong model, FedW2S explicitly controls *whether* and *how much* weak knowledge should be transferred through reliability filtering and discrepancy-aware target mixing, together with strong-to-weak feedback.

D Introduction to the Algorithm

Algorithm 1 summarizes the overall training procedure of FedW2S. Each round begins with local optimization of the lightweight weak models, followed by the aggregation step of the underlying federated method. After the warm-up stage, FedW2S performs bidirectional knowledge transfer through the public dataset: the server first aggregates client knowledge to construct weak supervision and updates the strong model using reliability-based filtering and discrepancy-aware target mixing; the resulting strong-model knowledge is then fed back to clients through compatibility-aware strong-to-weak distillation in the subsequent round. Notably, FedW2S is agnostic to the communication interface among clients and the server. Depending on the base FL method and communication

cost, clients can either transmit model parameters/updates, from which the server obtains the aggregated weak model and evaluates it on the public data, or directly communicate logits/predictive distributions on the public data. In practice, the more communication-efficient representation can be adopted without changing the core weak-to-strong learning procedure.

Algorithm 1 FedW2S

Require: Private datasets $\{\mathcal{D}_k\}_{k=1}^K$, public dataset $\mathcal{D}_p$, client weak models $\{f_k^w\}_{k=1}^K$, server strong model f^s, communication rounds R, warm-up rounds R_{warm}.
Ensure: Optimized server strong model f^s.
Initialize global weak model f^w. The strong model f^s uses LoRA or partial freezing.
for $t = 0, \ldots, R-1$ **do**
  Sample participating clients $\mathcal{S}_t$.
  for all $k \in \mathcal{S}_t$ **do**
    Update the client model using the base loss $\mathcal{L}_{\text{base}}^k$.
    if $t > R_{\text{warm}}$ **then**
      Compute compatibility-aware absorption weights between strong and weak predictions.
      Optimize weak model with local task loss and strong-to-weak feedback loss: $\mathcal{L}_k = \mathcal{L}_{\text{base}}^k + \lambda \mathcal{L}_{\text{S2W}}$.
    end if
  end for
  Aggregate client models/updates or public-data predictions according to the base FL method.
  if $t > R_{\text{warm}}$ **then**
    Construct global weak knowledge $\boldsymbol{p}^w$ from the aggregated weak model or client predictions.
    Perform reliability-based filtering to remove unreliable weak supervision.
    Construct discrepancy-aware mixed targets by balancing weak knowledge and strong model prior knowledge.
    Update server strong model using the weak-to-strong objective $\mathcal{L}_{\text{W2S}}$.
  end if
  Generate strong model predictions $\boldsymbol{q}^s$ on public data.
end for
return Strong model f^s

E Proofs of the Theoretical Results

We use natural logarithms throughout. Server targets and selection indicators are fixed during optimization. We consider $|S| > 0$; if S is empty, the server distillation update is skipped.

E.1 Target fitting and prediction retention

Let $z = \beta u + (1-\beta)w$. Expanding the definition of KL divergence gives

$$\begin{aligned} &\beta D_{\text{KL}}(u\|p) + (1-\beta) D_{\text{KL}}(w\|p) \\ &= D_{\text{KL}}(z\|p) + \beta D_{\text{KL}}(u\|z) + (1-\beta) D_{\text{KL}}(w\|z). \end{aligned} \tag{22}$$

The last two terms are constant with respect to the trainable prediction p. Therefore, fitting z is equivalent to weak-teacher fitting with weight $1-\beta$ and cached-server retention with weight β.

E.2 Gain of the mixed target

For any one-hot label e_y, the squared-norm identity gives

$$\begin{aligned} \|z_i - e_y\|_2^2 = \beta_i \|u_i - e_y\|_2^2 + (1-\beta_i)\|w_i - e_y\|_2^2 \\ - \beta_i(1-\beta_i)\|u_i - w_i\|_2^2. \end{aligned} \tag{23}$$

Taking the conditional expectation over Y given $X = x_i$ yields

$$b_i(z_i) = \beta_i b_i(u_i) + (1-\beta_i) b_i(w_i) - \beta_i(1-\beta_i)\|u_i - w_i\|_2^2. \tag{24}$$

Subtracting from $b_i(u_i)$ proves equation 12. Since $1 - \beta_i > 0$ on S, the gain is positive exactly when

$$b_i(w_i) - b_i(u_i) < \beta_i \|u_i - w_i\|_2^2.$$

The weak teacher's risk disadvantage must therefore be smaller than the benefit provided by mixing.

E.3 Proof of Theorem 1

Step 1: Bound the risk change by the prediction change. For probability vectors p, q and any one-hot label e_y,

$$\big|\|p - e_y\|_2^2 - \|q - e_y\|_2^2\big| \le (\|p - e_y\|_2 + \|q - e_y\|_2)\|p - q\|_2.$$

Each distance to e_y is at most $\sqrt{2}$. Taking the conditional expectation over Y therefore gives

$$|b_i(p) - b_i(q)| \le 2\sqrt{2}\|p - q\|_2. \tag{25}$$

We also need to relate prediction error to the KL fitting loss. For $h = p - q$, the positive and negative parts each have total mass $\|h\|_1/2$. Consequently, $\|h\|_2^2 \le \|h\|_1^2/2$. Pinsker's inequality then implies

$$\|p - q\|_2^2 \le \tfrac{1}{2}\|p - q\|_1^2 \le D_{\mathrm{KL}}(q\|p). \tag{26}$$

Step 2: Bound the selected samples. For $i \in S$, insert the mixed target between the initial and final predictions:

$$b_i(v_i) - b_i(u_i) = [b_i(v_i) - b_i(z_i)] - [b_i(u_i) - b_i(z_i)].$$

The second bracket is the ideal target gain. For the first bracket, equation 25 and equation 26 give

$$|b_i(v_i) - b_i(z_i)| \le 2\sqrt{2}\sqrt{D_{\mathrm{KL}}(z_i\|v_i)}.$$

Averaging and using the concavity of the square root,

$$\begin{aligned} \frac{1}{n_p}\sum_{i\in S}[b_i(v_i) - b_i(u_i)] &\le -\rho A_S + \frac{2\sqrt{2}\rho}{n_S}\sum_{i\in S}\sqrt{D_{\mathrm{KL}}(z_i\|v_i)} \\ &\le -\rho A_S + 2\sqrt{2}\rho\sqrt{\delta_S}. \end{aligned} \tag{27}$$

Step 3: Account for unselected samples. Training on S can also change predictions outside S through shared parameters. Applying equation 25 directly gives

$$\frac{1}{n_p}\sum_{i\notin S}[b_i(v_i) - b_i(u_i)] \le \frac{2\sqrt{2}}{n_p}\sum_{i\notin S}\|v_i - u_i\|_2 = 2\sqrt{2}D_{\bar{S}}. \tag{28}$$

Adding the selected and unselected bounds proves equation 14. When all public samples are selected, the unselected sum is zero. The current selected-only training loss does not explicitly penalize this drift.

E.4 Proof of Proposition 1

Step 1: Bound confidence and prediction differences. Set $T_i = \frac{1}{2}\|p_i - q_i\|_1$, the total variation distance. If j maximizes $q_{i,j}$, then $\max_c q_{i,c} - \max_c p_{i,c} \le q_{i,j} - p_{i,j} \le T_i$. Hence

$$[\max_c q_{i,c} - \max_c p_{i,c}]_+ \le T_i, \qquad \|p_i - q_i\|_2 \le \sqrt{2}T_i, \tag{29}$$

where the second inequality uses the zero-sum argument from equation 26.

Step 2: Relate total variation to JS divergence. Let $m_i = (p_i + q_i)/2$. The total variation distance from either p_i or q_i to m_i is $T_i/2$. By Pinsker's inequality,

$$d_i = \tfrac{1}{2}D_{\mathrm{KL}}(p_i\|m_i) + \tfrac{1}{2}D_{\mathrm{KL}}(q_i\|m_i) \ge \tfrac{1}{2}T_i^2. \tag{30}$$

Using the definition of a_i and $0 \le d_i \le \log 2$, we obtain

$$\begin{aligned} a_i \|p_i - q_i\|_2 &\le \sqrt{2} T_i^2 (1 - d_i/\log 2) \\ &\le 2\sqrt{2} d_i (1 - d_i/\log 2) \le \frac{\log 2}{\sqrt{2}}. \end{aligned} \tag{31}$$

The last inequality follows from $d(1 - d/\log 2) \le (\log 2)/4$ for $d \in [0, \log 2]$.

Step 3: Apply the softmax gradient formula. Because the target q_i and weight a_i are detached,

$$\nabla_{\ell_i} [a_i D_{\mathrm{KL}}(q_i \| p_i)] = a_i (p_i - q_i).$$

Combining this identity with Step 2 proves equation 15.

If the gate a_i^0 is cached at p_i^0 and later used at p_i while q_i remains fixed, define $d_i^0 = D_{\mathrm{JS}}(p_i^0, q_i)$. The triangle inequality gives the required correction:

$$a_i^0 \|p_i - q_i\|_2 \le 2\sqrt{2} d_i^0 (1 - d_i^0/\log 2) + a_i^0 \|p_i - p_i^0\|_2. \tag{32}$$

Thus the stated bound applies directly to gates evaluated at the current prediction; cached gates require the additional prediction-drift term.