Learning Compound Tasks without Task-specific Knowledge via Imitation and Self-supervised Learning

Sang-Hyun Lee [1 2] **Seung-Woo Seo** [2]

Abstract

Most real-world tasks are compound tasks that consist of multiple simpler sub-tasks. The main challenge of learning compound tasks is that we have no explicit supervision to learn the hierarchical structure of compound tasks. To address this challenge, previous imitation learning methods exploit task-specific knowledge, e.g., labeling demonstrations manually or specifying termination conditions for each sub-task. However, the need for task-specific knowledge makes it difficult to scale imitation learning to real-world tasks. In this paper, we propose an imitation learning method that can learn compound tasks without task-specific knowledge. The key idea behind our method is to leverage a self-supervised learning framework to learn the hierarchical structure of compound tasks. Our work also proposes a task-agnostic regularization technique to prevent unstable switching between sub-tasks, which has been a common degenerate case in previous works. We evaluate our method against several baselines on compound tasks. The results show that our method achieves state-of-the-art performance on compound tasks, outperforming prior imitation learning methods.

1. Introduction

Learning from demonstration (LfD) has been widely researched for decades, which has led to remarkable achievements (Ziebart et al., 2008; Wulfmeier et al., 2016; Finn et al., 2016; Ho & Ermon, 2016). However, we still cannot apply existing LfD algorithms to learn real-world tasks. One of the main reasons is that most real-world tasks are compound tasks that consist of a set of simpler sub-tasks.

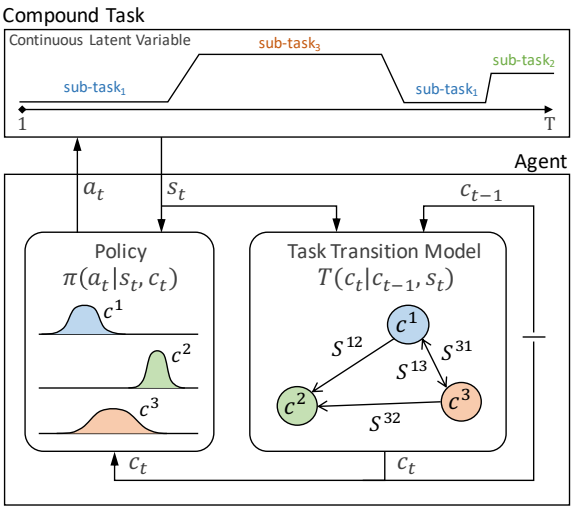


Figure 1. Overview of proposed method. Our work introduces the task transition model, which describes the hierarchical structure of compound tasks. The policies for each sub-task take the current sub-task as additional input where sub-tasks are encoded as continuous latent variables. S^{ij} represents a set of states relevant to the transition from sub-task c^i to sub-task c^j.

Compound tasks require an agent to learn their hierarchical structure and policies for each sub-task, but standard LfD algorithms cannot infer their hierarchical structure. To understand the hierarchical structure of compound tasks, we should identify sub-tasks and learn the relationships between them.

The main challenge of learning compound tasks is that we have no explicit supervisory signals to learn their hierarchical structure. Previous works have tried to overcome the challenge by leveraging task-specific knowledge in a variety of ways. One traditional approach is to manually label or segment demonstrations into sub-tasks (Manschitz et al., 2015; Li et al., 2017). However, this approach requires laborious and time-consuming processes, as raw demonstrations are unlabeled and unsegmented. Although several recent works have introduced approaches that allow compound tasks to be learned with weak supervision, e.g., desired sub-task orders or the total number of sub-tasks (Fox et al., 2017;

[1]ThorDrive, Seoul, South Korea [2]Department of Electrical and Computer Engineering, Seoul National University, Seoul, South Korea. Correspondence to: Sang-Hyun Lee <slee01@snu.ac.kr>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

Shiarlis et al., 2018; Sharma et al., 2018), they still need to exploit task-specific knowledge to learn compound tasks. Unfortunately, the need for task-specific knowledge limits the applicability of LfD approaches to real-world tasks.

In this paper, we propose an LfD method that allows compound tasks to be learned without task-specific knowledge. Our method introduces a model that represents the hierarchical structure of compound tasks. This model takes as input the previous sub-task and the current state and returns the current sub-task, where sub-tasks are encoded as continuous latent variables. We call this model the task transition model because it is similar to the state transition model in the Markov decision process (MDP). An agent in our framework interacts with the environments via the task transition model, where sub-policies are designed to take the current sub-task that is the output of the task transition model as an additional input. Figure 1 shows an overview of the proposed method.

To train the task transition model, our method leverages self-supervised learning in which embedded metadata are autonomously extracted from training inputs and used as supervision. In our case, we extract sub-task labels from unsegmented demonstrations and then use them as supervisory signals to train the task transition model. This approach allows us to learn the hierarchical structure of compound tasks without supervisory signals, which have been generated based on task-specific knowledge in previous works. Our method also introduces a task-agnostic regularization technique to prevent unstable sub-task transitions, whereas previous works address the degenerate case with task-specific knowledge, such as predefined transition conditions between sub-tasks (Le et al., 2018). The concept behind the regularization technique is the information bottleneck, which was first proposed in Tishby et al. (2000) to extract informative representation from an original input.

Our work presented here belongs to the class of imitation learning (IL), one of the main approaches for LfD. Specifically, the propsoed method is based on generative adversarial imitation learning (GAIL) (Ho & Ermon, 2016), which models LfD as an adversarial learning framework. While GAIL focuses on primitive tasks, we extend GAIL with the task transition model to learn compound tasks. The main contribution of our work is that we propose a novel IL method that can learn sub-policies and the hierarchical structure of compound tasks. To the best of our knowledge, this is the first IL method that can learn compound tasks without task-specific knowledge.

We conduct experiments on compound tasks from the OpenAI Gym benchmark suites (Brockman et al., 2016). In addition to these provided tasks, we also introduce new compound tasks — MountainToyCar and MountainToyCarContinuous — that are variants of classic control tasks from the OpenAI Gym. Although the tasks seem straightforward and simple, they require an agent to learn the hierarchical structure of both tasks and policies for each sub-task, which remains a major challenge for traditional LfD algorithms. Our results demonstrate that the proposed method leads to state-of-the-art performance on compound tasks, outperforming prior methods.

2. Related Work

In addition to IL, there are two other approaches for LfD: behavior cloning (BC) and inverse reinforcement learning (IRL). BC seeks to learn a policy from demonstrations using supervised learning. Although BC is the simplest approach of LfD, it requires a relatively large amount of demonstrations compared to other approaches. IRL finds the reward function that can explain demonstrations and learns the optimal policy from that reward function with reinforcement learning (RL). This contrasts with IL that directly learns the optimal policy from demonstrations without recovering a reward function. Although these all have achieved a wide range of success in challenging problems (Abbeel & Ng, 2004; Finn et al., 2016; Ho & Ermon, 2016) and have led to further research in various directions (Duan et al., 2017; Peng et al., 2018a), most prior works only focus on primitive tasks consisting of a single task without a hierarchical structure.

Several previous works have attempted to learn compound tasks with additional task-specific information (Meier et al., 2011; Yang et al., 2015; Xu et al., 2018; Yu et al., 2018). Manschitz et al. (2015) used segmented and labeled demonstrations to infer the transitions between consecutive movement primitives. Shiarlis et al. (2018) introduced an algorithm that can learn compound tasks with task sketches that explain the desired sequences of sub-tasks. Recently, Kipf et al. (2019) and Sharma et al. (2018) proposed algorithms that allow compound tasks to be learned without specifying sub-tasks in advance. However, the method proposed in Kipf et al. (2019) is built upon BC, which is inadequate for high-dimensional tasks due to compounding errors caused by covariate shift (Ross & Bagnell, 2010; Ross et al., 2011). The approach introduced in Sharma et al. (2018) can learn high-dimensional compound tasks but assumes access to a pre-trained model and information about the total number of sub-tasks to infer the hierarchical structure of compound tasks. Unlike these prior works, our method does not use task-specific knowledge or pre-trained models to learn compound tasks.

Information-theoretic concepts are widely used in the fields of RL and LfD. Mohamed and Rezende (2015) introduced an intrinsically motivated RL algorithm that trains an agent to take the actions that yield the highest intrinsic reward, where intrinsic rewards are defined as mutual information

between states and actions. Eysenbach et al. (2018) introduced DIAYN, which can learn diverse skills without a reward function by maximizing the mutual information between states and skills. Peng et al. (2018b) introduced a regularization technique for adversarial learning, referred to as variational discriminator bottleneck (VDB), which enforces a constraint on mutual information between input observations and a discriminator's internal representation. In our work, we apply a constraint on mutual information, conditioned on the previous sub-task, between the current sub-task and the current state to prevent unstable sub-task transitions.

3. Preliminaries

3.1. Markov Decision Process

The Markov Decision Process (MDP), which is defined by the tuple $(S, A, P, R, \rho_0, \gamma, T)$, is a framework for sequential decision-making problems. Here, S is the set of states, A is the set of actions, $P : S \times A \times S \to \mathbb{R}_+$ is the state transition model, $R : S \times A \to \mathbb{R}$ is a reward function, $\rho_0 : S \to \mathbb{R}_+$ is the initial state distribution, γ is the discount factor, and T is the horizon. An agent's behavior is defined by the policy $\pi : S \to P(A)$, which maps states to a probability distribution over actions. The core problem of the MDP is to find the optimal policy π^* that maximizes the expected cumulative rewards. RL is one of the approaches to solve the core problem when we do not know the model of environments.

3.2. Imitation Learning

The goal of IL is to learn the optimal policy based on expert demonstrations rather than the reward function. The framework is very useful in the fields where the reward function cannot be easily defined. Ho and Ermon (2016) proposed a novel IL method called GAIL, which is inspired by generative adversarial networks (GANs) (Goodfellow et al., 2014). GAIL consists of two networks, policy π and discriminator D, which are trained by adversarial learning. The policy aims to confuse the discriminator by generating expert-like actions. In contrast, the discriminator's goal is to distinguish between the behavior of experts and the agent being trained. Note that the discriminator can be used as a reward function for training the policy. The GAIL objective function is as follows:

$$\min_{\pi} \max_{D} \mathbb{E}_{\pi}[\log D(s,a)] + \mathbb{E}_{\pi_E}[\log(1 - D(s,a))] - \lambda H(\pi) \tag{1}$$

where π_E is the expert policy and $H(\pi) \triangleq \mathbb{E}_{\pi}[-\log \pi(a|s)]$ is the entropy of the policy being trained.

Although GAIL obtains huge performance gains over existing LfD methods on standard control benchmark tasks, it cannot consider the variations underlying the demonstrations. To handle these variations, Li et al. (2017) proposed InfoGAIL by extending GAIL with the concept of InfoGAN (Chen et al., 2016). They design the policy $\pi(a|s, c)$ to take factors of variations as additional inputs, where factors are encoded as discrete latent variables. In this setting, they maximize the mutual information $I(c\,; \tau_c)$ between the latent variables c and the generated trajectories $\tau_c \sim \pi(a|s, c)$ to identify the salient factors of variations. The objective function of InfoGAIL is as follows:

$$\min_{\pi, Q} \max_{D} \mathbb{E}_{\pi}[\log D(s,a)] + \mathbb{E}_{\pi_E}[\log(1 - D(s,a))] - \lambda_1 L_I(\pi, Q) - \lambda_2 H(\pi) \tag{2}$$

where $Q(c|s, a)$ is the approximation of the true posterior $p(c|s, a)$ and $L_I(\pi,Q) = \mathbb{E}_{c\sim p(c), a\sim\pi(\cdot|s,c)}[\log Q(c|s,a)] + H(c)$ is the variational lower bound of the mutual information $I(c\,; \tau)$. Note that InfoGAIL requires that demonstrations be segmented into salient factors in advance. Unlike this work, the goal of our method is to learn compound tasks from unsegmented demonstrations.

3.3. Self-supervised Learning

Self-supervised learning is a promising paradigm that allows us to learn without explicit supervision, such as hand-labeled data or extrinsic rewards. In this approach, informative metadata are extracted from training inputs and used as supervisory signals. There are many different forms of extracted metadata. Doersch et al. (2015) sampled pairs of patches randomly in the same image and used them as supervisory signals for visual representation learning. Pathak et al. (2017) defined the intrinsic rewards proportional to the prediction error of the state transition model, which encourages an agent to explore new states. This method enables an agent to be trained without extrinsic rewards, unlike traditional RL approaches. In our work, we leverage self-supervised learning to learn the hierarchical structure of compound tasks without task-specific information.

4. Proposed Method

Compound tasks require that an agent understand their hierarchical structure and infer policies for each sub-task. The central challenge of learning compound tasks is that we have no explicit supervision from which to learn the hierarchical structure of compound tasks. Although several recent works have proposed LfD methods that can learn compound tasks, their methods address the above challenge with various forms of task-specific information (Li et al., 2017; Xu et al., 2018; Sharma et al., 2018; Yu et al., 2018; Le et al., 2018). The requirement of task-specific knowledge makes it difficult to scale IL to real-world tasks. Here, we introduce an IL method that can learn compound tasks without task-specific knowledge.

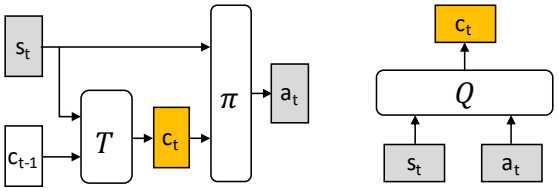


Figure 2. **Left:** Structure of task transition model and hierarchical policy. The agent in s_t interacts with the environments using its policy $\pi(a_t|s_t, c_t)$ and task transition model $T(c_t|c_{t-1}, s_t)$. **Right:** Structure of posterior. The posterior infers sub-tasks c_t from corresponding state-action pairs (s_t, a_t).

4.1. Problem Setting and Overview

We focus on the problem of learning compound tasks from demonstrations without task-specific knowledge. We assume that demonstrations $\{\tau_1, \tau_2, \ldots, \tau_N\}$ are neither labeled nor segmented into sub-tasks. Each demonstration consists of variable length of state–action pairs $\{(s_1, a_1), (s_2, a_2), \ldots, (s_T, a_T)\}$, and each state–action pair (s_t, a_t) has a corresponding sub-task c_t that is encoded as a latent variable. Most previous works have encoded sub-tasks as discrete latent variables while specifying the total number of sub-tasks in advance. However, sub-tasks in our work are encoded as continuous latent variables to avoid restricting the number of sub-tasks and stably represent the process of sub-task transitions in latent space. We believe that using continuous latent variables contributes to performance improvements in our work.

Our framework introduces a model that describes the hierarchical structure of compound tasks. The model takes as input the previous sub-task and the current state and returns the current sub-task. We call this model the task transition model $T(c_t|c_{t-1}, s_t)$ because it is similar to the state transition model $P(s_{t+1}|s_t, a_t)$ in the MDP. This model's structure comes from the idea that states include features that determine the relationship between sub-tasks. As shown in Figure 2, sub-policies $\pi(a_t|s_t, c_t)$ are designed to take the current sub-task as an additional input, which allows an agent to execute the optimal action for the current sub-task.

We learn compound tasks by alternating between two phases: 1) identifying sub-tasks and learning their sub-policies simultaneously and 2) modeling the relationship between the identified sub-tasks. These two phases are dependent on each other. In the remainder of this section, we describe how our method accomplishes each phase in detail.

4.2. Sub-task Identification and Sub-policy Learning

In order to identify sub-tasks, we maximize the mutual information $I(c_t; s_t, a_t)$ between sub-tasks c_t and corresponding state-action pairs (s_t, a_t) sampled from the policy $\pi(a_t|s_t, c_t)$ being trained. Maximizing mutual information

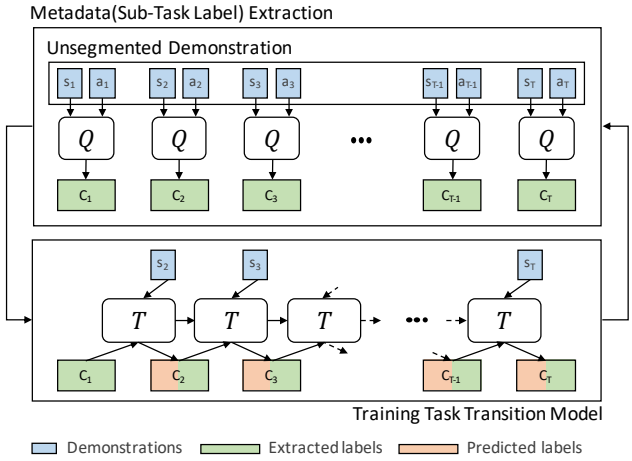


Figure 3. Overview of training procedure for task transition model. Note that the task transition model is represented with recurrent models, since it should learn long-term dependencies between sub-tasks.

encourages sub-tasks to determine what actions the agent chooses. In other words, it allows us to separate distinct types of policies in an unsupervised manner. The concept of maximizing mutual information can be instantiated by deriving the lower bound of the mutual information (Chen et al., 2016). The derived lower bound is estimated with samples drawn from the prior distribution $P(c_t)$ instead of the true posterior distribution $P(c_t|s_t, a_t)$ on sub-tasks.

Several previous works assume that the prior $P(c_t)$ is known before training (Li et al., 2017; Sharma et al., 2018), but the assumption requires pre-training step or task-specific knowledge, such as the total number of sub-tasks. To avoid this requirement, our work uses the task transition model as importance sampling distribution for estimating the lower bound of the mutual information $L_I(\pi, T, Q)$. This allows us to estimate the lower bound without task-specific knowledge because our method trains the task transition model to generate useful samples to estimate the lower bound. Mathematically, we can write the lower bound of the mutual information as follows:

$$
\begin{aligned}
&I(c_t\,; s_t, a_t) \\
&\geq \mathbb{E}_{P(c_t)}[\mathbb{E}_{\pi(a_t|s_t,c_t)}[\log Q(c_t|s_t, a_t)]] + H(c_t) \\
&= \mathbb{E}_{T(c_t|c_{t-1},s_t)}[\omega_t\, \mathbb{E}_{\pi(a_t|s_t,c_t)}[\log Q(c_t|s_t, a_t)]] + H(c_t) \\
&= L_I(\pi, T, Q) \qquad (3)
\end{aligned}
$$

where $Q(c_t|s_t, a_t)$ is the approximation of the true posterior $P(c_t|s_t, a_t)$ and ω_t is an importance weight $\frac{P(c_t)}{T(c_t|c_{t-1},s_t)}$. The detailed derivation is given in Appendix B.

To obtain sub-policies that can imitate distinct behaviors of demonstrations, we just add the above regularization term

to the GAIL objective in Equation (1), similar to Li et al. (2017). The modified objective can then be written as

$$\min_{\pi,Q}\max_{D}\mathbb{E}_{\pi,T}[\log D(s_t,a_t)] + \mathbb{E}_{\pi_E}[\log(1-D(s_t,a_t))] - \lambda_1 L_I(\pi,T,Q) - \lambda_2 H(\pi) \tag{4}$$

where λ_1 is the hyperparameter for the regularization term and λ_2 is the hyperparameter for the entropy term. Note that the task transition model is fixed when we update the policy, discriminator, and posterior with this objective.

4.3. Learning Relationship between Sub-tasks with Self-supervised Learning

This section describes how we learn the relationship between sub-tasks without task-specific knowledge while preventing unstable sub-task transitions. The key idea behind our approach is to leverage self-supervised learning to train the task transition model, which represents the hierarchical structure of compound tasks. In particular, we use the posterior $Q(c_t|s_t,a_t)$ to extract sub-task labels $\{c_1,c_2,\dots,c_T\}_i$ from sampled demonstrations $\{s_1,a_1,s_2,a_2,\dots,s_T,a_T\}_i$ and then use the extracted sequences of sub-task labels as supervisory signals in the training process for the task transition model. We hypothesize that demonstrations can provide us with diverse ways to compose sub-tasks. The objective for the task transition model can then be written as

$$\min_{T}\mathbb{E}_{Q(c_t|s_t,a_t)}[-\log(T(c_t|c_{t-1},s_t))] \tag{5}$$

where s and a are from sampled demonstrations. Figure 3 shows how our work trains the task transition model in a self-supervised manner.

When we learn the relationship between sub-tasks, one of the common degenerate cases that can cause unstable sub-task transitions is high-frequency switching between sub-tasks (Fox et al., 2017; Krishnan et al., 2017). It often happens near sub-task boundaries and causes sub-optimal behavior of an agent. Several prior works have exploited task-specific knowledge to tackle the issue, e.g., assuming access to desired sequences of sub-tasks or using a specific prior distribution for segment boundaries (Shiarlis et al., 2018; Kipf et al., 2019). Unlike previous works, we want to address the issue without task-specific knowledge.

Our method introduces a task-agnostic regularization technique to tackle the degenerate case without task-specific knowledge. Since the degenerate case can occur in our settings when the task transition model is overfitted or susceptible to features that are irrelevant to sub-task transitions, we need to make the task transition model robust to irrelevant features. To this end, the introduced regularization technique applies a constraint on mutual information $I(c_t\,;s_t|c_{t-1})$, conditioned on the previous sub-task, between the current sub-task and the current state. This enables the task transition model to focus on relevant features represented in the current state. The concept of a constraint on mutual information between related variables, known as information bottleneck, was first proposed in Tishby et al. (2000). Following previous works that utilize this concept (Alemi et al., 2016; Achille & Soatto, 2018), we instantiate our regularization technique with the variational upper bound on the conditional mutual information $I(c_t\,;s_t|c_{t-1})$ derived as follows:

$$\begin{aligned} I(c_t\,;s_t|c_{t-1}) &= \mathbb{E}_{p(s_t,c_{t-1})}[D_{KL}[T(c_t|c_{t-1},s_t)\|r(c_t)]] \\ &\quad - \underbrace{\mathbb{E}_{p(c_{t-1})}[D_{KL}[p(c_t|c_{t-1})\|r(c_t)]]}_{\geq 0} \\ &\leq \mathbb{E}_{p(s_t,c_{t-1})}[D_{KL}[T(c_t|c_{t-1},s_t)\|r(c_t)]] \end{aligned}$$

where $r(c_t)$ is an approximation of the marginal distribution $p(c_t|c_{t-1}) = \int T(c_t|c_{t-1},s_t)p(s_t)\,ds$. We modeled $r(c_t)$ as a normal distribution in our experiments. Appendix B describes the derivation for the variational upper bound in detail.

Consequently, we train the task transition model with the following regularized objective function:

$$\begin{aligned} &\min_{T}\mathbb{E}_{Q(c_t|s_t,a_t)}[-\log(T(c_t|c_{t-1},s_t))] \\ &\text{s.t.}\quad \mathbb{E}_{p(s_t,c_{t-1})}[D_{KL}[T(c_t|c_{t-1},s_t)\|r(c_t)] \leq I_c \end{aligned} \tag{6}$$

where I_c is the upper bound of the conditional mutual information $I(s_t\,;c_t|c_{t-1})$. The constrained optimization problem can be solved with a dual gradient descent (Boyd & Vandenberghe, 2004), where we alternate between minimizing the Lagrangian with respect to the task transition model and adjusting the dual variable β. The Lagrangian of the objective function is obtained by subsuming the constraint into the objective with the dual variable β as follows:

$$\begin{aligned} L(T,\beta) &= \mathbb{E}_{Q(c_t|s_t,a_t)}[-\log(T(c_t|c_{t-1},s_t))] \\ &\quad + \beta\left(\mathbb{E}_{p(s_t,c_{t-1})}[D_{KL}[T(c_t|c_{t-1},s_t)\|r(c_t)] - I_c\right), \end{aligned}$$

and the gradient of the Lagrangian dual function in terms of the dual variable β is

$$\mathbb{E}_{p(s_t,c_{t-1})}[D_{KL}[T(c_t|c_{t-1},s_t)\|r(c_t)] - I_c.$$

Thus, we can alternatively optimize the task transition model and adapt the dual variable according to

$$\begin{aligned} T &\leftarrow \arg\min_{T} L(T,\beta) \\ \beta &\leftarrow \beta + \alpha_\beta\big(\mathbb{E}_{p(s_t,c_{t-1})}[D_{KL}[T(c_t|c_{t-1},s_t)\|r(c_t)] - I_c\big), \end{aligned} \tag{7}$$

where α_β is the step size for the dual variable. In practice, we update the task transition model with a few gradient steps for each iteration. Appendix A provides further details of our training procedure.

	MountainToyCar		MountainToyCarContinuous		FetchPickAndPlace	
Method	Avg. Return	Success Rate	Avg. Return	Success Rate	Avg. Return	Success Rate
GAIL	-200.0 ± 0.0	0.00	-2.72 ± 1.33	0.00	-6.33 ± 6.14	0.77
BC	-200.0 ± 0.0	0.00	-14.34 ± 4.94	0.00	-10.56 ± 6.40	0.29
Ours	**-151.43 ± 4.28**	**1.0**	**91.94 ± 1.78**	**1.0**	**-3.13 ± 1.68**	**0.98**
CVAE	-200.0 ± 0.0	0.00	-32.65 ± 0.83	0.00	-11.07 ± 6.01	0.24
CVAE+T2M	-149.48 ± 35.34	0.87	90.48 ± 9.70	0.98	-10.53 ± 7.37	0.31

Table 1. Average returns and success rates computed over 100 episodes for each method. Although MountainToyCar and MountainToyCarContinuous are relatively simple compound tasks, GAIL and BC perform poorly for both tasks because they cannot infer their hierarchical structure. CVAE+T2M achieves results comparable with those of our method for both tasks but fails to learn FetchPickAndPlace, which is a complex and high-dimensional compound task. Our method outperforms the baselines across all the compound tasks. The results indicate that our work is more scalable and shows better performance than the baselines for compound tasks.

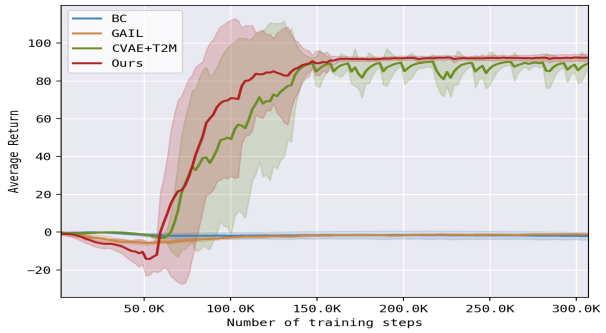


Figure 4. Learning curves for MountainToyCarContinuous. The darker-colored lines and shaded areas denote the average returns and standard deviations, respectively, computed over 10 random seeds.

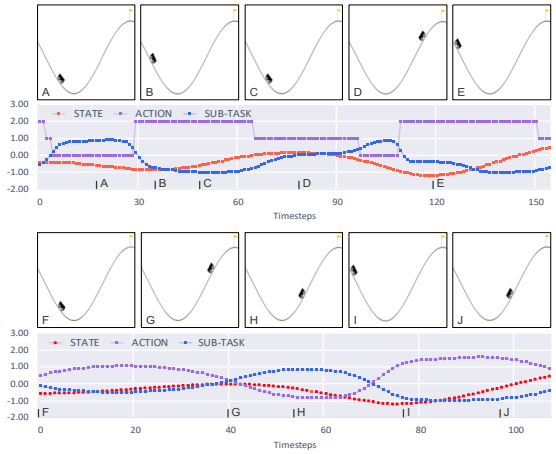


Figure 5. **Top:** Training results for MountainToyCar. The task has discrete action space: pushing left (0), no pushing (1), and pushing right (2). **Bottom:** Training results for MountainToyCarContinuous. The task has continuous action space: pushing left (negative value) and pushing right (positive value).

5. Experiments

Our experiments is designed to answer the following questions: 1) Can our method learn the hierarchical structure of compound tasks without task-specific knowledge? 2) Can our method learn policies for each sub-task without task-specific knowledge? 3) Can our method outperform previous state-of-the-art methods on compound tasks? 4) Can our method achieve performance comparable with that of prior works on primitive tasks? The reason we should answer the last question is that if our method does not work well on primitive tasks, we should investigate whether the target task types are primitive or compound before training. We want to demonstrate that our method can be applied without identifying the task type, which is sometimes not straightforward in practice.

To answer the above questions, we compare our method with several baselines: BC, GAIL, and conditional variational autoencoder (CVAE). BC and GAIL are standard LfD algorithms that cannot learn the hierarchical structure of compound tasks. Comparing our work with these methods shows how important understanding the hierarchical structure is when solving compound tasks. CVAE is a typical approach that can identify sub-tasks and learn their sub-policies. Since CVAE cannot learn the relationship between sub-tasks, we compare our method against CVAE augmented with the task transition model, which has the same architecture as ours. This shows how efficient our method is for learning compound tasks.

Here we briefly describe our experimental setup. We evaluate our method on both primitive and compound tasks. In order to generate demonstrations for each task, we trained expert agents with recent RL algorithms such as ACKTR (Wu et al., 2017) and TRPO (Schulman et al., 2015) in dense reward settings. The demonstrations were neither labeled nor segmented into sub-tasks. We encoded sub-tasks as one-dimensional continuous latent variables, and the uniform distribution was used to sample the initial latent variable for each episode. We would like to emphasize that no task-specific knowledge was used in our experiments. All of the experiments were performed on a PC with a 3.60 GHz Intel Core i7-9700K Processor, and a GeForce RTX 2080 Ti GPU. Appendix C contains further details on our experimental setup. In the remainder of this section, we discuss the results for each task. Note that we denote the task transition model as T2M in graphs and tables for brevity.

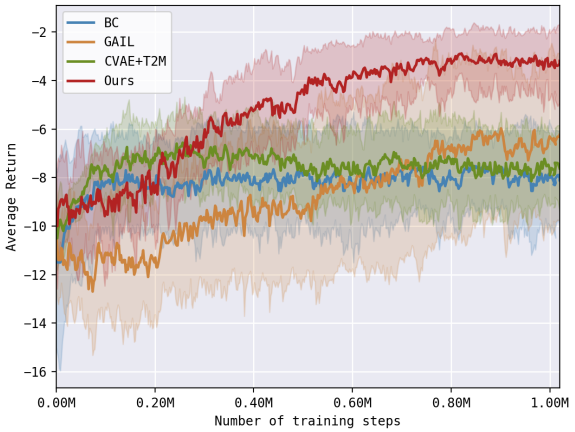


Figure 6. Learning curves for FetchPickAndPlace-v1. The darker-colored lines and shaded areas denote the average returns and standard deviations, respectively, computed over 10 random seeds.

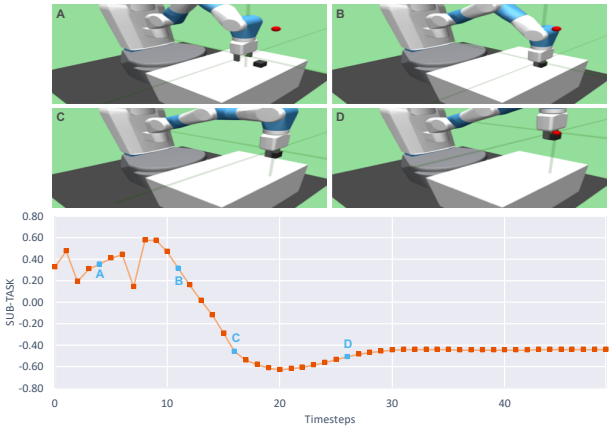


Figure 7. Sub-task transitions in FetchPickAndPlace-v1. Our work segments the episode into two sub-tasks: picking up a box from a table and moving that box to the target point. The first segment is encoded as latent variables ranging from 0.2 to 0.6, and the second segment is encoded as latent variables ranging from -0.6 to -0.4.

5.1. Compound Tasks

Classic Control Tasks: We introduce two low-dimensional compound tasks called MountainToyCar and MountainToyCarContinuous. These tasks are variants of MountainCar and MountainCarContinuous, which are classic control tasks provided by OpenAI Gym (Brockman et al., 2016). The goal of the origin tasks is to reach the target at the top of a hill on the right side. To achieve this goal, an agent should utilize momentum by alternating driving up the slope on either side. The introduced tasks have the same goal as the origin tasks but define the components of states differently. In contrast to the origin tasks where states include the position and velocity of an agent, states in the new tasks only consist of position. The modification requires that an agent infer the current sub-task and take different actions depending on the sub-task even in the same state, which poses a significant challenge to traditional LfD algorithms.

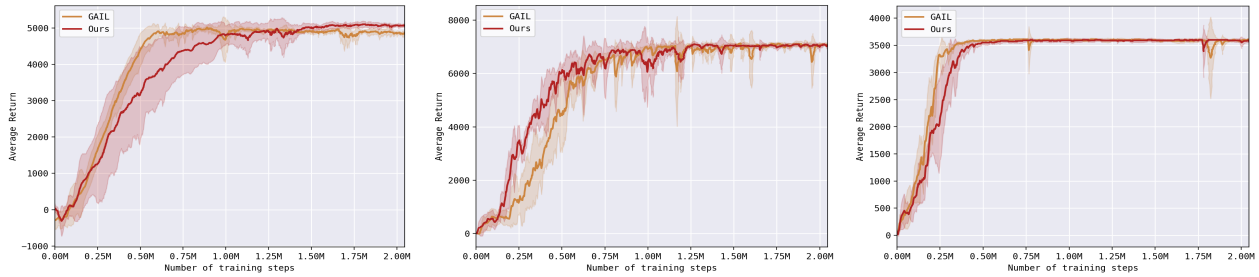
We show the learning curves for MountainToyCarContinuous in Figure 4 and summarize the numerical training results for MountainToyCar and MountainToyCarContinuous in Table 1. BC and GAIL fail to learn both tasks because they cannot infer their hierarchical structure. CVAE without the task transition model also cannot learn both tasks, where sub-task variables are randomly sampled at each time step. In contrast, our work and CVAE augmented with the task transition model successfully learn both tasks with high average returns. The results demonstrate that learning the hierarchical structure of compound tasks is critical for solving compound tasks. The plots in Figure 5 show the training results in an episode for each task. As shown in the snapshots A, C, H, and J, our method segments the left and right actions on each hill into different sub-tasks, which enables our agent to choose different actions even in the same state. In addition, we can also observe that our method composes the identified sub-tasks differently depending on the initial state for each episode to leverage momentum to reach the target point. These results indicate that our work allows the agent to learn the hierarchical structure of both tasks and their sup-policies.

Robotics Task: We evaluate our method on FetchPickAndPlace-v1, which is a complex and high-dimensional compound task from OpenAI Gym Robotics (Plappert et al., 2018). In this task, an agent has to pick up a box and then move it to a target point using its gripper. The box and target point are randomly generated for each episode. The task success is originally determined by whether an agent moves the box to within 0.05 of the target point, but we modified the threshold distance to 0.1 because we empirically observed that the original threshold was too strong to compare our method with the baselines. The most difficult part of the compound task is deciding when to close the gripper to pick up the box without knowing whether the box has been grabbed. We expect that an agent trained with our method can handle that tricky part by understanding the task's hierarchical structure.

Learning curves for FetchPickAndPlace-v1 are shown in Figure 6. We initialized the policy networks in our method and GAIL with pre-trained models through CVAE and BC, respectively. Our method significantly outperforms GAIL that achieves the best result among the baselines. This implies that understanding the hierarchical structure is important to achieve a performance gain for complex compound tasks. Although CVAE augmented with the task transition model can learn the hierarchical structure of com-

METHOD	HALFCHEETAH	WALKER2D	HOPPER
GAIL	**4872.84 ± 82.42**	**7031.15 ± 64.46**	**3597.89 ± 6.43**
BC	3664.40 ± 1730.83	4241.33 ± 2901.16	2233.27 ± 260.54
OURS	**5063.62 ± 76.39**	**7045.08 ± 54.44**	**3588.02 ± 4.50**
CVAE+T2M	3753.52 ± 1564.79	5059.42 ± 2522.85	2394.88 ± 240.69

Table 2. Average returns computed over 100 episodes for each method. All tasks are primitive tasks where an agent does not need to learn hierarchical policies. Although BC and CVAE+T2M perform properly in these tasks, they yield high variance due to compounding errors. Our method achieves performance comparable with that of GAIL across all the primitive tasks.



(a) HalfCheetah-v2 (b) Walker2d-v2 (c) Hopper-v2

Figure 8. Learning curves for primitive tasks. The darker-colored lines and shaded areas denote the average returns and standard deviations, respectively, computed over 10 random seeds. These graphs indicate that our method can successfully learn primitive tasks.

pound tasks, it cannot learn the high-dimensional task due to compounding errors, as discussed in Section 2. This result demonstrates that our method is more efficient than the baselines to deal with high-dimensional compound tasks.

Table 1 shows the average returns and success rates for FetchPickAndPlace-v1. We compute the results over 100 episodes based on the provided dense rewards, which are defined as the negative distance between a box and a target point. The results show that our method performs significantly better than the baselines. To our knowledge, our method achieves state-of-the-art performance on this task without even using task-specific knowledge, outperforming previous IL methods (Sharma et al., 2018). In Figure 7, we show the plot of sub-task variables inferred by our method with snapshots. This plot represents that our work segments the compound task into two sub-tasks: picking up a box from a table and moving the box to a target point. We show our agent's movements for the first sub-task in snapshots A and B, and those for the second sub-task in snapshots C and D. The plot also describes that the transition from the first to second sub-task is performed properly without unstable switching between them.

5.2. Primitive Tasks

We also conduct experiments on three primitive tasks: HalfCheetah-v2, Walker2d-v2, and Hopper-v2. The tasks are simulated using the MuJoCo physics engine (Todorov et al., 2012). We show the learning curves and training results for each task in Figure 8 and Table 2, respectively. The results show that all the baselines achieve relatively good performance, unlike in compound tasks. This implies that an agent does not need to learn hierarchical policies to solve primitive tasks, where states are in one-to-one correspondence with the optimal or expert-like actions. In addition, we also observe that our method achieves performance comparable with that of GAIL, which attains the best performance among the baselines. This result demonstrates that our method can be applied to any target task without identifying its type.

6. Conclusion

Our work present a novel IL method that can learn compound tasks without task-specific knowledge. The key idea is to leverage self-supervised learning to train the model that describes the hierarchical structure of compound tasks. We evaluate our method on several compound tasks, including the newly introduced tasks. Our experimental results demonstrate that the proposed method can learn their hierarchical structure and policies for each sub-task, outperforming previous methods. In future work, we will extend our work to address more difficult scenarios, such as long-term or image-based compound tasks. Another interesting direction for future work is to investigate how the sample efficiency of our method can be improved.

Acknowledgements

We appreciate Yeon-Jun Lee, Aregawi Tewodros, Yoon-Jae Jung, Sang-Hyun Kim, Doo-San Baek and Gyu-Min Oh for their helpful discussions. This work was supported by ThorDrive and the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT(2017R1E1A1A01075171).

References

Abbeel, P. and Ng, A. Y. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 1. ACM, 2004.

Achille, A. and Soatto, S. Information dropout: Learning optimal representations through noisy computation. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2897–2905, 2018.

Alemi, A. A., Fischer, I., Dillon, J. V., and Murphy, K. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

Boyd, S. and Vandenberghe, L. *Convex optimization*. Cambridge university press, 2004.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym, 2016. *arXiv preprint arXiv:1606.01540*, 2016.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pp. 2172–2180, 2016.

Doersch, C., Gupta, A., and Efros, A. A. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1422–1430, 2015.

Duan, Y., Andrychowicz, M., Stadie, B., Ho, O. J., Schneider, J., Sutskever, I., Abbeel, P., and Zaremba, W. One-shot imitation learning. In *Advances in neural information processing systems*, pp. 1087–1098, 2017.

Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

Finn, C., Levine, S., and Abbeel, P. Guided cost learning: Deep inverse optimal control via policy optimization. In *International Conference on Machine Learning*, pp. 49–58, 2016.

Fox, R., Krishnan, S., Stoica, I., and Goldberg, K. Multi-level discovery of deep options. *arXiv preprint arXiv:1703.08294*, 2017.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Ho, J. and Ermon, S. Generative adversarial imitation learning. In *Advances in neural information processing systems*, pp. 4565–4573, 2016.

Kipf, T., Li, Y., Dai, H., Zambaldi, V., Sanchez-Gonzalez, A., Grefenstette, E., Kohli, P., and Battaglia, P. Compile: Compositional imitation learning and execution. In *International Conference on Machine Learning*, pp. 3418–3428, 2019.

Krishnan, S., Fox, R., Stoica, I., and Goldberg, K. Ddco: Discovery of deep continuous options for robot learning from demonstrations. *arXiv preprint arXiv:1710.05421*, 2017.

Le, H. M., Jiang, N., Agarwal, A., Dudík, M., Yue, Y., and Daumé III, H. Hierarchical imitation and reinforcement learning. *arXiv preprint arXiv:1803.00590*, 2018.

Li, Y., Song, J., and Ermon, S. Infogail: Interpretable imitation learning from visual demonstrations. In *Advances in Neural Information Processing Systems*, pp. 3812–3822, 2017.

Manschitz, S., Kober, J., Gienger, M., and Peters, J. Learning movement primitive attractor goals and sequential skills from kinesthetic demonstrations. *Robotics and Autonomous Systems*, 74:97–107, 2015.

Meier, F., Theodorou, E., Stulp, F., and Schaal, S. Movement segmentation using a primitive library. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3407–3412. IEEE, 2011.

Mohamed, S. and Rezende, D. J. Variational information maximisation for intrinsically motivated reinforcement learning. In *Advances in neural information processing systems*, pp. 2125–2133, 2015.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 16–17, 2017.

Peng, X. B., Abbeel, P., Levine, S., and van de Panne, M. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics (TOG)*, 37(4):143, 2018a.

Peng, X. B., Kanazawa, A., Toyer, S., Abbeel, P., and Levine, S. Variational discriminator bottleneck: Improving imitation learning, inverse rl, and gans by constraining information flow. *arXiv preprint arXiv:1810.00821*, 2018b.

Plappert, M., Andrychowicz, M., Ray, A., McGrew, B., Baker, B., Powell, G., Schneider, J., Tobin, J., Chociej, M., Welinder, P., et al. Multi-goal reinforcement learning: Challenging robotics environments and request for research. *arXiv preprint arXiv:1802.09464*, 2018.

Ross, S. and Bagnell, D. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 661–668, 2010.

Ross, S., Gordon, G., and Bagnell, D. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635, 2011.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897, 2015.

Sharma, A., Sharma, M., Rhinehart, N., and Kitani, K. M. Directed-info gail: Learning hierarchical policies from unsegmented demonstrations using directed information. *arXiv preprint arXiv:1810.01266*, 2018.

Shiarlis, K., Wulfmeier, M., Salter, S., Whiteson, S., and Posner, I. Taco: Learning task decomposition via temporal alignment for control. *arXiv preprint arXiv:1803.01840*, 2018.

Tishby, N., Pereira, F. C., and Bialek, W. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

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.

Wu, Y., Mansimov, E., Grosse, R. B., Liao, S., and Ba, J. Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. In *Advances in neural information processing systems*, pp. 5279–5288, 2017.

Wulfmeier, M., Wang, D. Z., and Posner, I. Watch this: Scalable cost-function learning for path planning in urban environments. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2089–2095. IEEE, 2016.

Xu, D., Nair, S., Zhu, Y., Gao, J., Garg, A., Fei-Fei, L., and Savarese, S. Neural task programming: Learning to generalize across hierarchical tasks. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–8. IEEE, 2018.

Yang, Y., Li, Y., Fermuller, C., and Aloimonos, Y. Robot learning manipulation action plans by" watching" unconstrained videos from the world wide web. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Yu, T., Abbeel, P., Levine, S., and Finn, C. One-shot hierarchical imitation learning of compound visuomotor tasks. *arXiv preprint arXiv:1810.11043*, 2018.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., and Dey, A. K. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pp. 1433–1438. Chicago, IL, USA, 2008.