Recovering Reward Functions from Distributed Expert Demonstrations via Bi-level Maximum-likelihood Optimization

Guangyu Jiang, Shu Hong, Mahdi Imani, Nathaniel D. Bastian, and Tian Lan

***Abstract*—Inverse Reinforcement Learning (IRL) seeks to infer the latent reward function and the associated optimal policy from expert demonstrations. However, most current IRL methods assume centralized access to all trajectory data, which is impractical in real-world scenarios characterized by decentralized data sources and privacy concerns. To this end, this paper proposes a novel algorithm for federated maximum-likelihood IRL (F-ML-IRL) and provides a rigorous analysis of its convergence rate. The proposed F-ML-IRL leverages dual aggregation to update the shared global model and performs bi-level local updates: an upper-level learning task to optimize the parameterized reward function by maximizing the discounted likelihood of observing human expert trajectories under the current policy, and a lower-level learning task to find the optimal agent policy regarding the entropy-regularized discounted cumulative reward under the current reward function. We analyze the convergence rate of the proposed F-ML-IRL algorithm and show that the global model in F-ML-IRL converges to a stationary point for both the reward and policy parameters within finite time. That is, the log-distance between the recovered policy and the optimal policy, as well as the gradient of the likelihood objective, converges to zero. Evaluating our F-ML-IRL algorithm on high-dimensional robotic control tasks in MuJoCo, we show that it ensures convergence of the recovered reward in decentralized learning and outperforms centralized baselines due to its ability to utilize distributed data — attaining better recovered rewards than all baselines in 12 out of 20 tasks.**

***Index Terms*—Inverse Reinforcement Learning, Federated Learning, Bi-Level Optimization, Maximum Likelihood Estimation**

I. Introduction

Inverse Reinforcement Learning (IRL) seeks to recover the latent reward function underlying expert demonstrations, enabling subsequent agent training, teaming, and alignment with human intent. It formulates this as an inverse learning problem to model the preferences and goals of humans using observed behavior [1]. When a human expert's behavior is observed through demonstrations, IRL models the policy within a Markov Decision Process (MDP) framework and thereby replicates the optimal policy of the human expert [2]. The learned reward function can support various downstream tasks such as agent modeling, transfer learning, and fuzzy system control [1], [3], [4].

Despite significant advances in centralized IRL, with provably efficient algorithms such as Generative Adversarial Inverse Learning (GAIL) [5] and Maximum Likelihood IRL (ML-IRL) [6]–[8], practical deployment remains limited by privacy, communication, and scalability challenges. Human demonstration data are naturally distributed across personal devices, vehicles, and smart home systems, where privacy regulations and user expectations prohibit pooling raw trajectories on a central server. At the same time, transmitting large volumes of high-dimensional trajectory data can overwhelm bandwidth- constrained or intermittent networks with huge communication costs. Moreover, effectively leveraging the behavioral diversity of hundreds or even thousands of users demands a learning framework that scales gracefully without compromising each client's data ownership. These considerations collectively motivate a decentralized IRL framework that can recover a shared latent reward function from distributed human demonstrations while preserving data privacy.

To enable collaborative training with demonstration data from privacy-sensitive and decentralized clients, Federated Learning (FL) [9], [10] provides a promising solution. In FL, clients keep raw data locally and exchange only model updates, preserving privacy while leveraging distributed data sources. FL has proven effective for standard supervised and Reinforcement Learning (RL) settings [11], [12], including loss minimization [13], policy improvement [11], handling model heterogeneity [12], and communication-efficient optimization [10], [14]–[16]. However, IRL introduces a fundamentally different challenge: its bi-level optimization alternates between a lower level that recovers an optimal policy under a candidate reward function, and an upper level that adjusts reward parameters to maximize the likelihood of expert trajectories. These two coupled subproblems must be solved jointly, and naively combining them within FL's local-update and global-aggregate framework may fail to converge, since policy and reward parameters co-depend and interact across client updates. Therefore, it remains a significant open challenge to develop a decentralized learning framework for IRL, with provable convergence and time-complexity guarantees.

In this paper, we develop a novel framework, Federated Maximum-Likelihood IRL (F-ML-IRL), that enables decentralized recovery of a shared latent reward preserving data privacy. Our approach builds upon the ML-IRL formulation [6], [8] and extends it to a federated setting where human demonstration data are distributed across decentralized client devices. To address the intrinsic bi-level structure of IRL, each local training round of F-ML-IRL consists of two coupled learning tasks performed on each client. The *upper-level task* optimizes the parameterized reward function to maximize the discounted likelihood of observing local expert trajectories under the current policy, while the *lower-level task* solves

G. Jiang, S. Hong and T. Lan are with the Department of Electrical and Computer Engineering at George Washington University, M. Imani is with the Department of Electrical and Computer Engineering at Northeastern University, and N. Bastian is with the Department of Electrical Engineering & Computer Science at the United States Military Academy. Emails: {guangyu.jiang, shu.hong, tlan}@gwu.edu, m.imani@northeastern.edu, nathaniel.bastian@westpoint.edu.

an RL problem to find the agent's policy that maximizes the entropy-regularized discounted cumulative reward under the current reward function. To enable stable and synchronized updates across distributed clients, we propose a dual-aggregation method that aggregates both the reward parameters and action-value function models every T local training steps, which goes beyond traditional FL, which typically aggregates a single model parameter set. Further, we leverage Soft Q-learning [17] as the underlying RL method to ensure stability in the policy updates. Instead of fully solving the forward RL problem before updating the reward parameter, we perform one-step updates for both the recovered policy and the reward parameter alternately to improve efficiency. To the best of our knowledge, this is the first formulation and algorithmic solution for federated ML-IRL.

We conduct a rigorous convergence rate analysis of the proposed F-ML-IRL algorithm. Due to the tight coupling between the reward parameters and the recovered policy in IRL's bi-level optimization, the dual-aggregation method in our F-ML-IRL must be analyzed to understand its impact on convergence. By bounding the logarithmic distance between the estimated policy and the optimal policy by the distance between their corresponding Q-values, we control the variance introduced by local training by considering the time immediately after each global aggregation. Utilizing the γ-contraction property of soft Q-values, we establish the contraction property of the policy, and provide a convergence proof for the policy estimate. Moreover, we leverage the Lipschitz continuity of the reward parameter and the convergence of the policy estimate to show that the gradient of the global reward parameter converges to zero as the number of communications increases. These techniques enable us to show that F-ML-IRL's policy estimation and reward optimization both converge in finite time. The change in convergence speed due to the use of only decentralized clients and distributed data (rather than centralized learning) is characterized.

Our F-ML-IRL is implemented and evaluated on high-dimensional robotic control tasks in MuJoCo [18]. We compare its performance with several centralized learning baselines including Behavior Cloning (BC) [19], GAIL [5], and IRL methods like f-IRL [20] and ML-IRL [8]. We consider a non-iid data distribution, where clients have different local human expert demonstration data with varying performance levels. The baselines are evaluated using centralized data with two setups (i) a single client with medium-level demonstrations and (ii) a single client with a mixture of demonstrations of different levels. The results show that our F-ML-IRL could effectively leverage distributed data and client devices in learning, to achieve a similar or better recovered reward than the baselines, while meeting decentralization and data privacy restrictions. It is important to distinguish our formulation from existing centralized ML-IRL [8], which provides finite-time guarantees under pooled data with a single-loop update alternating policy improvement and likelihood ascent. By contrast, our setting requires federated training, where demonstration data remain local to clients due to privacy and communication constraints. This necessitates a bi-level federated program and a dual-aggregation scheme that synchronizes both Q-functions and reward parameters across clients. Our convergence analysis explicitly characterizes convergence rates in terms of the number of global rounds M and local steps T in an FL manner. These aspects highlight that our framework provides new guarantees and design elements beyond existing centralized approaches. The key contributions of this paper are summarized as follows:

- We propose a novel framework for federated maximum-likelihood IRL (F-ML-IRL). It enables decentralized IRL of a shared latent reward function, from distributed human demonstration data on decentralized client devices while preserving data privacy.
- To support the bi-level optimization structure in IRL – for jointly updating the optimal policy and the reward function estimate, the proposed F-ML-IRL algorithm leverages a dual-aggregation of the model parameters, which ensures convergence to optimal results.
- We prove the convergence and time-complexity of the proposed F-ML-IRL algorithm, with respect to local rounds T and aggregation steps M. We show that F-ML-IRL achieves convergence in finite time and will have faster convergence with smaller local rounds T.
- Our solution is evaluated on high-dimensional robotic control tasks in MuJoCo [18] and achieves a similar or higher recovered reward compared with Imitation Learning (IL) and IRL baselines that employ centralized learning.

The rest of the paper is organized as follows. Section II reviews related work and background. Section III introduces the proposed F-ML-IRL algorithm and Section IV presents its convergence analysis. Section V evaluates the proposed F-ML-IRL algorithm on high-dimensional robotic control tasks in MuJoCo [18]. Finally, Section VI concludes the paper.

II. Background and Related Works

A. Centralized IRL and IL

IRL aims to learn a latent reward function from human expert demonstration data, thereby freeing the forward RL problem from the requirement of specifying the reward function beforehand [21]. The recovered reward can be used to derive effective policies in RL [22] and IL [5]. Various formulations and solutions for the IRL problem have been explored. The Maximum Margin Planning algorithm frames the problem within a quadratic programming context [23]. Bayesian IRL models infer the posterior distribution of the reward function given a prior [24]. Probabilistic maximum entropy IRL methods favor stochastic policies using entropy regularization. In recent years, GAIL [5] has adopted a Generative Adversarial Networks [25] framework to recover the expert's policy, where a generator proposes new policies to confuse the discriminator, and the discriminator determines whether the trajectories following the generator's policy originate from human experts. In addition, recent works have studied IRL in continuous-time and structured multi-agent settings, such as efficient reward shaping for multiagent systems [26] and IRL in multiagent graphical games [27]. However, existing work has not considered the IRL problem with distributed data and decentralized clients, under data privacy.

ML-IRL has gained attention for its solid theoretical grounding and empirical performance [7], [8]. ML-IRL models the expert behavior within an MDP defined as $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \eta, r, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, $\mathcal{P}(s'|s,a)$ the transition probabilities, $\eta(\cdot)$ the initial state distribution, $r(s,a)$ the reward function, and $\gamma \in (0,1)$ the discount factor. The reward is parameterized as $r(s,a;\theta)$, where $\theta \in \mathbb{R}^d$ denotes the reward parameter vector. The goal is to recover the reward parameter θ such that the induced stochastic policy $\pi_{r_\theta}(a|s)$ explains the expert data $\mathcal{D} = \{\tau_j\}_{j=1}^K$, where each trajectory is a sequence $\tau_j = \{(s_t, a_t)\}_{t=0}^T$. The discounted log-likelihood of the data under the current reward-induced policy is:

$$\mathbb{E}_{\tau\sim\mathcal{D}}\left[\sum_{t\geq 0}\gamma^t\left(\log\pi_{r_\theta}(a_t|s_t) + \log\mathcal{P}(s_{t+1}|s_t,a_t)\right)\right]. \quad (1)$$

Since the transition dynamics $\mathcal{P}(\cdot)$ does not depend on the reward parameter θ, the optimization objective simplifies to:

$$l(\theta) = \mathbb{E}_{\tau\sim\mathcal{D}}\left[\sum_{t\geq 0}\gamma^t\log\pi_{r_\theta}(a_t|s_t)\right]. \quad (2)$$

ML-IRL maximizes $l(\theta)$ under the constraint that π_{r_θ} is the optimal policy under the reward r_θ, where the policy π_{r_θ} maximizes an entropy-regularized return:

$$\pi_{r_\theta} := \arg\max_\pi \mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t\left(r(s_t,a_t;\theta) + \mathcal{H}(\pi(\cdot|s_t))\right)\right], \quad (3)$$

with the entropy term defined as $\mathcal{H}(\pi(\cdot|s)) = -\sum_{a\in\mathcal{A}}\pi(a|s)\log\pi(a|s)$. The entropy regularization promotes stochasticity in the agent policy, improving exploration and generalization.

B. Federated Learning

Federated Learning (FL) is a decentralized training paradigm where multiple clients collaboratively learn a global model while keeping their raw data local. The typical FL objective is:

$$\min_{w\in\mathbb{R}^d} f(w) = \frac{1}{n}\sum_{i=1}^n f_i(w), \quad (4)$$

where each $f_i(w)$ is defined over local data $\mathcal{D}_i$. Early FL approaches like FedSGD [28] perform one local gradient step per communication round. FedAvg [9] improves communication efficiency by performing multiple local steps: $\omega^i \leftarrow \omega^i - \alpha\nabla f_i(\omega^i)$, where α is the learning rate. This reduces communication frequency while allowing clients to perform more local updates before aggregating at the server. FedAvg's convergence under non-i.i.d. data has been analyzed in [13]. Subsequently, methods such as FedProx [10], FedBN [14], MOON [15], and FedNova [16] have been developed to address non-i.i.d. data and accelerate the model training process [29]. Model-heterogeneous FL, where lightweight local models are derived from a shared global model, has been explored in [12]. However, existing FL methods assume a single-level objective. They could not be directly applied to the ML-IRL problem with decentralized clients, since ML-IRL requires a bi-level optimization involving both policy improvement and reward estimation using maximum likelihood. New algorithms need to be developed for decentralized ML-IRL with rigorous convergence analysis.

C. Federated IRL and IL

Few studies have addressed IL or IRL in a federated (client–server) setting. Federated Imitation Learning (FIL) enables robots or agents to train behavior cloning models by sharing model updates rather than raw demonstration data [30]. However, FIL remains a single-level supervised approach, which does not infer a reward function and therefore cannot leverage the generalization benefits of IRL. In the IRL domain, Distributed Inverse Constrained RL [31] considers a multi-agent setting in which agents share gradients to recover common reward and constraint functions. However, this method assumes synchronous, peer-to-peer communication over a fully connected network, which does not address federated scenarios with client-server orchestration. Moreover, they lack convergence guarantees when both reward and policy must be jointly learned under privacy and communication constraints. Several recent works have investigated federated or decentralized approaches to IRL and IL. A GAIL-powered asynchronous federated IRL framework has been proposed for 6G networks, highlighting adversarial learning in federated multi-agent environments [32]. Federated IRL has been explored for smart ICU decision support, demonstrating privacy-preserving inference in clinical domains [33]. Distributed IRL from streaming demonstrations considers continuously arriving, decentralized data [34], while decentralized adversarial IRL introduces decentralized adversarial IRL formulations for human–robot collaboration [35]. In addition, federated IL methods such as FedSkill [36] and FitLight [37] extend FL to demonstration-driven training for interpretable skill acquisition and traffic control. These approaches illustrate the growing importance of federated and decentralized frameworks for learning from demonstrations. However, they either rely on adversarial or supervised imitation objectives, focus on specific application domains, or lack rigorous convergence guarantees. Our work is distinct in proposing a maximum-likelihood federated IRL formulation with dual aggregation and finite-time analysis. To the best of our knowledge, no prior work has developed a federated algorithm with both reward inference (the IRL outer loop) and policy improvement (the IRL inner loop) under the federated setting.

D. Federated Bi-level Optimization

Bi-level optimization problems arise in hyperparameter tuning and meta-learning, where the outer objective depends on the solution to an inner learning problem. Federated bi-level optimization (FBO) methods such as FedNest [38] and FedDual [39] extend these problems to the federated setting by alternating between local inner-loop updates and outer-loop hypergradient aggregation. While these methods provide communication-efficient algorithms and theoretical guarantees, they typically assume supervised learning with

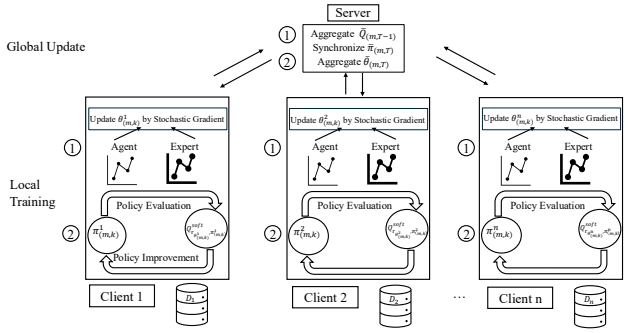


Fig. 1. Federated Maximum-Likelihood IRL (F-ML-IRL) framework. The objective is to recover a reward function r_θ from private expert trajectories $\mathcal{D}_1, \ldots, \mathcal{D}_n$ from n distributed clients. *Lower-level:* for a candidate reward r_θ, compute the entropy-regularized optimal policy π_{r_θ} via soft–Q evaluation and improvement. *Upper-level:* update θ to maximize the likelihood of expert data under π_{r_θ}. As the policy and reward updates depend on each other, F-ML-IRL uses a dual-aggregation strategy to synchronize both value/policy estimations and reward parameters across clients.

differentiable inner problems. In contrast, ML-IRL requires solving an RL problem in the inner loop (e.g., via soft Q-learning) and a likelihood-based reward optimization in the outer loop—posing challenges due to the non-differentiability of the inner problem and the coupling between reward and policy. Addressing this setting in FL presents unique challenges in convergence analysis and algorithm design.

III. Federated Maximum-Likelihood IRL

A. Problem Statement

We consider the federated inverse learning problem to recover a common reward function $r_\theta \colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ from n clients' private demonstration datasets $\{\mathcal{D}_i\}_{i=1}^n$. Each client holds a private dataset of expert demonstrations:

$$\mathcal{D}_i = \{\tau_j^i\}_{j=1}^K, \quad \tau_j^i = \{(s_t, a_t)\}_{t=0}^{T_i},$$

where each trajectory τ_j^i is generated by an unknown expert policy $\pi^{i*}(a|s)$, and T_i denotes the trajectory length for client i. We model the learning environment as an MDP $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \eta, \gamma)$ and employ the maximum-entropy IRL framework, as in the work of Ziebart *et al.* [6].Our goal is to learn a common reward function r_θ – parameterized by θ – from distributed data and to recover the corresponding optimal policy π_{r_θ}. The F-ML-IRL problem is formulated as the following bi-level optimization:

$$\begin{aligned} &\max_{\theta \in \mathbb{R}^d} \quad L(\theta) = \frac{1}{n} \sum_{i=1}^n l_i(\theta) \\ &\text{s.t. } \pi_{r_\theta} = \arg\max_\pi \mathbb{E}_\pi \left[\sum_{t=0}^\infty \gamma^t \left(r_\theta(s_t, a_t) + \mathcal{H}(\pi(\cdot|s_t)) \right) \right], \end{aligned} \tag{5}$$

where the local log-likelihood on client i is

$$l_i(\theta) = \mathbb{E}_{\tau \sim \mathcal{D}_i} \Big[\sum_{t=0}^\infty \gamma^t \log \pi_{r_\theta}(a_t|s_t) \Big]. \tag{6}$$

In the federated setting we consider, the goal is to recover a single shared reward function r_θ across all clients. Although demonstrations are distributed and may reflect different expertise levels, they are assumed to be guided by the same underlying reward structure. Consequently, the optimal stochastic policy π_{r_θ} induced by this reward provides a unified explanation for the heterogeneous expert data. This reflects the FL assumption that local data distributions are different views of the same latent environment.

Furthermore, the optimization problem in (5) has a bi-level structure. The *lower-level problem* solves for the entropy-regularized optimal policy π_{r_θ} given a candidate reward function r_θ, while the *upper-level problem* updates θ by maximizing the likelihood of the expert demonstrations under the induced policy π_{r_θ}. The explicit dependency of the outer optimization on the inner policy solution is what makes this formulation bi-level.

This inherent bi-level coupling in Problem (5) —where $L(\theta)$ requires solving the inner RL problem and the policy optimization depends on the current θ—renders standard single-level FL methods (e.g., FedAvg [9]) inapplicable. Instead, our F-ML-IRL algorithm employs a novel dual aggregation strategy that periodically synchronizes both the value/policy estimations and the reward parameters across all clients, ensuring consistent bi-level progress and provable convergence.

B. The Proposed F-ML-IRL Algorithm

To solve the federated bi-level IRL problem in (5), we present the F-ML-IRL algorithm with three modules: local policy improvement, local reward optimization, and global dual aggregation. We consider M global communication rounds and T local update steps within each round. $\pi^i_{(m,k)}$ and $\theta^i_{(m,k)}$ denote client i's policy and reward parameters at index (m, k), respectively, where $m = 0, \cdots, M-1$ is the global round index, and $k = 0, \cdots, T-1$ is the local update index.

At each local step, each client i first executes (in parallel) a policy update (on local data $\mathcal{D}_i$) through policy evaluation and improvement steps based on soft-Q learning to address the lower-level problem. Second, each client carries out a reward optimization, where the reward parameter gradient update is derived by contrasting sampled trajectories from both the expert policy and the current policy estimate. Next, after every T local steps and at the end of round m, we perform a dual aggregation of both the action-value function and the reward parameters to synchronize the local bi-level optimization of both policy and reward on decentralized clients. While our solution is inspired by FL, F-ML-IRL performs a dual aggregation with respect to the bi-level optimization in ML-IRL. The details of the algorithm are presented below. Its convergence rate is rigorously analyzed in this paper.

Our F-ML-IRL is illustrated in Fig. 1. We adopt the standard FL architecture, where a central server coordinates parameter aggregation. This is distinct from distributed IRL, which refers to parallelizing optimization across multiple machines that jointly process a single dataset. Different human expert demonstration data D_i are stored on different client devices. We perform local training for policy evaluation and improvement based on soft Q-learning to improve the local policy

$\pi^i_{(m,k)}$ under the current reward parameter $\theta^i_{(m,k)}$. We then sample trajectories from the current local policy and the human expert demonstration data $\mathcal{D}_i$, to update the reward parameter $\theta^i_{(m,k)}$. At local step k of round m, we use $Q^{\text{soft}}_{r_{\theta^i_{(m,k)}},\pi^i_{(m,k)}}(s,a)$ to denote the action-value function (i.e., Q-value) for action a and state s, with respect to the current agent policy estimate $\pi^i_{(m,k)}$ under the current reward parameter estimate $\theta^i_{(m,k)}$, on each client i. After every T steps of local training, we perform dual aggregation for the Q-value $\overline{Q}^{\text{soft}}_{(m,T-1)}$ and the reward parameter $\overline{\theta}_{(m,T)}$. To the best of our knowledge, this is the first paper to consider an ML-IRL problem in this FL context.

1) Local Bi-Level Updates: Each client i performs local bi-level updates in parallel at local iteration $k = 0, \cdots, T-1$, including policy evaluation and improvement, as well as reward optimization.

Local training for policy improvement.

Each client performs T local updates during each communication round $m = 0, \cdots, M-1$. At each local training iteration $k = 0, \cdots, T-1$ in communication round m, each local client starts with a shared model with parameters $\pi^i_{(m,0)}(\cdot|s)$ and $\theta^i_{(m,0)}$. During each local training round, we first evaluate the local policy $\pi^i_{(m,k)}(\cdot|s)$ by computing the Q-values $Q^i_{(m,k)}(\cdot,\cdot)$ under the fixed reward parameter θ^i for the i-th local client using the definitions of the soft value and Q functions in (7) and (8).

$$V^{\text{soft}}_{r_{\theta^i_{(m,k)}},\pi^i_{(m,k)}}(s) = \mathbb{E}_{\pi^i_{(m,k)},s_0=s}\Bigg[\sum_{t=0}^{\infty}\gamma^t\Big(r(s_t,a_t;\theta^i_{(m,k)}) + \mathcal{H}(\pi_{\theta^i_{(m,k)}}(\cdot|s_t))\Big)\Bigg], \tag{7}$$

$$Q^{\text{soft}}_{r_{\theta^i_{(m,k)}},\pi^i_{(m,k)}}(s,a) = r(s,a;\theta^i_{(m,k)}) + \gamma\mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}\Big[V^{\text{soft}}_{r_{\theta^i_{(m,k)}},\pi^i_{(m,k)}}(s')\Big]. \tag{8}$$

Then $\pi^i_{(m,k+1)}(\cdot|s)$ is updated according to the policy improvement step using soft Q-learning in (9). It does not assume an explicit policy function, but uses the Boltzmann distribution of the Q function, making the probability of choosing an action at some state s proportional to the exponential of the Q-value of this action-state pair.

$$\pi^i_{(m,k+1)}(a|s) \propto \exp(Q^{\text{soft}}_{r_{\theta^i_{(m,k)}},\pi^i_{(m,k)}}(s,a)), \forall s,a. \tag{9}$$

Local training for reward optimization.

For optimization toward the local reward parameter $\theta^i_{(m,k+1)}$, a stochastic gradient ascent method is adopted. The gradient of each local likelihood function $l_i(\theta)$ is given by (10), which is derived from Lemma 1 in [8].

$$\nabla l_i(\theta) = \mathbb{E}_{\tau_i\sim\mathcal{D}_i}\sum_{t\geq 0}\gamma^t\nabla_\theta r(s_t,a_t;\theta) - \mathbb{E}_{\tau_i\sim\pi_\theta}\sum_{t\geq 0}\gamma^t\nabla_\theta r(s_t,a_t;\theta). \tag{10}$$

We construct a stochastic estimator of the exact gradient $\nabla l_i(\theta^i_{(m,k)})$, approximating the optimal policy $\pi_{r_{\theta^i_{(m,k)}}}$ with the current policy $\pi^i_{(m,k+1)}$. Specifically, we sample one expert trajectory $\tau^{E_i}_{(m,k)} := \{(s_t,a_t)\}_{t\geq 0}$ from the local dataset $\mathcal{D}_i$ and one agent trajectory $\tau^{A_i}_{(m,k)} := \{(s_t,a_t)\}_{t\geq 0}$ from the current policy $\pi^i_{(m,k+1)}$. Then we use a stochastic estimate $g^i_{(m,k)}$ to approximate the exact gradient of the local likelihood objective function l_i for each local client in (11). The update of the reward (11) relies on both the local softmax policy $\pi^i_{(m,k+1)}$ through the agent trajectory $\tau^{A_i}_{(m,k)}$ and the local data $\mathcal{D}_i$ through the expert trajectory $\tau^{E_i}_{(m,k)}$.

$$g^i_{(m,k)} = h(\theta^i_{(m,k)};\tau^{E_i}_{(m,k)}) - h(\theta_{(m,k)};\tau^{A_i}_{(m,k)}). \tag{11}$$

where $h(\theta;\tau) = \sum_{t\geq 0}\gamma^t\nabla_\theta r(s_t,a_t;\theta)$. Finally, the local reward parameter $\theta^i_{(m,k)}$ is updated via gradient ascent as:

$$\theta^i_{(m,k+1)} = \theta^i_{(m,k)} + \alpha g^i_{(m,k)}. \tag{12}$$

where α is the learning rate for the reward parameter update. In practice, we can choose a diminishing step size $\alpha_{(m,k)} = \frac{\alpha_0}{(mT+k)^\sigma}$ for communication round m and local training iteration k, where $\alpha_0 > 0$ and $\sigma \in (0,1)$ are constants.

2) Server-Side Dual Aggregation: Every T local iterations, local Q-values and local reward parameters are communicated to the global server for aggregation, while the policy synchronization is performed based on the aggregated Q-values such that each local client has the same policy after the aggregation. We design the dual aggregation step after careful consideration. The reward update in (12) depends on how well the trajectories from the policy $\pi^i_{(m,k)}$ approximate the optimal policy $\pi_{r_{\theta^i_{(m,k)}}}$, while the policy $\pi^i_{(m,k)}$ relies on the Q-value update in (8). Therefore, our FL algorithm aims to improve the Q-value estimates for local clients by aggregating their Q-values (13).

$$\overline{Q}^{\text{soft}}_{(m,T-1)}(\cdot,\cdot) := \sum_{j=1}^{N} Q^j_{(m,T-2)}(\cdot,\cdot)/N. \tag{13}$$

We note that when the Q-values are represented by another network with parameter ψ, the aggregation of the Q-values will simply become aggregation of model parameters. The policy synchronization is automatically performed by policy improvement based on the aggregated Q-values and the resulting policy is sent to each local client for update such that each local client has the same policy after the Q aggregation in (14):

$$\overline{\pi}_{(m,T)}(\cdot|s) \propto \exp(\overline{Q}^{\text{soft}}_{(m,T-1)}(s,\cdot)), \forall s \in S. \tag{14}$$

Since ML-IRL requires a bi-level problem with respect to both the reward parameter and the recovered policy, we consider a dual aggregation that also applies to the reward parameter θ:

$$\overline{\theta}_{(m,T)} := \sum_{j=1}^{N}\theta^j_{(m,T-1)}/N. \tag{15}$$

After each dual aggregation, the global policy and reward parameters are sent to each local client as an initialization for future local training: $\pi^i_{(m,0)}(\cdot|s) = \overline{\pi}_{(m-1,T)}(\cdot|s)$ and $\theta^i_{(m,0)} = \overline{\theta}_{(m-1,T)}$ for all $i = 1,2,\ldots,N$.

In our formulation, we adopt uniform $1/N$ aggregation across clients for both Q-functions and reward parameters. This design is motivated by two considerations. First, prior work on maximum-likelihood IRL has shown that even a single trajectory can be sufficient for recovering a reward function that induces near-optimal policies; hence the effect of unequal sample sizes across clients is relatively minor in practice. Second, for analytical simplicity, we assume that each client holds the same number of trajectories, in which case uniform averaging and sample-size-weighted averaging coincide. Nevertheless, the framework can be extended to heterogeneous data sizes by weighting each client's contribution proportionally to its number of local trajectories. Our framework preserves data locality by keeping raw trajectories local and only communicating model quantities. While this reduces direct privacy risks compared to centralized IRL, it does not constitute a formal guarantee. Stronger federated privacy mechanisms (e.g., secure aggregation [40], differential privacy [41]) can be readily integrated into our protocol.

Algorithm 1 Federated Maximum Likelihood Inverse Reinforcement Learning (F-ML-IRL)

Input: Initialize reward parameter $\theta^i_{(0,0)}$ and policy $\pi^i_{(0,0)}$. Set aggregation period T, number of clients N, and local step size α.
for $m = 0, 1, \ldots, M-1$ **do**
  if $m > 0$ **then**
    Inherit $\pi^i_{(m,0)}(\cdot|s)$ and $\theta^i_{(m,0)}$ from last aggregation
  end if
  for $k = 0, \ldots, T-2$ **do**
    for $i = 1, 2, \ldots, N$ **do**
      Compute $Q^{\text{soft}}_{r_{\theta^i_{(m,k)}}, \pi^i_{(m,k)}}(\cdot, \cdot)$ using (8)
      Update $\pi^i_{(m,k+1)}(\cdot|s)$ based on (9)
      Sample expert trajectory $\tau^{E_i}_{(m,k)}$ from $\mathcal{D}_i$
      Sample trajectory $\tau^{A_i}_{(m,k)}$ from policy $\pi^i_{(m,k+1)}$
      Estimate gradient $g^i_{(m,k)}$ via (11)
      Update reward parameter $\theta^i_{(m,k+1)}$ via (12)
    end for
  end for
  Set $k = T-1$
  Aggregate $\overline{Q}^{\text{soft}}_{(m,k)}(\cdot, \cdot)$ by (13)
  Synchronize policies $\overline{\pi}_{(m,k+1)}(\cdot|s)$ using (14)
  Aggregate reward parameters $\overline{\theta}_{(m,k+1)}$ via (15)
end for

The entire process of the F-ML-IRL algorithm is summarized in Algorithm 1.

IV. Theoretical Analysis

In this section, we analyze the convergence of F-ML-IRL. We begin with key assumptions and auxiliary lemmas that will be used in the analysis, followed by our main convergence theorem and its detailed proof.

A. Assumptions

Assumption 1 (Ergodicity)**.** *For any policy π, assume the Markov chain with transition kernel $\mathcal{P}$ is irreducible and aperiodic under policy π. Then there exist constants $\kappa > 0$ and $\rho \in (0, 1)$ such that*

$$\sup_{s \in S} \|\mathbb{P}(s_t \in \cdot | s_0 = s, \pi) - \mu_\pi(\cdot)\|_{TV} \leq \kappa \rho^t, \quad \forall t \geq 0, \tag{16}$$

where $\|\cdot\|_{TV}$ is the total variation (TV) norm, and μ_π is the stationary state distribution under π.

Assumption 1 ensures a geometric mixing rate for the Markov chain, which is standard in the RL literature. For finite state spaces, this assumption is satisfied when the underlying Markov chain has a unique stationary distribution and mixes sufficiently fast.

Assumption 2 (Bounded Gradient and Lipschitz Property)**.** *For any $s \in \mathcal{S}$, $a \in \mathcal{A}$, and any reward parameter θ, the following conditions hold, where L_r and L_g are positive constants:*

$$\|\nabla_\theta r(s, a; \theta)\| \leq L_r, \quad \text{and} \tag{17}$$

$$\|\nabla_\theta r(s, a; \theta_1) - \nabla_\theta r(s, a; \theta_2)\| \leq L_g \|\theta_1 - \theta_2\|. \tag{18}$$

Assumption 2 ensures that the parameterized reward function has a bounded gradient and satisfies the Lipschitz smoothness condition, which are standard in optimization theory.

B. Important Lemmas

We first introduce two important lemmas that are used repeatedly in the convergence analysis.

Lemma 1. *Suppose Assumptions 1-2 hold. Given any reward parameters θ_1 and θ_2, the following results hold for any $s \in \mathcal{S}$ and $a \in \mathcal{A}$:*

$$\left|Q^{\text{soft}}_{r_{\theta_1}, \pi_{\theta_1}}(s, a) - Q^{\text{soft}}_{r_{\theta_2}, \pi_{\theta_2}}(s, a)\right| \leq L_q \|\theta_1 - \theta_2\|, \tag{19}$$

$$\|\nabla L(\theta_1) - \nabla L(\theta_2)\| \leq L_c \|\theta_1 - \theta_2\|, \tag{20}$$

where $Q^{\text{soft}}_{r_\theta, \pi_\theta}(\cdot, \cdot)$ denotes the soft Q-function under the reward function $r(\cdot, \cdot; \theta)$ and the policy π_θ.

Lemma 1 is from Lemma 2 in [8], where the positive constants L_q and L_c are also defined. The Lipschitz properties of the Q-value function and the gradient of the log-likelihood are essential for the convergence analysis, as they help control the distance between local and global models in the FL setting.

Lemma 2. *For any two policies $\pi(a|s)$ and $\pi'(a|s)$, the difference in their soft Q-values under some reward function r for a given state-action pair (s, a) is bounded as follows:*

$$\|Q^{\text{soft}}_{r,\pi} - Q^{\text{soft}}_{r,\pi'}\|_\infty \leq \frac{\gamma}{1-\gamma} \|\log(\pi) - \log(\pi')\|_\infty. \tag{21}$$

Controlling the distance between soft Q-values under different policies helps us analyze the optimality of the global policy with respect to the global reward parameter after aggregations.

Proof: Given the soft Q-function defined by the Bellman equation:

$$Q^{\text{soft}}_{r,\pi}(s, a) = \mathbb{E}_{s' \sim P(\cdot|s,a)} \Big[r(s, a) + \gamma \mathbb{E}_{a' \sim \pi(\cdot|s')} \big(Q^{\text{soft}}_{r,\pi}(s', a') - \log \pi(a'|s') \big) \Big]. \tag{22}$$

We can decompose the difference between soft Q-values under policies π and π' using the triangle inequality in (23):

$$\begin{aligned}&|Q^{\text{soft}}_{r,\pi}(s,a) - Q^{\text{soft}}_{r,\pi'}(s,a)| \\ &\leq \gamma\Bigg(\left|\mathbb{E}_{s'\sim P(\cdot|s,a)}\mathbb{E}_{a'\sim\pi(\cdot|s')}\left[Q^{\text{soft}}_{r,\pi}(s',a') - Q^{\text{soft}}_{r,\pi'}(s',a')\right]\right| \\ &\quad + \left|\mathbb{E}_{s'\sim P(\cdot|s,a)}\mathbb{E}_{a'\sim\pi(\cdot|s')}\left[\log\pi(a'|s') - \log\pi'(a'|s')\right]\right| \Bigg).\end{aligned} \tag{23}$$

We apply Jensen's inequality to the absolute value function and the second term in (23) involving the log policies:

$$\begin{aligned}&\left|\mathbb{E}_{s'\sim P(\cdot|s,a)}\mathbb{E}_{a'\sim\pi(\cdot|s')}\left[Q^{\text{soft}}_{r,\pi}(s',a') - Q^{\text{soft}}_{r,\pi'}(s',a')\right]\right| \\ &\leq \sup_{s',a'}\left|Q^{\text{soft}}_{r,\pi}(s',a') - Q^{\text{soft}}_{r,\pi'}(s',a')\right| \\ &\quad \left|\mathbb{E}_{s'\sim P(\cdot|s,a)}\mathbb{E}_{a'\sim\pi(\cdot|s')}\left[\log\pi(a'|s') - \log\pi'(a'|s')\right]\right| \\ &\leq \sup_{s',a'}\left|\log\pi(a'|s') - \log\pi'(a'|s')\right|.\end{aligned} \tag{24}$$

Substituting the inequalities from (24) into (38), we obtain:

$$\begin{aligned}&|Q^{\text{soft}}_{r,\pi}(s,a) - Q^{\text{soft}}_{r,\pi'}(s,a)| \\ &\leq \gamma \sup_{s',a'}|Q^{\text{soft}}_{r,\pi}(s',a') - Q^{\text{soft}}_{r,\pi'}(s',a')| \\ &\quad + \gamma \sup_{s',a'}|\log\pi(a'|s') - \log\pi'(a'|s')|.\end{aligned} \tag{25}$$

Since soft-Q values depend recursively on future rewards, we apply this bound recursively over n steps:

$$\begin{aligned}&|Q^{\text{soft}}_{r,\pi}(s_0=s,a_0=a) - Q^{\text{soft}}_{r,\pi'}(s_0=s,a_0=a)| \\ &\leq \gamma^n \sup_{s_n,a_n}|Q^{\text{soft}}_{r,\pi}(s_n,a_n) - Q^{\text{soft}}_{r,\pi'}(s_n,a_n)| \\ &\quad + \sum_{k=0}^{n-1}\gamma^k \sup_{s',a'}|\log\pi(a'|s') - \log\pi'(a'|s')|.\end{aligned} \tag{26}$$

As $n \to \infty$, the term $\gamma^n \sup_{s_n,a_n}|Q^{\text{soft}}_{r,\pi}(s_n,a_n) - Q^{\text{soft}}_{r,\pi'}(s_n,a_n)|$ tends to zero. The geometric series sums to $\sum_{k=0}^{\infty}\gamma^k = \frac{1}{1-\gamma}$. Finally, taking the infinity norm with respect to all s and a, we obtain the bound in (21) of Lemma 2. ■

C. Main Convergence Result

Theorem 1. *Under Assumptions 1-2, consider F-ML-IRL (Algorithm 1) with step size $\alpha_{(m,k)} = \alpha_0/(mT+k)^\sigma$, where $\alpha_0 > 0$ and $\sigma \in (0,1)$ are constants, and M is the total number of dual aggregations. Then the following convergence results hold: (i) for the policy estimate:*

$$\begin{aligned}&\frac{1}{M}\sum_{m=0}^{M-1}\mathbb{E}[\|\log(\overline{\pi}_{(m,T)}) - \log(\pi_{\theta^i_{(m,T-1)}})\|_\infty] \\ &=\mathcal{O}(M^{-1}\gamma^{T-1}) + \mathcal{O}(M^{-\sigma}T^{1-\sigma}),\end{aligned} \tag{27}$$

and (ii) for the reward optimization:

$$\begin{aligned}&\frac{1}{M}\sum_{m=0}^{M-1}\mathbb{E}[\|\nabla L(\overline{\theta}_{(m,T)})\|^2] \\ &=\mathcal{O}(M^{-1}) + \mathcal{O}(M^{-\sigma}T^{-\sigma}) + \mathcal{O}(M^{-1-\sigma}T^{1-\sigma}).\end{aligned} \tag{28}$$

Remarks: The convergence rate of both the policy estimate and reward parameter optimization depends on the number of global aggregation rounds M and the number of local training steps T. The policy and reward function parameters in F-ML-IRL converge at a rate of $M^{-\sigma}T^{1-\sigma}$ and $M^{-\sigma}T^{-\sigma}$, respectively. Since we have $\sigma \in (0,1)$ and T is often fixed, we note that due to dual-aggregation and the variance caused by local training on distributed datasets across decentralized clients, F-ML-IRL exhibits a slightly slower convergence rate, compared with standard centralized ML-IRL with a single client (whose convergence rate is $M^{-\sigma}$). From Equations (27) and (28), there exists a sweet spot with respect to the number of local training steps T, since γ^{T-1} and $T^{-\sigma}$ both decrease with T, while $T^{1-\sigma}$ increases. Exploring this trade-off will be considered in future work.

Compared with single-level federated empirical risk minimization methods (e.g., FedAvg), our bilevel formulation introduces additional approximation and estimation terms. In our bounds (27) and (28), these appear as extra terms such as γ^{T-1} and $M^{-\sigma}T^{-\sigma}$. Under standard smoothness assumptions, the dependence on the number of global rounds remains sublinear.

In the following, we first analyze the convergence of policy estimates and reduce it to the convergence of Q-values (Lemma 3). We then analyze the distance between Q-values using the Lipschitz property, tracing back to the start of each dual aggregation round (Lemma 4). In particular, we examine the extra distance between the estimated policy and the optimal policy caused by aggregation, seeking the contraction property of Q-value estimates between adjacent aggregation rounds (Lemma 5). Next, for reward optimization, we leverage the Lipschitz smoothness of the likelihood and control the discrepancy between the stochastic gradient and the true gradient (Lemma 6). This allows us to use the convergence of Q-values from the previous analysis to demonstrate the gradient convergence of the reward parameter.

For simplicity of notation, we use $Q^{\text{soft}}_{i,(m,t)}$ to denote $Q^{\text{soft}}_{r_{\theta^i_{(m,t)}},\pi^i_{(m,t)}}$, the action-value function at a given state for the local policy and reward parameter estimates at round (m,t). Similarly, $Q^{\text{soft}}_{i,(m,t)}*$ denotes $Q^{\text{soft}}_{r_{\theta^i_{(m,t)}},\pi^i_{\theta_{(m,t)}}}$, which is the Q-function for the optimal policy under the reward parameter at round (m,t) and $Q^{\text{soft}}_{\overline{(m,t)}}$ denotes $Q^{\text{soft}}_{r_{\overline{\theta}^i_{(m,T)}},\pi_{\overline{\theta}^i_{(m,T)}}}$, which represents the Q-function for the aggregated policy and reward parameter at the m'th aggregation.

1) Convergence of Policy Estimate $\overline{\pi}_{(m,T)}$: We analyze the convergence of policy estimate in three steps.

Step 1: To initiate the policy-error analysis, we first relate any deviation in the soft-Q estimates directly to a gap in the corresponding log-policies. Lemma 3 bounds the log-policy gap by the local soft-Q gap.

Lemma 3 (Log-Policy Gap via Soft-Q Difference)**.** *The distance between the aggregated policy and the optimal policy under the pre-aggregation local reward parameter is bounded by twice the sup-norm difference of their soft-Q functions.*

$$\begin{aligned}&\|\log(\overline{\pi}_{(m,T)}) - \log(\pi_{\theta^i_{(m,T-1)}})\|_\infty \\ &\leq 2\|\overline{Q}^{soft}_{(m,T-1)} - Q^{soft}_{i,(m,T-2)}*\|_\infty.\end{aligned} \tag{29}$$

Proof: We first analyze the approximation error between the logarithm of the synchronized policy $\log(\overline{\pi}_{(m,T)}(a|s))$ and the logarithm of the optimal policy corresponding to the previous local reward parameter $\log(\pi_{\theta^i_{(m,T-1)}}(a|s))$ for all i. Specifically, we aim to bound the difference $|\log(\overline{\pi}_{(m,T)}(a|s)) - \log(\pi_{\theta^i_{(m,T-1)}}(a|s))|$. This difference represents the discrepancy between the synchronized policy after the m-th global aggregation and the optimal policy corresponding to the previous local reward parameter $\theta^i_{(m,T-1)}$. We aim to show that the distance between the logarithms of the synchronized policy and the optimal policy can be bounded by the difference between their corresponding soft-Q values. Specifically, we want to bound $|\log(\overline{\pi}_{(m,T)}(a|s)) - \log(\pi_{\theta^i_{(m,T-1)}}(a|s))| \leq \Delta_Q$, where Δ_Q involves the difference between the soft-Q values $\overline{Q}^{\text{soft}}_{(m,T-1)}(s,a)$ and $Q^{\text{soft}}_{r_{\theta^i_{(m,T-2)}},\pi_{\theta^i_{(m,T-2)}}}(s,a)$. Recall that the policy is proportional to the exponential of the soft-Q value in (9). Thus, we can write (30):

$$\begin{aligned}\log(\overline{\pi}_{(m,T)}(a|s)) &= \log\left(\frac{\exp\left(\overline{Q}^{\text{soft}}_{(m,T-1)}(s,a)\right)}{\sum_{\tilde{a}}\exp(\overline{Q}^{\text{soft}}_{(m,T-1)}(s,\tilde{a}))}\right)\\ &= \overline{Q}^{\text{soft}}_{(m,T-1)}(s,a) - \log\left(\sum_{\tilde{a}}\exp\left(\overline{Q}^{\text{soft}}_{(m,T-1)}(s,\tilde{a})\right)\right).\end{aligned} \tag{30}$$

Since $\log(\pi_{\theta^i_{(m,T-1)}}(a|s))$ is the optimal policy under the reward parameter $\theta^i_{(m,T-1)}$, according to [17], it has the form $\pi_{\theta^i_{(m,T-1)}}(a|s) = \frac{\exp(Q^{\text{soft}}_{r_{\theta^i_{(m,T-2)}},\pi_{\theta^i_{(m,T-2)}}}(s,a))}{\sum_{\tilde{a}}\exp(Q^{\text{soft}}_{r_{\theta^i_{(m,T-2)}},\pi_{\theta^i_{(m,T-2)}}}(s,\tilde{a}))}$, and we could similarly write (31):

$$\begin{aligned}\log(\pi_{\theta^i_{(m,T-1)}}(a|s)) &= Q^{\text{soft}}_{r_{\theta^i_{(m,T-2)}},\pi_{\theta^i_{(m,T-2)}}}(s,a)\\ &\quad - \log\left(\sum_{\tilde{a}}\exp\left(Q^{\text{soft}}_{r_{\theta^i_{(m,T-2)}},\pi_{\theta^i_{(m,T-2)}}}(s,\tilde{a})\right)\right).\end{aligned} \tag{31}$$

Subtracting the two expressions in (30) and (31), using the triangle inequality, we can bound the absolute value of the difference by the sum of the absolute values:

$$\begin{aligned}&\left|\log(\overline{\pi}_{(m,T)}(a|s)) - \log(\pi_{\theta^i_{(m,T-1)}}(a|s))\right|\\ &\leq \left|\overline{Q}^{\text{soft}}_{(m,T-1)}(s,a) - Q^{\text{soft}}_{r_{\theta^i_{(m,T-2)}},\pi_{\theta^i_{(m,T-2)}}}(s,a)\right|\\ &\quad + \left|\log\left(\sum_{\tilde{a}}\exp\left(\overline{Q}^{\text{soft}}_{(m,T-1)}(s,\tilde{a})\right)\right)\right.\\ &\quad \left. - \log\left(\sum_{\tilde{a}}\exp\left(Q^{\text{soft}}_{r_{\theta^i_{(m,T-2)}},\pi_{\theta^i_{(m,T-2)}}}(s,\tilde{a})\right)\right)\right|.\end{aligned} \tag{32}$$

The second term in (32) involves the difference of logarithms of sums. We can bound it using properties of logarithms and the maximum difference of the soft-Q values. We utilize (33) (as referenced in Equation (47) of [8]):

$$\begin{aligned}&\left|\log\left(\sum_{\tilde{a}}\exp(Q_1(s,\tilde{a}))\right) - \log\left(\sum_{\tilde{a}}\exp(Q_2(s,\tilde{a}))\right)\right|\\ &\leq \max_{\tilde{a}}|Q_1(s,\tilde{a}) - Q_2(s,\tilde{a})|.\end{aligned} \tag{33}$$

Applying (33) to the second term in (32), we have:

$$\begin{aligned}&\left|\log\left(\sum_{\tilde{a}}\exp\left(\overline{Q}^{\text{soft}}_{(m,T-1)}(s,\tilde{a})\right)\right) -\right.\\ &\quad \left.\log\left(\sum_{\tilde{a}}\exp\left(Q^{\text{soft}}_{r_{\theta^i_{(m,T-2)}},\pi_{\theta^i_{(m,T-2)}}}(s,\tilde{a})\right)\right)\right|\\ &\leq \max_{\tilde{a}}\left|\overline{Q}^{\text{soft}}_{(m,T-1)}(s,\tilde{a}) - Q^{\text{soft}}_{r_{\theta^i_{(m,T-2)}},\pi_{\theta^i_{(m,T-2)}}}(s,\tilde{a})\right|.\end{aligned} \tag{34}$$

Combining the results from (32) and (34) leads to (35):

$$\begin{aligned}&\left|\log(\overline{\pi}_{(m,T)}(a|s)) - \log(\pi_{\theta^i_{(m,T-1)}}(a|s))\right|\\ &\leq \left|\overline{Q}^{\text{soft}}_{(m,T-1)}(s,a) - Q^{\text{soft}}_{r_{\theta^i_{(m,T-2)}},\pi_{\theta^i_{(m,T-2)}}}(s,a)\right|\\ &\quad + \max_{\tilde{a}}\left|\overline{Q}^{\text{soft}}_{(m,T-1)}(s,\tilde{a}) - Q^{\text{soft}}_{r_{\theta^i_{(m,T-2)}},\pi_{\theta^i_{(m,T-2)}}}(s,\tilde{a})\right|.\end{aligned} \tag{35}$$

Taking the infinity norm of (35) gives Lemma 3. ■

Step 2: By looking back to the time right after the last aggregation, where all local servers share the same reward parameter $\overline{\theta}_{(m-1,T)}$, we can bound the difference in **Step 1** by relating it to the difference in reward parameters using (12), (13), and (19). Combined with the γ-contraction property of the soft-Q update, we obtain Lemma 4, which provides a bound on the distance between the aggregated soft-Q value and the optimal soft-Q value under the pre-aggregation local reward parameter.

Lemma 4 (One-Round Soft-Q Aggregation Contraction)**.**

$$\begin{aligned}&\|\overline{Q}^{soft}_{(m,T-1)} - Q^{soft}_{i,(m,T-2)}\|_\infty\\ &\leq \gamma^{T-2}\|Q^{soft}_{r_{\overline{\theta}_{(m-1,T)}},\overline{\pi}_{(m-1,T)}} - Q^{soft}_{r_{\overline{\theta}_{(m-1,T)}},\pi_{\overline{\theta}_{(m-1,T)}}}\|_\infty + E_1,\end{aligned} \tag{36}$$

where $E_1 = 4\alpha\left(\frac{1-\gamma^{T-2}}{1-\gamma} + T - 2\right)L_q^2$ *captures the accumulated error terms.*

Proof: First, applying the aggregation definition in (13) and using the triangle inequality, we obtain:

$$\begin{aligned}&\|\overline{Q}^{\text{soft}}_{(m,T-1)} - Q^{\text{soft}}_{r_{\theta^i_{(m,T-2)}},\pi_{\theta^i_{(m,T-2)}}}\|_\infty\\ &= \left\|\sum_{j=1}^N\frac{1}{N}\left(Q^{\text{soft}}_{r_{\theta^j_{(m,T-2)}},\pi^j_{(m,T-2)}} - Q^{\text{soft}}_{r_{\theta^i_{(m,T-2)}},\pi_{\theta^i_{(m,T-2)}}}\right)\right\|_\infty\\ &\leq \frac{1}{N}\sum_{j=1}^N\left\|Q^{\text{soft}}_{r_{\theta^j_{(m,T-2)}},\pi^j_{(m,T-2)}} - Q^{\text{soft}}_{r_{\theta^i_{(m,T-2)}},\pi_{\theta^i_{(m,T-2)}}}\right\|_\infty.\end{aligned} \tag{37}$$

Therefore, we move to analyze $\|Q^{\text{soft}}_{r_{\theta^j_{(m,T-2)}},\pi^j_{(m,T-2)}} - Q^{\text{soft}}_{r_{\theta^i_{(m,T-2)}},\pi_{\theta^i_{(m,T-2)}}}\|_\infty$, which is the difference of soft-Q

values between two different local nodes, one under policy estimation, and the other under optimal policy. Looking back to the time right after the last aggregation, where all local servers have the same reward parameter $\overline{\theta}_{(m-1,T)}$, we could further bound this difference using the difference of reward parameters, since the differences of local reward parameters are introduced by the local increment at each internal iteration except for the aggregation round. We start by decomposing this difference into three terms. Using the triangle inequality, we bound the sum as:

$$\begin{aligned}
&\left\| Q^{\text{soft}}_{r_{\theta^j_{(m,T-2)}},\pi^j_{(m,T-2)}} - Q^{\text{soft}}_{r_{\theta^i_{(m,T-2)}},\pi_{\theta^i_{(m,T-2)}}} \right\|_\infty \\
\leq &\left\| Q^{\text{soft}}_{r_{\theta^j_{(m,T-2)}},\pi^j_{(m,T-2)}} - Q^{\text{soft}}_{r_{\theta^j_{(m,T-2)}},\pi_{\theta^j_{(m,T-2)}}} \right\|_\infty \\
&+ \left\| Q^{\text{soft}}_{r_{\theta^j_{(m,T-2)}},\pi_{\theta^j_{(m,T-2)}}} - Q^{\text{soft}}_{r_{\theta^j_{(m,0)}},\pi_{\theta^j_{(m,0)}}} \right\|_\infty \\
&+ \left\| Q^{\text{soft}}_{r_{\theta^i_{(m,0)}},\pi_{\theta^i_{(m,0)}}} - Q^{\text{soft}}_{r_{\theta^i_{(m,T-2)}},\pi_{\theta^i_{(m,T-2)}}} \right\|_\infty .
\end{aligned} \tag{38}$$

The first term is the difference between the soft-Q values under the same reward parameter $\theta^j_{(m,T-2)}$ but different policies $\pi^j_{(m,T-2)}$ and $\pi_{\theta^j_{(m,T-2)}}$. The second term is the difference due to the change in reward parameters from $\theta^j_{(m,T-2)}$ to $\theta^j_{(m,0)}$, with corresponding optimal policies, and the third term is similar to the second term but for node i, comparing $\theta^i_{(m,0)}$ and $\theta^i_{(m,T-2)}$. We are utilizing the fact that $\theta^j_{(m,0)} = \theta^i_{(m,0)}$ since they are initialized after the previous aggregation. Applying (19) to (38), we have:

$$\begin{aligned}
&\| Q^{\text{soft}}_{r_{\theta^j_{(m,T-2)}},\pi^j_{(m,T-2)}} - Q^{\text{soft}}_{r_{\theta^i_{(m,T-2)}},\pi_{\theta^i_{(m,T-2)}}} \|_\infty \\
\leq &\left\| Q^{\text{soft}}_{r_{\theta^j_{(m,T-2)}},\pi^j_{(m,T-2)}} - Q^{\text{soft}}_{r_{\theta^j_{(m,T-2)}},\pi_{\theta^j_{(m,T-2)}}} \right\|_\infty \\
&+ L_q \left\| \theta^j_{(m,T-2)} - \theta^j_{(m,0)} \right\| + L_q \left\| \theta^i_{(m,0)} - \theta^i_{(m,T-2)} \right\| .
\end{aligned} \tag{39}$$

Next, using (12), we express the differences in reward parameters in terms of gradient updates (40):

$$\begin{aligned}
&\| Q^{\text{soft}}_{r_{\theta^j_{(m,T-2)}},\pi^j_{(m,T-2)}} - Q^{\text{soft}}_{r_{\theta^i_{(m,T-2)}},\pi_{\theta^i_{(m,T-2)}}} \|_\infty \\
\leq &\left\| Q^{\text{soft}}_{r_{\theta^j_{(m,T-2)}},\pi^j_{(m,T-2)}} - Q^{\text{soft}}_{r_{\theta^j_{(m,T-2)}},\pi_{\theta^j_{(m,T-2)}}} \right\|_\infty \\
&+ L_q \alpha \left\| \sum_{k=0}^{T-3} g^j_{(m,k)} \right\| + L_q \alpha \left\| \sum_{k=0}^{T-2} g^i_{(m,k)} \right\| .
\end{aligned} \tag{40}$$

According to existing RL optimization analysis [32], these stochastic gradients are uniformly bounded as:

$$\| g^i_{(m,k)} \| \leq 2L_q. \tag{41}$$

Applying this gradient bound to the sums in (40), we obtain:

$$\begin{aligned}
&\| Q^{\text{soft}}_{r_{\theta^j_{(m,T-2)}},\pi^j_{(m,T-2)}} - Q^{\text{soft}}_{r_{\theta^i_{(m,T-2)}},\pi_{\theta^i_{(m,T-2)}}} \|_\infty \\
\leq &\left\| Q^{\text{soft}}_{r_{\theta^j_{(m,T-2)}},\pi^j_{(m,T-2)}} - Q^{\text{soft}}_{r_{\theta^j_{(m,T-2)}},\pi_{\theta^j_{(m,T-2)}}} \right\|_\infty \\
&+ 4(T-2)\alpha L_q^2.
\end{aligned} \tag{42}$$

Now we can analyze the Q-value difference between two policies on the same client. According to equation (57) in [8], the difference between Q-values under policy estimation and optimal policy satisfies:

$$\begin{aligned}
&\| Q^{\text{soft}}_{r_{\theta^i_{(m,k)}},\pi^i_{(m,k)}} - Q^{\text{soft}}_{r_{\theta^i_{(m,k)}},\pi_{\theta^i_{(m,k)}}} \|_\infty \\
&\leq \gamma \| Q^{\text{soft}}_{r_{\theta^i_{(m,k-1)}},\pi^i_{(m,k-1)}} - Q^{\text{soft}}_{r_{\theta^i_{(m,k-1)}},\pi_{\theta^i_{(m,k-1)}}} \|_\infty \\
&\quad + 4\alpha L_q^2, \quad 1 \leq k \leq T-1, \quad m \in \mathbb{N}, \quad \forall i,
\end{aligned} \tag{43}$$

which provides a bound of the local Q-value using the previous Q-value times a contraction factor γ plus some extra term. We use it to compare the aggregated Q-value in the m-th outer round with the aggregated Q-value in the $m-1$ -th outer round as:

$$\begin{aligned}
&\| Q^{\text{soft}}_{r_{\theta^i_{(m,k)}},\pi^i_{(m,k)}} - Q^{\text{soft}}_{r_{\theta^i_{(m,k)}},\pi_{\theta^i_{(m,k)}}} \|_\infty \\
&\leq \gamma^k \| Q^{\text{soft}}_{r_{\theta^i_{(m,0)}},\pi^i_{(m,0)}} - Q^{\text{soft}}_{r_{\theta^i_{(m,0)}},\pi_{\theta^i_{(m,0)}}} \|_\infty + \frac{1-\gamma^k}{1-\gamma} \cdot 4\alpha L_q^2, \\
&\quad m \in \mathbb{N}, \quad 1 \leq k \leq T-1.
\end{aligned} \tag{44}$$

Applying (44) to (42), we have:

$$\begin{aligned}
&\| Q^{\text{soft}}_{r_{\theta^j_{(m,T-2)}},\pi^j_{(m,T-2)}} - Q^{\text{soft}}_{r_{\theta^i_{(m,T-2)}},\pi_{\theta^i_{(m,T-2)}}} \|_\infty \\
&\leq \gamma^{T-2} \| Q^{\text{soft}}_{r_{\overline{\theta}_{(m-1,T)}},\overline{\pi}_{(m-1,T)}} - Q^{\text{soft}}_{r_{\overline{\theta}_{(m-1,T)}},\pi_{\overline{\theta}_{(m-1,T)}}} \|_\infty \\
&\quad + 4\alpha \left(\frac{1-\gamma^{T-2}}{1-\gamma} + T - 2 \right) L_q^2.
\end{aligned} \tag{45}$$

Finally, we substitute (45) into (37) to complete the proof. ■

Step 3: Using Lemma 2, we further bound the difference in Q-values corresponding to different policies with the same reward during the aggregation step.

Lemma 5 (Q-Value Contraction)**.** *Under the same conditions as Theorem 1, for any round m, the difference between aggregated and optimal Q-values satisfies:*

$$\begin{aligned}
&\| \overline{Q}^{soft}_{(m,T-1)} - Q^{soft}_{i,(m,T-2)} * \|_\infty \\
&\leq (1-\gamma)\gamma^{T-1} \| \overline{Q}^{soft}_{(m-1,T-1)} - Q^{soft}_{i,(m-1,T-2)} * \|_\infty + E_2,
\end{aligned} \tag{46}$$

where $E_2 = 2\frac{1-\gamma^{T-2}}{1-\gamma} + (1-\gamma)^2\gamma^{T-2} + \frac{1-\gamma}{\gamma}(2T-3) + 2(T-2)L_q^2$.

Lemma 5 establishes the contraction property for the Q-value estimates between successive aggregation rounds.

Proof: We further analyze $\| Q^{\text{soft}}_{r_{\overline{\theta}_{(m-1,T)}},\overline{\pi}_{(m-1,T)}} - Q^{\text{soft}}_{r_{\overline{\theta}_{(m-1,T)}},\pi_{\overline{\theta}_{(m-1,T)}}} \|_\infty$ and use the triangle inequality to

decompose it into three parts to acquire the same form of Q-value difference in the previous outer round as:

$$\begin{aligned}
&\|Q^{\text{soft}}_{r_{\overline{\theta}_{(m-1,T)}},\overline{\pi}_{(m-1,T)}} - Q^{\text{soft}}_{r_{\overline{\theta}_{(m-1,T)}},\pi_{\overline{\theta}_{(m-1,T)}}}\|_\infty \\
\le\ &\|Q^{\text{soft}}_{r_{\overline{\theta}_{(m-1,T)}},\overline{\pi}_{(m-1,T)}} - \overline{Q}^{\text{soft}}_{(m-1,T-1)}\|_\infty \\
&+ \|\overline{Q}^{\text{soft}}_{(m-1,T-1)} - Q^{\text{soft}}_{r_{\theta^i_{(m-1,T-1)}},\pi_{\theta^i_{(m-1,T-1)}}}\|_\infty \\
&+ \|Q^{\text{soft}}_{r_{\theta^i_{(m-1,T-1)}},\pi_{\theta^i_{(m-1,T-1)}}} - Q^{\text{soft}}_{r_{\overline{\theta}_{(m-1,T)}},\pi_{\overline{\theta}_{(m-1,T)}}}\|_\infty.
\end{aligned} \tag{47}$$

We bound the first term in (47) by introducing a middle term and applying triangle inequality (48):

$$\begin{aligned}
&\|Q^{\text{soft}}_{r_{\overline{\theta}_{(m-1,T)}},\overline{\pi}_{(m-1,T)}} - \overline{Q}^{\text{soft}}_{(m-1,T-1)}\|_\infty \\
\le\ &\|Q^{\text{soft}}_{r_{\overline{\theta}_{(m-1,T)}},\overline{\pi}_{(m-1,T)}} - Q^{\text{soft}}_{r_{\theta^j_{(m-1,T-2)}},\overline{\pi}_{(m-1,T)}}\|_\infty + \\
&\frac{1}{N}\sum_{j=1}^{N}\|Q^{\text{soft}}_{r_{\theta^j_{(m-1,T-2)}},\overline{\pi}_{(m-1,T)}} - Q^{\text{soft}}_{r_{\theta^j_{(m-1,T-2)}},\pi^j_{(m-1,T-2)}}\|_\infty.
\end{aligned} \tag{48}$$

For the first term in (48), we leverage Lemma 7 in [8]:

$$|Q^{\text{soft}}_{r_{\theta_1},\pi} - Q^{\text{soft}}_{r_{\theta_2},\pi}| \le L_q\|\theta_1-\theta_2\|, \forall\pi, \forall\theta_1,\theta_2, \forall s\in\mathcal{S}, \forall a\in\mathcal{A}. \tag{49}$$

Then we further bound the difference between θ's using (12) and (41):

$$\begin{aligned}
&\|Q^{\text{soft}}_{r_{\overline{\theta}_{(m-1,T)}},\overline{\pi}_{(m-1,T)}} - Q^{\text{soft}}_{r_{\theta^j_{(m-1,T-2)}},\overline{\pi}_{(m-1,T)}}\|_\infty \\
&\le L_q\|\frac{1}{N}\sum_{j=1}^{N}\theta^j_{(m-1,T-1)} - \theta^j_{(m-1,T-2)}\| \\
&\le \frac{L_q}{N}\sum_{j=1}^{N}(\|\theta^j_{(m-1,T-1)} - \theta^j_{(m-1,0)}\| \\
&\quad + \|\theta^i_{(m-1,0)} - \theta^i_{(m-1,T-2)}\|) \\
&= \frac{L_q}{N}\sum_{j=1}^{N}\left(\alpha\left\|\sum_{k=0}^{T-2} g^j_{(m,k)}\right\| + \alpha\left\|\sum_{k=0}^{T-3} g^i_{(m,k)}\right\|\right) \\
&\le 2(2T-3)\alpha L_q^2.
\end{aligned}$$

For the second term in (48), by Lemma 2:

$$\begin{aligned}
&\|Q^{\text{soft}}_{r_{\theta^j_{(m-1,T-2)}},\overline{\pi}_{(m-1,T)}} - Q^{\text{soft}}_{r_{\theta^j_{(m-1,T-2)}},\pi^j_{(m-1,T-2)}}\|_\infty \\
&\le \frac{1-\gamma}{\gamma}\|\log(\overline{\pi}_{(m-1,T)}) - \log(\pi^j_{(m-1,T-2)})\|_\infty.
\end{aligned} \tag{50}$$

Using Lemma 3, we bound the right-hand side of (50) as:

$$\begin{aligned}
&\|\log(\overline{\pi}_{(m-1,T)}) - \log(\pi^j_{(m-1,T-2)})\|_\infty \\
\le\ &2\|\overline{Q}^{\text{soft}}_{(m-1,T-1)} - Q^{\text{soft}}_{r_{\theta^i_{(m-1,T-3)}},\pi_{\theta^i_{(m-1,T-3)}}}\|_\infty \\
\le\ &2\Big(\|\overline{Q}^{\text{soft}}_{(m-1,T-1)} - Q^{\text{soft}}_{r_{\theta^i_{(m-1,T-2)}},\pi_{\theta^i_{(m-1,T-2)}}}\|_\infty + \\
&\|Q^{\text{soft}}_{r_{\theta^i_{(m-1,T-2)}},\pi_{\theta^i_{(m-1,T-2)}}} - Q^{\text{soft}}_{r_{\theta^i_{(m-1,T-3)}},\pi_{\theta^i_{(m-1,T-3)}}}\|_\infty\Big).
\end{aligned} \tag{51}$$

For the last term in (51), we can apply the Lipschitz property and gradient bounds as in (38) to obtain:

$$\begin{aligned}
&\|Q^{\text{soft}}_{r_{\theta^i_{(m-1,T-2)}},\pi_{\theta^i_{(m-1,T-2)}}} - Q^{\text{soft}}_{r_{\theta^i_{(m-1,T-3)}},\pi_{\theta^i_{(m-1,T-3)}}}\|_\infty \\
\le\ &2\alpha L_q^2.
\end{aligned} \tag{52}$$

Plugging the above results into (47), we have:

$$\begin{aligned}
&\|Q^{\text{soft}}_{r_{\overline{\theta}_{(m-1,T)}},\overline{\pi}_{(m-1,T)}} - Q^{\text{soft}}_{r_{\overline{\theta}_{(m-1,T)}},\pi_{\overline{\theta}_{(m-1,T)}}}\|_\infty \\
&\le \frac{1-\gamma}{\gamma}\|\overline{Q}^{\text{soft}}_{(m-1,T-1)} - Q^{\text{soft}}_{r_{\theta^i_{(m-1,T-2)}},\pi_{\theta^i_{(m-1,T-2)}}}\|_\infty \\
&\quad + 2(2T-3)\alpha L_q^2 + \frac{1-\gamma}{\gamma}\cdot 4\alpha L_q^2.
\end{aligned} \tag{53}$$

Finally, we substitute (53) into (36) and obtain (46). ■

Combining (36) and the results in **Steps 1-3**, and summing over rounds $m=1$ to M, we finally obtain the convergence rate of the policy estimation in (27) of Theorem 1.

Proof: Summing the inequality from $m=1$ to $m=M$ gives:

$$\begin{aligned}
&\sum_{m=1}^{M}\|\overline{Q}^{\text{soft}}_{(m,T-1)} - Q^{\text{soft}}_{r_{\theta^i_{(m,T-2)}},\pi_{\theta^i_{(m,T-2)}}}\|_\infty \\
&\le (1-\gamma)\gamma^{T-1}\sum_{m=1}^{M}\|\overline{Q}^{\text{soft}}_{(m-1,T-1)} - Q^{\text{soft}}_{r_{\theta^i_{(m-1,T-2)}},\pi_{\theta^i_{(m-1,T-2)}}}\|_\infty \\
&\quad + M\left[2\cdot\frac{1-\gamma^{T-2}}{1-\gamma} + (1-\gamma)^2\gamma^{T-2}\right. \\
&\quad \left. + \frac{1-\gamma}{\gamma}(2T-3) + 2(T-2)\right]L_q^2.
\end{aligned} \tag{54}$$

Rearranging the inequality, it follows that:

$$\begin{aligned}
&\big(1-(1-\gamma)\gamma^{T-1}\big)\sum_{m=1}^{M}\|\overline{Q}^{\text{soft}}_{(m,T-1)} - Q^{\text{soft}}_{r_{\theta^i_{(m,T-2)}},\pi_{\theta^i_{(m,T-2)}}}\|_\infty \\
&\le (1-\gamma)\gamma^{T-1}\Big(\|\overline{Q}^{\text{soft}}_{(0,T-1)} - Q^{\text{soft}}_{r_{\theta^i_{(0,T-2)}},\pi_{\theta^i_{(0,T-2)}}}\|_\infty \\
&\quad - \|\overline{Q}^{\text{soft}}_{(M,T-1)} - Q^{\text{soft}}_{r_{\theta^i_{(M,T-2)}},\pi_{\theta^i_{(M,T-2)}}}\|_\infty\Big) \\
&\quad + M\cdot\left[2\cdot\frac{1-\gamma^{T-2}}{1-\gamma} + (1-\gamma)^2\gamma^{T-2}\right. \\
&\quad \left. + \frac{1-\gamma}{\gamma}(2T-3) + 2(T-2)\right]L_q^2 \\
&\le (1-\gamma)\gamma^{T-1}\|\overline{Q}^{\text{soft}}_{(0,T-1)} - Q^{\text{soft}}_{r_{\theta^i_{(0,T-2)}},\pi_{\theta^i_{(0,T-2)}}}\|_\infty \\
&\quad + M\cdot\left[2\cdot\frac{1-\gamma^{T-2}}{1-\gamma} + (1-\gamma)^2\gamma^{T-2}\right. \\
&\quad \left. + \frac{1-\gamma}{\gamma}(2T-3) + 2(T-2)\right]L_q^2.
\end{aligned} \tag{55}$$

Denote $C_0 = \|\overline{Q}^{\text{soft}}_{(0,T-1)} - Q^{\text{soft}}_{r_{\theta^i_{(0,T-2)}},\pi_{\theta^i_{(0,T-2)}}}\|_\infty$. Dividing both sides by $M(1-(1-\gamma)\gamma^{T-1})$, we get:

$$\begin{aligned}
&\frac{1}{M}\sum_{m=1}^{M}\|\overline{Q}^{\text{soft}}_{(m,T-1)} - Q^{\text{soft}}_{r_{\theta^i_{(m,T-2)}},\pi_{\theta^i_{(m,T-2)}}}\|_\infty \\
&\le \frac{(1-\gamma)\gamma^{T-1}}{(1-(1-\gamma)\gamma^{T-1})M}\,C_0 \\
&\quad + \frac{1}{1-(1-\gamma)\gamma^{T-1}}\cdot\left[2\cdot\frac{1-\gamma^{T-2}}{1-\gamma} + (1-\gamma)^2\gamma^{T-2}\right. \\
&\quad \left. + \frac{1-\gamma}{\gamma}(2T-3) + 2(T-2)\right]L_q^2. \qquad (56)
\end{aligned}$$

Recall the step size is defined as $\alpha_{(m,t)} = \frac{\alpha_0}{(mT+t)^\sigma}$, with $\sigma > 0$. Then the policy convergence result (27) in Theorem 1 gives:

$$\begin{aligned}
&\frac{1}{M}\sum_{m=1}^{M}\|\overline{Q}^{\text{soft}}_{(m,T-1)} - Q^{\text{soft}}_{r_{\theta^i_{(m,T-2)}},\pi_{\theta^i_{(m,T-2)}}}\|_\infty \\
&= \mathcal{O}(M^{-1}\gamma^T) + \mathcal{O}(M^{-\sigma}T^{1-\sigma}). \qquad (57)
\end{aligned}$$

■

2) Convergence of the global reward parameter $\overline{\theta}_{(m,T)}$:
Step 1: By the Lipschitz smoothness of $L(\theta)$ from (20), and using the reward aggregation (15) and the update rule (12):

$$\begin{aligned}
&L(\overline{\theta}_{(m,T)}) \\
&\ge L(\overline{\theta}_{(m-1,T)}) + \left\langle \nabla L(\overline{\theta}_{(m,T)}),\, \overline{\theta}_{(m,T)} - \overline{\theta}_{(m-1,T)}\right\rangle \\
&\quad - \frac{L_c}{2}\|\overline{\theta}_{(m,T)} - \overline{\theta}_{(m-1,T)}\|^2 \\
&= L(\overline{\theta}_{(m-1,T)}) + \alpha\left\langle \nabla L(\overline{\theta}_{(m,T)}),\, \frac{1}{N}\sum_{j=1}^{N}\sum_{k=0}^{T-2} g^j_{(m,k)}\right\rangle \\
&\quad - \frac{L_c\alpha^2}{2}\left\|\frac{1}{N}\sum_{j=1}^{N}\sum_{k=0}^{T-2} g^j_{(m,k)}\right\|^2. \qquad (58)
\end{aligned}$$

Step 2: Next, we analyze how the stochastic gradient estimates $g^i_{(m,k)}$ relate to the true gradient $\nabla L(\theta_{(m-1,T)})$. Leveraging the reward update rule (12) and aggregation scheme (15), we can bound their difference and control the additional error terms introduced by the federated learning framework. This analysis allows us to express the gradient of the global reward parameter in terms of the discrepancy in Q-values.

Lemma 6 (Gradient-Q Relationship)**.** *Under the same conditions as Theorem 1, for any round m, the gradient of the global reward parameter satisfies:*

$$\begin{aligned}
&\alpha(T-1)\mathbb{E}[\|\nabla L(\overline{\theta}_{(m-1,T)})\|^2] \\
&\le \alpha C_1\mathbb{E}[\|\overline{Q}^{soft}_{(m-1,T-1)} - Q^{soft}_{i,(m-1,T-2)}\|_\infty] \qquad (59) \\
&\quad + \mathbb{E}[L(\overline{\theta}_{(m,T)}) - L(\overline{\theta}_{(m-1,T)})] + E_3,
\end{aligned}$$

where $C_1 = \frac{4(1-\gamma^{T-1})}{\gamma}L_q^2C_d\sqrt{|\mathcal{S}|\cdot|\mathcal{A}|}$ *and* $E_3 = 8\alpha L_q^3C_d\sqrt{|\mathcal{S}|\cdot|\mathcal{A}|}\cdot\frac{T-1-\frac{1-\gamma^{T-1}}{1-\gamma}}{1-\gamma} + \frac{(T-1)(3T-1)\alpha^2L_c^2L_q^2}{2} + \frac{4(1-\gamma^{T-1})}{1-\gamma}\alpha L_q^2C_d\sqrt{|\mathcal{S}|\cdot|\mathcal{A}|}\cdot[2(2T-3)\alpha L_q^2 + \frac{1-\gamma}{\gamma}\cdot 4\alpha L_q^2]$ *are two auxiliary variables.*

This lemma reveals how the gradient of the reward parameter relates to the discrepancy in Q-values, providing insight into the convergence behavior.

Proof: We compare $g^i_{(m,k)}$ with the true gradient of $L(\theta^i_{(m,k)})$ and leverage the fact that $\|\nabla L(\theta)\|_\infty \le 2L_q$:

$$\begin{aligned}
&L(\overline{\theta}_{(m,T)}) \\
&\ge L(\overline{\theta}_{(m-1,T)}) + \alpha\left\langle \nabla L(\overline{\theta}_{(m,T)}),\, \frac{1}{N}\sum_{j=1}^{N}\sum_{k=0}^{T-2} g^j_{(m,k)}\right\rangle \\
&\quad - \frac{L_c\alpha^2}{2}\left\|\frac{1}{N}\sum_{j=1}^{N}\sum_{k=0}^{T-2} g^j_{(m,k)}\right\|^2 \\
&\ge L(\overline{\theta}_{(m-1,T)}) + \alpha\Big\langle \nabla L(\overline{\theta}_{(m,T)}),\, \frac{1}{N}\sum_{j=1}^{N}\sum_{k=0}^{T-2} g^j_{(m,k)} \\
&\quad - (T-1)\nabla L(\overline{\theta}_{(m-1,T)})\Big\rangle + \alpha(T-1)\left\|\nabla L(\overline{\theta}_{(m-1,T)})\right\|^2 \\
&\quad - \frac{L_c\alpha^2}{2}\left\|\frac{1}{N}\sum_{j=1}^{N}\sum_{k=0}^{T-2} g^j_{(m,k)}\right\|^2 \\
&\ge L(\overline{\theta}_{(m-1,T)}) - 2\alpha L_q\left\|\frac{1}{N}\sum_{j=1}^{N}\sum_{k=0}^{T-2}\left[g^j_{(m,k)} - \nabla L(\overline{\theta}_{(m-1,T)})\right]\right\| \\
&\quad + \alpha(T-1)\left\|\nabla L(\overline{\theta}_{(m-1,T)})\right\|^2 - \frac{(T-1)^2L_cL_q^2\alpha^2}{2}. \qquad (60)
\end{aligned}$$

For $g^j(m,k)$ in (60), we evaluate its distance to $\nabla L(\theta^j_{(m,k)})$ and also consider the distance between $\nabla L(\theta^j_{(m,k)})$ and $\nabla L(\overline{\theta}_{(m,T)}$ with the help of the triangle inequality:

$$\begin{aligned}
&L(\overline{\theta}_{(m,T)}) \\
&\ge L(\overline{\theta}_{(m-1,T)}) - 2\alpha L_q\cdot\left[\frac{1}{N}\sum_{j=1}^{N}\sum_{k=0}^{T-2}\Big(\|g^j_{(m,k)} - \nabla L(\theta^j_{(m,k)})\|\right. \\
&\quad \left. + \|\nabla L(\theta^j_{(m,k)}) - \nabla L(\theta^j_{(m,0)})\|\Big)\right] \\
&\quad + \alpha(T-1)\left\|\nabla L(\overline{\theta}_{(m-1,T)})\right\|^2 - \frac{(T-1)^2L_cL_q^2\alpha^2}{2}. \qquad (61)
\end{aligned}$$

Taking expectation over both sides, we obtain (62):

$$\begin{aligned}
&\mathbb{E}[L(\overline{\theta}_{(m,T)})] \\
&\ge \mathbb{E}[L(\overline{\theta}_{(m-1,T)})] - 2\alpha L_q\cdot\left[\frac{1}{N}\sum_{j=1}^{N}\sum_{k=0}^{T-2}\Big(\right. \\
&\quad \left.\mathbb{E}\|g^j_{(m,k)} - \nabla L(\theta^j_{(m,k)})\| + \|\nabla L(\theta^j_{(m,k)}) - \nabla L(\theta_{(m,0)})\|\Big)\right] \\
&\quad + \alpha(T-1)\left\|\nabla L(\overline{\theta}_{(m-1,T)})\right\|^2 - \frac{(T-1)^2L_cL_q^2\alpha^2}{2}. \qquad (62)
\end{aligned}$$

According to equations (62) and (63) in [8], we have:

$$\mathbb{E}\left\|g^j_{(m,k)} - \nabla L(\theta^j_{(m,k)})\right\| \tag{63}$$
$$\leq 2L_q C_d\sqrt{|\mathcal{S}|\cdot|\mathcal{A}|}\cdot\mathbb{E}\left\|Q^{\text{soft}}_{r_{\theta^j_{(m,k)}},\pi^j_{(m,k)}} - Q^{\text{soft}}_{r_{\theta^j_{(m,k)}},\pi_{\theta^j_{(m,k)}}}\right\|_\infty.$$

Then, using the Lipschitz property of L in (20), we have:

$$\begin{aligned}
&\mathbb{E}[L(\overline{\theta}_{(m,T)})]\\
&\geq \mathbb{E}[L(\theta^i_{(m-1,T)})] - 2\alpha L_q\cdot\frac{1}{N}\sum_{j=1}^{N}\sum_{k=0}^{T-2}\Big(2L_qC_d\sqrt{|\mathcal{S}|\cdot|\mathcal{A}|}\cdot\\
&\mathbb{E}\left\|Q^{\text{soft}}_{r_{\theta^j_{(m,k)}},\pi^j_{(m,k)}} - Q^{\text{soft}}_{r_{\theta^j_{(m,k)}},\pi_{\theta^j_{(m,k)}}}\right\|_\infty\\
&+ L_c\mathbb{E}\left\|\theta^j_{(m,k)} - \theta^j_{(m,0)}\right\|\Big)\\
&+ \alpha(T-1)\mathbb{E}\left\|\nabla L(\overline{\theta}_{(m-1,T)})\right\|^2 - \frac{(T-1)^2L_cL_q^2\alpha^2}{2}.
\end{aligned} \tag{64}$$

Similar to (42), we have $\|\theta^j_{(m,k)} - \theta^j_{(m,0)}\| \leq 2k\alpha L_q$. Applying (44) to (64), we have (65):

$$\begin{aligned}
&\mathbb{E}[L(\overline{\theta}_{(m,T)})]\\
&= \mathbb{E}[L(\overline{\theta}_{(m-1,T)})] - \frac{4(1-\gamma^{T-1})}{1-\gamma}\cdot\alpha L_q^2C_d\sqrt{|\mathcal{S}|\cdot|\mathcal{A}|}\\
&\mathbb{E}\left\|Q^{\text{soft}}_{r_{\overline{\theta}_{(m-1,T)}},\overline{\pi}_{(m-1,T)}} - Q^{\text{soft}}_{r_{\overline{\theta}_{(m-1,T)}},\pi_{\overline{\theta}_{(m-1,T)}}}\right\|_\infty\\
&- 8\alpha L_q^3C_d\sqrt{|\mathcal{S}||\mathcal{A}|}\frac{T-1-\frac{1-\gamma^{T-1}}{1-\gamma}}{1-\gamma} - T(T-1)\alpha^2L_cL_q^2\\
&+ \alpha(T-1)\mathbb{E}\left\|\nabla L(\overline{\theta}_{(m-1,T)})\right\|^2 - \frac{(T-1)^2L_cL_q^2\alpha^2}{2}.
\end{aligned} \tag{65}$$

Plugging the result from (53) and rearranging terms, we obtain (59). ∎

By combining this with the convergence of the Q-value difference established in **Step 3** of the Policy Estimation proof, we obtain the desired convergence of the reward parameter.

Summing the inequality above from $m = 1$ to M and dividing both sides by $\alpha(T-1)M$, it holds that:

$$\begin{aligned}
&\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\left\|\nabla L(\overline{\theta}_{(m-1,T)})\right\|^2\\
&\leq \frac{\sum_{m=1}^{M}\mathbb{E}\|\overline{Q}^{\text{soft}}_{(m-1,T-1)} - Q^{\text{soft}}_{r_{\theta^i_{(m-1,T-2)}},\pi_{\theta^i_{(m-1,T-2)}}}\|_\infty}{M(T-1)}\\
&+ \frac{\mathbb{E}[L(\overline{\theta}_{(m,T)}) - L(\overline{\theta}_{(0,T)})]}{\alpha(T-1)M} + \frac{8L_q^3C_d\sqrt{|\mathcal{S}||\mathcal{A}|}}{M(1-\gamma)}\\
&+ \frac{16\alpha\left(1+\frac{1}{T-1}\cdot\frac{1-\gamma}{\gamma}\right)}{M(1-\gamma)}L_q^4C_d\sqrt{|\mathcal{S}|\cdot|\mathcal{A}|}\\
&+ \frac{(3T-1)\alpha L_cL_q^2}{2M}.
\end{aligned} \tag{66}$$

Since $L(\overline{\theta}_{(m,T)})$ is negative and $L(\overline{\theta}_{(0,T)})$ is a bounded constant, we substitute (57) into (66) and get (28) in Theorem 1.

V. Evaluations

In this section, we evaluate our proposed F-ML-IRL algorithm on high-dimensional robotic control tasks in MuJoCo. We first describe the experimental setup, including baseline methods, network architectures, and hyperparameters. We then present comprehensive evaluation results comparing F-ML-IRL with state-of-the-art centralized IRL and IL approaches.

A. Experiment Setup

We evaluated the proposed F-ML-IRL method on five high-dimensional robotic control tasks in MuJoCo [18], including Ant, HalfCheetah, Hopper, Humanoid, and Walker2d. These environments involve locomotion and coordination behaviors in simulated robotic systems and are widely used as standard benchmarks for RL and IRL. The purpose of our evaluation is twofold: (i) to verify that F-ML-IRL can successfully recover a reward function that explains distributed expert demonstrations, and (ii) to test whether policies trained under the reconstructed rewards achieve performance comparable to the original experts. Expert demonstrations are generated by pre-trained SAC agents with different expertise levels, and each client holds trajectories from one such agent. This setup allows us to test whether F-ML-IRL can unify heterogeneous, non-i.i.d. demonstrations into a shared reward model. For comparison, we selected several state-of-the-art baselines, including IL approaches that only learn the expert policy—specifically like BC [19] and GAIL [5], as well as IRL methods that simultaneously learn both a reward function and a policy, namely f-IRL [20] and ML-IRL [8]. To ensure fairness, we used Soft Actor-Critic (SAC) [42] as the base RL algorithm for all methods, as it incorporates elements of Soft Q-Learning and achieves strong performance using the actor-critic scheme. SAC provides a continuous-action analog of the discrete soft policy improvement, since its policy update step minimizes the KL divergence between the policy distribution and the Boltzmann distribution defined by the Q-function. The experiments were conducted on a server with AMD EPYC 7513 32-Core Processors and NVIDIA RTX A6000 GPUs.

We use Multi-Layer Perceptron (MLP) to represent both the policy and the Q network in SAC, as well as the reward function. We use ReLU as the activation function and Adam as the optimizer. The key parameters of the SAC model and the parameterization of the reward function remain consistent between the different algorithms and are detailed in Table I in the Supplementary Material. For F-ML-IRL, we choose $M = 200$ rounds and $T = 5$ local steps. The results for F-ML-IRL and all other baselines are averaged over three different random seeds. Our evaluation code is available at https://anonymous.4open.science/r/F-ML-IRL/. For f-IRL, we utilize the official implementation available at https://github.com/twni2016/f-IRL, which also includes implementations for BC and GAIL. The official implementation of ML-IRL can be found at https://github.com/Cloud0723/ML-IRL.

At each iteration, we sample 10 trajectories from the current local policy estimate and compare them with the expert demonstration to update the reward parameter. The reward levels of the expert demonstrations are shown in Table II in the

TABLE I

COMPARISON OF F-ML-IRL WITH BASELINES ON MUJOCO TASKS ACROSS DIFFERENT NUMBERS OF CLIENTS AND DEMONSTRATION TRAJECTORY LENGTHS. IN F-ML-IRL, WE CONSIDER A DECENTRALIZED SETTING WHERE EACH LOCAL AGENT HAS ACCESS TO UNIQUE LOCAL DATA AND CAN COMMUNICATE WITH OTHER AGENTS TO SHARE MODEL PARAMETERS

. In contrast, for the decentralized baselines, each agent has access to the same type of local data but trains its model independently, without the ability to collaborate or exchange information. F-ML-IRL outperforms all other baselines in 60% of the scenarios.

Environment	Setting	F-ML-IRL	ML-IRL	BC	GAIL	f-IRL
Ant	(3, 200)	**6534.84 ± 62.05**	6175.32 ± 43.75	984.07 ± 0.30	988.59 ± 0.30	5723.94 ± 172.60
	(3, 1000)	**6295.42 ± 76.83**	6213.52 ± 113.12	688.51 ± 134.76	988.44 ± 0.74	5662.66 ± 140.96
	(5, 200)	**6434.21 ± 56.04**	6233.98 ± 87.04	984.02 ± 0.28	988.98 ± 0.64	5735.55 ± 168.02
	(5, 1000)	**6444.92 ± 138.36**	6201.26 ± 89.45	775.82 ± 187.62	988.28 ± 0.28	5580.52 ± 125.58
HalfCheetah	(3, 200)	12162.63 ± 97.27	12639.25 ± 41.53	-0.62 ± 0.02	9578.49 ± 175.62	**13384.21 ± 163.46**
	(3, 1000)	12632.71 ± 354.49	**13288.66 ± 313.48**	343.69 ± 356.46	10054.14 ± 265.54	13177.06 ± 428.71
	(5, 200)	12045.92 ± 167.35	12651.53 ± 218.90	-0.61 ± 0.02	9353.45 ± 491.20	**13770.08 ± 674.19**
	(5, 1000)	12361.25 ± 11.47	13316.89 ± 263.44	199.90 ± 290.75	9214.57 ± 466.04	**13345.78 ± 517.94**
Hopper	(3, 200)	**3592.51 ± 18.01**	3579.04 ± 35.85	18.13 ± 0.00	1023.33 ± 1.60	3535.44 ± 25.67
	(3, 1000)	**3672.46 ± 88.91**	3662.74 ± 135.37	1212.10 ± 925.42	1025.56 ± 10.27	3457.41 ± 57.96
	(5, 200)	3549.39 ± 15.97	**3574.72 ± 90.54**	18.13 ± 0.01	1023.30 ± 6.83	3511.00 ± 160.35
	(5, 1000)	**3704.81 ± 61.84**	3655.13 ± 109.05	955.09 ± 857.06	1028.73 ± 9.26	3531.85 ± 66.46
Humanoid	(3, 200)	**5849.72 ± 35.67**	5835.57 ± 271.94	242.98 ± 0.58	4474.71 ± 338.27	5784.10 ± 167.77
	(3, 1000)	**5844.76 ± 50.18**	5804.65 ± 204.09	380.00 ± 199.10	4317.20 ± 265.47	5461.96 ± 112.67
	(5, 200)	**5899.00 ± 35.67**	5895.19 ± 183.68	243.02 ± 0.62	4465.39 ± 265.68	5597.98 ± 271.26
	(5, 1000)	**5820.34 ± 41.31**	5760.77 ± 218.80	443.60 ± 174.08	3756.19 ± 862.88	5505.29 ± 136.06
Walker2d	(3, 200)	4718.99 ± 92.28	4833.13 ± 223.49	8.27 ± 0.06	220.11 ± 142.69	**5585.27 ± 126.78**
	(3, 1000)	5655.09 ± 53.98	**5739.55 ± 52.47**	577.80 ± 218.45	135.54 ± 151.96	5704.69 ± 79.93
	(5, 200)	4480.87 ± 42.45	4903.23 ± 219.03	8.27 ± 0.05	208.88 ± 154.67	**5576.01 ± 174.81**
	(5, 1000)	**5792.80 ± 20.17**	5769.92 ± 133.93	473.49 ± 212.99	220.11 ± 142.69	5720.78 ± 77.70

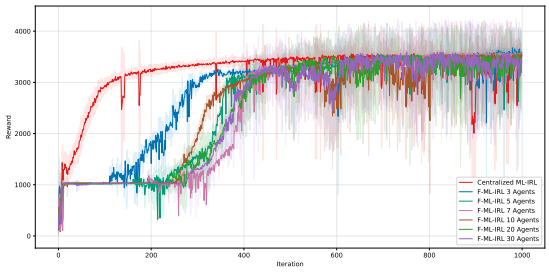


Fig. 2. Convergence of F-ML-IRL in Hopper Environment compared with centralized ML-IRL. As the number of clients (and thus the non-iid datasets) increases from 3 to 30, F-ML-IRL takes longer to converge and nevertheless achieves more significant improvement by leveraging distributed demonstration data on the clients.

Supplementary Material. For 3 agents, we use Data 3, 4, and 5; for 5 agents, Data 2, 3, 4, 5, and 6 are used. For 7 agents, all 7 data sets are distributed across different local clients.

B. Numerical Results

We evaluate different algorithms using the rewards associated with the recovered human expert policies in the original environment (same as the method adopted in previous work). We compare F-ML-IRL with the baselines on five MuJoCo tasks under non-iid data distributions, where each client contains different demonstration data corresponding to varying levels of expertise, and detailed demonstration data information is displayed in Table II in the Supplementary Material. For the baselines that rely on centralized learning, we average the results from centralized learning on different demonstration data, i.e., the same data allocation compared with F-ML-IRL but do not allow communication between individual agents.

The evaluation results are summarized in Table I. We have tested each algorithm and each MuJoCo task under 4 settings, i.e., with 3 or 5 clients and with demonstration trajectory lengths equal to 200 or 1000. For F-ML-IRL, we average the rewards across different local clients, while for all other centralized learning baselines, we let multiple local clients train their models using local data independently and average their rewards. As demonstrated in [8], even a single expert trajectory of length 1000 can lead to a well-recovered policy using ML-IRL. To investigate the performance of our model under the conditions of scarce and distributed data, we use a single expert trajectory of length 1000 and further reduce its length to 200 in the experiments.

We note that F-ML-IRL ensures convergence of the recovered reward in decentralized learning and achieves rewards comparable to or higher than those of the baselines in almost all settings and tasks. It even outperforms centralized baselines in more than half of the settings and tasks (12 out of 20), due to its ability to utilize distributed data. The performance of F-ML-IRL is pretty robust as the number of clients increases to 5 and the expert trajectory length reduces to 200. IL baselines like BC and GAIL generally have lower performance and even fail in some settings. Although ML-IRL performs generally well, it does not recover a satisfactory policy when data are limited. On the other hand, f-IRL performs relatively well in some scenarios, but underperforms ML-IRL and F-ML-IRL in most cases. In contrast, our F-ML-IRL consistently achieves similar or higher recovered rewards compared to all baselines, particularly maintaining robust performance even when data are limited.

We further illustrate the convergence of our F-ML-IRL algorithm compared to the centralized learning baseline using ML-IRL in the Hopper environment with trajectory length 1000, as shown in Figure 2. As the number of clients (and thus

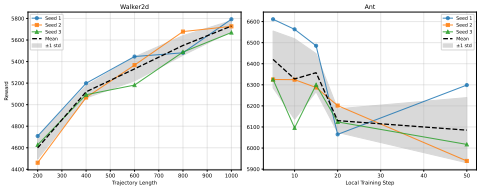


Fig. 3. Recovered reward of F-ML-IRL in Walker2d Environment using expert trajectories with lengths from 200 to 1000 and local step 5, and Ant environment using local step T's from 5 to 50 with trajectory length 1000. Results for 3 different random seeds are shown along with the mean and standard deviation. Longer trajectory lengths and smaller local step T's tend to improve the performance of the recovered policies.

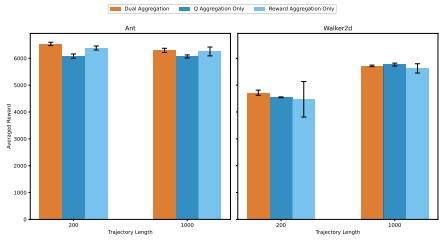


Fig. 4. Ablation Study for dual aggregation in Ant and Walker2d environments. Dual aggregation achieves higher reward compared with performing aggregation on only one level of the optimization problem.

the number of non-iid local datasets) increases (from 3 clients to 30 clients), it takes F-ML-IRL more rounds to converge, because of the increased variance introduced by local training on more participating clients and datasets. However, F-ML-IRL is able to converge to a higher recovered reward than the baseline. Centralized ML-IRL agents suffer from local single demonstration data. In contrast, as the number of clients and demonstration dataset increases, F-ML-IRL shows a more significant improvement by using distributed demonstration data on the clients, with strong scalability guarantee given the consistent performance across experiments with different numbers of agents as large as 30.

To evaluate the scalability of our F-ML-IRL algorithm, we varied the trajectory lengths from 200 to 1000, and compared the recovered rewards in the Walker2d environment with local step T set to be 5. Also, we set the local step T to be vary from 5 to 50 with trajectory length 1000 to study the impact of communication frequency in F-ML-IRL using the Ant environment. As shown in Fig. 3, longer trajectories lead to higher recovered rewards due to more complete strategic information included, while smaller local training steps between communications gives better performance in general.

We conducted an ablation study to highlight the importance of using dual aggregation for both Q-values and reward parameters. Specifically, we compared models that perform aggregation solely on Q-values or solely on reward parameters with the full model that applies dual aggregation. The impact of aggregation settings on the final reward attained by the recovered optimal policy is shown in Fig. 4. Both aggregations on the Q-values (Network parameters in realistic implementation) and reward parameters contribute to a better policy, while lack of Q aggregation may lead to a drastic variance increase.

We further demonstrate another baseline setting with two setups: (i) a single client with medium-level demonstrations, denoted as *medium* and (ii) a single client with a mixture of demonstrations of different levels, denoted as *mixed*. In either case, the total amount of local data per client remains the same in the experiments. We display the results and comparison using this setting to show that our F-ML-IRL could maintain reliable and satisfactory results under comparison with different data allocation of centralized learning. The full experimental results with different parameters are provided in Table III in the Supplementary Material. F-ML-IRL outperforms most of the baselines in both mixed and medium centralized expert data settings.

VI. Conclusions

This paper proposes F-ML-IRL for federated maximum-likelihood inverse reinforcement learning. It enables decentralized learning of a shared latent reward function from distributed human expert demonstration datasets on decentralized clients. The F-ML-IRL algorithm leverages a dual-aggregation to update the shared global model and performs bi-level local updates for inverse learning. We analyze the convergence and time-complexity of F-ML-IRL. Evaluation results on MuJoCo tasks show that F-ML-IRL ensures convergence of the recovered reward and achieves recovered rewards comparable to or higher than state-of-the-art baselines using centralized inverse learning. For further work, we plan to investigate further communication reduction and the use of heterogeneous local models in F-ML-IRL.

Acknowledgments

This work was supported in part by the U.S. Military Academy (USMA) under Cooperative Agreement No. W911NF-23-2-0175. The views and conclusions expressed in this paper are those of the authors and do not reflect the official policy or position of the U.S. Military Academy, U.S. Army, U.S. Department of War, or U.S. Government.

Appendix
Centralized ML-IRL Upper Bound

In addition to the prior baselines, we include a *Centralized ML-IRL (Upper Bound)* baseline in Table II in the appendix, which aggregates all expert demonstrations across clients into a single dataset and trains ML-IRL in the standard centralized manner. This represents the performance of an idealized centralized learner with full data access, and thus serves as an upper bound for our federated approach.

References

[1] S. Arora and P. Doshi, "A survey of inverse reinforcement learning: Challenges, methods and progress," 2020.
[2] S. Russell, "Learning agents for uncertain environments," in *Proceedings of the eleventh annual conference on Computational learning theory*, pp. 101–103, 1998.
[3] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
[4] W. Song, J. Ning, and S. Tong, "Inverse q-learning optimal control for takagi-sugeno fuzzy systems," *IEEE Transactions on Fuzzy Systems*, 2025.

TABLE II
COMPARE F-ML-IRL AND CENTRALIZED ML-IRL BASELINE ON MUJOCO TASKS, WITH DIFFERENT NUMBER OF CLIENTS AND DEMONSTRATION TRAJECTORY LENGTH. F-ML-IRL ACHIEVES SIMILAR LEVEL OF RECOVERED REWARD IN MOST OF THE SCENARIOS COMPARED WITH THE CENTRALIZED ML-IRL.

Environment	Setting	F-ML-IRL	Centralized ML-IRL
Ant	(3, 200)	6534.84 ± 62.05	6253.28 ± 57.63
	(3, 1000)	6295.42 ± 76.83	6323.10 ± 98.51
	(5, 200)	6434.21 ± 56.04	6289.45 ± 76.21
	(5, 1000)	6444.92 ± 138.36	6310.74 ± 54.22
HalfCheetah	(3, 200)	12162.63 ± 97.27	13031.89 ± 153.20
	(3, 1000)	12632.71 ± 354.49	13868.23 ± 165.83
	(5, 200)	12045.92 ± 167.35	13103.97 ± 254.75
	(5, 1000)	12361.25 ± 11.47	14061.00 ± 236.03
Hopper	(3, 200)	3592.51 ± 18.01	3585.90 ± 56.72
	(3, 1000)	3672.46 ± 88.91	3657.93 ± 77.29
	(5, 200)	3549.39 ± 15.97	3593.24 ± 85.87
	(5, 1000)	3704.81 ± 61.84	3658.47 ± 51.54
Humanoid	(3, 200)	5849.72 ± 35.67	5836.73 ± 46.31
	(3, 1000)	5844.76 ± 50.18	5709.95 ± 25.41
	(5, 200)	5899.00 ± 35.67	5897.49 ± 36.78
	(5, 1000)	5820.34 ± 41.31	5773.84 ± 54.05
Walker2d	(3, 200)	4718.99 ± 92.28	4951.47 ± 75.38
	(3, 1000)	5655.09 ± 53.98	5986.22 ± 121.91
	(5, 200)	4480.87 ± 42.45	5135.90 ± 89.01
	(5, 1000)	5792.80 ± 20.17	5948.10 ± 57.52

[5] J. Ho and S. Ermon, "Generative adversarial imitation learning," *Advances in neural information processing systems*, vol. 29, 2016.

[6] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey, *et al.*, "Maximum entropy inverse reinforcement learning.," in *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, vol. 8, pp. 1433–1438, Chicago, IL, USA, 2008.

[7] H. Ratia, L. Montesano, and R. Martinez-Cantin, "On the performance of maximum likelihood inverse reinforcement learning," *arXiv preprint arXiv:1202.1558*, 2012.

[8] S. Zeng, C. Li, A. Garcia, and M. Hong, "Maximum-likelihood inverse reinforcement learning with finite-time guarantees," *Advances in Neural Information Processing Systems*, vol. 35, pp. 10122–10135, 2022.

[9] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, pp. 1273–1282, PMLR, 2017.

[10] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[11] H. Jin, Y. Peng, W. Yang, S. Wang, and Z. Zhang, "Federated reinforcement learning with environment heterogeneity," in *International Conference on Artificial Intelligence and Statistics*, pp. 18–37, PMLR, 2022.

[12] H. Zhou, T. Lan, G. P. Venkataramani, and W. Ding, "Every parameter matters: Ensuring the convergence of federated learning with dynamic heterogeneous models reduction," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[13] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," *arXiv preprint arXiv:1907.02189*, 2019.

[14] X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou, "Fedbn: Federated learning on non-iid features via local batch normalization," *arXiv preprint arXiv:2102.07623*, 2021.

[15] Q. Li, B. He, and D. Song, "Model-contrastive federated learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10713–10722, 2021.

[16] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," *Advances in neural information processing systems*, vol. 33, pp. 7611–7623, 2020.

[17] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine, "Reinforcement learning with deep energy-based policies," in *International conference on machine learning*, pp. 1352–1361, PMLR, 2017.

[18] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033, IEEE, 2012.

[19] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," *Advances in neural information processing systems*, vol. 1, 1988.

[20] T. Ni, H. Sikchi, Y. Wang, T. Gupta, L. Lee, and B. Eysenbach, "f-irl: Inverse reinforcement learning via state marginal matching," in *Conference on Robot Learning*, pp. 529–551, PMLR, 2021.

[21] A. Y. Ng, S. Russell, *et al.*, "Algorithms for inverse reinforcement learning.," in *Icml*, vol. 1, p. 2, 2000.

[22] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the twenty-first international conference on Machine learning*, p. 1, 2004.

[23] N. D. Ratliff, J. A. Bagnell, and M. A. Zinkevich, "Maximum margin planning," in *Proceedings of the 23rd international conference on Machine learning*, pp. 729–736, 2006.

[24] D. Ramachandran and E. Amir, "Bayesian inverse reinforcement learning.," in *IJCAI*, vol. 7, pp. 2586–2591, 2007.

[25] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[26] V. S. Donge, B. Lian, F. L. Lewis, and A. Davoudi, "Efficient reward shaping for multiagent systems," *IEEE Transactions on Control of Network Systems*, vol. 12, no. 1, pp. 687–699, 2024.

[27] V. S. Donge, B. Lian, F. L. Lewis, and A. Davoudi, "Multiagent graphical games with inverse reinforcement learning," *IEEE Transactions on Control of Network Systems*, vol. 10, no. 2, pp. 841–852, 2022.

[28] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pp. 1310–1321, 2015.

[29] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *arXiv preprint arXiv:1610.02527*, 2016.

[30] B. Liu, L. Wang, M. Liu, and C.-Z. Xu, "Federated imitation learning: A novel framework for cloud robotic systems with heterogeneous sensor data," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3509–3516, 2020.

[31] S. Liu and M. Zhu, "Distributed inverse constrained reinforcement learning for multi-agent systems," *Advances in Neural Information Processing Systems*, vol. 35, pp. 33444–33456, 2022.

[32] S. S. Hassan, Y. M. Park, Y. K. Tun, W. Saad, Z. Han, and C. S. Hong, "Enhancing spectrum efficiency in 6g satellite networks: A gail-powered policy learning via asynchronous federated inverse reinforcement learning," *arXiv preprint arXiv:2409.18718*, 2024.

[33] W. Gong, L. Cao, Y. Zhu, F. Zuo, X. He, and H. Zhou, "Federated inverse reinforcement learning for smart icus with differential privacy," *IEEE Internet of Things Journal*, vol. 10, no. 21, pp. 19117–19124, 2023.

[34] S. Liu and M. Zhu, "Learning multi-agent behaviors from distributed and streaming demonstrations," *Advances in Neural Information Processing Systems*, vol. 36, pp. 53552–53564, 2023.

[35] P. Sengadu Suresh, Y. Gui, and P. Doshi, "Dec-airl: Decentralized adversarial irl for human-robot teaming," in *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pp. 1116–1124, 2023.

[36] Y. Jiang, W. Yu, D. Song, L. Wang, W. Cheng, and H. Chen, "Fedskill: Privacy preserved interpretable skill learning via imitation," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1010–1019, 2023.

[37] Y. Ye, Y. Zhou, Z. Liu, X. Du, H. Zhou, X. Lian, and M. Chen, "Fitlight: Federated imitation learning for plug-and-play autonomous traffic signal control," *arXiv preprint arXiv:2502.11937*, 2025.

[38] D. A. Tarzanagh, M. Li, C. Thrampoulidis, and S. Oymak, "Fednest: Federated bilevel, minimax, and compositional optimization," in *International Conference on Machine Learning*, pp. 21146–21179, PMLR, 2022.

[39] P. Sahoo, A. Tripathi, S. Saha, and S. Mondal, "Feddual: A dual-strategy with adaptive loss and dynamic aggregation for mitigating data heterogeneity in federated learning," *arXiv preprint arXiv:2412.04416*, 2024.

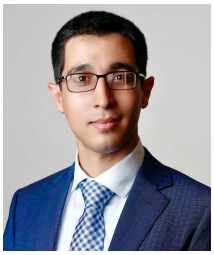
[40] J. So, C. He, C.-S. Yang, S. Li, Q. Yu, R. E Ali, B. Guler, and S. Avestimehr, "Lightsecagg: a lightweight and versatile design for secure aggregation in federated learning," *Proceedings of Machine Learning and Systems*, vol. 4, pp. 694–720, 2022.

[41] S. Hong, X. Lin, and L. Duan, "Lightweight federated learning with differential privacy and straggler resilience," in *IEEE INFOCOM 2025-IEEE Conference on Computer Communications*, pp. 1–10, IEEE, 2025.

[42] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*, pp. 1861–1870, PMLR, 2018.

Guangyu Jiang is a Ph.D. student in Electrical and Computer Engineering at George Washington University. He received his B.S. degree in Mathematics and Physics from Tsinghua University in 2021, and the M.A. degree in Statistics from Columbia University in 2023. His research interests include reinforcement learning, federated learning, and large language models.

Shu Hong is a Postdoctoral Associate in the Department of Electrical and Computer Engineering at George Washington University. She received the Ph.D. degree from Singapore University of Technology and Design, in 2023. Her research interests include trustworthy and efficient learning in edge, networked, and cyber-physical systems. She has received the N2Women Young Researcher Fellowship, the Best Poster/Demo Award at the ACM MobiHoc XR Security Workshop, the President's Graduate Fellowship, SUTD Outstanding Thesis Award, and multiple travel grants.



Mahdi Imani received the Ph.D. degree in Electrical and Computer Engineering from Texas A&M University, College Station, TX, in 2019. He is currently an Assistant Professor in the Department of Electrical and Computer Engineering at Northeastern University. His research areas include reinforcement learning, reasoning under uncertainty, and multi-agent systems. He is the recipient of the NIH NIBIB Trailblazer Award (2022), the NSF CISE CRII Award (2020), and the Oracle Research Award (2022). His work has received several Best Paper Finalist awards, including a spotlight paper at AAAI 2025 and CVPR 2026, the Best Poster/Demo Award at the MobiHoc 2025 XR Systems Workshop, and Best Paper Finalist awards at ACC 2023/2025, SYSID 2024, and Asilomar 2015.

Nathaniel D. Bastian (Senior Member, IEEE) received the Ph.D. degree in Industrial Engineering and Operations Research from Pennsylvania State University, University Park, PA, in 2016. He is currently an Assistant Professor in the Department of Electrical Engineering & Computer Science at the United States Military Academy at West Point, and he serves as Deputy Director of the Robotics Research Center and Principal Investigator of the Laboratory for Artificial Intelligence Research & Engineering (LAIRE). His primary research interests combine mathematical optimization, decision theory, machine learning, and statistical computing to design and develop secure, robust, and resilient AI-enabled autonomous C5ISRT systems. He has received $8M+ in research funding support from DARPA, NSA, OUSW, DEVCOM, AFRL, ONR, etc.

Tian Lan (Fellow, IEEE) received the Ph.D. degree from Princeton University, in 2010. He is currently a full professor of electrical and computer engineering with George Washington University. His research interests include machine learning, network optimization, and algorithms. He received six best paper or runner-up awards (including from IEEE INFOCOM, ACM MobiHoc, IEEE VR, IEEE GLOBECOM, and IEEE Signal Processing Society) and six industry research awards (including from Meta, CISCO, and AT&T), as well as a number of faculty recognition and innovation awards. He served as a member on Federal Communications Commission Technological Advisory Council (FCC TAC), TPC Co-Chair for IEEE INFOCOM 2026, Fellow at National Quantum Lab at UMD, and Associate Editor for IEEE/ACM Transactions on Networking.