2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

On First-Order Meta-Reinforcement Learning with Moreau Envelopes

Mohammad Taha Toghani†, Sebastian Perez-Salazar*, César A. Uribe†

***Abstract*— Meta-Reinforcement Learning (MRL) is a promising framework for training agents that can quickly adapt to new environments and tasks. In this work, we study the MRL problem under the policy gradient formulation, where we propose a novel algorithm that uses Moreau envelope surrogate regularizers to jointly learn a meta-policy that is adjustable to the environment of each individual task. Our algorithm, called Moreau Envelope Meta-Reinforcement Learning (MEMRL), learns a meta-policy that can adapt to a distribution of tasks by efficiently updating the policy parameters using a combination of gradient-based optimization and Moreau Envelope regularization. Moreau Envelopes provide a smooth approximation of the policy optimization problem, which enables us to apply standard optimization techniques and converge to an appropriate stationary point. We provide a detailed analysis of the MEMRL algorithm, where we show a sublinear convergence rate to a first-order stationary point for non-convex policy gradient optimization. We finally show the effectiveness of MEMRL on a multi-task 2D-navigation problem.**

I. Introduction

The Reinforcement Learning (RL) problem [1]–[4] studies the interaction of some agent with an environment to maximize a reward function. In this problem setup, the agent observes the environment's state, selects an action according to a policy, receives a reward and a new state, and updates its policy based on experience [1]. Policy Gradient Reinforcement Learning (PGRL) [5] is a subclass of RL methods that directly optimize the policy by following the gradient of the expected reward with respect to the policy parameters [6]. A neural network or another parametric function usually represents the policy, mapping states to action probabilities [7]. PGRL methods are advantageous because they can handle high-dimensional and continuous action spaces and integrate prior knowledge or structure into the policy [8].

Meta-Reinforcement Learning (MRL) focuses on enabling agents to learn how to learn or adapt rapidly to new tasks and environments [9]–[11]. In MRL, the agent is trained to perform (i.e., a higher reward in expectation) not just on a single task but on a range of tasks, each defined by distinct reward functions, initial state distributions, and transition dynamics [12]. Thus, the agent learns a meta-policy that can quickly adapt to new tasks by modifying its policy or value function based on a few experience samples from the new task [9], [13]. The meta-policy aims to optimize the expected cumulative reward across tasks rather than maximizing the reward for a single task. The MRL framework enables agents to generalize better to new tasks and contexts, making it a crucial area of research.

The primary motivation behind MRL is the need for agents to adapt rapidly to frequently changing environments or tasks, allowing them to become more efficient [14]–[16]. For instance, a robot that operates in various environments [15] or a recommender system that adapts to different user preferences [16]. MRL equips agents with the ability to generalize their skills and knowledge across tasks, allowing them to learn new tasks more efficiently by leveraging past experiences. By enabling agents to learn how to learn, MRL opens new possibilities for intelligent systems to adapt and succeed in complex and dynamic environments.

Some works discuss the fast adaptation in MRL via different techniques [17]–[19]. Ren et al. [17] develop an algorithm that can adapt to new tasks with preference-based feedback from a human oracle. The algorithm uses information theory techniques to design query sequences that maximize the information gained from human interactions while tolerating the inherent error of a non-expert human oracle. Melo [18] presents a method that uses the transformer architecture to mimic the memory reinstatement mechanism. The agent associates the recent past of working memories to build episodic memory recursively through the transformer layers. Zintgraf [19] proposes an MRL framework that can adapt to new tasks by learning a latent task representation and a task-conditioned policy. The framework uses variational inference techniques to infer the task representation from trajectories and optimize the policy with respect to task distribution.

The main challenge in MRL is finding a good representation of the tasks and meta-policy that enables efficient adaptation to new tasks while preserving knowledge learned from past tasks. Beck et al. [20] provide a comprehensive overview of the MRL problem setting, its variations, algorithms, and applications, along with the open challenges for this problem. Furthermore, Yu et al. [21] present a benchmark suite for MRL that includes robotic manipulation tasks with varying difficulty and diversity.

MRL algorithms use different techniques such as gradient-based methods [9], model-based [22] or model-free RL [23], and shared hierarchy (or representation) [24], [25] to tackle these challenges. Frans et al. [24] present MetaSH, an MRL algorithm that learns a shared hierarchy of policies that can generalize across tasks. This method uses a high-level policy that selects sub-policies based on the task context and a low-level policy that executes actions based on the sub-policy.

†Department of Electrical and Computer Engineering, *Computational Applied Mathematics and Operations Research, Rice University, Houston, TX, USA. This work was supported by the National Science Foundation under Grants #2211815 and #2213568. Email addresses: {mttoghani, sperez, cauribe}@rice.edu.

2023 62nd IEEE Conference on Decision and Control (CDC) | 979-8-3503-0124-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/CDC49753.2023.10383874

Finn et al. [9] introduce MAML, another MRL algorithm that learns a model-agnostic initialization that can be fine-tuned with a few gradient steps to new tasks. MAML can be applied to any model trained with gradient descent/ascent and loss functions.

The main challenge faced by Model-Agnostic Meta-Reinforcement Learning (MAMRL) and its variants [9], [10], [26] is the scalability in training. The MAMRL formulation requires computing and storing multiple gradients and Hessians for each task and updating the meta-policy with respect to these parameters. This may incur high computational and memory costs, especially for large-scale or continuous problems [27]. Moreover, the stability of training, which depends on the number of gradient steps, poses another challenge for MAMRL-based methods. Such issues affect the convergence and generalization of MAMRL and require careful tuning/trade-off. Some possible solutions are to use Hessian-free methods [28] or adaptive learning rates [29].

This work studies the MRL problem through gradient-based techniques. We formulate the MRL problem via surrogate cost functions [30], i.e., Moreau Envelope proximal operators [31], [32] and derive the first-order information for this framework building upon the policy gradient problem setup. Our main contribution is *lowering memory requirements* and *reducing arithmetic complexity* by removing the need for second-order information, i.e., a reduction from $\mathcal{O}(d^2)$ to $\mathcal{O}(d)$. We list our contributions as follows:

- We propose a novel framework for MRL via Moreau Envelopes and propose a novel first-order algorithm called MEMRL to maximize the proposed personalized value function.
- We present a detailed convergence analysis of the proposed algorithm under customary assumptions and show a sublinear convergence result for our proposed method.
- We finally present the performance of MEMRL on multi-task 2D-navigation on a discrete grid with a set of finite actions.

The remainder of this paper is arranged as follows. In Section II, we introduce the underlying problem setup, compare with prior works, and present our novel algorithm MEMRL for Meta-Reinforcement Learning with Moreau Envelopes. In Section III, we present the convergence result of our proposed algorithm along with the underlying assumptions. We provide a numerical experiment demonstrating the performance of our method in Section IV. Finally, we conclude the remarks and highlight future works in Section V.

II. Problem Setup & Algorithm

In this section, we first describe the Meta-Reinforcement Learning (MRL) problem setup and discuss the Policy Gradient Reinforcement Learning (PGRL) method via function approximation. Then, we explain the setup for MRL through Moreau Envelope auxiliary cost and a brief comparison with some relevant works. Finally, we present our method MEMRL for the underlying problem.

A. Policy Gradient Meta-Reinforcement Learning

We consider a set of (potentially infinite) Markov Decision Processes (MDPs) $\{\mathcal{M}_i\}_{i\in\mathcal{I}}$ that represent different tasks drawn from a distribution p over a finite time horizon[1] $\{0,1,\dots,H\}$. For each task $i\in\mathcal{I}$, we denote the states and actions by $\mathcal{S}_i$ and $\mathcal{A}_i$, respectively. In this setup, the initial state distribution is given by $\mu_i:\mathcal{S}_i\to\Delta(\mathcal{S}_i)$, where $\Delta(S_i)$ is the set of probability distributions over $\mathcal{S}_i$. Moreover, the transition kernel is denoted by $\mathcal{P}_i$, where $\mathcal{P}_i(s_i'|s_i,a_i)$ is the probability of transitioning from state $s_i\in\mathcal{S}_i$ to $s_i'\in\mathcal{S}_i$ by taking action $a_i\in\mathcal{A}_i$ for which a reward $r_i(s_i,a_i)$ is received according to its corresponding reward function $r_i:\mathcal{S}_i\times\mathcal{A}_i\to[0,R]$. Therefore, the value of a trajectory $\tau_i=(s_i^0,a_i^0,\dots,a_i^{H-1},s_i^H)$ can be defined as

$$\mathcal{R}_i(\tau_i)\coloneqq\sum_{h=0}^{H-1}\gamma^h r_i(s_i^h,a_i^h), \tag{1}$$

where $\gamma\in(0,1)$ is the discount factor for reward accumulation over time. Eventually, each task $i\in\mathcal{I}$ can be modeled as an MDP defined by the tuple $(\mathcal{S}_i,\mathcal{A}_i,\mathcal{P}_i,r_i,\mu_i,\gamma)$. In this setting, a random policy $\pi_i:\mathcal{S}_i\to\Delta(\mathcal{A}_i)$ determines the probability of each action a_i given a state s_i as $\pi_i(a_i|s_i)$. In *Policy Gradient Reinforcement Learning (PGRL)*, we parameterize the policy by a d-dimensional parameter $w\in\mathbb{R}^d$ (like a large neural network), i.e., $\pi_i(\cdot|\cdot;w)$. Therefore, the probability of trajectory τ_i is given by

$$q_i(\tau_i;w)\coloneqq\mu_i(s_i^0)\prod_{h=0}^{H-1}\pi_i(a_i^h|s_i^h;w)\prod_{h=0}^{H-1}\mathcal{P}_i(s_i^{h+1}|s_i^h,a_i^h). \tag{2}$$

Accordingly, the average reward value for each task $i\in\mathcal{I}$ is

$$J_i(w)\coloneqq\mathbb{E}_{\tau_i\sim q_i(\cdot;w)}\left[\mathcal{R}_i(\tau_i)\right], \tag{3}$$

which is a function of parameter w. In multi-task RL problems, we seek to find a joint parameter that maximizes the expected reward on all tasks $\mathcal{I}$:

$$J(w)\coloneqq\mathbb{E}_{i\sim p}\left[J_i(w)\right]. \tag{4}$$

The goal of policy gradient is to find a (sub)optimal parameter that maximizes the expected cumulative reward in (4) obtained by $\pi_i(\cdot|\cdot;w)$, for all $i\in\mathcal{I}$. The key idea behind PGRL is to use the gradient of value function with respect to the policy parameters w to update the policy in the direction that increases the expected cumulative reward.

In multi-task settings with heterogeneous environments, we seek to find a global policy $w\in\mathbb{R}^d$ that performs well by adapting quickly to each task, i.e., obtaining a personal policy $\theta_i\in\mathbb{R}^d$ through fine-tuning. We formulate the joint multi-task setup via *Moreau Envelope Meta-Reinforcement*

[1] A common assumption in RL is that the agent operates in an infinite-horizon setting, where the goal is to maximize the expected discounted or average reward over an infinite number of steps. However, in many practical scenarios, the agent may face a finite-horizon setting, where the goal is to maximize the expected discounted reward over a finite number of steps [33].

Learning cost (MEMRL)

$$\max_{w\in\mathbb{R}^d} V(w) \coloneqq \mathbb{E}_{i\sim p}\left[V_i(w)\right] \tag{5a}$$

$$\text{with} \quad V_i(w) \coloneqq \max_{\theta_i\in\mathbb{R}^d} \left[J_i(\theta_i) - \frac{\lambda}{2}\|\theta_i - w\|^2\right], \tag{5b}$$

where parameter $\lambda{\geq}0$ forms a trade-off on the similarity of policies for different tasks. The formulation in (5) is a bilevel optimization problem. A solution $w^\star \in \mathbb{R}^d$ to Problem (5a) is considered a meta-model that yields a task-personalized parameter $\theta_i^\star$ by maximizing Problem (5b). In the next subsection, we discuss the cost function in (5) and present a novel policy gradient-based algorithm with access to a first-order oracle to maximize this cost.

Related Works: The MRL problem has been studied [10], [11] for multi-task setups with parameter adjustment mainly under Model-Agnostic Meta-Learning (MAML) framework [9], where the goal is to maximize the following cost function:

$$\max_{w\in\mathbb{R}^d} V'(w) \coloneqq \mathbb{E}_{i\sim p}\left[V_i'(w)\right], \tag{6a}$$

$$\text{with} \quad V_i'(w) \coloneqq J_i(w + \alpha\nabla J_i(w)). \tag{6b}$$

This formulation suggests to find an initial policy that performs well after modification with one step of gradient ascent. Finn et al. [11] studied the online meta-learning setting under the MAML setup and Rajeswaran et al. [10] established one of the first theoretical results for this framework. Some other recent works have examined the complexity analysis of MAML in different contexts such as supervised meta-learning [28], [34]. Assuming the inner loop loss function is sufficiently smooth and strongly convex, iMAML converges to a first-order stationary point in the deterministic case. Moreover, works such as [26], [34], [35] study the impact of multi-step fine-tuning, an extended version of (6), under appropriate assumptions. Specifically, [26] establishes the first analysis for multi-step MRL with stochastic gradients via a novel algorithm called SGMRL.

Different variations of MAML setup require access to the second-order information in the underlying update rule for the policy gradient maximization. Moreover, in the RL setups, MAML with one gradient often fails to perform well in practice, hence the choice of multi-step MAML with multiple updates remains a crucial for MRL tasks. In the contrary, Moreau Envelope (ME) [36], [37] controls the trade-off between the similarity and divergence of personal parameters θ_i via the regularization parameter λ. Furthermore, the inner optimization problem can be maximized without the need for second-order information.

B. Meta-Reinforcement Learning with Moreau Envelopes

We start by presenting the first-order information of (3). According to [2], [26], [38], [39], the gradient of $J_i(\cdot)$ can be derived via the logarithm derivative trick as follows:

$$\nabla J_i(w) \coloneqq \mathbb{E}_{\tau_i\sim q_i(\cdot;w)}\left[g_i(\tau_i; w)\right], \tag{7}$$

where the stochastic policy gradient $g_i(\cdot; w)$ is given by

$$g_i(\tau_i; w) \coloneqq \sum_{h=0}^{H-1} \nabla_w \log \pi_i(a_i^h|s_i^h; w)\, \mathcal{R}_i^h(\tau_i), \tag{8a}$$

$$\text{where} \quad \mathcal{R}_i^h(\tau_i) \coloneqq \sum_{l=h}^{H-1} \gamma^l\, r_i(s_i^l, a_i^l). \tag{8b}$$

We drop parameter w from the gradient notation for simplicity of presentation in (8a). In this setup, $g_i(\cdot; w)$ is the score function that measures the sensitivity of the log-probability of the trajectory τ_i to the policy parameters w. Moreover, (1) and (8b) imply $\mathcal{R}_i^0(\tau_i) = \mathcal{R}_i(\tau_i)$.

To deal with the computational intractability of the full gradient in (7), we approximate this term by a stochastic policy gradient over a batch $\mathcal{D}_i$ of trajectories sampled from distribution $q_i(\cdot; w)$, i.e.,

$$\nabla\tilde{J}_i(\mathcal{D}_i; w) \coloneqq \frac{1}{|\mathcal{D}_i|}\sum_{\tau_i\in\mathcal{D}_i} g_i(\tau_i; w), \tag{9}$$

where $\nabla J_i(w){=}\mathbb{E}\left[\nabla\tilde{J}_i(\mathcal{D}_i; w)\right]$.

Next, we present the first-order information of the surrogate function $V_i(\cdot)$ in (5b). To compute the gradient, let

$$\hat{\theta}_i(w) \coloneqq \arg\max_{\theta_i\in\mathbb{R}^d} \left\{J_i(\theta_i) - \frac{\lambda}{2}\|\theta_i - w\|^2\right\}, \tag{10}$$

then, by taking derivative of $V_i(\cdot)$ with respect to w, we have

$$\nabla V_i(w) = \frac{\partial\hat{\theta}_i(w)}{\partial w}\nabla J_i(\hat{\theta}_i(w)) - \lambda\left[\frac{\partial\hat{\theta}_i(w)}{\partial w} - I\right]\left(\hat{\theta}_i(w) - w\right), \tag{11}$$

and due to first-order optimality,

$$\nabla J_i(\hat{\theta}_i(w)) - \lambda\left(\hat{\theta}_i(w) - w\right) = 0 \overset{(11)}{\Rightarrow} \tag{12}$$

$$\nabla V_i(w) = \lambda\left(\hat{\theta}_i(w) - w\right). \tag{13}$$

Therefore, one needs to obtain $\hat{\theta}_i(w)$ for computing $\nabla V_i(w)$. Note that given a set of parameters w, finding $\hat{\theta}_i(w)$ for a general policy function $\pi_i(\cdot|\cdot; w)$ is computationally intractable, since (i) deterministic gradient $\nabla J_i(\cdot)$ cannot be computed and (ii) an exact solution to (10) requires certain assumptions on the function class, e.g., quadratic functions [40]. Hence, we instead propose to maximize the following stochastic approximation with respect to parameter θ_i

$$\tilde{F}_i\left(\mathcal{D}_i; \theta_i, w\right) \coloneqq \tilde{J}_i\left(\mathcal{D}_i; \theta_i\right) - \frac{\lambda}{2}\left\|\theta_i - w\right\|^2, \tag{14}$$

where $\mathcal{D}_i$ is a batch of trajectories sampled from distribution $q_i(\cdot; \theta_i)$. Thus, it is sufficient to find an inexact solution $\tilde{\theta}_i(w)$ that satisfies

$$\left\|\nabla_{\theta_i}\tilde{F}_i\left(\mathcal{D}_i; \tilde{\theta}_i(w), w\right)\right\| \leq \nu, \tag{15}$$

Algorithm 1 MEMRL: First-Order Moreau Envelope Meta-Reinforcement Learning

1: **input:** regularization parameter λ, inexact approximation precision ν, meta stepsize α, task batch size B, trajectory batch size D.
2: **initialize:** $w^0 \in \mathbb{R}^d, t \leftarrow 0$
3: **repeat**
4: sample a batch of tasks $\mathcal{B}^t \subseteq \mathcal{I}$ with size B
5: **for** all tasks $i \in \mathcal{B}^t$ **do**
6: find $\tilde{\theta}_i(w^t)$ such that for a batch of trajectories $\mathcal{D}_i^t$ (of size D) sampled from $q_i(\cdot; \tilde{\theta}_i(w^t))$ to maximize $\tilde{F}_i(\cdot;\cdot,w^t)$ up to accuracy level ν with
$$\left\|\nabla \tilde{F}_i\left(\mathcal{D}_i^t; \tilde{\theta}_i(w^t), w^t\right)\right\| \leq \nu$$
7: **end for**
8: $w^{t+1} \leftarrow (1-\alpha\lambda)w^t + \frac{\alpha\lambda}{|\mathcal{B}^t|}\sum_{i\in\mathcal{B}^t}\tilde{\theta}_i(w^t)$
9: $t \leftarrow t+1$
10: **until** not converged
11: **output:**

for some approximation precision $\nu > 0$, where

$$\nabla_{\theta_i}\tilde{F}_i(\mathcal{D}_i;\theta_i,w) = \nabla\tilde{J}_i(\mathcal{D}_i;\theta_i) - \lambda(\theta_i - w). \tag{16}$$

Then, we can approximate the exact gradient $\nabla V_i(w)$ in (13) with

$$\nabla\tilde{V}_i(w) := \lambda(\tilde{\theta}_i(w) - w), \tag{17}$$

where $\tilde{\theta}_i(w)$ satisfies (15). Note that a small parameter ν provides a better approximation, thus less error in the solution of the algorithm.

We are now ready to propose MEMRL for solving the problem in (5). Algorithm 1 shows the pseudo-code for our method. Starting from a random initial set of parameter w^0, we perform an iterative method. At each round $t \geq 0$, we sample a batch of tasks $\mathcal{B}^t$ with size B and for each task $i \in \mathcal{B}^t$, we maximize $\nabla_{\theta_i}\tilde{F}_i(\cdot;\cdot,w^t)$ up to precision ν (Step 6 of Algorithm 1). Then, we use the approximate individual sub-optimal solutions $\tilde{\theta}_i(w^t)$ to approximate the gradient of $\nabla V_i(w^t)$ according to (17) and use this to aggregate and apply one step of gradient ascent in Step 8 of Algorithm 1.

The inexact optimization solver for the inner problem in Step 6 of Algorithm 1 can be any first-order policy gradient method. For example, let us describe an algorithm for the inexact optimizer:

1) $\theta_i^{t,0} \leftarrow w^t, k \leftarrow 0$,
2) sample a batch of trajectories $\mathcal{D}_i^{t,0}$ with size D with respect to $q_i(\cdot;\theta_i^{t,0})$,
3) While not $\left\|\nabla\tilde{F}_i\left(\mathcal{D}_i^{t,k};\theta_i^{t,k},w^t\right)\right\| \leq \nu$:
   a) sample a batch of trajectories $\mathcal{D}_i^{t,k}$ with size D with respect to $q_i(\cdot;\theta_i^{t,k})$,
   b) $\theta_i^{t,k+1} \leftarrow \theta_i^{t,k} + \beta\left[\nabla\tilde{J}_i(\mathcal{D}_i^{t,k};\theta_i^{t,k}) - \lambda(\theta_i^{t,k} - w^t)\right]$,
   c) $k \leftarrow k+1$,
4) $\tilde{\theta}_i(w^t) \leftarrow \theta_i^{t,k}$.

We will discuss the convergence of Algorithm 1 in the next section.

III. Convergence Result

We start this section by stating the underlying assumption for our analysis. Further, we present two auxiliary lemmas stating the properties of J_i and V_i functions. Finally, we show the convergence result of MEMRL as the main result of this work along with its proof.

Assumption 1 (Log-Probability Properties)**.** *The logarithm of the policy functions π_i are twice differentiable, for all $i \in \mathcal{I}$. Moreover, there exist constants G and L, such that for any task $i \in \mathcal{I}$ and state $s_i \in \mathcal{S}_i$, action $a_i \in \mathcal{A}_i$, and arbitrary parameter $w \in \mathbb{R}^d$,*

$$\|\nabla \log\pi_i(a_i|s_i;w)\| \leq G, \tag{18}$$
$$\left\|\nabla^2 \log\pi_i(a_i|s_i;w)\right\| \leq L. \tag{19}$$

This assumption is conventional in prior works on policy gradient optimization [10], [26], [39], [41], [42]. Particularly, one can see that for Softmax policy [26][Appendix D], which is customary in practice, both (18) and (19) hold. Moreover, recall that the reward functions $r_i(\cdot,\cdot)$ are nonnegative and bounded, i.e., there exists a constant R such that for all $i \in \mathcal{I}, a_i \in \mathcal{A}_i, r_i \in \mathcal{S}_i$, we have $0 \leq r_i(s_i,a_i) \leq R$.

Lemma 1 ([26], Lemma 1, Properties of J_i)**.** *Let Assumption 1 hold. Then, for all $i \in \mathcal{I}$ and $w \in \mathbb{R}^d$, and any batch of trajectories $\mathcal{D}_i$ sampled from distribution $q_i(\cdot;w)$, we have:*

$$\left\|\nabla J_i(w)\right\|, \left\|\nabla\hat{J}_i(\mathcal{D}_i;w)\right\| \leq \hat{G}, \tag{20}$$
$$\left\|\nabla^2 J_i(w)\right\|, \left\|\nabla^2\hat{J}_i(\mathcal{D}_i;w)\right\| \leq \hat{L}, \tag{21}$$

where $\hat{G} := \frac{GR}{(1-\gamma)^2}$ and $\hat{L} := \frac{(HG^2+L)R}{(1-\gamma)^2}$.

Lemma 1 indicates gradient boundedness and smoothness for the customary value function J_i, for each task $i \in \mathcal{I}$. Note that this work defines the trajectory up to H actions starting from s_0. Hence, the value $\hat{L}$ of is slightly different from [26]. By modifying the proof of Lemma 1, we may replace $\frac{1}{1-\gamma}$ with $\min\{\frac{1}{1-\gamma}, H\}$ due to the fact that the tasks are MDPs with finite time horizons.

Lemma 2 (Properties of V_i)**.** *Let Assumption 1 hold and $\lambda \geq \kappa\hat{L}$ for some κ>1, and $\hat{G}, \hat{L}$ as in Lemma 1. Then, for all $i \in \mathcal{I}$ and $w, v \in \mathbb{R}^d$, the following properties hold:*

$$\left\|\nabla V_i(w)\right\| \leq \hat{G}, \tag{22}$$
$$\left\|\nabla V_i(w) - \nabla V_i(v)\right\| \leq \tilde{L}\left\|w - v\right\|, \tag{23}$$

where $\tilde{L} := \frac{\lambda}{\kappa-1}$.

This lemma suggests that the Moreau Envelope surrogate value function in (5) has similar properties as the value function in (3). The upper bound on the gradient is the same, and the smoothness parameter depends on the regularization term λ. Also note that the global expected value function $V(\cdot)$ has similar boundedness and smoothness properties as each $V_i(\cdot)$, according to the definition in (5a).

We are now ready to present our main technical result.

Theorem 1 (MEMRL Convergence)**.** *Let Assumption 1 hold, $\lambda > \hat{L}$, and $\alpha = \frac{1}{4\tilde{L}}$. Then for any timestep $T \geq 4\tilde{L}^2$, the following property holds for the iterates of Algorithm 1:*

$$\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla V(w^t)\|^2 \leq \frac{8R}{(1-\gamma)\sqrt{T}} + \frac{\lambda^2\nu^2}{(\lambda-\hat{L})^2} + \frac{8\tilde{L}\hat{G}^2}{B\sqrt{T}} + \frac{8\tilde{L}\lambda^2\nu^2}{(\lambda-\hat{L})^2 B\sqrt{T}} + \frac{8\alpha\tilde{L}\lambda^2\hat{G}^2}{(\lambda-\hat{L})^2 BD\sqrt{T}},$$

where $\hat{G}, \hat{L}$ as in Lemma 1, and $\tilde{L}$ as in Lemma 2.

Theorem 1 implies a sublinear convergence rate for MEMRL algorithm to a first-order stationary point with oracle complexity $\mathcal{O}(1/\sqrt{T}) + \mathcal{O}(\nu^2)$. In other words, to reach a complexity of $\mathcal{O}(\varepsilon)$, it is sufficient to run the algorithm for $T = \mathcal{O}(1/\varepsilon^2)$ iterations via an inexact inner solver with precision $\nu = \mathcal{O}(\sqrt{\varepsilon})$.

The proofs for Lemma 2 and Theorem 1 can be found in [43].

IV. Numerical Experiment

In this section, we present numerical studies of our proposed MEMRL algorithm. We consider a discrete variation of 2D-navigation problem [9], [26], [44], [45] over a square grid, where the objective is to reach a specified destination by taking valid actions. The MRL setup for this problem narrates the scenario where the goal is to obtain a meta policy that can be easily adapted to perform well for multiple destinations.

Now, let us describe the problem setup. We consider a group of $|\mathcal{I}| = 3$ tasks where the objective of each task $i \in \mathcal{I}$ is to navigate to some corresponding destination $s_i^\star = \{x_i, y_i\}$ over an 11×11 grid, $\{-5, \ldots, 5\} \times \{-5, \ldots, 5\}$, starting from some random point. For example, check the left subfigure in Figure 1, where each star represents the destination locations for some task $i \in [3]$. The set of valid actions is limited to left, right, down, up, and pause,

$$\mathcal{A}_i = \{(0,0), (0,1), (0,-1), (1,0), (-1,0)\}. \tag{24}$$

The set of states $\mathbf{S}_i = \{-5, \ldots, 5\} \times \{-5, \ldots, 5\}$. We define the reward function inversely proportional [45] to the ℓ_1 distance of the agent's next state to the intended destination, i.e.,

$$r_i(a_i^h | s_i^h) = \exp(-\|s_i^{h+1} - s_i^\star\|_1), \tag{25}$$

where s_i^{h+1} is the next coordinate of the agent after taking action a_i^h. Finally, we parameterize the policy with a two-layer MLP network with softmax layer and 5 outputs by taking a two-dimensional input, i.e., the state of the agent yields a probability vector of the potential actions.

We implement Step 6 of Algorithm 1 via the first-order inexact optimizer which we described in Section II under fixed inner loop with K=8 steps. Moreover, we select $\lambda = 2$, $\alpha = 0.1$, $\beta = 0.02$, $\gamma = 0.99$, $B = 2$, and $D = 10$. The right subfigure of Figure 1 illustrates the performance of MEMRL for the 2D-navigation problem. We plot the stochastic reward function $\tilde{J}_i$ for each task across the optimization runtime. Moreover, we show present the underlying navigation before and after training with out method respectively in the left and middle subfigures of Figure 1.

V. Conclusions

We studied the Meta-Reinforcement Learning (MRL) problem and introduced MEMRL. This novel meta-reinforcement learning algorithm leverages Moreau Envelopes to achieve fast and stable policy adaptation with first-order optimization. We proved the convergence of our algorithm under non-convex policy gradient optimization and showed its performance over a discrete 2D-navigation problem with no need to second-order information. Our work opens up new directions for applying Moreau Envelopes to other meta-learning problems to resolve the scalability challenges for Hessian computation. A thorough study of the Multi-Agent MRL problem remains a future study for this work. We also leave the extended analysis for infinite horizon MDPs to future studies.

References

[1] C. Szepesvári, "Algorithms for reinforcement learning," *Synthesis lectures on artificial intelligence and machine learning*, vol. 4, no. 1, pp. 1–103, 2010.

[2] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[3] A. Agarwal, N. Jiang, S. M. Kakade, and W. Sun, "Reinforcement learning: Theory and algorithms," *CS Dept., UW Seattle, Seattle, WA, USA, Tech. Rep*, pp. 10–4, 2019.

[4] R. Liu and A. Olshevsky, "Temporal difference learning as gradient splitting," in *International Conference on Machine Learning*. PMLR, 2021, pp. 6905–6913.

[5] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," *Advances in neural information processing systems*, vol. 12, 1999.

[6] H. P. Van Hasselt, M. Hessel, and J. Aslanides, "When to use parametric models in reinforcement learning?" *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[7] J. Clifton and E. Laber, "Q-learning: Theory and applications," *Annual Review of Statistics and Its Application*, vol. 7, pp. 279–301, 2020.

[8] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.

[9] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International conference on machine learning*. PMLR, 2017, pp. 1126–1135.

[10] A. Rajeswaran, C. Finn, S. M. Kakade, and S. Levine, "Meta-learning with implicit gradients," *Advances in neural information processing systems*, vol. 32, 2019.

[11] C. Finn, A. Rajeswaran, S. Kakade, and S. Levine, "Online meta-learning," in *International Conference on Machine Learning*. PMLR, 2019, pp. 1920–1930.

[12] A. Nagabandi, I. Clavera, S. Liu, R. S. Fearing, P. Abbeel, S. Levine, and C. Finn, "Learning to adapt in dynamic, real-world environments through meta-reinforcement learning," *arXiv preprint arXiv:1803.11347*, 2018.

[13] R. Dorfman, I. Shenfeld, and A. Tamar, "Offline meta reinforcement learning–identifiability challenges and effective data collection strategies," *Advances in Neural Information Processing Systems*, vol. 34, pp. 4607–4618, 2021.

[14] I. Dasgupta, J. Wang, S. Chiappa, J. Mitrovic, P. Ortega, D. Raposo, E. Hughes, P. Battaglia, M. Botvinick, and Z. Kurth-Nelson, "Causal reasoning from meta-reinforcement learning," *arXiv preprint arXiv:1901.08162*, 2019.

[15] G. Schoettler, A. Nair, J. A. Ojea, S. Levine, and E. Solowjow, "Meta-reinforcement learning for robotic industrial insertion tasks," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9728–9735.

[16] Y. Wang, Y. Ge, L. Li, R. Chen, and T. Xu, "Offline meta-level model-based reinforcement learning approach for cold-start recommendation," *arXiv preprint arXiv:2012.02476*, 2020.

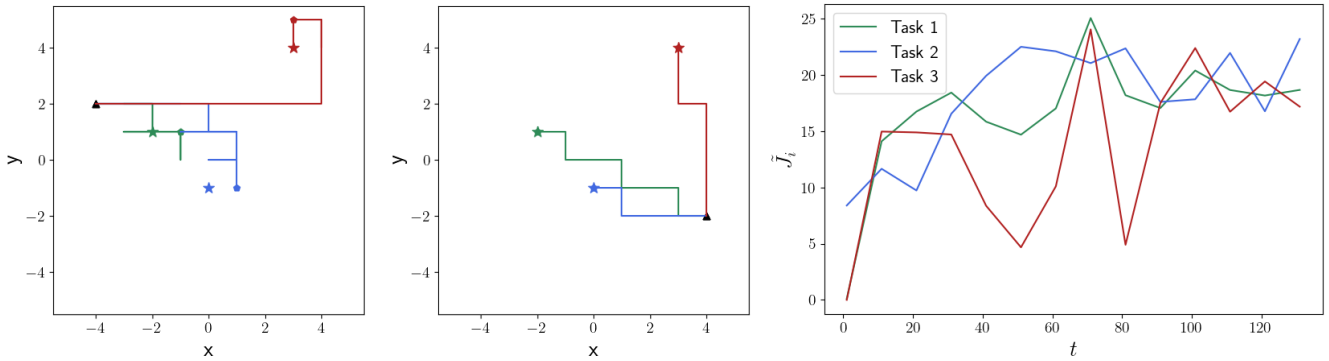

Fig. 1: The performance of our MEMRL algorithm on discrete 2D-navigation for $|\mathcal{I}|$=3 tasks with different underlying MDPs. **(Left)** The navigation map at iteration $t = 0$ starting from a random location (black triangle) on the grid. The stars indicate the destination of each task $i \in \mathcal{I}$. Pentagons indicate the end of a trajectory when it fails to reach its destination (star). **(Middle)** The navigation map at iteration $t = 120$, where the adapted meta-policy for each task is optimal. **(Right)** The evolution of individual reward functions given the adapted meta-policy on each task. Each curve is the empirical mean of the reward obtain over 10 independent trajectories conditioned on the approximated policy parameter $\tilde{\theta}_i^t$.

[17] Z. Ren, A. Liu, Y. Liang, J. Peng, and J. Ma, "Efficient meta reinforcement learning for preference-based fast adaptation," *arXiv preprint arXiv:2211.10861*, 2022.

[18] L. C. Melo, "Transformers are meta-reinforcement learners," in *International Conference on Machine Learning*. PMLR, 2022, pp. 15 340–15 359.

[19] L. Zintgraf, "Fast adaptation via meta reinforcement learning," Ph.D. dissertation, University of Oxford, 2022.

[20] J. Beck, R. Vuorio, E. Z. Liu, Z. Xiong, L. Zintgraf, C. Finn, and S. Whiteson, "A survey of meta-reinforcement learning," *arXiv preprint arXiv:2301.08028*, 2023.

[21] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine, "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning," in *Conference on robot learning*. PMLR, 2020, pp. 1094–1100.

[22] I. Clavera, J. Rothfuss, J. Schulman, Y. Fujita, T. Asfour, and P. Abbeel, "Model-based reinforcement learning via meta-policy optimization," in *Conference on Robot Learning*. PMLR, 2018, pp. 617–629.

[23] L. Li, Y. Huang, M. Chen, S. Luo, D. Luo, and J. Huang, "Provably improved context-based offline meta-rl with attention and contrastive learning," *arXiv preprint arXiv:2102.10774*, 2021.

[24] K. Frans, J. Ho, X. Chen, P. Abbeel, and J. Schulman, "Meta learning shared hierarchies," *arXiv preprint arXiv:1710.09767*, 2017.

[25] S. Zhang, L. Shen, and L. Han, "Learning meta representations for agents in multi-agent reinforcement learning," *arXiv preprint arXiv:2108.12988*, 2021.

[26] A. Fallah, K. Georgiev, A. Mokhtari, and A. Ozdaglar, "On the convergence theory of debiased model-agnostic meta-reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[27] J. Shin, H. B. Lee, B. Gong, and S. J. Hwang, "Large-scale meta-learning with continual trajectory shifting," *arXiv preprint arXiv:2102.07215*, 2021.

[28] A. Fallah, A. Mokhtari, and A. Ozdaglar, "On the convergence theory of gradient-based model-agnostic meta-learning algorithms," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 1082–1092.

[29] Y. Jiang, J. Konečnỳ, K. Rush, and S. Kannan, "Improving federated learning personalization via model agnostic meta learning," *arXiv preprint arXiv:1909.12488*, 2019.

[30] J. Kim, P. Toulis, and A. Kyrillidis, "Convergence and stability of the stochastic proximal point algorithm with momentum," in *Learning for Dynamics and Control Conference*. PMLR, 2022, pp. 1034–1047.

[31] J.-J. Moreau, "Proximité et dualité dans un espace hilbertien," *Bulletin de la Société mathématique de France*, vol. 93, pp. 273–299, 1965.

[32] N. Parikh and S. Boyd, "Proximal algorithms. number 3 in foundations and trends in optimization," 2013.

[33] V. VP, D. Bhatnagar *et al.*, "Finite horizon q-learning: Stability, convergence, simulations and an application on smart grids," *arXiv preprint arXiv:2110.15093*, 2021.

[34] K. Ji, J. Yang, and Y. Liang, "Multi-step model-agnostic meta-learning: Convergence and improved algorithms," *arXiv preprint arXiv:2002.07836*, vol. 2, 2020.

[35] M. T. Toghani, S. Lee, and C. A. Uribe, "Pars-push: Personalized, asynchronous and robust decentralized optimization," *IEEE Control Systems Letters*, vol. 7, pp. 361–366, 2022.

[36] C. Dinh, N. Tran, and T. Nguyen, "Personalized Federated Learning with Moreau Envelopes," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 394–21 405, 2020.

[37] M. T. Toghani, S. Lee, and C. A. Uribe, "PersA-FL: Personalized Asynchronous Federated Learning," *arXiv preprint arXiv:2210.01176*, 2022.

[38] J. Peters and S. Schaal, "Reinforcement learning of motor skills with policy gradients," *Neural networks*, vol. 21, no. 4, pp. 682–697, 2008.

[39] Z. Shen, A. Ribeiro, H. Hassani, H. Qian, and C. Mi, "Hessian aided policy gradient," in *International conference on machine learning*. PMLR, 2019, pp. 5729–5738.

[40] Z. Charles and J. Konečnỳ, "Convergence and accuracy trade-offs in federated learning and meta-learning," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 2575–2583.

[41] M. Papini, D. Binaghi, G. Canonaco, M. Pirotta, and M. Restelli, "Stochastic variance-reduced policy gradient," in *International conference on machine learning*. PMLR, 2018, pp. 4026–4035.

[42] A. Agarwal, S. M. Kakade, J. D. Lee, and G. Mahajan, "Optimality and approximation with policy gradient methods in markov decision processes," in *Conference on Learning Theory*. PMLR, 2020, pp. 64–66.

[43] M. Toghani, S. Perez-Salazar, and C. Uribe, "On first-order meta-reinforcement learning with moreau envelopes," *arXiv preprint arXiv:2305.12216*, 2023.

[44] P. Henderson, W.-D. Chang, F. Shkurti, J. Hansen, D. Meger, and G. Dudek, "Benchmark environments for multitask learning in continuous domains," *ICML Lifelong Learning: A Reinforcement Learning Approach Workshop*, 2017.

[45] J. Rothfuss, D. Lee, I. Clavera, T. Asfour, and P. Abbeel, "Promp: Proximal meta-policy search," *arXiv preprint arXiv:1810.06784*, 2018.