Free-Energy Advantage Functions for Policy Transfer to Noisy Environments with Safety Constraints

Pierre Haritz[1,2], Thomas Liebig[1,2]

[1]Chair of Artificial Intelligence, Faculty of Computer Science, TU Dortmund University, Dortmund, Germany
[2]Lamarr Institute for Machine Learning and Artificial Intelligence, Dortmund, Germany

Abstract

Training acting agents for the goal of controlling complex live systems on the system itself is often an unfeasible task, either due to the high cost or the potential dangers that might arise. In this paper, we take a step towards identifying ways to evaluate the transferability of models for the class of constrained Reinforcement Learning problems. Furthermore, we present an approach based on free-energy advantage functions to improve adaptability and in turn transferability for constrained Reinforcement Learning problems and subsequently manage to increase the performance of a baseline algorithm, CPO, with regard to safety constraints in noisy environments.

Keywords

reinforcement learning, transfer learning, safety

1. Introduction

AI systems can have significant real-world impact, and if not designed and deployed with safety in mind, they can cause harm to individuals, organizations, or society as a whole. Ensuring safety is crucial to prevent accidents, unintended consequences, or malicious uses of AI. When deploying trained models to large-scale industrial applications, unstable live systems can cause damage of economic or other nature. Because of the high complexity, cost, and potential danger of training live systems from scratch, usually, these models are trained on historical or simulation data, which may or may not accurately reflect the actual use case environment. Specifically, in some instances, knowledge of the actual environment dynamics is only partially available, and algorithms need to be able to handle situations where there is a degree of uncertainty. Classically, in control environments, robustness can be achieved with Model Predictive Control approaches ([1]) when plant dynamics are known.

Reinforcement Learning (RL) is a machine learning paradigm that includes a variety of algorithmic approaches, foremost in sequential decision-making environments. Recently, RL has become a promising way to solve sequential decision-making tasks such as in marketing, gaming, and control tasks, such as robotics and autonomous cars, where the aspect of safety and trustworthiness in the agent is an important factor.

We argue that in real-world applications that require safety guarantees, RL methods that transfer well could improve upon satisfying certain thresholds.

LWDA'23: Lernen, Wissen, Daten, Analysen. October 09–11, 2023, Marburg, Germany
pierre.haritz@tu-dortmund.de (P. Haritz); thomas.liebig@cs.tu-dortmund.de (T. Liebig)

Transfer learning is an established concept in areas such as image classification and natural language processing ([2]), with the goal of reducing training time for Machine Learning models and improving their performance. In this paper, we first give an overview of how Transfer Learning is interpreted in Reinforcement Learning and discuss the benefit of transferability in constrained Reinforcement Learning. Our contribution in this paper can be stated as such:

- We propose criteria to evaluate policy transfer in constrained RL.
- We present a method for improving performance regarding safety after transferring pre-trained policies to a noisy environment through the use of free-energy advantage functions.

2. Background and Related Work

In this section, we will introduce the mathematical framework for the problem setting.

2.1. Reinforcement Learning

Reinforcement Learning problems can typically be modeled with the help of a Markov Decision Process (MDP) $M = (S, A, T, \gamma, R)$ with a state space S, an action space A, a transition probability function $T : S \times A \times S \to [0, 1]$, a discount factor $\gamma \in [0, 1]$ and a reward function $R : S \times A \to \mathbb{R}$.

To extend this to safety critical problems, one possibility is to introduce a constraint cost function $C : S \times A \to \mathbb{R}$ analogue to the reward function and a safety threshold $c \in \mathbb{R}$. We define a Constrained Markov Decision Process (from now on referred to as CMDP) $M_C = (S, A, T, \gamma, R, C, c)$. We can calculate a weighted return value for constrained problems with $J_C(\pi) = \mathop{\mathbb{E}}_{\tau\sim\pi}[\sum_{t=0}^{\infty} \gamma^t C(s_t, a_t)]$ of a policy $\pi : S \to A$ with $\pi \in \Pi$ for the set of all policies Π and a trajectory $\tau = (s_0, a_0, s_1, a_1, \dots)$.
Let $\Pi_C = \{\pi \in \Pi : J_C(\pi) \leq c\}$ be the set of policies that satisfy the constraint c. Then we can calculate the optimal policy $\pi^* = \arg\max_{\pi\in\Pi_C} J(\pi)$.

In real-life applications of Reinforcement Learning, environment dynamics, especially state transitions, can be unknown. Therefore, we introduce a generalization of the MDP model by assuming transition probabilities $T^{\star}_{s,a} \in \Delta^S$ for finite states and actions and probability simplex $\Delta^S \subset \mathbb{R}^S_+$. A common way to learn the objective under the assumption of unknown transition probabilities is to maximize a lower bound on the return.

2.2. Transfer Learning in the Reinforcement Learning Context

In a mathematical sense, given a source domain $M_{\mathbf{S}}$ and a target domain $M_{\mathbf{T}}$, Transfer Learning (TL) is used to learn an optimal policy π^* for $M_{\mathbf{T}}$ by incorporating both external information from the source $\mathcal{I}_{\mathbf{S}}$ and internal information $\mathcal{I}_{\mathbf{T}}$ gathered from $M_{\mathbf{T}}$. The optimal policy can be written as $\pi^* = \arg\max_{\pi} \mathop{\mathbb{E}}_{x\sim\mu^t, a\sim\pi}[Q^{\pi}_M(x, a)]$ for initial set of states μ. Taylor and Stone [3] highlight the benefits of using transfer methods in RL tasks and categorize measurements as such:

- Performance improvement of the initial policy by transferring an agent from a source task to a target task.
- Performance improvement of the final learned policy of an agent on a target task by transferring.
- The gained total cumulative reward from a transfer strategy compared to a non-transfer strategy.
- The ratio of the total reward accumulated by the transfer learner and the total reward accumulated by the non-transfer learner.
- The reduction of learning time needed by the agent to achieve a pre-specified performance level via knowledge transfer.

In literature ([4]) a variety of TL approaches that fall under this category, are mentioned: In Imitation learning, the agent is trained to mimic a policy of a source policy, called the expert. This is a way of training without having access to feedback from the environment. A framework for Imitation Learning in partially-observable settings based on the Free-Energy Principle has been proposed in [5]. In cases where the reward signal is available, Learning from Demonstrations (LfD) is a possible way of training an agent. The way agents combine their knowledge (inter-agent or intra-agent) in Cooperative Multi-Agent RL can also be described as a form of TL.

In TL, domains can be described by MDPs, and any parts of it can have differences between the source and target domain. Consider state spaces $S_{\mathbf{S}}$ and $S_{\mathbf{T}}$. Any of these relations might be true, depending on the problem: $S_{\mathbf{S}} \subset S_{\mathbf{T}}$, $S_{\mathbf{S}} \equiv S_{\mathbf{T}}$ or $S_{\mathbf{S}} \subset S_{\mathbf{T}}$. Differences for the action spaces $A_{\mathbf{S}}$ and $A_{\mathbf{T}}$ are analogs. Since both state and action spaces can differ, reward functions can also be defined differently for both domains. Ultimately, trajectories can differ for problems where reaching a goal can be achieved differently (e.g., path-finding tasks).

This can be further extended to safety critical applications. Differing state spaces can be the result of failed sensors, differing action spaces are the result of hard constraints implemented by the system. Additionally, reward functions might yield different values in cases where sensors supply noisy data. In the case of CMDPs, for similar reasons, differences can be found in both constrained cost function and safety threshold.

On the topic of which kind of knowledge is transferable, we can define multiple forms. The transfer of trajectories is the main subject of LfD. Furthermore, the transfer of model dynamics is possible when an approximation by offline learning algorithms trained on historical data, and before getting transferred to an online system, is feasible. Offline RL algorithms usually mitigate the impact of the gap between real and estimated values by adding a pessimism factor to these learned values ([6]) or learned dynamic models ([7]).

The transfer of policies has been discussed by [8]. They propose to extend the exploration-exploitation choice with the option to reuse an older policy and consequently test the transfer performance. Reward Shaping (as presented in [9]) speeds up the RL training process by guiding the exploration process by transforming the reward function into a potential-based reward function.

Transfer by starting from prior distributions has been explored by [10]. Instead of finding trajectories that maximize expected rewards, inference formulations start from a prior distribution over trajectories, condition the desired outcome, such as achieving a goal state, and

then estimate the posterior distribution over trajectories consistent with this outcome. Since imitation learning provides a teacher policy to learn from, it interprets the teacher policy as a prior policy distribution.

3. Using Free-Energy Priors to improve Robustness after Policy Transfer

In real-world applications, such as robotics, it can be hard to separate signals from noise, especially at the early stages after deploying a learned strategy. We consider a scenario where there is a cost to receiving state data from an actor, e.g., sensor data from a robot's joints. Since we are considering the case of a *SimToReal*-transfer, we assume the existence of priors learned from simulation interactions. In this section, we propose the use of an advantage function over the simulation priors based on the free-energy principle to improve the agent's robustness.

3.1. Free-Energy Functions

Free-energy functions are fundamental concepts in thermodynamics and statistical mechanics that describe the energy available to do work in a system while accounting for both its internal energy and its entropy.

3.2. Quanitifying the cost of Control

Rubin et al. [11] borrow the term to define free-energy functions in the RL context to derive optimal policies and explore the tradeoff between value and control information. The idea is that optimal policies reflect a balance between maximizing expected rewards (value) and minimizing the information cost that comes with control.

With the help of information theory, we can quantify the expected cost of executing a policy π in state $s \in S$ as $\Delta I(s) = \sum_a \pi_s^{\mathbf{T}}(a) \log(\frac{\pi_s^{\mathbf{T}}(a)}{\pi_s^{\mathbf{S}}(a)})$ with $\Delta I(s_T) = 0$ for a terminal state s_T. With this, we are able to measure the relative entropy between the source policy $\pi^{\mathbf{S}}$ and target policy $\pi^{\mathbf{T}}$. The source policy is used by the agent in the absence of information from its new noisy environment. For any state s, $\Delta I(s)$ describes the minimal number of bits required to describe the outcome, or action sampled, of the random variable $a \sim \pi^{\mathbf{T}}$. In our case, it serves as a measure for the cost of control. Similar to the value function $V_\pi(s_0)$, we can define the total control information involved in executing policy π starting from the initial state s_0:

$$
\begin{aligned}
I_\pi(s_0) &= \lim_{T\to\infty} \mathbb{E}[\Delta I(s_t)] \\
&= \lim_{T\to\infty} \mathbb{E}\left[\sum_{t=0}^{T-1} \log \frac{\pi_{s_t}^{\mathbf{T}}(a_t)}{\pi_{s_t}^{\mathbf{S}}(a_t)} \right] \\
&= \lim_{T\to\infty} \mathbb{E}\left[\log \frac{Pr(a_0, a_1, \dots, a_{T-1} | s_0, \pi^{\mathbf{T}})}{Pr(a_0, a_1, \dots, a_{T-1} | s_0, \pi^{\mathbf{S}})} \right]
\end{aligned}
\tag{1}
$$

Here, the optimal target policy $\pi^{*,\mathbf{T}}$ should minimize the control information cost and at the same time maximize the reward while respecting environmental constraints.

3.3. Optimization with constrained Policies

In safety-critical domains, RL optimization problems are typically subject to constraints. For a distance measure $d : \Pi \times \Pi \to \mathbb{R}$ and step size δ, trust-region policy optimization algorithms makes sure that the new policy is within a so-called trust region of the previous one: $\pi_{t+1} = \arg\max_{\pi \in \Pi_\theta} J(\pi)$ s.t. $J_C(\pi) \leq c$ and $d(\pi, \pi_t) \leq \delta$. Here, $\Pi_\theta \subset \Pi$ denotes a θ-parameterized policy subset that filters for relevant parameters. Trust region algorithms for reinforcement learning ([12, 13], such as CPO, have policy updates of the form

$$\begin{aligned} \pi_{k+1} = & \arg\max_{\pi \in \Pi_\theta} \mathbb{E}_{s \sim d^{\pi_k}, a \sim \pi}[A^{\pi_k}(s,a)], \\ & \text{s.t. } D_{KL}(\pi \| \pi_k) \leq \delta \end{aligned} \tag{2}$$

where $D_{KL}(\pi || \pi_k) = \mathbb{E}_{s \sim d^{\pi_k}}[D_{KL}(\pi \| \pi_k)[s]])$, and $\delta > 0$ is the step size.

The advantage functions calculates the expected reward gain along a trajectory and is given by:

$$\begin{aligned} A^\pi(s,a) &= Q^\pi(s,a) - V^\pi(s) \\ &= \mathbb{E}_{\tau \sim \pi}[R(\tau) | s_0 = s, a_0 = a] - \mathbb{E}_{\tau \sim \pi}[R(\tau) | s_0 = s] \end{aligned} \tag{3}$$

The *trust region* is then defined by the set $\{\pi_\theta \in \Pi_\theta : D_{KL}(\pi \| \pi_k)\}$.
CPO solves the CMDP problem approximately by calculating the update

$$\begin{aligned} \pi_{k+1} = & \arg\max_{\pi \in \Pi_\theta} \mathbb{E}_{s \sim d^{\pi_k}, a \sim \pi}[A^{\pi_k}(s,a)] \\ & \text{s.t. } J_{C_i}(\pi_k) + \mathbb{E}_{s \sim d^{\pi_k}, a \sim \pi}\left[\frac{A^{\pi_k}_{C_i}(s,a)}{1-\gamma}\right] \leq c_i \\ & D_{KL}(\pi \| \pi_k) \leq \delta \end{aligned} \tag{4}$$

3.4. Using Free-Energy Functions to improve Transferability

We aim to use a free-energy function to derive optimal policies while balancing the tradeoff between value and information during exploring.

Early works ([14]) propose using advantage functions in noisy environments to mitigate undesired approximation effects by reducing the action gap ([15]). We assume a stochastic prior policy $\pi^{\mathbf{S}}(a|s)$ from the source task. Fox et al ([16]) propose that we can measure the information cost of a policy $\pi^{\mathbf{T}}(a|s)$ with $g^{\pi^{\mathbf{T}}}(s,a) = \log \frac{\pi^{\mathbf{T}}(a|s)}{\pi^{\mathbf{S}}(a|s)}$. The expected information cost of the target policy $\pi^{\mathbf{T}}$ can be written as $\mathbb{E}[g^{\pi^{\mathbf{T}}}(s_t, a_t|s)] = D_{KL}(\pi^{\mathbf{T}}_s \| \pi^{\mathbf{S}}_s)$. Considering the dynamics induced by the transition probabilities $T(s_{t+1}|s_t, a_t)$ of the underlying MDP, we can now consider the total discounted expected information cost for the target policy:

$$I^{\pi^{\mathbf{T}}}(s) = \sum_{t=0}^{\infty} \gamma^t D_{KL}(\pi^{\mathbf{T}}_{s_t, a_t} \| \pi^{\mathbf{S}}_{s_t, a_t}). \tag{5}$$

We define

$$F^{\pi^{\mathbf{T}}}(s,a) = V^{\pi^{\mathbf{T}}}(s) + \frac{1}{\beta} I^{\pi^{\mathbf{T}}}(s) \tag{6}$$

as a β-weighted free-energy function with β controlling the tradeoff between value and information. From this we get a state-action free-energy function

$$G^{\pi^{\mathbf{T}}}(s,a) = \mathbb{E}_{\theta}[R|s,a] + \gamma \mathbb{E}_{T}[F^{\pi^{\mathbf{T}}}(s')|s,a]. \tag{7}$$

Now, we define the free-energy advantage function as:

$$\begin{aligned} B^{\pi^{\mathbf{T}}}(s,a) &= G^{\pi^{\mathbf{T}}}(s,a) - V^{\pi^{\mathbf{T}}}(s) \\ &= \mathbb{E}_{\tau\sim\pi^{\mathbf{T}}}[C(\tau) + \frac{\gamma}{\beta} g^{\pi^{\mathbf{T}}}(s_{t+1}, a_{t+1})|s_0 = s, a_0 = a] - \mathbb{E}_{\tau\sim\pi^{\mathbf{T}}}[C(\tau)|s_0 = s] \end{aligned} \tag{8}$$

Here, $C(\tau)$ represents the cumulative sum of constraint costs along the trajectory τ.

Finally, we can calculate the free-energy advantage transfer policy update:

$$\begin{aligned} \pi_{k+1} = \ & \arg\max_{\pi\in\Pi_{\theta}} \mathbb{E}_{s\sim d^{\pi_k}, a\sim\pi}[B^{\pi_k}(s,a)] \\ & \text{s.t. } J_{C_i}(\pi_k) + \mathbb{E}_{s\sim d^{\pi_k}, a\sim\pi}\left[\frac{B_{C_i}^{\pi_k}(s,a)}{1-\gamma}\right] \leq c_i \\ & D_{KL}(\pi\|\pi_k) \leq \delta \end{aligned} \tag{9}$$

4. Results

In this section, we will present the evaluation framework, metrics and results.

4.1. Experiments

In this section we will first evaluate the performance of the Constrained Policy Optimization algorithm [17] for constrained RL problems. CPO yields better performance on constrained tasks than methods such as Trust Region Policy Optimization or Primal-Dual Optimization ([12, 18]). We conduct the experiments on an exemplary robotics learning task, specifically the *HalfCheetah* environment within the *MuJoCo*[1] physics engine embedded in *OpenAI Gym*[2]. The *HalfCheetah* is a two-dimensional simulated robot with six controllable joints, as depicted in figure 1. We use a continuous action space with $A = [-1, 1]^6$, where each entry of the action vector represents the torque *[Nm]* applied to the respective motorized joint. The constraint is placed on an angle, in which the HalfCheetah is considered to be *fallen over* and would not be able to recover to a standing position without external help.

4.2. Evaluating Transferability for Safety-Critical Applications

For safety-critical applications at any scale, the best direct improvement of TL would generally be starting from accurate prior distributions, because we can expect a reduced exploratory period. While this is expected to reduce training time, prevention of constraint violations is

[1]https://github.com/openai/mujoco-py
[2]https://github.com/openai/gym

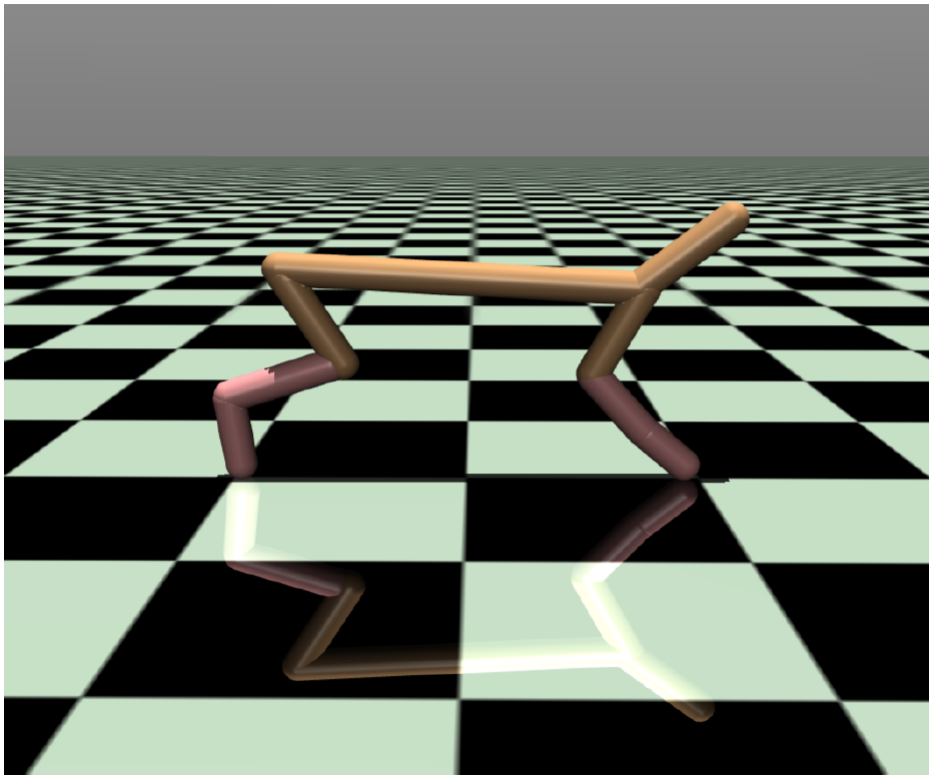

Figure 1: A rendering of the MuJoCo *HalfCheetah* environment in its initial state. Its controllable joints are highlighted in *red*.

not necessarily guaranteed. Having reliable algorithms should also make it possible to train an agent in a simulation and then transfer the model to safety-critical applications in the real world without violating constraints imposed by the task. We, therefore, extend the list by the following measurements:

- The ratio of total constraint cost accumulated by the transfer learner and total constraint cost accumulated by the non-transfer learner or between different transfer learners.
- The sum of constraint violations committed by the transfer learner compared to the non-transfer learner (or between multiple transfer learners) above a specified threshold.

Note that we hypothesize that measuring the robustness gained by simultaneously learning system dynamics ([19]) could a valid metric, which we intend to examine in the future.

4.3. Evaluation

We hence compare the CPO algorithm with and without free-energy advantage policy transfer (FEAT) on noisy environments with a noise factor $U_j \sim \mathcal{N}(1, \sigma)$ for every state variable index $j \in \{1, \dots, |s|\}$ by evaluation the post-transfer performance according to the formerly proposed criteria. In all experiments, we first pre-train an agent with an implementation of the CPO

algorithm in a simulated environment without noise for 2500 iterations. After the final iteration, the agent is able to control the *HalfCheetah* at a satisfactory level.

4.3.1. Comparison of ratios of total constraint costs

Figure 2 shows the mean constraint costs over a post-transfer training process of $T = 1000$ iterations. Our approach, CPO+FEAT (orange), manages to stay below the curve of the baseline approach, CPO (green).

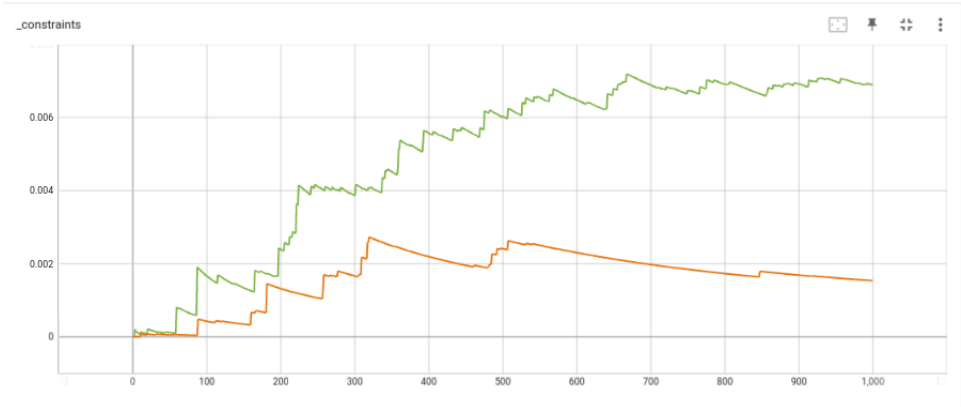

Figure 2: A comparison of mean constraint costs over $T = 1000$ iterations between CPO (green) and CPO with FEAT (orange) in a noisy environment with $\sigma = 0.1$.

4.3.2. Comparison of the sum of constraint violations

For the criterion of constraint violations, we define a constraint threshold c. Like above, we train the agents for a total of $T = 1000$ iterations. In a noisy environment with $\sigma = 0.1$, we evaluate both agents with a strict safety threshold of $c = 0.02$. Here, the value for c means that the *HalfCheetah* is not allowed to show signs of falling over. While CPO without FEAT violates the threshold 7.2% of the time, CPO with added FEAT evaluates at only 3.5%.
For $\sigma = 0.2$, we chose a higher threshold of $c = 0.15$ (the agent is allowed to appear unstable, but is not allowed to fall over). CPO without FEAT violates the threshold in 86.7% of iterations, while CPO with FEAT is significantly lower, with only 32.3% violations. Unfortunately, both algorithms still lack the necessary robustness to guarantee safety for environments with higher noise levels.

5. Conclusion and Future Work

In this paper, we highlighted how Transfer Learning can be interpreted in the context of constrained Reinforcement Learning and proposed a way that transferability can be evaluated. The experiments indicate that our approach improves the transferability of policies for constrained problems in the specific case of the Constrained Policy Optimization algorithm.

In the future, we aim to research further how this approach is applicable for similar policy based RL algorithms and extended this to a more general case. Furthermore, to reflect real-world problems more accurately, we plan to add further restrictions to the actor's perception of the environment, such as partial observability.

Acknowledgments

This research has been funded by the Federal Ministry of Education and Research of Germany and the state of North-Rhine Westphalia as part of the Lamarr-Institute for Machine Learning and Artificial Intelligence.

References

[1] M. V. Kothare, V. Balakrishnan, M. Morari, Robust constrained model predictive control using linear matrix inequalities, Automatica 32 (1996) 1361–1379.
[2] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, Q. He, A comprehensive survey on transfer learning, Proceedings of the IEEE 109 (2020) 43–76.
[3] M. E. Taylor, P. Stone, Transfer learning for reinforcement learning domains: A survey., Journal of Machine Learning Research 10 (2009).
[4] Z. Zhu, K. Lin, J. Zhou, Transfer learning in deep reinforcement learning: A survey, arXiv preprint arXiv:2009.07888 (2020).
[5] R. Ogishima, I. Karino, Y. Kuniyoshi, Reinforced imitation learning by free energy principle, arXiv preprint arXiv:2107.11811 (2021).
[6] S. Fujimoto, D. Meger, D. Precup, Off-policy deep reinforcement learning without exploration, in: International conference on machine learning, PMLR, 2019, pp. 2052–2062.
[7] R. Kidambi, A. Rajeswaran, P. Netrapalli, T. Joachims, Morel: Model-based offline reinforcement learning, Advances in neural information processing systems 33 (2020) 21810–21823.
[8] F. Fernández, J. García, M. Veloso, Probabilistic policy reuse for inter-task transfer learning, Robotics and Autonomous Systems 58 (2010) 866–871.
[9] T. Brys, A. Harutyunyan, M. E. Taylor, A. Nowé, Policy transfer using reward shaping., in: AAMAS, 2015, pp. 181–188.
[10] A. Abdolmaleki, J. T. Springenberg, Y. Tassa, R. Munos, N. Heess, M. Riedmiller, Maximum a posteriori policy optimisation, arXiv preprint arXiv:1806.06920 (2018).
[11] J. Rubin, O. Shamir, N. Tishby, Trading value and information in mdps, in: Decision Making with Imperfect Decision Makers, Springer, 2012, pp. 57–74.
[12] J. Schulman, S. Levine, P. Abbeel, M. Jordan, P. Moritz, Trust region policy optimization, in: International conference on machine learning, PMLR, 2015, pp. 1889–1897.
[13] J. Schulman, P. Moritz, S. Levine, M. Jordan, P. Abbeel, High-dimensional continuous control using generalized advantage estimation, arXiv preprint arXiv:1506.02438 (2015).
[14] L. C. Baird, Reinforcement learning in continuous time: Advantage updating, in: Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94), volume 4, IEEE, 1994, pp. 2448–2453.
[15] M. G. Bellemare, G. Ostrovski, A. Guez, P. Thomas, R. Munos, Increasing the action gap:

New operators for reinforcement learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 30, 2016.
[16] R. Fox, A. Pakman, N. Tishby, Taming the noise in reinforcement learning via soft updates, arXiv preprint arXiv:1512.08562 (2015).
[17] J. Achiam, D. Held, A. Tamar, P. Abbeel, Constrained policy optimization, in: International conference on machine learning, PMLR, 2017, pp. 22–31.
[18] Y. Chow, M. Ghavamzadeh, L. Janson, M. Pavone, Risk-constrained reinforcement learning with percentile risk criteria, The Journal of Machine Learning Research 18 (2017) 6070–6120.
[19] P. G. Sessa, I. Bogunovic, M. Kamgarpour, A. Krause, Mixed strategies for robust optimization of unknown objectives, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2020, pp. 2970–2980.