FedGRPO: Privately Optimizing Foundation Models with Group-Relative Rewards from Domain Clients

Gongxi Zhu[1], Hanlin Gu[2], Lixin Fan[2], Qiang Yang[3], Yuxing Han[1]*

[1]Shenzhen International Graduate School, Tsinghua University
[2]AI Lab, Webank
[3] Department of Data Science and Artificial Intelligence, The Hong Kong Polytechnic University
gx.zhu@foxmail.com, allengu@webank.com, yuxinghan@sz.tsinghua.edu.cn

Abstract

One important direction of Federated Foundation Models (FedFMs) is leveraging data from small client models to enhance the performance of a large server-side foundation model. Existing methods based on model level or representation level knowledge transfer either require expensive local training or incur high communication costs and introduce unavoidable privacy risks. We reformulate this problem as a reinforcement learning style evaluation process and propose FedGRPO, a privacy preserving framework comprising two modules. The first module performs competence-based expert selection by building a lightweight confidence graph from auxiliary data to identify the most suitable clients for each question. The second module leverages the "Group Relative" concept from the Group Relative Policy Optimization (GRPO) framework by packaging each question together with its solution rationale into candidate policies, dispatching these policies to a selected subset of expert clients, and aggregating solely the resulting scalar reward signals via a federated group–relative loss function. By exchanging reward values instead of data or model updates, FedGRPO reduces privacy risk and communication overhead while enabling parallel evaluation across heterogeneous devices. Empirical results on diverse domain tasks demonstrate that FedGRPO achieves superior downstream accuracy and communication efficiency compared to conventional FedFMs baselines.

Code — https://github.com/Liar-Mask/FedGRPO

1 Introduction

Federated Foundation Models (FedFMs) (Fan et al. 2025; Kang et al. 2023; Ren et al. 2025) present a promising paradigm that integrates the strong generalization capabilities of server-side Foundation Models (FMs) with the domain-specific expertise of client devices. One important goal in FedFMs is how to effectively leverage clients' domain knowledge to enhance the performance of FMs while preserving the privacy of local data.

Current methods for integrating domain knowledge from downstream clients into a server-side foundation model (FM) fall into two main categories: model-level transfer (Fan et al. 2023; Zhang et al. 2023) and synthetic data-level transfer (Yu et al. 2023; Abacha et al. 2024). In model-level transfer, each client fine-tunes a server-provided small portion of model parameters or an adapter on its local data and sends the updated parameters back for aggregation. In synthetic data-level transfer, clients generate domain-pertinent synthetic data, which is then uploaded to the server to enhance the FM's performance. Both approaches impose substantial communication overhead and expose sensitive information: the frequent exchange of parameters or synthetic data not only strains network bandwidth (Zhao et al. 2024) but also invites semi-honest adversaries to infer private data from the transmitted artifacts (Chen et al. 2024).

In order to reduce privacy leaking risk and communication overhead, we take a novel approach by which only model evaluation scores are transferred from clients to the server. Compared with model parameters or synthetic data, evaluation scores are orders of magnitude smaller in terms of amount of information to be transmitted, and moreover, the risk of privacy leaking is substantially reduced. In this framework, the server leverages evaluations from domain-specific clients on distributed problems and candidate solutions to simultaneously enhance model performance while preserving data privacy. *A fundamental requirement of this framework is that client evaluations should be accurate.* Specifically, it introduces two critical challenges: (1) how to select the most suitable clients for evaluating specific problems, based on their domain expertise; and (2) how to aggregate evaluations from multiple expert clients to effectively improve the foundation model.

To address these challenges, we propose FedGRPO, a federated-foundation-models (FedFMs) framework that integrates two complementary modules: (i) competence-based expert selection and (ii) Group-Relative Reward Aggregation. The first module constructs a confidence graph from auxiliary data to quantify each client's expertise on domain-specific queries, enabling the server to recruit the most competent subset of experts for every request. Building on "Group Relative" idea of Group Relative Policy Optimization (GRPO), the second module computes relative rewards among the selected experts to guide server-side updates. Concretely, the server maintains a global policy and periodically dispatches enhanced candidate policies—each encapsulating the original question alongside a detailed problem-

*Corresponding author.

solving rationale—to the chosen expert clients. Each client evaluates the received policy on its private in-domain data and returns only a scalar reward signal, thereby preserving data privacy. The server then aggregates these signals through a federated group-relative loss function, which balances contributions from multiple experts and iteratively refines the foundation model. FedGRPO offers three key advantages: (1) it leverages domain-specific user data to improve LLM training; (2) it enhances data privacy by exchanging only reward signals rather than raw data or model parameters; and (3) it improves computational efficiency by enabling parallel reward evaluation across multiple Clients. Our contributions are summarized as follows:

- We introduce FedGRPO, a novel reinforcement-learning-inspired FedFM pipeline that recasts large-model refinement as a reward-based evaluation process. By packaging each query with its solution rationale into candidate policies and leveraging a lightweight confidence graph for competence-based expert selection, FedGRPO enables efficient, parallelized evaluation across resource-constrained clients without requiring expensive local fine-tuning.
- We design a federated Group Relative Policy Optimization module that aggregates only scalar reward signals from selected expert clients via a group-relative loss. This mechanism eliminates the need to transmit raw data, model updates, or high-dimensional representations—substantially reducing communication overhead and mitigating privacy leakage.
- Extensive experiments show that FedGRPO: 1) enhances server-model reasoning using client reward feedback and group-relative loss, closely matching centralized GRPO performance; 2) effectively leverages heterogeneous domain expertise through competence-based expert selection even without ground-truth answers; and 3) avoids privacy leakage risks inherent in model-level or synthetic data-level transfer while maintaining smaller orders of magnitude communication overhead.

2 Related Work

2.1 Knowledge Transfer in FedFMs

Federated Foundation Models (FedFMs) (Fan et al. 2025; Kang et al. 2023; Ren et al. 2025) constitute a distributed learning paradigm facilitating the reciprocal exchange and adaptation of knowledge between server-hosted Foundation Models (FMs) and the domain-specific expertise residing on client devices. Existing methodologies for client-to-server knowledge transfer within the FedFMs framework can be broadly bifurcated into two primary categories: model-level and data-level transfer.

In the context of model-level transfer, the substantial scale of foundation models precludes clients from training the entire model locally for subsequent uploading and aggregation. To circumvent this limitation, researchers have extended existing Parameter-Efficient Fine-Tuning (PEFT) techniques to the federated learning setting, a methodology termed FedPEFT (Sun et al. 2024; Yi et al. 2023; Zhang et al. 2023). Within this framework, clients perform lightweight PEFT on the FM locally and subsequently transmit only the trained PEFT modules or adapters to the server. These components are then aggregated and integrated with the global FM.

Regarding data-level transfer schemes, a prevalent approach involves clients generating domain-pertinent synthetic data (Li et al. 2024a; Abacha et al. 2024; Hou et al. 2025). This synthetic data encapsulates domain knowledge while ostensibly preserving the privacy of the original dataset from which it was derived.The synthesized data is then uploaded to the server to enhance the FM's performance through further training. Such schemes can be augmented with differential privacy mechanisms to fortify privacy preservation.

Notwithstanding the efficacy of the aforementioned schemes in transferring domain-specific knowledge from clients to the server, the frequent transmission of adapters or synthetic data imposes considerable communication overhead (Zhao et al. 2024) and presents inherent risks of privacy leakage (Chen et al. 2024).

2.2 Reinforcement Learning for LLM Reasoning

Advances in LLM research have shifted focus from basic autoregressive token generation to complex reasoning tasks like mathematical problem-solving and code generation. Reinforcement learning (RL) enhances model reasoning through trial-and-error optimization, emerging as a key method for post-training LLMs (Chu et al. 2025). Proximal Policy Optimization (PPO) (Schulman et al. 2017) is a well-known traditional RL method that is widely used due to its robustness across various fields. However, it encounters challenges such as the need for costly online data collection, high computational overhead, and sensitivity to hyperparameters. Subsequently, offline RL methods, such as Direct Preference Optimization (DPO) (Rafailov et al. 2023), were proposed to eliminate the need for live interaction or data generation and allow users to tune the model more efficiently. Deepseek developed Group Relative Policy Optimization (GRPO) (Shao et al. 2024) method, which abandons the value model in favor of estimating the baseline based on group scores, significantly reducing training resource requirements. The advantages of this approach were further emphasized with its application on Deepseek-R1.

3 The Proposed Method

3.1 Problem Formulation

Setting. We consider a collaborative learning framework comprising a central server and K distributed clients. Each client $k \in \{1, \ldots, K\}$ possesses a local dataset $\mathcal{D}_k = \{(x_i^{(k)}, y_i^{(k)})\}_{i=1}^{n_k}$, where $x_i^{(k)} \in \mathcal{X}$ represents the input instance and $y_i^{(k)} \in \mathcal{Y}$ denotes the corresponding output. For instance, $x_i^{(k)}$ can be a question and $y_i^{(k)}$ its solution.

Each client trains a model on its domain-specific dataset $\mathcal{D}_k$ to have their domain-specific model θ_k. The central server hosts a large-scale pre-trained foundation model, denoted by θ_g, with the goal of leveraging domain knowledge from clients' private datasets. Suppose the central server

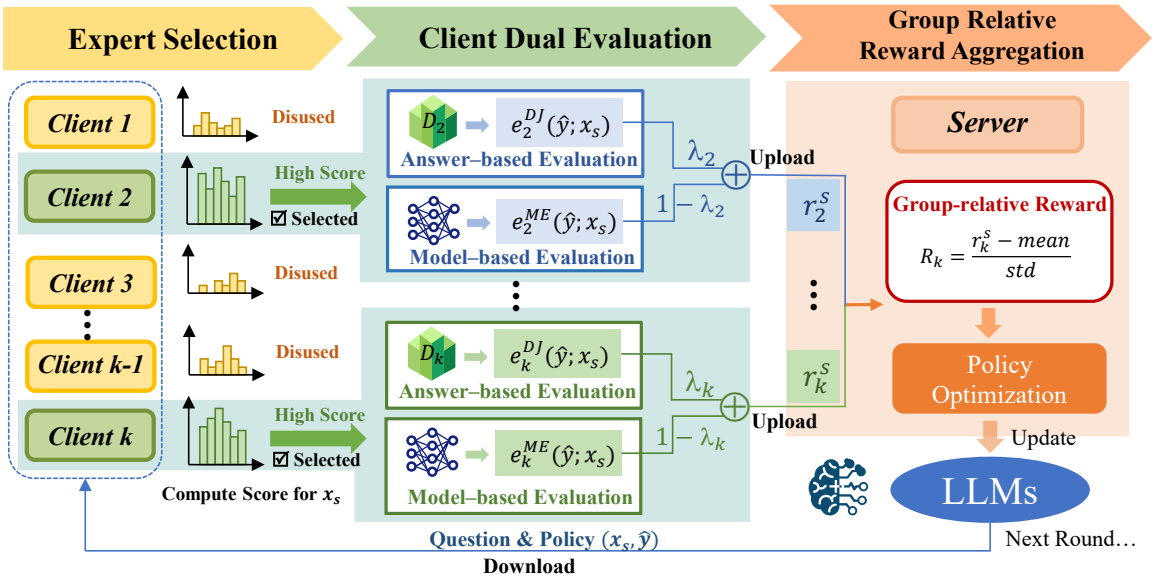


Figure 1: Overview of FedGRPO including three steps: 1) Expert selection to select an appropriate expert subset $\mathcal{C}(x_s)$ for every question x_s; 2) Dual evaluation on the select client $k \in \mathcal{C}(x_s)$ to compute rewards r_k^s for the policy $\hat{y}$; and 3) Group relative reward aggregation on server to get group-relative reward R_k and perform policy optimization to update LLMs.

holds only a negligible amount of auxiliary data, denoted by $\mathcal{D}_p = \{(x_{p,j}, y_{p,j})\}_{j=1}^{|\mathcal{D}_p|}$. The server aims to optimize:

$$\min_{\theta_g} \ell(\theta_g, \mathcal{D}_1, \ldots, \mathcal{D}_K, \mathcal{D}_p), \tag{1}$$

to enhance the utility of θ_g on domain-specific data. However, privacy constraints prohibit the server from directly accessing clients' raw data.

Threat model. We assume an honest-but-curious threat model, where the central server and participating clients follow the prescribed protocol but may attempt to infer additional information from the received messages. Existing approaches typically transmit models, gradients, or embeddings rather than raw data to the server; nevertheless, these strategies remain susceptible to privacy leakage (Zhu, Liu, and Han 2019).

Formulation. Motivated by reinforcement learning (RL), we propose to treat each client's model as a reward model that evaluates the server's generated outputs. In this design, clients provide only scalar reward signals to the server, thereby preventing the leakage of raw data. We formulate the server's optimization as an RL problem. Specifically, the server maintains a policy π_{θ_g}, parameterized by θ_g, which, given an input $x \in \mathcal{X}$ (e.g., a question), generates a candidate output $\hat{y} \in \mathcal{Y}$ (e.g., an answer or solution):

$$\hat{y} \sim \pi_{\theta_g}(\cdot|x). \tag{2}$$

Each client k evaluates the generated output $\hat{y}$ with respect to its local domain knowledge or ground-truth labels in $\mathcal{D}_k$, and returns a scalar reward:

$$r_k(\hat{y}, x) = f_{\theta_k}(\hat{y}; \mathcal{D}_k), \tag{3}$$

where f_{θ_k} denotes the reward function that derives reward score from the local data or client's locally trained model θ_k^* on dataset $\mathcal{D}_k$. The server aggregates individual client rewards $\{r_k(\hat{y}, x)\}_{k=1}^K$ through an aggregation function $A(\cdot)$, resulting in a global reward $R(\hat{y}|x)$. Consequently, the server's optimization objective can be expressed as:

$$\begin{aligned} J(\theta_g) &= \mathbb{E}_{x\sim\mathcal{X},\, \hat{y}\sim\pi_{\theta_g}(\cdot|x)}\Big[R(\hat{y}|x)\Big]. \\ &= \mathbb{E}_{x\sim\mathcal{X},\, \hat{y}\sim\pi_{\theta_g}(\cdot|x)}\Big[A(\{r_k(\hat{y}, x)\}_{k=1}^K)\Big]. \end{aligned} \tag{4}$$

However, this formulation poses two key challenges: (1) determining how to select suitable clients for evaluating specific tasks, considering the heterogeneity in their domain expertise; and (2) developing effective strategies to aggregate evaluations from multiple expert clients to enhance the performance of the foundation model.

3.2 FedGRPO

To tackle the aforementioned dual challenges, we propose FEDGRPO, which integrates two tightly coupled modules: *expert client selection* and *group-relative policy optimization* (GRPO). The expert-selection module estimates each client's competency on the auxiliary data and selects an appropriate subset of expert clients for every question; the GRPO module then transforms the raw scores provided by experts into a unified, scale-invariant reinforcement signal.

Competence-based expert selecting. For each unlabeled question (no answer) $x_s \sim \mathcal{D}_s$, the server (i) samples a provisional answer $\hat{y} \sim \pi_{\theta_g}(\cdot \mid x_s)$ and (ii) embeds the question with a frozen encoder ϕ derived from the foundation model, producing $\mathbf{z}_s = \phi(x_s) \in \mathbb{R}^d$. Based on the cosine similarity between $\mathbf{z}_s$ and auxiliary embeddings $\{\mathbf{z}_{p,j}\}_{j=1}^{|\mathcal{D}_p|} = \{\phi(x_{p,j})\}_{j=1}^{|\mathcal{D}_p|}$, the server retrieves the L most

similar labeled exemplars

$$\mathcal{G}(x_s) = \big\{(x_{p,\ell}, y_{p,\ell})\big\}_{\ell=1}^{L}, \quad \langle \mathbf{z}_s, \mathbf{z}_{p,1}\rangle \geq \cdots \geq \langle \mathbf{z}_s, \mathbf{z}_{p,L}\rangle. \tag{5}$$

The server distributes $\mathcal{G}(x_s)$ to all clients evaluate. Each client maintains a running accuracy estimate on auxiliary prompts; the server therefore possesses a *competence score* $r_k^p(x_s) \in [0,1]$ for every client k with respect to the neighborhood equation (5) as:

$$r_k^p = \frac{1}{L}\sum_{\ell=1}^{L} \mathbf{1}\big[e_k(y_{p,\ell}; x_{p,\ell}) = y_{p,\ell}\big].$$

It retains the M highest-scoring experts for each question x_s:

$$\mathcal{C}(x_s) = \underset{k}{\text{Top-M}}\big\{r_k^p(x_s)\big\}. \tag{6}$$

Algorithm 1: FEDGRPO training with competence-based expert selection

Input: auxiliary prompt set $\mathcal{D}_p$, unlabeled queries $\mathcal{D}_s$, global policy π_{θ_g}, client set $\mathcal{K}$; neighbourhood size L, experts per question M; learning rate η, epochs T

Pre-compute ℓ_2-normalised embeddings $\mathbf{z}_{p,j} = \phi(x_{p,j})$ for all $(x_{p,j}, y_{p,j}) \in \mathcal{D}_p$
for $t = 1$ **to** T **do**
  Sample question $x_s \sim \mathcal{D}_s$ and provisional answer $\hat{y} \sim \pi_{\theta_g}(\cdot \mid x_s)$
  Compute question embedding $\mathbf{z}_s = \phi(x_s)$
  Retrieve L nearest auxiliary exemplars $\mathcal{G}(x_s) = \{(x_{p,\ell}, y_{p,\ell})\}_{\ell=1}^{L}$
  Broadcast $\mathcal{G}(x_s)$ to all clients $k \in \mathcal{K}$
  for clients $k \in \mathcal{K}$ **in parallel do**
    Compute competence score $r_k^p(x_s) = \frac{1}{L}\sum_{\ell=1}^{L} \mathbf{1}\big[e_k(y_{p,\ell}; x_{p,\ell}) = y_{p,\ell}\big]$
    Send $r_k^p(x_s)$ to server
  end for
  Select experts $\mathcal{C}(x_s) = \text{Top-M}_k\{r_k^p(x_s)\}$ {Eq. equation (6)}
  Broadcast $\langle x_s, \hat{y}\rangle$ to experts $k \in \mathcal{C}(x_s)$
  for all experts $k \in \mathcal{C}(x_s)$ **in parallel do**
    Choose pathway $\lambda_k(x_s) \in \{0,1\}$ (AE if ground-truth exists, else ME)
    Compute private score $r_k^s = \lambda_k e_k^{\text{AE}}(\hat{y}; x_s) + (1 - \lambda_k) e_k^{\text{ME}}(\hat{y}; x_s)$
    Send r_k^s to server
  end for
  Compute group-relative reward

$$\mu_r = \frac{1}{M}\sum_{k\in\mathcal{C}} r_k^s, \quad \sigma_r = \sqrt{\frac{1}{M}\sum_{k\in\mathcal{C}}(r_k^s - \mu_r)^2}, \quad R_k = \frac{r_k^s - \mu_r}{\sigma_r + \epsilon} \tag{7}$$

  Update global policy $\theta_g \leftarrow \theta_g + \eta\, R_k\, \nabla_{\theta_g} \log \pi_{\theta_g}(\hat{y} \mid x_s)$
end for

Dual evaluation on the clients. For each training iteration, the server broadcasts the $\langle x_s, \hat{y}\rangle$ solely to the selected expert set $\mathcal{C}(x_s)$. Upon receiving the triplet $\langle x_s, \hat{y}, \mathcal{G}(x_s)\rangle$, each client k produces two scalar feedback signals: *private score* r_k^s for the unlabeled question and *auxiliary score* r_k^p for its retrieved neighbourhood. The precise mechanism depends on which evaluation criteria the client can apply.

(i) **Answer–based evaluation (AE).** If the question x_s (or a paraphrase thereof) appears in the client's private corpus $\mathcal{D}_k$ together with a trusted ground-truth answer $\tilde{y}_k(x_s)$, the client performs an *exact-answer check*:

$$e_k^{\text{AE}}(\hat{y}; x_s) \;=\; \mathbf{1}\big[\,\hat{y} = \tilde{y}_k(x_s)\,\big].$$

The resulting score is binary and therefore directly comparable across all AE clients.

(ii) **Model–based evaluation (ME).** If the question is *not* covered by $\mathcal{D}_k$, the client falls back to its self-trained reward model $f_{\theta_k^\star}$, yielding a real-valued score

$$e_k^{\text{ME}}(\hat{y}; x_s) \;=\; f_{\theta_k^\star}(\hat{y}; x_s) \in \mathbb{R}.$$

Typical choices for $f_{\theta_k^\star}$ include a small cross-entropy classifier or a learned rubric rubricator.

In practice a single client may support *both* criteria and dynamically select the appropriate pathway via a gating indicator $\lambda_k(x_s) \in \{0,1\}$:

$$e_k(\hat{y}; x_s) \;=\; \lambda_k(x_s)\, e_k^{\text{AE}}(\hat{y}; x_s) + \big(1 - \lambda_k(x_s)\big)\, e_k^{\text{ME}}(\hat{y}; x_s).$$

The private score is then $r_k^s = e_k(\hat{y}; x_s)$. This dual-evaluation design lets every client exploit the strongest knowledge source at its disposal (ground-truth answers when available, otherwise a learned evaluator) while providing the server with a question-specific competence signal r_k^p and a raw reward r_k^s suitable for GRPO aggregation.

Remark 1. *We also incorporate a format reward to ensure that the generated answers conform to the expected type, following the approach in (Shao et al. 2024).*

Group Relative Policy Optimization. The server retains only the M highest-scoring clients

$$\mathcal{C}(x_s) = \underset{k}{\text{Top-M}}\{\, r_k^p\},$$

disregarding the rest. This question-adaptive pruning ensures that subsequent optimization relies on the most competent—and therefore most reliable—experts. Let $\{r_k^s\}_{k\in\mathcal{C}}$ be the private scores returned by the selected experts. Define

$$\mu_r = \frac{1}{M}\sum_{k\in\mathcal{C}} r_k^s, \quad \sigma_r = \sqrt{\frac{1}{M}\sum_{k\in\mathcal{C}}(r_k^s - \mu_r)^2}, \tag{8}$$

$$R_k(x_s, \hat{y}) = \frac{r_k^s - \mu_r}{\sigma_r + \epsilon}. \tag{9}$$

Standardising in this manner eliminates scale discrepancies across evaluation modes and dampens outliers, yielding the group-relative reward R_k.

Finally, the server performs a policy-gradient step

$$\theta_g \;\leftarrow\; \theta_g + \eta\, R_k(x_s, \hat{y})\, \nabla_{\theta_g} \log \pi_{\theta_g}(\hat{y} \mid x_s),$$

thus reinforcing responses that outperform expert-group average and steadily improving π_{θ_g} on unlabeled server data.

Remark 2. *In this work, the term "group" denotes the aggregation of reward signals from multiple clients and corresponds to the ensemble of candidate policies in GRPO (Shao et al. 2024). To facilitate understanding, we use $\hat{y}$ to denote a single policy, whereas it could represents a collection of policies.*

4 Experimental Results

This section presents the empirical analysis of the proposed FedGRPO from comparison with baselines, communication efficiency and ablation studies.

4.1 Experimental Setup

Datasets & Models. We conduct experiments on two classic kinds of LLM tasks including math problem-solving and question-answering:

- For math problem-solving task, we use MATH-benchmark (Hendrycks et al. 2021) and OpenR1-Math (Hugging Face 2025) as training sets, and three different sizes of LLMs (Qwen2.5-3B (Yang et al. 2024a), Qwen2.5-Math-1.5B and Qwen2.5-Math-7B (Yang et al. 2024b)) as the server's foundation models. We evaluate the performance of the model on six widely used math problem test sets, including MATH500 (Hendrycks et al. 2021), Minerva (Lewkowycz et al. 2022), OlympiadBench (He et al. 2024), AIME 2024, AIME 2025, and AMC (Li et al. 2024b).
- For question-answering task, we apply 2WikiMultiHopQA (Ho et al. 2020) as training set and two LLMs (Qwen2.5-1.5B,Qwen2.5-3B (Yang et al. 2024a)) as server's foundation models. The performance is evaluated on the test sets of HotpotQA (Yang et al. 2018), 2WikiMultiHopQA (Ho et al. 2020) and MuSiQue (Trivedi et al. 2022) . Due to space limit, the experimental results of question-answering task are left in Appendix.

Each result presented is the average of three repeated experiments. For efficiency, 2,000 samples are randomly selected from the training set for each experiment.

Setting. This paper considers two settings for the dataset $\mathcal{D}_s$: 1) none of the K clients possess the ground-truth label $y(x_s)$ for samples in $\mathcal{D}_s$, so clients rely solely on model-based evaluation (see result in Sect. 4.3); 2) some clients have access to the ground-truth label $y(x_s)$, and thus can perform answer-based evaluation (see result in Sect. 4.2). We evaluate scenarios involving 4 to 20 clients in federated learning (FL), with up to 320 communication rounds. FedGRPO is assessed under both IID and Non-IID conditions. For the Non-IID setting, we simulate data heterogeneity using the Dirichlet distribution, $\mathrm{Dir}(\beta)$, with $\beta = 0.1$.

Baselines. To benchmark FedGRPO against other client-to-server knowledge transfer approaches in federated foundation models (FedFMs), we consider two representative methods: (1) **Fedpetuning** (Zhang et al. 2023) (model-level transfer): clients perform parameter-efficient local tuning and upload only the adapted modules or adapters to the server; (2) **DPSDA-FL** (Abacha et al. 2024) (synthetic data-level transfer): clients use a small foundation model to generate differentially private synthetic data from their local datasets, which are then sent to the server for centralized training. For both baselines, we evaluate models trained with both GRPO and supervised fine-tuning (SFT) objectives.

Furthermore, we compare FedGRPO with the following baselines: (3) **Zero-shot**: the direct performance of pre-trained LLMs with zero sample post-training, serving as a baseline to measure improvement; (4) **Central-GRPO** (Shao et al. 2024): the group relative policy optimization (GRPO) method proposed by Deepseek, applied in a centralized setting where all training data are aggregated, representing the ideal upper bound for FedGRPO.

Implementation & Evaluation Metric. The experiments are configured with a learning rate of 3.0×10^{-6} and a temperature of 0.7. Each question involved sampling 8 candidate policies for evaluation, with the maximum generation length limited to 2048 tokens. The auxiliary data owned by server has 100 samples and expert selection parameter L is set as 20, M is 2. More settings can be seen in Appendix.

We adopt Pass@1 accuracy on the test sets as our primary evaluation metric, following the methodology of (Shao et al. 2024) and a higher value indicates better performance. Additionally, we assess the communication overhead of the complete training process, measured in megabytes (MB), as an efficiency metric and the lower value means more efficiency.

4.2 Comparison with Baselines

We conduct a comprehensive performance evaluation of our proposed **FedGRPO** framework against a range of baselines, including FedFMs methods (model-level transfer method FedPETuning (Zhang et al. 2023) and synthetic data-level transfer method DPSDA-FL (Abacha et al. 2024)) and the centralized GRPO method. The detailed results, presented in Table 1, are evaluated across three model scales (1.5B, 3B, and 7B). Our analysis confirms the superior performance and effectiveness of FedGRPO.

Comparison with FedFMs Baselines. When benchmarked against existing FedFMs methods, FedGRPO consistently demonstrates significant and stable performance gains. Our approach substantially outperforms both Fedpetuning (model-level transfer) and DPSDA-FL (synthetic data-level transfer), irrespective of their underlying optimization objectives (GRPO or SFT). This superiority holds across all model sizes and both datasets. For instance, using the Qwen2.5-Math-7B model on the Math-benchmark, FedGRPO attains an average accuracy of 0.369, marking a substantial improvement over the best-performing federated baseline, DPSDA-FL+SFT (0.253). The performance advantage is also evident on smaller models; for the Qwen2.5-Math-1.5B model on Math-benchmark, FedGRPO (0.338) is clearly ahead of the next best federated method, DPSDA-FL+GRPO (0.275). These robust results validate that FedGRPO provides a more effective framework for leveraging clients' domain knowledge to enhance the performance of FMs compared to other FedFMs methods.

Dataset	Model	Method	Math	Minerva	AMC	Olympiad	AIME24	AIME25	Avg.
Math-benchmark	Qwen2.5-Math-1.5B	Zero-shot	0.316	0.081	0.272	0.203	0.074	0.034	0.163
		Fedpetuning+GRPO	0.342	0.121	0.279	0.233	0.072	0.048	0.183
		Fedpetuning+SFT	0.480	0.088	0.294	0.220	0.068	0.038	0.198
		DPSDA-FL+GRPO	0.630	0.275	0.285	0.067	**0.310**	**0.083**	0.275
		DPSDA-FL+SFT	0.498	0.129	0.284	0.260	0.079	0.049	0.217
		Central-GRPO	0.701	0.307	0.440	**0.353**	0.081	0.056	0.323
		FedGRPO	**0.716**	**0.313**	**0.451**	0.348	0.133	0.065	**0.338**
	Qwen2.5-3B	Zero-shot	0.118	0.092	0.063	0.046	0.006	0.006	0.055
		Fedpetuning+GRPO	0.384	0.129	0.156	0.132	0.017	0.006	0.137
		Fedpetuning+SFT	0.318	0.085	0.100	0.061	0.006	0.003	0.096
		DPSDA-FL+GRPO	0.448	0.148	0.099	0.176	**0.029**	**0.009**	0.152
		DPSDA-FL+SFT	0.428	0.134	0.073	**0.197**	0.027	0.007	0.144
		Central-GRPO	**0.474**	**0.210**	0.157	0.153	0.010	0.003	**0.168**
		FedGRPO	0.438	0.143	**0.195**	0.157	0.024	0.005	0.160
	Qwen2.5-Math-7B	Zero-shot	0.426	0.121	0.326	0.163	0.111	0.048	0.199
		Fedpetuning+GRPO	0.460	0.107	0.329	0.132	0.049	0.038	0.186
		Fedpetuning+SFT	0.400	0.107	0.182	0.132	0.049	0.018	0.148
		DPSDA-FL+GRPO	0.714	**0.323**	0.432	0.308	0.087	0.064	0.321
		DPSDA-FL+SFT	0.574	**0.323**	0.343	0.187	0.067	0.026	0.253
		Central-GRPO	**0.742**	0.320	**0.515**	0.364	**0.175**	0.101	**0.370**
		FedGRPO	0.738	0.321	0.504	**0.371**	0.167	**0.110**	0.369
OpenR1-Math	Qwen2.5-Math-1.5B	Zero-shot	0.316	0.081	0.272	0.203	0.074	0.034	0.163
		Fedpetuning+GRPO	0.348	0.114	0.272	0.210	0.049	0.040	0.172
		Fedpetuning+SFT	0.352	0.092	0.262	0.223	0.060	0.031	0.170
		DPSDA-FL+GRPO	0.574	0.162	0.318	0.242	0.043	0.034	0.229
		DPSDA-FL+SFT	0.480	0.110	0.281	0.259	0.055	0.042	0.205
		Central-GRPO	0.724	0.257	0.392	0.338	**0.114**	**0.069**	0.316
		FedGRPO	**0.740**	**0.290**	**0.414**	**0.353**	0.099	0.066	**0.327**
	Qwen2.5-3B	Zero-shot	0.118	0.092	0.063	0.046	0.006	0.006	0.055
		Fedpetuning+GRPO	0.400	0.125	0.152	0.127	0.017	0.010	0.139
		Fedpetuning+SFT	0.362	0.055	0.138	0.092	0.013	0.007	0.111
		DPSDA-FL+GRPO	0.392	0.154	0.127	0.117	**0.043**	**0.016**	0.142
		DPSDA-FL+SFT	0.416	0.121	0.171	0.169	0.016	0.013	0.151
		Central-GRPO	0.532	**0.248**	**0.305**	0.235	0.018	0.011	0.225
		FedGRPO	**0.546**	0.235	0.296	**0.240**	0.028	0.015	**0.227**
	Qwen2.5-Math-7B	Zero-shot	0.426	0.121	0.326	0.163	0.111	0.048	0.199
		Fedpetuning+GRPO	0.504	0.129	0.317	0.181	0.121	0.039	0.215
		Fedpetuning+SFT	0.497	0.143	0.297	0.189	0.072	0.008	0.201
		DPSDA-FL+GRPO	0.551	0.172	0.308	0.262	0.017	0.014	0.221
		DPSDA-FL+SFT	0.574	0.239	0.351	0.256	0.087	0.048	0.259
		Central-GRPO	0.755	0.325	0.529	0.370	0.180	0.113	0.379
		FedGRPO	**0.768**	**0.337**	**0.533**	**0.382**	**0.184**	**0.126**	**0.388**

Table 1: Performance results of FedGRPO and other methods with two trainsets on 1.5B, 3B and 7B models.

Comparison with Centralized GRPO. We compare our method against Centralized GRPO denoted as "Central-GRPO" in Tab.1, which represents a upper bound method trained on aggregated data. The results compellingly demonstrate that FedGRPO not only approaches but in several instances surpasses this centralized baseline. For example, on the OpenR1-Math dataset with the 7B model, FedGRPO achieves an average accuracy of 0.388, exceeding Central-GRPO's score of 0.379. A similar trend is observed with the 1.5B model on the same dataset. On the Math-benchmark, FedGRPO (0.369) performs almost identically to Central-GRPO (0.370) with the 7B model. This near-centralized performance demonstrates that our competence-based expert selection and dual evaluation mechanisms synergistically produce reliable reward signals, enabling effective and privacy-preserving server model optimization.

4.3 Performance Without Answers

We assess FedGRPO's robustness and effectiveness when ground-truth answers are unavailable. In this setting, clients employ locally trained reward models to evaluate policies and assign rewards, while FedGRPO selects appropriate rewards via competence-based expert selection. Specifically, we use a locally trained Qwen2.5-Math-1.5B model as the reward model under Non-IID $\beta = 0.1$; further details are provided in the Appendix. Notably, other baselines such as Fedpetuning and DPSDA-FL are inapplicable in this scenario due to the absence of reference answers.

As the results shown in Tab.2, FedGRPO demonstrates strong performance even in the absence of ground-truth answers. For the 1.5B model, FedGRPO achieves an average accuracy of 0.310, which is only slightly lower than the centralized upper bound (Central-GRPO, 0.323), and signifi-

Model	Method	Math500	Minerva	AMC	Olympiad	AIME24	AIME25	Avg.
Qwen2.5-Math-1.5B	Zero-shot	0.316	0.081	0.272	0.203	0.074	0.034	0.163
	Central-GRPO	0.701	0.307	0.440	0.353	0.081	0.056	0.323
	FedGRPO	0.681	0.272	0.414	0.317	0.114	0.061	0.310
Qwen2.5-3B	Zero-shot	0.118	0.092	0.063	0.046	0.006	0.006	0.055
	Central-GRPO	0.474	0.210	0.157	0.153	0.010	0.003	0.168
	FedGRPO	0.360	0.272	0.142	0.098	0.005	0.003	0.146
Qwen2.5-Math-7B	Zero-shot	0.426	0.121	0.326	0.163	0.111	0.048	0.199
	Central-GRPO	0.742	0.320	0.515	0.364	0.175	0.101	0.370
	FedGRPO	0.692	0.282	0.417	0.328	0.182	0.061	0.327

Table 2: Performance results of FedGRPO without ground-truth answers on Math-benchmark dataset.

cantly outperforms the zero-shot baseline (0.163). Similar trends are observed for larger models: with Qwen2.5-Math-7B, FedGRPO attains an average accuracy of 0.327, closely matching Central-GRPO's 0.369, and for Qwen2.5-3B, FedGRPO achieves 0.146 compared to Central-GRPO's 0.168.

4.4 Communication Efficiency

To evaluate FedGRPO's communication efficiency, we tested the communication overhead of FedGRPO (transmitting group rewards), Fedpetuning (transmitting LoRA trainable parameters), and DPSDA-FL (transmitting synthetic data) during a complete training process on 1.5B, 3B, and 7B models. The results are shown in Fig.2.

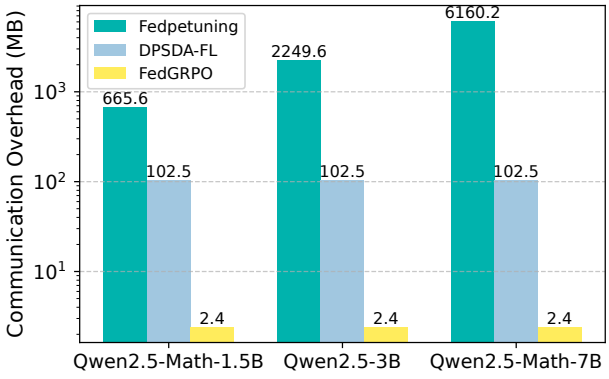
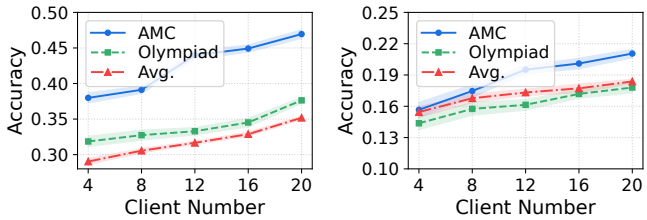


Figure 2: The communication overheaf of FedGRPO, Fedpetuning and DPSDA-FL.

From the results we can observe that FedGRPO requires only **2.4 MB** of reward signal transfer during FL, and remains constant regardless of model size because it only transmits a short reward value (1.0 or 0.0) for each policy. This is in stark contrast to the baselines: DPSDA-FL requires 102.5 MB (a 40× increase), while FedPETuning's overhead is even more prohibitive, scaling with the model size to a massive 6.1 GB for the 7B model. This represents a reduction in communication cost of two to three orders of magnitude, making FedGRPO a highly practical and scalable solution for real-world deployment.

4.5 Ablation Study

Client Number. We evaluate the performance of FedGRPO with client number K ranging from 4 to 20. As shown in Fig. 3, FedGRPO demonstrates consistent performance improvements as more clients participate. For the Qwen2.5-Math-1.5B model (Fig. 3a), the average accuracy increases steadily from approximately 0.29 with 4 clients to 0.36 with 20 clients. Notably, the accuracy on the AMC subset rises from about 0.37 to 0.47, and Olympiad accuracy improves from 0.32 to 0.38 as the client number grows. A similar trend is observed for the Qwen2.5-3B model (Fig. 3b). These consistent gains across different benchmarks confirm FedGRPO's ability to robustly aggregate distributed knowledge and benefit from larger federated networks.



(a) Qwen2.5-Math-1.5B (b) Qwen2.5-3B

Figure 3: Accuracy of FedGRPO on AMC, Olympiad, and all 6 testsets (averaged) across varying client numbers.

Expert Selection. We evaluated the impact of different selected experts number M on the performance of FedGRPO when performing competence-based expert selection. Due to space limit, the detail results are shown in Appendix.

5 Conclusion

In this work, we proposed FedGRPO, a privacy-preserving and communication-efficient framework for Federated Foundation Models (FedFMs), which leverages client-side domain expertise through reinforcement learning. By reformulating large model adaptation as a reward-based evaluation process, FedGRPO introduces two key innovations: competence-based expert selection and a federated Group-Relative Policy Optimization mechanism that aggregates scalar feedback instead of high-dimensional parameters or numerous synthetic data. These designs collectively ensure accurate evaluation, preserve user privacy, and support scalable deployment across heterogeneous clients. Empirical results show that FedGRPO not only closely matches centralized GRPO performance and surpasses other FedFMs methods, but also mitigates privacy risks in other FedFMs methods with significantly reduced communication overhead.

Acknowledgements

This work is supported by research fund of Tsinghua University - Tencent Joint Laboratory for Internet Innovation Technology.

References

Abacha, F. Z.; Teo, S. G.; Cordeiro, L. C.; and Mustafa, M. A. 2024. Synthetic data aided federated learning using foundation models. In *International Workshop on Trustworthy Federated Learning*, 106–118. Springer.

Chen, G.; Qin, Z.; Yang, M.; Zhou, Y.; Fan, T.; Du, T.; and Xu, Z. 2024. Unveiling the vulnerability of private fine-tuning in split-based frameworks for large language models: A bidirectionally enhanced attack. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2904–2918.

Chu, T.; Zhai, Y.; Yang, J.; Tong, S.; Xie, S.; Schuurmans, D.; Le, Q. V.; Levine, S.; and Ma, Y. 2025. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*.

Fan, T.; Gu, H.; Cao, X.; Chan, C. S.; Chen, Q.; Chen, Y.; Feng, Y.; Gu, Y.; Geng, J.; Luo, B.; et al. 2025. Ten challenging problems in federated foundation models. *IEEE Transactions on Knowledge and Data Engineering*.

Fan, T.; Kang, Y.; Ma, G.; Chen, W.; Wei, W.; Fan, L.; and Yang, Q. 2023. Fate-llm: A industrial grade federated learning framework for large language models. *arXiv preprint arXiv:2310.10049*.

He, C.; Luo, R.; Bai, Y.; Hu, S.; Thai, Z. L.; Shen, J.; Hu, J.; Han, X.; Huang, Y.; Zhang, Y.; et al. 2024. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*.

Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Ho, X.; Nguyen, A.-K. D.; Sugawara, S.; and Aizawa, A. 2020. Constructing A Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, 6609–6625.

Hou, C.; Wang, M.-Y.; Zhu, Y.; Lazar, D.; and Fanti, G. 2025. Private federated learning using preference-optimized synthetic data. *arXiv preprint arXiv:2504.16438*.

Hugging Face. 2025. Open R1: A fully open reproduction of DeepSeek-R1.

Kang, Y.; Fan, T.; Gu, H.; Zhang, X.; Fan, L.; and Yang, Q. 2023. Grounding foundation models through federated transfer learning: A general framework. *ACM Transactions on Intelligent Systems and Technology*.

Lewkowycz, A.; Andreassen, A.; Dohan, D.; Dyer, E.; Michalewski, H.; Ramasesh, V.; Slone, A.; Anil, C.; Schlag, I.; Gutman-Solo, T.; et al. 2022. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35: 3843–3857.

Li, H.; Zhao, X.; Guo, D.; Gu, H.; Zeng, Z.; Han, Y.; Song, Y.; Fan, L.; and Yang, Q. 2024a. Federated domain-specific knowledge transfer on large language models using synthetic data. *arXiv preprint arXiv:2405.14212*.

Li, J.; Beeching, E.; Tunstall, L.; Lipkin, B.; Soletskyi, R.; Huang, S.; Rasul, K.; Yu, L.; Jiang, A. Q.; Shen, Z.; et al. 2024b. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13(9): 9.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36: 53728–53741.

Ren, C.; Yu, H.; Peng, H.; Tang, X.; Zhao, B.; Yi, L.; Tan, A. Z.; Gao, Y.; Li, A.; Li, X.; et al. 2025. Advances and open challenges in federated foundation models. *IEEE Communications Surveys & Tutorials*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y.; Wu, Y.; et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Sun, Y.; Li, Z.; Li, Y.; and Ding, B. 2024. Improving lora in privacy-preserving federated learning. *arXiv preprint arXiv:2403.12313*.

Trivedi, H.; Balasubramanian, N.; Khot, T.; and Sabharwal, A. 2022. MuSiQue: Multihop Questions via Single-hop Question Composition. *Transactions of the Association for Computational Linguistics*, 10: 539–554.

Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Li, C.; Liu, D.; Huang, F.; Wei, H.; Lin, H.; Yang, J.; Tu, J.; Zhang, J.; Yang, J.; Yang, J.; Zhou, J.; Lin, J.; Dang, K.; Lu, K.; Bao, K.; Yang, K.; Yu, L.; Li, M.; Xue, M.; Zhang, P.; Zhu, Q.; Men, R.; Lin, R.; Li, T.; Xia, T.; Ren, X.; Ren, X.; Fan, Y.; Su, Y.; Zhang, Y.; Wan, Y.; Liu, Y.; Cui, Z.; Zhang, Z.; and Qiu, Z. 2024a. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115*.

Yang, A.; Zhang, B.; Hui, B.; Gao, B.; Yu, B.; Li, C.; Liu, D.; Tu, J.; Zhou, J.; Lin, J.; et al. 2024b. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W.; Salakhutdinov, R.; and Manning, C. D. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2369–2380.

Yi, L.; Yu, H.; Wang, G.; Liu, X.; and Li, X. 2023. pFedLoRA: Model-heterogeneous personalized federated learning with LoRA tuning. *arXiv preprint arXiv:2310.13283*.

Yu, Q.; Liu, Y.; Wang, Y.; Xu, K.; and Liu, J. 2023. Multimodal federated learning via contrastive representation ensemble. *arXiv preprint arXiv:2302.08888*.

Zhang, Z.; Yang, Y.; Dai, Y.; Wang, Q.; Yu, Y.; Qu, L.; and Xu, Z. 2023. Fedpetuning: When federated learning meets the parameter-efficient tuning methods of pre-trained language models. In *Annual Meeting of the Association of Computational Linguistics 2023*, 9963–9977. Association for Computational Linguistics (ACL).

Zhao, X.; Gu, H.; Fan, L.; Han, Y.; and Yang, Q. 2024. Disentangling data distribution for Federated Learning. *arXiv preprint arXiv:2410.12530*.

Zhu, L.; Liu, Z.; and Han, S. 2019. Deep leakage from gradients. *Advances in neural information processing systems*, 32.