PubSwap: Public-Data Off-Policy Coordination for Federated RLVR

Anupam Nayak* **Baris Askin*** **Muhammed Ustaomeroglu** **Carlee Joe-Wong** **Gauri Joshi**
Carnegie Mellon University

Abstract

Reasoning post-training with reinforcement learning from verifiable rewards (RLVR) is typically studied in centralized settings, yet many realistic applications involve decentralized private data distributed across organizations. Federated training is a natural solution, but scaling RLVR in this regime is challenging: full-model synchronization is expensive, and performing many local steps can cause severe client drift under heterogeneous data. We propose a federated RLVR framework that combines LoRA-based local adaptation with public-data-based off-policy steps to improve both communication efficiency and cross-client coordination. In particular, a small shared public dataset is used to periodically exchange and reuse response-level training signals across organizations, providing a lightweight anchor toward a more globally aligned objective without exposing private data. Our method selectively replaces locally incorrect responses with globally correct ones during public-data steps, thereby keeping training closer to the local policy while still benefiting from cross-client coordination. Across mathematical and medical reasoning benchmarks and models, our method consistently improves over standard baselines. Our results highlight a simple and effective recipe for federated reasoning post-training: combining low-rank communication with limited public-data coordination.

1 Introduction

Reinforcement learning with verifiable rewards (RLVR) has become the central component of post-training for obtaining large language models (LLMs) with strong reasoning ability. Unlike purely supervised approaches such as supervised fine-tuning (SFT) or distillation, RLVR methods like variants of Group relative policy optimization (GRPO) [Shao et al., 2024b] directly optimize models based on outcome-level reward signals derived from verifiable tasks, which has led to substantial gains in reasoning performance. Its success has driven adoption across a wide range of domains with checkable rewards, including mathematics [Shao et al., 2024b, Yu et al., 2025], coding [Zhu et al., 2024], scientific reasoning [Gunjal et al., 2025, Narayanan et al., 2025], and medical decision-making [Gunjal et al., 2025, Pan et al., 2025], finance [Zhu et al., 2025, Liu et al., 2025b], model alignment [Dineen et al., 2025], and lately to even incentivize reasoning via next token prediction [Dong et al., 2025].

In many important settings, the data required for reasoning post-training is mostly decentralized, with only a limited amount of public data available. This is especially true in personalized [Li et al., 2026] and domain-specific applications such as medicine, finance, cybersecurity, etc where relevant data is distributed across multiple organizations and cannot be shared freely because of privacy, confidentiality, and leakage concerns. For instance, medical reasoning input prompts may corres to patient histories, diagnosis reports, or clinical notes, while financial reasoning may rely on proprietary transaction data, internal risk reports, or client-specific portfolio information. This makes

*Equal contribution; corresponding email {anupamn,baskin}@andrew.cmu.edu

Accepted at ICML 2026 Workshop on RLxF: Reinforcement Learning from World Feedback.

data sharing and centralized training impractical, while relying solely on isolated private datasets can limit data diversity and reduce generalization beyond each local domain. Federated learning [McMahan et al., 2017] offers a natural framework for this setting. Instead of requiring organizations to pool their private data, federated learning trains a shared global model through distributed local steps, allowing data to remain on-device or within each institution. This paradigm has therefore emerged as a promising approach for building models from decentralized data while mitigating privacy and data-sharing concerns.

However, the scale of modern reasoning models makes federated training challenging. Frequent synchronization under full fine-tuning (FFT) incurs substantial communication overhead, as large model updates must be repeatedly transmitted and aggregated across clients. In some settings, gradients or optimizer states may also be maintained at higher precision, further increasing the systems cost. Standard approaches for reducing communication in federated learning include quantization Reisizadeh et al. [2020], Shlezinger et al. [2020], sparsification [Sattler et al., 2019, Li and Li, 2023], and sketching [Rothchild et al., 2020]. While effective in many regimes, these methods generally induce a communication–accuracy tradeoff, and compression can degrade downstream quality especially for reasoning-centric LLMs [Liu et al., 2025a]. This motivates the use of Parameter Efficient Finetuning (PEFT) based methods for Federated RLVR.

A complementary way to amortize communication is to perform more local steps between synchronization rounds. However, under heterogeneous client data, increasing local computation can exacerbate client drift and slow convergence [Wang et al., 2020]. Moreover, most of these methods do not exploit the limited public data often available in decentralized settings, especially for language tasks, where small amounts of public text data are typically accessible. Even when modest in size, such public data can provide a shared reference for coordination across clients under heterogeneous local distributions, an idea related in spirit to federated distillation [Jeong et al., 2018, Lin et al., 2020].

To address these challenges, we propose a two-fold solution in the form of a federated post-training framework. In our approach, each client performs local GRPO steps using Low-Rank Adaptation (LoRA) [Hu et al., 2022] rather than full fine-tuning, reducing both communication cost and client-side memory usage. In addition, we leverage shared public data to perform off-policy updates that provide lightweight synchronization across clients during local GRPO training. In particular, beyond standard federated parameter exchange, our method communicates evaluations of *public* data points produced by models hosted on local clients, which do not introduce additional privacy concerns. Together, these two components enable communication-efficient federated RLVR while preserving the ability to benefit from both private local data and limited shared public data. The remainder of the paper is organized as follows.

- Section 2 reviews the prior work relevant to our setting, and Section 3 introduces the problem formulation and our proposed methods.
- Section 4 presents empirical results demonstrating the effectiveness of our proposed `PubSwap` method with `Balanced` response aggregation on both mathematical and medical reasoning tasks on Qwen and LLama family of models, along with ablations on key hyperparameters. Section 5 concludes the paper.
- Additional experimental details, as well as theoretical derivations upper bounding the per-step drift induced by both our method and the FedAvg baseline, are deferred to the appendix.

2 Related Works

This work was inspired by the following previous studies conducted in related areas.

Decentralized training of Reasoning Models: Recent work has begun to examine distributed training paradigms for large models. In the large scale private-data setting, prior studies have considered collaborative training without direct data sharing, for example through localized mixture-of-experts (MoE) training followed by model merging [Shi et al., 2025]. MoE parameter efficient finetuning (PEFT) have also been explored for on-device training [Fan et al., 2025, Wagner et al., 2024, Singhal et al., 2026, Raje et al., 2025]. Meanwhile, Zhang et al. [2024] show that FedAvg-style optimization can be effective for instruction tuning. However, these works are largely restricted to supervised training with next-token prediction losss. In the RLVR setting, recent works study

decentralized training [Team et al., 2025, Zhang et al., 2026]. While they face related challenges, including communication overhead and drift arising from stale or off-policy updates, they focus on distributed training over shared public data rather than private data distributed across clients. Another closely connected application domain is personalized reasoning [Li et al., 2026, 2025], In a federated formulation, Wang et al. [2026] study GRPO based LLM training for personalized reasoning under heterogeneous rewards. Recent work has also extended federated learning to reinforcement-learning-based training of LLM agents, including GRPO-style methods [Chen et al., 2026].

Learning from public and private data: The use of representative public data has long been studied in federated learning. A common application is federated distillation, where public inputs are passed through client models trained on private data, and the resulting logits are used as teacher signals to distill a server-side model [Jeong et al., 2018, Lin et al., 2020, Cho et al., 2022, Shao et al., 2024a]. In LLM settings, public data has been used in several ways. The most common approach first tunes on public data and then fine-tunes or post-trains on private data for personalization or downstream adaptation [Yu et al., 2024, Hanke et al., 2024]. More recent work studies combining public and private data across multiple training stages, including during pretraining [Bu et al., 2024] and through public mid-training followed by federated learning on private data [Wang et al., 2024]. Another line of work uses public data to train generative models that synthesize samples matching private-data distributions [Yu et al., 2024, Hou et al., 2025]. More broadly, the value of public data has also been studied extensively in differential privacy [Alon et al., 2019]; most closely related to our setting, Jiang et al. [2025] analyze a public-private interleaving scheme for gradient-based optimization similar to ours.

Off policy Learning and GRPO: GRPO has been a key driver of recent progress in reasoning language models, motivating substantial effort toward improving its efficiency in terms of memory, wall-clock time, and compute. A recurring theme in this literature is off-policy training, where rollouts generated by a different behavioral policy, stale or otherwise non-current policies are reused to accelerate learning. Asynchronous GRPO based distributed training [Team et al., 2025, Zhang et al., 2026] is one important special case of this broader setting, since asynchrony naturally leads to stale-policy updates. Because such reuse introduces policy mismatch, it can degrade optimization through bias, instability, and drift. Previous work mitigates these effects through techniques such as careful importance sampling, reward shaping, asymmetric baselines, and clipped surrogate objectives [Zheng et al., 2026, Wan et al., 2026, Mroueh et al., 2025, Yan et al., 2025, Arnal et al., 2025]. These techniques are complementary to our approach and could be incorporated on top of our method to further improve performance.

Efficient Federated Training: Federated learning (FL) is the dominant paradigm for distributed training without direct data sharing. A large body of work has focused on improving FL by accelerating convergence [Jhunjhunwala et al., Pathak and Wainwright, 2020], reducing client drift under data heterogeneity [Li et al., 2020, Wang et al., 2020, Karimireddy et al., 2020, Acar et al., 2021], and improving communication efficiency [Konečnỳ et al., 2016, Reisizadeh et al., 2020, Shlezinger et al., 2020, Zakerinia et al., 2024]. However, these methods were largely developed for smaller-scale models and do not directly address the optimization, memory, and systems challenges that arise in LLM training. In the LLM setting, a common approach to communication- and memory-efficient federated adaptation is to combine FL with parameter-efficient fine-tuning methods such as LoRA [Wagner et al., 2024, Singhal et al., 2026, Raje et al., 2025, Cho et al., 2022, Zhang et al., 2024]. Although effective, these methods primarily focus on supervised fine-tuning objectives and are less suited beyond these settings to domains like RL post training.

3 Proposed Method

3.1 Preliminaries

GRPO: Group Relative Policy Optimization (GRPO) [Shao et al., 2024b] is a policy gradient method widely used for RL post training of LLMs. It replaces the critic model used to compute the advantage function in the Proximal Policy Optimization (PPO) [Schulman et al., 2017] with a relative advantage of the group. Let $(y_1 \cdots y_K) \sim \pi(\cdot|x)$ be IID responses produced by the model π to a prompt x then

the advantage is computed as

$$A_k(x, y_k) := \frac{r(x, y_k) - \bar{r}(x)}{\sigma(x)} \qquad \forall k \in \{1, 2 \cdots K\}, \tag{1}$$

where $r(x, y)$ is the reward associated with the response y for the prompt x, $\bar{r}(x)$ and $\sigma(x)$ are the group-mean and standard deviation across the responses $\{y_1 \cdots y_K\}$ to the same prompt x. $\{y_1 \cdots y_K\} \sim \pi_{\theta_{\text{old}}}(\cdot|x)$ are generated using a policy parameterized by $\theta_{\text{old}} \in \Theta$. The GRPO objective is then given as

$$J(\theta) = \mathop{\mathbb{E}}_{x \sim \mathcal{D}} \left[\frac{1}{K} \sum_{k=1}^{K} \frac{1}{|y_k|} \sum_{t=1}^{|y_k|} \min\left(\frac{\pi_\theta(y_{k,<t}|x)}{\pi_{\theta_{\text{old}}}(y_{k,<t}|x)} A_k, \text{clip}\left(\frac{\pi_\theta(y_{k,<t}|x)}{\pi_{\theta_{\text{old}}}(y_{k,<t}|x)}, 1-\epsilon, 1+\epsilon \right) A_k \right) \right] \tag{2}$$

where A_k is the shorthand for $A_k(x, y_k)$ and $y_{k<t}$ denotes all tokens in y_k before position t. A GRPO step has two substeps: (1) generate responses for the minibatch of prompts using $\pi_{\theta_{\text{old}}}$, which defines the objective, and (2) update the policy via gradient ascent, possibly with multiple ascent iterations (GRPO update). The updated policy then replaces $\pi_{\theta_{\text{old}}}$ for the next GRPO step.

3.2 Setup

We consider the standard federated learning framework consisting of N clients and a central server, widely used in previous literature [Jeong et al., 2018, McMahan et al., 2017]. The local dataset at each client $n \in [N]$ consists of query-verifiable answer pairs and is denoted by $D_n := \{(x_i^{(n)}, y_i^{(n)})\}_{i=1}^{|D_n|}$. In addition, there exists a shared public dataset D_{pub} that serves as a representative reference set, is accessible to all clients and the server. The public dataset is typically much smaller than the total amount of private training data residing across clients. For example, in a cross-hospital medical imaging setting, hospitals may retain private MRI datasets locally, while a public MRI dataset serves as a common reference set for everyone. The goal is to train a shared model that performs well across the query distributions of all clients, as measured by its performance on a held-out test dataset maintained by the server. This test data set is typically assumed to be representative of global training data, which is distributed heterogeneously across clients enabling the evaluation of the extent to which client-specific knowledge contributes to the learned model.

3.3 LoRA for communication-efficient federated aggregation

All clients are initialized from a shared pretrained model at the beginning of training. To reduce both the memory footprint of local fine-tuning and the communication overhead incurred during synchronization, we parameterize client updates using Low-Rank Adaptation (LoRA) [Hu et al., 2022]. Recent work suggests that LoRA is especially effective for RLVR [Wang et al., 2025], possibly because the RL training use the reward signal contains much less information than token-level supervised fine-tuning labels and can therefore often be captured and encoded easily using low-rank updates [Schulman and Lab, 2025].

Specifically, for a layer weight matrix in the pretrained backbone, we parameterize the global model at round p as

$$\mathbf{W}_p = \mathbf{W}_0 + \mathbf{B}_p \mathbf{A}_p, \tag{3}$$

where $\mathbf{W}_0 \in \mathbb{R}^{m \times d}$ denotes the frozen pretrained weight matrix, $\mathbf{B}_p \in \mathbb{R}^{m \times r}$ and $\mathbf{A}_p \in \mathbb{R}^{r \times d}$ are the global LoRA factors at round p, and $r \ll \min\{m, d\}$ is the adaptation rank. At the beginning of each round, the server broadcasts $(\mathbf{A}_p, \mathbf{B}_p)$ to all clients. Client n initializes its local factors from the global ones and, after local training, obtains updated factors $\left(\mathbf{A}_{p+1}^{(n)}, \mathbf{B}_{p+1}^{(n)}\right)$. During local training, the backbone $\mathbf{W}_0$ remains fixed and only the LoRA factors are optimized. We initialize $\mathbf{B}_0$ to the zero matrix and $\mathbf{A}_0$ randomly before training starts.

To preserve the communication benefits of low-rank updates during federated synchronization, we adopt the inexact aggregation strategy of FedIT [Zhang et al., 2024]. Rather than transmitting full dense parameter updates, each client communicates only its LoRA matrices, which are averaged separately across clients at the server to obtain the next round's parameters:

$$\mathbf{B}_{p+1} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{B}_{p+1}^{(n)}, \qquad \mathbf{A}_{p+1} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{A}_{p+1}^{(n)}. \tag{4}$$

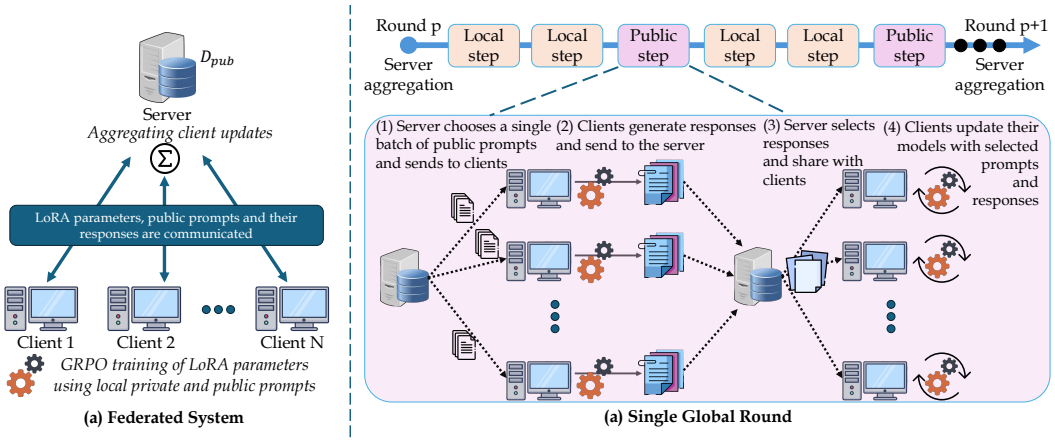

Figure 1: The figure illustrates our proposed `PubSwap` method, which alternates training on both public and private data. During a local step, each client performs a GRPO step on minibatches sampled from its own private dataset. During the public step, every client generates K responses for the same shared batch of prompts and sends these generations to the server. Based on the response aggregation method, the server then returns the responses to each client, which are used them to perform GRPO update. Models are aggregated after each communication round.

This reduces the per-layer communication cost from $2md$ to $2r(m+d)$, which is significantly smaller since $r \ll \min\{m, d\}$.

3.4 Local Optimization Algorithms

Here we will list two local optimization algorithms we use in our experiments

Local GRPO: During each communication round, every client performs τ local GRPO steps. At every such step, the client samples a size b minibatch of prompts from its local dataset, generates K responses per sample and applies a fixed number of gradient ascent steps to optimize the GRPO objective (Eq. 2). Once τ local steps are completed, the client sends its learned LoRA parameters to the server for aggregation, after which the global parameters aggregated across clients are communicated back to all clients.

`PubSwap`**:** The primary goal of this procedure is to leverage public data to mitigate client drift while still learning from the broader set of private local data. To this end, we interleave steps on private and public data during local training. In addition to the local-step parameter τ, the algorithm takes as input a `PubSwap` period $\tau_{\texttt{swap}} < \tau$. Within each `PubSwap` period, the first $\tau_{\texttt{swap}} - 1$ steps are standard GRPO steps using private data, while the final step is a public step performed locally on shared public data.

The public-data-based steps here are intended to mitigate client drift. We consider the following strategies for incorporating the public dataset:

- `Random`**:** The server first selects b prompts from the public dataset and communicates this set to all clients. Each client then generates K answers for each selected prompt and sends them back to the server. From the resulting pool of NK answers per prompt, the server uniformly samples K answers at random for each prompt and broadcasts them to all clients. Each client then performs a GRPO update using this shared global batch of K answers.
- `Balanced`**:** Similarly to `Random`, the server begins by sampling a batch of b prompts from the public dataset and broadcasting them to all clients. For each prompt, every client generates K i.i.d. responses and transmits them to the server. Let c_n denote the number of correct responses obtained by client n for a particular prompt. The response pool used for the local GRPO update at client n is constructed according to the following rule:
  - If $c_n \geq K/2$, the client forms its GRPO objective using only its own generated responses.
  - If $c_n < K/2$, the client replaces up to $K/2 - c_n$ responses with correct responses drawn from the globally collected pool, excluding its own generations.

Model	Method	MATH				DeepMath			
		τ=10	τ=40	τ=90	τ=120	τ=10	τ=40	τ=90	τ=120
Qwen3-1.7B	Base model	55.2	55.2	55.2	55.2	14.9	14.9	14.9	14.9
	FedAvg-GRPO	**77.9**	75.6	75.5	75.9	48.9	52.7	50.4	50.7
	FedProx-GRPO	77.5	76.0	76.5	75.6	50.6	52.3	48.0	47.7
	FedAvg-`PubSwap`	77.0	**76.7**	**76.9**	**76.6**	**51.1**	**53.0**	**53.3**	**55.8**
Qwen2.5-Math-1.5B	Base	58.4	58.4	58.4	58.4	34.9	34.9	34.9	34.9
	FedAvg-GRPO	**73.5**	**72.3**	71.9	71.1	51.0	52.3	50.5	49.3
	FedProx-GRPO	72.8	71.4	71.5	70.6	52.2	52.1	48.0	49.4
	FedAvg-`PubSwap`	72.4	71.6	**72.5**	**71.9**	**52.4**	**53.1**	**50.8**	**53.1**

Table 1: Pass @1 performance comparison on MATH [Hendrycks et al., 2021] and DeepMath [He et al., 2025] across different numbers of local steps (τ). The `Balanced` method and a `swap` period of 2 is used here for response aggregation with `PubSwap`. We tune baselines to the best performance, more details are presented in the Appendix.

This response aggregation strategy increases reward variance within the response set assigned to each client while replacing only locally incorrect responses with globally correct ones.

Both proposed methods modify only the first part of the GRPO step, namely response generation and objective definition, by swapping responses and defining the objective accordingly. The communication overhead associated with the public data steps is negligible compared to the cost of exchanging the LoRA matrices at the end of every communication round.

The proposed mechanisms are inherently off-policy, because part of the response set used for a local GRPO update may come from client models that have already drifted during the preceding private-data steps (See figure 1). In this sense, the resulting learning dynamics are intuitively closer to SFT, in which models learn from trajectories produced by other models. Moreover, under `Balanced`, a local answer is replaced by a global correct answer only when the local answer is incorrect. Consequently, the local policy learns both by downweighting locally incorrect answers and by leveraging both locally correct and globally correct answers to accelerate learning.

Compared to `Random`, the `Balanced` strategy is less off-policy in nature. This is because it replaces only locally incorrect responses, and only when the number of locally correct responses falls below $K/2$. As a result, the local GRPO objective is constructed using a response set that remains closer to the client's own policy. Moreover, when all responses to a prompt are either correct or incorrect, the GRPO advantage vanishes and the objective in Equation (2) becomes zero. For this reason, in `Balanced` we replace responses only up to the point where the resulting set contains an equal number of correct and incorrect answers. This maximizes reward variance within the subset [Xu et al., 2025], which has been observed to improve performance [Xu et al., 2025, Ye et al., 2025].

4 Experimental Results

4.1 Setup

Datasets and Models: We choose the Qwen family of models for our models, presenting results on Qwen2.5-MATH-1.5B [Yang et al., 2024], Qwen3-1.7B and Qwen3-4B-Instruct models [Yang et al., 2025]. For math reasoning, we evaluate our method against baselines on MATH [Hendrycks et al., 2021] and DeepMath [He et al., 2025]. MATH contains 12.5K competition-style high-school math problems with difficulty levels 1–5, whereas DeepMath is a much larger 103K-problem benchmark with predominantly higher-difficulty problems, making it substantially harder than MATH. We also test our proposed method on medical reasoning tasks for which we use a mixture of MedQA [Jin et al., 2021] and MedMCQA datasets [Pal et al., 2022] and the Llama3.2-3B-Instruct [Grattafiori et al., 2024] model.

Implementation details: Our implementation is based on the VERL framework [Sheng et al., 2025], and we adopt most of the recommended hyperparameter settings from VERL. In addition, we apply LoRA to all layers across models, using LoRA rank = 32 and LoRA $\alpha = 64$. Consequently, we use a learning rate of 1×10^{-5} rather than the standard 1×10^{-6} recommended in VERL for full

fine-tuning, following scaling law evidence suggesting that LoRA typically benefits from a learning rate roughly $10\times$ larger than that used for full fine-tuning [Schulman and Lab, 2025].We additionally adopt the clip higher strategy for importance weights proposed in DAPO [Yu et al., 2025], resulting in an asymmetric trust region of $(1-\epsilon_{\text{low}}, 1+\epsilon_{\text{high}})$, where $\epsilon_{\text{low}} = 0.2$ and $\epsilon_{\text{high}} = 0.25$. These settings are used consistently across all methods and baselines. Additional details are provided in the appendix.

Baselines: We report both the final checkpoint test accuracy and, in the appendix, the best checkpoint accuracy for all runs. In all math reasoning experiments reported in the main text, we train for 360 GRPO steps, with two gradient update steps per GRPO step, and evaluate multiple choices of the number of local steps (τ). As baselines, we compare our method against FedAvg + GRPO and FedProx [Li et al., 2020], a standard method for mitigating client drift in supervised federated learning. FedProx modifies the local minimization objective on client k to

Method	τ=10	τ=40	τ=90	τ=120
Base model	51.1	51.1	51.1	51.1
FedAvg-GRPO	66.9	**66.5**	67.0	67.4
FedAvg-`PubSwap`	**70.0**	65.4	**70.2**	**70.5**

Table 2: Pass@1 performance of the model Qwen3-4B-Instruct-2507 on DeepMath across different numbers of local steps (τ). The `Balanced` method and a `swap` period of 2 is used here for response aggregation with `PubSwap`.

$$\min_{\theta} \; F_k(\theta) + \frac{\mu}{2}\left\|\theta - \theta^{(t)}\right\|^2, \tag{5}$$

where $\theta^{(t)}$ denotes the global model parameters at the start of the communication round, before local training begins. For each reported model, we tuned $\mu \in \{0.001, 0.01, 0.1\}$ separately and report the best-performing choice. In our implementation, the FedProx penalty is applied only to the LoRA matrices separately. For all runs, we use a rollout temperature of 0.7, a maximum generation length of 2048 tokens, and no top-p or top-K sampling. Additional details are provided in the appendix.

4.2 Main Results

Mathematical reasoning tasks: Our main results for Qwen2.5-Math-1.5B and Qwen3-1.7B are reported in Table 1. Across both model sizes, our proposed `PubSwap` method with `Balanced` response aggregation outperforms both FedAvg+GRPO and FedProx for most choices of the local step parameter. Unless otherwise noted, all reported results for the `Balanced` variant use a `PubSwap` period of 2. Notably, the gains are larger at higher values of the local step (τ) parameter.

This regime is particularly important in practice, since larger τ reduces synchronization frequency and hence communication overhead. The stronger performance of our method in this setting suggests that it is especially effective in the communication-efficient regime where standard federated RL training becomes more challenging.

Our experiments further show that, unlike in many supervised federated learning settings, FedProx does not improve performance in our RLVR setup and in many cases underperforms even vanilla FedAvg+GRPO. One possible explanation is that constraining local updates in parameter space does not necessarily induce a corresponding constraint in policy/token space: for autoregressive models, small differences in token probabilities can compound through sampling and lead to substantially different output trajectories.

Method	τ=10	τ=40	τ=90	τ=120
Base model	49.2	49.2	49.2	49.2
FedAvg-GRPO	**58.7**	58.2	57.9	56.0
FedAvg-`PubSwap`	57.5	**59.4**	**58.5**	**58.1**

Table 3: Pass@1 performance of the model Llama3.2-3B-Instruct on medical reasoning across different numbers of local steps (τ). The `Balanced` method and a `swap` period of 2 is used here for response aggregation with `PubSwap`.

We additionally report results for Qwen3-4B-Instruct model on the DeepMath dataset in Table 2. Here, `PubSwap` outperforms FedAvg+GRPO at three out of four local-step settings and is broadly consistent with our main results.

Medical reasoning tasks: Our main results on the medical reasoning tasks are presented in Table 3. All the reported numbers correspond to 540 GRPO steps except for the τ=120 setting where the reported accuracy corresponds to 600 GRPO steps. Similar to the mathematical reasoning experiments,

our proposed PubSwap method outperforms the FedAvg+GRPO baseline in the settings where it is evaluated.

A common observation across both setups is that increasing the local-step parameter does not always degrade performance. One possible explanation is that when synchronization is too frequent, aggregation is performed after only a few steps, which can interrupt the optimization dynamics by resetting the local optimizer state at the end of each round. While this effect is less pronounced in smaller-scale settings, it may become more significant for larger models. Prior work has proposed alleviating this issue through server-side momentum [Reddi et al., 2020, Hsu et al., 2019], and such techniques could in principle be combined with our method to improve performance in the small-local-step regime. However, we expect them to be less important in the more practical setting where the number of local steps is sufficiently large. In our implementation, optimizer states are reinitialized after each aggregation round, so this issue affects both the baselines and the proposed method.

Additionally, we find that under the PubSwap method, Random can outperform Balanced response aggregation scheme when the number of local steps is small (10 or 40). This advantage does not persist at larger local-step values (90 or 120), where training can become unstable, especially for Qwen3-1.7B. One possible explanation is that as client models drift farther apart, responses generated by other clients become increasingly off-policy for the local model. The Balanced strategy alleviates this by retaining locally generated incorrect responses and by leaving the local response set unchanged on easier prompts where at least $K/2$ responses are already correct.

Method	τ=40	τ=90	τ=120
Base model	14.9	14.9	14.9
FedAvg+GRPO	52.0	51.6	51.3
PubSwap	**53.0**	**56.7**	**54.9**

Table 4: Pass@1 performance of the model Qwen3-1.7B on DeepMath across different numbers of local steps (τ) for a higher heterogeneity split.

4.3 Experiments on higher heterogeneity

In Table 4, we compare our method against the baseline under a higher-heterogeneity client split. The corresponding bubble plot of the client-wise topic distributions is provided in the Appendix section B. While the local datasets in Table 1 were constructed by sampling client distributions from a Dirichlet distribution with $\alpha = 0.3$, the split used in Table 4 corresponds to a more heterogeneous setting with $\alpha = 0.1$. We generate equal-sized client datasets following the procedure of Acar et al. [2021]. The Balanced method and a swap period of 2 is used here for response aggregation with PubSwap. The results show that, after 360 rounds of training, our proposed method outperforms FedAvg across different local-step configurations, consistent with the trends observed in the lower-heterogeneity setting.

Method	τ=80	τ=90	τ=120
FedAvg-GRPO	48.0	50.4	50.7
FedProx-GRPO	46.5	48.0	47.7
$\tau_{\texttt{swap}}$ = 2	52.2	53.3	**55.8**
$\tau_{\texttt{swap}}$ = 4	**54.3**	**54.3**	52.5
$\tau_{\texttt{swap}}$ = 8	53.1	53.4	52.6

Table 5: Pass@1 performance of Qwen3-1.7B across swap periods on DeepMath.

4.4 Ablation on swap period

In Tables 5 and 6, we study the effect of varying the PubSwap period for the Qwen3-1.7B and Qwen2.5-Math-1.5B models. All experiments are run for 360 GRPO steps except for the τ=80 column in Qwen3-1.7B, where the reported numbers correspond to the final accuracy after 320 steps. Increasing the PubSwap period reduces the frequency with which public data is incorporated into training, while decreasing it leads to more frequent public-data steps. Although more frequent public steps can in principle help limit drift through steps on shared prompts and response aggregation, they also increase the extent to which locally generated responses are replaced by globally correct responses, thereby introducing a

Method	τ=90	τ=120
FedAvg-GRPO	50.5	49.3
FedProx-GRPO	48.0	49.4
$\tau_{\texttt{swap}}$ = 2	51.0	53.1
$\tau_{\texttt{swap}}$ = 4	**52.3**	**53.6**
$\tau_{\texttt{swap}}$ = 8	51.9	51.7

Table 6: Pass@1 performance of Qwen2.5-Math-1.5B across different swap periods on DeepMath.

larger off-policy shift especially in later stages of local training where the models drift further. Thus, smaller PubSwap periods are not necessarily always beneficial.

While Balanced always induces higher reward variance within the rollout set used for model updates, which can often accelerate learning [Xu et al., 2025, Ye et al., 2025], this advantage may be attenuated in our setting. Unlike Xu et al. [2025], where all rollouts are on-policy, our method injects responses that can be off-policy relative to the local model, and this mismatch can reduce the benefit of increased variance. Nevertheless, across settings and models and for PubSwap periods of 2, 4, and 8, the proposed PubSwap method with Balanced aggregation consistently outperforms the baselines.

5 Conclusion

In this work, we proposed PubSwap, a federated RLVR framework that combines LoRA-based local adaptation with coordination through shared public data. By interleaving private-data GRPO steps with public-data off-policy steps, our method improves cross-client alignment while preserving communication efficiency and privacy. Across mathematical and medical reasoning benchmarks, multiple model families, and varying heterogeneity levels, Fedavg-PubSwap consistently outperformed the baselines, with the consistent gains appearing in high-local-step regimes where communication efficiency matters most. Promising directions for future work include adaptive training strategies along multiple axes, rather than only adjusting the overall balance between public and private data. These include curriculum based strategies for sampling private data and public prompts, adaptive assignment of public prompts to clients based on informativeness or coordination value, and smarter response aggregation schemes that provide each client with the external responses most useful for its current policy. Stronger off-policy correction methods may further improve performance.

Ethics Statement

Our work focuses on communication-efficient federated training, which can support collaborative model development without requiring raw data sharing across organizations. We do not identify any unique ethical concerns beyond those already associated with federated learning and reasoning model deployment, though standard considerations around privacy, data governance, and responsible downstream use still apply.

References

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

Noga Alon, Raef Bassily, and Shay Moran. Limits of private learning with access to public data. *Advances in neural information processing systems*, 32, 2019.

Charles Arnal, Gaëtan Narozniak, Vivien Cabannes, Yunhao Tang, Julia Kempe, and Remi Munos. Asymmetric REINFORCE for off-policy reinforcement learning: Balancing positive and negative rewards. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Zhiqi Bu, Xinwei Zhang, Sheng Zha, Mingyi Hong, and George Karypis. Pre-training differentially private models with limited public data. *Advances in Neural Information Processing Systems*, 37: 94652–94683, 2024.

Canyu Chen, Kangyu Zhu, Zhaorun Chen, Zhanhui Zhou, Shizhe Diao, Yiping Lu, Tian Li, Manling Li, and Dawn Song. Federated agent reinforcement learning, 2026.

Yae Jee Cho, Andre Manoel, Gauri Joshi, Robert Sim, and Dimitrios Dimitriadis. Heterogeneous ensemble knowledge transfer for training large models in federated learning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI) Main Track*, 2022.

Jacob Dineen, Aswin Rrv, Qin Liu, Zhikun Xu, Xiao Ye, Ming Shen, Zhaonan Li, Shijie Lu, Chitta Baral, Muhao Chen, and Ben Zhou. QA-LIGN: Aligning LLMs through constitutionally decomposed QA. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose,

and Violet Peng, editors, *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 20619–20642, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.1123. URL https://aclanthology.org/2025.findings-emnlp.1123/.

Qingxiu Dong, Li Dong, Yao Tang, Tianzhu Ye, Yutao Sun, Zhifang Sui, and Furu Wei. Reinforcement pre-training. *arXiv preprint arXiv:2506.08007*, 2025.

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

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. *arXiv preprint arXiv:2507.17746*, 2025.

Vincent Hanke, Tom Blanchard, Franziska Boenisch, Iyiola E Olatunji, Michael Backes, and Adam Dziedzic. Open llms are necessary for current private adaptations and outperform their closed alternatives. *Advances in Neural Information Processing Systems*, 37:1220–1250, 2024.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, et al. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *arXiv preprint arXiv:2504.11456*, 2025.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Charlie Hou, Mei-Yu Wang, Yige Zhu, Daniel Lazar, and Giulia Fanti. Private federated learning using preference-optimized synthetic data. In *Forty-second International Conference on Machine Learning*, 2025.

Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

Eunjeong Jeong, Seungeun Oh, Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *arXiv preprint arXiv:1811.11479*, 2018.

Divyansh Jhunjhunwala, Shiqiang Wang, and Gauri Joshi. Fedexp: Speeding up federated averaging via extrapolation. In *The Eleventh International Conference on Learning Representations*.

Shuli Jiang, Pranay Sharma, Zhiwei Steven Wu, and Gauri Joshi. The cost of shuffling in private gradient based optimization. *arXiv preprint arXiv:2502.03652*, 2025.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021.

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*, pages 5132–5143. PMLR, 2020.

Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

Shuyue Stella Li, Melanie Sclar, Hunter Lang, Ansong Ni, Jacqueline He, Puxin Xu, Andrew Cohen, Chan Young Park, Yulia Tsvetkov, and Asli Celikyilmaz. Prefpalette: Personalized preference modeling with latent attributes. *arXiv preprint arXiv:2507.13541*, 2025.

Shuyue Stella Li, Avinandan Bose, Faeze Brahman, Simon Shaolei Du, Pang Wei Koh, Maryam Fazel, and Yulia Tsvetkov. Personalized reasoning: Just-in-time personalization and why LLMs fail at it. In *The Fourteenth International Conference on Learning Representations*, 2026.

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.

Xiaoyun Li and Ping Li. Analysis of error feedback in federated non-convex optimization with biased compression: Fast convergence and partial participation. In *International Conference on Machine Learning*, pages 19638–19688. PMLR, 2023.

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.

Ruikang Liu, Yuxuan Sun, Manyi Zhang, Haoli Bai, Xianzhi Yu, Tiezheng Yu, Chun Yuan, and Lu Hou. Quantization hurts reasoning? an empirical study on quantized reasoning models. *arXiv preprint arXiv:2504.04823*, 2025a.

Zhaowei Liu, Xin Guo, Zhi Yang, Fangqi Lou, Lingfeng Zeng, Mengping Li, Qi Qi, Zhiqiang Liu, Yiyang Han, Dongpo Cheng, et al. Fin-r1: A large language model for financial reasoning through reinforcement learning. *arXiv preprint arXiv:2503.16252*, 2025b.

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*, pages 1273–1282. Pmlr, 2017.

Youssef Mroueh, Nicolas Dupuis, Brian Belgodere, Apoorva Nitsure, Mattia Rigotti, Kristjan Greenewald, Jiri Navratil, Jerret Ross, and Jesus Rios. Revisiting group relative policy optimization: Insights into on-policy and off-policy training. *arXiv preprint arXiv:2505.22257*, 2025.

Siddharth M Narayanan, James D Braza, Ryan-Rhys Griffiths, Albert Bou, Geemi Wellawatte, Mayk Caldas Ramos, Ludovico Mitchener, Samuel G Rodriques, and Andrew D White. Training a scientific reasoning model for chemistry. *arXiv preprint arXiv:2506.17238*, 2025.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on health, inference, and learning*, pages 248–260. PMLR, 2022.

Jiazhen Pan, Che Liu, Junde Wu, Fenglin Liu, Jiayuan Zhu, Hongwei Bran Li, Chen Chen, Cheng Ouyang, and Daniel Rueckert. Medvlm-r1: Incentivizing medical reasoning capability of vision-language models (vlms) via reinforcement learning. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 337–347. Springer, 2025.

Reese Pathak and Martin J Wainwright. Fedsplit: An algorithmic framework for fast federated optimization. *Advances in neural information processing systems*, 33:7057–7066, 2020.

Arian Raje, Baris Askin, Divyansh Jhunjhunwala, and Gauri Joshi. Ravan: Multi-head low-rank adaptation for federated fine-tuning. *arXiv preprint arXiv:2506.05568*, 2025.

Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, Ali Jadbabaie, and Ramtin Pedarsani. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In *International conference on artificial intelligence and statistics*, pages 2021–2031. PMLR, 2020.

Daniel Rothchild, Ashwinee Panda, Enayat Ullah, Nikita Ivkin, Ion Stoica, Vladimir Braverman, Joseph Gonzalez, and Raman Arora. Fetchsgd: Communication-efficient federated learning with sketching. In *International Conference on Machine Learning*, pages 8253–8265. PMLR, 2020.

Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Robust and communication-efficient federated learning from non-iid data. *IEEE transactions on neural networks and learning systems*, 31(9):3400–3413, 2019.

John Schulman and Thinking Machines Lab. Lora without regret. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20250929. https://thinkingmachines.ai/blog/lora/.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Jiawei Shao, Fangzhao Wu, and Jun Zhang. Selective knowledge sharing for privacy-preserving federated distillation without a good teacher. *Nature Communications*, 15(1):349, 2024a.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024b.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pages 1279–1297, 2025.

Weijia Shi, Akshita Bhagia, Kevin Farhat, Niklas Muennighoff, Pete Walsh, Jacob Morrison, Dustin Schwenk, Shayne Longpre, Jake Poznanski, Allyson Ettinger, et al. Flexolmo: Open language models for flexible data use. *arXiv preprint arXiv:2507.07024*, 2025.

Nir Shlezinger, Mingzhe Chen, Yonina C Eldar, H Vincent Poor, and Shuguang Cui. Uveqfed: Universal vector quantization for federated learning. *IEEE Transactions on Signal Processing*, 69: 500–514, 2020.

Raghav Singhal, Kaustubh Ponkshe, Rohit Vartak, Lav R. Varshney, and Praneeth Vepakomma. Fed-SB: A silver bullet for extreme communication efficiency and performance in (private) federated loRA fine-tuning. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856.

Prime Intellect Team, Sami Jaghouar, Justus Mattern, Jack Min Ong, Jannik Straube, Manveer Basra, Aaron Pazdera, Kushal Thaman, Matthew Di Ferrante, Felix Gabriel, et al. Intellect-2: A reasoning model trained through globally decentralized reinforcement learning. *arXiv preprint arXiv:2505.07291*, 2025.

Nicolas Wagner, Dongyang Fan, and Martin Jaggi. Personalized collaborative fine-tuning for on-device large language models. In *First Conference on Language Modeling*, 2024.

Xu Wan, Yansheng Wang, Wenqi Huang, and Mingyang Sun. Buffer matters: Unleashing the power of off-policy reinforcement learning in large language model reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026.

Boxin Wang, Yibo Zhang, Yuan Cao, Bo Li, Hugh McMahan, Sewoong Oh, Zheng Xu, and Manzil Zaheer. Can public large language models help private cross-device federated learning? In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 934–949, 2024.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33:7611–7623, 2020.

Shangshang Wang, Julian Asilis, Ömer Faruk Akgül, Enes Burak Bilgin, Ollie Liu, and Willie Neiswanger. Tina: Tiny reasoning models via lora. *arXiv preprint arXiv:2504.15777*, 2025.

Ziyao Wang, Daeun Jung, Yexiao He, Guoheng Sun, Zheyu Shen, Myungjin Lee, and Ang Li. Fedmoa: Federated grpo for personalized reasoning llms under heterogeneous rewards. *arXiv preprint arXiv:2602.00453*, 2026.

Yixuan Even Xu, Yash Savani, Fei Fang, and J Zico Kolter. Not all rollouts are useful: Down-sampling rollouts in llm reinforcement learning. *arXiv preprint arXiv:2504.13818*, 2025.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Chenlu Ye, Zhou Yu, Ziji Zhang, Hao Chen, Narayanan Sadagopan, Jing Huang, Tong Zhang, and Anurag Beniwal. Beyond correctness: Harmonizing process and outcome rewards through rl training. *arXiv preprint arXiv:2509.03403*, 2025.

Da Yu, Peter Kairouz, Sewoong Oh, and Zheng Xu. Privacy-preserving instructions for aligning large language models. *arXiv preprint arXiv:2402.13659*, 2024.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Hossein Zakerinia, Shayan Talaei, Giorgi Nadiradze, and Dan Alistarh. Communication-efficient federated learning with data and client heterogeneity. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, pages 3448–3456, 2024.

Han Zhang, RuibinZheng, ZEXUAN YI, Zhuo Zhang, Hanyang Peng, Hui Wang, Jiayin Qi, Binxing Fang, Ruifeng Xu, and Yue Yu. GEPO: Group expectation policy optimization for stable heterogeneous reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 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)*, pages 6915–6919. IEEE, 2024.

Haizhong Zheng, Jiawei Zhao, and Beidi Chen. Prosperity before collapse: How far can off-policy RL reach with stale data on LLMs? In *The Fourteenth International Conference on Learning Representations*, 2026.

Jie Zhu, Qian Chen, Huaixia Dou, Junhui Li, Lifan Guo, Feng Chen, and Chi Zhang. Dianjin-r1: Evaluating and enhancing financial reasoning in large language models. *arXiv preprint arXiv:2504.15716*, 2025.

Qihao Zhu, Daya Guo, Zhihong Shao, Dejian Yang, Peiyi Wang, Runxin Xu, Y Wu, Yukun Li, Huazuo Gao, Shirong Ma, et al. Deepseek-coder-v2: Breaking the barrier of closed-source models in code intelligence. *arXiv preprint arXiv:2406.11931*, 2024.

Appendix

LLM usage

We used LLMs minimally, focusing on making sentences more concise to fit the page limit.

A Supporting Theoretical Derivations

In this section, we analyze, in a simplified setting, the drift between two clients, n and n', induced by a single step of gradient descent on private data and by our `Balanced` method. This treatment is intended mainly to provide intuition for the behavior observed in the main text, rather than a fully general theorem-level statement. To keep the exposition compact, we do the analysis in a simplified setting and omit routine technical details:

- The result is derived for SGD instead of AdamW and uses only one gradient update per GRPO step.
- We model heterogeneity and estimation noise assumptions directly at the objective-function level, rather than through data distributions and sampling (as is common in most federated learning analyses). This abstraction allows us to sidestep the technical complexities of the GRPO objective, including clipped importance ratios, gradient clipping induced bounded gradients, and KL regularization.

Analogous conclusions can be derived for the exact analytical expressions in GRPO with more extensive assumptions and bookkeeping.

A.1 Understanding the Drift induced by Private Gradient Descent steps

In this section, we quantify the drift between clients n and n' by deriving an upper bound on the effect of a single gradient descent step computed on a mini-batch of their respective private datasets. At the beginning of the communication round, all clients start with the same model θ_0. Let Δ_t be the drift between the clients n and n' at local step t defined as

$$\Delta_t := \theta_n^{(t)} - \theta_{n'}^{(t)}.$$

For the private local GRPO step, the parameters are updated as

$$\theta_n^{(t+1)} = \theta_n^{(t)} + \eta \hat{g}_n^{(t)}, \qquad \theta_{n'}^{(t+1)} = \theta_{n'}^{(t)} + \eta \hat{g}_{n'}^{(t)},$$

where $\hat{g}_n^{(t)}$ is the stochastic GRPO gradient computed using the private minibatch $B_n^{(t)}$ at client n. Define the minibatch-conditioned expected private gradient

$$g_B^{\text{priv}}(\theta, \theta') := \frac{1}{b} \sum_{x \in B} \mathbb{E}_{\mathcal{Y} \sim \pi_{\theta'}(\cdot|x)^{\otimes K}} \left[\nabla_\theta \ell_{\text{GRPO}}(\theta; x, \mathcal{Y}) \right].$$

For one gradient update per GRPO step $\theta = \theta'$, one can write the gradient term as

$$g_B^{\text{priv}}(\theta) := \frac{1}{b} \sum_{x \in B} \mathbb{E}_{\mathcal{Y} \sim \pi_{\theta}(\cdot|x)^{\otimes K}} \left[\nabla_\theta \ell_{\text{GRPO}}(\theta; x, \mathcal{Y}) \right].$$

where $\ell_{\text{GRPO}}(\theta; x, \mathcal{Y})$ denotes the GRPO loss at point θ with importance ratio also computed with the sampling policy θ on a rollout batch $\mathcal{Y}$ for the prompt x. The response sampling noise in $g_n^{(t)}$ can be quantified as follows

$$\hat{g}_n^{(t)} = g_{B_n^{(t)}}^{\text{priv}}(\theta_n^{(t)}) + \xi_n^{(t)},$$

Now,

$$\Delta_{t+1} = \Delta_t + \eta \left(\hat{g}_n^{(t)} - \hat{g}_{n'}^{(t)} \right), \tag{6}$$

so

$$\|\Delta_{t+1}\| \leq \|\Delta_t\| + \eta \left\| \hat{g}_n^{(t)} - \hat{g}_{n'}^{(t)} \right\|.$$

Adding and subtracting $g_{B_n^{(t)}}^{\text{priv}}(\theta_{n'}^{(t)})$ gives

$$\hat{g}_n^{(t)} - \hat{g}_{n'}^{(t)} = \left(g_{B_n^{(t)}}^{\text{priv}}(\theta_n^{(t)}) - g_{B_n^{(t)}}^{\text{priv}}(\theta_{n'}^{(t)}) \right) + \left(g_{B_n^{(t)}}^{\text{priv}}(\theta_{n'}^{(t)}) - g_{B_{n'}^{(t)}}^{\text{priv}}(\theta_{n'}^{(t)}) \right) + \left(\xi_n^{(t)} - \xi_{n'}^{(t)} \right).$$

Assumption 1 (Lipschitz private on-policy gradient) *There exists a constant $L_{\text{priv}} > 0$ such that for any minibatch B across clients of size b and any $\theta, \theta' \in \Theta$,*

$$\left\| g_B^{\text{priv}}(\theta) - g_B^{\text{priv}}(\theta') \right\| \leq L_{\text{priv}} \|\theta - \theta'\|,$$

where

$$g_B^{\text{priv}}(\theta) := \frac{1}{b} \sum_{x \in B} \mathbb{E}_{\mathcal{Y} \sim \pi_\theta(\cdot|x)^{\otimes K}} [\nabla_\theta \ell_{\text{GRPO}}(\theta; x, \mathcal{Y})].$$

Since each local GRPO step performs a single gradient update from on-policy rollouts, this map jointly varies the sampling distribution and the optimization variable. This is stronger than requiring Lipschitzness in the optimization variable alone with a fixed sampling policy, which is sufficient for our case, as it additionally requires that the expected GRPO gradient varies smoothly as the response distribution shifts with the policy. Now using assumption 1 we have

$$\left\| g_B^{\text{priv}}(\theta) - g_B^{\text{priv}}(\theta') \right\| \leq L_{\text{priv}} \|\theta - \theta'\|. \tag{7}$$

and using equations 7 and 6

$$\left\| \theta_n^{(t+1)} - \theta_{n'}^{(t+1)} \right\| \leq (1 + \eta L_{\text{priv}}) \left\| \theta_n^{(t)} - \theta_{n'}^{(t)} \right\| + \eta H_{n,n'}^{(t)} + \eta \left(\left\| \xi_n^{(t)} \right\| + \left\| \xi_{n'}^{(t)} \right\| \right),$$

where

$$H_{n,n'}^{(t)} := \left\| g_{B_n^{(t)}}^{\text{priv}}(\theta_{n'}^{(t)}) - g_{B_{n'}^{(t)}}^{\text{priv}}(\theta_{n'}^{(t)}) \right\|.$$

Further define the population private gradient to quantify the noise from minibatch selection

$$g_n^{\text{priv}}(\theta) := \mathbb{E}_{B_n} \left[g_{B_n}^{\text{priv}}(\theta) \right] \quad \text{and} \quad H_{n,n'}^{(t)} \leq \zeta_{n,n'}^{(t)} + \delta_n^{(t)} + \delta_{n'}^{(t)},$$

where

$$\zeta_{n,n'}^{(t)} := \left\| g_n^{\text{priv}}(\theta_{n'}^{(t)}) - g_{n'}^{\text{priv}}(\theta_{n'}^{(t)}) \right\|, \qquad \delta_n^{(t)} := \left\| g_{B_n^{(t)}}^{\text{priv}}(\theta_{n'}^{(t)}) - g_n^{\text{priv}}(\theta_{n'}^{(t)}) \right\|.$$

Therefore,

$$\left\| \theta_n^{(t+1)} - \theta_{n'}^{(t+1)} \right\| \leq (1 + \eta L_{priv}) \left\| \theta_n^{(t)} - \theta_{n'}^{(t)} \right\| + \eta \zeta_{n,n'}^{(t)} + \eta(\delta_n^{(t)} + \delta_{n'}^{(t)}) + \eta \left(\left\| \xi_n^{(t)} \right\| + \left\| \xi_{n'}^{(t)} \right\| \right).$$

Note that the factor contributing to a large drift here will be L_{priv} this is taken as the maximum smoothness parameter over any mini batch *across users* (Assumption 1). Additionally, the term

$$\zeta_{n,n'}^{(t)} := \left\| g_n^{\text{priv}}(\theta_{n'}^{(t)}) - g_{n'}^{\text{priv}}(\theta_{n'}^{(t)}) \right\|,$$

is dictated by the dissimilarity of the avg gradient between the client n and n', and can be large in heterogeneous settings.

A.2 Understanding the Drift induced by `Balanced` steps using public data

Define the shared on-policy GRPO gradient at time t as the expected gradient on the dataset $B_{\text{pub}}^{(t)}$ chosen for the `Balanced` step at round t

$$g_{\text{pub}}^{(t)}(\theta) := \frac{1}{b} \sum_{x \in B_{\text{pub}}^{(t)}} \mathbb{E}_{\mathcal{Y} \sim \pi_\theta(\cdot|x)^{\otimes K}} \left[\nabla_\theta \ell_{\text{GRPO}}(\theta; x, \mathcal{Y}) \right].$$

Define the `Balanced` gradient at client i on the public minibatch as

$$g_{B_{\text{pub}}^{(t)}, i}^{\texttt{Balanced}} \left(\theta; \Theta^{(t)} \right) := \frac{1}{b} \sum_{x \in B_{\text{pub}}^{(t)}} \mathbb{E}_{\mathcal{Y} \sim Q_i^{\texttt{Balanced}}(\cdot|x;\, \theta,\, \Theta^{(t)})} \left[\nabla_\theta \ell_{\text{GRPO}}(\theta; x, \mathcal{Y}) \right],$$

where $Q_i^{\texttt{Balanced}}(\cdot \mid x;\, \theta,\, \Theta^{(t)})$ denotes the distribution over response groups at client i after applying the `Balanced` replacement rule: client i first generates K responses from $\pi_\theta(\cdot \mid x)$, then replaces up

to $\left(\frac{K}{2} - c_i^{(t)}(x)\right)_+$ incorrect responses with correct responses drawn from the globally collected pool $\bigcup_{j\neq i}\{\text{responses from } \pi_{\theta_j^{(t)}}(\cdot \mid x)\}$. The dependence on $\Theta^{(t)} := (\theta_1^{(t)}, \ldots, \theta_N^{(t)})$ enters through this donor pool. For each client i, define the `Balanced` distortion

$$e_i^{(t)}\left(\theta, \Theta^{(t)}\right) := g^{\texttt{Balanced}}_{B^{(t)}_{\text{pub}}, i}\left(\theta; \Theta^{(t)}\right) - g^{(t)}_{\text{pub}}(\theta).$$

Then the stochastic `Balanced` gradient can be written as

$$\hat{g}_i^{\texttt{Balanced},(t)} = g^{(t)}_{\text{pub}}\left(\theta_i^{(t)}\right) + e_i^{(t)}\left(\theta_i^{(t)}, \Theta^{(t)}\right) + \xi_i^{\texttt{Balanced},(t)}.$$

where $\xi_i^{\texttt{Balanced},(t)}$ is the noise term arising from randomness in response sampling. Hence

$$\begin{aligned}\hat{g}_n^{\texttt{Balanced},(t)} - \hat{g}_{n'}^{\texttt{Balanced},(t)} &= \left(g^{(t)}_{\text{pub}}\left(\theta_n^{(t)}\right) - g^{(t)}_{\text{pub}}\left(\theta_{n'}^{(t)}\right)\right) \\ &\quad + \left(e_n^{(t)}\left(\theta_n^{(t)}, \Theta^{(t)}\right) - e_{n'}^{(t)}\left(\theta_{n'}^{(t)}, \Theta^{(t)}\right)\right) \\ &\quad + \left(\xi_n^{\texttt{Balanced},(t)} - \xi_{n'}^{\texttt{Balanced},(t)}\right).\end{aligned}$$

Therefore,

$$\begin{aligned}\left\|\theta_n^{(t+1)} - \theta_{n'}^{(t+1)}\right\| \leq\ & \left\|\theta_n^{(t)} - \theta_{n'}^{(t)}\right\| + \eta\left\|g^{(t)}_{\text{pub}}(\theta_n^{(t)}) - g^{(t)}_{\text{pub}}(\theta_{n'}^{(t)})\right\| \\ & + \eta\left\|e_n^{(t)}\left(\theta_n^{(t)}, \Theta^{(t)}\right) - e_{n'}^{(t)}\left(\theta_{n'}^{(t)}, \Theta^{(t)}\right)\right\| \\ & + \eta\left(\left\|\xi_n^{\texttt{Balanced},(t)}\right\| + \left\|\xi_{n'}^{\texttt{Balanced},(t)}\right\|\right).\end{aligned}$$

Assumption 2 (Lipschitz public on-policy gradient) *For any minibatch $B \subseteq D_{\text{pub}}$ of size b, define*

$$g_B^{\text{pub}}(\theta) := \frac{1}{b}\sum_{x\in B}\mathbb{E}_{\mathcal{Y}\sim\pi_\theta(\cdot|x)^{\otimes K}}\left[\nabla_\theta \ell_{\text{GRPO}}(\theta; x, \mathcal{Y})\right].$$

There exists a constant $L_{\text{pub}} > 0$ such that for all such B and all $\theta, \theta' \in \Theta$,

$$\left\|g_B^{\text{pub}}(\theta) - g_B^{\text{pub}}(\theta')\right\| \leq L_{\text{pub}}\|\theta - \theta'\|.$$

Note that at each round t, the server samples a minibatch $B^{(t)}_{\text{pub}} \subseteq D_{\text{pub}}$, and we write $g^{(t)}_{\text{pub}}(\theta) := g^{\text{pub}}_{B^{(t)}_{\text{pub}}}(\theta)$ for brevity.

$$\begin{aligned}\left\|\theta_n^{(t+1)} - \theta_{n'}^{(t+1)}\right\| \leq (1+\eta L_{\text{pub}})\left\|\theta_n^{(t)} - \theta_{n'}^{(t)}\right\| + \eta\left\|e_n^{(t)}(\theta_n^{(t)}, \Theta^{(t)}) - e_{n'}^{(t)}(\theta_{n'}^{(t)}, \Theta^{(t)})\right\| \\ + \eta\left(\left\|\xi_n^{\texttt{Balanced},(t)}\right\| + \left\|\xi_{n'}^{\texttt{Balanced},(t)}\right\|\right).\end{aligned}$$

Defining $\omega_i^{(t)}(\theta, \Theta^{(t)}) := \left\|e_i^{(t)}(\theta, \Theta^{(t)})\right\|$, we further get

$$\begin{aligned}\left\|\theta_n^{(t+1)} - \theta_{n'}^{(t+1)}\right\| \leq (1+\eta L_{\text{pub}})\left\|\theta_n^{(t)} - \theta_{n'}^{(t)}\right\| + \eta\omega_n^{(t)}\left(\theta_n^{(t)}, \Theta^{(t)}\right) + \eta\omega_{n'}^{(t)}\left(\theta_{n'}^{(t)}, \Theta^{(t)}\right) \\ + \eta\big(\|\xi_n^{\texttt{Balanced},(t)}\| + \|\xi_{n'}^{\texttt{Balanced},(t)}\|\big).\end{aligned}$$

For the public data step with `Balanced` aggregation, recall

$$e_i^{(t)}(\theta, \Theta^{(t)}) = g^{\texttt{Balanced}}_{B^{(t)}_{\text{pub}}, i}(\theta; \Theta^{(t)}) - g^{(t)}_{\text{pub}}(\theta).$$

For a public prompt x, let $\mathcal{Y}_i^{(0)}(x)$ denote the fully local on-policy GRPO group for client i, and let $\mathcal{Y}_i^{(\texttt{Balanced})}(x)$ denote the final group used by `Balanced`. Writing

$$\mathcal{G}(\theta; x, \mathcal{Y}) := \nabla_\theta \ell_{\text{GRPO}}(\theta; x, \mathcal{Y}),$$

we have

$$e_i^{(t)}\left(\theta, \Theta^{(t)}\right) = \frac{1}{b} \sum_{x \in B_{\text{pub}}^{(t)}} \left(\mathbb{E}\left[\mathcal{G}(\theta; x, \mathcal{Y}_i^{(\texttt{Balanced})}(x))\right] - \mathbb{E}\left[\mathcal{G}(\theta; x, \mathcal{Y}_i^{(0)}(x))\right] \right).$$

Define the random variable number $M_i^{(t)}(x, \theta)$ denoting of replacements on prompt x by `Balanced` at client i

$$M_i^{(t)}(x, \theta) \leq \left(\frac{K}{2} - c_i^{(t)}(x) \right)_+, \qquad \alpha_i^{(t)}(x) := \frac{M_i^{(t)}(x, \theta)}{K} \leq \frac{1}{2}.$$

Define the one-replacement GRPO distortion

$$\beta_i^{(t)}(x; \theta) := \sup_{(\mathcal{Y}, \widetilde{\mathcal{Y}})} \left\| \mathbb{E}\big[\mathcal{G}(\theta; x, \widetilde{\mathcal{Y}}) - \mathcal{G}(\theta; x, \mathcal{Y})\big] \right\|,$$

where the supremum is over all pairs $(\mathcal{Y}, \widetilde{\mathcal{Y}})$ that differ in exactly one coordinate, (happens when one local response in $\mathcal{Y}$ is replaced by a donor-correct response in $\widetilde{\mathcal{Y}}$), while the other $K-1$ responses are identical. If $M_i^{(t)}(x, \theta) = m$, let $\mathcal{Y}_i^{(0)}(x), \mathcal{Y}_i^{(1)}(x), \ldots, \mathcal{Y}_i^{(m)}(x)$ be the sequence of intermediate groups obtained by performing the m replacements one at a time, so that $\mathcal{Y}_i^{(m)}(x) = \mathcal{Y}_i^{(\texttt{Balanced})}(x)$. Then

$$\mathcal{G}(\theta; x, \mathcal{Y}_i^{(m)}(x)) - \mathcal{G}(\theta; x, \mathcal{Y}_i^{(0)}(x)) = \sum_{s=1}^{m} \Big(\mathcal{G}(\theta; x, \mathcal{Y}_i^{(s)}(x)) - \mathcal{G}(\theta; x, \mathcal{Y}_i^{(s-1)}(x)) \Big),$$

and hence

$$\left\| \mathbb{E}[\mathcal{G}(\theta; x, \mathcal{Y}_i^{(\texttt{Balanced})}(x))] - \mathbb{E}[\mathcal{G}(\theta; x, \mathcal{Y}_i^{(0)}(x))] \right\| \leq M_i^{(t)}(x, \theta)\, \beta_i^{(t)}(x; \theta).$$

Therefore,

$$\omega_i^{(t)}(\theta) \leq \frac{1}{b} \sum_{x \in B_{\text{pub}}^{(t)}} \mathbb{E}[M_i^{(t)}(x)]\, \beta_i^{(t)}(x; \theta) = K \cdot \frac{1}{b} \sum_{x \in B_{\text{pub}}^{(t)}} \mathbb{E}[\alpha_i^{(t)}(x)]\, \beta_i^{(t)}(x; \theta).$$

Hence, the additional drift induced by the `Balanced` step is controlled by the product of two factors: the fraction of responses that must be replaced, and the sensitivity of the GRPO gradient to a single such replacement.

This decomposition clarifies when the off-policy drift of `Balanced` is small. First, $\alpha_i^{(t)}(x)$ is small when the local model already performs reasonably well on the public prompt, so that few or no replacements are needed. This regime naturally becomes more common later in training or on easier public prompts. Second, $\beta_i^{(t)}(x; \theta)$ is small when donor-correct responses remain reasonably aligned with the local policy on the shared public prompt, so that replacing one response does not substantially perturb the GRPO update. In particular, $\beta_i^{(t)}(x; \theta)$ is expected to be smaller when clients have not drifted too far apart on the public data and when the GRPO group statistics are stable, since replacing a single response then only mildly changes the group-relative advantages - This partially formalizes the intuition that `Balanced` has more off policy tendency compared to FedAvg + GRPO.

Additionally, the drift now depends on L_{pub} rather than L_{priv}. This can be a lot smaller, since L_{pub} represents the maximum smoothness over mini-batches in the public dataset, whereas L_{priv} is taken over all mini-batches across clients—a substantially larger and typically more heterogeneous set.

The above perspective also explains why `Balanced` is particularly appealing under strong client heterogeneity. For a purely private local step, the client-drift term is driven directly by the discrepancy between private gradients across heterogeneous clients. In contrast, under `Balanced`, all clients are anchored to the same public minibatch, and the remaining discrepancy enters only through the off-policy distortion terms $\omega_i^{(t)}$. Therefore, whenever private-data heterogeneity is large while the public anchor remains effective that is, whenever In words, heterogeneity create the regime in which public-data coordination is most useful.

B Additional Experimental Details and Results

B.1 Client Dataset split

For federated training, we partition both datasets across 4 clients using heterogeneous splits (based on topics); full details are provided in the appendix. In the MATH setting, each client receives 2,500 distinct non-overlapping samples, while the server maintains 1,250 public samples and a held-out test set of 1,250 samples. For DeepMath, we first randomly subsample 20K problems from the original 103K-problem corpus; each client then receives 4,000 samples, while the server again maintains 3,000 public samples and 1,000 held-out test samples. For the medical reasoning experiments, each client possesses 3956 datapoints while the size of the test set is 1977 and public dataset also has 1977 datapoints.

Notably, in both settings, the public dataset is substantially smaller than the total amount of private data distributed across clients. The code required to reproduce the results presented in the paper is included in the supplementary material.

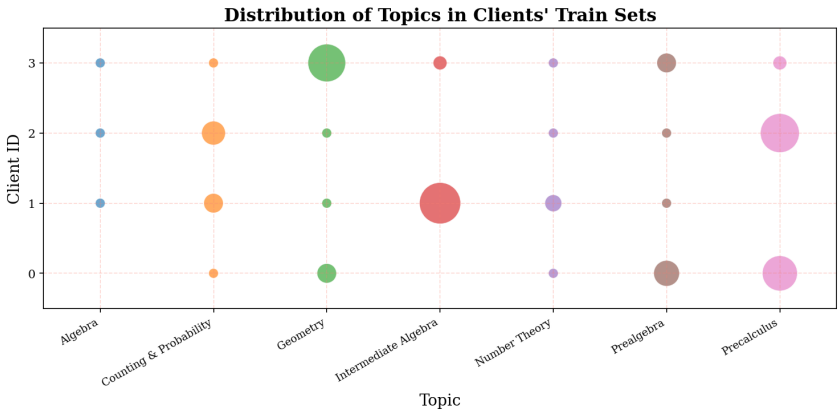

Figure 2: Bubble plot (quantized by 100 samples) of per client topic distribution heterogeneity for the MATH dataset experiments

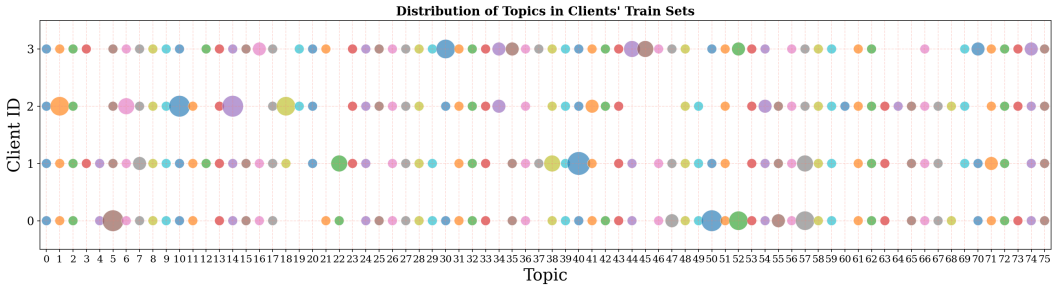

Figure 3: Bubble plot (quantized by 100 samples) of per client topic distribution (topic number) heterogeneity for the DeepMath dataset experiments

B.2 Best checkpoint results

In tables 7, 8 we report the best checkpoint test performance of the models throughout the training rounds for our main experiments. The final checkpoint numbers are the same as the best checkpoint numbers in most high local step size regimes.

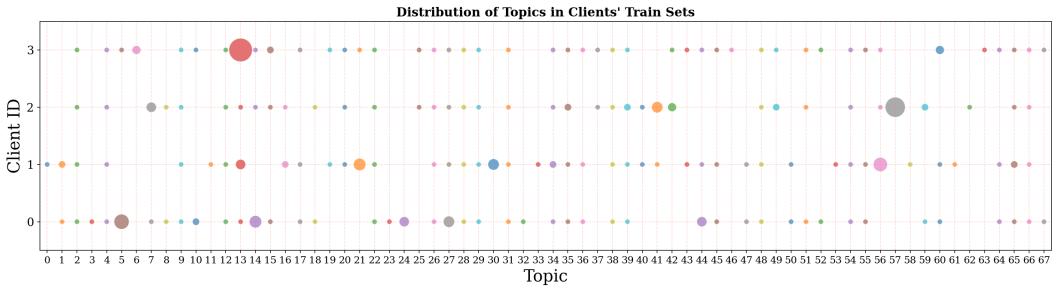

Figure 4: Bubble plot (quantized by 100 samples) of per client topic distribution heterogeneity for the high heterogeneity DeepMath dataset experiments (Table 4)

Model	Method	MATH				DeepMath			
		τ=10	τ=40	τ=90	τ=120	τ=10	τ=40	τ=90	τ=120
Qwen3-1.7B	Base model	55.2	55.2	55.2	55.2	14.9	14.9	14.9	14.9
	FedAvg-GRPO	**78.3**	76.1	76.3	75.9	49.5	52.7	50.4	50.7
	FedProx-GRPO	77.5	76.7	76.5	75.6	52.0	52.3	48.0	47.7
	FedAvg-`PubSwap`	77.0	**76.9**	**76.9**	**76.6**	**53.8**	**54.6**	**53.3**	**55.8**
Qwen2.5-Math-1.5B	Base model	58.4	58.4	58.4	58.4	34.9	34.9	34.9	34.9
	FedAvg-GRPO	**73.7**	73.1	71.9	71.1	53.1	52.3	50.5	49.3
	FedProx-GRPO	72.8	71.4	71.5	70.6	52.6	52.1	48.0	49.4
	FedAvg-`PubSwap`	73.1	**73.5**	**73.2**	**71.9**	**54.5**	**53.1**	**51.0**	**53.1**

Table 7: Best checkpoint pass@1 performance comparison on MATH [Hendrycks et al., 2021] and DeepMath [He et al., 2025] across different numbers of local steps (τ). The `Balanced` method and a `swap` period of 2 is used here for response aggregation with `PubSwap`.

Method	τ=10	τ=40	τ=90	τ=120
Base model	49.2	49.2	49.2	49.2
FedAvg-GRPO	**59.7**	58.9	57.9	57.7
FedAvg-`PubSwap`	58.3	**59.5**	**58.5**	**58.1**

Table 8: Best checkpoint pass@1 performance of the model Llama3.2-3B-Instruct on medical reasoning across different numbers of local steps (τ). The `Balanced` method and a `swap` period of 2 is used here for response aggregation with `PubSwap`.

Parameter	**Value**	**Parameter**	**Value**
Pretrained Model	Qwen3-1.7B	Num Clients	4
Grad epochs per GRPO step	2	Train Batch Size	8
Max prompt length	1024	Max response length	2048
LoRA rank	32	LoRA alpha	64
LoRA modules	all linear	Gen. per prompt	8
Learning rate	1×10^{-5}	Clip ratio low	0.2
Weight decay	0.01	Gradient clip	1.0
Clip ratio high	0.25	KL loss coefficient	0.0001
Entropy coefficient	0	Rollout engine	vllm
Rollout temperature	0.7	Validation temperature	0.7
Validation batch size	512	Remove padding	Enabled
Device	2/4 $\times$ NVIDIA H100	Aggregation method	FedIT

Table 9: Configuration for Qwen3-1.7B

Parameter	Value	Parameter	Value
Pretrained Model	Qwen2.5-Math-1.5B	Num Clients	4
Grad epochs per GRPO step	2	Train Batch Size	8
Max prompt length	1024	Max response length	2048
LoRA rank	32	LoRA alpha	64
LoRA modules	all linear	Gen. per prompt	8
Learning rate	1×10^{-5}	Clip ratio low	0.2
Weight decay	0.01	Gradient clip	1.0
Clip ratio high	0.25	KL loss coefficient	0.0001
Entropy coefficient	0	Rollout engine	vllm
Rollout temperature	0.7	Validation temperature	0.7
Validation batch size	512	Remove padding	Enabled
Device	2/4 $\times$ NVIDIA H100	Aggregation method	FedIT

Table 10: Configuration for Qwen2.5-Math-1.5B

Parameter	Value	Parameter	Value
Pretrained Model	Qwen3-4B-Instruct-2507	Num Clients	4
Grad epochs per GRPO step	2	Train Batch Size	4
Max prompt length	1024	Max response length	2048
LoRA rank	32	LoRA alpha	64
LoRA modules	all linear	Gen. per prompt	8
Learning rate	1×10^{-5}	Clip ratio low	0.2
Weight decay	0.01	Gradient clip	1.0
Clip ratio high	0.25	KL loss coefficient	0.0001
Entropy coefficient	0	Rollout engine	vllm
Rollout temperature	0.7	Validation temperature	0.7
Validation batch size	512	Remove padding	Enabled
Device	2/4 $\times$ NVIDIA H100	Aggregation method	FedIT

Table 11: Configuration for Qwen3-4B-Instruct

Parameter	Value	Parameter	Value
Pretrained Model	Llama-3.2-3B-Instruct	Num Clients	4
Grad epochs per GRPO step	2	Train Batch Size	8
Max prompt length	1024	Max response length	2048
LoRA rank	32	LoRA alpha	64
LoRA modules	q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj	Gen. per prompt	8
Learning rate	1×10^{-5}	Clip ratio low	0.2
Weight decay	0.01	Gradient clip	1.0
Clip ratio high	0.25	KL loss coefficient	0.0001
Entropy coefficient	0	Rollout engine	vllm
Rollout temperature	0.7	Validation temperature	0.7
Validation batch size	512	Remove padding	Enabled
Device	4$\times$NVIDIA H100	Aggregation method	FedIT

Table 12: Configuration for Llama-3.2-3B-Instruct