Multi-task Hierarchical Adversarial Inverse Reinforcement Learning

Jiayu Chen [1] Dipesh Tamboli [2] Tian Lan [3] Vaneet Aggarwal [1 2 4]

Abstract

Multi-task Imitation Learning (MIL) aims to train a policy capable of performing a distribution of tasks based on multi-task expert demonstrations, which is essential for general-purpose robots. Existing MIL algorithms suffer from low data efficiency and poor performance on complex long-horizontal tasks. We develop Multi-task Hierarchical Adversarial Inverse Reinforcement Learning (MH-AIRL) to learn hierarchically-structured multi-task policies, which is more beneficial for compositional tasks with long horizons and has higher expert data efficiency through identifying and transferring reusable basic skills across tasks. To realize this, MH-AIRL effectively synthesizes context-based multi-task learning, AIRL (an IL approach), and hierarchical policy learning. Further, MH-AIRL can be adopted to demonstrations without the task or skill annotations (i.e., state-action pairs only) which are more accessible in practice. Theoretical justifications are provided for each module of MH-AIRL, and evaluations on challenging multi-task settings demonstrate superior performance and transferability of the multi-task policies learned with MH-AIRL as compared to SOTA MIL baselines.

1. Introduction

The generalist robot, which can autonomously perform a wide range of tasks, is one of the essential targets of robotic learning. As an important approach, Imitation Learning (IL) enables the agent to learn policies based on expert demonstrations and is especially effective for problems where it's difficult to discover task solutions autonomously through Reinforcement Learning (RL). To train a general-purpose agent, Multi-task/Meta Imitation Learning (MIL) algorithms (Finn et al., 2017b; Deisenroth et al., 2014; Singh et al., 2020) have been proposed to learn a parameterized policy that is a function of both the current observation and the task and is capable of performing a range of tasks following a particular distribution. The key insight of these algorithms is that the successful control for one task can be informative for other related tasks. However, a critical challenge for them is to acquire enough data for the agent to generalize broadly across tasks. Typically, a large number of demonstrations are required for each task in that distribution, and the required amount increases with task difficulty. Moreover, the learned multi-task policy cannot be transferred to tasks out of that distribution (Yu et al., 2019; Ghasemipour et al., 2019), which limits its general use.

Hierarchical Imitation Learning (HIL) has the potential to reduce the required demonstrations. In HIL, the agent learns a two-level policy, which can be modeled with the option framework (Sutton et al., 1999), from the expert data. Specifically, the low-level policies (i.e., skills) are designated to accomplish certain subtasks in a complex task, while the high-level policy is for scheduling the switch among the skills to solve the entire task. For multi-task settings, learning a hierarchical policy enables the agent to identify basic skills that can be useful in solving a distribution of tasks and to transfer them across tasks during training. In this case, each skill can be trained with demonstrations from different tasks rather than limited to a single one, and, with the shared skills, an agent mainly needs to update its high-level policy rather than learning an entire policy for each task. The expert data efficiency is significantly improved since demonstrations among different tasks are reused for learning skills and the burden of multi-task policy learning becomes lower. Further, in RL and IL, hierarchies exhibit a number of benefits, including better performance on long-horizontal complex tasks (Florensa et al., 2017; Jing et al., 2021) and the possibility of skill transfer between distinct tasks (Andreas et al., 2017).

In this paper, we propose MH-AIRL to introduce hierarchies to MIL. As discussed above, such hierarchies can improve expert data efficiency so that the agent can achieve supe-

[1] School of Industrial Engineering, Purdue University, West Lafayette, IN 47907, USA [2] Elmore Family School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, USA [3] Department of Electrical and Computer Engineering, George Washington University, Washington DC 20052, USA [4] Department of Computer Science and AI Initiative, King Abdullah University of Science and Technology, Thuwal 23955, KSA. Correspondence to: Jiayu Chen <chen3686@purdue.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

rior performance based on a limited number of demonstrations. Further, basic skills can be extracted from the learned policies and reused in out-of-distribution tasks for better transferability (i.e., addressing the core concern of multi-task learning). For example, it enables locomotion skills to be reused for multiple goal-achieving tasks of the same robot agent, yet in distinct scenarios. Different from previous Multi-task Hierarchical IL (MHIL) algorithms (Fox et al., 2019; Yu et al., 2018a; Gao et al., 2022; Bian et al., 2022), MH-AIRL is context-based and thus can be applied to demonstrations without any (skill or task) annotations, which are more accessible in practice. To this end, we extend both the multi-task learning and imitation learning modules (i.e., the core components of MIL), with the option framework (i.e., the hierarchical learning module). For multi-task learning, we condition the learned policy on a Hierarchical Latent Context Structure, where the task code and skill segmentation serve as the global and local context variables respectively. To compel the casual relationship of learned policy and latent variables, we start from the definition of mutual information and directed information and derive an easier-to-handle lower bound for each of them, serving as the optimization objectives. For imitation learning, we propose H-AIRL, which redefines a SOTA IL algorithm – AIRL (Fu et al., 2017) in an extended state-action space to enable our algorithm to recover a hierarchical policy (rather than a monolithic one) from expert trajectories. Finally, an actor-critic framework – HPPO is proposed to synthesize the optimization of the three modules above.

The contributions are as follows: (1) Our work presents the first MHIL algorithm based on demonstrations without any (skill or task) annotations, i.e., state-action pairs only. This greatly generalizes the applicability of our algorithm and reduces the cost of building expert datasets. (2) The newly-proposed H-AIRL and HPPO can be independently used for Hierarchical IL and RL, respectively. They are shown to achieve improved performance than SOTA HIL and HRL baselines. (3) We provide theoretical proof and ablation study for each algorithm module, and show the superiority of our algorithm through comparisons with SOTA baselines on a series of challenging multi-task settings from Mujoco (Todorov et al., 2012) and D4RL (Fu et al., 2020).

2. Related Work

Machine Learning has found successful applications across a wide array of sectors such as transportation (Al-Abbasi et al., 2019; Chen et al., 2021; Luo et al., 2022; Ma et al., 2020), manufacturing (Peddireddy et al., 2021; Fu et al., 2021), networking (Balachandran et al., 2014; Geng et al., 2023), robotics (Gao et al., 2022; Gonzalez et al., 2023), etc. In the field of robotics, one of the key objectives is developing a 'generalist' robot, capable of executing a multitude of tasks with human-like precision. To achieve this, multi-task robotic learning proves to be a highly effective methodology. In this section, we succinctly delineate Multi-task IL and Multi-task HIL, illustrating the contributions and significance of our research in this evolving field.

Multi-task/Meta IL algorithms have been proposed to learn a parameterized policy, which is capable of performing a range of tasks following a particular distribution, from a mixture of expert demonstrations. Based on the meta/multi-task learning techniques used, current MIL algorithms can be categorized as gradient-based or context-based. Gradient-based MIL, such as (Finn et al., 2017b; Yu et al., 2018b), integrates a gradient-based meta learning algorithm — MAML (Finn et al., 2017a) with supervised IL to train a policy that can be fast adapted to a new task with one-step gradient update. Context-based MIL, such as (Ghasemipour et al., 2019; Yu et al., 2019), learns a latent variable to represent the task contexts and trains a policy conditioned on the task context variable. Thus, with the corresponding task variable, the policy can be directly adopted to a new task setting. However, these algorithms do not make use of the option framework to learn a hierarchical policy like ours. In Section 5.1, we compare our algorithm with MIL baselines from both categories and show that it achieves better performance on a wide range of challenging long-horizon tasks.

Multi-task HIL aims at recovering a multi-task hierarchical policy based on expert demonstrations from a distribution of tasks, which synthesizes the advantages of Multi-task IL and HIL. We present here the previous study in this area. The algorithms proposed in (Fox et al., 2019) and (Duminy et al., 2021) are limited to a certain type of robot. They provide predefined subtask decomposition, like picking and placing dishes, to simplify hierarchical learning, and have access to segmented expert demonstrations. However, our algorithm is proposed to automatically discover a hierarchical policy from unsegmented demonstrations and the discovered policy should capture the subtask structure of the demonstrations without supervision. In (Yu et al., 2018a), they propose to let the robot learn a series of primitive skills from corresponding demonstrations first, and then learn to compose learned primitives into multi-stage skills to complete a task. Thus, they predefine the types of skills and provide demonstrations corresponding to each skill. Also, in their setting, each new task has to be a sequence of predefined skills. A very recent work (Gao et al., 2022) integrates MAML and the option framework for MHIL. Like (Bian et al., 2022) and (Devin et al., 2019), this algorithm can be applied to demonstrations without the skill annotations, but these demonstrations have to be categorized by the task, in accordance with the requirements of MAML. Consequently, our research introduces the first MHIL algorithm that relies on demonstrations devoid of task or skill annotations. This makes it significantly more practical for real-world applications.

3. Background

In this section, we introduce Adversarial Inverse Reinforcement Learning (AIRL), Context-based Meta Learning, and the One-step Option Framework, corresponding to the three components of our algorithm: IL, multi-task learning, and hierarchical policy learning, respectively. They are based on the Markov Decision Process (MDP), denoted by $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mu, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition function ($\mathcal{P}_{S_t,A_t}^{S_{t+1}} \triangleq \mathcal{P}(S_{t+1}|S_t, A_t)$), $\mu : \mathcal{S} \to [0, 1]$ is the distribution of the initial state, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and $\gamma \in (0, 1]$ is the discount factor.

3.1. Adversarial Inverse Reinforcement Learning

While there are several other ways to perform IL, such as supervised imitation (e.g., Behavioral Cloning (BC) (Pomerleau, 1991)) and occupancy matching (e.g., GAIL (Ho & Ermon, 2016)), we adopt Inverse Reinforcement Learning (IRL) because it uses not only the expert data but also self-exploration of the agent with the recovered reward function for further improvement (Ng & Russell, 2000; Wang et al., 2021). Comparisons with BC- and GAIL-based algorithms will be provided in Section 5. IRL aims to infer an expert's reward function from demonstrations, based on which the expert's policy can be recovered. Maximum Entropy IRL (Ziebart et al., 2008) solves IRL as a maximum likelihood estimation (MLE) problem shown as Equation 1. $\tau_E \triangleq (S_0, A_0, \cdots, S_T)$ denotes the expert trajectory. Z_ϑ is the partition function which can be calculated with $Z_\vartheta = \sum_{\tau_E} \widehat{P}_\vartheta(\tau_E)$.

$$\begin{aligned} &\max_{\vartheta} \mathbb{E}_{\tau_E}\left[\log P_\vartheta(\tau_E)\right] = \max_{\vartheta} \mathbb{E}_{\tau_E}\left[\log \frac{\widehat{P}_\vartheta(\tau_E)}{Z_\vartheta}\right], \\ &\widehat{P}_\vartheta(\tau_E) = \mu(S_0) \prod_{t=0}^{T-1} \mathcal{P}_{S_t,A_t}^{S_{t+1}} \exp(\mathcal{R}_\vartheta(S_t, A_t)) \end{aligned} \tag{1}$$

Since Z_ϑ is intractable for problems with large state-action space, the authors of (Fu et al., 2017) propose AIRL to solve this MLE problem in a sample-based manner, through alternatively training a discriminator f_ϑ and policy network π in an adversarial setting. The discriminator is trained by minimizing the cross-entropy loss between the expert demonstrations τ_E and generated samples τ by π:

$$\min_{\vartheta} \sum_{t=0}^{T-1} -\mathbb{E}_{\tau_E}\left[\log D_\vartheta^t\right] - \mathbb{E}_{\tau}\left[\log(1 - D_\vartheta^t)\right] \tag{2}$$

Here, $D_\vartheta^t = D_\vartheta(S_t, A_t) = \frac{\exp(f_\vartheta(S_t,A_t))}{\exp(f_\vartheta(S_t,A_t))+\pi(A_t|S_t)}$. Meanwhile, the policy π is trained with RL using the reward function defined as $\log D_\vartheta^t - \log(1 - D_\vartheta^t)$. It is shown that, at optimality, f_ϑ can serve as the recovered reward function $\mathcal{R}_\vartheta$ and π is the recovered expert policy.

3.2. Context-based Meta Learning

We consider the Meta IRL setting: given a distribution of tasks $P(\mathcal{T})$, each task sampled from $P(\mathcal{T})$ has a corresponding MDP, and all of them share the same $\mathcal{S}$ and $\mathcal{A}$ but may differ in μ, $\mathcal{P}$, and $\mathcal{R}$. The goal is to train a flexible policy π on a set of training tasks sampled from $P(\mathcal{T})$, which can be quickly adapted to unseen test tasks sampled from the same distribution. As a representative, context-based Meta IRL algorithms (Ghasemipour et al., 2019; Yu et al., 2019) introduce the latent task variable C, which provides an abstraction of the corresponding task $\mathcal{T}$, so each task can be represented with its distinctive components conditioning on C, i.e., $(\mu(S_0|C), \mathcal{P}(S'|S, A, C), \mathcal{R}(S, A|C))$. These algorithms learn a context-conditioned policy $\pi(A|S, C)$ from the multi-task expert data, through IRL and by maximizing the mutual information (Cover, 1999) between the task variable C and the trajectories from $\pi(A|S, C)$. Thus, given C for a new task, the corresponding $\pi(A|S, C)$ can be directly adopted. Context-based methods can adopt off-policy data, making them more align with the goal of our work – learning from demonstrations. Thus, we choose context-based Meta IRL as our base algorithm.

Given expert trajectories sampled from a distribution of tasks (i.e., $C \sim prior(\cdot)$) and assuming that the demonstrative trajectories of each task are from a corresponding expert policy $\pi_E(\tau_E|C)$, context-based Meta IRL recovers both the task-conditioned reward function $\mathcal{R}_\vartheta(S, A|C)$ and policy $\pi(S, A|C)$ by solving an MLE problem:

$$\begin{aligned} &\max_{\vartheta} \mathbb{E}_{C\sim prior(\cdot), \tau_E\sim\pi_E(\cdot|C)}\left[\log P_\vartheta(\tau_E|C)\right], \\ &P_\vartheta(\tau_E|C) \propto \mu(S_0|C) \prod_{t=0}^{T-1} \mathcal{P}_{S_t,A_t,C}^{S_{t+1}} \, e^{\mathcal{R}_\vartheta(S_t,A_t|C)} \end{aligned} \tag{3}$$

where $\mathcal{P}_{S_t,A_t,C}^{S_{t+1}} \triangleq \mathcal{P}(S_{t+1}|S_t, A_t, C)$. Like Equation 1, this can be efficiently solved through AIRL. We provide the AIRL framework to solve Equation 3 in Appendix A.1.

3.3. One-step Option Framework

As proposed in (Sutton et al., 1999), an option $Z \in \mathcal{Z}$ can be described with three components: an initiation set $I_Z \subseteq \mathcal{S}$, an intra-option policy $\pi_Z(A|S) : \mathcal{S} \times \mathcal{A} \to [0, 1]$, and a termination function $\beta_Z(S) : \mathcal{S} \to [0, 1]$. An option Z is available in state S if and only if $S \in I_Z$. Once the option is taken, actions are selected according to π_Z until it terminates stochastically according to β_Z, i.e., the termination probability at the current state. A new option will be activated by a high-level policy $\pi_{\mathcal{Z}}(Z|S) : \mathcal{S} \times \mathcal{Z} \to [0, 1]$ once the previous option terminates. In this way, $\pi_{\mathcal{Z}}(Z|S)$ and $\pi_Z(A|S)$ constitute a hierarchical policy for a certain task. Hierarchical policies tend to have superior performance on complex long-horizontal tasks which can be broken down into a series of subtasks (Chen et al., 2022a;b;c;d).

The one-step option framework (Li et al., 2021) is proposed to learn the hierarchical policy without the extra need to justify the exact beginning and breaking condition of each option, i.e., I_Z and β_Z. First, it assumes that each option is available at each state, i.e., $I_Z = \mathcal{S}, \forall Z \in \mathcal{Z}$. Second, it drops β_Z through redefining the high-level and low-level (i.e., intra-option) policies as $\pi_\theta(Z|S, Z')$ (Z': the option in the last timestep) and $\pi_\phi(A|S, Z)$ respectively and implementing them as end-to-end neural networks with the Multi-Head Attention (MHA) mechanism (Vaswani et al., 2017), which enables it to temporally extend options in the absence of the termination function. Intuitively, if Z' still fits S, $\pi_\theta(Z|S, Z')$ will assign a larger attention weight to Z' and thus has a tendency to continue with it; otherwise, a new option with better compatibility will be sampled. Then, the option is sampled at each timestep rather than after the last one terminates. With this simplified framework, we only need to train the hierarchical policy, i.e., π_θ and π_ϕ, of which the structure design with MHA is in Appendix A.2.

4. Proposed Approach

In this section, we propose Multi-task Hierarchical AIRL (MH-AIRL) to learn a multi-task hierarchical policy from a mixture of expert demonstrations. **First**, the learned policy is multi-task by conditioning on the task context variable C. Given $C \sim prior(\cdot)$, the policy can be directly adopted to complete the corresponding task. In practice, we can usually model a class of tasks by specifying the key parameters of the system and their distributions (i.e., $prior(C)$), including the property of the agent (e.g., mass and size), circumstance (e.g., friction and layout), and task setting (e.g., location of the goals). In this case, directly recovering a policy, which is applicable to a class of tasks, is quite meaningful. **Second**, for complex long-horizontal tasks which usually contain subtasks, learning a monolithic policy to represent a structured activity can be challenging and inevitably requires more demonstrations. In contrast, a hierarchical policy can make full use of the subtask structure and has the potential for better performance. Moreover, the learned low-level policies can be transferred as basic skills to out-of-distribution tasks for better transferability, while the monolithic policy learned with previous Meta IL algorithms cannot.

In Section 4.1 and 4.2, we extend context-based Meta Learning and AIRL with the option framework, respectively. In Section 4.3, we synthesize the three algorithm modules and propose an actor-critic framework for optimization.

4.1. Hierarchical Latent Context Structure

As mentioned in Section 3.2, the current task for the agent is encoded with the task variable C, which serves as the **global** context since it is consistent through the episode. As mentioned in Section 3.3, at each step, the hierarchical policy agent will first decide on its option choice Z using π_θ and then select the primitive action based on the low-level policy π_ϕ corresponding to Z. In this case, the policy learned should be additionally conditioned on Z besides the task code C, and the option choice is specific to each timestep $t \in \{0, \cdots, T\}$, so we view the option choices $Z_{0:T}$ as the **local** latent contexts. C and $Z_{0:T}$ constitute a hierarchical latent context structure shown as Figure 1. Moreover, real-world tasks are often compositional, so the agent requires to reason about the subtask at hand while dealing with the global task. $Z_{0:T}$ and C provide a hierarchical embedding, which enhances the expressiveness of the policy trained with MH-AIRL, compared with context-based Meta IL which only employs the task context. **In this section**, we define the mutual and directed information objectives to enhance the causal relationship between the hierarchical policy and the global & local context variables which the policy should condition on, as an extension of context-based Meta-IL with the one-step option model.

Context-based Meta IL algorithms establish a connection between the policy and task variable C, so that the policy can be adapted among different task modes according to the task context. This can be realized through maximizing the mutual information between the trajectory generated by the policy and the corresponding C, i.e., $I(X_{0:T}; C)$, where $X_{0:T} = (X_0, \cdots, X_T) = ((A_{-1}, S_0), \cdots, (A_{T-1}, S_T)) = \tau$. A_{-1} is a dummy variable. On the other hand, the local latent variables $Z_{0:T}$ have a directed causal relationship with the trajectory $X_{0:T}$ shown as the probabilistic graphical model in Figure 1. As discussed in (Massey et al., 1990; Sharma et al., 2019), this kind of connection can be established by maximizing the directed information (a.k.a., causal information) flow from the trajectory to the latent factors of variation, i.e., $I(X_{0:T} \to Z_{0:T})$. In our multi-task framework, we maximize the conditional directed information $I(X_{0:T} \to Z_{0:T}|C)$, since for each task c, the corresponding $I(X_{0:T} \to Z_{0:T}|C = c)$ should be maximized.

Directly optimizing the mutual or directed information objective is computationally infeasible, so we instead maximize their variational lower bounds as follows: (Please refer to Appendix B.1 and B.2 for the definition of mutual and directed information and derivations of their lower bounds. For simplicity, we use X^T to represent $X_{0:T}$, and so on.)

$$\begin{aligned} L^{MI} &\triangleq H(C) + \mathop{\mathbb{E}}_{X^T, Z^T, C} \log P_\psi(C|X_{0:T}) \\ L^{DI} &\triangleq \sum_{t=1}^{T} [\mathop{\mathbb{E}}_{X^t, Z^t, C} \log P_\omega(Z_t|X_{0:t}, Z_{0:t-1}, C) \\ &\qquad + H(Z_t|X_{0:t-1}, Z_{0:t-1}, C)] \end{aligned} \tag{4}$$

where $H(\cdot)$ denotes the entropy, P_ψ and P_ω are the variational estimation of the posteriors $P(C|X_{0:T})$ and

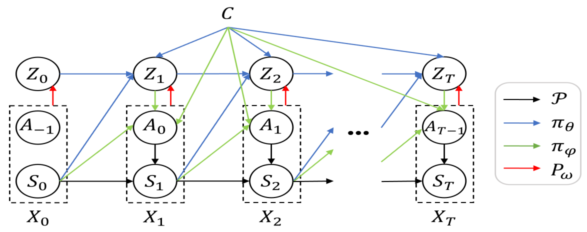


Figure 1. Illustration of the hierarchical latent context structure and its implementation with the one-step option model.

$P(Z_t|X_{0:t}, Z_{0:t-1}, C)$ which cannot be calculated directly. P_ψ and P_ω are implemented as neural networks, $H(C)$ is constant, and $H(Z_t|X_{0:t-1}, Z_{0:t-1}, C)$ is the entropy of the output of the high-level policy network (Appendix B.1), so L^{MI} and L^{DI} can be computed in real-time. Moreover, the expectation on X^t, Z^t, C in L^{MI} and L^{DI} can be estimated in a Monte-Carlo manner (Sutton & Barto, 2018): $C \sim prior(\cdot)$, $(X_{0:t}, Z_{0:t}) \sim P_{\theta,\phi}(\cdot|C)$, where $P_{\theta,\phi}(X_{0:t}, Z_{0:t}|C)$ is calculated by: (See Appendix B.1.)

$$\mu(S_0|C)\prod_{i=1}^{t}[\pi_\theta(Z_i|S_{i-1}, Z_{i-1}, C)\cdot \pi_\phi(A_{i-1}|S_{i-1}, Z_i, C)\mathcal{P}_{S_{i-1},A_{i-1},C}^{S_i}] \tag{5}$$

Combining Equation 4 and 5, we can get the objectives with respect to π_θ and π_ϕ, i.e., the hierarchical policy defined in the one-step option model. By maximizing L^{MI} and L^{DI}, the connection between the policy and the hierarchical context structure can be established and enhanced. In L^{MI} and L^{DI}, we also introduce two variational posteriors P_ψ and P_ω and update them together with π_θ and π_ϕ. An analogy of our learning framework with Variational Autoencoder (VAE) (Kingma & Welling, 2014) is provided in Appendix B.3, which provides another perspective to understand the proposed objectives.

4.2. Hierarchical AIRL

In this section, we consider how to recover the task-conditioned hierarchical policy from a mixture of expert demonstrations $\{(X^E_{0:T}, Z^E_{0:T}, C_E)\}$. Current algorithms, like AIRL (Fu et al., 2017) or Meta AIRL (Ghasemipour et al., 2019; Yu et al., 2019), can not be directly adopted since they don't take the local latent codes $Z^E_{0:T}$ into consideration. Thus, we propose a novel hierarchical extension of AIRL, denoted as H-AIRL, as a solution, which is also part of our contributions. Further, it's usually difficult to annotate the local and global latent codes, i.e., $Z^E_{0:T}$ and C_E, of an expert trajectory $X^E_{0:T}$, so we propose an Expectation-Maximization (EM) adaption of H-AIRL as well to learn the multi-task hierarchical policy based on only the unstructured expert trajectories $\{X^E_{0:T}\}$.

First, we define the task-conditioned hierarchical policy. When observing a state S_t at timestep $t \in \{0, \cdots, T-1\}$ during a certain task C, the agent needs first to decide on its option choice based on S_t and its previous option choice Z_t using the high-level policy $\pi_\theta(Z_{t+1}|S_t, Z_t, C)$, and then decide on the action with the corresponding low-level policy $\pi_\phi(A_t|S_t, Z_{t+1}, C)$. Thus, the task-conditioned hierarchical policy can be acquired with the chain rule as:

$$\begin{aligned}&\pi_\theta(Z_{t+1}|S_t, Z_t, C)\cdot\pi_\phi(A_t|S_t, Z_{t+1}, C)\\&= \pi_{\theta,\phi}(Z_{t+1}, A_t|S_t, Z_t, C) = \pi_{\theta,\phi}(\widetilde{A}_t|\widetilde{S}_t, C)\end{aligned} \tag{6}$$

where the first equality holds because of the one-step Markov assumption (i.e., $\pi_\phi(A_t|S_t, Z_t, Z_{t+1}, C) = \pi_\phi(A_t|S_t, Z_{t+1}, C)$), $\widetilde{S}_t \triangleq (S_t, Z_t)$ and $\widetilde{A}_t \triangleq (Z_{t+1}, A_t)$ denote the extended state and action space respectively.

Next, by substituting (S_t, A_t) with $(\widetilde{S}_t, \widetilde{A}_t)$ and τ_E with the hierarchical trajectory $(X_{0:T}, Z_{0:T})$ in Equation 3, we can get an MLE problem shown as Equation 7, from which we can recover the task-conditioned hierarchical reward function and policy. The derivation is in Appendix C.1.

$$\begin{aligned}&\max_{\vartheta}\mathbb{E}_{C,(X^T,Z^T)\sim\pi_E(\cdot|C)}\left[\log P_\vartheta(X^T, Z^T|C)\right],\\&P_\vartheta(X_{0:T}, Z_{0:T}|C) \propto \widehat{P}_\vartheta(X_{0:T}, Z_{0:T}|C)\\&= \mu(S_0|C)\prod_{t=0}^{T-1}\mathcal{P}_{S_t,A_t,C}^{S_{t+1}}\, e^{\mathcal{R}_\vartheta(S_t,Z_t,Z_{t+1},A_t|C)}\end{aligned} \tag{7}$$

Equation 7 can be efficiently solved with the adversarial learning framework shown as Equation 8 ($C, C_E \sim prior(\cdot), (X^E_{0:T}, Z^E_{0:T}) \sim \pi_E(\cdot|C_E)$, and $(X_{0:T}, Z_{0:T}) \sim \pi_{\theta,\phi}(\cdot|C)$). At optimality, we can recover the hierarchical policy of the expert as $\pi_{\theta,\phi}$ with these objectives, of which the justification is provided in Appendix C.2.

$$\begin{aligned}\min_{\vartheta} &-\mathbb{E}_{C_E,(X^E_{0:T},Z^E_{0:T})}\sum_{t=0}^{T-1}\log D_\vartheta(\widetilde{S}^E_t, \widetilde{A}^E_t|C_E)\\&-\mathbb{E}_{C,(X_{0:T},Z_{0:T})}\sum_{t=0}^{T-1}\log(1-D_\vartheta(\widetilde{S}_t, \widetilde{A}_t|C)),\\\max_{\theta,\phi} &\, L^{IL} = \mathbb{E}_{C,(X_{0:T},Z_{0:T})}\sum_{t=0}^{T-1}R^t_{IL}\end{aligned} \tag{8}$$

where the reward function $R^t_{IL} = \log D^t_\vartheta - \log(1-D^t_\vartheta)$ and $D^t_\vartheta = D_\vartheta(\widetilde{S}_t, \widetilde{A}_t|C) = \frac{\exp(f_\vartheta(\widetilde{S}_t,\widetilde{A}_t|C))}{\exp(f_\vartheta(\widetilde{S}_t,\widetilde{A}_t|C))+\pi_{\theta,\phi}(\widetilde{A}_t|\widetilde{S}_t,C)}$.

In practice, the unstructured expert data $\{X^E_{0:T}\}$, i.e., trajectories only, is more accessible. In this case, we can view the latent contexts as hidden variables in a hidden Markov model (HMM) (Eddy, 1996) shown as Figure 1 and adopt an EM-style adaption to our algorithm, where we use the variational posteriors introduced in Section 4.1 to sample the corresponding $C_E, Z_{0:T}^E$ for each $X_{0:T}^E$. In the E step, we sample the global and local latent codes with $C_E \sim P_{\overline{\psi}}(\cdot|X_{0:T}^E), Z_{0:T}^E \sim P_{\overline{\omega}}(\cdot|X_{0:T}^E, C_E)$. $P_{\overline{\psi}}$ and $P_{\overline{\omega}}$ represent the posterior networks for C and $Z_{0:T}$ respectively, with the parameters $\overline{\psi}$ and $\overline{\omega}$, i.e., the old parameters before being updated in the M step. Then, in the M step, we optimize the hierarchical policy and posteriors with Equation 4 and 8. Note that the expert data used in the first term of Equation 8 should be replaced with $(X_{0:T}^E, Z_{0:T}^E, C_E)$ collected in the E step. By this adaption, we can get the solution of the original MLE problem (Equation 7), i.e., the recovered expert policy $\pi_{\theta,\phi}$, with only unstructured expert data, which is proved in Appendix C.3.

4.3. Overall Framework

In Section 4.1, we propose $L^{MI}(\theta, \phi, \psi)$ and $L^{DI}(\theta, \phi, \omega)$ to establish the causal connection between the policy and hierarchical latent contexts. Then, in Section 4.2, we propose H-AIRL to recover the hierarchical policy from multi-task expert demonstrations, where the policy is trained with the objective $L^{IL}(\theta, \phi)$. In this section, we introduce our method to update the hierarchical policy and posteriors with these objectives, and describe the overall algorithm framework. Detailed derivations of $\nabla_{\theta,\phi,\psi}L^{MI}$, $\nabla_{\theta,\phi,\omega}L^{DI}$ and $\nabla_{\theta,\phi}L^{IL}$ are in Appendix D.1, D.2, and D.3, respectively.

First, the variational posteriors P_ψ and P_ω can be updated with the gradients shown in Equation 9 through Stochastic Gradient Descent (SGD) (Bottou, 2010).

$$\begin{aligned}\nabla_\psi L^{MI} &= \mathop{\mathbb{E}}_{C,X^T,Z^T} \nabla_\psi \log P_\psi(C|X_{0:T}) \\ \nabla_\omega L^{DI} &= \sum_{t=1}^{T} \mathop{\mathbb{E}}_{C,X^t,Z^t} \nabla_\omega \log P_\omega(Z_t|X^t, Z^{t-1}, C)\end{aligned} \tag{9}$$

Next, the gradients with respect to θ and ϕ, i.e., the hierarchical policy, are computed based on the overall objective:

$$L = \alpha_1 L^{MI} + \alpha_2 L^{DI} + \alpha_3 L^{IL} \tag{10}$$

where $\alpha_{1:3}$ are the weights (only the ratios $\frac{\alpha_1}{\alpha_3}, \frac{\alpha_2}{\alpha_3}$ matter) and fine-tuned as hyperparameters. Based on L, we can get the unbiased gradient estimators with respect to θ and ϕ: (Derivations are in Appendix D.4.)

$$\begin{aligned}\nabla_\theta L = &\mathop{\mathbb{E}}_{C,X^T,Z^T}[\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(Z_t|S_{t-1}, Z_{t-1}, C) \cdot \\ &(Ret_t - b^{high}(S_{t-1}, Z_{t-1}|C))] \\ \nabla_\phi L = &\mathop{\mathbb{E}}_{C,X^T,Z^T}[\sum_{t=1}^{T} \nabla_\phi \log \pi_\phi(A_{t-1}|S_{t-1}, Z_t, C) \cdot \\ &(Ret_t - b^{low}(S_{t-1}, Z_t|C))]\end{aligned} \tag{11}$$

$$\begin{aligned}Ret_t = &\alpha_1 \log P_\psi(C|X_{0:T}) \\ &+ \sum_{i=t}^{T}[\alpha_2 \log \frac{P_\omega(Z_i|X^i, Z^{i-1}, C)}{\pi_\theta(Z_i|S_{i-1}, Z_{i-1}, C)} + \alpha_3 R_{IL}^{i-1}]\end{aligned} \tag{12}$$

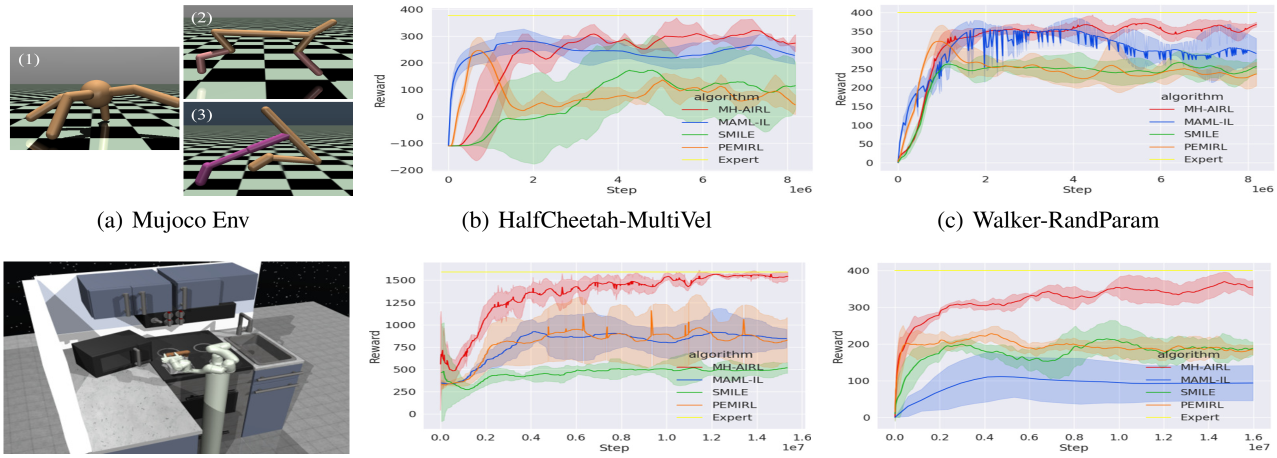
Ret_t represents the return at timestep t, while b^{high} and b^{low} are the baseline terms for training π_θ and π_ϕ, respectively. Further, we claim that the advantage functions for training π_θ and π_ϕ are given by $Ret_t - b^{high}(S_{t-1}, Z_{t-1}|C)$ and $Ret_t - b^{low}(S_{t-1}, Z_t|C)$, respectively, based on which we can optimize the hierarchical policy via off-the-shelf RL algorithms. In our implementation, we adopt PPO (Schulman et al., 2017) to train π_θ and π_ϕ with their corresponding advantage functions, respectively. This forms a novel Hierarchical RL (HRL) algorithm – HPPO, which has shown superiority over RL and HRL baselines in our experiment.

In Appendix D.5, we provide the overall algorithm as Algorithm 1 and illustrate the interactions among the networks in MH-AIRL in Figure 5.

5. Evaluation and Main Results

MH-AIRL is proposed to learn a multi-task hierarchical policy from a mixture of (unstructured) expert demonstrations. The learned policy can be adopted to any task sampled from a distribution of tasks. In this section: (1) We provide an ablation study with respect to the three main components of our algorithm: context-based multi-task/meta learning, option/hierarchical learning, and imitation learning. (2) We show that the hierarchical policy learning can significantly improve the agent's performance on challenging long-horizontal tasks. (3) Through qualitative and quantitative results, we show that our algorithm can capture the subtask structure within the expert demonstrations and that the learned basic skills for the subtasks (i.e., options) can be transferred to tasks not within the task distribution to aid learning, for better transferability.

The evaluation is based on three Mujoco (Todorov et al., 2012) locomotion tasks and the Kitchen task from the D4RL benchmark (Fu et al., 2020). All of them are with continuous state & action spaces, and contain compositional subtask structures to make them long-horizontal and a lot more challenging. To be specific: (1) In **HalfCheetah-MultiVel**, the goal velocity v is controlled by a 1-dim Gaussian context variable. The HalfCheetah agent is required to speed up to $v/2$ first, then slow down to 0, and finally achieve v. (2) In **Walker-RandParam**, the Walker agent must achieve the goal velocity 4 in three stages, i.e., [2, 0, 4]. Meanwhile, the mass of the agent changes among different tasks, which is controlled by an 8-dim Gaussian context variable. (3) In **Ant-MultiGoal**, a 3D Ant agent needs to reach a certain goal, which is different in each task and controlled



(a) Mujoco Env (b) HalfCheetah-MultiVel (c) Walker-RandParam

(d) Kitchen Env (e) Ant-MultiGoal (f) Kitchen-MultiSeq

Figure 2. (a) Multi-stage Mujoco locomotion tasks, where (1)-(3) show Ant, HalfCheetah, and Walker agent, respectively. (d) The Kitchen task. (b)(c)(e)(f) Comparison results of MH-AIRL with SOTA Meta Imitation Learning baselines on the four challenging tasks.

by a 2-dim Gaussian context variable (polar coordinates). Moreover, the agent must go through certain subgoals. For example, if the goal is (x, y) and $|x| > |y|$, the agent must go along $[(0, 0), (x, 0), (x, y)]$. (4) In **Kitchen-MultiSeq**, there are seven different subtasks, like manipulating the microwave, kettle, cabinet, switch, burner, etc. Each task requires the sequential completion of four specific subtasks. Twenty-four permutations are chosen and so 24 tasks, each of which is sampled with the same probability and controlled by a discrete context variable (input as one-hot vectors). Note that the states of the robot agents only contain their original states (defined by Mujoco or D4RL) and the task context variable, and do not include the actual task information, like the goal (velocity) and subgoal list. The task information is randomly generated by a parametric model of which the parameter is used as the context variable (i.e., the Gaussian vectors as mentioned above). The mapping between context variables and true task information is unknown to the learning agent. This makes the learning problem more challenging and our algorithm more general, since a vector of standard normal variables can be used to encode multiple types of task information.

These scenarios are designed to evaluate our algorithm on a wide range of multi-task setups. First, the agent needs to adapt across different reward functions in (1) and (3) since the rewarding state changes, and adjust across different transition functions in (2) since the mass change will influence the robotic dynamics. Next, different from (1)-(3), discrete context variables are adopted in (4), and (4) provides more realistic and challenging robotic tasks for evaluation. The expert data for Mujoco tasks are from expert agents trained with an HRL algorithm (Zhang & Whiteson, 2019) and specifically-designed rewards. While for the Kitchen task, we use the human demonstrations provided by (Gupta et al., 2019). Note that the demonstrations (state-action pairs only) do not include the rewards, task or option variables. Codes for reproducing all the results are on https://github.com/LucasCJYSDL/Multi-task-Hierarchical-AIRL.

5.1. Effect of Hierarchical Learning

In this part, we evaluate whether the use of options can significantly improve the learning for challenging compound multi-task settings. We compare MH-AIRL with SOTA Meta Imitation Learning (MIL) baselines which also aim to train a policy that can be fast adapted to a class of related tasks but does not adopt options in learning. Context-based MIL, such as **PEMIRL** (Yu et al., 2019) and **SMILE** (Ghasemipour et al., 2019), learns a context-conditioned policy that can be adopted to any task from a class by applying the task variable. While the policy learned with Gradient-based MIL, such as **MAML-IL** (Finn et al., 2017b) which integrates MAML (Finn et al., 2017a) (a commonly-adopted Meta Learning algorithm) and Behavioral Cloning (BC), has to be updated with gradients calculated from trajectories of the new task, before being applied. We select PEMIRL, SMILE, and MAML-IL from the two major categories of MIL as our baselines. All the algorithms are trained with the same expert data, and evaluated on the same set of test tasks (not contained in the demonstrations). Note that, unlike the others, MAML-IL requires expert data of each test task besides the task variable when testing and requires the expert demonstrations to be categorized by the task when training, which may limit its use in practical scenarios. Our algorithm is trained based on unstructured demonstrations and is only provided with the task context variable for testing.

Table 1. Numeric results of the ablation study

	HalfCheetah-MultiVel	Walker-RandParam	Ant-MultiGoal	Kitchen-MultiSeq
Expert	376.55 ± 11.12	399.95 ± 1.43	1593.17 ± 40.91	400.00 ± 0.00
MH-AIRL (ours)	$\mathbf{292.79 \pm 15.99}$	$\mathbf{357.59 \pm 12.10}$	$\mathbf{1530.82 \pm 15.18}$	$\mathbf{352.59 \pm 15.12}$
MH-GAIL (ours)	211.32 ± 52.74	268.92 ± 49.29	1064.78 ± 180.28	212.13 ± 25.25
H-AIRL (ours)	126.85 ± 21.92	225.48 ± 12.87	533.80 ± 40.69	83.97 ± 10.95
Option-GAIL	-44.89 ± 51.95	132.01 ± 54.75	383.05 ± 13.52	204.73 ± 56.41
DI-GAIL	56.77 ± 49.76	225.22 ± 14.01	328.06 ± 19.89	131.79 ± 53.29

In Figure 2, we record the change of the episodic reward (i.e., the sum of rewards for each step in an episode) on the test tasks as the number of training samples increases. The training is repeated 5 times with different random seeds for each algorithm, of which the mean and standard deviation are shown as the solid line and shadow area, respectively. Our algorithm outperforms the baselines in all tasks, and the improvement is more significant as the task difficulty goes up (i.e., in Ant & Kitchen), which shows the effectiveness of hierarchical policy learning especially in complex tasks. MAML-IL makes use of more expert information in both training and testing, but its performance gets worse on more challenging tasks. This may be because it is based on BC, which is a supervised learning algorithm prone to compounding errors (Ross et al., 2011).

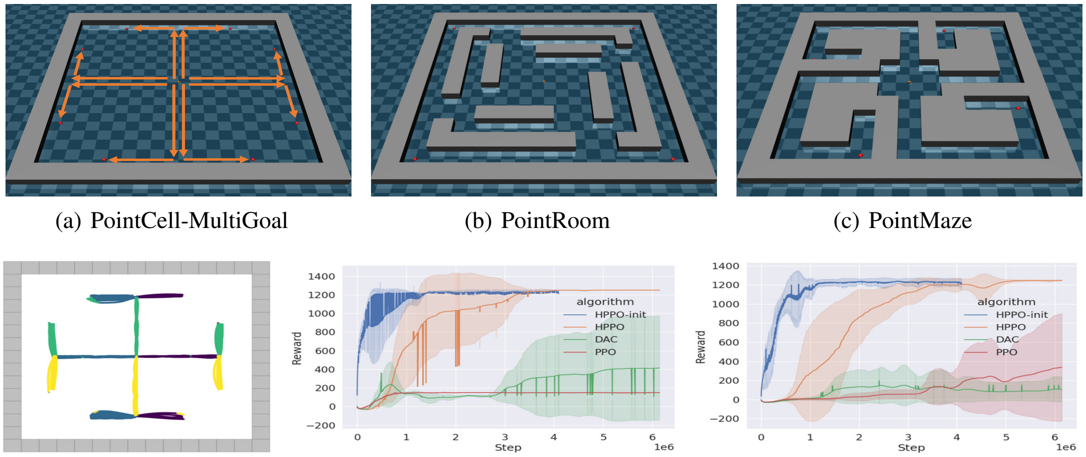
5.2. Ablation Study

We proceed to show the effectiveness of the IL and context-based multi-task learning components through an ablation study. We propose two ablated versions of our algorithm: (1) **MH-GAIL** – a variant by replacing the AIRL component of MH-AIRL with GAIL (Ho & Ermon, 2016) (another commonly-used IL algorithm), of which the details are in Appendix E.2. (2) **H-AIRL** – a version that does not consider the task context C, which means P_ψ (i.e., the posterior for C) is not adopted, L^{MI} is eliminated from Equation 10, and other networks do not use C as input. H-AIRL can be viewed as a newly-proposed HIL algorithm since it integrates the option framework and IL. To be more convincing, we also use two SOTA HIL algorithms – **Option-GAIL** (Jing et al., 2021) and **DI-GAIL** (Sharma et al., 2019), as the baselines. The training with the HIL algorithms is based on the same multi-task expert data as ours.

In Appendix E.1, we provide the plots of the change of episodic rewards on the test tasks. The training with each algorithm is repeated for 5 times with different random seeds. For each algorithm, we compute the average episodic reward after the learning converges in each of the 5 runs, and record the mean and standard deviation in Table 1 as the convergence performance. First, we can see that our algorithm performs the best on all tasks over the ablations, showing the effectiveness of all the main modules of our algorithm. Second, MH-GAIL performs better than HIL baselines, showing the necessity of including the context-based multi-task learning component. Without this component, HIL algorithms can only learn an average policy for a class of tasks from the mixture of multi-task demonstrations. Last, H-AIRL, the newly-proposed HIL algorithm, performs better than the SOTA HIL baselines on Mujoco tasks. A comprehensive empirical study on H-AIRL is provided in (Chen et al., 2022e).

5.3. Analysis on the Learned Hierarchical Policy

In this section, we do the case study to analyze if the learned hierarchical policy can capture the sub-task structure in the demonstrations, and if the learned options can be transferred to tasks out of the task distribution. Capturing the subtask structures in real-life tasks can be essential for the (multi-task) policy learning, because: (1) It is more human-like to split a complex task into more manageable subtasks to learn separately and then synthesize these skills to complete the whole task. (2) In some circumstances, the basic skills learned from one task setting can be reused in other task settings so the agent only needs to update its high-level policy over the same skill set, significantly lowering the learning difficulty. We test our algorithm on Mujoco-MultiGoal (Figure 3(a)) where the agent is required to achieve a goal corresponding to the task variable (2-dim Gaussian). The expert demonstrations include 100 goal locations in the Cell and the expert agent only moves horizontally or vertically. We test the learned hierarchical policy on 8 sparsely distributed goal locations, of which the trajectories are shown as Figure 3(d). We can see: (1) Four options (labeled with different colors) are discovered based on the demonstrations, each of which corresponds to a particular forward direction (green: up, yellow: down, etc.). These options are shared among the tasks. (2) The agent knows how to switch among the options to complete the tasks in stages (i.e., horizontal and vertical) with the learned high-level policy. Thus, our algorithm can effectively capture the compositional structure within the tasks and leverage it in the multi-task policy learning, which explains its superior performance. More analysis results of the learned hierarchical policy on HalfCheetah-MultiVel and Walker-RandParam are in Appendix E.3.



(a) PointCell-MultiGoal

(b) PointRoom

(c) PointMaze

(d) Recovered Hier Policy

(e) Transfer Learning on PointRoom

(f) Transfer Learning on PointMaze

Figure 3. (a) The environment for multi-task learning with MH-AIRL. (d) Visualization of the learned hierarchical policy in (a). (b)(c) New task settings for evaluating the learned options in (a). (e)(f) Comparison results on (b)(c) between our proposed HRL algorithm (i.e., HPPO) initialized with the transferred options (i.e., HPPO-init) and other SOTA HRL and RL baselines.

Next, previous Meta/Multi-task Learning algorithms can learn a policy for a class of tasks whose contexts follow a certain distribution, but the learned policy cannot be transferred as a whole to tasks out of this class. In contrast, our algorithm recovers a hierarchical policy, of which the low-level part can be reused as basic skills for new tasks not necessarily in the same class, resulting in substantially improved transferability of the learned policy. To show this, we reuse the options discovered in PointCell as the initialization of the low-level part of the hierarchical policy for the goal-achieving tasks in new scenarios – PointRoom and PointMaze (Figure 3(b) and 3(c)). In each scenario, we select 4 challenging goals (starting from the center point) for evaluation, which are labeled as red points in the figure. Unlike the other evaluation tasks, we provide the agent sparse reward signals (a positive reward for reaching the goal only) instead of expert data, so they are RL rather than IL tasks. We use HPPO proposed in Section 4.3 and initialize it with the transferred options (i.e., HPPO-init). To be more convincing, we use two other SOTA HRL and RL algorithms – **DAC** (Zhang & Whiteson, 2019) and **PPO** (Schulman et al., 2017), as baselines. In Figure 3(e) and 3(f), we plot the episodic reward change in the training process of each algorithm, where the solid line and shadow represent the mean and standard deviation of the performance across the 4 different goals in each scenario. We can see that the reuse of options significantly accelerate the learning process and the newly proposed HRL algorithm performs much better than the baselines. Note that the other algorithms are trained for more episodes since they do not adopt the transferred options. We show that, in scenarios for which we do not have expert data or dense rewards, we can make use of the basic skills learned from expert demonstrations for similar task scenarios to effectively aid the learning, which provides a manner to bridge IL and RL.

6. Conclusion and Discussion

In this paper, we propose MH-AIRL to learn a hierarchical policy that can be adopted to perform a class of tasks, based on a mixture of multi-task unannotated expert data. We evaluate our algorithm on a series of challenging robotic multi-task settings. The results show that the multi-task hierarchical policies trained with MH-AIRL perform significantly better than the monotonic policies learned with SOTA Multi-task/Meta IL baselines. Further, with MH-AIRL, the agent can capture the subtask structures in each task and form a skill for each subtask. The basic skills can be reused for different tasks in that distribution to improve the expert data efficiency, and can even be transferred to more distinct tasks out of the distribution to solve long-timescale sparse-reward RL problems.

The primary limitation of our study is the inherent complexity of the overall framework, which comprises five networks as depicted in Figure 5. This complexity arises from our algorithm's integration of AIRL, context-based Meta IL, and the option framework. This amalgamation introduces certain challenges in the training process, particularly in determining the optimal number of training iterations for each network within each learning episode. After careful fine-tuning, we established a training iteration ratio of 1:3:10 for the discriminator, hierarchical policy, and variational posteriors, respectively. Despite this complexity, our evaluations across a wide variety of tasks utilized a consistent set of hyperparameters, showing the robustness of our approach.

References

Al-Abbasi, A. O., Ghosh, A., and Aggarwal, V. Deeppool: Distributed model-free algorithm for ride-sharing using deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 20(12):4714–4727, 2019.

Andreas, J., Klein, D., and Levine, S. Modular multitask reinforcement learning with policy sketches. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 166–175, 2017.

Balachandran, A., Aggarwal, V., Halepovic, E., Pang, J., Seshan, S., Venkataraman, S., and Yan, H. Modeling web quality-of-experience on cellular networks. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pp. 213–224, 2014.

Bian, X., Maldonado, O. M., and Hadfield, S. SKILL-IL: disentangling skill and knowledge in multitask imitation learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 7060–7065. IEEE, 2022.

Bottou, L. Large-scale machine learning with stochastic gradient descent. In *Proceedings of the 19th International Conference on Computational Statistics*, pp. 177–186. Springer, 2010.

Chen, J., Umrawal, A. K., Lan, T., and Aggarwal, V. Deepfreight: A model-free deep-reinforcement-learning-based algorithm for multi-transfer freight delivery. In *Proceedings of the 31st International Conference on Automated Planning and Scheduling*, pp. 510–518, 2021.

Chen, J., Aggarwal, V., and Lan, T. ODPP: A unified algorithm framework for unsupervised option discovery based on determinantal point process. *CoRR*, abs/2212.00211, 2022a.

Chen, J., Chen, J., Lan, T., and Aggarwal, V. Multi-agent covering option discovery based on kronecker product of factor graphs. *IEEE Transactions on Artificial Intelligence*, pp. 1–13, 2022b. doi: 10.1109/TAI.2022.3195818.

Chen, J., Chen, J., Lan, T., and Aggarwal, V. Scalable multi-agent covering option discovery based on kronecker graphs. In *Advances in Neural Information Processing Systems*, volume 35, pp. 30406–30418, 2022c.

Chen, J., Haliem, M., Lan, T., and Aggarwal, V. Multi-agent deep covering option discovery. *CoRR*, abs/2210.03269, 2022d.

Chen, J., Lan, T., and Aggarwal, V. Option-aware adversarial inverse reinforcement learning for robotic control. *arXiv preprint arXiv:2210.01969*, 2022e.

Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A. C., and Bengio, Y. A recurrent latent variable model for sequential data. In *Advances in Neural Information Processing Systems 28*, pp. 2980–2988, 2015.

Cover, T. M. *Elements of information theory*. John Wiley & Sons, 1999.

Deisenroth, M. P., Englert, P., Peters, J., and Fox, D. Multi-task policy search for robotics. In *IEEE International Conference on Robotics and Automation*, pp. 3876–3881. IEEE, 2014.

Devin, C., Geng, D., Abbeel, P., Darrell, T., and Levine, S. Compositional plan vectors. In *Advances in Neural Information Processing Systems 32*, pp. 14963–14974, 2019.

Duminy, N., Nguyen, S. M., Zhu, J., Duhaut, D., and Kerdreux, J. Intrinsically motivated open-ended multi-task learning using transfer learning to discover task hierarchy. *CoRR*, abs/2102.09854, 2021.

Eddy, S. R. Hidden markov models. *Current Opinion in Structural Biology*, 6(3):361–365, 1996. ISSN 0959-440X.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 1126–1135, 2017a.

Finn, C., Yu, T., Zhang, T., Abbeel, P., and Levine, S. One-shot visual imitation learning via meta-learning. In *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78, pp. 357–368, 2017b.

Florensa, C., Duan, Y., and Abbeel, P. Stochastic neural networks for hierarchical reinforcement learning. In *Proceedings of the 5th International Conference on Learning Representations*. OpenReview.net, 2017.

Fox, R., Berenstein, R., Stoica, I., and Goldberg, K. Multi-task hierarchical imitation learning for home automation. In *Proceedings of the 15th IEEE International Conference on Automation Science and Engineering*, pp. 1–8. IEEE, 2019.

Fu, J., Luo, K., and Levine, S. Learning robust rewards with adversarial inverse reinforcement learning. *CoRR*, abs/1710.11248, 2017.

Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. D4RL: datasets for deep data-driven reinforcement learning. *CoRR*, abs/2004.07219, 2020.

Fu, X., Peddireddy, D., Aggarwal, V., and Jun, M. B.-G. Improved dexel representation: A 3-d cnn geometry descriptor for manufacturing cad. *IEEE Transactions on Industrial Informatics*, 18(9):5882–5892, 2021.

Galvin, D. Three tutorial lectures on entropy and counting. *arXiv preprint arXiv:1406.7872*, 2014.

Gao, C., Jiang, Y., and Chen, F. Transferring hierarchical structures with dual meta imitation learning. In *Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 762–773, 2022.

Geng, N., Bai, Q., Liu, C., Lan, T., Aggarwal, V., Yang, Y., and Xu, M. A reinforcement learning framework for vehicular network routing under peak and average constraints. *IEEE Transactions on Vehicular Technology*, 2023.

Ghasemipour, S. K. S., Gu, S., and Zemel, R. S. Smile: Scalable meta inverse reinforcement learning through context-conditional policies. In *Advances in Neural Information Processing Systems 32*, pp. 7879–7889, 2019.

Gonzalez, G., Balakuntala, M., Agarwal, M., Low, T., Knoth, B., Kirkpatrick, A. W., McKee, J., Hager, G., Aggarwal, V., Xue, Y., et al. Asap: A semi-autonomous precise system for telesurgery during communication delays. *IEEE Transactions on Medical Robotics and Bionics*, 5(1):66–78, 2023.

Gupta, A., Kumar, V., Lynch, C., Levine, S., and Hausman, K. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. In *Proceedings of the 3rd Annual Conference on Robot Learning*, volume 100, pp. 1025–1037, 2019.

Higgins, I., Matthey, L., Pal, A., Burgess, C. P., Glorot, X., Botvinick, M. M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

Ho, J. and Ermon, S. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems 29*, pp. 4565–4573, 2016.

Jensen, J. L. W. V. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta mathematica*, 30(1):175–193, 1906.

Jing, M., Huang, W., Sun, F., Ma, X., Kong, T., Gan, C., and Li, L. Adversarial option-aware hierarchical imitation learning. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 5097–5106, 2021.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In *Proceedings of the 2nd International Conference on Learning Representations*, 2014.

Kosiorek, A. R., Sabour, S., Teh, Y. W., and Hinton, G. E. Stacked capsule autoencoders. In *Advances in Neural Information Processing Systems 32*, pp. 15486–15496, 2019.

Li, C., Song, D., and Tao, D. The skill-action architecture: Learning abstract action embeddings for reinforcement learning. In *Submissions of the 9th International Conference on Learning Representations*, 2021.

Luo, X., Ma, X., Munden, M., Wu, Y.-J., and Jiang, Y. A multisource data approach for estimating vehicle queue length at metered on-ramps. *Journal of Transportation Engineering, Part A: Systems*, 148(2):04021117, 2022.

Ma, X., Karimpour, A., and Wu, Y.-J. Statistical evaluation of data requirement for ramp metering performance assessment. *Transportation Research Part A: Policy and Practice*, 141:248–261, 2020. ISSN 0965-8564.

Mangal, S., Joshi, P., and Modak, R. LSTM vs. GRU vs. bidirectional RNN for script generation. *CoRR*, abs/1908.04332, 2019.

Massey, J. et al. Causality, feedback and directed information. In *Proc. Int. Symp. Inf. Theory Applic.(ISITA-90)*, pp. 303–305, 1990.

Ng, A. Y. and Russell, S. Algorithms for inverse reinforcement learning. In *Proceedings of the 7th International Conference on Machine Learning*, pp. 663–670. Morgan Kaufmann, 2000.

Ng, A. Y., Harada, D., and Russell, S. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the 16th International Conference on Machine Learning*, pp. 278–287. Morgan Kaufmann, 1999.

Peddireddy, D., Fu, X., Shankar, A., Wang, H., Joung, B. G., Aggarwal, V., Sutherland, J. W., and Jun, M. B.-G. Identifying manufacturability and machining processes using deep 3d convolutional networks. *Journal of Manufacturing Processes*, 64:1336–1348, 2021. ISSN 1526-6125.

Pomerleau, D. Efficient training of artificial neural networks for autonomous navigation. *Neural Computation*, 3(1): 88–97, 1991.

Ross, S., Gordon, G. J., and Bagnell, D. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, volume 15, pp. 627–635, 2011.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

Sharma, M., Sharma, A., Rhinehart, N., and Kitani, K. M. Directed-info GAIL: learning hierarchical policies from unsegmented demonstrations using directed information. In *Proceedings of the 7th International Conference on Learning Representations*. OpenReview.net, 2019.

Singh, A., Jang, E., Irpan, A., Kappler, D., Dalal, M., Levine, S., Khansari, M., and Finn, C. Scalable multi-task imitation learning with autonomous improvement. In *2020 IEEE International Conference on Robotics and Automation*, pp. 2167–2173. IEEE, 2020.

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

Sutton, R. S., Precup, D., and Singh, S. P. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112: 181–211, 1999.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pp. 5998–6008, 2017.

Wang, P., Liu, D., Chen, J., Li, H., and Chan, C. Decision making for autonomous driving via augmented adversarial inverse reinforcement learning. In *IEEE International Conference on Robotics and Automation*, pp. 1036–1042. IEEE, 2021.

Yu, L., Yu, T., Finn, C., and Ermon, S. Meta-inverse reinforcement learning with probabilistic context variables. In *Advances in Neural Information Processing Systems 32*, pp. 11749–11760, 2019.

Yu, T., Abbeel, P., Levine, S., and Finn, C. One-shot hierarchical imitation learning of compound visuomotor tasks. *CoRR*, abs/1810.11043, 2018a.

Yu, T., Finn, C., Xie, A., Dasari, S., Zhang, T., Abbeel, P., and Levine, S. One-shot imitation from observing humans via domain-adaptive meta-learning. *arXiv preprint arXiv:1802.01557*, 2018b.

Zhang, S. and Whiteson, S. DAC: the double actor-critic architecture for learning options. In *Advances in Neural Information Processing Systems 32*, pp. 2010–2020, 2019.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., and Dey, A. K. Maximum entropy inverse reinforcement learning. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, pp. 1433–1438, 2008.

A. Appendix on the Background and Related Works

A.1. AIRL Framework to Solve Equation 3

For each task C, we need to recover the task-specific reward function $\mathcal{R}_\vartheta(S, A|C)$ and policy $\pi(A|S, C)$ based on the corresponding expert trajectories $\tau_E \sim \pi_E(\cdot|C)$ which can be solved by AIRL as mentioned in Section 3.1. Thus, we have the following objective functions for training, which is a simple extension of AIRL (Ghasemipour et al., 2019; Yu et al., 2019):

$$\min_{\vartheta} \mathbb{E}_C \left[-\mathbb{E}_{\tau_E \sim \pi_E(\cdot|C)} \left[\sum_{t=0}^{T-1} \log D_\vartheta(S_t, A_t|C) \right] - \mathbb{E}_{\tau \sim \pi(\cdot|C)} \left[\sum_{t=0}^{T-1} \log(1 - D_\vartheta(S_t, A_t|C)) \right] \right] \tag{13}$$

$$\max_{\pi} \mathbb{E}_C \left[\mathbb{E}_{\tau \sim \pi(\cdot|C)} \left[\sum_{t=0}^{T-1} \log D_\vartheta(S_t, A_t|C) - \log(1 - D_\vartheta(S_t, A_t|C)) \right] \right] \tag{14}$$

where $D_\vartheta(S, A|C) = \exp(f_\vartheta(S, A|C))/[\exp(f_\vartheta(S, A|C)) + \pi(A|S, C)]$.

A.2. Implementation of the Hierarchical Policy in the One-step Option Model

In this section, we give out the detailed structure design of the hierarchical policy introduced in Section 3.3, i.e., $\pi_\theta(Z|S, Z')$ and $\pi_\phi(A|S, Z)$, which is proposed in (Li et al., 2021). This part is not our contribution, so we only provide the details for the purpose of implementation.

As mentioned in Section 3.3, the structure design is based on the Multi-Head Attention (MHA) mechanism (Vaswani et al., 2017). An attention function can be described as mapping a query, i.e., $q \in \mathbb{R}^{d_k}$, and a set of key-value pairs, i.e., $K = [k_1 \cdots k_n]^T \in \mathbb{R}^{n \times d_k}$ and $V = [v_1 \cdots v_n]^T \in \mathbb{R}^{n \times d_v}$, to an output. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key. To be specific:

$$Attention(q, K, V) = \sum_{i=1}^{n} \left[\frac{\exp(q \cdot k_i)}{\sum_{j=1}^{n} \exp(q \cdot k_j)} \times v_i \right] \tag{15}$$

where q, K, V are learnable parameters, $\frac{\exp(q \cdot k_i)}{\sum_{j=1}^{n} \exp(q \cdot k_j)}$ represents the attention weight that the model should pay to item i. In MHA, the query and key-value pairs are first linearly projected h times to get h different queries, keys and values. Then, an attention function is performed on each of these projected versions of queries, keys and values in parallel to get h outputs which are then be concatenated and linearly projected to acquire the final output. The whole process can be represented as Equation 16, where $W_i^q \in \mathbb{R}^{d_k \times d_k}, W_i^K \in \mathbb{R}^{d_k \times d_k}, W_i^V \in \mathbb{R}^{d_v \times d_v}, W^O \in \mathbb{R}^{nd_v \times d_v}$ are the learnable parameters.

$$MHA(q, K, V) = Concat(head_1, \cdots, head_h)W^O,\ head_i = Attention(qW_i^q, KW_i^K, VW_i^V) \tag{16}$$

In this work, the option is represented as an N-dimensional one-hot vector, where N denotes the total number of options to learn. The high-level policy $\pi_\theta(Z|S, Z')$ has the structure shown as:

$$q = linear(Concat[S, W_C^T Z']),\ dense_Z = MHA(q, W_C, W_C),\ Z \sim Categorical(\cdot|dense_Z) \tag{17}$$

$W_C \in \mathbb{R}^{N \times E}$ is the option context matrix of which the i-th row represents the context embedding of the option i. W_C is also used as the key and value matrix for the MHA, so $d_k = d_v = E$ in this case. Note that W_C is only updated in the MHA module. Intuitively, $\pi_\theta(Z|S, Z')$ attends to all the option context embeddings in W_C according to S and Z'. If Z' still fits S, $\pi_\theta(Z|S, Z')$ will assign a larger attention weight to Z' and thus has a tendency to continue with it; otherwise, a new skill with better compatibility will be sampled.

As for the low-level policy $\pi_\phi(A|S, Z)$, it has the following structure:

$$dense_A = MLP(S, W_C^T Z),\ A \sim Categorical/Gaussian(\cdot|dense_A) \tag{18}$$

where MLP represents a multilayer perceptron, A follows a categorical distribution for the discrete case or a gaussian distribution for the continuous case. The context embedding corresponding to Z, i.e., $W_C^T Z$, instead of Z only, is used as input of π_ϕ since it can encode multiple properties of the option Z (Kosiorek et al., 2019).

B. Appendix on the Hierarchical Latent context Structure

B.1. A Lower Bound of the Directed Information Objective

In this section, we give out the derivation of a lower bound of the directed information from the trajectory sequence $X_{0:T}$ to the local latent context sequence $Z_{0:T}$ conditioned on the global latent context C, i.e., $I(X_{0:T} \rightarrow Z_{0:T}|C)$ as follows:

$$\begin{aligned}
I(X_{0:T} \rightarrow Z_{0:T}|C) &= \sum_{t=1}^{T}[I(X_{0:t}; Z_t|Z_{0:t-1}, C)] \\
&= \sum_{t=1}^{T}[H(Z_t|Z_{0:t-1}, C) - H(Z_t|X_{0:t}, Z_{0:t-1}, C)] \\
&\geq \sum_{t=1}^{T}[H(Z_t|X_{0:t-1}, Z_{0:t-1}, C) - H(Z_t|X_{0:t}, Z_{0:t-1}, C)] \\
&= \sum_{t=1}^{T}[H(Z_t|X_{0:t-1}, Z_{0:t-1}, C) + \\
&\qquad \sum_{\substack{X_{0:t},C,\\ Z_{0:t-1}}} P(X_{0:t}, Z_{0:t-1}, C) \sum_{Z_t} P(Z_t|X_{0:t}, Z_{0:t-1}, C) \log P(Z_t|X_{0:t}, Z_{0:t-1}, C)]
\end{aligned} \tag{19}$$

In Equation 19, $I(Var_1; Var_2|Var_3)$ denotes the conditional mutual information, $H(Var_1|Var_2)$ denotes the conditional entropy, and the inequality holds because of the basic property related to conditional entropy: increasing conditioning cannot increase entropy (Galvin, 2014). $H(Z_t|X_{0:t-1}, Z_{0:t-1}, C)$ is the entropy of the high-level policy $\pi_\theta(Z_t|S_{t-1}, Z_{t-1})$, where the other variables in $X_{0:t-1}, Z_{0:t-1}$ are neglected due to the one-step Markov assumption, and more convenient to obtain. Further, the second term in the last step can be processed as follows:

$$\begin{aligned}
&\sum_{Z_t} P(Z_t|X_{0:t}, Z_{0:t-1}, C) \log P(Z_t|X_{0:t}, Z_{0:t-1}, C) \\
&= \sum_{Z_t} P(Z_t|X_{0:t}, Z_{0:t-1}, C) \left[\log \frac{P(Z_t|X_{0:t}, Z_{0:t-1}, C)}{P_\omega(Z_t|X_{0:t}, Z_{0:t-1}, C)} + \log P_\omega(Z_t|X_{0:t}, Z_{0:t-1}, C)\right] \\
&= D_{KL}(P(\cdot|X_{0:t}, Z_{0:t-1}, C)||P_\omega(\cdot|X_{0:t}, Z_{0:t-1}, C)) + \sum_{Z_t} P(Z_t|X_{0:t}, Z_{0:t-1}, C) \log P_\omega(Z_t|X_{0:t}, Z_{0:t-1}, C) \\
&\geq \sum_{Z_t} P(Z_t|X_{0:t}, Z_{0:t-1}, C) \log P_\omega(Z_t|X_{0:t}, Z_{0:t-1}, C)
\end{aligned} \tag{20}$$

where $D_{KL}(\cdot)$ denotes the Kullback-Leibler (KL) Divergence which is non-negative (Cover, 1999), $P_\omega(Z_t|X_{0:t}, Z_{0:t-1}, C)$ is a variational estimation of the posterior distribution of Z_t given $X_{0:t}$ and $Z_{0:t-1}$, i.e., $P(Z_t|X_{0:t}, Z_{0:t-1}, C)$, which is modeled as a recurrent neural network with the parameter set ω in our work. Based on Equation 19 and 20, we can obtain a lower bound of $I(X_{0:T} \rightarrow Z_{0:T}|C)$ denoted as L^{DI}:

$$L^{DI} = \sum_{t=1}^{T}[\sum_{\substack{X_{0:t},C,\\ Z_{0:t}}} P(X_{0:t}, Z_{0:t}, C) \log P_\omega(Z_t|X_{0:t}, Z_{0:t-1}, C) + H(Z_t|X_{0:t-1}, Z_{0:t-1}, C)] \tag{21}$$

Note that the joint distribution $P(X_{0:t}, Z_{0:t}, C)$ has a recursive definition as follows:

$$\begin{aligned}
&P(X_{0:t}, Z_{0:t}, C) = prior(C)P(X_{0:t}, Z_{0:t}|C) \\
&= prior(C)P(X_t|X_{0:t-1}, Z_{0:t}, C)P(Z_t|X_{0:t-1}, Z_{0:t-1}, C)P(X_{0:t-1}, Z_{0:t-1}|C)
\end{aligned} \tag{22}$$

$$P(X_0, Z_0|C) = P((S_0, A_{-1}), Z_0|C) = \mu(S_0|C) \tag{23}$$

where $\mu(S_0|C)$ denotes the distribution of the initial states for task C. Equation 23 holds because A_{-1} and Z_0 are dummy variables which are only for simplifying notations and never executed and set to be constant across different tasks. Based on

Equation 22 and 23, we can get:

$$\begin{aligned}
P(X_{0:t}, Z_{0:t}, C) &= prior(C)\mu(S_0|C)\prod_{i=1}^{t} P(Z_i|X_{0:i-1}, Z_{0:i-1}, C)P(X_i|X_{0:i-1}, Z_{0:i}, C) \\
&= prior(C)\mu(S_0|C)\prod_{i=1}^{t} P(Z_i|X_{0:i-1}, Z_{0:i-1}, C)P((S_i, A_{i-1})|X_{0:i-1}, Z_{0:i}, C) \\
&= prior(C)\mu(S_0|C)\prod_{i=1}^{t} P(Z_i|X_{0:i-1}, Z_{0:i-1}, C)P(A_{i-1}|X_{0:i-1}, Z_{0:i}, C)\mathcal{P}(S_i|S_{i-1}, A_{i-1}, C) \\
&= prior(C)\mu(S_0|C)\prod_{i=1}^{t} \pi_\theta(Z_i|S_{i-1}, Z_{i-1}, C)\pi_\phi(A_{i-1}|S_{i-1}, Z_i, C)\mathcal{P}(S_i|S_{i-1}, A_{i-1}, C)
\end{aligned} \tag{24}$$

In Equation 24, $prior(C)$ is the known prior distribution of the task context C, $\mathcal{P}(S_i|S_{i-1}, A_{i-1}, C)$ is the transition dynamic of task C, $P(Z_i|X_{0:i-1}, Z_{0:i-1}, C)$ and $P(A_{i-1}|X_{0:i-1}, Z_{0:i}, C)$ can be replaced with π_θ and π_ϕ, respectively, due to the one-step Markov assumption.

To sum up, we can adopt the high-level policy, low-level policy and variational posterior to get an estimation of the lower bound of the directed information objective through Monte Carlo sampling (Sutton & Barto, 2018) according to Equation 21 and 24, which can then be used to optimize the three networks.

B.2. A Lower Bound of the Mutual Information Objective

In this section, we give out the derivation of a lower bound of the mutual information between the trajectory sequence $X_{0:T}$ and its corresponding task context C, i.e., $I(X_{0:T}; C)$.

$$\begin{aligned}
&I(X_{0:T}; C) = H(C) - H(C|X_{0:T}) \\
&= H(C) + \sum_{X_{0:T}} P(X_{0:T}) \sum_{C} P(C|X_{0:T}) \log P(C|X_{0:T}) \\
&= H(C) + \sum_{X_{0:T}} P(X_{0:T}) \sum_{C} P(C|X_{0:T}) \log \frac{P(C|X_{0:T})}{P_\psi(C|X_{0:T})} + \sum_{X_{0:T}, C} P(X_{0:T}, C) \log P_\psi(C|X_{0:T}) \\
&= H(C) + \sum_{X_{0:T}} P(X_{0:T}) D_{KL}(P(\cdot|X_{0:T}||P_\psi(\cdot|X_{0:T})) + \sum_{X_{0:T}, C} P(X_{0:T}, C) \log P_\psi(C|X_{0:T}) \\
&\geq H(C) + \sum_{X_{0:T}, C} P(X_{0:T}, C) \log P_\psi(C|X_{0:T}) \\
&= H(C) + \sum_{C} prior(C) \sum_{X_{0:T}} P(X_{0:T}|C) \log P_\psi(C|X_{0:T}) \\
&= H(C) + \sum_{C} prior(C) \sum_{X_{0:T}, Z_{0:T}} P(X_{0:T}, Z_{0:T}|C) \log P_\psi(C|X_{0:T})
\end{aligned} \tag{25}$$

In Equation 25, $H(\cdot)$ denotes the entropy, $prior(C)$ denotes the known prior distribution of the task context C, $P(X_{0:T}, Z_{0:T}|C)$ can be calculated with Equation 24 by setting $t = T$, and $P_\psi(C|X_{0:T})$ is a variational estimation of the posterior distribution $P(C|X_{0:T})$ which is implemented as a recurrent neural network with the parameter set ψ. Note that the inequality holds because the KL-Divergence, i.e., $D_{KL}(\cdot)$, is non-negative.

B.3. The Analogy with the VAE Framework

Variational Autoencoder (VAE) (Kingma & Welling, 2014) learns a probabilistic **encoder** $P_\eta(V|U)$ and **decoder** $P_\xi(U|V)$ which map between data U and latent variables V by optimizing the evidence lower bound (ELBO) on the marginal distribution $P_\xi(U)$, assuming the prior distributions $P^U(\cdot)$ and $P^V(\cdot)$ over the data and latent variables respectively. The authors of (Higgins et al., 2017) extend the VAE approach by including a parameter β to control the capacity of the latent V,

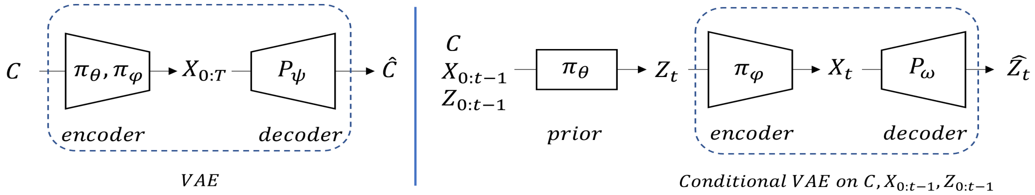


Figure 4. The analogy of our learning framework with the VAE structure.

of which the ELBO is:

$$\max_{\eta,\xi} \mathop{\mathbb{E}}_{\substack{U\sim P^U(\cdot)\\ V\sim P_\eta(\cdot|U)}} \left[\log P_\xi(U|V) - \beta D_{KL}(P_\eta(V|U)||P^V(V))\right] \tag{26}$$

The first term can be viewed as the reconstruction accuracy of the data U from V, and the second term works as a regularizer for the distribution of the latent variables V, where D_{KL} denotes the KL Divergence (Cover, 1999). VAE can efficiently solve the posterior inference problem for datasets with continuous latent variables where the true posterior is intractable, through fitting an approximate inference model P_ξ (i.e., the variational posterior). The variational lower bound, i.e., ELBO, can be straightforwardly optimized using standard stochastic gradient methods, e.g., SGD (Bottou, 2010).

As shown in Figure 4, the optimization of L^{MI} (Equation 4) can be viewed as using π_θ and π_ϕ as the encoder and P_ψ as the decoder and then minimizing the reconstruction error of C from $X_{0:T}$, and the regularizer term in Equation 26 is neglected (i.e., $\beta = 0$). As for the optimization of L^{DI} (Equation 4), at each timestep t, π_ϕ and P_ω form a conditional VAE between Z_t and X_t, which is conditioned on the history information and task code, i.e., $(X_{0:t-1}, Z_{0:t-1}, C)$, with the prior distribution of Z_t provided by π_θ. Compared with the VAE objective (i.e., Equation 26), π_ϕ and P_ω in L^{DI} work as the encoder and decoder respectively; π_θ provides the prior, which corresponds to $P^U(\cdot)$.

Both P_ψ and P_ω use sequential data as input and thus are implemented with RNN. The variational posterior for the task code, i.e., $P_\psi(C|X_{0:T})$ takes the trajectory $X_{0:T}$ as input and is implemented as a bidirectional GRU (Mangal et al., 2019) to make sure that both the beginning and end of the trajectory are equally important. On the other hand, the variational posterior for the local latent code, i.e., $P_\omega(Z_t|X_{0:t}, Z_{0:t-1}, C)$, is modeled as $P_\omega(Z_t|X_t, Z_{t-1}, C, h_{t-1})$, where h_{t-1} is the internal hidden state of an RNN. h_{t-1} is recursively maintained with the time series using the GRU rule, i.e., $h_{t-1} = GRU(X_{t-1}, Z_{t-2}, h_{t-2})$, to embed the history information in the trajectory, i.e., $X_{0:t-1}$ and $Z_{0:t-2}$. Note that the RNN-based posterior has been used and justified in the process for sequential data (Chung et al., 2015).

C. Appendix on Hierarchical AIRL

C.1. Derivation of the MLE Objective

In Equation 27, Z_0 is a dummy variable which is assigned before the episode begins and never executed. It's implemented as a constant across different episodes, so we have $P(S_0, Z_0|C) = P(S_0|C) = \mu(S_0|C)$, where $\mu(\cdot|C)$ denotes the initial state distribution for task C. On the other hand, we have $P(S_{t+1}, Z_{t+1}|S_t, Z_t, Z_{t+1}, A_t, C) = P(Z_{t+1}|S_t, Z_t, Z_{t+1}, A_t, C)P(S_{t+1}|S_t, Z_t, Z_{t+1}, A_t, C) = \mathcal{P}(S_{t+1}|S_t, A_t, C)$, since the transition dynamic $\mathcal{P}$ is irrelevant to the local latent codes Z and only related the task context C.

$$\begin{aligned}
P_{\vartheta}(X_{0:T}, Z_{0:T}|C) &\propto \mu(\widetilde{S}_0|C) \prod_{t=0}^{T-1} \mathcal{P}(\widetilde{S}_{t+1}|\widetilde{S}_t, \widetilde{A}_t, C) \exp(\mathcal{R}_{\vartheta}(\widetilde{S}_t, \widetilde{A}_t|C)) \\
&= P(S_0, Z_0|C) \prod_{t=0}^{T-1} P(S_{t+1}, Z_{t+1}|S_t, Z_t, Z_{t+1}, A_t, C) \exp(\mathcal{R}_{\vartheta}(S_t, Z_t, Z_{t+1}, A_t|C)) \\
&= \mu(S_0|C) \prod_{t=0}^{T-1} \mathcal{P}(S_{t+1}|S_t, A_t, C) \exp(\mathcal{R}_{\vartheta}(S_t, Z_t, Z_{t+1}, A_t|C))
\end{aligned} \tag{27}$$

C.2. Justification of the Objective Function Design in Equation 8

In this section, we prove that by optimizing the objective functions shown in Equation 8, we can get the solution of the MLE problem shown as Equation 7, i.e., the task-conditioned hierarchical reward function and policy of the expert.

In Appendix A of (Fu et al., 2017), they show that the discriminator objective (the first equation in 8) is equivalent to the MLE objective (Equation 7) where f_{ϑ} serves as R_{ϑ}, when $D_{KL}(\pi(\tau)||\pi_E(\tau))$ is minimized. The same conclusion can be acquired by simply replacing $\{S_t, A_t, \tau\}$ with $\{(S_t, Z_t), (Z_{t+1}, A_t), (X_{0:T}, Z_{0:T})\}$, i.e., the extended definition of the state, action and trajectory, in the original proof, which we don't repeat here. Then, we only need to prove that $\mathbb{E}_C\left[D_{KL}(\pi_{\theta,\phi}(X_{0:T}, Z_{0:T}|C)||\pi_E(X_{0:T}, Z_{0:T}|C))\right]$ can be minimized through the second equation in 8:

$$\begin{aligned}
&\max_{\theta,\phi} \mathbb{E}_{C\sim prior(\cdot),(X_{0:T},Z_{0:T})\sim\pi_{\theta,\phi}(\cdot|C)} \sum_{t=0}^{T-1} R_{IL}^t \\
&= \max_{\theta,\phi} \mathop{\mathbb{E}}_{C,X_{0:T},Z_{0:T}} \left[\sum_{t=0}^{T-1} \log D_{\vartheta}(S_t, Z_t, Z_{t+1}, A_t|C) - \log(1 - D_{\vartheta}(S_t, Z_t, Z_{t+1}, A_t|C))\right] \\
&= \max_{\theta,\phi} \mathop{\mathbb{E}}_{C,X_{0:T},Z_{0:T}} \left[\sum_{t=0}^{T-1} f_{\vartheta}(S_t, Z_t, Z_{t+1}, A_t|C) - \log \pi_{\theta,\phi}(Z_{t+1}, A_t|S_t, Z_t, C)\right] \\
&= \max_{\theta,\phi} \mathop{\mathbb{E}}_{C,X_{0:T},Z_{0:T}} \left[\sum_{t=0}^{T-1} f_{\vartheta}(S_t, Z_t, Z_{t+1}, A_t|C) - \log(\pi_{\theta}(Z_{t+1}|S_t, Z_t, C)\pi_{\phi}(A_t|S_t, Z_{t+1}, C))\right] \\
&= \max_{\theta,\phi} \mathop{\mathbb{E}}_{C,X_{0:T},Z_{0:T}} \left[\log \frac{\prod_{t=0}^{T-1} \exp(f_{\vartheta}(S_t, Z_t, Z_{t+1}, A_t|C))}{\prod_{t=0}^{T-1} \pi_{\theta}(Z_{t+1}|S_t, Z_t, C)\pi_{\phi}(A_t|S_t, Z_{t+1}, C)}\right] \\
&\iff \max_{\theta,\phi} \mathop{\mathbb{E}}_{C,X_{0:T},Z_{0:T}} \left[\log \frac{\prod_{t=0}^{T-1} \exp(f_{\vartheta}(S_t, Z_t, Z_{t+1}, A_t|C))/Z_{\vartheta}^C}{\prod_{t=0}^{T-1} \pi_{\theta}(Z_{t+1}|S_t, Z_t, C)\pi_{\phi}(A_t|S_t, Z_{t+1}, C)}\right]
\end{aligned} \tag{28}$$

Note that $Z_{\vartheta}^C = \sum_{X_{0:T},Z_{0:T}} \widehat{P}_{\vartheta}(X_{0:T}, Z_{0:T}|C)$ (defined in Equation 7) is the normalized function parameterized with ϑ, so the introduction of Z_{ϑ}^C will not influence the optimization with respect to θ and ϕ and the equivalence at the last step holds. Also, the second equality shows that the task-conditioned hierarchical policy is recovered by optimizing an entropy-regularized policy objective where f_ϑ serves as R_ϑ. Further, we have:

$$\begin{aligned}
&\max_{\theta,\phi} \mathop{\mathbb{E}}_{C,X_{0:T},Z_{0:T}} \left[\log \frac{\prod_{t=0}^{T-1} \exp(f_\vartheta(S_t, Z_t, Z_{t+1}, A_t|C))/Z_\vartheta^C}{\prod_{t=0}^{T-1} \pi_\theta(Z_{t+1}|S_t, Z_t, C)\pi_\phi(A_t|S_t, Z_{t+1}, C)}\right] \\
&= \max_{\theta,\phi} \mathop{\mathbb{E}}_{C,X_{0:T},Z_{0:T}} \left[\log \frac{\mu(S_0|C)\prod_{t=0}^{T-1} \mathcal{P}(S_{t+1}|S_t, A_t, C)\prod_{t=0}^{T-1} \exp(f_\vartheta(S_t, Z_t, Z_{t+1}, A_t|C))/Z_\vartheta^C}{\mu(S_0|C)\prod_{t=0}^{T-1} \mathcal{P}(S_{t+1}|S_t, A_t, C)\prod_{t=0}^{T-1} \pi_\theta(Z_{t+1}|S_t, Z_t, C)\pi_\phi(A_t|S_t, Z_{t+1}, C)}\right] \\
&= \max_{\theta,\phi} \mathbb{E}_{C\sim prior(\cdot),(X_{0:T},Z_{0:T})\sim\pi_{\theta,\phi}(\cdot|C)} \left[\log \frac{\pi_E(X_{0:T}, Z_{0:T}|C)}{\pi_{\theta,\phi}(X_{0:T}, Z_{0:T}|C)}\right] \\
&= \max_{\theta,\phi} \mathbb{E}_{C\sim prior(\cdot)} \left[-D_{KL}(\pi_{\theta,\phi}(X_{0:T}, Z_{0:T}|C)||\pi_E(X_{0:T}, Z_{0:T}|C))\right] \\
&\iff \min_{\theta,\phi} \mathbb{E}_{C\sim prior(\cdot)} \left[D_{KL}(\pi_{\theta,\phi}(X_{0:T}, Z_{0:T}|C)||\pi_E(X_{0:T}, Z_{0:T}|C))\right]
\end{aligned} \tag{29}$$

where the second equality holds because of the definition of π_E (Equation 7 with f_ϑ serving as R_ϑ) and $\pi_{\theta,\phi}$ (Equation 34).

C.3. Justification of the EM-style Adaption

Given only a dataset of expert trajectories, i.e., $D_E \triangleq \{X_{0:T}\}$, we can still maximize the likelihood estimation $\mathbb{E}_{X_{0:T}\sim D_E}[\log P_\vartheta(X_{0:T})]$ through an EM-style adaption: (We use $X_{0:T}, C, Z_{0:T}$ instead of $X_{0:T}^E, C_E, Z_{0:T}^E$ for simplicity.)

$$\begin{aligned}
&\mathbb{E}_{X_{0:T}\sim D_E}[\log P_\vartheta(X_{0:T})] = \mathbb{E}_{X_{0:T}\sim D_E}\left[\log\left[\sum_{C,Z_{0:T}} P_\vartheta(X_{0:T}, C, Z_{0:T})\right]\right] \\
&= \mathbb{E}_{X_{0:T}\sim D_E}\left[\log\left[\sum_{C,Z_{0:T}} \frac{P_\vartheta(X_{0:T}, C, Z_{0:T})}{P_{\overline{\vartheta}}(C, Z_{0:T}|X_{0:T})} P_{\overline{\vartheta}}(C, Z_{0:T}|X_{0:T})\right]\right] \\
&= \mathbb{E}_{X_{0:T}\sim D_E}\left[\log\left[\mathbb{E}_{(C,Z_{0:T})\sim P_{\overline{\vartheta}}(\cdot|X_{0:T})} \frac{P_\vartheta(X_{0:T}, C, Z_{0:T})}{P_{\overline{\vartheta}}(C, Z_{0:T}|X_{0:T})}\right]\right] \\
&\geq \mathbb{E}_{X_{0:T}\sim D_E}\left[\mathbb{E}_{(C,Z_{0:T})\sim P_{\overline{\vartheta}}(\cdot|X_{0:T})} \log \frac{P_\vartheta(X_{0:T}, C, Z_{0:T})}{P_{\overline{\vartheta}}(C, Z_{0:T}|X_{0:T})}\right] \\
&= \mathbb{E}_{X_{0:T}\sim D_E, C\sim P_{\overline{\psi}}(\cdot|X_{0:T}), Z_{0:T}\sim P_{\overline{\varpi}}(\cdot|X_{0:T},C)}\left[\log \frac{P_\vartheta(X_{0:T}, C, Z_{0:T})}{P_{\overline{\vartheta}}(C, Z_{0:T}|X_{0:T})}\right] \\
&= \mathbb{E}_{X_{0:T},C,Z_{0:T}}[\log P_\vartheta(X_{0:T}, C, Z_{0:T})] - \mathbb{E}_{X_{0:T},C,Z_{0:T}}\left[\log P_{\overline{\vartheta}}(C, Z_{0:T}|X_{0:T})\right] \\
&= \mathbb{E}_{X_{0:T},C,Z_{0:T}}[\log P_\vartheta(X_{0:T}, Z_{0:T}|C)] - \mathbb{E}_{X_{0:T},C,Z_{0:T}}\left[-\log prior(C) + \log P_{\overline{\vartheta}}(C, Z_{0:T}|X_{0:T})\right]
\end{aligned} \tag{30}$$

where we adopt the Jensen's inequality (Jensen, 1906) in the 4-th step. Also, we note that $P_{\overline{\psi},\overline{\varpi}}(C, Z_{0:T}|X_{0:T})$ provides a posterior distribution of $(C, Z_{0:T})$, which corresponds to the generating process led by the hierarchical policy. As justified in C.2, the hierarchical policy is trained with the reward function parameterized with $\overline{\vartheta}$. Thus, the hierarchical policy is a function of $\overline{\vartheta}$, and the network $P_{\overline{\psi},\overline{\varpi}}$ corresponding to the hierarchical policy provides a posterior distribution related to the parameter set $\overline{\vartheta}$, i.e., $(C, Z_{0:T}) \sim P_{\overline{\vartheta}}(\cdot|X_{0:T}) \iff C \sim P_{\overline{\psi}}(\cdot|X_{0:T}), Z_{0:T} \sim P_{\overline{\varpi}}(\cdot|X_{0:T}, C)$, due to which the 5-th step holds. Note that $\overline{\vartheta}, \overline{\psi}, \overline{\varpi}$ denote the parameters ϑ, ψ, ω before being updated in the M step.

In the second equality of Equation 30, we introduce the sampled global and local latent codes in the E step as discussed in Section 4.2. Then, in the M step, we optimize the objectives shown in Equation 4 and 8 for iterations, by replacing the samples in the first term of Equation 8 with $(X_{0:T}, C, Z_{0:T})$ collected in the E step. This is equivalent to solve the MLE problem: $\max_\vartheta \mathbb{E}_{X_{0:T}\sim D_E, C\sim P_{\overline{\psi}}(\cdot|X_{0:T}), Z_{0:T}\sim P_{\overline{\varpi}}(\cdot|X_{0:T},C)}[\log P_\vartheta(X_{0:T}, Z_{0:T}|C)]$, which is to maximize a lower bound of the original objective, i.e., $\mathbb{E}_{X_{0:T}\sim D_E}[\log P_\vartheta(X_{0:T})]$, as shown in the last step of Equation 30. Thus, the original objective can be optimized through this EM procedure. Note that the second term in the last step is a function of the old parameter $\overline{\vartheta}$ so that it can be overlooked when optimizing with respect to ϑ.

C.4. State-only Adaption of H-AIRL

In AIRL (Fu et al., 2017), they propose a two-component design for the discriminator as follows:

$$f_{\vartheta,\zeta}(S_t, S_{t+1}) = g_\vartheta(S_t) + \gamma h_\zeta(S_{t+1}) - h_\zeta(S_t) \tag{31}$$

where γ is the discount factor in MDP. Based on $f_{\vartheta,\zeta}(S_t, S_{t+1})$, they can further get $D_{\vartheta,\zeta}(S_t, S_{t+1})$ which is used in Equation 2 for AIRL training. As proved in (Fu et al., 2017), g_ϑ, h_ζ and $f_{\vartheta,\zeta}$ can recover the true reward, value and advantage function, respectively, under deterministic environments with a state-only ground truth reward. With this state-only design, the recovered reward function is disentangled from the dynamics of the environment in which it was trained, so that it can be directly transferred to environments with different transition dynamics, i.e., $\mathcal{P}$, for the policy training. Moreover, the additional shaping term h_ζ helps mitigate the effects of unwanted shaping on the reward approximator g_ϑ (Ng et al., 1999). This design can also be adopted to H-AIRL (Equation 8) by redefining Equation 31 on the extended state space (first defined in Section 4.2):

$$\begin{aligned} f_{\vartheta,\zeta}(\widetilde{S}_t, \widetilde{S}_{t+1}|C) &= g_\vartheta(\widetilde{S}_t|C) + \gamma h_\zeta(\widetilde{S}_{t+1}|C) - h_\zeta(\widetilde{S}_t|C) \\ &= g_\vartheta(S_t, Z_t|C) + \gamma h_\zeta(S_{t+1}, Z_{t+1}|C) - h_\zeta(S_t, Z_t|C) \end{aligned} \tag{32}$$

In this way, we can recover a hierarchical reward function conditioned on the task context C, i.e., $g_\vartheta(S_t, Z_t|C)$, which avoids unwanted shaping and is robust enough to be directly applied in a new task with different dynamic transition distribution from $prior(C)$. The proof can be done by simply replacing the state S in the original proof (Appendix C of (Fu et al., 2017)) with its extended definition $\widetilde{S}$, so we don't repeat it here.

D. The Proposed Actor-Critic Algorithm for Training

D.1. Gradients of the Mutual Information Objective Term

The objective function related to the mutual information:

$$L^{MI} = \sum_C prior(C) \sum_{X_{0:T}, Z_{0:T}} P(X_{0:T}, Z_{0:T}|C) \log P_\psi(C|X_{0:T}) \tag{33}$$

After introducing the one-step Markov assumption to Equation 24, we can calculate $P(X_{0:T}, Z_{0:T}|C)$ as Equation 34, where π_θ and π_ϕ represent the hierarchical policy in the one-step option framework.

$$P(X_{0:T}, Z_{0:T}|C) = \mu(S_0|C) \prod_{t=1}^{T} \pi_\theta(Z_t|S_{t-1}, Z_{t-1}, C)\pi_\phi(A_{t-1}|S_{t-1}, Z_t, C)\mathcal{P}(S_t|S_{t-1}, A_{t-1}, C) \tag{34}$$

First, the gradient with respect to ψ is straightforward as Equation 35, which can be optimized as a standard likelihood maximization problem.

$$\nabla_\psi L^{MI} = \sum_C prior(C) \sum_{X_{0:T}, Z_{0:T}} P(X_{0:T}, Z_{0:T}|C) \nabla_\psi \log P_\psi(C|X_{0:T}) \tag{35}$$

Now we give out the derivation of $\nabla_\theta L^{MI}$:

$$\begin{aligned} \nabla_\theta L^{MI} &= \sum_C prior(C) \sum_{X_{0:T}, Z_{0:T}} \nabla_\theta P_{\theta,\phi}(X_{0:T}, Z_{0:T}|C) \log P_\psi(C|X_{0:T}) \\ &= \sum_C prior(C) \sum_{X_{0:T}, Z_{0:T}} P_{\theta,\phi}(X_{0:T}, Z_{0:T}|C) \nabla_\theta \log P_{\theta,\phi}(X_{0:T}, Z_{0:T}|C) \log P_\psi(C|X_{0:T}) \\ &= \mathop{\mathbb{E}}_{\substack{C, X_{0:T}, \\ Z_{0:T}}} \left[\nabla_\theta \log P_{\theta,\phi}(X_{0:T}, Z_{0:T}|C) \log P_\psi(C|X_{0:T})\right] \\ &= \mathop{\mathbb{E}}_{\substack{C, X_{0:T}, \\ Z_{0:T}}} \left[\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(Z_t|S_{t-1}, Z_{t-1}, C) \log P_\psi(C|X_{0:T})\right] \end{aligned} \tag{36}$$

where the last equality holds because of Equation 34. With similar derivation as above, we have:

$$\nabla_\phi L^{MI} = \mathop{\mathbb{E}}_{\substack{C, X_{0:T}, \\ Z_{0:T}}} \left[\sum_{t=1}^{T} \nabla_\phi \log \pi_\phi(A_{t-1}|S_{t-1}, Z_t, C) \log P_\psi(C|X_{0:T})\right] \tag{37}$$

D.2. Gradients of the Directed Information Objective Term

Next, we give out the derivation of the gradients related to the directed information objective term, i.e., L^{DI}. We denote the two terms in Equation 21 as L_1^{DI} and L_2^{DI} respectively. Then, we have $\nabla_{\theta,\phi}L^{DI} = \nabla_{\theta,\phi}L_1^{DI} + \nabla_{\theta,\phi}L_2^{DI}$. The derivations are as follows:

$$\begin{aligned}
\nabla_\theta L_1^{DI} &= \sum_{t=1}^{T}\sum_{C} prior(C) \sum_{X_{0:t},Z_{0:t}} \nabla_\theta P_{\theta,\phi}(X_{0:t}, Z_{0:t}|C) \log P_\omega(Z_t|X_{0:t}, Z_{0:t-1}, C) \\
&= \sum_{t=1}^{T}\sum_{C} prior(C) \sum_{X_{0:t},Z_{0:t}} P_{\theta,\phi}(X_{0:t}, Z_{0:t}|C) \sum_{i=1}^{t} \nabla_\theta \log \pi_\theta(Z_i|S_{i-1}, Z_{i-1}, C) \log P_\omega^t \\
&= \sum_{t=1}^{T}\sum_{C} prior(C) \sum_{X_{0:t},Z_{0:t}} \sum_{\substack{X_{t+1:T},\\ Z_{t+1:T}}} P_{\theta,\phi}(X_{0:T}, Z_{0:T}|C) \sum_{i=1}^{t} \nabla_\theta \log \pi_\theta(Z_i|S_{i-1}, Z_{i-1}, C) \log P_\omega^t \\
&= \sum_{t=1}^{T}\sum_{C} prior(C) \sum_{X_{0:T},Z_{0:T}} P_{\theta,\phi}(X_{0:T}, Z_{0:T}|C) \sum_{i=1}^{t} \nabla_\theta \log \pi_\theta(Z_i|S_{i-1}, Z_{i-1}, C) \log P_\omega^t \\
&= \sum_{C} prior(C) \sum_{X_{0:T},Z_{0:T}} P_{\theta,\phi}(X_{0:T}, Z_{0:T}|C) \sum_{t=1}^{T} \log P_\omega^t \sum_{i=1}^{t} \nabla_\theta \log \pi_\theta(Z_i|S_{i-1}, Z_{i-1}, C) \\
&= \sum_{C} prior(C) \sum_{X_{0:T},Z_{0:T}} P_{\theta,\phi}(X_{0:T}, Z_{0:T}|C) \sum_{i=1}^{T} \nabla_\theta \log \pi_\theta(Z_i|S_{i-1}, Z_{i-1}, C) \sum_{t=i}^{T} \log P_\omega^t \\
&= \mathop{\mathbb{E}}_{\substack{C,X_{0:T},\\ Z_{0:T}}} \left[\sum_{i=1}^{T} \nabla_\theta \log \pi_\theta(Z_i|S_{i-1}, Z_{i-1}, C) \sum_{t=i}^{T} \log P_\omega(Z_t|X_{0:t}, Z_{0:t-1}, C) \right] \\
&= \mathop{\mathbb{E}}_{\substack{C,X_{0:T},\\ Z_{0:T}}} \left[\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(Z_t|S_{t-1}, Z_{t-1}, C) \sum_{i=t}^{T} \log P_\omega(Z_i|X_{0:i}, Z_{0:i-1}, C) \right]
\end{aligned} \tag{38}$$

where $P_\omega^t = P_\omega(Z_t|X_{0:t}, Z_{0:t-1}, C)$ for simplicity. The second equality in Equation 38 holds following the same derivation in Equation 36. Then, the gradient related to L_2^{DI} is:

$$\begin{aligned}
\nabla_\theta L_2^{DI} &= \nabla_\theta \sum_{t=1}^{T} H(Z_t|X_{0:t-1}, Z_{0:t-1}, C) \\
&= -\nabla_\theta[\sum_{t=1}^{T}\sum_{C} prior(C) \sum_{X_{0:t-1},Z_{0:t}} P_{\theta,\phi}(X_{0:t-1}, Z_{0:t}|C) \log P(Z_t|X_{0:t-1}, Z_{0:t-1}, C)] \\
&= -\nabla_\theta[\sum_{t=1}^{T}\sum_{C} prior(C) \sum_{X_{0:t-1},Z_{0:t}} P_{\theta,\phi}(X_{0:t-1}, Z_{0:t}|C) \log \pi_\theta(Z_t|S_{t-1}, Z_{t-1}, C)] \\
&= -\nabla_\theta[\sum_{C} prior(C) \sum_{X_{0:T},Z_{0:T}} P_{\theta,\phi}(X_{0:T}, Z_{0:T}|C) \sum_{t=1}^{T} \log \pi_\theta(Z_t|S_{t-1}, Z_{t-1}, C)] \\
&= -[\sum_{C} prior(C) \sum_{X_{0:T},Z_{0:T}} \nabla_\theta P_{\theta,\phi}(X_{0:T}, Z_{0:T}|C) \sum_{t=1}^{T} \log \pi_\theta(Z_t|S_{t-1}, Z_{t-1}, C) + \\
&\qquad \sum_{C} prior(C) \sum_{X_{0:T},Z_{0:T}} P_{\theta,\phi}(X_{0:T}, Z_{0:T}|C) \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(Z_t|S_{t-1}, Z_{t-1}, C)]
\end{aligned} \tag{39}$$

$$\begin{aligned}
&= -\mathop{\mathbb{E}}_{\substack{C,X_{0:T},\\ Z_{0:T}}}\left[\sum_{t=1}^{T}\nabla_{\theta}\log\pi_{\theta}(Z_t|S_{t-1},Z_{t-1},C)\left[\sum_{i=1}^{T}\log\pi_{\theta}(Z_i|S_{i-1},Z_{i-1},C)+1\right]\right] \\
&= -\mathop{\mathbb{E}}_{\substack{C,X_{0:T},\\ Z_{0:T}}}\left[\sum_{t=1}^{T}\nabla_{\theta}\log\pi_{\theta}(Z_t|S_{t-1},Z_{t-1},C)\sum_{i=t}^{T}\log\pi_{\theta}(Z_i|S_{i-1},Z_{i-1},C)\right]
\end{aligned} \tag{40}$$

The third equality holds because we adopt the one-step Markov assumption, i.e., the conditional probability distribution of a random variable depends only on its parent nodes in the probabilistic graphical model (shown as Figure 1). The fourth equality holds out of similar derivation as steps 2-4 in Equation 38. The last equality can be obtained with Equation 46 in the next section, where we prove that any term which is from $\sum_{i=1}^{T}\log\pi_{\theta}(Z_i|S_{i-1},Z_{i-1},C)+1$ and not a function of Z_t will not influence the gradient calculation in Equation 39 and 40.

With similar derivations, we have:

$$\nabla_{\phi}L_1^{DI} = \mathop{\mathbb{E}}_{\substack{C,X_{0:T},\\ Z_{0:T}}}\left[\sum_{t=1}^{T}\nabla_{\phi}\log\pi_{\phi}(A_{t-1}|S_{t-1},Z_t,C)\sum_{i=t}^{T}\log P_{\omega}(Z_i|X_{0:i},Z_{0:i-1},C)\right] \tag{41}$$

$$\nabla_{\phi}L_2^{DI} = -\mathop{\mathbb{E}}_{\substack{C,X_{0:T},\\ Z_{0:T}}}\left[\sum_{t=1}^{T}\nabla_{\phi}\log\pi_{\phi}(A_{t-1}|S_{t-1},Z_t,C)\sum_{i=t}^{T}\log\pi_{\theta}(Z_i|S_{i-1},Z_{i-1},C)\right] \tag{42}$$

As for the gradient with respect to ω, it can be computed with:

$$\nabla_{\omega}L^{DI} = \nabla_{\omega}L_1^{DI} = \sum_{t=1}^{T}\sum_{C}prior(C)\sum_{X_{0:t},Z_{0:t}}P_{\theta,\phi}(X_{0:t},Z_{0:t}|C)\nabla_{\omega}\log P_{\omega}(Z_t|X_{0:t},Z_{0:t-1},C) \tag{43}$$

Still, for each timestep t, it's a standard likelihood maximization problem and can be optimized through SGD.

D.3. Gradients of the Imitation Learning Objective Term

We consider the imitation learning objective term L^{IL}, i.e., the trajectory return shown as:

$$L^{IL} = \sum_{C}prior(C)\sum_{X_{0:T},Z_{0:T}}P_{\theta,\phi}(X_{0:T},Z_{0:T}|C)\sum_{i=0}^{T-1}R_{IL}(S_i,Z_i,Z_{i+1},A_i|C) \tag{44}$$

Following the similar derivation with Equation 36, we can get:

$$\nabla_{\theta}L^{IL} = \mathop{\mathbb{E}}_{\substack{C,X_{0:T},\\ Z_{0:T}}}\left[\sum_{t=1}^{T}\nabla_{\theta}\log\pi_{\theta}(Z_t|S_{t-1},Z_{t-1},C)\sum_{i=0}^{T-1}R_{IL}(S_i,Z_i,Z_{i+1},A_i|C)\right] \tag{45}$$

Further, we note that for each $t\in\{1,\cdots,T\}$, $\forall i<t-1$, we have:

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\substack{C,X_{0:T},\\ Z_{0:T}}}\left[\nabla_{\theta}\log\pi_{\theta}(Z_t|S_{t-1},Z_{t-1},C)R_{IL}(S_i,Z_i,Z_{i+1},A_i|C)\right] \\
&= \sum_{C}prior(C)\sum_{X_{0:T},Z_{0:T}}P_{\theta,\phi}(X_{0:T},Z_{0:T}|C)\nabla_{\theta}\log\pi_{\theta}(Z_t|S_{t-1},Z_{t-1},C)R_{IL}(S_i,Z_i,Z_{i+1},A_i|C) \\
&= \sum_{C}prior(C)\sum_{\substack{X_{0:t-1},\\ Z_{0:t}}}\sum_{\substack{X_{t:T},\\ Z_{t+1:T}}}P_{\theta,\phi}(X_{0:T},Z_{0:T}|C)\nabla_{\theta}\log\pi_{\theta}(Z_t|S_{t-1},Z_{t-1},C)R_{IL}^{i} \\
&= \sum_{C}prior(C)\sum_{\substack{X_{0:t-1},\\ Z_{0:t}}}P_{\theta,\phi}(X_{0:t-1},Z_{0:t}|C)\nabla_{\theta}\log\pi_{\theta}(Z_t|S_{t-1},Z_{t-1},C)R_{IL}^{i}
\end{aligned} \tag{46}$$

$$\begin{aligned}
&= \sum_{C} prior(C) \sum_{\substack{X_{0:t-1},\\ Z_{0:t-1}}} P_{\theta,\phi}(X_{0:t-1}, Z_{0:t-1}|C) \sum_{Z_t} \pi_\theta(Z_t|S_{t-1}, Z_{t-1}, C) \nabla_\theta \log \pi_\theta(Z_t|S_{t-1}, Z_{t-1}, C) R_{IL}^i \\
&= \sum_{C} prior(C) \sum_{\substack{X_{0:t-1},\\ Z_{0:t-1}}} P_{\theta,\phi}(X_{0:t-1}, Z_{0:t-1}|C) R_{IL}^i \sum_{Z_t} \pi_\theta(Z_t|S_{t-1}, Z_{t-1}, C) \nabla_\theta \log \pi_\theta(Z_t|S_{t-1}, Z_{t-1}, C) \\
&= \sum_{C} prior(C) \sum_{\substack{X_{0:t-1},\\ Z_{0:t-1}}} P_{\theta,\phi}(X_{0:t-1}, Z_{0:t-1}|C) R_{IL}^i \sum_{Z_t} \nabla_\theta \pi_\theta(Z_t|S_{t-1}, Z_{t-1}, C) \\
&= \sum_{C} prior(C) \sum_{\substack{X_{0:t-1},\\ Z_{0:t-1}}} P_{\theta,\phi}(X_{0:t-1}, Z_{0:t-1}|C) R_{IL}^i \nabla_\theta \sum_{Z_t} \pi_\theta(Z_t|S_{t-1}, Z_{t-1}, C) \\
&= \sum_{C} prior(C) \sum_{\substack{X_{0:t-1},\\ Z_{0:t-1}}} P_{\theta,\phi}(X_{0:t-1}, Z_{0:t-1}|C) R_{IL}(S_i, Z_i, Z_{i+1}, A_i|C) \nabla_\theta 1 = 0
\end{aligned} \tag{47}$$

where $R_{IL}^i = R_{IL}(S_i, Z_i, Z_{i+1}, A_i|C)$ for simplicity. We use the law of total probability in the third equality, which we also use in the later derivations. The fifth equality holds because $i < t-1$ and $R_{IL}(S_i, Z_i, Z_{i+1}, A_i|C)$ is irrelevant to Z_t. Based on Equation 45 and 46, we have:

$$\nabla_\theta L^{IL} = \mathop{\mathbb{E}}_{\substack{C, X_{0:T},\\ Z_{0:T}}} \left[\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(Z_t|S_{t-1}, Z_{t-1}, C) \sum_{i=t-1}^{T-1} R_{IL}(S_i, Z_i, Z_{i+1}, A_i|C) \right] \tag{48}$$

With similar derivations, we can obtain:

$$\nabla_\phi L^{IL} = \mathop{\mathbb{E}}_{\substack{C, X_{0:T},\\ Z_{0:T}}} \left[\sum_{t=1}^{T} \nabla_\phi \log \pi_\phi(A_{t-1}|S_{t-1}, Z_t, C) \sum_{i=t-1}^{T-1} R_{IL}(S_i, Z_i, Z_{i+1}, A_i|C) \right] \tag{49}$$

D.4. The Overall Unbiased Gradient Estimator

To sum up, the gradients with respect to θ and ϕ can be computed with $\nabla_{\theta,\phi} L = \nabla_{\theta,\phi}(\alpha_1 L^{MI} + \alpha_2 L^{DI} + \alpha_3 L^{IL})$, where $\alpha_{1:3} > 0$ are the weights for each objective term and fine-tuned as hyperparameters. Combining Equation (36, 38, 39, 48) and Equation (37, 41, 42, 49), we have the actor-critic learning framework shown as Equation 11, except for the baseline terms, b^{high} and b^{low}.

Further, we claim that Equation 11 provides unbiased estimation of the gradients with respect to θ and ϕ. We proof this by showing that $\mathbb{E}\left[\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta^t b^{high}(S_{t-1}, Z_{t-1}|C)\right] = \mathbb{E}\left[\sum_{t=1}^{T} \nabla_\phi \log \pi_\phi^t b^{low}(S_{t-1}, Z_t|C)\right] = 0$, as follows:

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\substack{C, X_{0:T},\\ Z_{0:T}}} \left[\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(Z_t|S_{t-1}, Z_{t-1}, C) b^{high}(S_{t-1}, Z_{t-1}|C) \right] \\
&= \sum_{C} prior(C) \sum_{X_{0:T}, Z_{0:T}} P_{\theta,\phi}(X_{0:T}, Z_{0:T}|C) \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(Z_t|S_{t-1}, Z_{t-1}, C) b^{high}(S_{t-1}, Z_{t-1}|C) \\
&= \sum_{C} prior(C) \sum_{t=1}^{T} \sum_{X_{0:T}, Z_{0:T}} P_{\theta,\phi}(X_{0:T}, Z_{0:T}|C) \nabla_\theta \log \pi_\theta(Z_t|S_{t-1}, Z_{t-1}, C) b^{high}(S_{t-1}, Z_{t-1}|C) \\
&= \sum_{C} prior(C) \sum_{t=1}^{T} \sum_{X_{0:t-1}, Z_{0:t}} P_{\theta,\phi}(X_{0:t-1}, Z_{0:t}|C) \nabla_\theta \log \pi_\theta(Z_t|S_{t-1}, Z_{t-1}, C) b^{high}(S_{t-1}, Z_{t-1}|C)
\end{aligned}$$

$$\sum_{X_{0:t-1},Z_{0:t}} P_{\theta,\phi}(X_{0:t-1},Z_{0:t}|C)\nabla_\theta \log\pi_\theta(Z_t|S_{t-1},Z_{t-1},C)b^{high}(S_{t-1},Z_{t-1}|C)$$
$$=\sum_{\substack{X_{0:t-1},\\Z_{0:t-1}}} P_{\theta,\phi}(X_{0:t-1},Z_{0:t-1}|C)\sum_{Z_t}\pi_\theta(Z_t|S_{t-1},Z_{t-1},C)\nabla_\theta \log\pi_\theta(Z_t|S_{t-1},Z_{t-1},C)b^{high}(S_{t-1},Z_{t-1}|C)$$
$$=\sum_{X_{0:t-1},Z_{0:t-1}} P_{\theta,\phi}(X_{0:t-1},Z_{0:t-1}|C)b^{high}(S_{t-1},Z_{t-1}|C)\sum_{Z_t}\nabla_\theta\pi_\theta(Z_t|S_{t-1},Z_{t-1},C)$$
$$=\sum_{X_{0:t-1},Z_{0:t-1}} P_{\theta,\phi}(X_{0:t-1},Z_{0:t-1}|C)b^{high}(S_{t-1},Z_{t-1}|C)\nabla_\theta 1=0$$

$$\mathop{\mathbb{E}}_{\substack{C,X_{0:T},\\Z_{0:T}}}\left[\sum_{t=1}^{T}\nabla_\phi\log\pi_\phi(A_{t-1}|S_{t-1},Z_t,C)b^{low}(S_{t-1},Z_t|C)\right]$$
$$=\sum_C prior(C)\sum_{X_{0:T},Z_{0:T}} P_{\theta,\phi}(X_{0:T},Z_{0:T}|C)\sum_{t=1}^{T}\nabla_\phi\log\pi_\phi(A_{t-1}|S_{t-1},Z_t,C)b^{low}(S_{t-1},Z_t|C)$$
$$=\sum_C prior(C)\sum_{t=1}^{T}\sum_{X_{0:T},Z_{0:T}} P_{\theta,\phi}(X_{0:T},Z_{0:T}|C)\nabla_\phi\log\pi_\phi(A_{t-1}|S_{t-1},Z_t,C)b^{low}(S_{t-1},Z_t|C)$$
$$=\sum_C prior(C)\sum_{t=1}^{T}\sum_{X_{0:t},Z_{0:t}} P_{\theta,\phi}(X_{0:t},Z_{0:t}|C)\nabla_\phi\log\pi_\phi(A_{t-1}|S_{t-1},Z_t,C)b^{low}(S_{t-1},Z_t|C)$$
$$=\sum_C prior(C)\sum_{t=1}^{T}\sum_{X_{0:t-1},Z_{0:t}} P_{\theta,\phi}(X_{0:t-1},Z_{0:t}|C)\sum_{X_t}P_\phi(X_t|X_{0:t-1},Z_{0:t},C)\cdot$$
$$\nabla_\phi\log\pi_\phi(A_{t-1}|S_{t-1},Z_t,C)b^{low}(S_{t-1},Z_t|C)$$
$$=\sum_C prior(C)\sum_{t=1}^{T}\sum_{X_{0:t-1},Z_{0:t}} P_{\theta,\phi}(X_{0:t-1},Z_{0:t}|C)\sum_{A_{t-1}}\pi_\phi(A_{t-1}|S_{t-1},Z_t,C)\cdot$$
$$\nabla_\phi\log\pi_\phi(A_{t-1}|S_{t-1},Z_t,C)b^{low}(S_{t-1},Z_t|C)\sum_{S_t}\mathcal{P}(S_t|S_{t-1},A_{t-1},C)$$
$$=\sum_C prior(C)\sum_{t=1}^{T}\sum_{X_{0:t-1},Z_{0:t}} P_{\theta,\phi}(X_{0:t-1},Z_{0:t}|C)b^{low}(S_{t-1},Z_t|C)\sum_{A_{t-1}}\pi_\phi(A_{t-1}|S_{t-1},Z_t,C)\cdot$$
$$\nabla_\phi\log\pi_\phi(A_{t-1}|S_{t-1},Z_t,C)$$
$$=\sum_C prior(C)\sum_{t=1}^{T}\sum_{X_{0:t-1},Z_{0:t}} P_{\theta,\phi}(X_{0:t-1},Z_{0:t}|C)b^{low}(S_{t-1},Z_t|C)\sum_{A_{t-1}}\nabla_\phi\pi_\phi(A_{t-1}|S_{t-1},Z_t,C)$$

$$=\sum_C prior(C)\sum_{t=1}^{T}\sum_{X_{0:t-1},Z_{0:t}} P_{\theta,\phi}(X_{0:t-1},Z_{0:t}|C)b^{low}(S_{t-1},Z_t|C)\nabla_\phi\sum_{A_{t-1}}\pi_\phi(A_{t-1}|S_{t-1},Z_t,C)$$
$$=\sum_C prior(C)\sum_{t=1}^{T}\sum_{\substack{X_{0:t-1},\\Z_{0:t}}} P_{\theta,\phi}(X_{0:t-1},Z_{0:t}|C)b^{low}(S_{t-1},Z_t|C)\nabla_\phi 1=0$$

Algorithm 1 Multi-task Hierarchical Adversarial Inverse Reinforcement Learning (MH-AIRL)

```
1: Input: Prior distribution of the task variable $prior(C)$, expert demonstrations $\{X^E_{0:T}\}$ (If the task or option annotations, i.e., $\{C_E\}$ or $\{Z^E_{0:T}\}$, are provided, the corresponding estimation in Step 6 is not required.)
2: Initialize the hierarchical policy $\pi_\theta$ and $\pi_\phi$, discriminator $f_\vartheta$, posteriors for the task context $P_\psi$ and option choice $P_\omega$
3: for each training episode do
4:     Generate $M$ trajectories $\{(C, X_{0:T}, Z_{0:T})\}$ by sampling the task $C \sim prior(\cdot)$ and then exploring it with $\pi_\theta$ and $\pi_\phi$
5:     Update $P_\psi$ and $P_\omega$ by minimizing $L^{MI}$ and $L^{DI}$ (Eq. 9) using SGD with $\{(C, X_{0:T}, Z_{0:T})\}$
6:     Estimate the expert global and local latent codes with $P_\psi$ and $P_\omega$, i.e., $C_E \sim P_\psi(\cdot|X^E_{0:T})$, $Z^E_{0:T} \sim P_\omega(\cdot|X^E_{0:T}, C_E)$
7:     Update $f_\vartheta$ by minimizing the cross entropy loss in Eq. 8 based on $\{(C, X_{0:T}, Z_{0:T})\}$ and $\{(C_E, X^E_{0:T}, Z^E_{0:T})\}$
8:     Train $\pi_\theta$ and $\pi_\phi$ by HPPO, i.e., Eq. 11, based on $\{(C, X_{0:T}, Z_{0:T})\}$ and $f_\vartheta$ which defines $D_\vartheta$ and $R_{IL}$
9: end for
```

D.5. Illustrations of Interactions among Networks in MH-AIRL

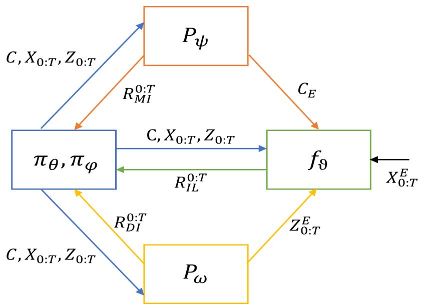
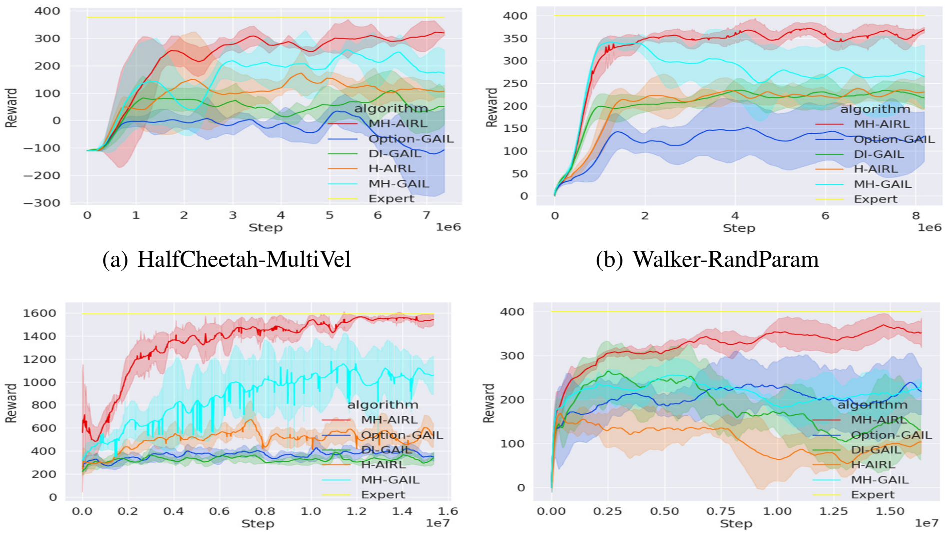


Figure 5. Interactions among the five networks in our learning system.

There are in total five networks to learn in our system: the high-level policy π_θ, low-level policy π_ϕ, discriminator f_ϑ, variational posteriors for the task context P_ψ and option context P_ω. Algorithm 1 shows in details how to coordinate their training process. To be more intuitive, we provide Figure 5 for illustrating the interactions among them. P_ψ and P_ω are trained with the trajectories (i.e., $\{(C, X_{0:T}, Z_{0:T})\}$) generated by the hierarchical policy $\pi_{\theta,\phi}$, and can provide the reward signals $R^{0:T}_{MI}$ and $R^{0:T}_{DI}$ for training $\pi_{\theta,\phi}$, which are defined as $\alpha_1 \log P_\psi(C|X_{0:T})$ and $\alpha_2 \log \frac{P_\omega(Z_i|X^i, Z^{i-1}, C)}{\pi_\theta(Z_i|S_{i-1}, Z_{i-1}, C)}$ $(i \in \{1, \cdots, T\})$ in Equation 11, respectively. On the other hand, the discriminator f_ϑ is trained to distinguish the expert demonstrations $\{(C_E, X^E_{0:T}, Z^E_{0:T})\}$ and generated samples $\{(C, X_{0:T}, Z_{0:T})\}$, where C_E and $\{Z^E_{0:T}\}$ can be estimated from P_ψ and P_ω if not provided. Then, the AIRL reward term $R^{0:T}_{IL}$ can be obtained based on the output of f_ϑ. Last, the hierarchical policy $\pi_{\theta,\phi}$ can be trained by maximizing the return defined with $R^{0:T}_{MI}$, $R^{0:T}_{DI}$, and $R^{0:T}_{IL}$ (i.e., Eq. 11).

E. Appendix on Evaluation Results

E.1. Plots of the Ablation Study



(a) HalfCheetah-MultiVel

(b) Walker-RandParam

(c) Ant-MultiGoal

(d) Kitchen-MultiSeq

Figure 6. Comparison results of MH-AIRL with the ablated versions (MH-GAIL & H-AIRL) and SOTA Hierarchical Imitation Learning (HIL) baselines (Option-GAIL & DI-GAIL) on the four evaluation tasks. Our algorithm outperforms the baselines in all the tasks, especially in the more challenging ones (Ant & Kitchen). MH-GAIL performs better than the other baselines which do not contain the multi-task learning component. H-AIRL, an ablation and HIL algorithm, has better performance than other SOTA HIL baselines on the Mujoco tasks.

E.2. Implementation Details of MH-GAIL

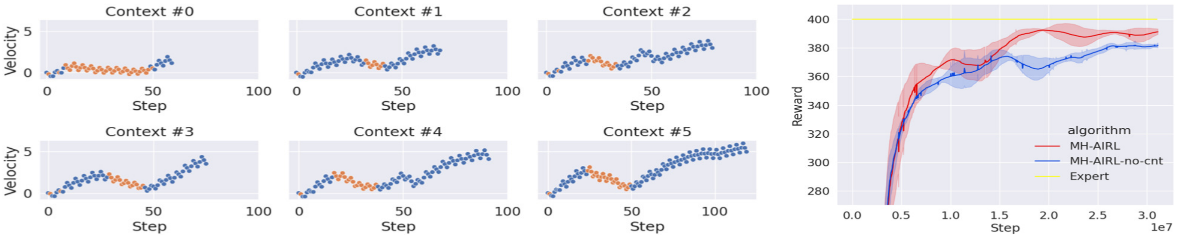
MH-GAIL is a variant of our algorithm by replacing the AIRL component with GAIL. Similar with Section 4.2, we need to provide an extension of GAIL with the one-step option model, in order to learn a hierarchical policy. The extension method follows Option-GAIL (Jing et al., 2021) which is one of our baselines. MH-GAIL also uses an adversarial learning framework that contains a discriminator D_ϑ and a hierarchical policy $\pi_{\theta,\phi}$, for which the objectives are as follows:

$$\begin{aligned} &\max_{\vartheta} \mathbb{E}_{C\sim prior(\cdot),(S,A,Z,Z')\sim\pi_E(\cdot|C)}\left[\log(1-D_\vartheta(S,A,Z,Z'|C))\right]+ \\ &\qquad \mathbb{E}_{C\sim prior(\cdot),(S,A,Z,Z')\sim\pi_{\theta,\phi}(\cdot|C)}\left[\log D_\vartheta(S,A,Z,Z'|C)\right] \\ &\max_{\theta,\phi} L^{IL} = \max_{\theta,\phi} \mathbb{E}_{C\sim prior(\cdot),(X_{0:T},Z_{0:T})\sim\pi_{\theta,\phi}(\cdot|C)} \sum_{t=0}^{T-1} R_{IL}^t,\ R_{IL}^t = -\log D_\vartheta(S_t,A_t,Z_{t+1},Z_t|C) \end{aligned} \tag{50}$$

where (S,A,Z,Z') denotes (S_t,A_t,Z_{t+1},Z_t), $t=\{0,\cdots,T-1\}$. It can be observed that the definition of R_{IL}^t have changed. Moreover, the discriminator D_ϑ in MH-GAIL is trained as a binary classifier to distinguish the expert demonstrations (labeled as 0) and generated samples (labeled as 1), and does not have a specially-designed structure like the discriminator D_ϑ in MH-AIRL, which is defined with f_ϑ and $\pi_{\theta,\phi}$, so that it cannot recover the expert reward function.

E.3. Analysis of the Learned Hierarchical Policy on HalfCheetah-MultiVel and Walker-RandParam

First, we randomly select 6 task contexts for HalfCheetah-MultiVel and visualize the recovered hierarchical policy as the velocity change of each episode in Figure 7(a). It can be observed that the agent automatically discovers two options (Option 1: blue, Option 2: orange) and adopts Option 1 for the acceleration phase ($0 \rightarrow v/2$ or $0 \rightarrow v$) and Option 2 for the deceleration phase ($v/2 \rightarrow 0$). This shows that MH-AIRL can capture the compositional structure within the tasks very well and transfer the learned basic skills to boost multi-task policy learning.



(a) Results on HalfCheetah-MultiVel

(b) Results on Walker-RandParam

Figure 7. (a) Velocity change of the HalfCheetah agent in the test tasks with different goal velocities, where the agent adopts Option 1 (blue) and 2 (orange) when increasing and decreasing the speed, respectively. (b) For Walker-RandParam, the basic skills must adapt to the task setting, so the learning performance would drop without conditioning the low-level policy (i.e., option) on the task context.

Second, we note that, for some circumstances, the basic skills need to be conditioned on the task context. For the Mujoco-MultiGoal/MultiVel tasks, the basic skills (e.g., Option 2: decreasing the velocity) can be directly transferred among the tasks in the class and the agent only needs to adjust its high-level policy according to the task variable (e.g., adopting Option 2 when achieving $v/2$). However, for tasks like Walker-RandParam, the skills need to adapt to the tasks, since the mass of the agent changes and so do the control dynamics. As shown in Figure 7(b), the learning performance would drop without conditioning the low-level policy (i.e., option) on the task context, i.e., MH-AIRL-no-cnt.