

# FEDERATED AGENT REINFORCEMENT LEARNING

**Anonymous authors**

Paper under double-blind review

## ABSTRACT

Autonomous AI Agents powered by LLMs have shown remarkable abilities in diverse domains. However, the training process typically require *centralized* collection of large amounts of real-world user data, posing substantial privacy and regulatory concerns. To this end, we explore a new *decentralized* training paradigm, namely **FEDAGENT** (Federated Agent Reinforcement Learning), which enables collaborative learning of AI agents across distributed clients without sharing local data. Moreover, we construct the first decentralized agent learning environment **FEDAGENTGYM**, which includes four types of LLM agents, two application scenarios (WebShop and ALFWorld), three variations of decentralized settings, and three newly defined heterogeneity challenges (*Preference Heterogeneity*, *Coverage Heterogeneity*, and *Hardness Heterogeneity*), to systematically investigate its effectiveness and impact factors. Extensive theoretical and empirical studies show that FEDAGENT can have comparable performance to the centralized training paradigm and exhibit strong robustness against heterogeneities, which shows the feasibility of training AI agents without sacrificing data privacy. The code is available [here](#).

## 1 INTRODUCTION

The rapid advancement of AI agents, especially those powered by Large Language Models (LLMs), has demonstrated remarkable capabilities across diverse domains, from web navigation to embodied environments (Zhang et al., 2025; Gao et al., 2025; Liu et al., 2025). However, training these agents typically requires *centralized* access to vast amounts of users’ real-world task query and trajectory data, which are inherently privacy-sensitive and hard to acquire due to regulatory compliance. Thus, a foundational question is: *how to train AI agents while protecting users’ data privacy?*

In this paper, we explore a new *decentralized* training paradigm, namely **FEDAGENT (Federated Agent Reinforcement Learning)**, which enables collaborative learning of AI agents, particularly LLMs, across distributed clients without sharing local data. In each round, the server distributes the current model to selected clients, who then train locally on their own data and send back their updated models. The server aggregates these updates by averaging them to create an improved global model for the next round. This process repeats iteratively, facilitating distributed LLM agent training while preserving data privacy since only model parameters, not raw data, are exchanged.

Compared with the previous federated learning literature, FEDAGENT is faced with fundamentally new challenges. The majority of existing federated learning research has concentrated on supervised classification tasks. There are also recent works that have explored federated reinforcement learning (FRL) for traditional RL settings (Liu et al., 2024; Qi et al., 2021; Kairouz et al., 2021). However, both of them operate under distinct assumptions compared to LLM agent learning. Supervised federated learning is usually built on static data distributions and one-shot predictions, while traditional FRL typically assumes simple rewards, well-defined state and action spaces. In contrast, LLM agent learning involves *diverse task formulations, multi-step natural language reasoning, and complex environment interactions*, which create entirely new challenges for federated paradigms.

To systematically investigate the effectiveness of this new training paradigm as well as the impact factors, we built the first decentralized agent learning environment **FEDAGENTGYM**, which incorporate four types of LLM agents (Qwen2.5- $\{1.5, 3, 7\}$ B-Instruct and Llama-3.2-3B-Instruct), two applications (WebShop and ALFWorld), three variations of decentralized settings (samples per client, clients selected per communication round, and local training epochs per client per round).

Importantly, since the existing heterogeneity challenges in federated learning have mostly been defined in the context of supervised classification tasks (Ye et al., 2023; Gao et al., 2022), which

054  
055  
056  
057  
058  
059  
060  
061  
062  
063  
064  
065  
066  
067  
068  
069  
070  
071  
072  
073  
074  
075  
076  
077  
078  
079  
080  
081  
082  
083  
084  
085  
086  
087  
088  
089  
090  
091  
092  
093  
094  
095  
096  
097  
098  
099  
100  
101  
102  
103  
104  
105  
106  
107

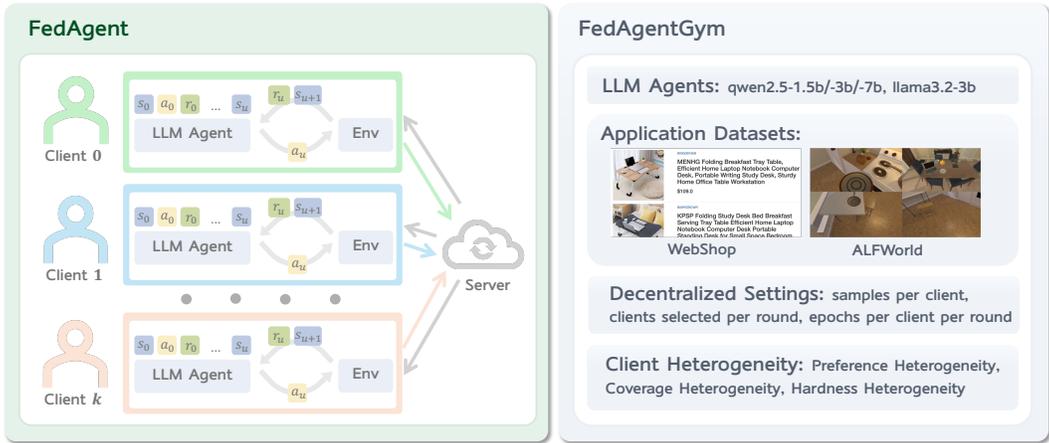


Figure 1: An Illustration of FEDAGENT and FEDAGENTGYM.

focus on label skew, feature shift, or quantity imbalance, they are not directly applicable to LLM agent learning. Thus, we propose three new and orthogonal definitions of client heterogeneity unique to decentralized AI agent learning: **Preference Heterogeneity**, where clients may prefer distinct types of tasks; **Coverage Heterogeneity**, where the task sampling scope may vary across clients; **Hardness Heterogeneity**, where the overall difficulty of tasks may differ among clients. Moreover, we carefully design three novel client partitioning strategies PREFERENCEPARTITION, COVERAGEPARTITION, and HARDNESSPARTITION accordingly, grounded in mathematical techniques such as *Gaussian Noise*, *Multinomial Sampling* and *Beta Distribution*. These strategies allow us to precisely control the extent of one type of heterogeneity across clients with a single hyperparameter, while keeping the other characteristics of the client distribution unchanged. We then incorporate them into FEDAGENTGYM to isolate and analyze the impact of each form of heterogeneity on FEDAGENT separately.

To validate the effectiveness of FEDAGENT, we first conduct a theoretical analysis on its convergence. Then, through extensive and systematic empirical studies with FEDAGENTGYM, we demonstrate that FEDAGENT consistently outperforms *local agent training* and can achieve performance comparable with *centralized agent training*, despite never sharing local data. Moreover, FEDAGENT exhibits strong robustness to the aforementioned preference, coverage, and hardness heterogeneity challenges, while revealing sensitivities to certain decentralized configurations. Overall, our studies show the potential of scalable agent learning without sacrificing data privacy, provide valuable insights that inform practical deployment, and open new research directions in the field of agent learning.

Our contributions can be summarized as follows:

- We explored a new decentralized paradigm of training AI agents, namely **FEDAGENT (Federated Agent Reinforcement Learning)**, which enables collaborative agent learning across distributed clients without sharing local data. We also provide a theoretical analysis on its convergence.
- We propose to categorize the new client heterogeneity challenges in decentralized agent learning into *Preference Heterogeneity*, *Coverage Heterogeneity*, and *Hardness Heterogeneity*. To investigate how each type of heterogeneity affects the performance, we introduce three novel client partitioning methods: PREFERENCEPARTITION, COVERAGEPARTITION, and HARDNESSPARTITION.
- We constructed the first decentralized agent learning environment **FEDAGENTGYM**, which includes four types of LLM agents, two applications (WebShop and ALFWorld), three variations of decentralized settings, and three heterogeneity challenges, to analyze the performance of FEDAGENT systematically and controllably, and offer insights to guide future development.
- Extensive studies show that FEDAGENT not only beats the single-client local training paradigm but also can achieve comparable performance to the centralized agent learning paradigm. Furthermore, FEDAGENT shows high robustness against the three types of heterogeneity challenges. We also provide insights on its sensitivity to different decentralized settings.
- We release our code and environment as an extendable open-source library to inspire more future works in this new direction. The link to the repository is available [here](#).

**Algorithm 1** FEDAGENT with Client and Server training

---

**Require:** Total clients  $K$ , rounds  $T$ , clients-per-round  $M$ , local steps  $\tau$ , learning rate  $\eta$   
**Ensure:** Final LLM-based global policy parameters  $\theta_{\text{final}}$

- 1: Initialize global policy parameters  $\theta_0$  (an LLM)
- 2: **for**  $t = 0$  **to**  $T - 1$  **do**
- 3:     **Server:** sample client subset  $S_t \subset [K]$  with  $|S_t| = M$  (uniform without replacement)
- 4:     **Server:** broadcast  $\theta_t$  to all  $k \in S_t$
- 5:     **for each**  $k \in S_t$  **in parallel do**
- 6:         Set local iterate  $\theta_{k,t,0} \leftarrow \theta_t$
- 7:         **for**  $i = 0$  **to**  $\tau - 1$  **do**
- 8:             Collect a mini batch of trajectories  $B_{k,t,i}$  using policy  $\pi_{\theta_{k,t,i}}$  in environment  $\mathcal{M}_k$
- 9:             Estimate policy gradient for  $J_k(\theta_{k,t,i})$  on client  $k$ :  

$$g_{k,t,i} \leftarrow \nabla_{\theta} \hat{J}_k(\theta_{k,t,i}; B_{k,t,i}) \quad (\text{e.g., GRPO})$$
- 10:             Local update:  $\theta_{k,t,i+1} \leftarrow \theta_{k,t,i} + \eta g_{k,t,i}$
- 11:         **end for**
- 12:         Client returns local model  $\theta_{k,t,\tau}$   $\triangleright$  equivalently  $\Delta\theta_{k,t} = \theta_{k,t,\tau} - \theta_t$
- 13:     **end for**
- 14:     **Server:** Aggregation via model averaging:  

$$\theta_{t+1} \leftarrow \frac{1}{M} \sum_{k \in S_t} \theta_{k,t,\tau} \quad (\text{equivalently } \theta_{t+1} = \theta_t + \frac{1}{M} \sum_{k \in S_t} (\theta_{k,t,\tau} - \theta_t))$$
- 15: **end for**
- 16: **return**  $\theta_{\text{final}} \leftarrow \theta_T$

---

## 2 FEDAGENT: FEDERATED AGENT REINFORCEMENT LEARNING

As shown in Algorithm 1, we consider a federated reinforcement learning setup for FEDAGENT. A population of clients are indexed by  $k \in [K] = \{0, \dots, K-1\}$ . Training proceeds for communication rounds  $t = 0, \dots, T-1$ . At round  $t$ , the server samples a subset  $S_t \subset [K]$  of size  $|S_t| = M$  uniformly without replacement, broadcasts the current global policy parameters  $\theta_t$ , and aggregates the participating clients’ locally updated parameters.

**LLM Agent Training.** The agent is a parametric policy  $\pi_{\theta}$  (an LLM) that, conditioned on a task description  $c$  and an interaction history  $h_u$  up to step  $u$ , produces an action  $a_u \sim \pi_{\theta}(\cdot | h_u, c)$ . An action often contains both a *sequence of free-form tokens* (i.e., the agent’s intermediate reasoning) and *environment-facing choice* (e.g., tool API calling). Each client  $k$  operates in a Markov Decision Process (MDP) environment  $\mathcal{M}_k = (\mathcal{S}_k, \mathcal{A}_k, P_k, r_k, \rho_k, \gamma)$  with state space  $\mathcal{S}_k$ , action space  $\mathcal{A}_k$ , transition kernel  $P_k$ , reward function  $r_k$ , initial-state distribution  $\rho_k$ , and discount  $\gamma \in (0, 1]$ . Client  $k$  also has a distribution  $\mathcal{D}_k$  over textual task descriptions  $c \in \mathcal{C}_k$ . Fix  $k$  and a task description  $c \sim \mathcal{D}_k$ . The agent interacts with  $\mathcal{M}_k$  for horizon  $H$ , producing a trajectory:

$$\chi = (c, s_0, a_0, r_0, \dots, s_H), \quad s_0 \sim \rho_k(\cdot | c), \quad a_u \sim \pi_{\theta}(\cdot | h_u, c), \quad s_{u+1} \sim P_k(\cdot | s_u, a_u, c).$$

The discounted return of  $\chi$  is  $R(\chi) = \sum_{u=0}^{H-1} \gamma^u r_u$ . It is worth noting that when the LLM agents generate  $H$  consecutive textual actions  $(a_0, \dots, a_{(H-1)})$  in a trajectory  $\chi$ , each action may span thousands of tokens, considering LLM agents’ long reasoning capacity (DeepSeek-AI et al., 2025). This makes token-level credit assignment across the trajectory particularly challenging.

**Local objective (client  $k$ ).** Client  $k$  aims to maximize the expected episodic return of the policy on its own environments and tasks:

$$J_k(\theta) = \mathbb{E}_{c \sim \mathcal{D}_k} \mathbb{E}_{\chi \sim (\pi_{\theta}, \mathcal{M}_k, c)} [R(\chi)]. \quad (1)$$

During round  $t$ , each participating client initializes a local iterate at the broadcast model,  $\theta_{k,t,0} \leftarrow \theta_t$ , and performs  $\tau$  steps of stochastic policy optimization. At local step  $i \in \{0, \dots, \tau - 1\}$ , the client collects a batch of trajectories  $B_{k,t,i} = \{\chi^{(b)}\}_{b=1}^{N_{k,t,i}}$  by interacting with  $\mathcal{M}_k$  under  $\pi_{\theta_{k,t,i}}$  and computes a policy-gradient estimate:

$$g_{k,t,i} = \nabla_{\theta} \widehat{J}_k(\theta_{k,t,i}; B_{k,t,i}) = \frac{1}{N_{k,t,i}} \sum_{b=1}^{N_{k,t,i}} \sum_{u=0}^{H(\chi^{(b)})-1} \nabla_{\theta} \log \pi_{\theta_{k,t,i}}(a_u^{(b)} | h_u^{(b)}, c^{(b)}) \widehat{A}_u^{(b)}, \quad (2)$$

where  $\widehat{A}_u^{(b)}$  is any valid return/advantage signal (e.g., a GRPO-style estimator (Shao et al., 2024)). The local update is  $\theta_{k,t,i+1} = \theta_{k,t,i} + \eta g_{k,t,i}$ , ( $i = 0, \dots, \tau - 1$ ) with step size  $\eta > 0$ . After  $\tau$  steps the client returns its local model  $\theta_{k,t,\tau}$  (equivalently the update  $\Delta\theta_{k,t} = \theta_{k,t,\tau} - \theta_t$ ) to the server.

**Global objective and aggregation.** The federated learning goal is to maximize a weighted average of client objectives:

$$J_{\text{global}}(\theta) = \sum_{k=0}^{K-1} w_k J_k(\theta), \quad w_k \geq 0, \quad \sum_{k=0}^{K-1} w_k = 1. \quad (3)$$

In the FEDAGENT, the server uses uniform model averaging over the  $M$  participating clients each round (i.e.,  $w_k = \frac{1}{K}$  conceptually, with partial participation realized by  $S_t$ ). Upon receiving the  $\tau$ -step local models  $\{\theta_{k,t,\tau}\}_{k \in S_t}$ , the server performs model averaging:  $\theta_{t+1} = \frac{1}{M} \sum_{k \in S_t} \theta_{k,t,\tau} = \theta_t + \frac{1}{M} \sum_{k \in S_t} (\theta_{k,t,\tau} - \theta_t)$ . After  $T$  rounds the server outputs  $\theta_{\text{final}} = \theta_T$ .

### 3 FEDAGENTGYM: A DECENTRALIZED AGENT LEARNING ENVIRONMENT

#### 3.1 LLM AGENTS AND APPLICATION DATASETS

FEDAGENTGYM is designed as an environment to investigate the impact factors of training AI agents, especially LLM, in a decentralized way. It includes four types of LLM agents, including Qwen2.5- $\{1.5, 3, 7\}$ B-Instruct and Llama-3.2-3B-Instruct, and two challenging application datasets (WebShop (Yao et al., 2022) and ALFWorld (Shridhar et al., 2020)), which require complex reasoning process and multi-step environment interactions. We adopt these two datasets to simulate the real-world scenarios where data privacy concerns are paramount.

WebShop is a web-based interactive platform that evaluates LLM agents within authentic e-commerce scenarios. Task completion requires agents to navigate a simulated HTML shopping interface to locate, browse, and purchase appropriate items. The dataset features an extensive catalog of over 1.1 million products paired with 12,000 user instructions, creating a rich and varied action space.

ALFWorld provides an embodied simulation benchmark that evaluates LLM agents’ capacity for sequential decision-making tasks. Each scenario presents the agent with a textual objective that must be achieved through iterative environment interaction. The dataset encompasses 3,827 task instances spanning six types of household activities: Pick & Place (Pick), Examine in Light (Look), Clean & Place (Clean), Heat & Place (Heat), Cool & Place (Cool), and Pick Two & Place (Pick2).

#### 3.2 DECENTRALIZED SETTINGS

We comprehensively examine the impact of different decentralized settings on FEDAGENT performance across three critical dimensions. First, we vary **the number of samples per client**, which determines the sampling scope for each LLM agent’s exploration of the action space and response generation, directly affecting both the diversity of experiences collected and the quality of policy gradient estimates. Second, we change **the number of clients selected per communication round**, controlling both the computational parallelism and the degree of heterogeneity in exploration strategies incorporated during global model aggregation. Third, we adjust **the number of local training batches per client per round**, governing the extent of local optimization on the sampled trajectories before synchronization with the central server. These parameters collectively influence fundamental trade-offs between exploration diversity, communication overhead, and convergence stability in the federated setting. Through extensive studies across these dimensions, we characterize how different decentralized training design choices affect the final policy performance of FEDAGENT.

#### 3.3 HETEROGENEITY CHALLENGES

To systematically evaluate how FEDAGENT performs under realistic client distributions, we propose three novel and orthogonal heterogeneity definitions, as conventional heterogeneity dimensions in federated classification tasks (e.g., feature or label skew) (Ye et al., 2023; Gao et al., 2022) are not

216 directly applicable. We also propose the corresponding client partitioning strategies, allowing us to  
 217 understand the individual impact of different heterogeneity types separately.

218  
 219 **Preference Heterogeneity: When Clients Have Different Task Preferences.** In real-world  
 220 federated learning, different clients often *prefer distinct types of tasks*. For example, in the ALFWorld,  
 221 some users might frequently interact with kitchen-related tasks (like “put the apple in the fridge”),  
 222 while others primarily encounter bedroom tasks (like “examine the lamp”). In WebShop, some may  
 223 have mostly electronics searches while others mainly focus on clothing or home goods.

224 To simulate this preference heterogeneity, we propose the PREFERENCEPARTITION algorithm. The  
 225 pseudo code is illustrated in Algorithm 2 in Appendix B.1. We model this by starting with the global  
 226 distribution of task categories and introducing controlled noise to create client-specific preferences.  
 227 Specifically, we add **Gaussian Noise** to the log-probabilities of the global category distribution,  
 228 apply softmax normalization, and use the resulting probabilities to sample  $L$  instructions per client  
 229 via **Multinomial Sampling**. This approach allows precise control over client distributions with a  
 230 hyperparameter  $\omega$  on topical preference heterogeneity, while maintaining the same total dataset size  
 231 and per-client instruction count. More specifically, small noise values produce clients with similar  
 232 task distributions, while larger noise creates highly specialized clients with distinct preferences.

233 **Coverage Heterogeneity: When Clients Have Different Task Sampling Scopes.** Even when  
 234 clients encounter similar types of tasks, they may face vastly different quantities. A larger quantity of  
 235 tasks indicates *coverage of a broader sampling scope per epoch* in reinforcement learning (we follow  
 236 the setting in (Feng et al., 2025) to iteratively sample with replacement from the local data each  
 237 epoch), while the sampling size remains fixed. Importantly, this differs from the quantity imbalance  
 238 in conventional supervised federated classification tasks, where *training proceeds over the entire*  
 239 *dataset each epoch*. In WebShop, for instance, some users might have extensive browsing histories  
 240 with hundreds of product interactions, while others have only completed a few shopping sessions.

241 To model this coverage heterogeneity, we develop the COVERAGEPARTITION algorithm. The pseudo  
 242 code is shown in Algorithm 3. We fix a global overlap target  $r$  (representing the average number of  
 243 clients that see each instruction) and draw each client’s data quantity from a **Beta Distribution**, which  
 244 we then map to the range  $[L_{\min}, L_{\max}]$ . Task instructions are allocated to clients using weighted  
 245 sampling without replacement to satisfy both individual client quotas and the global overlap constraint.  
 246 This method isolates the effect of task sampling scope on FEDAGENT performance while keeping  
 247 the underlying task distribution consistent across clients. Also, this method controls the extent of  
 248 coverage heterogeneity via hyperparameter  $\xi$  without impacting the overall mean of client quantities.

249 **Hardness Heterogeneity: When Clients Face Different Task Difficulties.** A particularly im-  
 250 portant but often overlooked source of heterogeneity is the overall difficulty of tasks that different  
 251 clients encounter, which can be quantified by the *success rate* of tasks. For example, in ALFWorld,  
 252 some clients might consistently face simple navigation tasks with high success rates, while others  
 253 encounter complex multi-step reasoning tasks that frequently result in failure.

254 As demonstrated in Algorithm 4, our proposed HARDNESSPARTITION algorithm addresses this by  
 255 partitioning the task instruction pool into “successful” and “unsuccessful” examples with a pretrained  
 256 checkpoint. Then, using our COVERAGEPARTITION method, we first distribute successful instructions  
 257 according to a **Beta Distribution** that determines each client’s success rate. We then fill remaining  
 258 slots with unsuccessful examples sampled uniformly, ensuring all clients have exactly  $L$  instructions.  
 259 This method enables us to study how different success rate distributions, which are controlled by a  
 260 hyperparameter  $\xi'$  and measures the extent of hardness of task distributions for each client, affect  
 261 FEDAGENT while maintaining consistent dataset sizes and global overlap patterns across all clients.

## 262 4 THEORETICAL ANALYSIS ON CONVERGENCE

263 **Theorem 1** (Convergence of FEDAGENT). *Under Assumptions 1–5, suppose that at each communi-  
 264 cation round  $t$  the server uniformly samples without replacement a subset  $S_t \subset [K]$  of size  $M \leq K$   
 265 and aggregates only those clients’ updates:  $\theta_{t+1} = \theta_t + \frac{1}{M} \sum_{k \in S_t} \Delta \theta_{k,t}$ , with the same local inner  
 266 loop and notation as in Algorithm 1. Let each selected client perform  $\tau$  local steps with stepsize  $\eta$ .  
 267 Choose the stepsize  $\eta = \frac{1}{L\tau}$  and let  $\tilde{\theta}$  be a uniform random iterate drawn from  $\{\theta_t\}_{t=0}^{T-1}$ . Then*

268  
 269 
$$\mathbb{E} \left[ J(\theta^*) - J(\tilde{\theta}) \right] \leq \frac{L}{\mu T} (J(\theta^*) - J(\theta_0)) + \frac{1}{2\mu} \left[ \frac{G^2 + \sigma^2}{M} + \frac{2(K - M)}{M(K - 1)} \zeta^2 + \frac{(\tau - 1)^2}{2\tau^2} (G^2 + \sigma^2) \right].$$

In particular, the convergence rate remains  $O(1/T)$ . The “noise floor” comprises: (i) an  $O(1/M)$  local stochastic term, (ii) an  $O(\frac{K-M}{M(K-1)})$  heterogeneity term due to client sampling, and (iii) an  $O(\frac{(\tau-1)^2}{\tau^2})$  local-drift term that vanishes when  $\tau = 1$ .

**Remark.** Theorem 1 establishes that the FEDAGENT paradigm converges to a neighborhood of the optimum under the PL and smoothness conditions. With stepsize  $\eta = 1/(L\tau)$  and  $T$  communication rounds, the suboptimality decomposes into (i) a transient term that decays as  $O(1/T)$  and (ii) a time-invariant “noise floor”. The floor tightens with a larger  $M$ , vanishes in the full-participation and single-step homogeneous limit ( $M = K$ ,  $\tau = 1$ ,  $\zeta^2 \approx 0$ ), and otherwise quantifies the computation-communication trade-off. The proof of Theorem 1 is in Appendix C. The key implications are:

- Convergence rate ( $O(1/T)$ ):** The term  $\frac{L}{\mu T}(J(\theta^*) - J(\theta_0))$  exhibits a linear-in- $1/T$  convergence rate with respect to the number of communication rounds. Better conditioning (smaller  $L/\mu$ ) accelerates approach towards the asymptotic regime.
- Effect of partial participation ( $O(1/M)$ ):** The variance term  $\frac{G^2 + \sigma^2}{M}$  decays inversely with the number of participating clients each round. Increasing  $M$  reduces stochastic noise in the aggregated update. In the limit  $M = K$ , it matches the variance level under full participation.
- Client sampling and heterogeneity ( $O(\frac{K-M}{M(K-1)})$ ):** The middle term  $\frac{K-M}{M(K-1)} \zeta^2$  is induced by client sampling each round without replacement under the heterogeneity assumption. It vanishes when  $M = K$  and grows with smaller  $M$  and larger heterogeneity  $\zeta^2$ , implying the potential benefits of a larger number of clients each round, stratified or clustered client sampling.
- Local-drift from multiple local steps ( $O(\frac{(\tau-1)^2}{\tau^2})$ ):** Performing  $\tau > 1$  local steps introduces a bias captured by  $\frac{(\tau-1)^2}{2\tau^2}(G^2 + \sigma^2)$ . This term is 0 at  $\tau = 1$  and approaches  $\frac{1}{2}(G^2 + \sigma^2)$  as  $\tau \rightarrow \infty$ , quantifying the classic trade-off between fewer communications and increased drift.
- Noise floor and tuning guidelines:** The bracketed expression in Theorem 1 is a  $T$ -independent error floor. Once the  $O(1/T)$  term becomes negligible as the number of rounds grows, additional rounds do not improve the bound unless one (a) increases  $M$ , (b) reduces heterogeneity (e.g., via smarter client selection that lowers  $\zeta^2$ ), or (c) decreases  $\tau$  to curb local drift.

## 5 MAIN EXPERIMENTS

**Experiment Setup.** In this section, we aim to investigate the performance of FEDAGENT under a uniform client distribution, which is independent of the aforementioned three types of client heterogeneities. We partitioned the whole dataset (WebShop or ALFWorld) into 100 clients. Each client has 100 task instructions and there is a potential overlap between clients. 2 clients are randomly selected each round. Each client is trained for 3 epochs per round, with a total of 70 rounds and 210 epochs overall. For each epoch, 64 tasks are sampled iteratively with replacement from local data.

As for FEDAGENT, we adopt GRPO (Shao et al., 2024) for policy optimization. Then, following the literature in federated learning (Liu et al., 2024), we select two typical baselines: **Centralized Agent Training** and **Local Agent Training**. Centralized Agent Training uses the full dataset (*i.e.*, 64 tasks are sampled iteratively from the whole dataset each epoch), while Local Agent Training uses only a specific client’s dataset (we selected client index 21, 42, or 84 as the baselines). Both of them run for the same total epochs as FEDAGENT and also adopt GRPO for policy optimization.

**Result Analysis.** As shown in Table 1, **FEDAGENT consistently outperforms Local Agent Training and achieves comparable performance to Centralized Agent Training**. For instance, on ALFWorld using Qwen2.5-1.5B-Instruct, FEDAGENT achieves a 61.7% success rate compared to local training variants that range from 47.7% to 57.0%, while nearly matching the centralized training performance of 57.8%. This pattern is consistently observed across different model scales (1.5B, 3B, 7B), model architectures (qwen and llama), and client indexes (21, 42, and 84). Similarly, on the WebShop benchmark, FEDAGENT maintains this advantage with Qwen2.5-7B-Instruct achieving 68.9% success rate versus local training with different indexes ranging from 33.6% to 49.2%, while remaining competitive with centralized training at 64.7%. These results demonstrate the advantage of FEDAGENT in achieving competitive performance while preserving users’ data privacy inherently.

Method	ALFWorld							WebShop	
	Pick	Look	Clean	Heat	Cool	Pick2	All	Score	Succ.
<i>Qwen2.5-1.5B-Instruct</i>									
Local (Client 21)	42.9	25.0	38.5	37.5	14.3	14.3	29.7	69.9	57.0
Local (Client 42)	50.0	37.5	<b>76.9</b>	25.0	42.9	14.3	45.3	75.1	53.1
Local (Client 84)	50.0	37.5	46.2	25.0	28.6	0.0	34.4	72.7	47.7
Centralized	64.3 $\pm$ 4.8	37.5 $\pm$ 0.9	69.2 $\pm$ 6.1	<b>50.0</b> $\pm$ 2.2	42.9 $\pm$ 3.8	28.6 $\pm$ 0.4	51.6 $\pm$ 3.0	79.9 $\pm$ 4.7	57.8 $\pm$ 5.7
<b>FEDAGENT</b>	<b>80.0</b> $\pm$ 4.2	<b>75.0</b> $\pm$ 1.7	53.8 $\pm$ 4.3	37.5 $\pm$ 1.3	<b>83.3</b> $\pm$ 4.7	<b>50.0</b> $\pm$ 1.0	<b>64.1</b> $\pm$ 2.8	<b>83.2</b> $\pm$ 4.5	<b>61.7</b> $\pm$ 1.8
<i>Qwen2.5-3B-Instruct</i>									
Local (Client 21)	41.5	12.5	34.9	<b>51.0</b>	18.9	21.2	31.3	59.8	55.0
Local (Client 42)	46.5	37.5	24.4	15.0	33.7	33.3	28.2	61.3	59.3
Local (Client 84)	22.8	27.5	39.1	46.3	48.3	36.5	29.9	77.6	58.6
Centralized	94.1 $\pm$ 0.9	<b>80.0</b> $\pm$ 2.5	<b>64.3</b> $\pm$ 1.4	42.9 $\pm$ 2.6	50.0 $\pm$ 2.7	22.2 $\pm$ 5.2	62.5 $\pm$ 4.2	70.0 $\pm$ 1.5	53.9 $\pm$ 2.8
<b>FEDAGENT</b>	<b>95.5</b> $\pm$ 4.3	62.5 $\pm$ 3.0	49.7 $\pm$ 1.7	47.5 $\pm$ 2.4	<b>85.3</b> $\pm$ 3.6	<b>45.1</b> $\pm$ 2.1	<b>65.2</b> $\pm$ 3.9	<b>85.5</b> $\pm$ 3.4	<b>63.1</b> $\pm$ 3.1
<i>Qwen2.5-7B-Instruct</i>									
Local (Client 21)	35.5	25.0	61.0	25.9	35.8	<b>45.2</b>	38.4	70.9	49.2
Local (Client 42)	29.0	45.0	18.8	25.6	15.9	38.0	42.1	78.2	33.6
Local (Client 84)	34.7	47.5	44.4	51.3	40.1	21.8	35.7	60.6	39.3
Centralized	93.7 $\pm$ 4.5	82.5 $\pm$ 2.1	<b>71.5</b> $\pm$ 3.3	47.9 $\pm$ 3.7	63.2 $\pm$ 3.8	31.9 $\pm$ 1.0	73.3 $\pm$ 4.0	78.8 $\pm$ 2.8	64.7 $\pm$ 1.6
<b>FEDAGENT</b>	<b>94.5</b> $\pm$ 2.3	<b>85.0</b> $\pm$ 4.1	56.0 $\pm$ 0.8	<b>62.5</b> $\pm$ 1.2	<b>86.7</b> $\pm$ 2.9	42.8 $\pm$ 3.4	<b>75.5</b> $\pm$ 2.9	<b>89.0</b> $\pm$ 4.1	<b>68.9</b> $\pm$ 3.8
<i>Llama-3.2-3B-Instruct</i>									
Local (Client 21)	39.8	50.0	17.9	40.0	20.7	34.0	38.1	65.3	50.5
Local (Client 42)	18.2	55.0	41.9	34.3	41.0	25.0	35.0	67.0	51.0
Local (Client 84)	29.9	32.5	39.0	18.9	18.8	<b>37.6</b>	29.7	70.2	55.7
Centralized	72.4 $\pm$ 4.6	<b>62.5</b> $\pm$ 4.5	59.3 $\pm$ 3.1	45.2 $\pm$ 0.5	53.7 $\pm$ 2.2	27.9 $\pm$ 3.0	54.9 $\pm$ 2.9	72.3 $\pm$ 3.7	56.2 $\pm$ 1.6
<b>FEDAGENT</b>	<b>83.7</b> $\pm$ 1.7	57.5 $\pm$ 6.0	<b>60.6</b> $\pm$ 3.4	<b>55.9</b> $\pm$ 0.9	<b>65.3</b> $\pm$ 2.8	24.9 $\pm$ 3.1	<b>61.2</b> $\pm$ 3.3	<b>74.4</b> $\pm$ 4.9	<b>57.8</b> $\pm$ 3.2

Table 1: **Performance Comparison on ALFWorld and WebShop.** We report the averaged performance and the corresponding standard deviation for Centralized Training and FEDAGENT over three random seeds. For ALFWorld, the **Success Rate (%)** is reported for each subtask as well as for the overall dataset. For WebShop, both the **Task Score (%)** and the **Success Rate (%)** are reported.

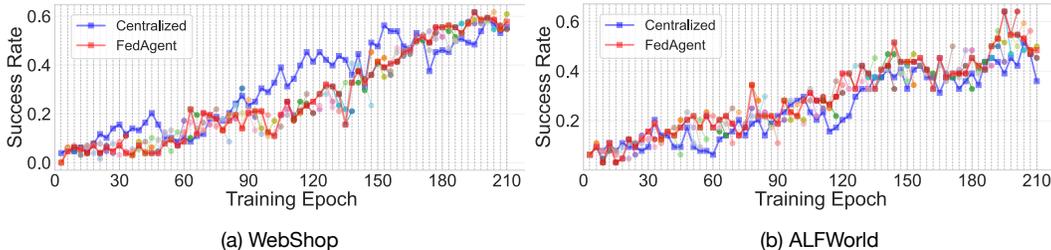


Figure 2: **Training Dynamics of FEDAGENT and Centralized Training.** Circle marks with different colors indicate the model performance after training on *specific selected clients each round*. The red line refers to the performance of the *aggregated models on server* throughout the training process.

Figure 2 shows the whole training dynamics of FEDAGENT and Centralized Agent Training with Qwen2.5-1.5B-Instruct on WebShop and ALFWorld datasets. **Both paradigms ultimately converge to similar success rates despite different training dynamics** ( $\sim 0.6$  for WebShop,  $\sim 0.5$  for ALFWorld). In WebShop (left), both approaches demonstrate steady monotonic improvement, with centralized training initially outperforming FEDAGENT until approximately epoch 120, after which both converge to similar success rates around 0.6. In contrast, ALFWorld (right) exhibits relatively more volatile training dynamics with frequent performance fluctuations for both methods, ultimately converging to success rates around 0.5. This further illustrates that FEDAGENT can achieve comparable performance with centralized training.

## 6 IMPACT OF DIFFERENT DECENTRALIZED SETTINGS

**Experiment Setup.** In this section, we aim to study the impact of different decentralized settings on FEDAGENT in FEDAGENTGYM by systematically varying three key hyperparameters across two

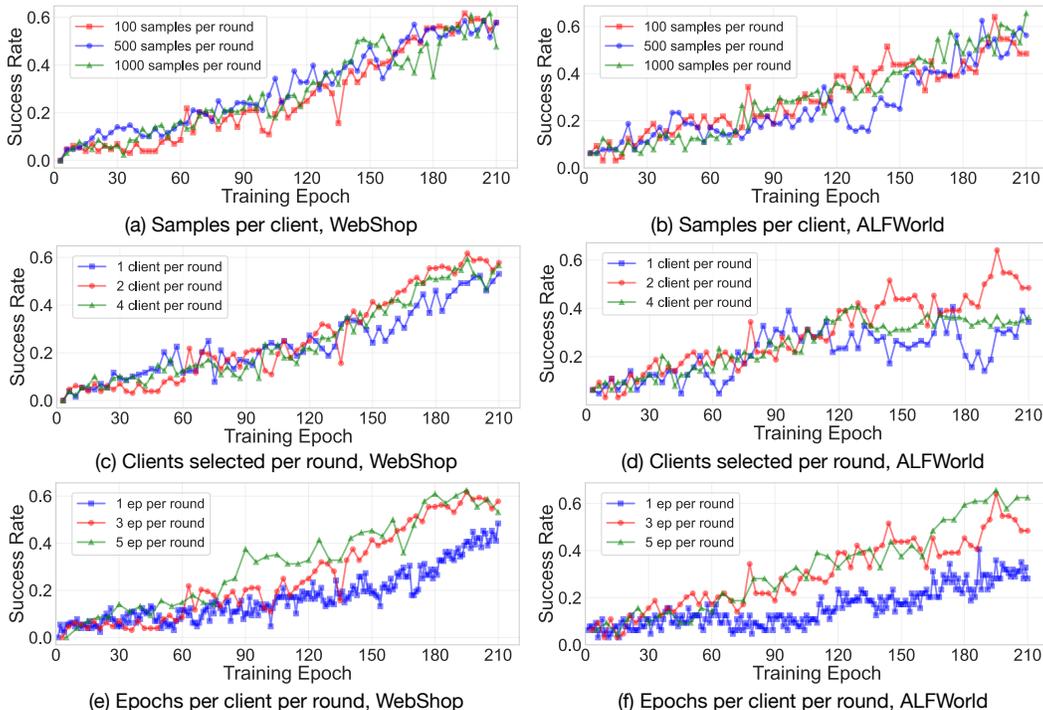


Figure 3: Training Dynamics of FEDAGENT in Different Decentralized Settings.

different datasets (WebShop and ALFWorld). We adopt Qwen2.5-1.5B-Instruct for all configurations. The experimental setup examines: (1) **samples per client**. We test 100, 500, and 1,000 tasks per client to understand how task sampling scope affects FEDAGENT learning dynamics; (2) **clients selected per round**. We compare 1, 2, and 4 participating clients each round to analyze the effect of federation scale on performance; and (3) **epochs per client per round**. We evaluate 1, 3, and 5 local training epochs to determine the optimal number of local computations before aggregation. Since we keep the total number of epochs the same at 210 for all configurations, 1, 3, and 5 local training epochs correspond to 210, 70, and 42 total rounds, respectively.

**Result Analysis.** The results in Figure 3 demonstrate that **FEDAGENT exhibits distinct sensitivity patterns towards decentralized settings, depending on the specific hyperparameter and dataset.** First, it shows notable sensitivity to the number of epochs per client per round. Moving from 1 to 5 epochs per round leads to significant performance gains, especially after around 100 training epochs, highlighting that shallow local updates are insufficient to unlock the full potential of FEDAGENT. On ALFWorld, FEDAGENT is also sensitive to the number of clients selected per round, with 2 clients per round outperforming 1 or 4, suggesting that too few or too many clients could hinder convergence. By contrast, FEDAGENT appears insensitive to the number of samples per client, as performance curves largely overlap across 100, 500, and 1,000 samples per round, suggesting that the task sampling scope for one client beyond a certain threshold may not be the limiting factor. Our studies offer valuable insights on the practical deployment of FEDAGENT and also suggest that **optimal federated agent learning configurations are environment-dependent.**

## 7 IMPACT OF HETEROGENEITY CHALLENGES

**Experiment Setup.** In this section, we aim to study the impact of different heterogeneity challenges on FEDAGENT in FEDAGENTGYM. As shown in Appendix B.2, we can leverage our proposed client partitioning strategies PREFERENCEPARTITION, COVERAGEPARTITION, and HARDNESSPARTITION to precisely control the extent of one form of heterogeneity (Preference, Coverage, or Hardness Heterogeneity) across clients with a hyperparameter  $\omega$ ,  $\xi$ , or  $\xi'$ . We keep the number of total epochs as 210 and the number of all clients as 100, which are consistent with the main experiments. We adopt Qwen2.5-1.5B-Instruct in the experiments.

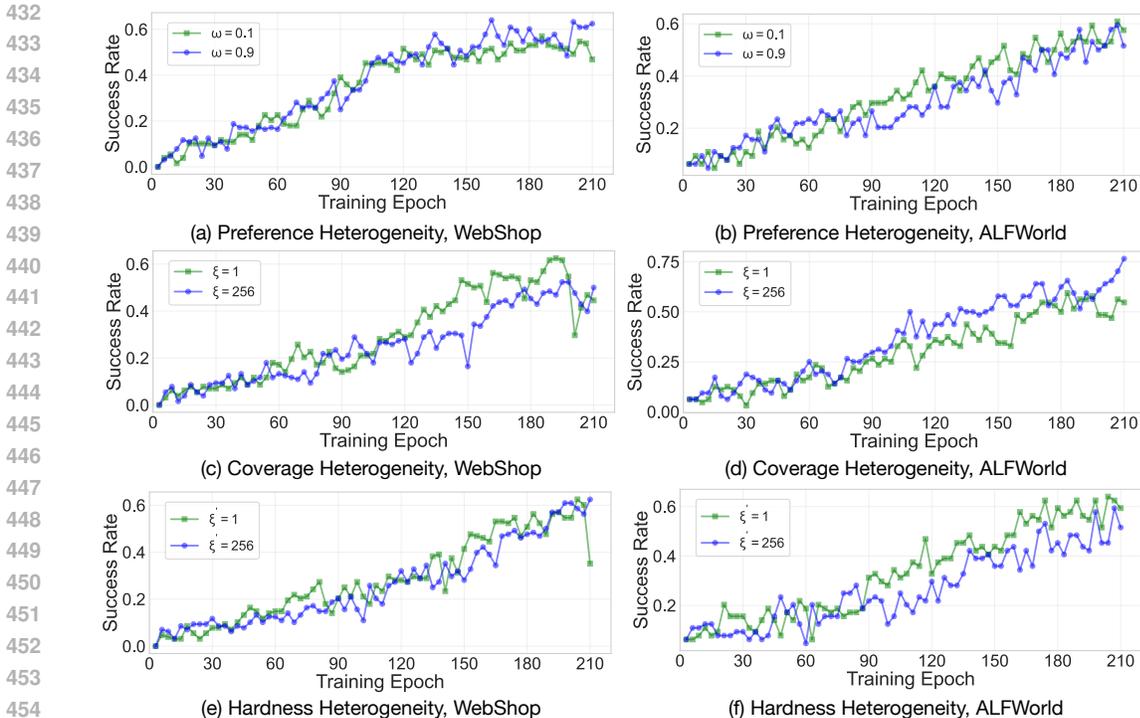


Figure 4: Training Dynamics of FEDAGENT in Different Heterogeneity Challenges.

**Result Analysis.** As shown in Figure 4, FEDAGENT shows high robustness against the three heterogeneity challenges. Across all scenarios, preference heterogeneity (panels a,b), coverage heterogeneity (panels c,d), and hardness heterogeneity (panels e,f), even when comparing low heterogeneity settings ( $\omega = 0.1$ ,  $\xi = 256$ ,  $\xi' = 256$ ) against high heterogeneity settings ( $\omega = 0.9$ ,  $\xi = 1$ ,  $\xi' = 1$ ), FEDAGENT consistently achieves strong success rates that steadily improve throughout training. The learning curves show that FEDAGENT maintains stable convergence behavior in both WebShop and ALFWorld environments regardless of heterogeneity intensity, with success rates generally reaching 0.5-0.6 by the end of training. Crucially, the performance degradation is minimal even under extreme heterogeneity conditions, indicating that FEDAGENT has great potential to handle real-world scenarios across the full spectrum of heterogeneity challenges.

## 8 RELATED WORK

RL has been instrumental in empowering LLM agents to function effectively in dynamic and open-ended environments. Initial studies leveraged traditional RL approaches like DQN (Mnih et al., 2015) for training LLM agents in text-based gaming environments (Narasimhan et al., 2015). Subsequent research began incorporating value-based techniques across broader agent applications such as Android device manipulation (Rawles et al., 2023) and embodied environments like ALFWorld (Shridhar et al., 2020). Contemporary methods have expanded RL training to encompass sophisticated web-based and application-specific tasks (Zhou et al., 2024; Putta et al., 2024). In previous works, real-world task queries and trajectories have been essential for training AI agents in practical applications. However, they are becoming increasingly difficult to acquire due to privacy concerns. Our work makes an initial effort to explore training AI agents without compromising user data privacy.

## 9 CONCLUSION

In this work, we explored FEDAGENT (Federated Agent Reinforcement Learning), a new collaborative paradigm to train AI agent, particularly LLMs, across distributed clients, and built FEDAGENTGYM, the first decentralized agent learning environment. Extensive theoretical and empirical studies demonstrate that FEDAGENT can achieve performance on par with centralized training and maintain strong robustness to heterogeneities. Our work validates the feasibility of training AI agents while protecting user data privacy and charts new research directions in agent learning.

486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539

## ETHICS STATEMENT

This research on federated agent reinforcement learning aims to address the critical privacy concerns in AI agent training by developing decentralized paradigms that eliminate centralized data collection, ensuring all user data remains distributed across local clients throughout the training process.

## REPRODUCIBILITY STATEMENT

The link to the code repository is available [here](#). The proof of Theorem 1 is in Appendix C.

## REFERENCES

- 540  
541  
542 Jalaj Bhandari and Daniel Russo. Global optimality guarantees for policy gradient methods. *Operations Research*, 72(5):1906–1927, 2024.
- 543  
544 William G. Cochran. *Sampling Techniques*. John Wiley & Sons, New York, 3rd edition, 1977. ISBN  
545 0-471-16240-X. See § 2.6.
- 546  
547 DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu,  
548 Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu,  
549 Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao  
550 Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan,  
551 Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao,  
552 Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding,  
553 Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang  
554 Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong,  
555 Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao,  
556 Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang,  
557 Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang,  
558 Qinyu Chen, Qiusi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L.  
559 Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang,  
560 Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng  
561 Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanbiao Zhao, Wen Liu, Wenfeng  
562 Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan  
563 Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang,  
564 Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen,  
565 Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li,  
566 Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang,  
567 Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan,  
568 Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia  
569 He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong  
570 Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha,  
571 Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang,  
572 Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li,  
573 Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen  
574 Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv: 2501.12948*, 2025.
- 574 Flint Xiaofeng Fan, Cheston Tan, Yew-Soon Ong, Roger Wattenhofer, and Wei-Tsang Ooi. Fedrlhf:  
575 A convergence-guaranteed federated framework for privacy-preserving and personalized rlhf. In  
576 *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*,  
577 AAMAS’25, pp. 713–721, Richland, SC, 2025. International Foundation for Autonomous Agents  
578 and Multiagent Systems. ISBN 9798400714269.
- 579 Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm  
580 agent training. *arXiv preprint arXiv:2505.10978*, 2025.
- 581  
582 Dashan Gao, Xin Yao, and Qiang Yang. A survey on heterogeneous federated learning. *arXiv preprint*  
583 *arXiv:2210.04505*, 2022.
- 584 Huan-ang Gao, Jiayi Geng, Wenyue Hua, Mengkang Hu, Xinzhe Juan, Hongzhang Liu, Shilong Liu,  
585 Jiahao Qiu, Xuan Qi, Yiran Wu, et al. A survey of self-evolving agents: On path to artificial super  
586 intelligence. *arXiv preprint arXiv:2507.21046*, 2025.
- 587 Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin  
588 Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Ad-  
589 vances and open problems in federated learning. *Foundations and trends® in machine learning*,  
590 14(1–2):1–210, 2021.
- 591  
592 Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-  
593 gradient methods under the polyak-łojasiewicz condition. In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 795–811. Springer, 2016.

- 594 Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and  
595 Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In  
596 *International conference on machine learning*, pp. 5132–5143. PMLR, 2020.
- 597  
598 Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. Tighter theory for local sgd on identical  
599 and heterogeneous data. In *International conference on artificial intelligence and statistics*, pp.  
600 4519–4529. PMLR, 2020.
- 601  
602 Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith.  
603 Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*,  
604 2:429–450, 2020.
- 605  
606 Bang Liu, Xinfeng Li, Jiayi Zhang, Jinlin Wang, Tanjin He, Sirui Hong, Hongzhang Liu, Shaokun  
607 Zhang, Kaitao Song, Kunlun Zhu, et al. Advances and challenges in foundation agents: From  
608 brain-inspired intelligence to evolutionary, collaborative, and safe systems. *arXiv preprint*  
*arXiv:2504.01990*, 2025.
- 609  
610 Bingyan Liu, Nuoyan Lv, Yuanchun Guo, and Yawen Li. Recent advances on federated learning: A  
611 systematic survey. *Neurocomputing*, 597:128019, 2024.
- 612  
613 Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare,  
614 Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control  
615 through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.
- 616  
617 Karthik Narasimhan, Tejas Kulkarni, and Regina Barzilay. Language understanding for text-based  
618 games using deep reinforcement learning. *arXiv preprint arXiv:1506.08941*, 2015.
- 619  
620 Pranav Putta, Edmund Mills, Naman Garg, Sumeet Motwani, Chelsea Finn, Divyansh Garg, and  
621 Rafael Rafailov. Agent q: Advanced reasoning and learning for autonomous ai agents. *arXiv*  
*preprint arXiv:2408.07199*, 2024.
- 622  
623 Jiaju Qi, Qihao Zhou, Lei Lei, and Kan Zheng. Federated reinforcement learning: Techniques,  
624 applications, and open challenges. *arXiv preprint arXiv:2108.11887*, 2021.
- 625  
626 Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy Lillicrap. An-  
627 droidinthewild: A large-scale dataset for android device control. *Advances in Neural Information*  
*Processing Systems*, 36:59708–59728, 2023.
- 628  
629 Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang,  
630 Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathemat-  
631 ical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.
- 632  
633 Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew  
634 Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv*  
*preprint arXiv:2010.03768*, 2020.
- 635  
636 Sebastian U Stich. Local sgd converges fast and communicates little. *arXiv preprint arXiv:1805.09767*,  
637 2018.
- 638  
639 Blake E Woodworth, Kumar Kshitij Patel, and Nati Srebro. Minibatch vs local sgd for heterogeneous  
640 distributed learning. *Advances in Neural Information Processing Systems*, 33:6281–6292, 2020.
- 641  
642 Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable  
643 real-world web interaction with grounded language agents. *Advances in Neural Information*  
*Processing Systems*, 35:20744–20757, 2022.
- 644  
645 Mang Ye, Xiuwen Fang, Bo Du, Pong C Yuen, and Dacheng Tao. Heterogeneous federated learning:  
646 State-of-the-art and research challenges. *ACM Computing Surveys*, 56(3):1–44, 2023.
- 647  
648 Rui Yuan, Robert M Gower, and Alessandro Lazaric. A general sample complexity analysis of  
649 vanilla policy gradient. In *International Conference on Artificial Intelligence and Statistics*, pp.  
650 3332–3380. PMLR, 2022.

648 Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou,  
649 Zhongzhi Li, Xiangyuan Xue, Yijiang Li, et al. The landscape of agentic reinforcement learning  
650 for llms: A survey. *arXiv preprint arXiv:2509.02547*, 2025.  
651  
652 Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. Archer: Training language  
653 model agents via hierarchical multi-turn rl. *arXiv preprint arXiv:2402.19446*, 2024.  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701

702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755

# Content of Appendix

- A LLM Usage Disclosure** **15**
  
- B More Details of Heterogeneity Challenges** **16**
  - B.1 Pseudo Code for Client Partitioning Strategies . . . . . 16
  - B.2 Client Distributions under Partitioning Strategies . . . . . 17
    - B.2.1 Preference Heterogeneity . . . . . 17
    - B.2.2 Coverage Heterogeneity . . . . . 19
    - B.2.3 Hardness Heterogeneity . . . . . 21
  
- C Proof of the Convergence** **23**
  - C.1 Proof Sketch . . . . . 23
  - C.2 A More Detailed Proof . . . . . 24

756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809

## A LLM USAGE DISCLOSURE

We hereby disclose that Large Language Models (LLMs) are utilized solely for the purposes of grammar correction and textual refinement.

## B MORE DETAILS OF HETEROGENEITY CHALLENGES

### B.1 PSEUDO CODE FOR CLIENT PARTITIONING STRATEGIES

---

#### Algorithm 2 PREFERENCEPARTITION

---

**Require:** Category pools  $\{\mathcal{I}_c\}_{c=1}^C$  with sizes  $n_c$ ; total clients  $K$ ; per-client set size  $L$ ; jitter  $\omega$   
**Ensure:** Client datasets  $X_1, \dots, X_K$  with  $|X_k| = L$

- 1:  $p_c \leftarrow n_c / \sum_{j=1}^C n_j$ ;  $\ell_c \leftarrow \log \frac{p_c}{1-p_c}$  ▷ global mix + logit anchors
- 2: **for**  $k = 1$  to  $K$  **do**
- 3:  $z_c \sim \mathcal{N}(\ell_c, \omega^2)$  for  $c = 1 \dots C$ ;  $q_c \leftarrow \exp(z_c) / \sum_j \exp(z_j)$  ▷ larger  $\omega \Rightarrow$  higher variance
- 4:  $(a_1, \dots, a_C) \sim \text{Multinomial}(L; q_1, \dots, q_C)$  ▷ category counts for client  $k$
- 5: **if** any  $a_c > n_c$  **then** set  $a_c \leftarrow \min(a_c, n_c)$  and redistribute leftover by  $q$  to classes with spare capacity ▷ capacity fix within a set
- 6:  $X_k \leftarrow \bigcup_{c=1}^C \text{SAMPLEWITHOUTREPLACEMENT}(\mathcal{I}_c, a_c)$
- 7: **end for**
- 8: **return**  $\{X_k\}_{k=1}^K$

---



---

#### Algorithm 3 COVERAGEPARTITION

---

**Require:** total items  $N$  (indexed  $1:N$ ); total clients  $K$ ; per-client bounds  $(L_{\min}, L_{\text{avg}}, L_{\max})$  with  $L_{\min} \leq L_{\text{avg}} \leq L_{\max}$ ; dispersion  $\xi$ ; desired average replicas per item  $r$   
**Ensure:** Client datasets  $X_1, \dots, X_K$

- 1:  $T \leftarrow \lfloor rN \rfloor$  ▷ total assignments (sum of all  $|X_k|$ ); keeps global overlap fixed
- 2: **assert**  $KL_{\min} \leq T \leq KL_{\max}$  ▷ feasibility under per-client bounds
- 3:  $\mu \leftarrow (L_{\text{avg}} - L_{\min}) / (L_{\max} - L_{\min})$ ;  $\alpha \leftarrow \mu\xi$ ,  $\beta \leftarrow (1 - \mu)\xi$  ▷ Beta params with mean fixed at  $L_{\text{avg}}$
- 4: **Sample**  $x_k \sim \text{Beta}(\alpha, \beta)$  for  $k = 1 \dots K$  ▷ larger  $\xi \Rightarrow$  lower variance (sizes closer to  $L_{\text{avg}}$ )
- 5:  $u_k \leftarrow L_{\min} + x_k(L_{\max} - L_{\min})$ ;  $u_k \leftarrow u_k \cdot \frac{T}{\sum_j u_j}$  ▷ shape then renormalize to sum  $T$
- 6:  $n_k \leftarrow \text{ROUNDTOSUM}(u, T, [L_{\min}, L_{\max}])$  ▷ largest remainder with clipping to  $[L_{\min}, L_{\max}]$
- 7:  $m \leftarrow \lfloor r \rfloor$ ,  $M \leftarrow \lceil r \rceil$ ,  $H \leftarrow T - mN$
- 8: **Set**  $q_i \leftarrow M$  for any  $H$  items;  $q_i \leftarrow m$  otherwise
- 9: **Initialize**  $X_k \leftarrow \emptyset$ ,  $\text{rem}_k \leftarrow n_k$  for all  $k$
- 10: **for**  $i = 1$  to  $N$  **do** ▷ weighted, no-replacement placement across clients
- 11:  $\mathcal{A} \leftarrow \{k : \text{rem}_k > 0\}$ ; choose  $q_i$  distinct  $k \in \mathcal{A}$  with  $\text{Pr}(k) \propto \text{rem}_k$
- 12: **Add** item  $i$  to each chosen  $X_k$  and decrement the corresponding  $\text{rem}_k$
- 13: **end for**
- 14: **return**  $\{X_k\}_{k=1}^K$

---



---

#### Algorithm 4 HARDNESSPARTITION

---

**Require:** total items  $N$  (indexed  $1:N$ ); disjoint index sets  $\mathcal{S}$  (successful) and  $\mathcal{U}$  (unsuccessful) with  $\mathcal{S} \cup \mathcal{U} = \{1:N\}$ ; total clients  $K$ ; per-client set size  $L$ ; Hyperparameters for COVERAGEPARTITION: bounds  $(\ell, c, h)$  with  $h \leq L$ , dispersion  $\xi'$ , overlap  $r$   
**Ensure:** client datasets  $X_1, \dots, X_K$  with  $|X_k| = L$

- 1:  $\{Y_k\}_{k=1}^K \leftarrow \text{COVERAGEPARTITION}(|\mathcal{S}|, K, (\ell, c, h), \xi', r)$  ▷ larger  $\xi' \Rightarrow$  lower variance
- 2: **for**  $k = 1$  to  $K$  **do**
- 3:  $m_k \leftarrow L - |Y_k|$ ;  $F_k \leftarrow \text{SAMPLEWITHOUTREPLACEMENT}(\mathcal{U}, m_k)$
- 4:  $X_k \leftarrow Y_k \cup F_k$
- 5: **end for**
- 6: **return**  $\{X_k\}_{k=1}^K$

---

B.2 CLIENT DISTRIBUTIONS UNDER PARTITIONING STRATEGIES

B.2.1 PREFERENCE HETEROGENEITY

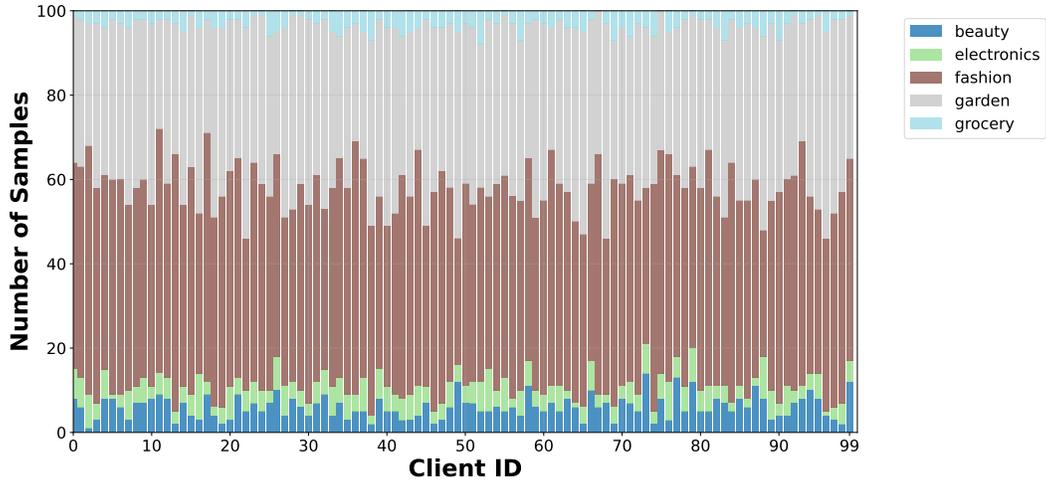


Figure 5: Client Distribution under Preference Heterogeneity (WebShop,  $\omega = 0.1$ ).

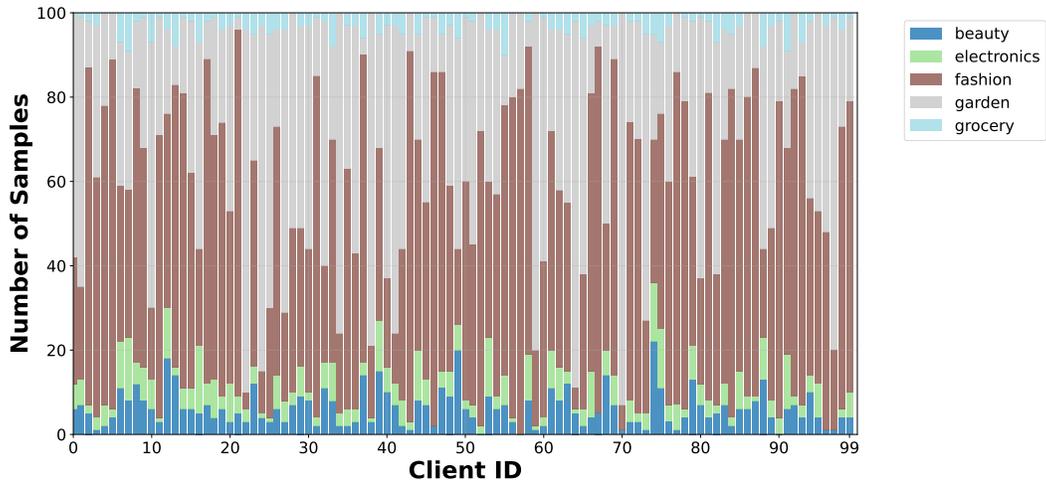


Figure 6: Client Distribution under Preference Heterogeneity (WebShop,  $\omega = 0.9$ ).

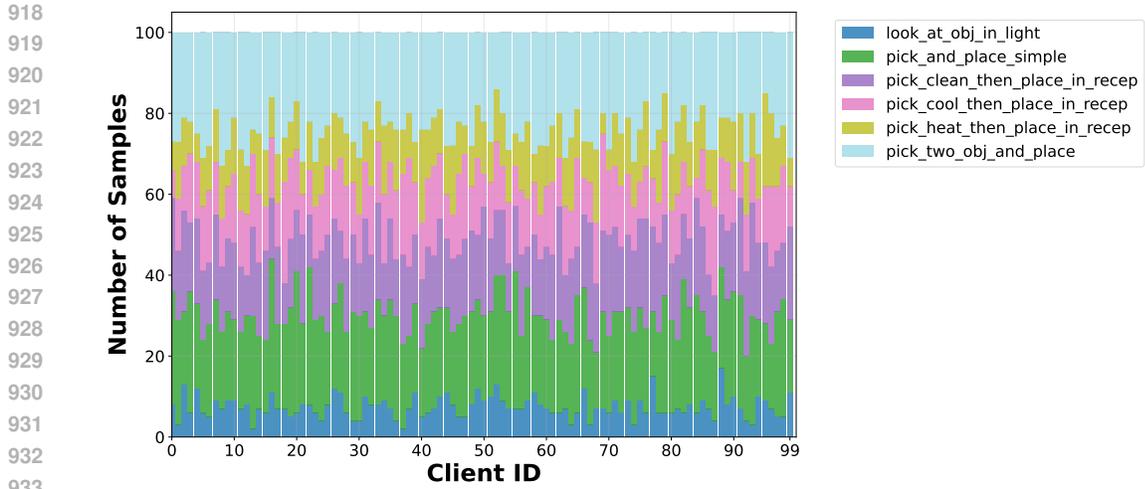


Figure 7: Client Distribution under Preference Heterogeneity (ALFWorld,  $\omega = 0.1$ ).

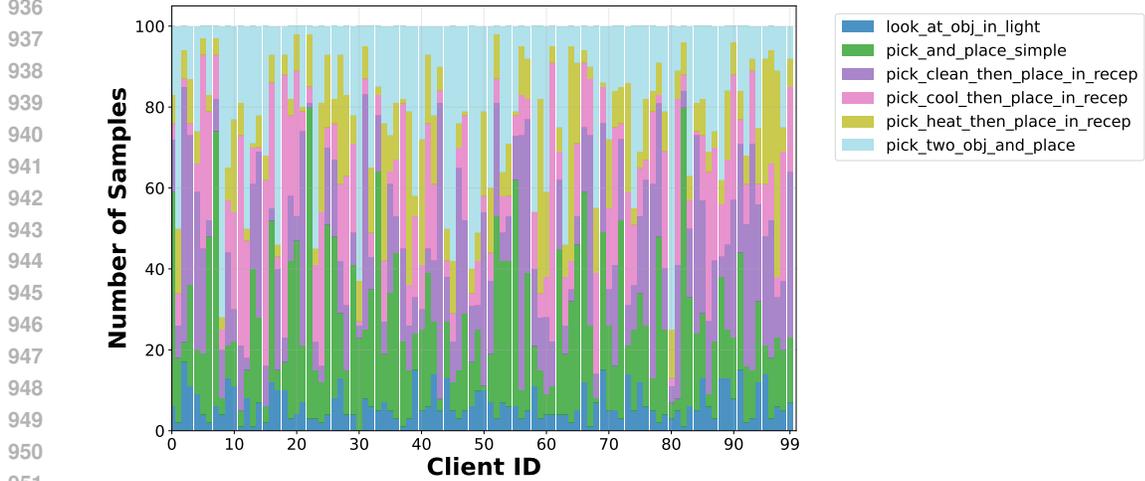


Figure 8: Client Distribution under Preference Heterogeneity (ALFWorld,  $\omega = 0.9$ ).

952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971

B.2.2 COVERAGE HETEROGENEITY

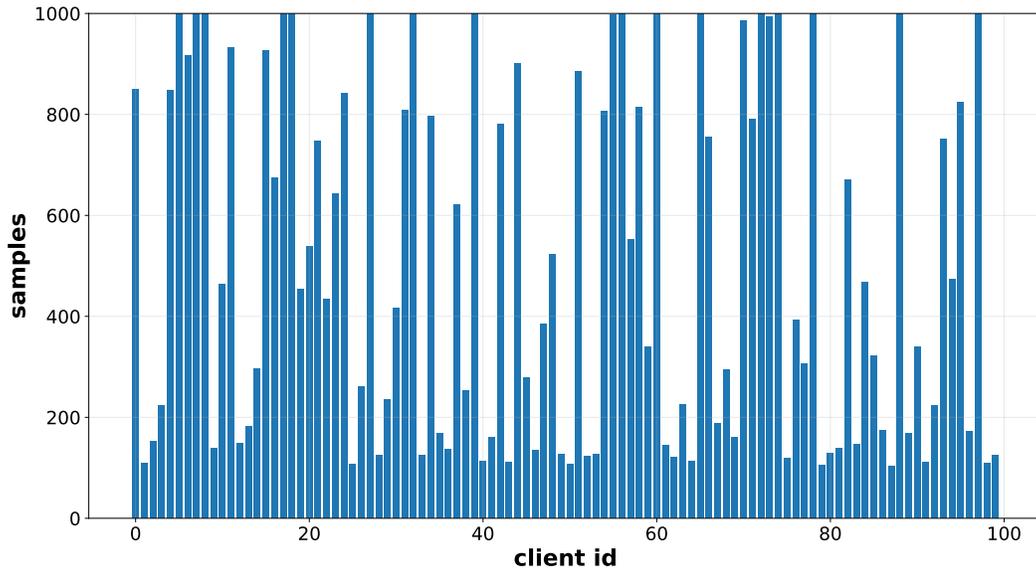


Figure 9: Client Distribution under Coverage Heterogeneity (WebShop,  $\xi = 1$ ).

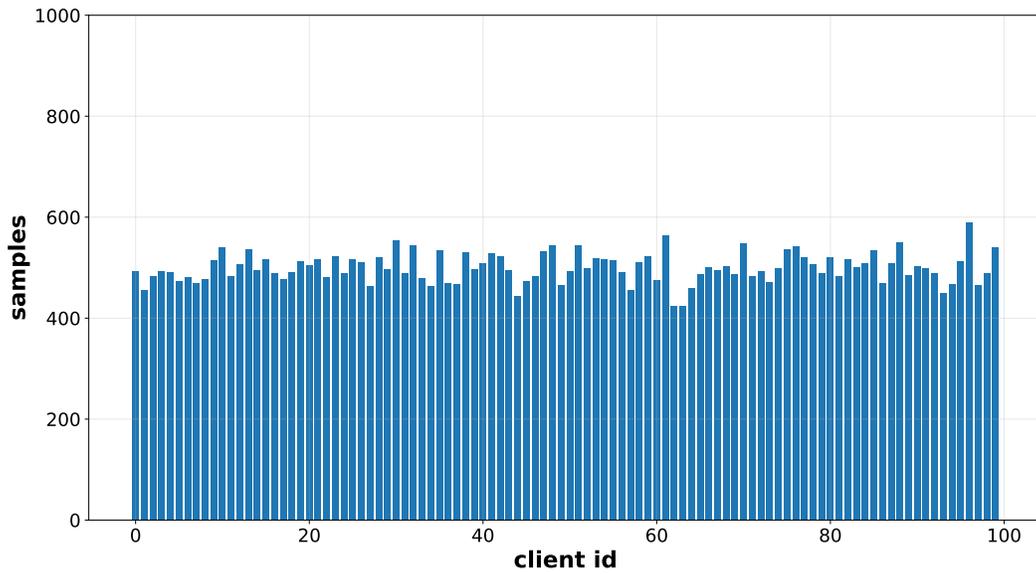


Figure 10: Client Distribution under Coverage Heterogeneity (WebShop,  $\xi = 256$ ).

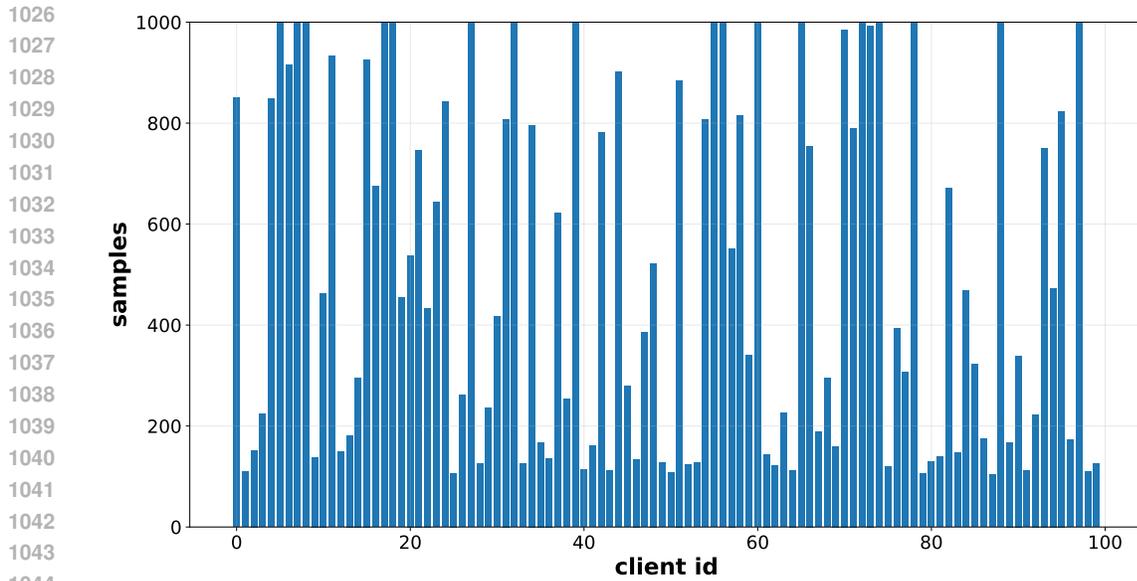


Figure 11: Client Distribution under Coverage Heterogeneity (ALFWorld,  $\xi = 1$ ).

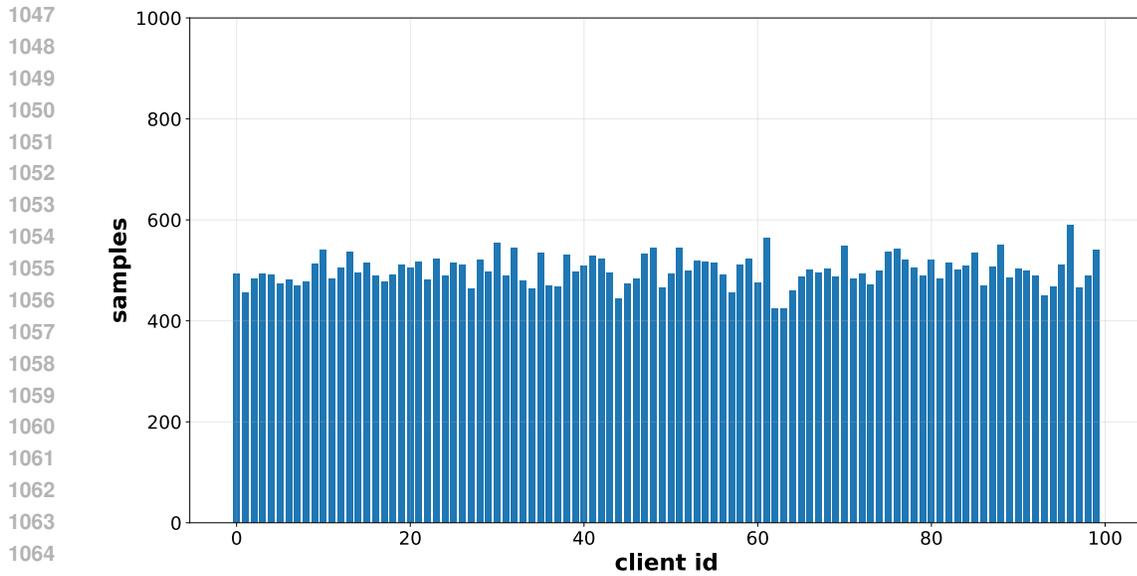


Figure 12: Client Distribution under Coverage Heterogeneity (ALFWorld,  $\xi = 256$ ).

B.2.3 HARDNESS HETEROGENEITY

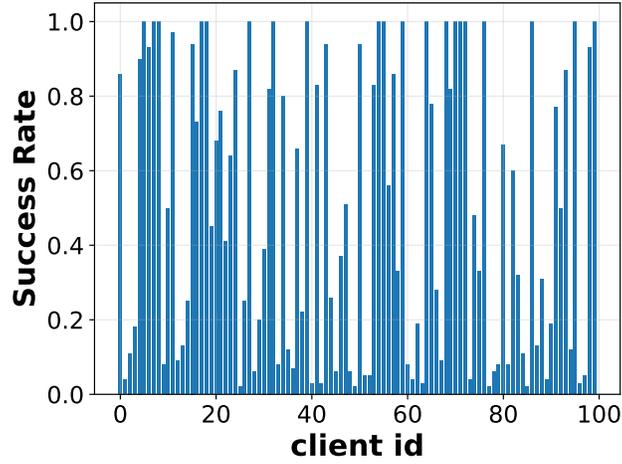


Figure 13: Client Distribution under Coverage Heterogeneity (WebShop,  $\xi' = 1$ ).

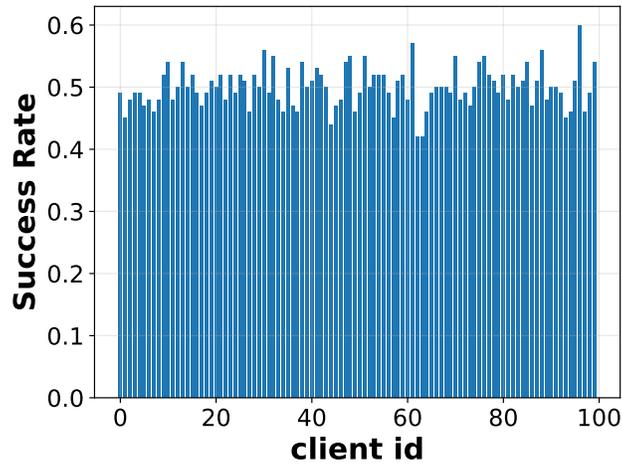


Figure 14: Client Distribution under Coverage Heterogeneity (WebShop,  $\xi' = 256$ ).

1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187

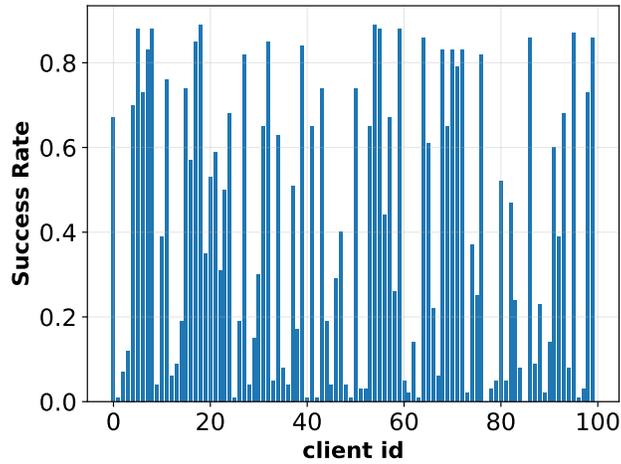


Figure 15: Client Distribution under Coverage Heterogeneity (ALFWorld,  $\xi' = 1$ ).

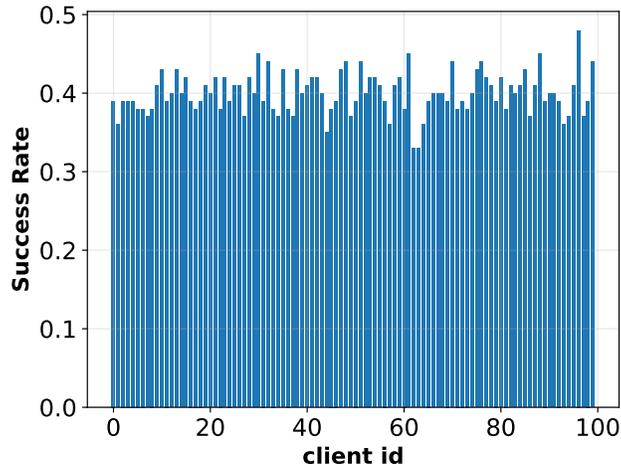


Figure 16: Client Distribution under Coverage Heterogeneity (ALFWorld,  $\xi' = 256$ ).

## C PROOF OF THE CONVERGENCE

**Assumption 1** ( $L$ -smooth gradients). For all  $\theta, \theta' \in \mathbb{R}^d$  and  $k \in [K]$ , the client objectives have  $L$ -Lipschitz gradients:

$$\|\nabla J_k(\theta) - \nabla J_k(\theta')\| \leq L\|\theta - \theta'\|.$$

**Assumption 2** ( $G$ -bounded gradients). For all  $\theta \in \mathbb{R}^d$  and  $k \in [K]$ , the full gradients are bounded:

$$\|\nabla J_k(\theta)\| \leq G.$$

**Assumption 3** ( $\sigma$ -bounded variance). For all  $\theta \in \mathbb{R}^d$  and  $k \in [K]$ , the stochastic gradient estimator has bounded variance:

$$\mathbb{E}[\|\nabla J_k(\theta; B) - \nabla J_k(\theta)\|^2] \leq \sigma^2,$$

where  $\nabla J_k(\theta; B)$  denotes the mini-batch gradient.

**Assumption 4** (Polyak–Łojasiewicz (PL) condition). The global objective satisfies, for some  $\mu > 0$  and  $\theta^* = \arg \max_{\theta} J(\theta)$ ,

$$2\mu(J(\theta^*) - J(\theta)) \leq \|\nabla J(\theta)\|^2, \quad \forall \theta \in \mathbb{R}^d.$$

**Assumption 5** (Bounded client heterogeneity). There exists  $\zeta^2$  such that for all  $\theta$ ,

$$\frac{1}{K} \sum_{k=0}^{K-1} \|\nabla J_k(\theta) - \nabla J(\theta)\|^2 \leq \zeta^2, \quad \text{where } \nabla J(\theta) = \frac{1}{K} \sum_{k=0}^{K-1} \nabla J_k(\theta).$$

**Remark.** Assumptions 1-3 are standard in stochastic optimization literature. As for Assumption 4 (the PL condition), in practice, policy-gradient methods that constrain update size, such as trust-region approaches or proximal policy methods, yield smoother policy updates, making the PL assumption more tenable. Recent works have likewise employed PL-type conditions to obtain convergence guarantees for non-convex reinforcement learning objectives (Bhandari & Russo, 2024; Karimi et al., 2016; Yuan et al., 2022), supporting our adoption of this assumption. Assumption 5 is a common “bounded heterogeneity” condition used to control client drift in federated learning analyses (Li et al., 2020; Karimireddy et al., 2020; Stich, 2018; Khaled et al., 2020; Woodworth et al., 2020).

**Theorem 1** (Convergence of FEDAGENT). Under Assumptions 1–5, suppose that at each communication round  $t$  the server uniformly samples without replacement a subset  $S_t \subset [K]$  of size  $M \leq K$  and aggregates only those clients’ updates:  $\theta_{t+1} = \theta_t + \frac{1}{M} \sum_{k \in S_t} \Delta \theta_{k,t}$ , with the same local inner loop and notation as in alg:fedagent. Let each selected client perform  $\tau$  local steps with stepsize  $\eta$ . Choose the stepsize  $\eta = \frac{1}{L\tau}$  and let  $\tilde{\theta}$  be a uniform random iterate drawn from  $\{\theta_t\}_{t=0}^{T-1}$ . Then

$$\mathbb{E}\left[J(\theta^*) - J(\tilde{\theta})\right] \leq \frac{L}{\mu T} (J(\theta^*) - J(\theta_0)) + \frac{1}{2\mu} \left[ \frac{G^2 + \sigma^2}{M} + \frac{2(K-M)}{M(K-1)} \zeta^2 + \frac{(\tau-1)^2}{2\tau^2} (G^2 + \sigma^2) \right].$$

In particular, the convergence rate remains  $O(1/T)$ . The “noise floor” comprises: (i) an  $O(1/M)$  local stochastic term, (ii) an  $O(\frac{K-M}{M(K-1)})$  heterogeneity term due to client sampling, and (iii) an  $O(\frac{(\tau-1)^2}{\tau^2})$  local-drift term that vanishes when  $\tau = 1$ .

### C.1 PROOF SKETCH

*Proof sketch.* Our proof generally follows the proof of Theorem 4.1 in (Fan et al., 2025), with key modifications on second-moment bound for the aggregated update and local-drift term.

Let  $u_{k,t} := \sum_{i=0}^{\tau-1} g_{k,t,i}$  be client  $k$ ’s aggregate local stochastic gradients in round  $t$ , and  $\bar{g}_{k,t} := \frac{1}{\tau} \sum_{i=0}^{\tau-1} g_{k,t,i}$ . Define the round-average  $\bar{g}_t := \frac{1}{M} \sum_{k \in S_t} \bar{g}_{k,t}$ , so the server update is  $\Delta_t = \theta_{t+1} - \theta_t = \eta\tau \bar{g}_t$ .

**(1) One-step descent.** By  $L$ -smoothness of  $J$ ,

$$\mathbb{E}[J(\theta_{t+1}) \mid \theta_t] \geq J(\theta_t) + \eta\tau \left(1 - \frac{L\eta\tau}{2}\right) \|\nabla J(\theta_t)\|^2 - \frac{L}{2} \eta^2 \tau^2 \mathbb{E}\|\bar{g}_t - \nabla J(\theta_t)\|^2 \quad (4)$$

1242 **(2) Variance–bias decomposition with Finite-Population Correction (FPC).** Decompose

$$1243 \bar{g}_t - \nabla J(\theta_t) = \underbrace{\left( \frac{1}{M} \sum_{k \in S_t} \nabla J_k(\theta_t) - \nabla J(\theta_t) \right)}_{\text{client sampling}} + \underbrace{\frac{1}{M} \sum_{k \in S_t} (\bar{g}_{k,t} - \mathbb{E} \bar{g}_{k,t})}_{\text{local stochastic noise}} + \underbrace{\frac{1}{M} \sum_{k \in S_t} b_{k,t}}_{\text{local drift}},$$

1244 where  $b_{k,t} := \mathbb{E}[\bar{g}_{k,t} | \theta_t] - \nabla J_k(\theta_t)$ . The three terms are bounded as follows:

$$1245 \mathbb{E} \left\| \frac{1}{M} \sum_{k \in S_t} \nabla J_k(\theta_t) - \nabla J(\theta_t) \right\|^2 = \frac{(K-M)}{M(K-1)} \cdot \frac{1}{K} \sum_{k=1}^K \|\nabla J_k(\theta_t) - \nabla J(\theta_t)\|^2$$

$$1246 \leq \frac{2(K-M)}{M(K-1)} \zeta^2, \quad (\text{FPC})$$

$$1247 \mathbb{E} \left\| \frac{1}{M} \sum_{k \in S_t} (\bar{g}_{k,t} - \mathbb{E} \bar{g}_{k,t}) \right\|^2 \leq \frac{G^2 + \sigma^2}{M}, \quad (\text{local noise})$$

$$1248 b_{k,t} = \frac{1}{\tau} \sum_{i=0}^{\tau-1} (\nabla J_k(\theta_{k,i}) - \nabla J_k(\theta_t)), \quad \mathbb{E} \|b_{k,t}\|^2 \leq \frac{L^2 \eta^2 (\tau-1)^2}{2} (G^2 + \sigma^2),$$

$$1249 \Rightarrow \mathbb{E} \left\| \frac{1}{M} \sum_{k \in S_t} b_{k,t} \right\|^2 \leq \frac{L^2 \eta^2 (\tau-1)^2}{2} (G^2 + \sigma^2). \quad (\text{local drift})$$

1250 Combining,

$$1251 \mathbb{E} \|\bar{g}_t - \nabla J(\theta_t)\|^2 \leq \frac{G^2 + \sigma^2}{M} + \frac{2(K-M)}{M(K-1)} \zeta^2 + \frac{L^2 \eta^2 (\tau-1)^2}{2} (G^2 + \sigma^2). \quad (5)$$

1252 **(3) PL and averaging.** Let  $\delta_t := \mathbb{E}[J(\theta^*) - J(\theta_t)]$ . Applying the PL condition  $\|\nabla J(\theta_t)\|^2 \geq 2\mu \delta_t$  in Equation (4) yields the linear recursion

$$1253 \delta_{t+1} \leq \left( 1 - 2\mu\eta\tau \left( 1 - \frac{L\eta\tau}{2} \right) \right) \delta_t + \frac{L}{2} \eta^2 \tau^2 \left( \frac{G^2 + \sigma^2}{M} + \frac{2(K-M)}{M(K-1)} \zeta^2 + \frac{L^2 \eta^2 (\tau-1)^2}{2} (G^2 + \sigma^2) \right),$$

1254 where, in forming Equation (5), we control the sampling-drift mixed term via Young's inequality  $2\langle X, Y \rangle \leq \|X\|^2 + \|Y\|^2$  (thus inflating the sampling and drift pieces by a factor of 2). With  $\eta = \frac{1}{L\tau}$

1255 the contraction becomes  $1 - \mu/L$ , and the drift contribution simplifies to  $\frac{(\tau-1)^2}{2\tau^2} (G^2 + \sigma^2)$ . Unrolling the recursion and averaging the gaps gives

$$1256 \frac{1}{T} \sum_{t=0}^{T-1} \delta_t \leq \frac{L}{\mu T} \delta_0 + \frac{1}{2\mu} \left( \frac{G^2 + \sigma^2}{M} + \frac{2(K-M)}{M(K-1)} \zeta^2 + \frac{(\tau-1)^2}{2\tau^2} (G^2 + \sigma^2) \right).$$

1257 Finally, let  $\tilde{\theta}$  be drawn uniformly at random from  $\{\theta_t\}_{t=0}^{T-1}$ ; then  $\mathbb{E}[J(\theta^*) - J(\tilde{\theta})] = \frac{1}{T} \sum_{t=0}^{T-1} \delta_t$ , which yields the stated bound.

1258  $\square$

## 1259 C.2 A MORE DETAILED PROOF

1260 *Proof.* We give a more detailed proof as follows. Our proof generally follows the proof of Theorem 4.1 in (Fan et al., 2025). Throughout, write  $\mathbb{E}_t[\cdot] := \mathbb{E}[\cdot | \theta_t]$ . Let each selected client  $k \in S_t$  perform  $\tau$  local stochastic policy-gradient steps with per-step gradients  $g_{k,t,i}$ ,  $i = 0, \dots, \tau-1$ ,

$$1261 \theta_{k,t,0} = \theta_t, \quad \theta_{k,t,i+1} = \theta_{k,t,i} + \eta g_{k,t,i}, \quad \mathbb{E}[g_{k,t,i} | \theta_{k,t,i}] = \nabla J_k(\theta_{k,t,i}), \quad \mathbb{E}\|g_{k,t,i}\|^2 \leq G^2 + \sigma^2.$$

1262 Define the client's round aggregates  $u_{k,t} := \sum_{i=0}^{\tau-1} g_{k,t,i}$  and  $\bar{g}_{k,t} := \frac{1}{\tau} u_{k,t}$ , and the server's round average

$$1263 \bar{g}_t := \frac{1}{M} \sum_{k \in S_t} \bar{g}_{k,t}, \quad \Delta_t := \theta_{t+1} - \theta_t = \eta\tau \bar{g}_t.$$

1264 We analyze  $J(\theta_{t+1})$  by Assumption 1  $L$ -smoothness of  $J$ :

$$1265 J(\theta_{t+1}) \geq J(\theta_t) + \langle \nabla J(\theta_t), \Delta_t \rangle - \frac{L}{2} \|\Delta_t\|^2 \quad (6)$$

**A. One-step progress.** Let  $g_t^* := \nabla J(\theta_t)$  and  $e_t := \bar{g}_t - g_t^*$ . With  $\eta = \frac{1}{L\tau}$  we have  $\Delta_t = \frac{1}{L}(g_t^* + e_t)$  and thus

$$\begin{aligned} \langle g_t^*, \Delta_t \rangle - \frac{L}{2} \|\Delta_t\|^2 &= \frac{1}{L} (\|g_t^*\|^2 + \langle g_t^*, e_t \rangle) - \frac{1}{2L} \|g_t^* + e_t\|^2 \\ &= \frac{1}{L} (\|g_t^*\|^2 + \langle g_t^*, e_t \rangle) - \frac{1}{2L} (\|g_t^*\|^2 + 2\langle g_t^*, e_t \rangle + \|e_t\|^2) \\ &= \frac{1}{2L} \|g_t^*\|^2 - \frac{1}{2L} \|e_t\|^2. \end{aligned} \quad (7)$$

Plugging equation 7 into equation 6 and taking  $\mathbb{E}_t[\cdot]$ ,

$$\mathbb{E}_t[J(\theta_{t+1})] \geq J(\theta_t) + \frac{1}{2L} \|g_t^*\|^2 - \frac{1}{2L} \mathbb{E}_t \|e_t\|^2 \quad (8)$$

Thus the entire task reduces to bounding  $\mathbb{E}_t \|e_t\|^2$ .

**B. Variance–bias decomposition of  $\mathbb{E}_t \|e_t\|^2$ .** We decompose  $e_t$  into three parts:

$$e_t = \underbrace{\left( \frac{1}{M} \sum_{k \in S_t} \nabla J_k(\theta_t) - \nabla J(\theta_t) \right)}_{\text{client sampling}} + \underbrace{\frac{1}{M} \sum_{k \in S_t} (\bar{g}_{k,t} - \mathbb{E}_t \bar{g}_{k,t})}_{\text{local stochastic noise}} + \underbrace{\frac{1}{M} \sum_{k \in S_t} b_{k,t}}_{\text{local drift}}, \text{ where } b_{k,t} := \mathbb{E}_t[\bar{g}_{k,t}] - \nabla J_k(\theta_t). \quad (9)$$

We now bound the mean-squared norm of each contribution. (All bounds hold component-wise and hence for the Euclidean norm.)

**Lemma 1** (FPC: client-sampling variance). *Let  $x_1, \dots, x_K \in \mathbb{R}^d$ ,  $\bar{x} = \frac{1}{K} \sum_k x_k$ , and  $S$  be a uniform size- $M$  sample without replacement with  $|S| = M$  where  $1 \leq M \leq K$ . Then*

$$\mathbb{E} \left\| \frac{1}{M} \sum_{k \in S} x_k - \bar{x} \right\|^2 = \frac{K-M}{M(K-1)} \cdot \frac{1}{K} \sum_{k=0}^{K-1} \|x_k - \bar{x}\|^2.$$

*Proof.* Let  $\Sigma := \frac{1}{K} \sum_{k=0}^{K-1} (x_k - \bar{x})(x_k - \bar{x})^\top$ . This is the standard finite-population correction (Cochran, 1977):  $\text{Cov}(\frac{1}{M} \sum_{k \in S} x_k) = \frac{K-M}{M(K-1)} \Sigma$ . Taking trace on both sides yields the claim since  $\mathbb{E} \|Z - \mathbb{E} Z\|^2 = \text{tr Cov}(Z)$ .  $\square$

Applying Lemma 1 with  $x_k = \nabla J_k(\theta_t)$  and using  $\nabla J(\theta_t) = \frac{1}{K} \sum_{k=0}^{K-1} \nabla J_k(\theta_t)$ , we obtain

$$\mathbb{E}_t \left\| \frac{1}{M} \sum_{k \in S_t} \nabla J_k(\theta_t) - \nabla J(\theta_t) \right\|^2 \leq \frac{K-M}{M(K-1)} \zeta^2. \quad (10)$$

**Lemma 2** (Local stochastic noise). *With the standing bounded-second-moment assumption, for each client  $k$  and round  $t$ ,  $\mathbb{E}_t \|\bar{g}_{k,t} - \mathbb{E}_t \bar{g}_{k,t}\|^2 \leq G^2 + \sigma^2$ . Moreover, conditioned on  $\theta_t$  and  $S_t$ , the per-client noises are independent across  $k \in S_t$ . Consequently,*

$$\mathbb{E}_t \left\| \frac{1}{M} \sum_{k \in S_t} (\bar{g}_{k,t} - \mathbb{E}_t \bar{g}_{k,t}) \right\|^2 \leq \frac{G^2 + \sigma^2}{M}. \quad (11)$$

*Proof.* Since  $\bar{g}_{k,t} = \frac{1}{\tau} \sum_{i=0}^{\tau-1} g_{k,t,i}$  and  $\mathbb{E} \|g_{k,t,i}\|^2 \leq G^2 + \sigma^2$ , we have  $\mathbb{E}_t \|\bar{g}_{k,t}\|^2 \leq G^2 + \sigma^2$ , hence  $\mathbb{E}_t \|\bar{g}_{k,t} - \mathbb{E}_t \bar{g}_{k,t}\|^2 \leq \mathbb{E}_t \|\bar{g}_{k,t}\|^2 \leq G^2 + \sigma^2$ . Independence across clients (conditional on  $\theta_t, S_t$ ) implies that variances add, yielding equation 11.  $\square$

**Lemma 3** (Local drift/bias bound). *Let*

$$b_{k,t} := \mathbb{E}_t[\bar{g}_{k,t}] - \nabla J_k(\theta_t), \quad \bar{g}_{k,t} = \frac{1}{\tau} \sum_{i=0}^{\tau-1} g_{k,t,i},$$

where the local iterates satisfy  $\theta_{k,t,0} = \theta_t$ ,  $\theta_{k,t,i+1} = \theta_{k,t,i} + \eta g_{k,t,i}$ ,  $\mathbb{E}[g_{k,t,i} \mid \theta_{k,t,i}] = \nabla J_k(\theta_{k,t,i})$ , and  $\mathbb{E}\|g_{k,t,i}\|^2 \leq G^2 + \sigma^2$ . If  $J_k$  is  $L$ -smooth, then

$$\mathbb{E}_t \|b_{k,t}\|^2 \leq \frac{L^2 \eta^2 (\tau - 1)^2}{4} (G^2 + \sigma^2). \quad (12)$$

Moreover, for any sampled set  $S_t$  of size  $M$ ,

$$\mathbb{E}_t \left\| \frac{1}{M} \sum_{k \in S_t} b_{k,t} \right\|^2 \leq \frac{1}{M} \sum_{k \in S_t} \mathbb{E}_t \|b_{k,t}\|^2 \leq \frac{L^2 \eta^2 (\tau - 1)^2}{4} (G^2 + \sigma^2). \quad (13)$$

*Proof.* By definition and  $L$ -smoothness of  $J_k$ ,

$$b_{k,t} = \frac{1}{\tau} \sum_{i=0}^{\tau-1} (\nabla J_k(\theta_{k,t,i}) - \nabla J_k(\theta_t)) = \frac{1}{\tau} \sum_{i=1}^{\tau-1} H_{k,t,i} (\theta_{k,t,i} - \theta_t),$$

where each  $H_{k,t,i}$  is a (mean-value) linear map with operator norm  $\|H_{k,t,i}\| \leq L$ . Using the local recursion  $\theta_{k,t,i} - \theta_t = \eta \sum_{j=0}^{i-1} g_{k,t,j}$  and swapping sums gives

$$b_{k,t} = \frac{\eta}{\tau} \sum_{j=0}^{\tau-2} \left( \sum_{i=j+1}^{\tau-1} H_{k,t,i} \right) g_{k,t,j} =: \frac{\eta}{\tau} \sum_{j=0}^{\tau-2} A_{k,t,j} g_{k,t,j},$$

with  $A_{k,t,j} := \sum_{i=j+1}^{\tau-1} H_{k,t,i}$  and hence  $\|A_{k,t,j}\| \leq \sum_{i=j+1}^{\tau-1} \|H_{k,t,i}\| \leq L(\tau - 1 - j)$ .

Applying the weighted Cauchy-Schwarz inequality,

$$\left\| \sum_j A_{k,t,j} g_{k,t,j} \right\|^2 \leq \left( \sum_j \|A_{k,t,j}\| \right) \left( \sum_j \frac{\|A_{k,t,j} g_{k,t,j}\|^2}{\|A_{k,t,j}\|} \right) \leq \left( \sum_j \|A_{k,t,j}\| \right) \left( \sum_j \|A_{k,t,j}\| \|g_{k,t,j}\|^2 \right),$$

and taking  $\mathbb{E}_t$  together with  $\mathbb{E}\|g_{k,t,j}\|^2 \leq G^2 + \sigma^2$  yields

$$\mathbb{E}_t \|b_{k,t}\|^2 \leq \frac{\eta^2}{\tau^2} \left( \sum_{j=0}^{\tau-2} \|A_{k,t,j}\| \right)^2 (G^2 + \sigma^2) \leq \frac{\eta^2}{\tau^2} \left( L \sum_{j=0}^{\tau-2} (\tau - 1 - j) \right)^2 (G^2 + \sigma^2).$$

Since  $\sum_{j=0}^{\tau-2} (\tau - 1 - j) = \sum_{m=1}^{\tau-1} m = \frac{\tau(\tau-1)}{2}$ , we obtain

$$\mathbb{E}_t \|b_{k,t}\|^2 \leq L^2 \eta^2 \frac{(\tau - 1)^2}{4} (G^2 + \sigma^2),$$

which is equation 12. For the client average, convexity of the squared norm (or Jensen) gives

$$\mathbb{E}_t \left\| \frac{1}{M} \sum_{k \in S_t} b_{k,t} \right\|^2 \leq \frac{1}{M} \sum_{k \in S_t} \mathbb{E}_t \|b_{k,t}\|^2,$$

and the second inequality in equation 13 follows by applying equation 12 to each  $k \in S_t$ .  $\square$

With Lemma 3 in place, combining equation 10, equation 11, equation 13, equation 9, and Young's inequality  $2\langle X, Y \rangle \leq \|X\|^2 + \|Y\|^2$  gives the (assumption-free) second-moment control

$$\mathbb{E}_t \|e_t\|^2 \leq \frac{G^2 + \sigma^2}{M} + \frac{2(K - M)}{M(K - 1)} \zeta^2 + \frac{L^2 \eta^2 (\tau - 1)^2}{2} (G^2 + \sigma^2). \quad (14)$$

**C. Closing the one-step inequality.** Insert equation 14 into equation 8 and use  $\eta = \frac{1}{L\tau}$  to get

$$\mathbb{E}_t [J(\theta_{t+1})] \geq J(\theta_t) + \frac{1}{2L} \|\nabla J(\theta_t)\|^2 - \frac{1}{2L} \left[ \frac{G^2 + \sigma^2}{M} + \frac{2(K - M)}{M(K - 1)} \zeta^2 + \frac{(\tau - 1)^2}{2\tau^2} (G^2 + \sigma^2) \right] \quad (15)$$

1404 **D. PL inequality, recursion, and averaging.** Let  $\delta_t := \mathbb{E}[J(\theta^*) - J(\theta_t)]$ . By the PL condition,  
 1405  $\|\nabla J(\theta_t)\|^2 \geq 2\mu \delta_t$ . Taking total expectation of equation 15 and using  $\eta = \frac{1}{L^\tau}$  together with the  
 1406 variance-bias bound that includes the mixed-term control (i.e.,  $2\langle S, D \rangle \leq \|S\|^2 + \|D\|^2$ ), we obtain  
 1407

$$1408 \delta_{t+1} \leq \left(1 - \frac{\mu}{L}\right) \delta_t + \frac{1}{2L} \left[ \frac{G^2 + \sigma^2}{M} + \frac{2(K-M)}{M(K-1)} \zeta^2 + \frac{(\tau-1)^2}{2\tau^2} (G^2 + \sigma^2) \right]. \quad (16)$$

1411 Summing equation 16 over  $t = 0, \dots, T-1$  and dividing by  $T$ , and noting that  $\sum_{t=0}^{T-1} (\delta_{t+1} - \delta_t) =$   
 1412  $\delta_T - \delta_0 \leq \delta_0$ , yields

$$1413 \frac{1}{T} \sum_{t=0}^{T-1} \delta_t \leq \frac{L}{\mu T} \delta_0 + \frac{1}{2\mu} \left[ \frac{G^2 + \sigma^2}{M} + \frac{2(K-M)}{M(K-1)} \zeta^2 + \frac{(\tau-1)^2}{2\tau^2} (G^2 + \sigma^2) \right].$$

1416 Finally, let  $\tilde{\theta}$  be drawn uniformly from  $\{\theta_t\}_{t=0}^{T-1}$ . Then  $\mathbb{E}[J(\theta^*) - J(\tilde{\theta})] = \frac{1}{T} \sum_{t=0}^{T-1} \delta_t$ , which gives  
 1417 the claimed bound.  $\square$   
 1418  
 1419  
 1420  
 1421  
 1422  
 1423  
 1424  
 1425  
 1426  
 1427  
 1428  
 1429  
 1430  
 1431  
 1432  
 1433  
 1434  
 1435  
 1436  
 1437  
 1438  
 1439  
 1440  
 1441  
 1442  
 1443  
 1444  
 1445  
 1446  
 1447  
 1448  
 1449  
 1450  
 1451  
 1452  
 1453  
 1454  
 1455  
 1456  
 1457