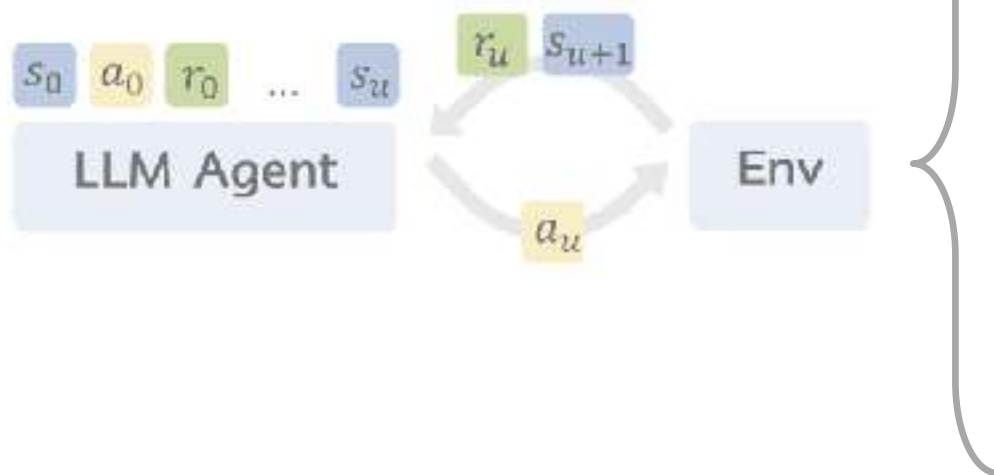


Federated Agent Reinforcement Learning

Centralized Agent Training

- Training AI agents typically requires *centralized* access to vast amounts of users' real-world task query and trajectory data.



Centralized Agent Training

- Training AI agents typically requires *centralized* access to vast amounts of users' real-world task query and trajectory data.

How to train AI agents while protecting users' data privacy?

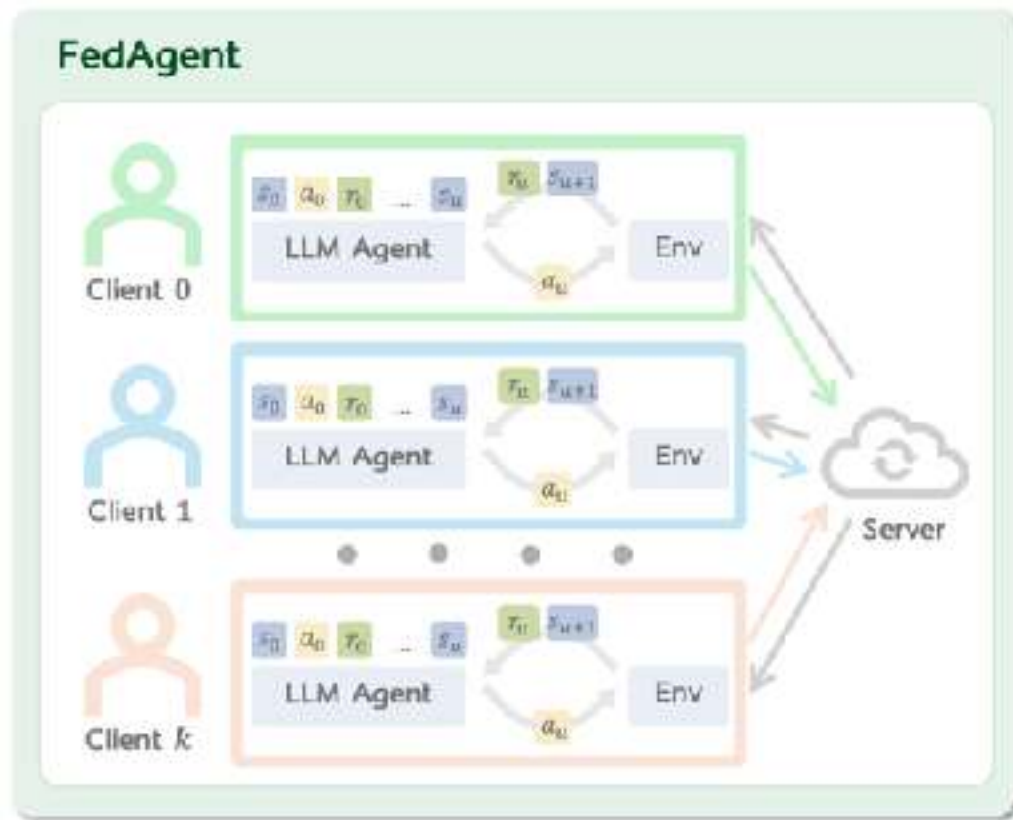
a_u



ALFWorld

FedAgent: Federated Agent Reinforcement Learning

- We explored a new *decentralized* paradigm of training AI agents, namely **FedAgent**.
- It enables collaborative agent learning across distributed clients without sharing local data, which preserve users' data privacy.



FedAgent: Federated Agent Reinforcement Learning

- 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.

Algorithm 1 FedAGENT with Client and Server training

Require: Total clients K , rounds T , clients-per-round M , local steps r , learning rate η

Ensure: Final LLM-based global policy parameters θ_{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  $r - 1$  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  $k$ :

$$g_{k,t,i} \leftarrow \nabla_{\Delta} J_k(\theta_{k,t,i}; B_{k,t,i}) \quad (\text{e.g., GRPO})$$

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  $\theta_{k,t,r}$   $\triangleright$  equivalently  $\Delta \theta_{k,t} = \theta_{k,t,r} - \theta_t$ 
13:  end for
14:  Server: Aggregation via model averaging:

$$\theta_{t+1} \leftarrow \frac{1}{M} \sum_{k \in S_t} \theta_{k,t,r} \quad (\text{equivalently } \theta_{t+1} = \theta_t - \frac{1}{M} \sum_{k \in S_t} (\theta_{k,t,r} - \theta_t))$$

15: end for
16: return  $\theta_{\text{final}} \leftarrow \theta_T$ 

```

FedAgentGym: A Decentralized Agent Learning Environment

- 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.
- Four types of LLM agents, two applications, three variations of decentralized settings, three new heterogeneity challenges.



Convergence of FedAgent

- FedAgent converges to a neighborhood of the optimum under the PL and smoothness conditions.

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 Algorithm 1. Let each selected client perform τ local steps with stepsize η . 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*

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

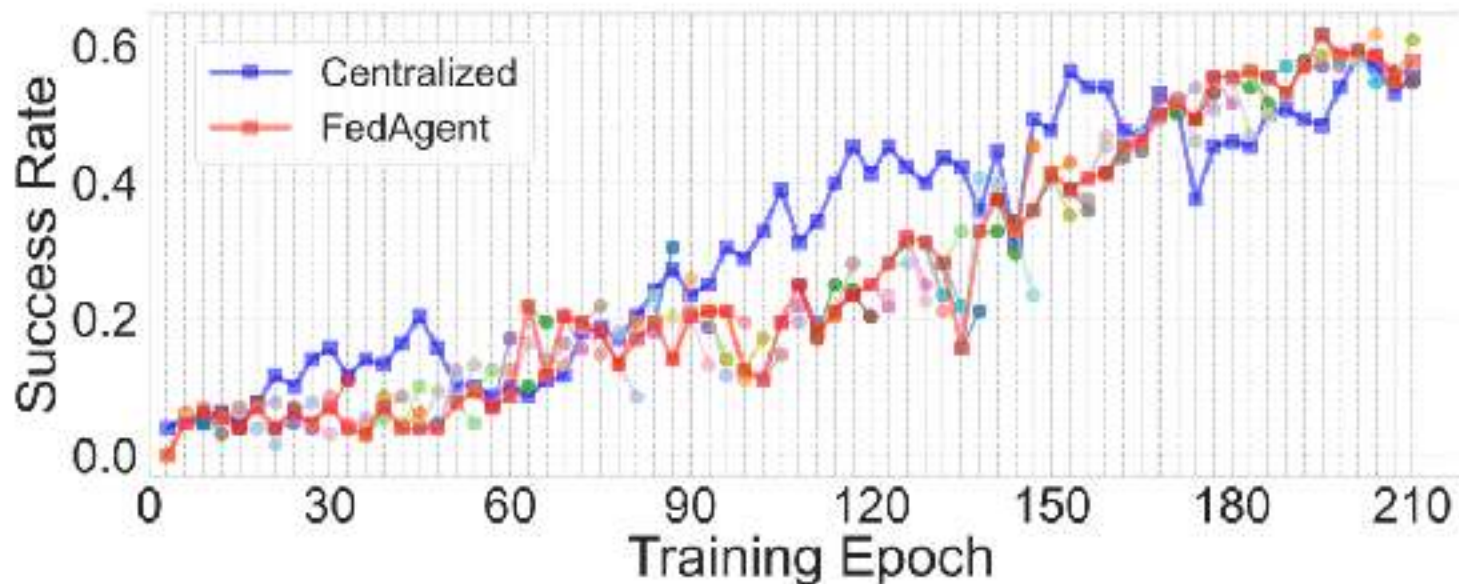
Is FedAgent Comparable to Centralized Agent Learning?

- Setup: 100 clients, each client has 100 task instructions, 2 clients are randomly selected each round, 3 epochs per round, a total of 70 rounds and 210 epochs overall.
- **FedAgent** consistently outperforms **Local Agent Training** and achieves comparable performance to **Centralized Agent Training**.

Method	ALFWorld							WebShop	
	Pick	Look	Clean	Heat	Cool	Pick2	All	Score	Success
<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	76.9	25.0	62.9	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 _{±4.0}	37.5 _{±0.0}	69.2 _{±6.1}	50.0 _{±4.2}	42.9 _{±3.9}	28.6 _{±0.4}	51.6 _{±3.0}	79.9 _{±4.7}	57.8 _{±5.7}
FEDAGENT	80.0 _{±4.2}	75.0 _{±1.7}	53.8 _{±1.3}	77.5 _{±1.0}	83.3 _{±1.7}	50.0 _{±1.0}	64.1 _{±2.8}	83.2 _{±1.5}	61.7 _{±1.4}
<i>Qwen2.5-3B-Instruct</i>									
Local (Client 21)	41.5	12.5	34.9	51.0	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	68.2	36.5	29.9	77.6	58.6
Centralized	94.1 _{±6.9}	80.0 _{±2.2}	64.3 _{±1.4}	42.9 _{±5.6}	50.0 _{±2.7}	22.2 _{±5.2}	62.5 _{±4.2}	70.0 _{±1.5}	53.9 _{±2.3}
FEDAGENT	95.5 _{±4.8}	62.5 _{±8.6}	49.7 _{±1.7}	47.5 _{±2.5}	85.3 _{±3.5}	45.1 _{±5.1}	65.2 _{±3.0}	85.5 _{±3.4}	63.1 _{±3.1}
<i>Qwen2.5-7B-Instruct</i>									
Local (Client 21)	25.5	25.0	51.0	25.9	35.6	45.2	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 _{±4.6}	82.5 _{±2.1}	71.5 _{±3.3}	47.9 _{±5.7}	63.2 _{±3.3}	31.9 _{±1.0}	73.3 _{±4.0}	78.8 _{±2.8}	64.7 _{±1.1}
FEDAGENT	94.5 _{±2.8}	85.0 _{±4.1}	56.0 _{±0.3}	62.5 _{±1.0}	86.7 _{±2.3}	42.8 _{±3.4}	75.5 _{±2.0}	89.0 _{±4.1}	68.9 _{±3.5}
<i>Llama 3.2 3B-Instruct</i>									
Local (Client 21)	29.8	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	37.6	29.7	70.2	55.7
Centralized	72.4 _{±4.0}	62.5 _{±4.2}	59.3 _{±3.1}	45.2 _{±6.5}	53.7 _{±2.2}	27.9 _{±5.0}	54.9 _{±2.3}	72.3 _{±3.7}	56.2 _{±1.1}
FEDAGENT	83.7 _{±1.7}	57.5 _{±6.6}	60.6 _{±3.4}	55.9 _{±6.0}	65.3 _{±2.5}	24.9 _{±3.1}	61.2 _{±3.3}	74.4 _{±4.0}	57.8 _{±3.2}

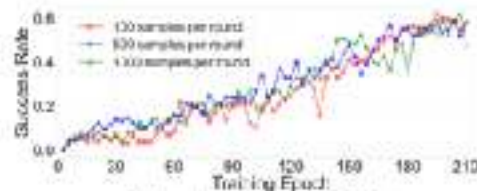
Is FedAgent Comparable to Centralized Agent Learning?

- Both paradigms (FedAgent and Centralized Agent Training) ultimately **converge to similar success rates** despite different training dynamics.

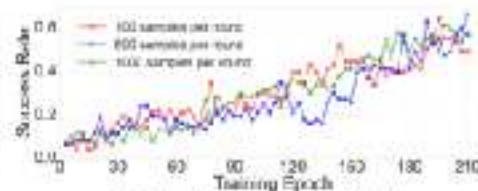


Impact of Decentralized Settings

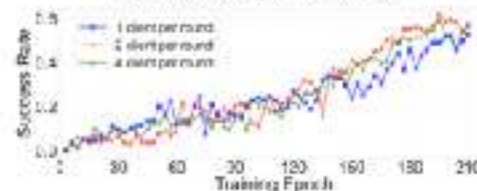
- Three different decentralized settings: (1) samples per client; (2) clients selected per round; (3) epochs per client per round.
- FedAgent exhibits distinct sensitivity patterns towards decentralized settings, depending on the specific hyperparameter and dataset.
- Optimal FedAgent configurations are environment-dependent.



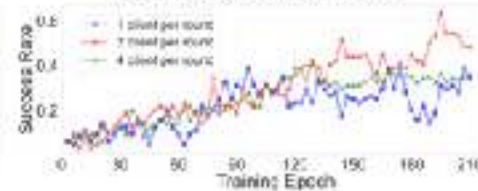
(a) Samples per client, WebShop



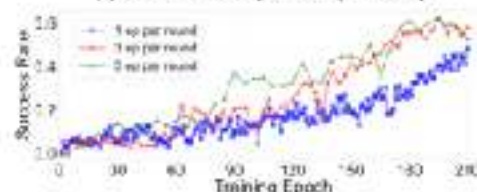
(b) Samples per client, ALFWorld



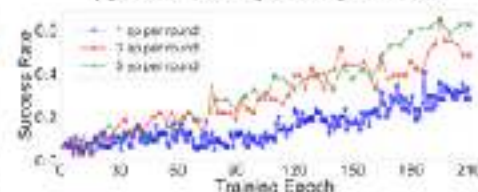
(c) Clients selected per round, WebShop



(d) Clients selected per round, ALFWorld



(e) Epochs per client per round, WebShop



(f) Epochs per client per round, ALFWorld

Impact of Heterogeneity Challenges

- Three different heterogeneity challenges: Preference, Coverage, or Hardness Heterogeneity.
- FedAgent shows high robustness against the three heterogeneity challenges.
- FedAgent has great potential to handle real-world scenarios across the full spectrum of heterogeneity challenges.

