Soft Option Transfer

Jinke He *
Interactive Intelligence Group
Delft University of Technology
J.He-4@tudelft.nl

Maximilian Igl
Department of Computer Science
University of Oxford
maximilian.igl@gmail.com

Matthew Smith
Department of Computer Science
University of Oxford
matt.ja.smith@gmail.com

Wendelin Boehmer
Department of Computer Science
University of Oxford
wendelin.boehmer@cs.ox.ac.uk

Shimon Whiteson
Department of Computer Science
University of Oxford
shimon.whiteson@cs.ox.ac.uk

Abstract

Transferring learnt options in hierarchical RL can yield poor performance when they are even slightly misaligned to the new task. This paper introduces *soft option transfer*: the given options are treated as a prior to learn task-specific option posteriors. This combines the fast exploration of transferred options with the flexibility to adjust them if need be. We investigate our approach in the taxi domain with varying option applicability and exploration complexity. The experiments demonstrate a clear advantage over flat policies and 'hard' options augmented with primitive actions.

1 Introduction

Most RL methods train an agent from scratch, even when experiences from similar tasks are readily available. This is in stark contrast to humans, who easily transfer skills across tasks, domains, and contexts.

One popular approach to transfer knowledge in RL (see e.g. Dietterich, 2000; Barreto et al., 2017, for others), is the options framework (Sutton et al., 1999). Options are temporally extended actions that can be learnt and transferred to new tasks. While learning a set of options can greatly improve exploration for new tasks, and thereby speed up learning (Sutton et al., 1999), there are currently no flexible methods for reusing learnt options. In most existing work, the learnt options are fixed and only a new master policy over options is learnt for the new task. This is severely restrictive: if the options are even slightly misaligned, it may not be possible to represent a high performing policy. Furthermore, previous work (Jong et al., 2008) has shown, and we confirm their results, that augmenting options with atomic actions to overcome this restrictiveness can hurt exploration. Using a set of fixed options can therefore accelerate learning, but also limit the range of tasks where optimal performance can be achieved.

Recently, Igl et al. (2019) formulated options using the RL as Inference framework (Levine, 2018) in multi-task settings. This allows learning of "soft options" for each task, which are regularized

*Work performed as a master student at University of Oxford

Learning Transferable Skills Workshop at NeurIPS 2019, Vancouver, Canada.

against a shared option prior. This method learns options that are a useful prior for a range of tasks, but each task can adjust them freely if need be. Based on this idea, this paper investigates *soft option transfer*, a new method to solve RL tasks with given options, even if these options cannot initially represent a high performing policy. We treat the given options as a fixed prior and learn another set of posterior options that may deviate from them, if it results in higher expected return for the new task. We demonstrate that this method allows a more flexible reuse of options to speed up learning. Furthermore, we provide additional insights into the more fundamental question of when knowledge transfer can be expected to be advantageous. In particular, we investigate the impact of task-difference (i.e., misspecification of options) and exploration complexity.

2 Related Work

A variety of approaches have been proposed that use a hierarchical composition of skills (Sutton et al., 1999; McGovern and Barto, 2001; Gregor et al., 2016; Eysenbach et al., 2019; Vezhnevets et al., 2017; Nachum et al., 2018; Dayan and Hinton, 1993; Bacon et al., 2017; Thrun and Schwartz, 1995). However, using skills in a transfer setting has been investigated significantly less often, especially in settings where environmental changes lead to a *misspecification* of the previously obtained skills. We argue that this is the more relevant scenario as behaviors rarely translate exactly from one task to another. One approach to adapt to changing requirements is to allow the skills to be modified by a higher-level controller (Schaul et al., 2015; Heess et al., 2016; Haarnoja et al., 2018). This restores the required flexibility of skills but might not guide exploration sufficiently to overcome hard exploration tasks. Lastly, some recent works have explored the RL as Inference framework for transfer. Goyal et al. (2019) uses the regularization cost to identify decision states while Tirumala et al. (2019) also investigates settings in which the task remains the same while the physical body of the agent changes. Contrary to our work in which we transfer the learnt skills, they transfer the master policy while re-learning the lower-level skill.

3 Background and Methodology

This paper follows the formalism of Igl et al. (2019) to transfer misspecified options. We consider several tasks $i \in \mathcal{T}$ drawn from a task distribution with $\xi(i)$, each described as a Markov Decision Process (MDP) $\langle \mathcal{S}_i, \mathcal{A}_i, \rho_i, P_i, r_i \rangle$. To simplify transfer, all tasks share a state space $\mathcal{S}_i := \mathcal{S}$ and action space $\mathcal{A}_i := \mathcal{A}$, but can have distinct initial state distributions ρ_i, transition dynamics $P_i(s_{t+1}|s_t, a_t)$ and reward functions $r_i(s_t, a_t)$. The goal is to maximize the expected discounted sum of rewards $J := \mathbb{E}_{\pi_\theta}[\sum_{t=0}^{\infty} \gamma^t r_t]$, optimized w.r.t. the parameters θ of the agent's hierarchical policy π_θ.

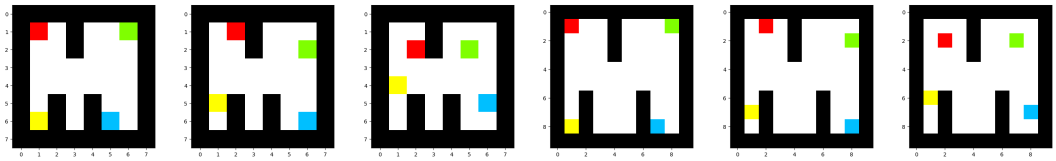
In the options framework (Sutton et al., 1999), an option o consists of an initiation set $\mathcal{I}_o \subseteq \mathcal{S}$, intra-option policy p^L, and termination policy p^T. We assume for simplicity $\mathcal{I}_o := \mathcal{S}, \forall o \in \mathcal{O}$, and that all options share parameters by conditioning both policies on the executed option o_t. The intra-option policy $p^L(a_t|s_t, o_t)$ is therefore executed until a terminal state s_t with $b_t = 1, b_t \sim p^T(b_t|s_t, o_{t-1})$ is reached. After termination of the previous option, the master policy $\pi_\theta^H(o_t|s_t)$ chooses a new option.

MSOL (Igl et al., 2019) employs the RL as Inference framework (Todorov, 2008; Toussaint, 2009; Kappen et al., 2012; Levine, 2018) to simultaneously solve multiple tasks. Each task optimizes its own soft options $\langle \pi_\theta^L(a_t|s_t, o_t), \pi_\theta^T(b_t|s_t, o_{t-1}) \rangle$, which are regularized (with parameter β) to resemble a common option prior $\langle p^L(a_t|s_t, o_t), p^T(b_t|s_t, o_{t-1}) \rangle$. The loss for episode $\{s_t, b_t, o_t, a_t, r_t\}_{t=1}^T$ is:

$$\mathcal{L}(\theta) \quad := \quad -\sum_{t=1}^{T} \mathbb{E}_{\pi_\theta}\Big[r_t - \beta \ln \Big(\frac{\pi_\theta^T(b_t|s_t,o_{t-1})}{p^T(b_t|s_t,o_{t-1})} \frac{b_t\, \pi_\theta^H(o_t|s_t) + (1-b_t)\delta_{o_t o_{t-1}}}{b_t/|\mathcal{O}| + (1-b_t)\delta_{o_t o_{t-1}}} \frac{\pi_\theta^L(a_t|s_t,o_t)}{p^L(a_t|s_t,o_t)} \Big) \Big] . \qquad (1)$$

After training both the posterior of each task and the parameterized prior over all tasks, MSOL transfers the learnt options to new tasks from $\mathcal{T}$. In the test task, MSOL trains the master policy, which is initialized randomly, and the posterior, which is initialized with the parameters of the prior, with the same loss while keeps the prior fixed.

This paper investigates what happens when the test task is *not* from $\mathcal{T}$ and the learnt options are ill equipped to solve it. In principle, any set of options could be used as a prior, including hand-crafted options and those learnt by other methods like option-critic (Bacon et al., 2017). However, for the sake of simplicity we train the transferred options with MSOL, and leave a transfer from other methods for future work. The option priors trained on $\mathcal{T}$ can be misspecified for a new task $i \notin \mathcal{T}$



(a) 8x8 original (b) 8x8 modified (c) 8x8 further-modified (d) 10x10 original (e) 10x10 modified (f) 10x10 further-modified

Figure 1: Taxi environments with different dimensions and potential pickup/dropoff locations. Options are learnt in the original domain (a+d) and transferred to tasks with modified (b+e) or further-modified (c+f) pickup/dropoff locations. Larger grid dimensions (d-f) require more exploration.

and we investigate how well the posterior of the learnt soft-options in MSOL can compensate for that misspecification.

We observe empirically that options learnt end-to-end are often almost deterministic, which results in poor exploration for even slightly different tasks. To address this, we propose to increase the stochasticity of the given options, for example, by increasing the softmax temperatures in the given options' policies p^T and p^L over discrete action spaces.

4 Experiments

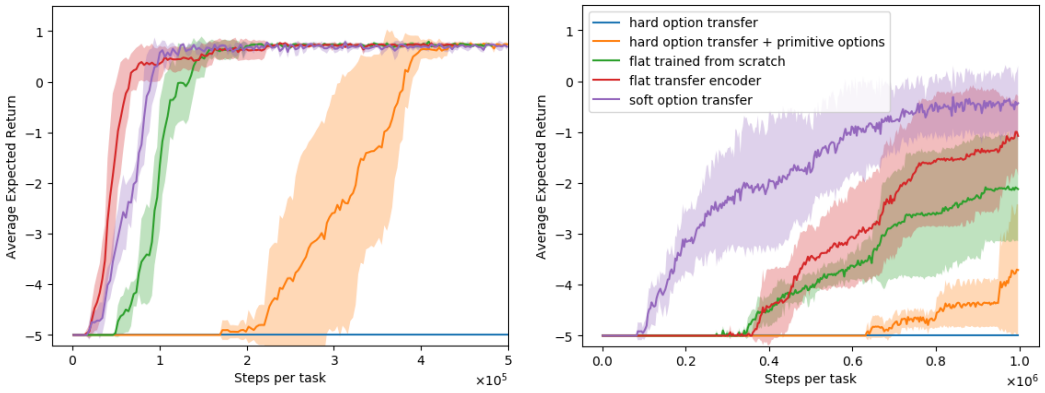
In the following, we compare soft option transfer against two sets of baselines. First, we have two 'hard' option transfer agents that freeze the options before transfer and only learn a new master policy over them. In one of them, the set of options is augmented by primitive options, i.e., atomic actions, which in principle allows it to express any policy and therefore removes the limitation on the representational capacity of the hierarchical policy. However, we show that under option misspecification, soft options easily outperform both baselines. This confirms previous results showing that augmenting the action space of an agent with hard options can hurt exploration (Jong et al., 2008).

The second set of baselines addresses the more general question: In which settings does option transfer benefit exploration and learning? To do so, we compare against two flat A2C agents that do not reuse options, one of which is trained from scratch and another transfers an 'encoder': the first 3 layers of an A2C agent solving the tasks in the original environments.

We conduct experiments in the Taxi domain motivated by Dietterich (2000) and Igl et al. (2019). Each environment is a grid world with obstacles and four potential locations for the pickup and dropoff of a passenger (see Figure 1). A *task* in an environment is defined as a combination of distinct pickup and dropoff locations, which yields 12 tasks in each environment. To gain reward, the agent needs to first move to the pickup location, execute a special pickup/dropoff action, move to the dropoff location and execute the same action again, which ends the episode. We use 4 options that are learnt from the 12 tasks of the 8x8 or 10x10 original environments (Figures 1a and 1d) by MSOL (Igl et al., 2019). Note that, to ensure generality of options, task-relevant information is withheld from them, similar to previous work, for example, (Galashov et al., 2018).

To evaluate the utility of (soft) option transfer in different settings, we vary the environment in two ways: how misspecified the learnt options are and how difficult the exploration is. To induce misspecification, we shift the pickup and dropoff locations by one grid cell, called *modified* environment (Figures 1b and 1e) or two grid cells, called *further modified* (Figures 1c and 1f). The difficulty of exploration is determined by the size of the grid, as larger environments (Figures 1d to 1f) are considerably harder.

Figure 2a plots the average expected return in the 12 tasks of the 8x8 modified environment. The hard option transfer agent fails because using the given set of options, the agent is never able to finish a single task. Interestingly, additional access to atomic actions learns much more slowly than the agent that is trained from scratch. A similar finding is reported by Jong et al. (2008), who show that under random exploration, mixing options with atomic actions can hurt performance.



(a) in the 8x8 modified environment

(b) in the 10x10 modified environment

Figure 2: Performance of soft option transfer and baseline methods in 12 tasks of the modified environments shown in Figures 1b and 1e. Note that soft option transfer outperforms hard option transfer significantly. Flat policies solve the task in the smaller environment (left) similarly well, but soft options have a clear advantage in larger environments (right) when even misaligned options improve exploration significantly.

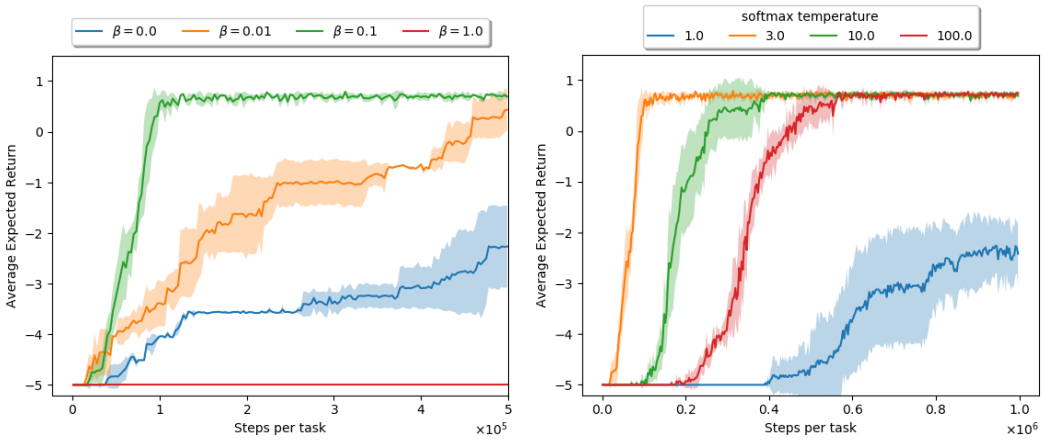


Figure 3: Left: Performance of soft option transfer agents trained with various regularization parameters β. Right: Performance of soft option transfer agents whose option posteriors are initialized to the same options but are regularized towards option priors that are softened with different softmax temperatures. Both of the transfer experiments are conducted in the 8x8 modified environments.

Because exploration in tasks of the 8x8 modified environment is not hard enough, transferring a state representation given by an encoder slightly outperforms soft options. By contrast, for larger grid sizes, Figure 2b demonstrates that in these tasks soft option transfer learns much faster and achieves higher performance than all baselines: because the exploration challenge on those tasks is much harder, even misspecified options accelerate learning as long as they are flexible, i.e., soft.

We also compare the soft option with flat agents in the 8x8 and 10x10 further-modified environments (Figures 1c and 1f), which are easier to explore, as pickup and dropoff locations are closer to each other, but the given options are more misspecified (see Figure 4 in the appendix).

The key feature of soft option transfer that enables faster learning is that it uses option priors to guide the policy search. However, the advantage of this guidance depends on how much it can help to overcome hard exploration and how much it distracts the agent from finding the optimal policy. This trade-off is captured by soft options in the temperature parameter β. For example, in the further misspecified environment (Figure 4), the performance of soft options can be raised by lowering β compared to the less misspecified setting. To explore this further, in Figure 3a, we compare soft

option transfer agents trained with different regularization parameters β. With $\beta = 1.0$, the agent refuses to deviate from the option priors because of the strong regularization and does not learn anything. With $\beta = 0.0$ and $\beta = 0.01$, the agents learn much more slowly than the agent with $\beta = 0.1$, due to the lack of guidance to overcome the hard exploration. In all cases, the policies are initialized to the prior, showing the importance of regularization in addition to initialization.

Lastly, in Figure 3b, we compare soft option transfer agents, where we initialized the posteriors to the given options softened by the same softmax temperature 3.0, but regularized towards priors with varying softmax temperatures, which has similar effects to tuning β. The figure shows that even with the same initial posterior options and the same temperature parameter $\beta = 0.1$, the priors have a large impact. On the one hand, very deterministic option priors, with softmax temperature 1.0, prevent the agent from exploring sufficiently to find an optimal policy. On the other hand, agents with large softmax temperatures, like 10.0 and 100.0, learn more slowly because regularizing posterior options towards prior options that are very stochastic results in the loss of guided exploration. All other presented results were computed with a softmax temperature of 3.0.

References

Bacon, P.-L., Harb, J., and Precup, D. (2017). The option-critic architecture. In *AAAI*, pages 1726–1734.

Barreto, A., Dabney, W., Munos, R., Hunt, J. J., Schaul, T., Van Hasselt, H., and Silver, D. (2017). Successor features for transfer in reinforcement learning. In *Advances in neural information processing systems*, pages 4056–4066.

Dayan, P. and Hinton, G. E. (1993). Feudal reinforcement learning. In *Advances in neural information processing systems*, pages 271–278.

Dietterich, T. G. (2000). Hierarchical Reinforcement Learning with the MAXQ Value Function Decomposition. *Journal of Artificial Intelligence Research*, 13:227–303.

Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. (2019). Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations*.

Galashov, A., Jayakumar, S., Hasenclever, L., Tirumala, D., Schwarz, J., Desjardins, G., Czarnecki, W. M., Teh, Y. W., Pascanu, R., and Heess, N. (2018). Information asymmetry in kl-regularized rl.

Goyal, A., Islam, R., Strouse, D., Ahmed, Z., Botvinick, M., Larochelle, H., Bengio, Y., and Levine, S. (2019). InfoBot: Transfer and Exploration via the Information Bottleneck. *arXiv preprint arXiv:1901.10902*.

Gregor, K., Rezende, D. J., and Wierstra, D. (2016). Variational intrinsic control. *arXiv preprint arXiv:1611.07507*.

Haarnoja, T., Hartikainen, K., Abbeel, P., and Levine, S. (2018). Latent Space Policies for Hierarchical Reinforcement Learning. In *International Conference on Machine Learning*, pages 1846–1855.

Heess, N., Wayne, G., Tassa, Y., Lillicrap, T., Riedmiller, M., and Silver, D. (2016). Learning and transfer of modulated locomotor controllers. *arXiv preprint arXiv:1610.05182*.

Igl, M., Gambardella, A., Nardelli, N., Siddharth, N., Böhmer, W., and Whiteson, S. (2019). Multitask Soft Option Learning. *arXiv preprint arXiv:1904.01033*.

Jong, N. K., Hester, T., and Stone, P. (2008). The utility of temporal abstraction in reinforcement learning. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, volume 1, pages 294–301.

Kappen, H. J., Gómez, V., and Opper, M. (2012). Optimal control as a graphical model inference problem. *Machine learning*, 87(2):159–182.

Levine, S. (2018). Reinforcement Learning and Control as Probabilistic Inference: Tutorial and Review. *arXiv preprint arXiv:1805.00909*.

McGovern, A. and Barto, A. G. (2001). Automatic Discovery of Subgoals in Reinforcement Learning using Diverse Density. In *International Conference on Machine Learning*, pages 361–368.

Nachum, O., Gu, S., Lee, H., and Levine, S. (2018). Data-efficient hierarchical reinforcement learning. *arXiv preprint arXiv:1805.08296*.

Schaul, T., Horgan, D., Gregor, K., and Silver, D. (2015). Universal value function approximators. In *International Conference on Machine Learning*, pages 1312–1320.

Sutton, R. S., Precup, D., and Singh, S. (1999). Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1-2):181–211.

Thrun, S. and Schwartz, A. (1995). Finding structure in reinforcement learning. In *Advances in neural information processing systems*, pages 385–392.

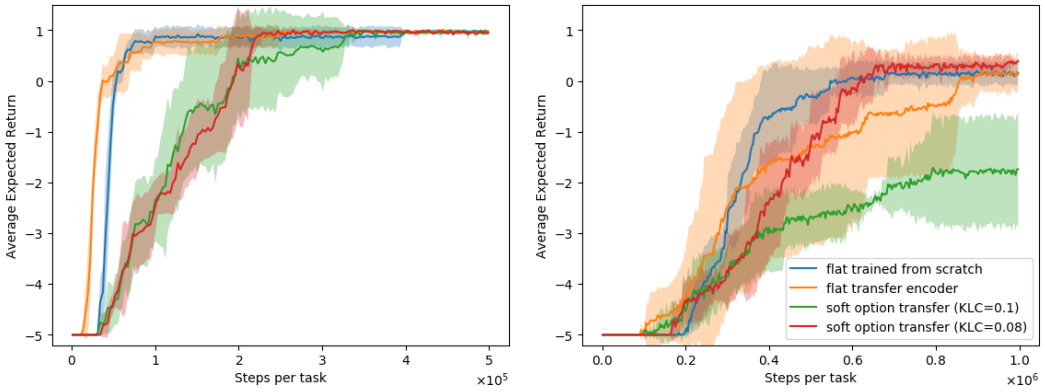
Tirumala, D., Noh, H., Galashov, A., Hasenclever, L., Ahuja, A., Wayne, G., Pascanu, R., Teh, Y. W., and Heess, N. (2019). Exploiting hierarchy for learning and transfer in kl-regularized rl. *arXiv preprint arXiv:1903.07438*.

Todorov, E. (2008). General duality between optimal control and estimation. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 4286–4292. IEEE.

Toussaint, M. (2009). Robot trajectory optimization using approximate inference. In *Proceedings of the 26th annual international conference on machine learning*, pages 1049–1056. ACM.

Vezhnevets, A. S., Osindero, S., Schaul, T., Heess, N., Jaderberg, M., Silver, D., and Kavukcuoglu, K. (2017). Feudal networks for hierarchical reinforcement learning. In *International Conference on Machine Learning*, pages 3540–3549.

Appendix



(a) in the 8x8 further-modified environment

(b) in the 10x10 further-modified environment

Figure 4: Performance of soft option transfer and baseline methods in 12 tasks of the further modified environments shown in Figures 1c and 1f, in which the potential pickup and dropoff locations are pushed further away from those in the original environments, making the tasks easier for exploration and learnt options further misspecified in the tasks. Although reducing the strength of KL regularization can improve the performance of the soft option transfer agents, in these environments the advantage of reusing misspecified options is lost.