Approximated Behavioral Metric-based State Projection for Federated Reinforcement Learning

Zengxia Guo[1,2], **Bohui An**[1,2], **Zhongqi Lu**[1,2*]

[1]College of Artificial Intelligence, China University of Petroleum-Beijing, China
[2]Hainan Institute of China University of Petroleum (Beijing), Sanya, Hainan, China
2024211272@student.cup.edu.cn, 2024211271@student.cup.edu.cn, zhongqi@cup.edu.cn

Abstract

Federated reinforcement learning (FRL) methods usually share the encrypted local state or policy information and help each client to learn from others while preserving everyone's privacy. In this work, we propose that sharing the approximated behavior metric-based state projection function is a promising way to enhance the performance of FRL and concurrently provides an effective protection of sensitive information. We introduce FedRAG, a FRL framework to learn a computationally practical projection function of states for each client and aggregating the parameters of projection functions at a central server. The FedRAG approach shares no sensitive task-specific information, yet provides information gain for each client. We conduct extensive experiments on the DeepMind Control Suite to demonstrate insightful results.

1 Introduction

In recent years, federated learning has emerged as a new approach to enable data owners to collaboratively train each one's improved local model with the help of the privacy preserved information from others [Yang *et al.*, 2019a; Yang *et al.*, 2019b; Li *et al.*, 2020a; Wei *et al.*, 2020; Lyu *et al.*, 2020]. Federated reinforcement learning (FRL) applies federated learning principles to reinforcement learning [Zhuo *et al.*, 2019]. In FRL, multiple clients, each with their own local environments, collaborate to learn a collective optimal policy [Qi *et al.*, 2021].

Aggregating knowledge from clients in non-identical environments allows FRL to explore a huge state-action space, enhance sample efficiency and accelerate the learning process [Wang *et al.*, 2020]. However, FRL faces unique challenges primarily due to the different local environments and diverse data distributions among clients. In FRL, clients may experience very different states and rewards in their own environment, resulting in diverse data distribution. This diversity may lead to significant differences in the learning model, making it difficult for clients to converge to a robust common policy [Zhao *et al.*, 2018]. Additionally, FRL must ensure that sensitive information remains protected from exposure to other clients or the central server [Zhu *et al.*, 2019; Anwar and Raychowdhury, 2021].

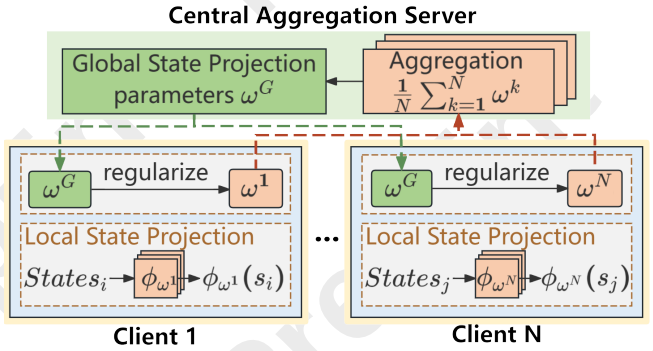

Figure 1: Framework of FedRAG. Periodically, the local state projection function parameters ω^k are synchronized to a central server. Then the central server distributes the averaged parameters to the clients. For each client, a regularization term is incorporated to ensure that the client's local state projection parameters follow the global updates.

Previous researches found that learning representation based behavioral metric can significantly accelerate the reinforcement learning process and enhance the generality of policy [Zhang *et al.*, 2020a; Agarwal *et al.*, 2021; Kemertas and Aumentado-Armstrong, 2021]. This method involves learning a state projection function by evaluating the behavioral similarities between states, which are measured in terms of rewards and state transition probabilities. The state projection function is valuable to the learning process, yet it does not reveal any sensitive task-specific information. In the FRL settings, clients would not directly share the rewards and state information because of the privacy issues. Therefore, sharing the parameters of the state projection function could be a promising research direction for FRL.

In this work, we propose the Federated Reinforcement Learning with Reducing Approximation Gap (FedRAG), a novel FRL framework to share parameters of state projection functions and to learn a local behavioral metric-based state projection function for each client. We detail FedRAG's network architecture in Figure 1, emphasizing how client collaboration is achieved through shared state projection functions. The global state projection function is formed by aggregat-

*Corresponding author.

ing local state projection functions, each trained with behavioral metrics to capture the unique transition dynamics and rewards of its respective environment. By integrating these locally learned features, the global state projection function reflects the diverse dynamics and rewards across different environments. Periodically, each client's local state projection function is replaced with the global state projection function, while the L2 regularization is continuously applied to maintain alignment throughout the learning process. Together, these mechanisms improve local state projection function and strategies that are robust and adaptable across varied environments.

The main contributions are as follows:

- We propose FedRAG, a novel federated reinforcement learning framework to share the projection function of states, instead of traditionally sharing the encrypted states information. Subsequent analysis show that our method is beneficial to privacy-preserving as a side-effect.
- Under the FedRAG framework, we introduce a behavioral metric-based state projection function and develop its practical approximation algorithm in Federated Learning settings. Empirical results demonstrate our method is effective.

2 Related Work

Federated Learning. Federated Learning (FL) was first introduced in FedAvg [McMahan *et al.*, 2017], where training data remains distributed across mobile devices, and a shared model is learned by aggregating locally computed updates through iterative model averaging. Subsequently, FedProx [Li *et al.*, 2020b] addresses system heterogeneity and statistical variability in federated networks. It incorporates a proximal term into local optimizations, allowing for variable computational efforts across devices, which helps stabilize diverse local updates. To accommodate the inherent heterogeneity in FL, Per-FedAvg [Fallah *et al.*, 2020] was developed as a personalized approach. This method adapts Model-Agnostic Meta-Learning (MAML) to provide a suitable initial model that quickly adapts to each user's local data after training. Another innovation, pFedMe [T Dinh *et al.*, 2020] tackles the statistical diversity among clients by utilizing Moreau envelopes as client-specific regularized loss functions, effectively decoupling personalized model optimization from global model learning.

Federated Representation Learning. Recently, federated representation learning, which focuses on training models to extract effective feature representations directly from raw data, has become increasingly popular. LG-FedAvg [Liang *et al.*, 2020] optimizes for compact local representations on each device alongside a global model spanning all devices. FedRep [Collins *et al.*, 2021] learns a shared data representation among clients while maintaining unique local heads to enhance each client's model quality. Model Contrastive Learning (MOON) [Li *et al.*, 2021a] improves local update consistency by maximizing alignment between representations learned from local and global models. Additionally, the Federated Prototype-wise Contrastive Learning (FedPCL) approach [Tan *et al.*, 2022] was introduced, leveraging pre-trained neural networks as backbones to enable knowledge sharing through class prototypes while constructing client-specific representations using prototype-wise contrastive learning. FedCA [Zhang *et al.*, 2023] aggregates representations from each client, aligning them with a base model trained on public data to mitigate inconsistencies and misalignment in the representation space across clients. TurboSVM-FL [Wang *et al.*, 2024] accelerates convergence in federated classification tasks by employing support vector machines for selective aggregation and applying max-margin spread-out regularization on class embeddings. Despite these advancements, research in federated representation learning specific to reinforcement learning remains limited.

Federated Reinforcement Learning. Federated Reinforcement Learning enables clients to collaboratively learn a unified policy while preserving privacy by avoiding the exchange of raw trajectories. FedPG-BR [Fan *et al.*, 2021] addresses convergence and fault tolerance against adversarial attacks or random failures in homogeneous environments using variance-reduced policy gradients. However, it does not tackle the challenges of heterogeneous environments, which is the focus of our work. To address environmental heterogeneity, Jin *et al.* [2022] introduced QAvg and PAvg algorithms, employing value function-based and policy gradient methods. They further proposed personalized policies that embed environment-specific state transitions into low-dimensional vectors, improving generalization and efficiency. Similarly, Tang *et al.* [2022] developed FeSAC, based on the soft actor-critic framework, which isolates local policies from global integration and employs trend models to adapt to regional disparities. Building on these advancements, our work focuses on learning a federated behavioral metric-based state projection function to effectively generalize across diverse environments.

Behavioral Metrics-based Representation Learning. Behavioral metric-based representation learning aims to create an embedding space that preserves behavioral similarities based on transitions and immediate rewards. Bisimulation metrics [Ferns *et al.*, 2011] measure state behavioral similarities in probabilistic transition systems for continuous state-space Markov Decision Processes (MDPs). On-policy bisimulation metrics [Castro, 2020] focus on behaviors specific to a given policy π, incorporating a reward difference term and the Wasserstein distance between dynamics models. To address the computational challenges associated with the Wasserstein distance, the MICo distance [Castro *et al.*, 2021] was developed to compare dynamics model distributions by measuring the distance between sampled subsequent states. The Conservative State-Action Discrepancy [Liao *et al.*, 2023] separates the learning of the RL policy from the metric itself, focusing on the most divergent reward outcomes between states taking the same actions to define similarity in the embedding space. Chen and Pan [2022] propose the Reducing Approximation Gap distance to recursively measure expected states over dynamics models, focusing on sampling from the policy π rather than the dynamics models. This approach re-

duces approximation errors and is particularly effective for representation learning. In our work, we apply approximation behavior metric-based representation learning to develop local state projection functions, capturing task-relevant behavioral similarities within each client's environment. Federated Learning then allows for sharing the parameters of these local projection function, enabling clients to benefit from generalized state representations across diverse environments.

3 Preliminaries

This section highlights the Federated Soft Actor-Critic (FeSAC) variant central to our research. Soft Actor-Critic (SAC) is an off-policy actor-critic algorithm based on the maximum entropy reinforcement learning framework [Haarnoja *et al.*, 2018a]. It aims to maximize cumulative future rewards and entropy to enhance robustness and exploration while preventing convergence to suboptimal policies. FeSAC extends SAC to a federated setting, enabling collaborative training among clients operating in diverse environments while ensuring data privacy. The global environment $E = \{E^1, E^2, \ldots, E^N\}$ is composed of N distinct local environments, and each client k operates within its own unique local environment E^k. The transition probabilities differ across local environments, i.e., $P(s^i_{t+1}|s^i_t, a) \neq P(s^j_{t+1}|s^j_t, a), i \neq j$.

As our study focuses on applying approximated behavioral metric-based representation learning to FRL, we introduce the state projection function when discussing FeSAC. In the scope of representation learning for deep RL, a state projection function ϕ_{ω^k} maps a high-dimensional state to low-dimensional vector, from which the policy $\pi_{\psi^k}(a|\phi_{\omega^k}(s))$ is learned. We configure all critic networks, target critic networks, and action networks to take the state representation $\phi_{\omega^k}(s)$ as input instead of the raw state s.

Unlike traditional FRL, the objective of FeSAC is to derive a set of maximum entropy policies that are specifically optimized for their respective local environments. The target policy $\widetilde{\pi}^k$ for client k in its local environment E^k is

$$\widetilde{\pi}^k = \arg\max_{\pi^k} \sum_{t=0}^{T} \mathbb{E}_{(s^k_t, a^k_t)\sim\tau_{\pi^k}} \Big[\gamma^t r(s^k_t, a^k_t) + \alpha^k \mathcal{H}\big(\pi^k(\cdot \mid \phi_{\omega^k}(s^k_t))\big)\Big], \quad (1)$$

where s^k_t and a^k_t represent the state and action made by client k in its local environment E^k at time t; τ_{π^k} refers to the trajectory generated by the policy π^k of client k, which encompasses the sequence of states and actions over time; γ^k is the discount rate; α^k is the entropy regularization coefficient used to control the importance of entropy; $\mathcal{H}(\pi^k(\cdot|\phi_{\omega^k}(s^k_t))) = \mathbb{E}[-log\pi^k(\cdot|\phi_{\omega^k}(s^k_t))]$ represents the entropy of the policy.

To evaluate the impact of the policy on local environments, the soft state value is defined as:

$$V(s^k_t) = \mathop{\mathbb{E}}_{a^k_t\sim\pi_{\psi^k}} \big[Q_{\theta^k}(\phi_{\omega^k}(s^k_t), a^k_t) - \alpha^k \log \pi_{\psi^k}(a^k_t|\phi_{\omega^k}(s^k_t))\big], \quad (2)$$

where Q_{θ^k} denote the local critic Q network for client k.

Each client adjusts its local Q-network to approximate the global Q-network, thus leveraging global knowledge while retaining its own characteristics:

$$L_Q(\theta^k) = \mathbb{E}_{(s^k_t, a^k_t, r^k_t, s^k_{t+1})\sim\mathcal{D}^k} \Bigg[Q_{\theta^k}\big(\phi_{\omega^k}(s^k_t), a^k_t\big) - \big(r^k_t + \gamma V_{\bar{\theta}}(s^k_{t+1})\big) \Bigg]^2, \quad (3)$$

where $V_{\bar{\theta}}$ denotes use the target critic Q networks to calculate the soft state value.

In FeSAC, the target critic Q network refers to the global critic Q network, which is broadcasted by the server to all clients. The global critic Q network $Q_{\bar{\theta}}$ is formed by aggregating the local critic Q networks of each client through soft updates, considering the reward differences of state-action pairs in each client's environment to obtain a value estimation in a global context:

$$Q_{\bar{\theta}} \leftarrow \epsilon Q_{\theta^k} + (1-\epsilon) Q_{\bar{\theta}}, \quad k \in \{1, 2, \ldots, N\}, \quad (4)$$

where ϵ is the aggregation factor.

The updated local Q-network then guides the update of the local policy, which keeps the local variability as well as learning the implicit trend of the global environment:

$$L_\pi(\psi^k) = \mathop{\mathbb{E}}_{s^k_t\sim\mathcal{D}^k} \Bigg[\mathop{\mathbb{E}}_{a^k_t\sim\pi_{\psi^k}(\cdot|\phi_{\omega^k}(s^k_t))} [\alpha^k \log \pi_{\psi^k}(a^k_t \mid \phi_{\omega^k}(s^k_t)) - Q_{\theta^k}(\phi_{\omega^k}(s^k_t), a^k_t)] \Bigg]. \quad (5)$$

The temperature parameter α^k is adapted to balance exploration and exploitation by controlling the relative importance of the entropy term in the policy's objective. The update objective for α^k in client k is as follows [Haarnoja *et al.*, 2018b]:

$$L_\alpha(\alpha^k) = \mathop{\mathbb{E}}_{s^k_t\sim\mathcal{D}^k} \Big[\mathbb{E}_{a^k_t\sim\pi_{\psi^k}(\cdot|\phi_{\omega^k}(s^k_t))} [\alpha^k \log \pi_{\psi^k}(a^k_t|\phi_{\omega^k}(s^k_t)) - \alpha^k \bar{\mathcal{H}}] \Big], \quad (6)$$

where $\bar{\mathcal{H}}$ is a target entropy level to tune the degree of exploration and $\bar{\mathcal{H}} = -|\mathcal{A}|$.

4 Methodology

In this section, we present the problem formulation for federated reinforcement learning with heterogeneous environments, introduce the approximated behavioral metric-based state projection function, propose the FedRAG framework and provide a theoretical analysis of its privacy preserving.

4.1 Problem Formulation

In federated reinforcement learning with heterogeneous environments, N clients each interact with their own unique local environment E^k, each modeled as a unique Markov Decision Process (MDP): $\{S^k, A, R^k, P^k, \gamma\}$. Each client has a unique state space S^k, reward function $R^k(s, a)$, and state transition dynamics $P^k(s'|s, a)$, reflecting the diversity of their environments, while sharing a common action space

A and discount factor γ. A central server facilitates collaboration by periodically aggregating and distributing shared model parameters, specifically the state projection function ϕ_ω in FedRAG. This function maps local states to a shared embedding space, enabling clients to benefit from collective learning while preserving privacy. FedRAG optimizes local policies $\pi^k\left(a|\phi_{\omega^k}\left(s\right)\right)$ by sharing the parameters of ϕ_ω, aiming to maximize cumulative reward and entropy:

$$\widetilde{\pi}^k = \arg\max_{\pi^k} \frac{1}{N}\sum_{k=1}^{N}\left\{\sum_{t=0}^{\infty}\mathbb{E}_{(s_t^k,a_t^k)\sim\tau_{\pi^k}}\left[\gamma^t R^k\left(s_t^k,a_t^k\right)\right.\right. \\ \left.\left. + \alpha^k\mathcal{H}\left(\pi_{\psi^k}\left(\cdot\mid\phi_{\omega^k}\left(s_t^k\right)\right)\right)\right]\right\}, \tag{7}$$

where $a_t^k \sim \pi^k(\cdot|\phi_{\omega^k}(s_t^k))$, $s_{t+1}^k \sim P^k(\cdot|s_t^k,a_t^k)$. To preserve data privacy, only the parameters of the state projection function ω are shared between clients and the server, while raw states, rewards, and transition dynamics remain local to each client, ensuring sensitive information is not exchanged while enabling effective federated learning.

4.2 Client RAG Distance

In FeSAC, clients across different environments share knowledge by aligning their local Q networks with the global Q network, enabling optimal local policies while adapting to changes. However, in complex environments, clients may struggle to capture task-relevant information (see Section 5.2), leading to unclear global perceptions and hindering adaptation to environmental changes.

To enhance generalization in complex environments, we introduce behavior metric-based representation learning. This approach learns robust state representations that filter out task-irrelevant background information, speeding up the learning process and improving policy generalization across diverse environments.

For each client k, behavioral metric-based representation learning is to learn a local state encoding network $\phi_{\omega^k}: S^k \to \mathbb{R}^n$ with parameters ω^k, which can be cast as a minimization problem of the loss between the distance on the embedding space, $\hat{d}(\phi_{\omega^k}\left(s_i^k\right),\phi_{\omega^k}\left(s_j^k\right))$, and the corresponding behavior metric, $d^\pi(s_i^k,s_j^k)$, between any pair of states s_i^k and s_j^k:

$$L_\phi(\omega^k) = \mathbb{E}\left[\left(\hat{d}(\phi_{\omega^k}(s_i^k),\phi_{\omega^k}(s_j^k)) - d^\pi(s_i^k,s_j^k)\right)^2\right]. \tag{8}$$

The Reducing Approximation Gap (RAG) distance is a behavioral metric that measures the absolute difference between the reward expectations of two states and the distance between the next state expectations of dynamics models. And it is defined as follows:

$$d^\pi(s_i^k,s_j^k) = \left|\mathbb{E}_{a_i^k\sim\pi^k} r_{s_i^k}^{a_i^k} - \mathbb{E}_{a_j^k\sim\pi^k} r_{s_j^k}^{a_j^k}\right| \\ + \gamma\mathbb{E}_{a_i^k\sim\pi^k,a_j^k\sim\pi^k} d^\pi(\mathbb{E}[s_{i+1}^k],\mathbb{E}[s_{j+1}^k]), \tag{9}$$

where $\mathbb{E}_{a_i^k\sim\pi^k} r_{s_i^k}^{a_i^k}$ represents the expected reward obtained by taking action a_i^k in state s_i^k under the policy π^k of client k, $\mathbb{E}[s_{i+1}^k] = \mathbb{E}_{s_{i+1}^k\sim P_{s_i^k}^{a_i^k}}[s_{i+1}^k]$ is the expectation value of next state over the dynamics model $P(s_i^k,a_i^k)$.

Then the approximation of RAG relax the computationally intractable reward difference term without introducing any approximate gap, as shown below:

$$d^\pi(s_i^k,s_j^k) \\ = \sqrt{\mathbb{E}_{a_i^k\sim\pi^k,a_j^k\sim\pi^k}\left[\left(r_{s_i^k}^{a_i^k} - r_{s_j^k}^{a_j^k}\right)^2\right] - \mathrm{Var}[r_{s_i^k}] - \mathrm{Var}[r_{s_j^k}]} \\ + \gamma\mathbb{E}_{a_i^k\sim\pi^k,a_j^k\sim\pi^k} d^\pi\left(\mathbb{E}[s_{i+1}^k],\mathbb{E}[s_{j+1}^k]\right). \tag{10}$$

Since the reward variance $\mathrm{Var}[r_{s_i^k}]$ is computationally intractable, we can learn a neural network approximator to estimate it by assuming that the reward r_{s^k} on state s^k is Gaussian distributed. Let $\hat{R}_{\xi^k}(s^k) = \{\hat{\mu}(r_{s^k}),\hat{\sigma}(r_{s^k})\}$ be the learned reward function approximation parameterized by ξ^k, which outputs a Gaussian distribution. The loss function is:

$$L_{\hat{R}}(\xi^k) = \mathbb{E}_{(s^k,r^k)\sim\mathcal{D}^k}\left[\left(\frac{r^k-\hat{\mu}(r_{s^k})}{2\hat{\sigma}(r_{s^k})}\right)^2\right], \tag{11}$$

where $\hat{\mu}$ and $\hat{\sigma}$ are the mean and the standard deviation, respectively.

Similarly, in order to estimate the expected next states $\mathbb{E}[s_{i+1}^k]$, we learn a dynamics model $\hat{P}_{\eta^k}(\phi_{\omega^k}(s),a) = \{\hat{\mu}(\hat{P}^a_{\phi_{\omega^k}(s)}),\hat{\sigma}(\hat{P}^a_{\phi_{\omega^k}(s)})\}$ for each client, which outputs a Gaussian distribution over the next state embedding:

$$L_{\hat{P}}(\eta^k) = \mathbb{E}_{\mathcal{D}^k}\left[\left(\frac{\phi_{\omega^k}(s_{i+1}^k) - \hat{\mu}\left(\hat{P}^{a_i^k}_{\phi_{\omega^k}(s_i^k)}\right)}{2\hat{\sigma}\left(\hat{P}^{a_i^k}_{\phi_{\omega^k}(s_i^k)}\right)}\right)^2\right]. \tag{12}$$

Based on the above approximation, the RAG loss for each client can be defined as:

$$L_{\mathrm{RAG}}(\phi_{\omega^k}) = \mathbb{E}_{\mathcal{D}^k} \\ \left[\left(\hat{d}\left(\phi_{\omega^k}(s_i^k),\phi_{\omega^k}(s_j^k)\right) - \gamma\hat{d}(\hat{\mu}(\hat{P}^{a_i^k}_{\phi_{\omega^k}(s_i^k)}),\hat{\mu}(\hat{P}^{a_j^k}_{\phi_{\omega^k}(s_j^k)}))\right)^2\right. \\ \left. - \left(\left|r_{s_i^k}^{a_i^k} - r_{s_j^k}^{a_j^k}\right|^2 - (\hat{\sigma}(r_{s_i^k}))^2 - (\hat{\sigma}(r_{s_j^k}))^2\right)\right]^2, \tag{13}$$

where $\mathcal{D}^k$ represents the replay buffer or the set of data collected from environment E^k by the RL algorithm, e.g. SAC.

Considering that the behavior metric has non-zero self-distance, the distance on the embedding space adopts the approximate form proposed in MICo [Castro *et al.*, 2021], which produces a non-zero self-distance and helps in maintaining proximity between similar states rather than pushing them apart:

$$\hat{d}(\phi(s_i^k),\phi(s_j^k)) = \|\phi(s_i^k)\|^2 + \|\phi(s_j^k)\|^2 + K\varphi(\phi(s_i^k),\phi(s_j^k)), \tag{14}$$

while φ is absolute angle distance and K is a hyperparameter.

Algorithm 1 FedRAG algorithm

Initialize local networks ϕ_{ω^k}, $\phi_{\bar{\omega}^k}$, Q_{θ^k}, $Q_{\bar{\theta}^k}$, π_{ψ^k}, $\hat{R}_{\xi^k}$, $\hat{P}_{\eta^k}$ for each client $k \in \{1, 2, \ldots, N\}$, and global network ϕ_{ω^G} at the server.
Synchronize local and global parameters: $\omega^k, \bar{\omega}^k \leftarrow \omega^G$ for each client k.
Initialize empty replay memory $\mathcal{D}^k$ for each client k.
while running **do**
  for each client k **do**
    Observe state s_t from local environment E^k, sample action $a_t \sim \pi(\cdot\,|\phi_{\omega^k}(s_t))$ and execute.
    Receive reward $r_t \leftarrow R(s_t, a_t)$ and transition to next state $s_{t+1} \sim P(\cdot\,|s_t, a_t)$.
    Store transition (s_t, a_t, r_t, s_{t+1}) in $\mathcal{D}^k$.
    Update local networks Q_{θ^k}, π_{ψ^k}, α^k, $\hat{P}_{\eta^k}$, $\hat{R}_{\xi^k}$, ϕ_{ω^k} via gradient descent using Eq. 3,5,6,12,11,15
    Softly update target networks: $\bar{\theta}^k \leftarrow \tau_Q \theta^k + (1 - \tau_Q)\bar{\theta}^k$, $\bar{\omega}^k \leftarrow \tau_\phi \omega^k + (1 - \tau_\phi)\bar{\omega}^k$.
    if running n iterations **then**
      Upload ω^k to federated center node.
    end if
  end for
  if in federated center node **then**
    Aggregate global parameters: $\omega^G \leftarrow \frac{1}{N}\sum_{k=1}^{N} \omega^k$.
    Broadcast updated global parameters: $\omega^k, \bar{\omega}^k \leftarrow \omega^G$ for all clients.
  end if
end while

4.3 FedRAG Framework

Under the federated learning framework, we share the parameter ω of the state projection function ϕ_ω. The FedRAG framework operates with multiple clients and a federated central node. Each client k generates local parameters ω^k for the state projection function and updates policy networks based on their local environment. The federated central node collects these local parameters ω^k from all clients, aggregates them into a global distribution, and then distributes the updated global parameters back to the clients. Each client uses the state projection $\phi_{\omega^k}(s)$ as input for both the actor and critic networks. We assume that global ω follows a Gaussian distribution,with each client learning only a portion of the overall distribution. Therefore, we add a Gaussian regularization term after the RAG regression function Eq. 13, leading to the new loss formulation:

$$L_{\text{FedRAG}}(\phi_{\omega^k}) = \mathbb{E}_{\mathcal{D}^k} \Bigg[\left(\hat{d}\left(\phi_{\omega^k}(s_i^k), \phi_{\omega^k}(s_j^k)\right) - \gamma \hat{d}(\hat{\mu}(\hat{P}^{a_i^k}_{\phi_{\omega^k}(s_i^k)}), \hat{\mu}(\hat{P}^{a_j^k}_{\phi_{\omega^k}(s_j^k)})) \right)^2 - \left(\left| r^{a_i^k}_{s_i^k} - r^{a_j^k}_{s_j^k} \right|^2 - (\hat{\sigma}(r_{s_i^k}))^2 - (\hat{\sigma}(r_{s_j^k}))^2 \right) \Bigg]^2 + \frac{\lambda}{2}\|\omega^k - \omega^G\|_2^2, \tag{15}$$

where ω^G represents the expectation of the global Gaussian distribution. The regularization term helps reduce environmental heterogeneity, thereby enhancing collaborative learning, as demonstrated in Section 5.6.

The proposed FedRAG is detailed in Algorithm 1. During the FL process, we upload ω^k to the server periodically. According to the central limit theorem, we approximate the global Gaussian distribution by aggregating the mean of all local ω^k at the server. The server then distributes the results to each client, aligning local learning with the global distribution. By averaging local state projection function parameters, FedRAG integrates the specialized features learned in each client's environment. Each client can maintain its own local training advantages while incorporating the global nature, and perform better when dealing with data outside of its own.

4.4 Effectiveness of Anti-attack

Note that the data we aim to protect is not directly uploaded, potentially reducing the need for additional privacy techniques such as differential privacy or homomorphic encryption. Below, we analyze the privacy-preserving properties of our approach.

One of the major issues in federated learning is preserving privacy. In our analysis, we consider the existence of semi-honest adversaries. The adversaries may launch privacy attacks to snoop on the training data of other participants by analyzing periodic updates (e.g., gradients) of the joint model during training [Zhu *et al.*, 2019]. Such kind of attacks is referred to as Bayesian inference attack [Zhang *et al.*, 2022].

A Bayesian inference attack is an optimization process that aims to infer the private variable D_k to best fit client k protected exposed information W_k^S as

$$\begin{aligned} d^* &= \arg\max_d \log(f_{D_k|W_k^S}(d|w)) \\ &= \arg\max_d \log(\frac{f_{W_k^S|D_k}(w|d) f_{D_k}(d)}{f_{W_k^S}(w)}) \\ &= \arg\max_d [\log f_{W_k^S|D_k}(w|d) + \log f_{D_k(d)}] \end{aligned} \tag{16}$$

where $f_{D_k|W_k^S}(d|w)$ is the posterior of D_k given the protected variable W_k^S. According to Bayes's theorem, maximizing the log-posterior $f_{D_k|W_k^S}(d|w)$ on D_k involves maximizing summation of $\log\left(f_{W_k^S|D_k}(d|w)\right)$ and $\log\left(f_{D_k}(d)\right)$. The former one aims to find D_k to best match W_k^S, and the latter one aims to make the prior of D_k more significant. The learned conditional distribution $f_{D_k|W_k^S}$ from the Bayesian inference attack reflects the dependency between W_k^S and D_k, which determines the amount of information that adversaries may infer about D_k after observing W_k^S. However, in our approach, the parameter ω that we participate in federated learning is related to the representation function ϕ of the state. From the loss $L_{FedRAG}(\phi_\omega)$ in Equation 15, we can also see that ω is only related to the mapped state and reward, and has nothing to do with our private data state. Therefore, our proposed FedRAG protects the privacy of local state information to a certain extent.

5 Experiment

5.1 Experimental Settings

In this section, we evaluate the effectiveness and generalization of FedRAG using DeepMind Control Suite (DMC).

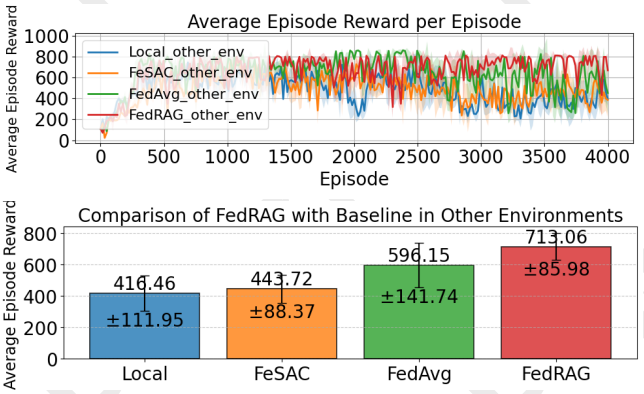

Figure 2: Comparison of FedRAG with Baseline in other environments.

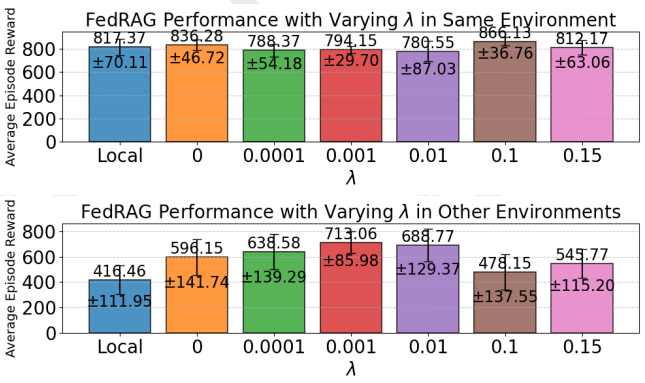

Figure 3: The results of varying lambda. In the above experiment, the training data and testing data are from environments with same setting, while in the below experiment, they are come from environments with different settings.

The DMC is a benchmark for control tasks in continuous action spaces with visual input [Tassa *et al.*, 2018]. We simulated different environments by modifying key physical parameters for several tasks: pole length (cartpole-swing), torso length (cheetah-run), finger distal length (finger-spin), and torso length (walker-walk).

As described in the previous section, each client projects state observation to the embedding space by using the approximated behavioral metric-based local state projection network, and updates local SAC network for policy evaluation and improvement. We perform experiments on 2 settings: 1) **Local**: clients can only interact and update local network in their own different environments without information sharing; 2) **Federated**: clients interact with their respective environments, update local network with information sharing according to federated methods.

In our study, we render 84×84 pixels and stack 3 frames as observation at each time step. We set an episode to consist of 125 environment steps, training over a total of 4000 episodes, which equates to 500,000 steps. For each setting, we evaluate the performance of each clients in both the same and other environments every 16 local update episodes. In the federated learning scenario, every 4 episodes, clients upload their local parameters, which the server then aggregates and redistributes as global parameters.

5.2 FedRAG vs. Baseline Performance Comparison

As illustrated in Figure 2, we compared our proposed FedRAG method ($\lambda = 0.001$) with FedAvg (equivalent to FedRAG with $\lambda = 0$), FeSAC, and Local methods in the Cart-Pole task with varying pole lengths. We assessed the average episode reward and standard deviation achieved by the clients in other environments. The results show that clients in the Local group, trained exclusively in their own environment without federated learning, struggled to adapt to other environments, resulting in the lowest performance. FeSAC had limited effectiveness in capturing task-relevant information in complex states, leading to only modest performance improvements. In contrast, FedRAG outperformed FedAvg by effectively integrating the global state projection function during local updates, resulting in significant performance gains in other environments.Unlike traditional FRL methods, FedRAG transmits only lightweight state projection parameters, significantly reducing communication overhead. While maintaining the state projection function incurs minor computational cost, it accelerates the RL process and enhances model generalization, yielding an overall benefit.

5.3 the Hyper-parameter of FedRAG

To investigate the impact of the regularization term in FedRAG (Eq.15), we evaluated the Local baseline and FedRAG with varying λ values in both same and other environments, as shown in Figure 3. In other environments, increasing λ strengthens the alignment between local and global state projection functions, promoting parameter sharing and enhancing cross-environment generalization. However, overly large λ constrains local updates, slowing convergence by limiting adaptation to local dynamics. The best performance was achieved at $\lambda = 0.001$. In the same environment, performance remained stable with minor fluctuations, demonstrating the robustness of FedRAG. This reveals a trade-off: larger λ promotes generalization across clients via global knowledge sharing, but may hinder specialization for local dynamics due to gradient conflicts from heterogeneous environments. Overall, while performance remained stable in the same environment across all λ values, notable improvements were observed in other environments, confirming the effectiveness of our federated approach.

5.4 Performance Improvement for Federated Learning

In Figure 4, we compare the performance of the FedRAG method ($\lambda = 0.1/0.001$) with the Local approach by evaluating average episode rewards in both the same and other environments. The Local approach limits clients to their own environments, resulting in local optimal policies that poorly generalize. In contrast, FedRAG aggregates local state projection functions on a central server to create a global state projection function. By sharing this global function during

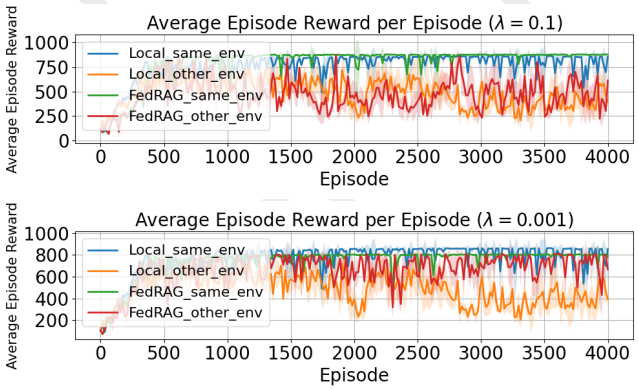

Figure 4: Comparison of Local and FedRAG with $\lambda = 0.1/0.001$ in same or other environments.

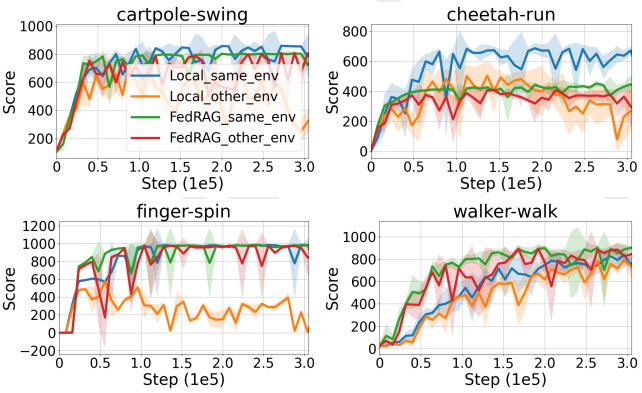

Figure 5: Experimental results on various DMC tasks.

local updates, clients benefit from cross-environment knowledge sharing while maintaining data privacy. With $\lambda = 0.1$, FedRAG enhances local performance by leveraging shared knowledge to overcome local optima, while also improving performance in other environments. At $\lambda = 0.001$, FedRAG achieves the best results in other environments with minimal loss in the same environment, demonstrating strong generalization and robustness across diverse settings.

5.5 FedRAG Performance on Various DeepMind Control Tasks

To evaluate the robustness and effectiveness of our method, we conducted experiments on several tasks from DMC and compared the average episode rewards of clients using our FedRAG method with $\lambda = 0.001$ and the non-federated Local method in both same and other environments, as illustrated in Figure 5. In cartpole-swing and finger-spin tasks, FedRAG significantly outperformed the Local method in other environments while maintaining near-optimal performance in the same environment. This success stems from its federated approach, which integrates global knowledge while preserving local training advantages. In cheetah-run task, Local clients trained only on their own environments exhibited declining performance in other environments over time. In contrast, FedRAG maintained stable performance in other environments,

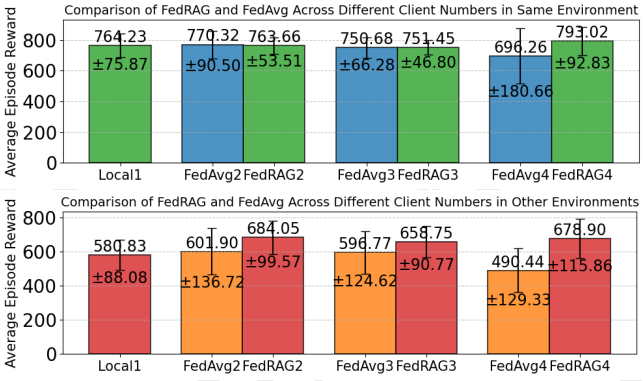

Figure 6: Comparison of FedRAG and FedAvg across different client numbers in same and other environments.

benefiting from global knowledge. By the end of training, FedRAG outperformed the Local method in cross-environment evaluations. In walker-walk task, FedRAG demonstrated faster convergence and higher episode rewards across all environments, benefiting from federated state projection functions that enhanced task-relevant feature extraction and generalization. These results confirm the robustness and generalization of FedRAG across diverse tasks and environments.

5.6 FedRAG Performance with Increasing Clients and Environmental Heterogeneity

We evaluated FedRAG's performance in heterogeneous environments with increasing client numbers N, comparing it to FedAvg. Using N pole lengths sampled uniformly from $[0.9, 1]$, we created diverse CartPole environments, with N ranging from 1 to 4. As N increased, greater environmental heterogeneity hindered policy convergence, while additional clients provided more learning information. As shown in Figure 6, FedAvg's performance deteriorated significantly in other environments, while FedRAG remained stable across both same and other settings, demonstrating its robustness to heterogeneity. This stability is attributed to FedRAG's behavior metric-based state projection, which captures task-relevant features and filters out environment-specific noise, enabling effective generalization. The L2 regularization ensures local updates align with the global model, capturing shared dynamics across clients and effectively adapting the projection function as the client count increases.

6 Conclusion

Sharing the parameters of the approximated behavior metric-based state projection function enhances the performance of FRL and protects sensitive local information. In this work, we propose FedRAG, a FRL framework that shares the parameters of the state projections among clients. Under the FedRAG framework, we introduce a behavioral metric-based state projection function and develop its practical approximation algorithm in Federated Learning settings. We conduct empirical studies on several reinforcement learning tasks to verify the effectiveness of our proposed method.

Acknowledgements

This work is supported by the Science Foundation of China University of Petroleum, Beijing (Grant No. 2462023YJRC024) and the Frontier Interdisciplinary Exploration Research Program of China University of Petroleum, Beijing (Grant No. 2462024XKQY003). Zhongqi Lu is the corresponding author.

References

[Abadi *et al.*, 2016] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[Agarwal *et al.*, 2021] Rishabh Agarwal, Marlos C Machado, Pablo Samuel Castro, and Marc G Bellemare. Contrastive behavioral similarity embeddings for generalization in reinforcement learning. *arXiv preprint arXiv:2101.05265*, 2021.

[Anwar and Raychowdhury, 2021] Aqeel Anwar and Arijit Raychowdhury. Multi-task federated reinforcement learning with adversaries. *CoRR*, abs/2103.06473, 2021.

[Castro *et al.*, 2021] Pablo Samuel Castro, Tyler Kastner, Prakash Panangaden, and Mark Rowland. Mico: Improved representations via sampling-based state similarity for markov decision processes. *Advances in Neural Information Processing Systems*, 34:30113–30126, 2021.

[Castro, 2020] Pablo Samuel Castro. Scalable methods for computing state similarity in deterministic markov decision processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10069–10076, 2020.

[Chen and Pan, 2022] Jianda Chen and Sinno Pan. Learning representations via a robust behavioral metric for deep reinforcement learning. *Advances in Neural Information Processing Systems*, 35:36654–36666, 2022.

[Collins *et al.*, 2021] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *International conference on machine learning*, pages 2089–2099. PMLR, 2021.

[Fallah *et al.*, 2020] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in neural information processing systems*, 33:3557–3568, 2020.

[Fan *et al.*, 2021] Xiaofeng Fan, Yining Ma, Zhongxiang Dai, Wei Jing, Cheston Tan, and Bryan Kian Hsiang Low. Fault-tolerant federated reinforcement learning with theoretical guarantee. *Advances in Neural Information Processing Systems*, 34:1007–1021, 2021.

[Fan *et al.*, 2023] Flint Xiaofeng Fan, Yining Ma, Zhongxiang Dai, Cheston Tan, Bryan Kian Hsiang Low, and Roger Wattenhofer. Fedhql: Federated heterogeneous q-learning. *arXiv preprint arXiv:2301.11135*, 2023.

[Fang and Qian, 2021] Haokun Fang and Quan Qian. Privacy preserving machine learning with homomorphic encryption and federated learning. *Future Internet*, 13(4):94, 2021.

[Ferns *et al.*, 2011] Norm Ferns, Prakash Panangaden, and Doina Precup. Bisimulation metrics for continuous markov decision processes. *SIAM Journal on Computing*, 40(6):1662–1714, 2011.

[Geyer *et al.*, 2017] Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

[Haarnoja *et al.*, 2018a] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

[Haarnoja *et al.*, 2018b] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

[Jin *et al.*, 2022] Hao Jin, Yang Peng, Wenhao Yang, Shusen Wang, and Zhihua Zhang. Federated reinforcement learning with environment heterogeneity. In *International Conference on Artificial Intelligence and Statistics*, pages 18–37. PMLR, 2022.

[Kemertas and Aumentado-Armstrong, 2021] Mete Kemertas and Tristan Aumentado-Armstrong. Towards robust bisimulation metric learning. *Advances in Neural Information Processing Systems*, 34:4764–4777, 2021.

[Li *et al.*, 2020a] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37(3):50–60, 2020.

[Li *et al.*, 2020b] 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.

[Li *et al.*, 2021a] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10713–10722, 2021.

[Li *et al.*, 2021b] Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, Yuan Li, Xu Liu, and Bingsheng He. A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):3347–3366, 2021.

[Liang *et al.*, 2020] Paul Pu Liang, Terrance Liu, Liu Ziyin, Nicholas B Allen, Randy P Auerbach, David Brent, Ruslan Salakhutdinov, and Louis-Philippe Morency. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*, 2020.

[Liao *et al.*, 2023] Weijian Liao, Zongzhang Zhang, and Yang Yu. Policy-independent behavioral metric-based representation for deep reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37:8746–8754, 06 2023.

[Lyu *et al.*, 2020] Lingjuan Lyu, Han Yu, and Qiang Yang. Threats to federated learning: A survey, 2020.

[McMahan *et al.*, 2017] 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.

[Mohassel and Rindal, 2018] Payman Mohassel and Peter Rindal. Aby3: A mixed protocol framework for machine learning. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, pages 35–52, 2018.

[Mothukuri *et al.*, 2021] Viraaji Mothukuri, Reza M Parizi, Seyedamin Pouriyeh, Yan Huang, Ali Dehghantanha, and Gautam Srivastava. A survey on security and privacy of federated learning. *Future Generation Computer Systems*, 115:619–640, 2021.

[Park and Lim, 2022] Jaehyoung Park and Hyuk Lim. Privacy-preserving federated learning using homomorphic encryption. *Applied Sciences*, 12(2):734, 2022.

[Qi *et al.*, 2021] Jiaju Qi, Qihao Zhou, Lei Lei, and Kan Zheng. Federated reinforcement learning: Techniques, applications, and open challenges. *arXiv preprint arXiv:2108.11887*, 2021.

[Reddi *et al.*, 2020] 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.

[T Dinh *et al.*, 2020] Canh T Dinh, Nguyen Tran, and Josh Nguyen. Personalized federated learning with moreau envelopes. *Advances in neural information processing systems*, 33:21394–21405, 2020.

[Tan *et al.*, 2022] Yue Tan, Guodong Long, Jie Ma, Lu Liu, Tianyi Zhou, and Jing Jiang. Federated learning from pre-trained models: A contrastive learning approach. *Advances in neural information processing systems*, 35:19332–19344, 2022.

[Tang *et al.*, 2022] Fengxiao Tang, Yilin Yang, Xin Yao, Ming Zhao, and Nei Kato. Fesac: Federated learning-based soft actor-critic traffic offloading in space-air-ground integrated network. *arXiv preprint arXiv:2212.02075*, 2022.

[Tassa *et al.*, 2018] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

[Truex *et al.*, 2019] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM workshop on artificial intelligence and security*, pages 1–11, 2019.

[Wang *et al.*, 2020] Hao Wang, Zakhary Kaplan, Di Niu, and Baochun Li. Optimizing federated learning on non-iid data with reinforcement learning. In *IEEE INFOCOM 2020-IEEE conference on computer communications*, pages 1698–1707. IEEE, 2020.

[Wang *et al.*, 2024] Mengdi Wang, Anna Bodonhelyi, Efe Bozkir, and Enkelejda Kasneci. Turbosvm-fl: Boosting federated learning through svm aggregation for lazy clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 15546–15554, 2024.

[Wei *et al.*, 2020] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE transactions on information forensics and security*, 15:3454–3469, 2020.

[Yang *et al.*, 2019a] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol.*, 10(2), January 2019.

[Yang *et al.*, 2019b] Qiang Yang, Yang Liu, Yong Cheng, Yan Kang, Tianjian Chen, and Han Yu. Federated learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 13:1–207, 12 2019.

[Zhang *et al.*, 2020a] Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. *CoRR*, abs/2006.10742, 2020.

[Zhang *et al.*, 2020b] Chengliang Zhang, Suyi Li, Junzhe Xia, Wei Wang, Feng Yan, and Yang Liu. {BatchCrypt}: Efficient homomorphic encryption for {Cross-Silo} federated learning. In *2020 USENIX annual technical conference (USENIX ATC 20)*, pages 493–506, 2020.

[Zhang *et al.*, 2022] Xiaojin Zhang, Hanlin Gu, Lixin Fan, Kai Chen, and Qiang Yang. No free lunch theorem for security and utility in federated learning. *ACM Transactions on Intelligent Systems and Technology*, 14(1):1–35, 2022.

[Zhang *et al.*, 2023] Fengda Zhang, Kun Kuang, Long Chen, Zhaoyang You, Tao Shen, Jun Xiao, Yin Zhang, Chao Wu, Fei Wu, Yueting Zhuang, et al. Federated unsupervised representation learning. *Frontiers of Information Technology & Electronic Engineering*, 24(8):1181–1193, 2023.

[Zhao *et al.*, 2018] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

[Zhu *et al.*, 2019] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients, 2019.

[Zhuo *et al.*, 2019] Hankz Hankui Zhuo, Wenfeng Feng, Yufeng Lin, Qian Xu, and Qiang Yang. Federated deep reinforcement learning. *arXiv preprint arXiv:1901.08277*, 2019.