Enhancing Online Reinforcement Learning with Meta-Learned Objective from Offline Data

Shilong Deng[1], Zetao Zheng[1,2], Hongcai He[1], Paul Weng[3], Jie Shao[1,2*]

[1]University of Electronic Science and Technology of China, Chengdu, China
[2]Sichuan Artificial Intelligence Research Institute, Yibin, China
[3]Data Science Research Center, Duke Kunshan University, Kunshan, China
{sldeng, hehongcai}@std.uestc.edu.cn, {ztzheng, shaojie}@uestc.edu.cn, paul.weng@dukekunshan.edu.cn

Abstract

A major challenge in Reinforcement Learning (RL) is the difficulty of learning an optimal policy from sparse rewards. Prior works enhance online RL with conventional Imitation Learning (IL) via a handcrafted auxiliary objective, at the cost of restricting the RL policy to be sub-optimal when the offline data is generated by a non-expert policy. Instead, to better leverage valuable information in offline data, we develop Generalized Imitation Learning from Demonstration (GILD), which meta-learns an objective that distills knowledge from offline data and instills intrinsic motivation towards the optimal policy. Distinct from prior works that are exclusive to a specific RL algorithm, GILD is a flexible module intended for diverse vanilla off-policy RL algorithms. In addition, GILD introduces no domain-specific hyperparameter and minimal increase in computational cost. In four MuJoCo tasks with sparse rewards, we show that three RL algorithms enhanced with GILD significantly outperform state-of-the-art methods.

Introduction

Reinforcement Learning (RL), which learns through trial and error experience to maximize the cumulative reward, has achieved great success in various dense reward tasks (Wang et al. 2023; Wu et al. 2023). However, RL agents still struggle to learn the optimal policy from real-world scenarios with sparse rewards. For instance, there might be a reward only if a navigation robot reaches the goal, with no reward feedback on the numerous intermediate steps taken to arrive.

To address the challenge of sparse rewards, prior works improve online RL with conventional Imitation Learning (IL) by guiding the agent to acquire reward signals that are essential for policy improvement (Mendonca et al. 2019; Fujimoto and Gu 2021; Rengarajan et al. 2022a). These RL+IL methods augment RL with conventional IL via a handcrafted auxiliary objective, which constrains the agent to stay close to behaviors observed in offline demonstration data. However, striking a balance between RL and IL remains intractable, especially when the agent is fed with sub-optimal demonstrations generated by humans. As shown in Figure 1, conventional IL guides the agent to obtain reward signals in early stage, but restricts the learned policy to be sub-optimal in later stage. This observation leads to the following research question: *Is it possible to leverage sub-optimal offline demonstrations for viable online RL with sparse rewards, while not restricting the policy to be sub-optimal?* A natural answer is to manually control or decay the influence of imitation on policy optimization with some pre-defined schedule, but at the cost of either spending massive time on hyperparameter tuning or being exclusive to a specific RL algorithm (Fujimoto and Gu 2021; Rengarajan et al. 2022a,b).

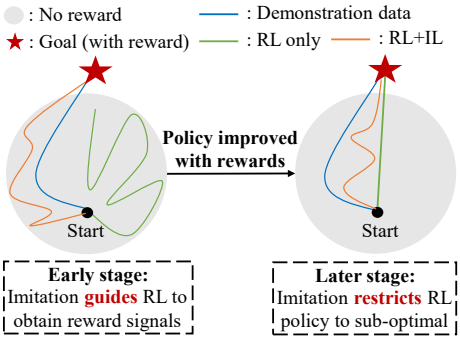

Figure 1: Illustration of RL+IL with sparse rewards. Conventional IL guides RL to obtain reward signals in early stage (left), while restricting RL policy to be sub-optimal in later stage (right).

By contrast, our key insight is to enhance online RL with a meta-learned objective that leverages valuable information in sub-optimal offline demonstrations, instead of RL with a handcrafted objective in conventional IL. To achieve this, we develop Generalized Imitation Learning from Demonstration (GILD), a flexible module intended for diverse vanilla off-policy RL algorithms. We devise a novel bi-level optimization framework for RL algorithms enhanced with GILD, with meta-optimization of GILD at the upper level and meta-training of RL at the lower level supported by the meta-learned objective. We select off-policy RL as the vanilla algorithm due to its superior sample efficiency compared with on-policy alternatives. The advantage of sample efficiency extends to meta-optimization, which updates GILD such that the policy learned with RL+GILD is superior to that with RL+IL. We emphasize that, in contrast to prior works that either augment RL with a handcrafted IL

*Corresponding author: Jie Shao.

objective or are exclusive to a specific RL algorithm, GILD meta-learns a general IL objective and is intended for diverse vanilla off-policy RL algorithms.

Our main results are as follows:

i. GILD meta-learns a general IL objective to enhance online RL via distilling knowledge from offline demonstrations, rather than relying on a handcrafted IL objective in conventional IL. To the best of our knowledge, GILD is the first to meta-learn an objective to deal with sparse rewards.

ii. We integrate GILD with three vanilla off-policy RL algorithms (DDPG (Lillicrap et al. 2016), TD3 (Fujimoto, van Hoof, and Meger 2018), and SAC (Haarnoja et al. 2018)) and evaluate them on four challenging MuJoCo tasks with sparse rewards. Experiments show that the RL+GILD methods not only outperform the vanilla RL methods and the conventional RL+IL variants, but also attain asymptotic performance to the optimal policy.

iii. To further analyze the impact of GILD, we present several visualizations including trajectories in a goal-reaching task and parameter optimization paths in MuJoCo. These visualizations demonstrate the aptitude of GILD at distilling knowledge from sub-optimal demonstrations and instilling intrinsic motivation that guides the RL agent towards the optimal policy.

iv. Finally, we observe that GILD converges exceptionally fast, making it feasible to utilize RL+GILD at a few warm-start (e.g., 1% of total) time steps and subsequently drop GILD (RL only) to speed up training. This highlights the potential to enhance RL with minimal computational cost while achieving significant improvement.

Related Work

Our work is mainly related to RL+IL, single-task meta-RL and objective learning, which we discuss below. The closest methods to our approach are LOGO (Rengarajan et al. 2022b) (RL+IL) and Meta-Critic (Zhou et al. 2020) (objective learning).

RL+IL. We focus on reinforcement learning enhanced with imitation learning (RL+IL) under sparse rewards, with the key idea of utilizing demonstrations to assist policy learning. Prior works have sought to (i) explicitly imitate behavior with demonstrations to accelerate standard RL learning (Mendonca et al. 2019; Fujimoto and Gu 2021) or guide the RL agent towards non-zero reward regions of state-action spaces (Rengarajan et al. 2022a), (ii) distill the information within the demonstrations into an implicit prior (Singh et al. 2021; Hakhamaneshi et al. 2022) or combine multiple explicit and implicit priors obtained from demonstrations (Yan, Schwing, and Wang 2022), and (iii) obtain guidance from implicit imitation via aligning with the behavior policy measured by KL-divergence (Rengarajan et al. 2022b). These methods strike a balance between RL and IL at the cost of either spending massive time on hyperparameter tuning or being exclusive to a specific RL algorithm. Distinct from prior works, we propose a flexible module named GILD, which is intended for diverse vanilla online RL algorithms, to distill knowledge from offline demonstrations with a meta-learned objective.

Single-task meta-RL. With the aim of accelerating learning or improving performance, single-task meta-RL can meta-learn various RL components, including (i) discount factor in scalar form (Xu, van Hasselt, and Silver 2018) or vector form (Yin, Yan, and Xu 2023), (ii) reward function as an additive intrinsic reward from data collected by RL (Zheng, Oh, and Singh 2018) or as the entire rewards from human preference data (Liu et al. 2022) and (iii) weights for training samples to achieve better task awareness in model-based RL (Yuan et al. 2023). By contrast, our proposed GILD meta-learns a general IL objective from offline demonstrations and automatically strikes a balance between RL and IL.

Objective learning. Different from the aforementioned works that employ only a common objective function, objective learning in RL or supervised learning aims to learn an objective. The learned objective function has been exploited to (i) provide guidance for accelerate learning in standard RL (Xu et al. 2019, 2020; Zhou et al. 2020), (ii) teach the training of a student RL model (Wu et al. 2018; Fan et al. 2018; Huang et al. 2019; Hai et al. 2023), and (iii) improve generalization or robustness to novel tasks with different dynamics (Baik et al. 2021; Jin et al. 2023; Neyman and Roughgarden 2023). Replacing conventional IL objective, our approach enhances online RL with a meta-learned objective from offline demonstrations.

Preliminaries

Standard RL. Reinforcement learning typically considers an infinite horizon Markov Decision Process (MDP), which is represented as a tuple $< \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma >$, with state space $\mathcal{S}$, action space $\mathcal{A}$, reward function $\mathcal{R}$, transition dynamics $\mathcal{P}$, and discount factor γ. At each timestep, given state $s \in \mathcal{S}$, an RL agent takes action $a \in \mathcal{A}$ based on its policy ϕ, and receives reward $r = \mathcal{R}(s, a)$ and new state s' following the transition dynamics $p(s'|s, a) \in \mathcal{P}$. The objective function of policy ϕ, known as the expected return, is defined as $\mathcal{L}^{RL}(\phi) = -\mathbb{E}_{s\sim p, a\sim\phi}[\sum_{t=0}^{\infty} \gamma^t r_t]$. With a bit abuse of notation, we use ϕ to refer to both stochastic and deterministic policy, as GILD is proposed for RL algorithms with both stochastic policy (SAC) and deterministic policy (DDPG and TD3).

Off-policy RL usually measures the objective with an actor-critic architecture for superior sample efficiency via reusing past experience (s, a, r, s') stored in the replay buffer $\mathcal{D}$. The critic parameterized by θ, learns an action-value function, which is defined as $Q_\theta(s, a) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_{t+1} | s_0 = s, a_0 = a]$, to evaluate the expected return following policy ϕ starting from state s and action a. The critic is updated to minimize the Mean-Square Bellman Error (MSBE) function:

$$\begin{aligned}\theta^* &= \arg\min_\theta \mathcal{L}^{\text{MSBE}}(\theta) \\ &= \arg\min_\theta \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\Big[Q_\theta(s, a) - \Big(r + \gamma Q_\theta\big(s', \phi(s')\big)\Big)\Big].\end{aligned} \tag{1}$$

Algorithm 1: RL+GILD

Input: Actor ϕ, critic θ, GILD ω, demonstration data $\mathcal{D}^{\text{dem}}$, and empty replay buffer $\mathcal{D}$

while not converging **do**
  Collect data from the environment and store in $\mathcal{D}$;
  meta-training:
  Sample (s, a, r, s') from $\mathcal{D}$, and $(s^{\text{d}}, a^{\text{d}})$ from $\mathcal{D}^{\text{dem}}$;
  Update critic θ via Eq. (5);
  Pseudo-update actor $\hat{\phi}$ with RL+IL via Eq. (6);
  Update actor ϕ with RL + GILD via Eq. (7);
  meta-optimization:
  Update GILD ω via Eq. (11);
end while

The policy ϕ, known as the actor, is updated to minimize the loss given by the critic:

$$\phi^* = \arg\min_{\phi} \mathcal{L}_{\theta}^{\text{RL}}(\phi) = \arg\min_{\phi} \mathop{\mathbb{E}}_{s\sim\mathcal{D}} \Big[- Q_{\theta}\big(s, \phi(s)\big)\Big]. \quad (2)$$

RL+IL. The most commonly used form of IL is Behaviour Cloning (BC), which focuses on imitating behaviors in demonstration data $\mathcal{D}^{\text{dem}}$ using supervised learning. The supervised learning objective for it is defined as $\mathcal{L}^{\text{IL}}(\phi) = N^{-1}\sum_{(s,a)\in\mathcal{D}^{\text{dem}}}(\phi(s) - a)^2$ for the deterministic policy and $\mathcal{L}^{\text{IL}}(\phi) = -N^{-1}\sum_{(s,a)\in\mathcal{D}^{\text{dem}}}\log(\pi_{\phi}(a|s))$ for the stochastic policy. Recent online RL approaches (Mendonca et al. 2019; Fujimoto and Gu 2021; Rengarajan et al. 2022a) utilize IL as an auxiliary objective added to the update steps of an RL policy, to push the policy towards behaviors in demonstrations:

$$\phi^* = \arg\min_{\phi} \big(\mathrm{w}_{\text{rl}}\mathcal{L}^{RL}(\phi) + \mathrm{w}_{\text{il}}\mathcal{L}^{IL}(\phi)\big), \quad (3)$$

where w_{rl} and w_{il} are hyperparameters that control the influence of RL and IL on policy optimization.

Methodology

In this section, we present off-policy RL augmented by GILD, which is formalized as a bi-level optimization framework, with (i) meta-optimization of GILD at the upper level and (ii) meta-training of RL at the lower level supported by the meta-learned objective. Following notations in off-policy RL and meta-RL, we denote the parameters of actor, critic, and GILD network as ϕ, θ, and ω respectively. We denote the objective learned by GILD ω as $\mathcal{L}_{\omega}^{\text{GILD}}(\phi)$, whose input depends on actor parameter ϕ.

Overview

GILD aims to enhance online RL with a meta-learned objective $\mathcal{L}_{\omega}^{\text{GILD}}(\phi)$ that distills knowledge from sub-optimal offline demonstrations, rather than relying on conventional IL via supervised learning. More specially, GILD is updated with meta-loss $\mathcal{L}_{\theta}^{\text{meta}}(\phi)$, which optimizes GILD in the direction that the policy learned with RL+GILD is superior to policy with RL+IL. See Algorithm 1 for a pseudocode of bi-level paradigm and Figure 2 for a workflow of bi-level

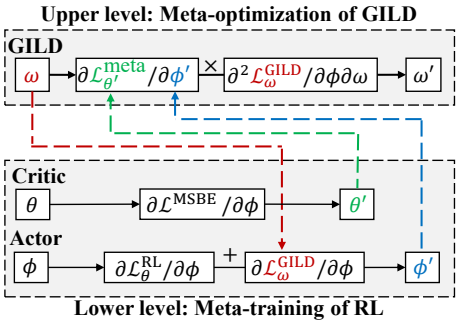

Figure 2: Workflow of the bi-level optimization framework, with meta-optimization of GILD at the upper level and meta-training of RL at the lower level supported by $\mathcal{L}_{\omega}^{\text{GILD}}$.

optimization. The overall objective is formulated as:

$$\min_{\omega} \quad \mathcal{L}_{\theta^*}^{\text{meta}}(\phi^*),$$
$$s.t. \quad \begin{cases} \phi^* = \arg\min_{\phi} \big(\mathcal{L}_{\theta^*}^{\text{RL}}(\phi) + \mathcal{L}_{\omega}^{\text{GILD}}(\phi)\big), \\ \theta^* = \arg\min_{\theta} \big(\mathcal{L}^{\text{MSBE}}(\theta)\big), \end{cases} \quad (4)$$

where meta-training at the lower level includes conventional critic learning and policy learning supported by GILD. Thanks to meta-optimization of GILD at the upper level, the policy learned with RL+GILD could be superior to that learned with RL+IL in Eq. (3). This bi-level optimization enables GILD to distill knowledge from offline data and instills in the online RL agent the intrinsic motivation towards optimal policy, thus not restricting RL policy to be sub-optimal.

General Imitation Learning Objective

As discussed, a policy trained using non-expert demonstrations via Eq. (3) is restricted to be sub-optimal. We address this issue with general IL objective, which enhances RL by leveraging valuable information in sub-optimal demonstrations $\mathcal{D}^{\text{dem}}$. The supervised learning objective function for IL can be formalized as $\mathcal{L}^{\text{IL}}(\phi) = f(\phi; \mathcal{D}^{\text{dem}})$, with a handcrafted loss function $f(\cdot)$ (e.g., mean square error), which restricts agent around the behavior policy. We devise GILD as a neural network parametrized by ω to meta-learn a general update function $f_{\omega}(\cdot)$, which produces a general IL objective $\mathcal{L}_{\omega}^{\text{GILD}}(\phi) = f_{\omega}(\phi; \mathcal{D}^{\text{dem}})$.

We implement GILD as a three-layer fully connected network for the following considerations: (i) GILD should be flexible to be integrated with diverse vanilla off-policy RL algorithms; (ii) For the feasibility to be applied to downstream tasks (e.g., use convolutional neural networks as GILD's backbone for image-based autonomous driving task), GILD ought to introduce no domain-specific hyperparameter; (iii) GILD is supposed to enhance off-policy RL without reducing the superior sample efficiency.

Building connection between lower-level and upper-level. (i) Upper-to-lower: To update the RL policy, the general IL objective $\mathcal{L}_{\omega}^{\text{GILD}}(\cdot)$ outputted by GILD must be differentiable w.r.t. policy parameter ϕ, which means the input of GILD should depend on the actor. This is satisfied in an end-to-end manner: GILD takes the combination

of demonstration state-action pair $(s^{\mathrm{d}}, a^{\mathrm{d}})$ and actor's action $a = \phi(s^{\mathrm{d}})$ as the input. (ii) Lower-to-upper: To update GILD, the meta-loss, which is the action-value function $Q_\theta(\cdot)$ for sample efficiency consideration, must be differentiable w.r.t. GILD parameter ω. As depicted in Figure 2, the connection between θ and ω is built as follows. First, θ is differentiable w.r.t. ϕ since $Q_\theta\big(s, \phi(s)\big)$ takes action $\phi(s)$ as the input. Second, ϕ is differentiable w.r.t. ω since it is updated with $\mathcal{L}_\omega^{\mathrm{GILD}}(\phi)$. Therefore, θ is differentiable w.r.t. ω.

Bi-Level Optimization

After defining general IL objective $\mathcal{L}_\omega^{\mathrm{GILD}}(\phi)$ and building a connection for bi-level optimization, we divide the bi-level objective in Eq. (4) into meta-training (lower-level) and meta-optimization (upper-level) to solve them respectively. Note that we omit tricks (e.g., target network and entropy regularizer) used in different off-policy algorithms here for simplicity. Details for three vanilla off-policy RL algorithms enhanced with GILD are presented in Deng et al. (2025).

Lower-level: meta-training. After collecting a set $\mathcal{D}$ of transitions (s, a, r, s') through interacting with the environment, off-policy RL reuses these past experiences to update critic and actor sequentially. The critic is updated with a batch of N transitions to minimize the MSBE function as:

$$\theta^{(k+1)} = \theta^{(k)} - \alpha \nabla_\theta \frac{1}{N} \sum_{(s,a,r,s') \sim \mathcal{D}} \Big[Q_\theta(s,a) - \Big(r + \gamma Q_\theta\big(s', \phi(s')\big)\Big)\Big]^2 \Big|_{\theta^{(k)}, \phi^{(k)}}, \tag{5}$$

where $\theta^{(k+1)}$ denotes the updated parameter $\theta^{(k)}$ at step k, α is the learning rate, and γ is the discount factor.

Before updating the actor, we *pseudo-update* the actor with RL+IL. The pseudo-updated actor is intended for computing the meta-loss later, which guides the policy learned with RL+GILD to be potentially superior to that with RL+IL. Pseudo-update means that we do not directly update actor $\phi^{(k)}$, but update a copy of the current actor $\hat{\phi}^{(k)}$:

$$\hat{\phi}^{(k+1)} = \hat{\phi}^{(k)} - \alpha \nabla_{\hat{\phi}} \Big[\mathrm{w_{rl}} \frac{1}{N} \sum_{(s,a) \sim \mathcal{D}} -Q_\theta\big(s, \hat{\phi}(s)\big) + \mathrm{w_{il}} \mathcal{L}^{IL}(\hat{\phi}) \Big] \Big|_{\theta^{(k+1)}, \hat{\phi}^{(k)}}, \tag{6}$$

where α is the learning rate, $\mathcal{L}^{IL}(\hat{\phi})$ is the conventional IL objective used in Eq. (3), and $\mathrm{w_{rl}}$ and $\mathrm{w_{il}}$ are hyperparameters that control the influence of RL and IL on policy optimization. Following TD3+BC (Fujimoto and Gu 2021), an approach for off-policy RL+IL, we assign the hyperparameters as $\mathrm{w_{rl}} = \beta / \frac{1}{N} \sum_{s,a} |Q_\theta(s,a)|$ and $\mathrm{w_{il}} = 1$ for off-policy RL+IL baselines in our experiment, with β=2.5 provided by the authors. Following EMRLD (Rengarajan et al. 2022a), an approach for on-policy RL+IL, we set $\mathrm{w_{rl}} = 1$ and $\mathrm{w_{il}} = 1$ for on-policy RL+IL baselines.

After the pseudo-update, the actor is updated to minimize both objectives given by the critic and GILD:

$$\phi^{(k+1)} = \phi^{(k)} - \alpha \nabla_{\phi} \Big[\frac{1}{N} \sum_{(s,a) \sim \mathcal{D}} -Q_\theta\big(s, \phi(s)\big) + \mathcal{L}_\omega^{GILD}(\phi) \Big] \Big|_{\theta^{(k+1)}, \phi^{(k)}, \omega^{(k)}}, \tag{7}$$

where $\mathcal{L}_\omega^{\mathrm{GILD}}(\phi) = N^{-1} \sum f_\omega\big(s^{\mathrm{d}}, a^{\mathrm{d}}, \phi(s^{\mathrm{d}})\big)$ with a batch of N state-action pairs $(s^{\mathrm{d}}, a^{\mathrm{d}})$ sampled from demonstrations $\mathcal{D}^{\mathrm{dem}}$.

Upper-level: meta-optimization. The intuition of the meta-loss is to update GILD ω in the direction that the policy learned with RL+GILD is superior to that with RL+IL. This superiority could be measured quantitatively by the difference in action-value function $Q_\theta(\cdot)$ as:

$$\mathcal{L}_\theta^{\mathrm{meta}}(\phi) = \frac{1}{N} \sum_{s^{\mathrm{val}} \sim \mathcal{D}} \Big[\tanh\Big(Q_\theta\big(s^{\mathrm{val}}, \phi(s^{\mathrm{val}})\big) - Q_\theta\big(s^{\mathrm{val}}, \hat{\phi}(s^{\mathrm{val}})\big)\Big)\Big] \Big|_{\theta^{(k+1)}, \phi^{(k+1)}, \hat{\phi}^{(k+1)}, \omega^{(k)}}, \tag{8}$$

where s^{val} is the validation states sampled from past experiences for sample efficiency consideration, ϕ is from performing Eq. (7), and $\hat{\phi}$ is from performing Eq. (6). The derivative of $\mathcal{L}_\theta^{\mathrm{meta}}(\phi)$ w.r.t. ω is calculated using the chain rule:

$$\begin{aligned} \frac{\partial \mathcal{L}_\theta^{\mathrm{meta}}(\phi)}{\partial \omega} &= \frac{\partial \mathcal{L}_\theta^{\mathrm{meta}}(\phi)}{\partial \phi} \cdot \frac{\partial \phi}{\partial \omega} \Big|_{\theta^{(k+1)}, \phi^{(k+1)}, \omega^{(k)}} \\ &= \frac{\partial \mathcal{L}_\theta^{\mathrm{meta}}(\phi)}{\partial \phi} \cdot g_\omega^{(k)} \Big|_{\theta^{(k+1)}, \phi^{(k+1)}, \omega^{(k)}}, \end{aligned} \tag{9}$$

where $g_\omega^{(k)}$, which is actually the second-order derivative, can be obtained as follows. Since $Q(\cdot)$ in Eq. (7) is a constant c that is independent of ω, we simplify $\phi^{(k+1)}$ and $g_\omega^{(k)}$ as:

$$\begin{aligned} \phi^{(k+1)} &= \phi^{(k)} - \alpha \frac{\partial \mathcal{L}_\omega^{\mathrm{GILD}}(\phi)}{\partial \phi} + c \Big|_{\phi^{(k)}, \omega^{(k)}}, \\ g_\omega^{(k)} &= \frac{\partial \phi^{(k+1)}}{\partial \omega} = -\alpha \frac{\partial^2 \mathcal{L}_\omega^{\mathrm{GILD}}(\phi)}{\partial \phi \partial \omega} \Big|_{\phi^{(k)}, \omega^{(k)}}. \end{aligned} \tag{10}$$

Combining Eq. (9) and Eq. (10), we get the derivative w.r.t. ω. Then, ω is meta-optimized as:

$$\omega^{(k+1)} = \omega^{(k)} + \alpha^2 \frac{\partial \mathcal{L}_\theta^{\mathrm{meta}}(\phi)}{\partial \phi} \Big|_{\phi^{(k+1)}, \omega^{(k)}} \cdot \frac{\partial^2 \mathcal{L}_\omega^{\mathrm{GILD}}(\phi)}{\partial \phi \partial \omega} \Big|_{\phi^{(k)} \omega^{(k)}}. \tag{11}$$

Experiments

Research questions. Our experiments are designed to investigate the following research questions: **RQ1**: What is the enhancement of RL+GILD compared with RL+IL and objective learning methods? **RQ2**: How does GILD enhance RL compared with conventional IL? **RQ3**: What are the effects of different meta-loss designs and warm-start steps for GILD? **RQ4**: How to mitigate the computational cost of GILD?

Benchmarks and vanilla RL algorithms. We conduct experiments on four challenging MuJoCo tasks with sparse rewards. Following EMRLD (Rengarajan et al. 2022a), the agent gets a reward only after it has moved a certain number of units along the correct direction, making the rewards sparse. We take three popular off-policy RL algorithms as our vanilla algorithms, which are DDPG (Lillicrap et al. 2016), TD3 (Fujimoto, van Hoof, and Meger 2018), and

Algorithm	Hopper-v2	Walker2d-v2	HalfCheetah-v2	Ant-v2	Point2D Navigation
DDPG	2122.9±590.7	1519.4±881.8	3349.1±1489.6	339.0±109.2	19.7±13.3
DDPG+IL	2378.4±906.1	1867.8±489.5	5603.9±1129.9	575.1±215.4	47.5±16.8
DDPG+GILD (ours)	2804.0±235.4	2632.1±373.0	9987.7±511.9	971.6±296.7	71.0±8.7
TD3	1320.8±413.9	1426.6±1413.0	3251.3±1135.4	1712.3±562.8	24.0±10.7
TD3+IL	2437.9±890.2	2488.5±903.7	5843.9±1321.0	2660.2±395.5	55.8±13.8
TD3+GILD (ours)	**3538.6±104.6**	4113.6±280.5	9997.6±754.9	4864.6±699.1	75.1±9.7
SAC	2235.1±569.6	1643.2±809.5	3946.2±485.2	2106.8±718.0	43.6±17.2
SAC+IL	2989.6±263.3	3102.1±476.5	6503.2±802.5	3370.8±466.3	67.1±14.4
SAC+GILD (ours)	3470.6±85.2	**4840.4±243.8**	**11161.5±552.6**	**5335.3±246.9**	**79.8±6.5**
PPO	1332.5±1356.33	6.3±13.3	-10.3±514.2	637.6±191.3	23.6±15.5
PPO+IL	1831.7±279.8	2649.5±86.8	2781.3±61.7	1759.8±7.7	43.2±8.6
LOGO	3465.80±88.2	4537.5±293.4	5264.0±486.5	4589.5±992.9	77.8±8.0
Meta-Critic	3185.2±526.9	3807.0±1377.1	6811.6±3981.0	1588.9±782.8	68.6±23.7
DiffAIL	2494.3±77.5	2848.3±153.4	5978.6±237.0	3650.1±183.9	51.3±4.1

Table 1: Comparison on max average return of three vanilla off-policy RL algorithms, RL+IL and RL+GILD, along with (on-policy or state-of-the-art) methods. Results are run on sparse environments over 5 trials, and "±" captures the standard deviation over trials. Max value for each category is underlined, and max value overall is in bold.

SAC (Haarnoja et al. 2018). We use open-source implementations of "OurDDPG"[1], TD3[2], and SAC[3].

Baselines. In addition to the above three vanilla RL algorithms and their RL+IL variants, we run the following (state-of-the-art) algorithms using either author-provided or open-source implementation: (i) **LOGO**: We re-run Learning Online with Guidance Offline (LOGO) (Rengarajan et al. 2022b), which merges TRPO (on-policy RL) with an additional policy step using sub-optimal demonstration data. (ii) **Meta-Critic**: We re-run Meta-Critic (Zhou et al. 2020), which meta-learns an additional objective for off-policy RL. (iii) **DiffAIL**: We re-run Diffusion Adversarial Imitation Learning (DiffAIL) (Wang et al. 2024), which introduces the diffusion model into adversarial IL. (iv) **PPO** and **PPO+IL**: We re-run PPO (Schulman et al. 2017) and its RL+IL variant to compare with on-policy RL. (v) **Expert** and **Behavior**: Following LOGO (Rengarajan et al. 2022b), we train vanilla RL algorithms in the dense reward environment to provide three Expert baselines. We use the partially trained Expert that is still at a sub-optimal stage as the Behavior baselines to provide demonstration data for the corresponding RL+IL and RL+GILD algorithms.

Implementation details. To ensure a fair and identical experimental evaluation across algorithms, we train the (RL+IL and RL+GILD) variant using the same hyperparameters as their vanilla algorithms and introduce no domain-specific parameters. We train off-policy algorithms for 1 million steps with sparse rewards and evaluate them every 5000 steps with dense rewards. On-policy algorithms are trained with more steps (e.g., 30 million) to ensure convergence. Results are averaged over five random seeds and the standard deviation is shown with the shaded region or error bar. Our code is available at https://github.com/slDeng1003/GILD.

[1]https://github.com/sfujim/TD3/blob/master/OurDDPG.py
[2]https://github.com/sfujim/TD3/blob/master/TD3.py
[3]https://github.com/pranz24/pytorch-soft-actor-critic

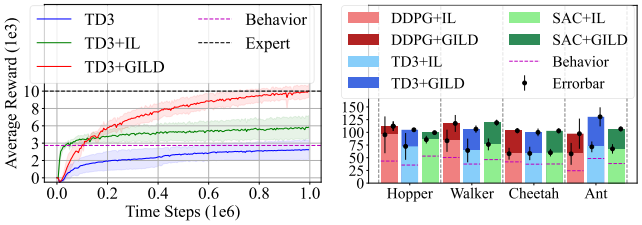

Figure 3: Learning curve with mean-std in the HalfCheetah task (left) and average normalized score (right) in the MuJoCo task(s) with sparse rewards. We normalized the scores using max average return of Expert (with a score of 100).

RQ1: Comparison w. RL+IL & Objective Learning

The max average returns for all methods are summarized in Table 1. We display the most representative learning curve of vanilla TD3 algorithms with its corresponding TD3+IL and TD3+GILD variants in Figure 3, and more learning curves are in Deng et al. (2025). Besides, Figure 3 presents the average normalized score of vanilla algorithms, their corresponding variants, and Behavior algorithms. Scores are normalized using the max average return of Expert (with a score of 100).

In all four benchmarks, our RL+GILD methods significantly outperform the other baselines, while vanilla algorithms fail in most cases due to the sparsity of reward. Learning curve of TD3+IL rises quickly in the initial stage of learning, indicating the agent obtains non-zero rewards via imitation, which underscores the necessity of imitating demonstrations. However, the policy learned by TD3+IL is restricted to be sub-optimal, while TD3+GILD smoothly surpasses the Behavior policy and attain asymptotic or superior performance to the Expert policy, emphasizing the benefit of leveraging insights from sub-optimal demonstrations. LOGO exhibits comparable performance to RL+GILD across several tasks, albeit with noticeably lower

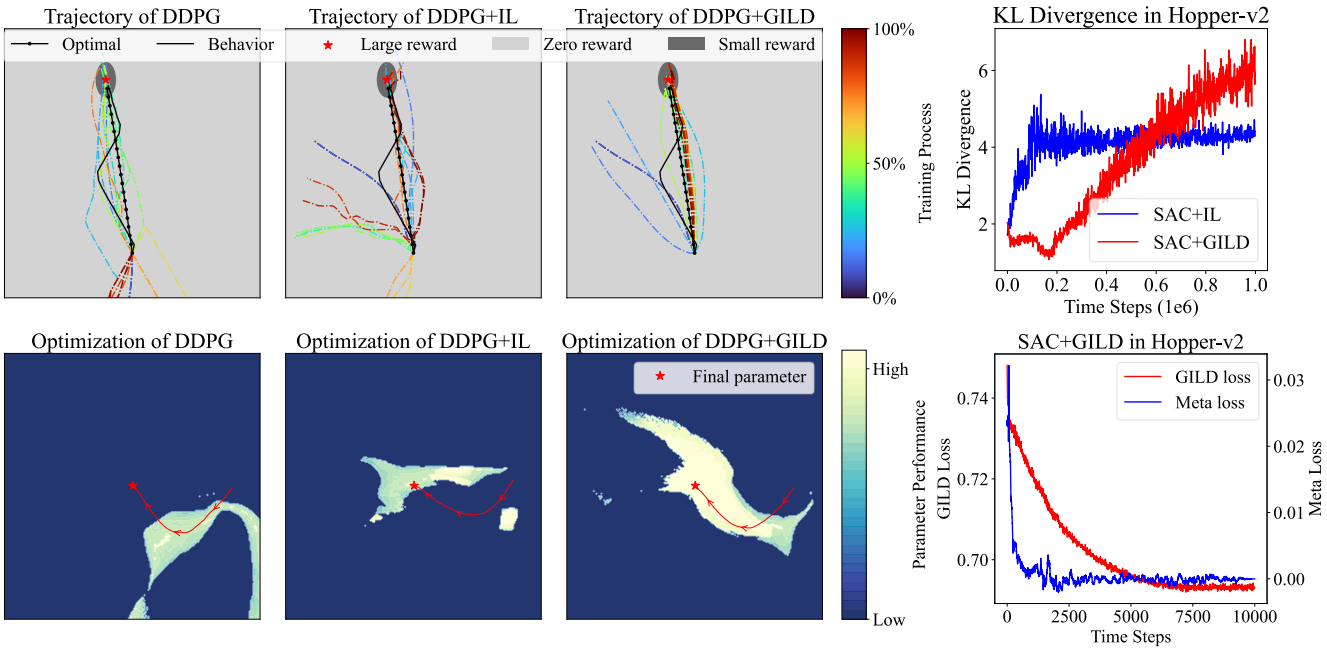

Figure 4: (i) Left: Visualization of evaluation trajectories and corresponding policy optimization paths for DDPG, DDPG+IL, DDPG+GILD in Point2D Navigation. The red star denotes the goal to reach, as well as parameters for the final policy. (ii) Right: KL divergence and loss analysis for SAC+IL and SAC+GILD.

sample efficiency and slower learning speed as shown in Table 3. Meta-Critic achieves commendable performance in a subset of benchmarks, although it struggles to reach the Expert performance due to its inability to utilize information in demonstrations. DiffAIL does not attain good metrics because it relies heavily on the quality of offline data collected by sub-optimal policy. More results for RQ1 are in Deng et al. (2025).

RQ2: Visualization and Loss Analysis

To investigate how GILD enhances the vanilla RL algorithms compared with conventional IL, we (i) visualize the evaluation trajectories and corresponding optimization paths of DDPG, DDPG+IL and DDPG+GILD, (ii) display the KL divergence of SAC+IL and SAC+GILD with the Behavior policy, and (iii) plot the value of general IL objective and meta-loss to demonstrate the convergence of GILD. Following Meta-Critic, curves are uniformly smoothed for clarity. Further visualization results are in Deng et al. (2025).

(i) Trajectory visualization: We run DDPG, DDPG+IL and DDPG+GILD in Point2D Navigation (Rengarajan et al. 2022a), a 2-dimensional goal-reaching environment with $|\mathcal{S}|$=2, $|\mathcal{A}|$=2. We plot the trajectories of the on-learning model at each evaluation at the top-left of Figure 4, and different training periods serve as colors of each trajectory. On the one hand, trajectories of DDPG+IL in the early stage are quite similar to the Behavior trajectories, indicating that the agent quickly learns a policy close to the Behavior policy via imitation. However, trajectories of DDPG+IL in the later stage deviate to the wrong direction towards the goal (red star), due to the incongruity between RL and conventional IL. On the other hand, DDPG+GILD eliminates the incongruity by leveraging the valuable information in demonstrations, with trajectories consistently resemble the optimal after the initial stage.

(ii) Optimization path visualization: Corresponding to the aforementioned trajectories, we display policy optimization paths (red line with arrow) in the parameter space at the bottom-left of Figure 4. Following network visualization in Li et al. (2018), we apply principal component analysis to reduce the dimension of policy parameter ϕ, and take the top-2 representative components for plotting on the 2D surface. Every point on the surface represents a policy. These policies are densely evaluated over 10 episodes to get the average reward values, which serve as colors of the points. The policy optimization paths demonstrate that DDPG+GILD moves directly and quickly to the high reward area (brighter color) on the surface, while the vanilla DDPG and DDPG+IL struggle to move beyond the low reward area (darker color) and finally learn a sub-optimal or bad policy.

(iii) KL divergence analysis: The stochastic policy in SAC provides feasibility to calculate the KL divergence between the learning policy and the Behavior policy. We display it at the top-right of Figure 4 and find that policy learned by SAC+IL is constrained to be similar to Behavior due to the handcrafted objective. By contrast, the policy learned by SAC+GILD leverages knowledge distilled from demonstrations and moves beyond the Behavior policy with consistently rising KL divergence after the early stage.

(iv) Loss analysis: We plot values of general IL objective $\mathcal{L}_{\omega}^{\text{GILD}}$ and meta-loss $\mathcal{L}_{\theta}^{\text{meta}}$ at the bottom-right of Figure 4, which demonstrates that GILD converges exception-

Algorithm	Meta-loss in Eq. (8)	Intuitive meta-loss
DDPG+GILD	**971.6+-296.7**	883.1+-254.8
TD3+GILD	**4864.6+-699.1**	4259.4+-716.3
SAC+GILD	**5335.3+-246.9**	4851.0+-218.5

Table 2: Ablation study on different designs of meta-loss applied to three RL+GILD methods in the sparse Ant benchmark. Max value for each method is in bold.

ally quickly (within 1% of total steps) under the supervision of meta-loss. Meta-loss drops rapidly around zero after 1000 steps, verifying that GILD has distilled most of the knowledge in demonstrations from $t_{\text{ws}} \times B \div N \approx 640$ times of processing each data, where t_{ws}=10000 is the warm-start steps, B=256 is the batch size, and $N \approx 4000$ is the number of samples. As we will discuss later in RQ3 and RQ4, GILD's rapid convergence indicates that we can utilize GILD with a few warm-start (ws) steps (e.g., 1% of total steps) and subsequently drop GILD to speed up training.

RQ3: Ablation on Meta-Loss and Warm-Start

(i) Ablation on meta-loss design: As discussed in Methodology, $Q_\theta(\hat{\phi})$ in the meta-loss is independent to GILD parameter ω, so the most intuitive meta-loss is defined as $\mathcal{L}_\theta^{\text{meta}}(\phi) = \mathbb{E}[Q_\theta(\phi)]$. This intuitive meta-loss aims to maximize the performance of policy ϕ updated with GILD ω. We evaluate these two meta-loss designs in the most challenging sparse Ant-v2 benchmark ($|\mathcal{S}| = 111$, $|\mathcal{A}| = 8$) and report the max average return in Table 2. We find that GILD with meta-loss in Eq. (8) outperforms GILD with the intuitive meta-loss, which also improves vanilla RL algorithms.

(ii) Ablation on GILD warm-start (ws): As discussed in RQ2, GILD converges quickly with a few warm-start (1% of total) steps. To investigate the influence of warm-start on the performance, we implement different warm-start steps on the RL+GILD methods. Training of a policy learned by RL+GILD+1%ws is split into two training stages: (i) RL+GILD stage: during 0%-1% steps, we train policy with RL+GILD, where GILD has not converged; (ii) RL-only stage: during 1%-100% steps, we train policy with vanilla RL, where GILD has converged.

For example, DDPG+GILD+1%ws trains the policy with RL+GILD at 0%-1% of total steps and with vanilla DDPG at 1%-100% of total steps. Figure 5 shows the max average return for three RL+GILD methods trained with different warm-start steps in the sparse Ant-v2 benchmark. Overall, GILD converges within 1% of total steps and improves slightly with more steps, indicating GILD's great potential to enhance RL with minimal computational cost.

RQ4: Computational Efficiency Analysis

We evaluate the average run time of training each algorithm to convergence over four MuJoCo tasks, using either author-provided or open-source implementations. The results are reported in Table 3. Unsurprisingly, on-policy algorithms take a longer time to converge due to lower sample efficiency than off-policy algorithms, especially for

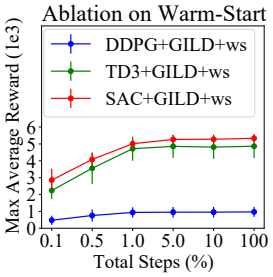

Figure 5: Ablation on warm-start steps. GILD converges within 1% of total steps.

Algorithm	Off-policy			On-policy	
	DDPG	TD3	SAC	PPO	LOGO
Vanilla RL	**1h58m**	**2h13m**	**4h40m**	23h51m	67h36m
RL+IL	2h58m	3h6m	6h4m	26h31m	-
RL+MC	7h23m	7h58m	15h47m	-	-
RL+GILD	4h21m	4h35m	9h24m	-	-
RL+GILD+1%ws	2h5m	2h18m	4h59m	-	-

Table 3: Average run time comparison for all methods over four MuJoCo tasks, "1%ws" denotes 1% (of total training steps) as warm-start steps, and "-" denotes no such a combination. Off-policy methods with the shortest time are in bold, and the second shortest are underlined.

LOGO which calculates KL-divergence at each time step. Although RL+GILD takes longer training time than RL+IL, RL+GILD with 1% (of total) warm-start steps significantly reduces training time, while achieving superior performance (as shown in Figure 5). Overall, vanilla RL algorithms enhanced with GILD warm-start take less than half of the computational cost of these (state-of-the-art) off-policy and on-policy algorithms. We recommend 1% (of total training steps) as warm-start steps for a minimal increase in computational cost while significantly improving performance. In more complex tasks, GILD might converge slower due to a larger amount of offline data and a higher dimensionality of data (e.g., image data).

Conclusion

We develop GILD, a flexible module that meta-learns a general imitation learning objective function from offline data to enhance diverse vanilla off-policy RL algorithms with sparse rewards. Introducing no domain-specific hyperparameter and minimal increase in computational cost, GILD is intended for diverse vanilla off-policy RL algorithms. We show that RL+GILD significantly improve upon baselines in four challenging environments.

Limitation and future work. GILD is conceived within the single-task meta-RL framework, which necessitates RL agents to learn from scratch upon encountering unseen tasks. This inherently limits the extensibility of GILD to few-shot learning scenarios. In future work, we plan to evolve GILD into the multi-task meta-RL framework, thereby addressing challenges in few-shot learning paradigms.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 62276047 and No. 62176154), National Foreign Expert Project of China (No. H20240938) and Sichuan Science and Technology Program (No. 2025HJRC0021).

References

Baik, S.; Choi, J.; Kim, H.; Cho, D.; Min, J.; and Lee, K. M. 2021. Meta-Learning with Task-Adaptive Loss Function for Few-Shot Learning. In *2021 IEEE/CVF International Conference on Computer Vision*, 9445–9454.

Deng, S.; Zheng, Z.; He, H.; Weng, P.; and Shao, J. 2025. Enhancing Online Reinforcement Learning with Meta-Learned Objective from Offline Data. *CoRR*, abs/2501.07346.

Fan, Y.; Tian, F.; Qin, T.; Li, X.; and Liu, T. 2018. Learning to Teach. In *6th International Conference on Learning Representations*.

Fujimoto, S.; and Gu, S. S. 2021. A Minimalist Approach to Offline Reinforcement Learning. In *Advances in Neural Information Processing Systems, 34*, 20132–20145.

Fujimoto, S.; van Hoof, H.; and Meger, D. 2018. Addressing Function Approximation Error in Actor-Critic Methods. In *Proceedings of the 35th International Conference on Machine Learning*, 1582–1591.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *Proceedings of the 35th International Conference on Machine Learning*, 1856–1865.

Hai, Z.; Pan, L.; Liu, X.; Liu, Z.; and Yunita, M. 2023. L2T-DLN: Learning to Teach with Dynamic Loss Network. In *Advances in Neural Information Processing Systems, 36*.

Hakhamaneshi, K.; Zhao, R.; Zhan, A.; Abbeel, P.; and Laskin, M. 2022. Hierarchical Few-Shot Imitation with Skill Transition Models. In *The Tenth International Conference on Learning Representations*.

Huang, C.; Zhai, S.; Talbott, W.; Bautista, M. Á.; Sun, S.; Guestrin, C.; and Susskind, J. M. 2019. Addressing the Loss-Metric Mismatch with Adaptive Loss Alignment. In *Proceedings of the 36th International Conference on Machine Learning*, 2891–2900.

Jin, T.; Liu, J.; Rouyer, C.; Chang, W.; Wei, C.; and Luo, H. 2023. No-Regret Online Reinforcement Learning with Adversarial Losses and Transitions. In *Advances in Neural Information Processing Systems, 36*.

Li, H.; Xu, Z.; Taylor, G.; Studer, C.; and Goldstein, T. 2018. Visualizing the Loss Landscape of Neural Nets. In *Advances in Neural Information Processing Systems, 31*, 6391–6401.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2016. Continuous control with deep reinforcement learning. In *4th International Conference on Learning Representations*.

Liu, R.; Bai, F.; Du, Y.; and Yang, Y. 2022. Meta-Reward-Net: Implicitly Differentiable Reward Learning for Preference-based Reinforcement Learning. In *Advances in Neural Information Processing Systems, 35*.

Mendonca, R.; Gupta, A.; Kralev, R.; Abbeel, P.; Levine, S.; and Finn, C. 2019. Guided Meta-Policy Search. In *Advances in Neural Information Processing Systems, 32*, 9653–9664.

Neyman, E.; and Roughgarden, T. 2023. No-Regret Learning with Unbounded Losses: The Case of Logarithmic Pooling. In *Advances in Neural Information Processing Systems, 36*.

Rengarajan, D.; Chaudhary, S.; Kim, J.; Kalathil, D.; and Shakkottai, S. 2022a. Enhanced Meta Reinforcement Learning via Demonstrations in Sparse Reward Environments. In *Advances in Neural Information Processing Systems, 35*.

Rengarajan, D.; Vaidya, G.; Sarvesh, A.; Kalathil, D. M.; and Shakkottai, S. 2022b. Reinforcement Learning with Sparse Rewards using Guidance from Offline Demonstration. In *The Tenth International Conference on Learning Representations*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. *CoRR*, abs/1707.06347.

Singh, A.; Liu, H.; Zhou, G.; Yu, A.; Rhinehart, N.; and Levine, S. 2021. Parrot: Data-Driven Behavioral Priors for Reinforcement Learning. In *9th International Conference on Learning Representations*.

Wang, B.; Wu, G.; Pang, T.; Zhang, Y.; and Yin, Y. 2024. DiffAIL: Diffusion Adversarial Imitation Learning. In *Thirty-Eighth AAAI Conference on Artificial Intelligence*, 15447–15455.

Wang, T.; Torralba, A.; Isola, P.; and Zhang, A. 2023. Optimal Goal-Reaching Reinforcement Learning via Quasimetric Learning. In *International Conference on Machine Learning*, 36411–36430.

Wu, J.; Ma, H.; Deng, C.; and Long, M. 2023. Pre-training Contextualized World Models with In-the-wild Videos for Reinforcement Learning. In *Advances in Neural Information Processing Systems, 36*.

Wu, L.; Tian, F.; Xia, Y.; Fan, Y.; Qin, T.; Lai, J.; and Liu, T. 2018. Learning to Teach with Dynamic Loss Functions. In *Advances in Neural Information Processing Systems, 31*, 6467–6478.

Xu, K.; Ratner, E.; Dragan, A. D.; Levine, S.; and Finn, C. 2019. Learning a Prior over Intent via Meta-Inverse Reinforcement Learning. In *Proceedings of the 36th International Conference on Machine Learning*, 6952–6962.

Xu, Z.; van Hasselt, H.; and Silver, D. 2018. Meta-Gradient Reinforcement Learning. In *Advances in Neural Information Processing Systems, 31*, 2402–2413.

Xu, Z.; van Hasselt, H. P.; Hessel, M.; Oh, J.; Singh, S.; and Silver, D. 2020. Meta-Gradient Reinforcement Learning with an Objective Discovered Online. In *Advances in Neural Information Processing Systems, 33*.

Yan, K.; Schwing, A. G.; and Wang, Y. 2022. CEIP: Combining Explicit and Implicit Priors for Reinforcement Learning with Demonstrations. In *Advances in Neural Information Processing Systems, 35*.

Yin, H.; Yan, S.; and Xu, Z. 2023. Distributional Meta-Gradient Reinforcement Learning. In *The Eleventh International Conference on Learning Representations*.

Yuan, H.; Dou, H.; Jiang, X.; and Deng, Y. 2023. Task-aware world model learning with meta weighting via bi-level optimization. In *Advances in Neural Information Processing Systems, 36*.

Zheng, Z.; Oh, J.; and Singh, S. 2018. On Learning Intrinsic Rewards for Policy Gradient Methods. In *Advances in Neural Information Processing Systems, 31*, 4649–4659.

Zhou, W.; Li, Y.; Yang, Y.; Wang, H.; and Hospedales, T. M. 2020. Online Meta-Critic Learning for Off-Policy Actor-Critic Methods. In *Advances in Neural Information Processing Systems, 33*.