FedKL: Tackling Data Heterogeneity in Federated Reinforcement Learning by Penalizing KL Divergence

Zhijie Xie and Shenghui Song, Senior Member, IEEE

Abstract—One of the fundamental issues for Federated Learning (FL) is data heterogeneity, which causes accuracy degradation, slow convergence, and the communication bottleneck issue. Although the impact of data heterogeneity on supervised FL has been widely studied, the related investigation for Federated Reinforcement Learning (FRL) is still in its infancy. In this paper, we first define the type and level of data heterogeneity for FRL systems. By inspecting the connection between the global and local objective functions, we prove that local training can benefit the global objective, if the local update is properly penalized by the total variation (TV) distance between the local and global policies. A necessary condition for the global policy to be learnable from the local environments is also derived, which is directly related to the heterogeneity level. Based on the theoretical result, a Kullback-Leibler (KL) divergence based penalty is proposed to directly constrain the model outputs in the distribution space and the convergence proof of the proposed algorithm is also provided. By jointly penalizing the divergence of the local policy from the global policy with a global penalty and penalizing each iteration of the local training with a local penalty, the proposed method achieves a better trade-off between training speed (step size) and convergence. Experiment results on two popular Reinforcement Learning (RL) experiment platforms demonstrate the advantage of the proposed algorithm over existing methods in accelerating and stabilizing the training process with heterogeneous data.

Index Terms—Federated reinforcement learning, data heterogeneity, policy gradient.

I. INTRODUCTION

FEDERATED Learning (FL) was proposed as a distributed Machine Learning (ML) scheme where data from distributed devices are utilized for training without violating the user privacy. The idea has been successfully implemented (e.g. the Google Keyboard) [1] and attracted much research attention [2], [3]. One of the key challenges for FL is the communication bottleneck due to the frequent communication of big ML models. There are mainly two methods to mitigate the issue, i.e., reducing the communication workload per round and decreasing the number of communication rounds. To this end, model compression has been proposed to reduce the communication workload per round [4], [5], and another line of research tried to improve the convergence speed to minimize the number of communication rounds [6], [7], [8]. Unfortunately, data heterogeneity between different devices significantly slows the convergence down [9], [10].

As an important ML paradigm, Reinforcement Learning (RL) trains the policy of an agent in an iterative way by interacting with the environment. In each time step, the agent makes a decision to manipulate the environment and receives a reward for its action, where the objective of the training is to maximize the long term reward [11]. Due to the distributed data and computing power in large-scale applications such as autonomous driving, the training of RL algorithms under the FL framework is inevitable. Unfortunately, many challenges faced by supervised FL, e.g., the data heterogeneity and communication bottleneck, are still valid and even worse for FRL [12]. For example, centralized policy gradient methods already suffer from high variance which is detrimental to the convergence speed and training performance [13], [14], [15], and data heterogeneity imposes another layer of difficulty for the convergence of FRL.

Data heterogeneity in FL refers to the situation where the data collected by different devices are not independent and identically distributed (IID) [3], [16]. Despite its empirical success, FedAvg does not perform well on highly skewed non-IID data [10], [17], [18]. As a result, many innovative methods have been proposed to tackle the data heterogeneity issue [9]. Data based approaches aim to modify the data distribution to address the root of the problem. For example, a data-sharing strategy [17] was proposed to create a small but globally shared dataset to mitigate the impact of heterogeneous data. On the other hand, algorithm based approaches try to solve the problem by improving the local training algorithms. Along this line of research, FedProx [19] added a weighted proximal term to the local objective function to constrain the divergence of the local model from the global model. Finally, system based approaches focus on client clustering and system level optimization [20] to counterbalance the bias introduced by non-IID data and speed up convergence.
The heterogeneity issue of FRL is different from that of the supervised FL and the related study is very limited [12]. There are mainly two types of RL methods, namely, the action-value method and the policy gradient (PG) method [11], [12]. The first performs action selection according to the value function, while the second utilizes ML models to approximate the policy and update it by gradient-based algorithms. Data heterogeneity affects different RL methods in different ways, and thus requires different solutions. For the action-value method, Lifelong Reinforcement Learning (LFRL) [21] explored the combination of Q-learning and FL in real-world applications, where the knowledge learned in one particular environment is transferred to another. Federated RL (FedRL) [22] considered the scenario where some agents can only observe part of the environment and have no access to the reward. To handle this issue, agents maintain their individual Q-networks in addition to the global Q-network, and exchange their predicted Q values for cooperation. For policy gradient methods, Federated Transfer Reinforcement Learning (FTRL) [23] proposed an online transfer process to numerically align the scale of the observations and actions in different environments so that the policy learned in one environment is portable to another. Given that devices of the same type may have different dynamics, a FRL architecture, which adopts Proximal Policy Gradient (PPO) and Multi-Agent Deep Deterministic Policy Gradient (MADDPG) [24], [25], [26] to FRL in the Internet of Things (IoT) networks, was proposed in [27] and [28]. This FRL scheme alternates between sharing gradient and transferring mature models to other agents. For offline learning, data heterogeneity exists in different local datasets. To address this problem, Federated Dynamic Treatment Regime (FDTR) [29] proposed a federated policy optimization algorithm for offline RL. There are also works that aim to improve the communication efficiency with heterogeneous data [30], [31]. For instance, Federated Multi-Agent Reinforcement Learning (FMARL) [31] proposed to gradually decay the gradients during local training to avoid model divergence and improve communication efficiency. QAvg and PAvg [32] theoretically proved that federated Q-Learning and federated policy gradient can converge in the tabular case when agents’ environments are heterogeneous.

In this paper, we will investigate the data heterogeneity issue in FRL systems whose agents employ PG methods with an advantage estimator [15], [33]. For that purpose, we first classify the types of data heterogeneity in FRL and define the heterogeneity level. By analyzing the connection between the global and local objective functions, we prove that although it is possible to improve the global objective by updating the local policy according to the local environment, the local update must be properly penalized. Based on the theory, we propose an algorithm that utilizes a Kullback–Leibler (KL) divergence based term to tackle data heterogeneity and name it the FedKL algorithm. The proposed KL penalty will be referred to as the global penalty because it guarantees that the local training of one agent will not diverge too far away from the global policy. Note that this global penalty is different from the per-iteration penalty utilized in centralized RL [24]. We will investigate the benefits of adding the global penalty and how its interplay with the local per-iteration penalty helps mitigate the data heterogeneity issue in FRL. Experiment results show that FedKL is able to stabilize training and speed up convergence when compared with existing algorithms, including FedAvg, FedProx and FMARL.

The major contributions of this work can be summarized as follows:

- We classify the data heterogeneity of FRL into two types, due to different initial state distributions and different environment dynamics, respectively. Then, we define the level of heterogeneity, based on which a necessary condition for the local policy improvement to be beneficial for the global objective is derived.
- We prove that the improvement of the local policy on one agent can benefit the global objective, as long as the local policy update is properly penalized by the total variation (TV) distance between the local and global policies. Based on the theoretical result, we propose the FedKL algorithm and adopt several approximations to simplify the implementation of FedKL. Convergence proof of the proposed algorithm is also provided.
- Experiment results show that FedKL outperforms existing algorithm based solutions with faster convergence and better stabilization. Several interesting insights are also obtained: 1) The proposed heterogeneity level is very effective and can be utilized to predict the learning behavior; 2) The KL divergence based penalty is more effective than the penalty defined in the parameter space; and 3) The proposed global penalty enables a larger learning step size, which speeds up convergence without causing divergence.

II. PRELIMINARIES

A. Federated Learning

A typical FL system consists of one central server and \( N \) agents (clients/devices), where the data of the agents are jointly utilized to train a global model without violating their privacy. In particular, the global optimization problem can be formulated as [18]

\[
\min_{\theta} f(\theta) = \sum_{n=1}^{N} q_n f_n(\theta),
\]

where \( f_n(\theta) \) represents the local objective function, e.g. the loss function, of the \( n \)-th agent with respect to the model parameters \( \theta \). \( q_n \) is a weighting factor often set to \( q_n = l_n/L \), with \( l_n \) denoting the number of data points at the \( n \)-th agent and \( L = \sum_{n=1}^{N} l_n \). In each communication round, \( K \) out of the total \( N \) agents are selected and the central server broadcasts the global model to the selected agents. Then, each agent updates its local model utilizing the local data and uploads the trained model to the server for model aggregation. This process will repeat for some rounds until a preset learning objective is achieved.

B. Federated Reinforcement Learning

The system setup of FRL is similar, i.e., one central server federates the learning of \( N \) distributed agents. In the
Maximize its expected discounted reward over the long run.

State \( s \) is controlled by a stochastic policy \( \pi \) distributions and environment dynamics. An agent's behavior interacts with its environment to collect \( T \) time-steps of data, and optimizes its local policy according to the local objective with gradient ascent methods. At the end of each round, the training results will be uploaded to the server for aggregation. An example of FRL systems for autonomous driving is illustrated in Fig. 1.

We model the local training problem of each agent as a finite Markov Decision Process (MDP). Accordingly, the FRL system consists of \( N \) finite MDPs \( \{(S, \mu_n, A, P_n, \gamma) | n \in \{1, \ldots, N\} \} \), where \( S \) denotes a finite set of states, \( \mu_n \) represents the initial state distribution of the \( n \)-th MDP, \( A \) is a finite set of actions and \( \gamma \in (0, 1) \) is the discount factor. The dynamics function \( P_n(s', r | s, a) : S \times A \rightarrow S \times A \rightarrow \mathbb{R} \) represents the probability that the \( n \)-th MDP transits from state \( s \) to \( s' \) and obtains a reward of \( r \in \mathbb{R} \) after taking action \( a \), which characterizes the environment dynamics of the \( n \)-th MDP [11]. We can define the transition probability \( P_n(s', r | s, a) : S \times S \times A \rightarrow \mathbb{R} \) as \( P_n(s', r | s, a) = \sum_{a' \in A} P_n(s', r | s, a') \). Furthermore, based on the dynamics function, we can define the reward function \( R_n(s, a) : S \times A \rightarrow \mathbb{R} \) as \( R_n(s, a) = \sum_{s' \in S} \sum_{a' \in A} P_n(s', r | s, a') \), which gives the expected reward of the \( n \)-th MDP for taking action \( a \) in state \( s \). As a result, the \( n \)-th MDP \( M_n \) can be represented by a 5-tuple \( (S, \mu_n, A, P_n, \gamma) \) sharing the same state and action space with other agents, but with possibly different initial state distributions and environment dynamics. An agent's behavior is controlled by a stochastic policy \( \pi_n : S \times A \rightarrow [0, 1] \) which outputs the probability of taking an action \( a \) in a given state \( s \). Specifically, each agent aims to train a local policy to maximize its expected discounted reward over the long run

\[
\eta_n(\pi) = \mathbb{E}_{s_0 \sim \mu_n, a_t \sim \pi, s_{t+1} \sim P_n} \left[ \sum_{t=0}^{\infty} \gamma^t R_n(s_t, a_t) \right], \tag{2}
\]

where the notation \( \mathbb{E}_{s_0 \sim \mu_n, a_t \sim \pi, s_{t+1} \sim P_n} \) indicates that the reward is averaged over all states and actions according to the initial state distribution, the transition probability, and the policy. Accordingly, the optimization problem for FRL can be formulated as

\[
\max_{\pi} \eta(\pi) = \sum_{n=1}^{N} \eta_n(\pi). \tag{3}
\]

Note that \( \mu_n \) is added to the formulation in (2) to represent the different initial state distributions for different agents and will be discussed in detail later. The above formulation covers both IID and non-IID cases. In particular, the different MDPs, i.e., different initial state distributions and environment dynamics, represent the heterogeneous environments experienced by different agents (non-IID). All MDPs will be identical for the IID case [34].

C. Policy Gradient Methods

In this section, we will introduce the policy gradient (PG) method [15], which is taken by the agents to update their local policies in this paper. While action-value methods learn the values of actions and then select actions based on their estimated values, PG methods directly learn the decision-making policy.

To update the policy by gradient ascent methods, we need to estimate the policy gradient. For that purpose, we first define the state-value function and the action-value function. Specifically, the state-value function \( V_{\pi, P_n}(s) \) of a state \( s \) is the expected return when the agent starts from state \( s \) and follows policy \( \pi \) thereafter:

\[
V_{\pi, P_n}(s) = \mathbb{E}_{s_0, P_n} \left[ \sum_{t=0}^{\infty} \gamma^t R(s_{t+1}, a_{t+1}) | s_0 = s \right], \tag{4}
\]

where the expectation is performed over actions sampled from policy \( \pi \) and the next state is sampled from the transition.
probability $P_n$. Similarly, the action-value function $Q_{\pi, P_n}(s, a)$ of a state-action pair is the expected return when the agent starts from state $s$, takes the action $a$, and follows policy $\pi$ thereafter:

$$Q_{\pi, P_n}(s, a) = \mathbb{E}_{\pi, P_n} \left[ \sum_{t=0}^{\infty} \gamma^t R(s_{t+1}, a_{t+1}) | s_t = s, a_t = a \right].$$

By subtracting the state-value function from the action-value function, we can define the advantage function $A_{\pi, P_n}(s, a) = Q_{\pi, P_n}(s, a) - V_{\pi, P_n}(s)$, which measures the value of action $a$ in state $s$. The authors of [15] proved that the parameters $\theta$ of a differentiable policy $\pi(\theta)$ can be updated using the policy gradient

$$\hat{g} = \mathbb{E}_{\pi, P_n} \left[ \nabla_{\theta} \log \pi(a | s) A_{\pi, P_n}(s, a) \right],$$

where $\nabla_{\theta}$ denotes the derivative with respect to $\theta$.

However, the high variance of the gradient estimation often leads to large policy updates and causes instability. As a result, the step size must be limited. For that purpose, [35], [36] investigated the performance gap for taking two different policies $\pi$ and $\pi'$,

$$\eta_n(\pi') - \eta_n(\pi) = \mathbb{E}_{s \sim \rho_{\pi', \mu_n, P_n}} \left[ \mathbb{E}_{a \sim \pi'(s | a)} \left[ A_{\pi, P_n}(s, a) \right] \right],$$

where

$$\rho_{\pi, \mu_n, P_n}(s) = \sum_{t=0}^{\infty} \gamma^t \Pr(s_t = s | \pi, \mu_n, P_n),$$

denotes the unnormalized discounted visitation frequency.

Equation (6) implies that the reward can be increased by any policy update that improves the expected advantage. However, it is difficult to optimize the right-hand side (RHS) of (6) since $\rho_{\pi', \mu_n, P_n}$ is dependent on $\pi'$. Therefore, the following surrogate objective was proposed

$$L_{\pi, \mu_n, P_n}(\pi') = \eta_n(\pi) + K_{\pi, \mu_n, P_n}(\pi'),$$

where

$$K_{\pi, \mu_n, P_n}(\pi') = \mathbb{E}_{s \sim \rho_{\pi, \mu_n, P_n}} \left[ \mathbb{E}_{a \sim \pi'(s | a)} \left[ A_{\pi, P_n}(s, a) \right] \right]$$

denotes the policy advantage that measures the advantage of policy $\pi'$ over $\pi$. $L_{\pi, \mu_n, P_n}$ is an approximation to $\eta_n(\pi')$ but easier to optimize because it replaces the dependency of $\rho_{\pi', \mu_n, P_n}$ on $\pi'$ with that of $\rho_{\pi, \mu_n, P_n}$ on $\pi$.

Although the above step involves an approximation, it has been shown that a sufficiently small step that improves $L_{\pi, \mu_n, P_n}(\pi')$ will also improve $\eta_n(\pi')$. Specifically, the following policy improvement bound was derived for any policies $\pi$ and $\pi'$

$$\eta_n(\pi') \geq L_{\pi, \mu_n, P_n}(\pi') - c_n' D_{KL}^{\text{max}}(\pi, \pi'),$$

or equivalently

$$\eta_n(\pi') - \eta_n(\pi) \geq K_{\pi, \mu_n, P_n}(\pi') - c_n' D_{KL}^{\text{max}}(\pi, \pi'),$$

where $c_n' = \frac{4 \max_{s,a} |A_{\pi, P_n}(s, a)|}{(1-\gamma)^2}$ is a constant and $D_{KL}^{\text{max}}(\pi, \pi') = \max_{s} D_{KL}(\pi(s) || \pi'(s))$ denotes the maximum KL divergence between two policies $\pi$ and $\pi'$ among all states. As a result, one can optimize Equation (2) by optimizing the surrogate objective. A tighter bound for the policy improvement was later derived [37]

$$\eta_n(\pi') - \eta_n(\pi) \geq K_{\pi, \mu_n, P_n}(\pi') - c_n' D_{KL}^{\text{max}}(\pi, \pi'),$$

where $c_n' = \frac{2 \max_{s,a} |A_{\pi, P_n}(s, a)|}{(1-\gamma)^2}$ is upper bounded by $c_n'$ and $D_{TV}(\pi(\cdot | s) || \pi'(\cdot | s)) = \frac{1}{2} \sum_{s} |\pi(a | s) - \pi'(a | s)|$ denotes the TV distance between two discrete probability distributions $\pi(\cdot | s)$ and $\pi'(\cdot | s)$. For the continuous case, it is straightforward to replace the summation by integration. In this work, (12) is useful to derive the policy improvement bound of FRL.

### III. DATA HETEROGENEITY AND ITS IMPACT ON FRL

In this section, we will first investigate the types of data heterogeneity in FRL and define the heterogeneity level. Then, by studying the connection between the local and global objectives, we will investigate how the local update can help the global objective and derive a necessary condition for the global policy to be learnable.

#### A. Types of Data Heterogeneity in FRL

Given (7), the expected discounted reward for the $n$-th MDP $\mathcal{M}_n$, defined in (2), can be written as

$$\eta_n(\pi) = \sum_{s} \rho_{\pi, \mu_n, P_n}(s) \sum_{a} \pi(a | s) R_n(s, a).$$

It can be observed that the performance of a policy on $\mathcal{M}_n$ is jointly determined by its initial state distribution $\mu_n$ and the dynamics function $P_n$, both of which can be either stochastic or deterministic. As a result, agents with different initial state distributions and environment dynamics will end up with different policies, causing the heterogeneity issue. Therefore, we classify the data heterogeneity in FRL into two types:

1. Heterogeneity due to initial state distribution;
2. Heterogeneity due to environment dynamics.

For ease of illustration, we consider two extreme cases where only one type of heterogeneity exists. Consider two finite MDPs, $\mathcal{M}_1$ and $\mathcal{M}_2$, with the same environment dynamics, $P_1 = P_2$ but different initial state distributions, $\mu_1 \neq \mu_2$. This corresponds to the situation where two self-driving vehicles have the same car model and training routine, but tend to enter the same road from different entrances. Now, consider the other extreme case, where $\mu_1 = \mu_2$ but $P_1 \neq P_2$, i.e., the two MDPs have the same initial state distribution but different environment dynamics. This can be utilized to model the situation where two different cars always enter the same road at the same entrance. The two types of heterogeneity can occur simultaneously. But, the heterogeneity due to different initial state distributions is of more concern for episodic tasks whose on-policy distribution is different from the stationary distribution [11] and heavily relies on the initial state distribution.
B. Level of Heterogeneity

In the following, we define a measurement regarding the level of heterogeneity which will be utilized to investigate the connection between the global and local policy advantages. For ease of illustration, we define $D_{\pi,\mu_n,P_n}$ as a $|S| \times |S|$ diagonal matrix with $p_{\pi,\mu_n,P_n}(s_i)$ being the $i$-th diagonal entry, where $|S|$ denotes the cardinality of $S$. Similarly, we denote $A_{\pi}$ as a $|S| \times |A|$ matrix whose $(i,j)$th entry is $\pi(a_j|s_i)$, and $A_{\pi,\mu_n}$ as a $|S| \times |A|$ matrix whose $(i,j)$th entry is $A_{\pi,\mu_n}(s_i,a_j)$. With the above notation, we can rewrite the policy advantage in (9) as

$$A_{\pi,\mu_n,P_n}(\pi^t) = \sum_s \rho_{\pi,\mu_n,P_n}(s) \sum_a \pi^t(a|s) A_{\pi,\mu_n}(s,a)$$

$$= \text{Tr} \left( D_{\pi,\mu_n,P_n} A_{\pi,\mu_n} \Pi_{\pi}^T \right),$$

where $\text{Tr}(\cdot)$ denotes the trace operator. It can be observed that the advantage of taking a new policy $\pi^t$ on a specific agent $n$ is realized through the matrix product $D_{\pi,\mu_n,P_n} A_{\pi,\mu_n}$. Thus, we propose the following definition regarding the level of heterogeneity. Without loss of generality, we assume that all states are reachable, i.e., $\rho_{\pi,\mu_n,P_n}(s) > 0$ for all $s$ and $\pi$.

**Definition I: (Level of heterogeneity).** We define the level of heterogeneity of the $n$-th agent under policy $\pi$ by the Frobenius norm of the matrix

$$B_{\pi,\mu_n,P_n} = \sum_{k=1}^N q_k D_{\pi,\mu_n,P_n}^{-1} D_{\pi,\mu_n,P_n} A_{\pi,\mu_n} - A_{\pi,\mu_n}$$

as $\|B_{\pi,\mu_n,P_n}\|_F$.

Note that $B_{\pi,\mu_n,P_n}$ measures the deviation of the policy advantage of the $n$-th agent from the average policy advantage of all agents. Given the local policy advantage directly affects the policy gradient, its deviation from the average will lead to potential gradient divergence. For the IID case, $B_{\pi,\mu_n,P_n} = 0$ and thus $\|B_{\pi,\mu_n,P_n}\|_F = 0$ for arbitrary policy $\pi$. With non-IID data, $\|B_{\pi,\mu_n,P_n}\|_F$ will be positive and its magnitude indicates the level of heterogeneity of the $n$-th agent. Intuitively, if the level of heterogeneity of one agent is too high, i.e., the policy advantage calculated over its MDP is too different from that of the others, improving the global policy according to the local environment may not be beneficial for the whole system. In Corollary I, we will prove a necessary condition, with respect to $\|B_{\pi,\mu_n,P_n}\|_F$, for the global policy to be learnable from the $n$-th local environment.

C. Connection Between Global and Local Objectives

In FRL, the local objective $\eta_n(\pi)$ serves as a proxy of the global objective $\eta(\pi)$. Specifically, each agent tries to optimize its local objective, but the global objective is the actual performance measure. Given the global policy $\pi^t$ in the $t$-th training round, the $n$-th agent interacts with its environment and obtains a new local policy $\pi_{n}^{t+1}$. Without access to any information of other agents, the $n$-th agent can only update the local policy by optimizing the local objective $\eta_n(\pi)$. However, a good local performance does not necessarily lead to a good global performance, especially with data heterogeneity. The question we want to answer is how we can optimize the local policy to best improve the global performance. For that purpose, we start from the local improvement.

It follows from (12) that

$$\eta_n(\pi_{n}^{t+1}) - \eta_n(\pi^t) \geq \mathcal{A}_{\pi,\mu_n,P_n}(\pi_{n}^{t+1}) - \mathcal{A}_{\pi,\mu_n,P_n}(\pi^t)$$

for all agents.

Remark II: Note that Corollary I includes the IID case as a special case. In particular, without data heterogeneity, the LHS of (18) will be zero. Thus, the necessary condition holds as long as the RHS is greater than zero, which indicates that it is
much easier for the local training to help the global training. This makes sense because for the IID case, all agents have the same MDP, thus any improvement for one agent will probably be beneficial for all agents.

However, the average policy advantage is not a direct measure of the final performance, as it is a proxy for the improvement of the global objective. A more useful result should directly measure the improvement in terms of the improvement of the global objective. A more useful result measure of the final performance, as it is a proxy for the

Corollary II: Based on Theorem I, we have the following inequality showing the relation between the global objective and the local update:

\[ \eta(\pi_n^{t+1}) \geq \sum_{k=1}^{N} q_k \eta_k(\pi^t) + A_{t,\mu_n,P_n}(\pi_n^{t+1}) - (\alpha_{t,n} + \beta_{t,n})E_{s \sim \rho_n^{t+1}} [D_{TV}(\pi^t(s)\|\pi_n^{t+1}(s))] \\
- \delta_{t,n}D_{TV}(\pi^t, \pi_n^{t+1}), \tag{19} \]

where \( \delta_{t,n} = 2 \sum_{k=1}^{N} q_k \frac{4\epsilon_k}{1-\gamma} D_{TV}(\rho_{t,n}, \pi_n^{t+1}) \), \( \beta_{t,n} = \sum_{k=1}^{N} q_k \frac{4\epsilon_k}{1-\gamma} \), and \( \epsilon_k = \max_{s,a} |A_{t,n}(s,a)| \) denotes the maximum absolute value of the advantage function over all state and action pairs.

The proof is given in Appendix C. The LHS of (19) represents the performance of the n-th local policy \( \pi_n^{t+1} \) over all agents. The first term on the RHS of (19) is a constant, representing the return of the current policy, and the second term represents the policy advantage of \( \pi_n^{t+1} \) over \( \pi_n^t \) in the n-th agent. Corollary II indicates that it is possible to improve a local policy’s average performance on all agents by improving the local policy advantage. In particular, by penalizing the local update with the TV distance, weighted by a coefficient \( \alpha_{t,n} \) proportional to the heterogeneity level, we may improve the performance of a local policy over all environments. In fact, it can be proved that optimizing the RHS of (19) will effectively improve the expected discounted reward on all agents. Please refer to Appendix D for more details.

The last two terms in (19) are penalty terms representing the expected TV distance and the maximum TV distance between \( \pi_n^{t+1} \) and \( \pi_n^t \). The two penalties indicate that even if it is possible to improve the global performance by optimizing the local objective, the improvement only happens when the local policy \( \pi_n^{t+1} \) is close to the global policy \( \pi_n^t \). Given the first term on the RHS of (19) is a constant, Corollary II suggests the following objective for the local training

\[ h_n(\pi_n^{t+1}; \pi^t) = A_{t,\mu_n,P_n}(\pi_n^{t+1}) - (\alpha_{t,n} + \beta_{t,n})E_{s \sim \rho_n^{t+1}} [D_{TV}(\pi^t(s)\|\pi_n^{t+1}(s))] \\
- \delta_{t,n}D_{TV}(\pi^t, \pi_n^{t+1}). \tag{20} \]

Remark III: There are three penalty terms in (20), namely the \( \alpha_{t,n}, \beta_{t,n} \), and \( \delta_{t,n} \) related penalty terms, penalizing the difference between the local and global policies. Thus, all of them can be regarded as a global penalty. However, they come from different sources and were added for different purposes. The \( \alpha_{t,n} \) penalty occurred when investigating the connection between the local and global policy advantage, i.e.,

Theorem I, to tackle the data heterogeneity, while the \( \beta_{t,n} \) and \( \delta_{t,n} \) penalty came from (11) and (12) to constrain the policy update at one agent.

Remark IV: The IID case is covered by Corollary II. This special case occurs when all agents share the same initial state distribution and environment dynamics, i.e. the level of heterogeneity \( \|B_{\pi^t,M_n,P_n}\|_p = 0 \) and hence \( \alpha_{t,n} = 0 \) and \( \beta_{t,n} \) becomes the constant \( \epsilon \) in (10) for all agents. As a result, the \( \alpha_{t,n} \) and \( \delta_{t,n} \) penalty term will disappear. However, the \( \beta_{t,n} \) and \( \delta_{t,n} \) penalty terms are still valid. This indicates that, even with IID data, a global penalty term is needed, and data heterogeneity will require a higher global penalty.

Next, we show the convergence of policy updates based on (20). The theoretical analysis is motivated by the convergence analysis of Heterogeneous-Agent Trust Region Policy Optimization (HATPRO) [38]. Similar to [32], we focus our discussion on the tabular case, i.e. we assume the aggregation step can linearly combine local policies to obtain the global policy. A summary of this learning procedure is presented in Algorithm 1.

**Algorithm 1** Federated Policy Iteration With Monotonic Improvement Guarantee. Tabular Case

Require: \( \pi^0 \)
1: for round \( t = 0,1,\ldots \) do
2: Synchronize the global model \( \pi^t \) to every agents.
3: Compute the exact values of \( \beta_{t,n} \).
4: for agent \( k = 1,2,\ldots,N \) do
5: Compute the exact values of \( \alpha_{t,k} \) and \( \delta_{t,k} \).
6: Optimize \( \pi_{k}^{t+1} = \arg \max \pi \ [h_k(\pi; \pi^t)] \).
7: Upload \( \pi_{k}^{t+1} \) and \( q_k \) to the central server.
8: end for
9: The central server aggregates the \( \theta \)'s as \( \pi_{t+1}(a|s) \leftarrow \sum_{k=1}^{N} q_k \pi_{k}^{t+1}(a|s), \forall s,a. \)
10: end for

**Theorem II:** Algorithm 1 is guaranteed to generate a monotonically improving sequence of policies, i.e. \( \eta(\pi^{t+1}) \geq \eta(\pi^t) \) for \( t \in \mathbb{N} \).

The proof is given in Appendix E. With Theorem II, the convergence of Algorithm 1 can be given by the following theorem.

**Theorem III:** A sequence of policies \( \pi^t \) \( t=0 \) generated by Algorithm 1 is convergent.

The proof is given in Appendix F. When policies are parameterized with tabular methods, Theorem II and III are exact. For non-linear parameterization, the algorithm can be similarly implemented, but the error term introduced by the aggregation step should be considered.

**IV. THE PROPOSED ALGORITHM: FEDKL**

Directly optimizing the objective in (20) is difficult due to several issues. First, it is impractical to compute the last terms of (20) by traversing all states. Moreover, despite it’s theoretical importance, the TV distance does not have a closed-form expression by far. Second, the advantage function has to be estimated by sampling. Third, one agent cannot access
information of other agents, making it impossible to determine the coefficients in (20). In the following, we tackle these issues one by one.

A. TV Distance

To handle the first issue, we replace the maximum TV distance over all states $D_{TV}(\pi^t, \pi_{n}^{t+1})$ by its average value $\mathbb{E}_{s \sim \rho_{s,t,n}, \mu_{n}, P_{n}} [D_{TV}(\pi^t(s)||\pi_{n}^{t+1}(s))]$. It has been shown that this heuristic approximation has similar empirical performance as the one with the maximum divergence [35]. Furthermore, we replace the TV distance in (20) by the KL divergence based on Pinsker’s inequality [39], i.e., $D_{TV}(p||q) \leq \sqrt{2D_{KL}(p||q)}$, and obtain

$$h_n(\pi_{n}^{t+1} || \pi^t) \geq A_{\pi_{n}, \mu_{n}, P_{n}}(\pi_{n}^{t+1}) - (\alpha_{\pi_{n}, \mu_{n}, \beta_{\pi_{n}} + \delta_{\pi_{n}, \mu_{n}}})$$

$$= \mathbb{E}_{s \sim \rho_{s,t,n}, \mu_{n}, P_{n}} \left[ \frac{1}{2} D_{KL}(\pi^t(s)||\pi_{n}^{t+1}(s)) \right] .$$

(21)

B. Approximation of the Policy Advantage and Penalty Terms

To optimize (21), we need to calculate the policy advantage and the penalty terms. To this end, we assume all agents adopt the training algorithm proposed by [24]. Next, we show how we can approximate (21) by Monte Carlo simulation. In each training round, the agent will perform $I$ iterations of local training. Let $\pi_{n}^{t+1}$ denote the local policy of the $n$-th agent after $i$ iterations of local training with $i = 0, \ldots, I$, where $\pi_{n,0} = \pi^t$ and $\pi_{n,i+1} = \pi_{n,i}^{t+1}$. Thus, the objective function for the $i$-th iteration of the $n$-th agent to obtain $\pi_n^{t+1}$ from $\pi_{n,i}^{t+1}$ can be given by

$$h_n(\pi_{n,i}^{t+1} || \pi^t) = \mathbb{E}_{s \sim \rho_{s,t,n,i}, \mu_{n}, P_{n}} \left[ \frac{1}{2} D_{KL}(\pi^t(s)||\pi_{n,i}^{t+1}(s)) \right] \cdot$$

$$-c_1 1 \left( \frac{1}{2} D_{KL}(\pi^t(s)||\pi_{n,i}^{t+1}(s)) \right)$$

$$+ c_2 D_{KL}(\pi_{n,i+1}^{t+1}(s)||\pi_{n,i}^{t+1}(s)) ,$$

(22)

where $c_1$ is used to approximate $(\alpha_{\pi_{n}, \mu_{n}} + \beta_{\pi_{n}} + \delta_{\pi_{n}, \mu_{n}})$, the last term is the KL penalty induced by the local training, and $c_2$ is used to approximate $c_i$ defined in (10). Here, $\omega(s, a) = \frac{\rho_{s,t,n,i}(s)}{\rho_{s,t,n,i}^{t+1}(s)} \pi_{n,i}^{t+1}(a|s)$ is the importance sampling ratio where we estimate $A_{\pi_{n}, \mu_{n}, P_{n}}(\pi_{n,i}^{t+1})$ by states and actions sampled from $\rho_{s,t,n,i}, \mu_{n}, P_{n}$ and $\pi_{n,i}^{t+1}$. The expected value of the first term in (22) is an unbiased estimation of the policy advantage $A_{\pi_{n}, \mu_{n}, P_{n}}(\pi_{n,i}^{t+1})$. However, it is difficult and inefficient to compute $\omega(s, a)$ directly. Therefore, we ignore the difference in the discounted visitation frequency and use $\omega(s, a) = \frac{\pi_{n,i}^{t+1}(a|s)}{\pi_{n,i+1}^{t+1}(a|s)}$, which yields a slightly biased estimation of the policy advantage but with smaller variance.

In the following discussion, we will refer to the second and third terms of (22) as the global and local penalties, respectively. In particular, the second term penalizes the divergence of one agent’s policy from the global policy and the third term penalizes the policy outputs in two consecutive iterations.

Given both the two penalties are related to KL divergence, we will refer to the proposed algorithm as FedKL.

C. Adaptive Penalty Coefficients

In FRL, the initial state distribution and dynamics function of one agent is not known to others. Thus, the penalty coefficients $c_1$ and $c_2$ cannot be known locally. In this paper, we adopt the adaptive method in [24] to determine the coefficients. Specifically, we introduce two hyper-parameters, $d_{local}$ and $d_{global}$, as the target local and global KL divergence. Then, $c_1$ and $c_2$ will be adjusted in an adaptive manner so that the target $d_{local}$ and $d_{global}$ are achieved in each policy update. After initializing $c_1$ and $c_2$, we perform a three-phase update in each policy update as follows:

- Perform policy update using E epochs of minibatch gradient ascent to optimize $h_n(\pi_{n,i}^{t+1} || \pi^t, \pi_{n,i}^{t+1})$ in (22).

- Compute

$$d = \mathbb{E}_{s \sim \rho_{s,t,n,i+1}^{t+1}, \mu_{n}, P_{n}} \left[ D_{KL}(\pi_{n,i+1}^{t+1}(s)||\pi_{n,i}^{t+1}(s)) \right]$$

$$- \text{If } d < d_{local}/1.1, c_2 \leftarrow c_2/2$$

$$- \text{If } d > d_{local} \times 1.1, c_2 \leftarrow c_2 \times 2$$

- Compute

$$d = \sqrt{2} D_{KL}(\pi^t(s)||\pi_{n,i}^{t+1}(s))$$

$$- \text{If } d < d_{global}/1.1, c_2 \leftarrow c_2/2$$

$$- \text{If } d > d_{global} \times 1.1, c_2 \leftarrow c_2 \times 2$$

Specifically, in each training round, FedKL performs several iterations of SGD update. After each iteration, FedKL updates its estimation of $c_1$ and $c_2$ for the next iteration so that the target local and global KL divergence are achieved.

Remark V: A larger target KL divergence ($d_{global}$ and $d_{local}$) will lead to a smaller penalty coefficient on the KL divergence ($c_1$ and $c_2$). Given the KL divergence between two policies before and after training represents the learning step size, the target KL divergence can be regarded as a target learning step size. Thus, $d_{global}$ and $d_{local}$ can help maintain a constant step size for each training round and iteration, respectively.

The heuristics for tuning $d_{local}$ and $d_{global}$ will be discussed in the experiments section. The training procedure of FedKL is illustrated in Algorithm 2.

V. EXPERIMENTS

In the experiments, we will compare the proposed FedKL with other algorithm-level solutions for tackling the

![Fig. 2. Reacher's field splitting. Sub-fields in grey are reachable by the robotic arm. Sub-environments are created based on these sub-fields.](image-url)
Algorithm 2 FedKL Using Adaptive Penalty Coefficients

Require: $K, I, T, E, \pi^0$
1: for round $t = 0,1, \ldots$ do
2: Selects a subset of $K$ agents according to some criteria.
3: Synchronize the global model to every selected agents.
4: for agent $k = 1, 2, \ldots, K$ do
5: for iteration $i = 1, 2, \ldots, I$ do
6: Run policy $\pi_{k,i}^{t+1}$ in environment for $T$ timesteps.
7: Optimize $h_k(\pi_{k,i}^{t+1}; \pi^t, \pi_{n,i-1}^{t+1})$ for $E$ epochs to obtain $\pi_{k,i}^{t+1}$.
8: end for
9: Upload $\theta_{k}^{t+1}$ and $l_k$ to the central server.
10: end for
11: The central server aggregates the $\theta$’s as $\theta^{t+1} = \sum_{k=1}^{K} l_k \theta_{k}^{t+1}$.
12: end for

heterogeneity issue, including FedAvg, FedProx and FMARL. In particular, agents of the FedAvg system will perform local training utilizing the algorithm proposed in [24], with the local penalty term. Agents of the FedProx system will deploy the same algorithm as FedAvg and add the proximal term [19]. Agents of the FMARL system will deploy the same algorithm as FedAvg and utilize the gradient decay scheme in [31].

On the other hand, agents of the proposed FedKL system will implement the same algorithm as FedAvg, but with the global KL divergence penalty. All codes can be found at: https://github.com/ShiehShieh/FedKL.

A. Experiment Platforms

To the best of the authors’ knowledge, there are limited works focusing on the data heterogeneity issue in FRL, making it difficult to find benchmarks for comparison purposes. Therefore, we made several lightweight modifications to popular RL simulators to accommodate the FRL setting. There are two widely used simulation environments for evaluating the performance of RL controllers, namely the MuJoCo simulator [40], [41] and the Flow simulator [42]. In the following, we introduce the two simulators and the settings for FRL experiments.

1) Heterogeneous Robotic Tasks: In Experiment 1, we consider a robotic task in the Reacher-v2 environment provided by OpenAI Gym [41]. As shown in Fig. 2a, the agent actuates a robotic arm to move the end-effector to reach a target, which is denoted by a red sphere and spawn at a random position in a $0.4 \times 0.4$ field. The environment consists of a 11-dimensional state space, a 2-dimensional action space, and a reward function that determines the reward based on the norm of the action vector as well as the distance between the end-effector and the target.

To simulate the heterogeneity due to different initial state distributions, we split the field into $Q$ sub-fields and create $Q$ sub-environments. For each sub-environment, we assume the target will always be placed in one of the $Q$ sub-fields. Note that this corresponds to an extreme setup with different initial state distributions. The case with $Q = 60$ is illustrated in Fig. 2b. To simulate the heterogeneity due to different environment dynamics, we add noise to each action as in [28]. Specifically, we add noise to distort all actions such that different devices have different transition probabilities. Let $m_{noise}$ and $\sigma_{noise}$ denote the mean and variance of the added noise. We assume $m_{noise} = 0$ and pick different variances to represent different levels of heterogeneity.

2) Traffic Control by Isolated RL Agents: In Experiment 2, we consider the traffic control in the “figure eight” network [42]. As shown in Fig. 3, there are in total 14 vehicles on the road with half of them controlled by RL agents and the rest controlled by the Intelligent Driver Model (IDM) controller provided by the SUMO library. The IDM controller can be regarded as the human-driver model. This experiment was designed for mixed-autonomy, where a centralized controller coordinates a set of connected autonomous vehicles (CAVs) to improve the system-level velocities.

Some modifications are required to run FRL experiments in the above network. Fortunately, there are a few modifications proposed by [31], where autonomous vehicles (AVs) are not connected by a centralized controller but isolated agents. Each RL-controlled vehicle is able to observe the position and speed of itself, and that of the two neighboring vehicles. As a result, each state is represented by a 6-dimensional vector. Since each agent is controlled by its local policy, every action is a scalar specifying the acceleration for the current time-step. Further assume that the average speed of all vehicles is accessible to all AVs.

By default, AVs and human-driven vehicles are positioned alternatively, as shown in Fig. 3a. If we utilize ‘h’ and ‘r’ to represent the human-driven vehicle and AVs, respectively, the default placement can be written as $[h, r, h, r, h, r, h, r, h, r, h, r, h, r, r]$. To simulate the heterogeneity due to different initial state distributions, we fix the starting position of all AVs such that everyone starts from the same position in each run. Given the AVs have access to the information about the two neighboring vehicles, we may create different environment dynamics by adjusting the placement pattern. For example, the car placement $[h, r, h, h, r, h, r, h, h, r, r, r]$, as shown in Fig. 3b, will lead to different environment dynamics.

Fig. 3. Vehicle Placement Schema. Red cars are human-driven vehicles. White cars are RL-controlled vehicles.

| Algorithm 2 FedKL Using Adaptive Penalty Coefficients |
|------------------------------------------------------|
| Require: $K, I, T, E, \pi^0$                        |
| 1: for round $t = 0,1, \ldots$ do                    |
| 2: Selects a subset of $K$ agents according to some criteria. |
| 3: Synchronize the global model to every selected agents. |
| 4: for agent $k = 1, 2, \ldots, K$ do               |
| 5: for iteration $i = 1, 2, \ldots, I$ do          |
| 6: Run policy $\pi_{k,i}^{t+1}$ in environment for $T$ timesteps. |
| 7: Optimize $h_k(\pi_{k,i}^{t+1}; \pi^t, \pi_{n,i-1}^{t+1})$ for $E$ epochs to obtain $\pi_{k,i}^{t+1}$. |
| 8: end for                                           |
| 9: Upload $\theta_{k}^{t+1}$ and $l_k$ to the central server. |
| 10: end for                                          |
| 11: The central server aggregates the $\theta$’s as $\theta^{t+1} = \sum_{k=1}^{K} l_k \theta_{k}^{t+1}$. |
| 12: end for                                          |
3) Implementation Details: For both experiments, we use neural networks to represent policies as in [35] and [42]. Specifically, fully-connected Multilayer Perceptrons (MLPs) with tanh non-linearity are utilized for two experiments with hidden layers (64, 64) and (100, 50, 25) [42], respectively. Generalized Advantage Estimation (GAE) [33] is used to estimate the advantage function. The value functions for two experiments are represented by MLPs with tanh non-linearity and hidden layers (64, 64) and (256, 256), respectively. The learning rate of one algorithm will remain the same for one simulation without decaying, but it will be tuned for different simulations. The hyperparameters $d_{local}, d_{global}$ and $\mu^1$ are carefully tuned so that they are near-optimal. In each training round, we set $K = 3, I = 20$, and $T = 2048$ for Experiment 1, and $K = 7, I = 50$, and $T = 1500$ for Experiment 2. For both experiments, we set the discount factor $\gamma$, batch size, and the discount factor for GAE, i.e., $\lambda$ in [33], as 0.99, 64, and 0.95, respectively.

B. Level of Heterogeneity and Its Impact

In this section, we will investigate the type and level of heterogeneity and how they affect the performance of FRL. For ease of illustration, we will use Experiment 1 to analyze the heterogeneity type and level. Detailed performance comparison between different algorithms and in-depth discussion regarding the key parameters will be given in the next section based on results from Experiment 2.

To illustrate the impact of data heterogeneity, we show in Fig. 4 the changes of $G = \|D_{\pi,\mu_n,\rho_n}A_{\pi,\rho_n}\|_F - \|D_{\pi,\mu_n,\rho_n}B_{\pi,\mu_n,\rho_n}\|_F$ with respect to the training round. Each term in (18) is rescaled by $D_{\pi,\mu_n,\rho_n}$ to overcome the numerical instability introduced by the inverse operation $D_{\pi,\mu_n,\rho_n}^{-1}$. Note that according to Corollary 1, the local training is useful only when $\|A_{\pi,\rho_n}\|_F > \|B_{\pi,\mu_n,\rho_n}\|_F$. Thus, we may regard $G$ as the potential of one agent to contribute to the global policy. In particular, if we treat $\|D_{\pi,\mu_n,\rho_n}A_{\pi,\rho_n}\|_F$ as the local training potential, then $G$ is the local potential penalized by the heterogeneity level, i.e., even if there is potential improvement from local training, it may not be helpful for the global policy when the heterogeneity level is high. We consider four different scenarios, including the IID case, the case with different initial state distributions ($Q = 60$), and two cases with different levels of environment dynamics, where the heterogeneity level is controlled by the variance of the action noises. It can be observed that $G$ decreases with training because training will reduce agents’ potential to further contribute to the global policy. On the other hand, comparing the two cases with different environment dynamics, we notice that higher level of heterogeneity will lead to smaller $G$, indicating that heterogeneity will limit the potential contribution of one agent. Note that the comparison between different types of heterogeneity may not be meaningful, as their local advantage functions may have different ranges. Fluctuations of the curves are due to limited number of simulations and matrix estimation error.

To further illustrate the impact of the heterogeneity level, we show in Fig. 5 the training performance with large action noise. The red (vertical) lines indicate the training rounds where $G < 0$ happens. It can be observed that, for those rounds, the global training will slow down, which agrees with Corollary 1, i.e., $G < 0$ indicates that the associate training is not helpful for the global policy. In Figs. 6 and 7, we show the performance comparison of FedAvg, FedProx, FMARL, and FedKL with different initial state distributions and different transition probabilities, respectively. It can be observed that with the same learning step size, FedKL can converge to stable performance, but
both FedAvg, FedProx, and FMARL may diverge or fluctuate. We can certainly reduce the step size for FedAvg, FedProx, and FMARL, but this will lead to slow learning speed. See the next section for more discussions.

C. Advantage of the KL-Based Global Penalty

In the following, we show the results of Experiment 2 where both types of heterogeneity are present. The best performance of each algorithm is shown in Fig. 8, where the averaged total reward over three runs is reported. Each run uses the same seed for the SUMO simulator but different seeds for the parameter initialization of the neural network. It can be observed that FedKL converges much faster than FedAvg, FedProx and FMARL. The faster convergence of FedKL indicates a larger step size. Then, a natural question is whether we can increase the step size of FedAvg, FedProx and FMARL to speed up their training. In Fig. 9, we show the results where the $d_{local}$ for FedAvg, FedProx and FMARL are increased. As a result, all of them converge at a similar speed as FedKL, but they diverge after reaching the maximum reward. The above results provide empirical evidence that directly penalizing the model output by KL divergence is more effective in handling the data heterogeneity than penalizing model divergence in the parameter space. Furthermore, applying a proper global KL penalty can prevent divergence from happening without loss of learning speed.

1) Why KL Penalty is Better?: It can be observed from Fig. 9 (after Round 120) that the proximal term of FedProx can slow down the divergence, but failed in avoiding it. This is because the proximal term $\|\theta^t_n - \theta^t\|$ constrains the learning in the model parameter space by the $L^2$-norm. Although the change in the model parameter space will lead to changes in the output distribution space (actions) and eventually influence the training objective, it is more effective to directly constrain the training in the distribution space [43], [44], which justifies the performance advantage of the proposed KL-divergence based penalty.

2) Why Global Penalty Helps?: We now investigate how $d_{global}$ helps speed up convergence and stabilize training. Assume there are $I$ iterations of local training in each round and let $d_i$ denote the policy update in the $i$-th iteration. We consider two cases, without and with the global penalty (target step size), respectively, as illustrated in Fig. 10. We use the red (dashed) and green arrows to represent the optimal optimization direction of the global and local objectives, respectively. Note that they are not on the same direction due to data heterogeneity. For the first case, we only apply the local penalty which constrains the step size for each iteration with $|d_i| \leq d_{local}$, i.e., the radius of the green circle in part (a) of Fig. 10. Let $d_{max}$ denote the highest tolerable step size in one training round, without causing training divergence. Thus, we need $d_{local} \leq d_{max}/I$, which is determined by considering the worst case scenario where all updates are on the same direction. However, the worst case happens with a very low chance. As a result, $d_{local}$ is pessimistically small for most training. Performance of FedAvg in Fig. 8 corresponds to the case that $d_{local}$ is small enough to avoid divergence, but causes slow convergence. Performance of FedAvg in Fig. 9

Fig. 7. Comparison of FedAvg, FedProx, FMARL, and FedKL on modified Reacher-V2 environments with different environment dynamics (action noise $\sim N(0, 0.4)$). For all algorithms, the learning rate is 0.001. For FedProx, $\mu = 0.02$. For FMARL, $\lambda = 0.9999$. For FedKL, $d_{global} = 0.6$. Results are averaged across three runs.

Fig. 8. Learning curves of FedAvg, FedProx, FMARL, and FedKL for Experiment 2. The best performance is shown for each algorithm. For all algorithms, the learning rate is 0.01. For FedAvg, $d_{local} = 0.0002$, $\mu = 0.001$. For FedProx, $d_{local} = 0.0002$, $\mu = 0.001$. For FedKL, $d_{local} = 0.0003$, $d_{global} = 0.15$. Results are averaged across three runs.

Fig. 9. The settings are the same as Fig. 8, but all algorithms have same local penalty, i.e. $d_{local} = 0.0003$.

Fig. 10. The effect of $d_{global}$ and $d_{local}$. (a) Without the global penalty, agents must use a small $d_{local}$ at the cost of slow learning. (b) With the global penalty, agents can explore a wider region (faster learning) without worrying about divergence.
corresponds to the case that \( d_{\text{local}} \) is large enough for fast training, but causes divergence.

Now consider the second case with \( d_{\text{global}} \), denoted by the radius of the red circle in part (b) of Fig. 10. Under such circumstances, we only need to guarantee \( \sum_{i=1}^{N} d_i \leq d_{\text{global}} \leq d_{\text{max}} \) and \( |d_i| \leq d_{\text{local}} \). As a result, \( d_{\text{local}} \) can be much larger than \( d_{\text{max}}/I \), leading to a much larger per-iteration step size, represented by the larger radius of the green circle in part (b) of Fig. 10. Thus, adding \( d_{\text{global}} \) enables a larger \( d_{\text{local}} \) and their joint efforts can achieve a better trade-off between training speed (step size) and convergence.

3) Heuristics for Tuning Target Step Size: The global KL divergence is a good indicator of model divergence and we usually observe a large global KL divergence followed by a drastic performance degradation. This observation suggests a heuristic way for tuning the global step-size target \( d_{\text{global}} \). For example, one can first run the algorithm withoutactivating the global KL penalty (set \( c_1 \) to 0). Then, one may find the optimal value of \( d_{\text{local}} \) and observe the average value of the KL divergence between \( \pi_t \) and \( \pi_{n+1} \) for certain number of rounds. It was observed that using a \( d_{\text{global}} \) slightly larger than the average, together with a \( d_{\text{local}} \) larger than the optimal value found in previous runs, can significantly speed up learning in the early stage of training.

D. Communication-Efficient FRL

There are many benefits for solving the data heterogeneity issue, including faster convergence, stable performance, etc. Faster convergence will also lead to fewer rounds of communication and mitigate the communication bottleneck issue. For example, to achieve a total reward of 350 in Experiment 2, it needs around 110 rounds of communication for both FedAvg, FedProx, and FMARL, but only 50 rounds for FedKL, representing a saving of more than 50% of the communication workload.

Local agents in FedKL use the Actor-Critic architecture [15], [25], where the actor refers to the policy \( \pi \) and the critic refers to the value function or the advantage function. One possible way to reduce communication workload is to keep the critic local since it is not needed for inference. We will investigate this idea in future works.

Another issue for large-scale FRL applications like autonomous driving is the model aggregation over many agents moving in a large area. A possible solution is the hierarchical FL scheme [45], where aggregations happen in two layers. First, local aggregations among agents within a small region are performed on a local server. Then, the aggregation results from multiple local servers are sent to a global server for global aggregation. The local and global aggregations can happen with different frequencies, which may significantly reduce the communication workload. Unfortunately, the convergence analysis for the hierarchical scheme with data heterogeneity is not available, and it will be interesting to investigate the performance of FedKL under such a hierarchical framework.

VI. CONCLUSION

In this paper, we investigated the data heterogeneity issue in FRL systems. We first classified the types of data heterogeneity in FRL and defined the heterogeneity level. It was proved that although optimizing the local policy may improve the global policy, the deviation of the local policy from the global one must be properly penalized. A necessary condition for the local policy improvement to be beneficial for the global objective was derived with respect to the heterogeneity level. Based on the theoretical result, a KL-divergence based global penalty term was proposed, which was shown to be able to enable a larger local step size without causing model divergence. The convergence proof of the proposed algorithm was also provided. Experiment results demonstrated the advantage of the proposed scheme in speeding up and stabilizing the training.

One conclusion we can draw from this work is that heterogeneity not only reduces the contribution of one agent’s local training to the global policy, but also increases the risk of divergence. It is thus important to balance the learning speed and convergence with a proper learning step size. Due to the lack of global information in the agents, a feasible solution is to maximize the learning speed and avoid divergence by jointly tuning the local and global penalty. A big challenge for implementing large-scale FRL is the model aggregation among many moving agents and it is critical to develop distributed aggregation schemes with heterogeneous data.

APPENDIX A

PROOF OF THEOREM I

Proof: Following [46], we define the “absolute” value of a matrix \( X \) as

\[
|X| = (X^T X)^{\frac{1}{2}}, \quad |X|^\dagger = (XX^\dagger)^{\frac{1}{2}},
\]

(23)

where \( X^\dagger \) denotes the conjugate transpose of \( X \). In other words, \( |X| \) can be regarded as the square-root of \( X^\dagger X \). For a real matrix \( X \), \( X^\dagger X \) is symmetric and positive semi-definite. Thus, there must be a unique square root \( |X| \) of \( X^T X \) that is real and positive semi-definite. For a given scalar \( x \), the above operator will degenerate to the absolute value operator \( |x| = (xx^\dagger)^{\frac{1}{2}} \).

To understand the impact of the local optimization on the global objective, we define the gap between the global and local policy advantage for taking the updated local policy of the \( n \)-th agent \( \pi_{n+1} \) as

\[
G(\pi_{n+1}^t) = \sum_{k=1}^{N} q_k \mathbb{A}_{\pi_t, \mu_k, P_n} \left( \pi_{n+1}^t \right) - \mathbb{A}_{\pi_t, \mu_n, P_n} \left( \pi_{n+1}^t \right).
\]

(24)

Utilizing the vector notation from (14), we can calculate the gap as

\[
G(\pi_{n+1}^t) = \text{Tr} \left( \sum_{k=1}^{N} q_k \mathbf{D}_{\pi_t, \mu_k, P_n} \mathbf{D}^{-1}_{\pi_t, \mu_n, P_n} \mathbf{D}_{\pi_t, \mu_k, P_n} \mathbf{A}_{\pi_t, P_n} \mathbf{D}^T_{\pi_{n+1}^t} \right)

(25)
\]

\[
= \text{Tr} \left( \sum_{k=1}^{N} q_k \mathbf{A}_{\pi_t, P_n} \mathbf{D}^T_{\pi_{n+1}^t} \mathbf{D}_{\pi_t, \mu_k, P_n} \mathbf{D}^{-1}_{\pi_t, \mu_n, P_n} \mathbf{D}_{\pi_t, \mu_k, P_n} \mathbf{A}_{\pi_t, P_k} \right)
\]
Thus, we can further obtain
\[ G(\pi^{t+1}_n - \pi^t_n) \leq \|B_{\pi^{t+1}_n, \mu_n} - B_{\pi^t_n, \mu_n}\|_F \cdot \sqrt{\sum_{s,a} |\rho_{\pi^{t+1}_n, \mu_n}(s) - \rho_{\pi^t_n, \mu_n}(s)|^2}. \] (37)

Given \( \sum_{n} \pi^t(a|s)A_{\pi^t, \mu_n}(s, a) = 0, \forall s \in S, \) we have
\[ \tilde{A}_{\pi^{t+1}_n, \mu_n} \leq \tilde{A}_{\pi^t_n, \mu_n} \]
\[ = \sum_{s} \rho_{\pi^{t+1}_n, \mu_n}(s) \sum_{a} \pi^{t+1}_n(a|s)A_{\pi^t_n, \mu_n}(s, a) \]
\[ - \sum_{s} \rho_{\pi^t_n, \mu_n}(s) \sum_{a} \pi^t(a|s)A_{\pi^t_n, \mu_n}(s, a) \]
\[ = \sum_{s} \rho_{\pi^{t+1}_n, \mu_n}(s) \sum_{a} \left[ \pi^{t+1}_n(a|s) - \pi^t(a|s) \right] A_{\pi^t_n, \mu_n}(s, a) \]
\[ = \tilde{A}_{\pi^{t+1}_n, \mu_n} \left( \left( \pi^{t+1}_n - \pi^t \right) \right). \] (38)

It thus follows from (24) that
\[ G(\pi^{t+1}_n) = G(\pi^{t+1}_n - \pi^t_n). \] (39)

By combining (39) and (37), we can obtain
\[ G(\pi^{t+1}_n) \leq \|B_{\pi^{t+1}_n, \mu_n} - B_{\pi^t_n, \mu_n}\|_F \cdot \sqrt{\sum_{s,a} |\rho_{\pi^{t+1}_n, \mu_n}(s) - \rho_{\pi^t_n, \mu_n}(s)|^2}. \] (40)

Based on (24), we can obtain
\[ \sum_{k=1}^{N} q_k \tilde{A}_{\pi^{t+1}_n, \mu_k, \rho_k} \leq \tilde{A}_{\pi^{t+1}_n, \mu_n} \leq \tilde{A}_{\pi^t_n, \mu_n} + \|B_{\pi^{t+1}_n, \mu_n} - B_{\pi^t_n, \mu_n}\|_F \cdot \sqrt{\sum_{s,a} |\rho_{\pi^{t+1}_n, \mu_n}(s) - \rho_{\pi^t_n, \mu_n}(s)|^2}. \] (41)

Given \( \sqrt{|a| + |b|} \leq \sqrt{|a|^2 + |b|^2} = |a| + |b|, \) we can obtain
\[ \sum_{k=1}^{N} q_k \tilde{A}_{\pi^{t+1}_n, \mu_k, \rho_k} \leq \tilde{A}_{\pi^{t+1}_n, \mu_n} \leq \tilde{A}_{\pi^t_n, \mu_n} + \|B_{\pi^{t+1}_n, \mu_n} - B_{\pi^t_n, \mu_n}\|_F \cdot \sqrt{\sum_{s,a} |\rho_{\pi^{t+1}_n, \mu_n}(s) - \rho_{\pi^t_n, \mu_n}(s)|^2}. \] (42)

Denote the TV distance between \( \pi(\cdot|s) \) and \( \pi'(\cdot|s) \) as \( D_{TV}(\pi(\cdot|s)||\pi'(\cdot|s)) = \frac{1}{2} \sum_{a} |\pi(a|s) - \pi'(a|s)|. \) We can finally obtain
\[ \sum_{k=1}^{N} q_k \tilde{A}_{\pi^{t+1}_n, \mu_k, \rho_k} \leq \tilde{A}_{\pi^{t+1}_n, \mu_n} \leq \tilde{A}_{\pi^t_n, \mu_n} \}
\[ - \alpha_{\pi^{t+1}_n, \mu_n} \sum_{a} \rho_{\pi^{t+1}_n, \mu_n}(s) \left[ D_{TV}(\pi(\cdot|s)||\pi^{t+1}_n(\cdot|s)) \right]. \] (43)

where \( \alpha_{\pi^{t+1}_n, \mu_n} = 2 \|B_{\pi^{t+1}_n, \mu_n} - B_{\pi^t_n, \mu_n}\|_F. \)

\(^2\)Note that we have used the notation \( (\pi' - \pi)(s, a) = \pi'(a|s) - \pi(a|s), \forall s \in S, a \in A. \)
APPENDIX B
PROOF OF COROLLARY I

Proof: It follows from (14) that
\[ A_{\pi^t, \mu_n, p_n}(\pi_n^{t+1}) = \text{Tr} \left( D_{\pi^t, \mu_n, p_n} A_{\pi^t, \mu_n, p_n} \Pi_{\pi_n^{t+1}}^T \right). \] (44)
By following the same procedure as from (27) to (43), we can prove
\[ A_{\pi^t, \mu_n, p_n}(\pi_n^{t+1}) \leq 2 \| A_{\pi^t, \mu_n, p_n} \|_F \cdot \mathbb{E}_{s \sim \rho^t, \mu_n, p_n} \left[ D_{TV}(\pi^t(\cdot|s)\| \pi_n^{t+1}(\cdot|s)) \right]. \] (45)
Thus, if \( \| A_{\pi^t, \mu_n, p_n} \|_F \leq \| B_{\pi^t, \mu_n, p_n} \|_F \), then
\[ A_{\pi^t, \mu_n, p_n}(\pi_n^{t+1}) \leq 2 \| B_{\pi^t, \mu_n, p_n} \|_F \cdot \mathbb{E}_{s \sim \rho^t, \mu_n, p_n} \left[ D_{TV}(\pi^t(\cdot|s)\| \pi_n^{t+1}(\cdot|s)) \right]. \] (46)
As a result, the right-hand side of (43) will be less or equal to zero, where the equality holds when \( \pi_n^{t+1} = \pi^t \). This indicates that there is no benefit from further updating \( \pi^t \) at the n-th agent, as there will be no improvement for the global objective. Thus, (18) is a necessary condition for the local update to be able to improve the global objective.

APPENDIX C
PROOF OF COROLLARY II

Proof: It follows from (3) that the global discounted reward at time \( t + 1 \) can be given by
\[ \eta(\pi_n^{t+1}) = \sum_{k=1}^{N} q_k \eta_k(\pi_n^{t+1}). \] (47)
By substituting (12) into (47), we can obtain
\[ \eta(\pi_n^{t+1}) \geq \sum_{k=1}^{N} q_k \left( L_{\pi^t, \mu_k, p_k}(\pi_n^{t+1}) - c_k \mathbb{E}_{s \sim \rho^t, \mu_k, p_k} \left[ D_{TV}(\pi^t(\cdot|s)\| \pi_n^{t+1}(\cdot|s)) \right] \right). \] (48)
By substituting (8) into (48) and bounding \( c_k \mathbb{E} \) from below by \( c_k^C \) from below by \( c_k^C \), we have
\[ \eta(\pi_n^{t+1}) \geq \sum_{k=1}^{N} q_k \left( \eta_k(\pi^t) + A_{\pi^t, \mu_k, p_k}(\pi_n^{t+1}) - c_k^C \mathbb{E}_{s \sim \rho^t, \mu_k, p_k} \left[ D_{TV}(\pi^t(\cdot|s)\| \pi_n^{t+1}(\cdot|s)) \right] \right), \] (49)
where \( c_k^C = \frac{4 \rho \gamma}{(1 - \gamma)^2} \), and \( c_k = \max_s |A_{\pi^t, \mu_k, p_k}(s,a)| \) denotes the maximum absolute value of the advantage function over all states and action pairs.
To remove the dependency on \( \rho^t, \mu_k, p_k \), we leverage the following inequality
\[ \mathbb{E}_{s \sim \rho^t, \mu_k, p_k} \left[ D_{TV}(\pi^t(\cdot|s)\| \pi_n^{t+1}(\cdot|s)) \right] \leq D_{TV}(\pi^t(\cdot|s)\| \pi_n^{t+1}(\cdot|s)) \leq D_{TV}(\pi^t(\cdot|s)\| \pi_n^{t+1}(\cdot|s)) \leq D_{TV}(\pi^t(\cdot|s)\| \pi_n^{t+1}(\cdot|s)). \] (50)
where \( D_{TV}^\max(\pi, \pi') = \max_s D_{TV}(\pi(\cdot|s)\| \pi'(\cdot|s)) \) denotes the maximum TV distance between two policies \( \pi \) and \( \pi' \) among all states. It follows from (50) that
\[ \mathbb{E}_{s \sim \rho^t, \mu_k, p_k} \left[ D_{TV}(\pi^t(\cdot|s)\| \pi_n^{t+1}(\cdot|s)) \right] \leq D_{TV}(\pi^t(\cdot|s)\| \pi_n^{t+1}(\cdot|s)) + 2D_{TV}^\max(\pi^t, \pi_n^{t+1})D_{TV}(\rho^t, \rho_n^{t+1}). \] (51)

Substituting (17) and (51) into (49) gives
\[ \eta(\pi_n^{t+1}) \geq \sum_{k=1}^{N} q_k \eta_k(\pi^t) + A_{\pi^t, \mu_k, p_k}(\pi_n^{t+1}) - (\alpha_{\pi^t, n} + \beta_{\pi^t}) \mathbb{E}_{s \sim \rho^t, \mu_k, p_k} \left[ D_{TV}(\pi^t(\cdot|s)\| \pi_n^{t+1}(\cdot|s)) \right] - \delta_{\pi^t, n} D_{TV}^\max(\pi^t, \pi_n^{t+1}), \] (52)
where \( \delta_{\pi^t, n} = \sum_{k=1}^{N} q_k \frac{4 \rho \gamma}{(1 - \gamma)^2} \).

APPENDIX D
EFFECTIVENESS OF IMPROVING THE LOCAL OBJECTIVE

Let \( g(\pi_n^{t+1}) \) denote the RHS of (19) as a function of \( \pi_n^{t+1} \). Thus, we have the following relation
\[ \eta(\pi_n^{t+1}) \geq g(\pi_n^{t+1}), \] (53)
\[ \eta(\pi^t) \geq g(\pi^t), \] (54)
\[ \eta(\pi_n^{t+1}) - \eta(\pi^t) \geq g(\pi_n^{t+1}) - g(\pi^t). \] (55)
The first inequality comes from (19). The equality in (54) can be checked by replacing \( \pi_n^{t+1} \) with \( \pi^t \) in the RHS of (19). Finally, subtracting (54) from (53) gives (55). The above conditions guarantee that we can improve the reward \( \eta(\pi_n^{t+1}) \) at each iteration \( t \) by improving the surrogate function \( g(\pi_n^{t+1}) \). In particular, the surrogate function \( g(\pi) \) minimizes \( \eta(\pi) \) with equality at \( \pi^t \) [35, 47].

APPENDIX E
PROOF OF THEOREM II

Lemma I: (Linearity of policy advantage \( A_\pi \) and surrogate objective \( L_\pi \)). Given arbitrary policies \( \pi, \tilde{\pi}, \) and \( \pi' \), we define
\[ \tilde{\pi}(s, a) = \gamma \tilde{\pi}(s, a) + (1 - \gamma) \pi'(s, a), \] \( \forall s \in S, a \in A, \gamma \leq 1 \). As the linear combination of policies \( \tilde{\pi} \) and \( \pi' \). Then, the policy advantage of \( \tilde{\pi} \) over \( \pi \) is the linear combination of individual policy advantages,
\[ A_{\pi}(\tilde{\pi}) = \gamma A_{\pi}(\tilde{\pi}) + (1 - \gamma) A_{\pi}(\pi'), \] (56)
\[ L_\pi(\tilde{\pi}) = \gamma L_\pi(\tilde{\pi}) + (1 - \gamma) L_\pi(\pi'). \] (57)
Proof: For any \( \pi, \tilde{\pi}, \pi', \) and \( 0 \leq \gamma \leq 1 \),
\[ L_{\pi}(\tilde{\pi}) = \eta(\pi) + \sum_s \rho(s) \sum_a \tilde{\pi}(a|s) A_\pi(s, a) + (1 - \gamma) \sum_s \rho(s) \sum_a \pi'(a|s) A_\pi(s, a) = \eta(\pi) + \gamma A_{\pi}(\tilde{\pi}) + (1 - \gamma) A_{\pi}(\pi'), \]
\[ = \gamma L_{\pi}(\tilde{\pi}) + (1 - \gamma) L_{\pi}(\pi'). \]
This proves (57), and (56) follows from the second equality.
Lemma II: In the tabular case, for all states, the TV distance between $\pi^{t+1}(|s)$ and $\pi^t(|s)$ is upper bounded by the expectation of the TV distance between local policies $\pi_k^{t+1}(|s)$ and $\pi_k^t(|s)$.

Proof: For all states $s$,

\[
D_{TV}(\pi^t(|s), \pi^{t+1}(|s)) = \frac{1}{2} \left\| \pi^t(|s) - \pi^{t+1}(|s) \right\|_1
\]

By (12) and bounding $c^\rho_n$ by $c^\rho_n$, we can obtain

\[
\eta(\pi^{t+1}) \geq \sum_{n=1}^{N} q_n \eta_n \left( \sum_{k=1}^{N} q_k \pi_k^{t+1} \right) \tag{58}
\]

By the linearity of advantage function, i.e. Lemma I, we have

\[
\eta(\pi^{t+1}) \geq \eta(\pi^t) + \sum_{n=1}^{N} q_n \left\{ \hat{A}_{\pi^t,\mu_n} p_n \left( \sum_{k=1}^{N} q_k \pi_k^{t+1} \right) - c^\rho_n \mathbb{E}_{s \sim \rho_{\pi_n, \mu_n}} [D_{TV}(\pi^t(s)||\pi^{t+1}(s))] \right\}. \tag{59}
\]

By Lemma II, we have

\[
\eta(\pi^{t+1}) \geq \eta(\pi^t) + \sum_{n=1}^{N} q_n \left\{ \sum_{k=1}^{N} q_k \hat{A}_{\pi^t,\mu_n} p_n (\pi_k^{t+1}) \right\}
\]

By exchanging the summation order, we have

\[
\eta(\pi^{t+1}) \geq \eta(\pi^t) + \sum_{n=1}^{N} q_n \mathbb{E}_{s \sim \rho_{\pi_n, \mu_n}} \left[ D_{TV}(\pi^t(s)||\pi^{t+1}(s)) \right].
\]

By following the same procedure as Appendix C, we have

\[
\eta(\pi^{t+1}) - \eta(\pi^t) \geq \sum_{n=1}^{N} q_n \left\{ \hat{A}_{\pi^t,\mu_n} p_n (\pi_k^{t+1}) - (\alpha_{\pi^t,k} + \beta_{\pi^t}) \mathbb{E}_{s \sim \rho_{\pi_n, \mu_n}} [D_{TV}(\pi^t(s)||\pi^{t+1}(s))] \right\}. \tag{61}
\]

This proves that the RHS of (61) serves as a lower bound for the performance difference between $\pi^{t+1}$ and $\pi^t$. Moreover, optimizing the RHS of (61) in each round can monotonically improve the performance since the RHS of (61) minorizes its LHS, where the equality holds when $\pi_k^{t+1} = \pi^t$, $\forall k = 1, \ldots, N$ (Appendix D). This completes the proof of Theorem II.

APPENDIX F

PROOF OF THEOREM III

This proof uses some lemmas and corollaries from [38].

Lemma III: (Continuity of $\rho_{\pi}$, [38, Lemma 4]). The (unnormalized) discounted visitation frequency $\rho_{\pi}$ is continuous on the policy space.

Lemma IV: (Continuity of $Q_{\pi}$, [38, Lemma 5]). $Q_{\pi}$ is Lipschitz-continuous on the policy space.

Corollary III: ( [38, Corollary 1]). Due to the continuity of $Q_{\pi}$, the following functions are Lipschitz-continuous on the policy space: The state value function $V_{\pi}$, the advantage function $A_{\pi}$, and the expected total reward $\eta(\pi)$.

Lemma V: (Continuity of policy advantage, [38, Lemma 6]). Let $\pi$ and $\hat{\pi}$ be policies. Fixing $\hat{\pi}$, the policy advantage $A_{\pi}(\hat{\pi})$ is continuous on the policy space.

Lemma VI: (Continuity of $\|B_{\pi,\mu_n,\mu_n}\|_{F}$). For every agent $k = 1, \ldots, N$, the level of heterogeneity is continuous on the policy space.

Proof: To simplify the proof, we expand the matrix norm $\|B_{\pi,\mu_n,\mu_n}\|_{F}$ as follows

\[
\|B_{\pi,\mu_n,\mu_n}\|_{F}^2 = \left\| \sum_{k=1}^{N} q_k D_{\pi,\mu_n,\mu_n}^{-1} D_{\pi,\mu_k,\mu_k} A_{\pi} p_n - A_{\pi} p_n \right\|_{F}^2
\]

The continuity of $\|B_{\pi,\mu_n,\mu_n}\|_{F}$ follows from the assumption that every state is reachable and the continuity of $\rho_{\pi}(s)$ and $A_{\pi}$.

Lemma VII: (Continuity of the objective). For every agent $k = 1, \ldots, N$, the local objective $h_k(\pi; \pi)$ in (20) is continuous on the policy space.

Proof: By Lemma V, $A_{\pi,\mu_n,\mu_n}(\pi')$ is continuous. By Corollary III and Lemma VI, $\alpha_{\pi,k}$, $\beta_{\pi}$ and $\delta_{\pi,k}$ are
This proves that, for any limit point of the policy sequence induced by Algorithm 1, \( \max_k [h_k(\pi; \pi^t)] = 0 \) for every agent \( k = 1, \ldots, N \), indicating that the sequence converges to a FedKL-stationary point. \( \square \)
