# Off-Dynamics Reinforcement Learning: Training for Transfer with Domain Classifiers

**Benjamin Eysenbach** [*] [1] [2] **Swapnil Asawa** [*] [3] **Shreyas Chaudhari** [*] [2] **Ruslan Salakhutinov** [2] **Sergey Levine** [1] [4]

## Abstract

We propose a simple, practical, and intuitive approach for domain adaptation in reinforcement learning. Our approach stems from the idea that the agent's experience in the source domain should look similar to its experience in the target domain. Building off of a probabilistic view of RL, we formally show that we can achieve this goal by *compensating for the difference in dynamics by modifying the reward function*. This modified reward function is simple to estimate by learning auxiliary classifiers that distinguish source-domain transitions from target-domain transitions. Intuitively, the modified reward function penalizes the agent for visiting states and taking actions in the source domain which are not possible in the target domain. Our approach is applicable to domains with continuous states and actions and does not require learning a model of the dynamics.

## 1. Introduction

Reinforcement learning (RL) is often touted as a promising approach for costly and risk-sensitive applications, yet learning to act in those domains directly is costly and risky. How can an intelligent agent learn to solve tasks in environments in which it cannot practice? In this paper we study the problem of domain adaptation in reinforcement learning (RL). In the context of RL, domains refer to different environments (MDPs) that have different dynamics (transition functions). Our aim is to learn a policy in the source domain that will achieve high reward in a different target domain, using a limited amount of experience from the target domain.

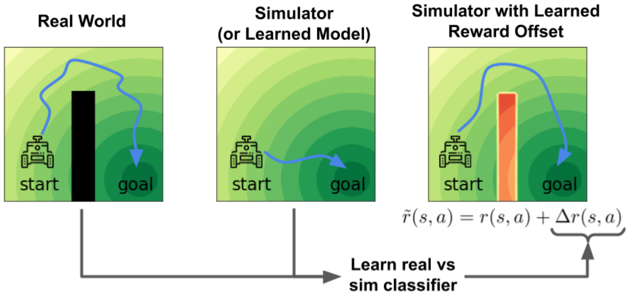

*Figure 1.* We will learn a policy for a target domain *(Left)* using experience from a source domain with different dynamics *(Center)*. *(Right)* Our method modifies the reward function to force the agent to learn behaviors that will be feasible in the target domain.

RL algorithms today require a large amount of experience in the *target domain*. Experience in the target domain is expensive to collect: it costs time (e.g., when the target domain is the real world, we cannot progress faster than real-time); it costs money (e.g., a robot might break itself); it could even be dangerous to humans (Matsakis, 2018). For many tasks, such as assistive robotics and self-driving cars, we may have access to a different but structurally similar *source domain*. While the source domain has different dynamics than the target domain, experience in the source domain is much cheaper to collect. For example, a computer simulation of the real world can run much faster than real time, collecting (say) a year of experience in an hour; it is much cheaper to simulate 1000 robot manipulators in parallel than to maintain 1000 robot manipulators. The source domain need not be a simulator, but rather could be any "practice" facility, such as a "farm" of robot arms (Levine et al., 2018), a "playpen" for learning to walk (Raibert, 2019), or a testing facility for self-driving vehicles (Madrigal, 2018).

Domain adaptation in RL is challenging because strategies which are effective in the source domain may not be effective in the target domain. For example, a good approach to driving a car around a dry racetrack (the source domain) may entail aggressive acceleration and cutting corners. If the target domain is an icy, public road, this approach may cause the car to skid off the road or hit oncoming traffic. While prior work has thoroughly studied the domain adaptation of *observations* in RL (Bousmalis et al., 2018; Ganin et al., 2016; Higgins et al., 2017), it ignores the domain adaptation of the *dynamics*. For space constraints, we defer the discussion of related work to Appendix A.

This paper presents a simple and practical approach for domain adaptation in RL, illustrated in Fig. 1. Our ap-

[*]Equal contribution [1]Google Brain [2]Carnegie Mellon University [3]University of Pittsburgh [4]UC Berkeley. Correspondence to: Benjamin Eysenbach <beysenba@cs.cmu.edu>.

*Proceedings of the 37th International Conference on Machine Learning*, Vienna, Austria, PMLR 119, 2020. Copyright 2020 by the author(s).

proach stems from the idea that the agent's experience in the source domain should look similar to its experience in the target domain. Building off of a probabilistic view of RL, we formally show that we can achieve this goal by *compensating for the difference in dynamics by modifying the reward function*. This modified reward function is simple to estimate by learning auxiliary classifiers that distinguish source-domain transitions from target-domain transitions. Because our method learns a classifier, rather than a dynamics model, we expect it to handle high-dimensional tasks better than model-based methods, a conjecture supported by experiments on the 111-dimensional Ant task. Intuitively, the modified reward function penalizes the agent for visiting states and taking actions where the source domain and target domain differ. The agent is penalized for taking transitions which would indicate whether the agent is interacting with the source or target domain. On a range of discrete and continuous control tasks, we both illustrate the mechanics of our approach and demonstrate its scalability to higher-dimensional tasks. broadly-applicable approach for learning from inaccurate models.

## 2. Preliminaries

Our problem setting will consider two MDPs: $\mathcal{M}_{\text{source}}$ represents the source domain (e.g., a practice facility, simulator, or learned approximate model of the target domain) while $\mathcal{M}_{\text{target}}$ represents a the target domain. We assume that the two domains have the same state space $\mathcal{S}$, action space $\mathcal{A}$, reward function $r$, and initially state distribution $p_1(s_1)$; the only difference between the domains is the dynamics, $p_{\text{source}}(s_{t+1} \mid s_t, a_t)$ and $p_{\text{target}}(s_{t+1} \mid s_t, a_t)$. We will learn a Markovian policy $\pi_\theta(a \mid s)$, parametrized by $\theta$. Our objective is to learn a policy $\pi$ that maximizes rewards on $\mathcal{M}_{\text{target}}$, $\mathbb{E}_{\pi, \mathcal{M}_{\text{target}}}[\sum_t \gamma^t r(s_t, a_t)]$. We now define our problem setting:

**Definition 1.** ***Domain Adaptation for RL*** *is the problem of using interactions in the source MDP $\mathcal{M}_{source}$ together with a small number of interactions in the target MDP $\mathcal{M}_{target}$ to acquire a policy that achieves high reward in the target MDP, $\mathcal{M}_{target}$.*

We will assume every transition with non-zero probability in the target domain will have non-zero probability in the source domain:

$$p_{\text{target}}(s_{t+1} \mid s_t, a_t) > 0 \implies p_{\text{source}}(s_{t+1} \mid s_t, a_t) > 0$$

for all states $s_t, s_{t+1} \in \mathcal{S}$ and actions $a_t \in \mathcal{A}$. This assumption is very weak, and common in work on importance sampling (Koller & Friedman, 2009, §12.2.2).

## 3. A Variational Perspective on Domain Adaptation in RL

The probabilistic inference interpretation of RL (Kappen, 2005; Todorov, 2007; Toussaint, 2009; Ziebart, 2010; Rawlik et al., 2013; Levine, 2018) treats the reward function as defining a desired distribution over trajectories. The agent's task is to sample from this distribution by picking trajectories with probability proportional to their exponentiated reward. This section will reinterpret this model in the context of domain transfer, showing that domain adaptation of *dynamics* can be done by modifying the *rewards*.

To apply this model to domain adaptation, define $p(\tau)$ as the desired distribution over trajectories in the target domain,

$$p(\tau) \propto p_1(s_1) \left( \prod_t p_{\text{target}}(s_{t+1} \mid s_t, a_t) \right) \exp \left( \sum_t r(s_t, a_t) \right),$$

and $q(\tau)$ as our agent's distribution over trajectories in the source domain,

$$q(\tau) = p_1(s_1) \prod_t p_{\text{source}}(s_{t+1} \mid s_t, a_t) \pi_\theta(a_t \mid s_t).$$

As noted in Section 2, we assume both trajectory distributions have the same initial state distribution. Our aim is to learn a policy whose behavior in the source domain both receives high reward and has high likelihood under the target domain dynamics. We codify this objective by minimizing the reverse KL divergence between these two distributions:

$$\min_{\pi(a|s), q(s'|s,a)} D_{\text{KL}}(q \parallel p) =$$
$$- \mathbb{E}_q \left[ \sum_t r(s_t, a_t) + \mathcal{H}_\pi[a_t \mid s_t] + \Delta r(s_{t+1}, s_t, a_t) \right] + c,$$

where

$$\Delta r(s_{t+1}, s_t, a_t) \triangleq \log p(s_{t+1} \mid s_t, a_t) - \log q(s_{t+1} \mid s_t, a_t).$$

The constant $c$ is the partition function of $p(\tau)$, which is independent of the policy and source dynamics. While $\Delta r$ is defined as the difference of transition probabilities, in Sec. 4.1 we show how to estimate $\Delta r$ without learning transition probabilities directly. In the special case where the source and target dynamics are equal, the correction term $\Delta r$ is zero and we recover maximum entropy RL (Ziebart, 2010; Todorov, 2007). We emphasize that our reward correction is different from prior work that adds $\log \beta(a \mid s)$ to the reward to regularize the policy to be close to the behavior policy $\beta$ (Schulman et al., 2015; Jaques et al., 2017; Schroecker & Isbell, 2020; Abdolmaleki et al., 2018; Jaques et al., 2019; Thananjeyan et al., 2020).

In the case where the source dynamics are *not* equal to the true dynamics, this objective is not the same as maximum entropy RL on trajectories sampled from the source domain. Instead, this objective suggests a corrective term $\Delta r$ that should be added to the reward function to account for the discrepancy between the source and target dynamics. The correction term, $\Delta r$, is quite intuitive. If a transition $(s_t, a_t, s_{t+1})$ has equal probability in the source and target domains, then $\Delta r(s_t, a_t) = 0$ so no correction is applied. For transitions that are likely in the source but are unlikely

**Algorithm 1** Domain Adaptation with Rewards from Classifiers [DARC]

```
for $t = 1, \cdots$, num iterations do
    $\mathcal{D}_{\text{source}} \leftarrow \mathcal{D}_{\text{source}} \cup \text{ROLLOUT}(\pi, \mathcal{M}_{\text{source}})$
    if $t \mod r = 0$ then
        $\mathcal{D}_{\text{target}} \leftarrow \mathcal{D}_{\text{target}} \cup \text{ROLLOUT}(\pi, \mathcal{M}_{\text{target}})$
    $\theta \leftarrow \theta - \eta \nabla_\theta \ell(\theta)$
    $\tilde{r}(s_t, a_t, s_{t+1}) \leftarrow r(s_t, a_t) + \Delta r(s_t, a_t, s_{t+1})$
    $\pi \leftarrow \text{MAXENT RL}(\pi, \mathcal{D}_{\text{source}}, \tilde{r})$
return $\pi$
```

in the target domain, $\Delta r < 0$, so the agent is penalized for "exploiting" inaccuracies or discrepancies in the source domain by taking these transitions. For the example environment in Figure 1, transitions through the center of the environment are blocked in the target domain but not in the source domain. For these transitions, $\Delta r$ would serve as a large penalty, discouraging the agent from taking these transitions and instead learning to navigate around the wall. Appendix C presents additional interpretations of $\Delta r$ in terms of coding theory, mutual information, and a constraint on the discrepancy between the source and target dynamics. Appendix B shows how prior work that adapts observation is a special case of our approach.

## 4. Domain Adaptation with Learned Rewards

The variational perspective on model-based RL in the previous section suggests that we should modify the reward in the source domain by adding $\Delta r$. While $\Delta r$ is defined above in terms of transition probabilities, we will show below how it can be estimated via binary classification, without learning an explicit dynamics model. We then use this observation to develop a practical algorithm for off-dynamics RL.

### 4.1. Estimating the Reward Correction with Classifiers

The transition probabilities in the modified reward function are rarely known and are hard to estimate. Instead, we show that we can estimate this log ratio using a pair of (learned) binary classifiers, which will infer whether transitions came from the source or target domain by rewriting $\Delta r(s_t, a_t, s_{t+1})$ as

$$\underline{\log p(\text{target} \mid s_t, a_t, s_{t+1})} - \underline{\log p(\text{target} \mid s_t, a_t)}$$

$$- \underline{\log p(\text{source} \mid s_t, a_t, s_{t+1})} + \underline{\log p(\text{source} \mid s_t, a_t)}$$

The orange terms are the difference in logits from the classifier conditioned on $s_t, a_t, s_{t+1}$, while the blue terms are the difference in logits from the classifier conditioned on just $s_t, a_t$. Intuitively, $\Delta r$ answers the following question: for the task of predicting whether a transition came from the source or target domain, how much better can you perform after observing $s_{t+1}$?

### 4.2. Algorithm Summary

Our algorithm, Domain Adaptation with Rewards from Classifiers (DARC), is presented in Alg. 1. The algorithm modifies an existing MaxEnt RL algorithm to additionally learn two classifiers, $q_{\theta_{\text{SAS}}}(\text{target} \mid s_t, a_t, s_{t+1})$ and $q_{\theta_{\text{SAS}}}(\text{target} \mid s_t, a_t)$. We use the classifiers to modify the rewards from the *source* domain, and apply MaxEnt RL to this experience. We use SAC (Haarnoja et al., 2018) as our MaxEnt RL algorithm, but emphasize that DARC is applicable to any MaxEnt RL algorithm (e.g., on-policy, off-policy, and model-based). Code has been released.[1]

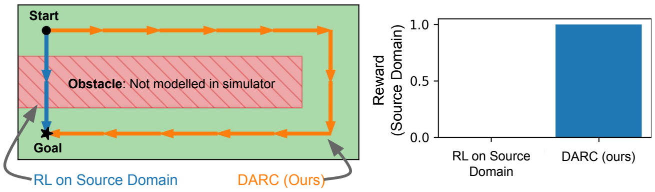

*Figure 2.* **Tabular example of off-dynamics RL**

## 5. Experiments

We start with a didactic experiment to build intuition for the mechanics of our method, and then evaluate on more complex tasks. Appendix D.3 includes additional experiments.

**Illustrative example.** We start with a simple gridworld example, shown on the right, where we can apply our method without function approximation. The goal is to navigate from the top left to the bottom left. The real environment contains an obstacle (shown in red), which is not present in the source domain. If we simply apply RL on the source domain, we obtain a policy that navigates directly to the goal (blue arrows), and will fail when used in the target domain. We then apply our method: we collect trajectories from the source domain and real world to fit the two tabular classifiers. These classifiers give us a modified reward, which we use to learn a policy in the source domain. The modified reward causes our learned policy to navigate around the obstacle.

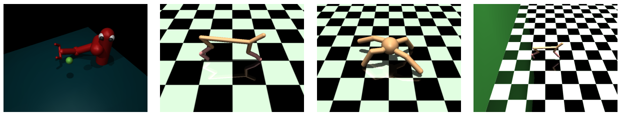

*Figure 3.* **Environments**: (L to R) broken reacher, broken half cheetah, broken ant, and half cheetah obstacle.

**Scaling to more complex tasks.** We now apply DARC to the more complex tasks shown in Fig. 3. We define three tasks by crippling one of the joints of each robot in the target domain, but using the fully-functional robot in the source domain. We use three simulated robots taken from OpenAI Gym (Brockman et al., 2016): 7 DOF reacher, half cheetah, and ant. The broken reacher is based on the task described by Vemula et al. (2020). We also include a task where the shift in dynamics is external to the robot, by modifying the

[1] https://github.com/google-research/google-research/tree/master/darc

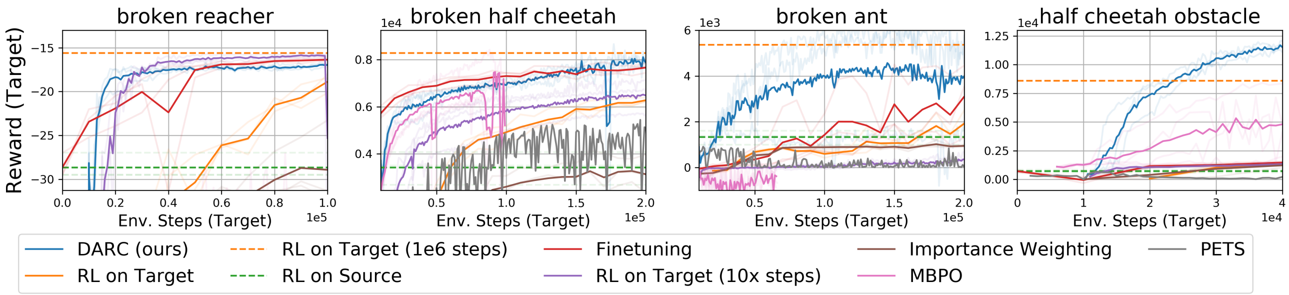

*Figure 4.* **DARC compensates for crippled robots and obstacles**: We apply DARC to four continuous control tasks: three tasks (broken reacher, half cheetah, and ant) which are crippled in the target domain but not the source domain, and one task (half cheetah obstacle) where the source domain omits the obstacle from the target domain. Note that naïvely ignoring the shift in dynamics (green dashed line) performs quite poorly, while directly learning on the crippled robot requires an order of magnitude more experience than our method.

cheetah task to reward the agent for running both forward and backwards. It is easier to learn to run backwards, but the target domain contains an obstacle that prevents the agent from running backwards.

We compare our method to seven baselines. **RL on Source** and **RL on Target** directly perform RL on the source and target domains, respectively. The **Finetuning** baseline takes the result of running RL on the source domain, and further finetunes the agent on the target domain. The **Importance Weighting** baseline performs RL on importance-weighted samples from the source domain; the importance weights are $\exp(\Delta r)$. To account for the fact that our method performs more gradient updates per environment step in the source domain, we trained a version of the RL on source baseline likewise does 10 gradient updates per source domain step. Finally, we compared against two model-based RL methods: MBPO (Janner et al., 2019) and PETS (Chua et al., 2018).

We show the results of this experiment in Fig. 4, plotting the reward on the *target* domain as a function of the number of transitions in the *target* domain. On all tasks, the RL on source baseline (shown as a dashed line because it observes no target transitions) performs considerably worse than the optimal policy from RL on the target domain, suggesting that good policies for the source domain are suboptimal for the target domain. Nonetheless, on three of the four tasks our method matches (or even surpasses) the asymptotic performance of doing RL on the target domain, despite never doing RL on experience from the target domain, and despite observing 5 - 10x less experience from the target domain. On the broken reacher and broken half cheetah tasks, we observe that finetuning on the target domain performs on par with our method. On the simpler broken reacher task, just doing RL on the target domain with a large number of gradient steps works quite well (we did not tune this parameter for our method). However, as we scale to the more complex broken ant and half cheetah obstacle tasks, we observe that all baselines perform poorly.

To gain more intuition for our method, we recorded the reward correction $\Delta r$ throughout training on the broken reacher environment. In this experiment, we ran RL on the source domain for 100k steps before switching to our method. Said another way, we ignored $\Delta r$ for the first 100k steps of training. As shown in Fig. 5, $\Delta r$ steadily decreases during these first 100k steps, suggesting that the agent is learning a strategy that takes transitions where the source domain and target domain have different dynamics: the agent is making use of its broken joint. After 100k steps, when we maximize the combination of task reward and reward correction $\Delta r$, we observe that $\Delta r$ increases, so the agent's transitions in the source domain are increasingly consistent with target domain dynamics. After around 1e6 training steps $\Delta r$ is zero: the agent has learned a strategy that uses transitions that are indistinguishable between the source and target domains.

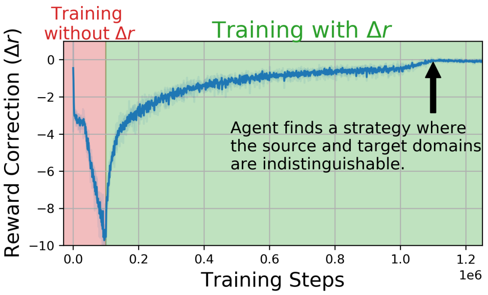

*Figure 5.* Without the reward correction, the agent takes transitions where the source domain and target domains are dissimilar; after adding the reward correction, the agent's transitions in the source domain are increasingly plausible under the target domain.

## 6. Discussion

In this paper, we proposed a simple, practical, and intuitive approach for domain adaptation to changing dynamics in RL. We formally motivate this method from a novel variational perspective on domain adaptation in RL, which suggests that we can compensate for differences in dynamics via the reward function. Experiments on a range of tasks show that our method can leverage the source domain to learn policies that work well in the target domain, despite observing only a handful of transitions from the target domain.

## References

Abdolmaleki, A., Springenberg, J. T., Tassa, Y., Munos, R., Heess, N., and Riedmiller, M. Maximum a posteriori policy optimisation. *arXiv preprint arXiv:1806.06920*, 2018.

Achiam, J., Held, D., Tamar, A., and Abbeel, P. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 22–31. JMLR. org, 2017.

Attias, H. Planning by probabilistic inference. In *AISTATS*. Citeseer, 2003.

Baktashmotlagh, M., Harandi, M. T., Lovell, B. C., and Salzmann, M. Domain adaptation on the statistical manifold. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2481–2488, 2014.

Berkenkamp, F., Turchetta, M., Schoellig, A., and Krause, A. Safe model-based reinforcement learning with stability guarantees. In *Advances in neural information processing systems*, pp. 908–918, 2017.

Bickel, S., Brückner, M., and Scheffer, T. Discriminative learning for differing training and test distributions. In *Proceedings of the 24th international conference on Machine learning*, pp. 81–88, 2007.

Bousmalis, K., Irpan, A., Wohlhart, P., Bai, Y., Kelcey, M., Kalakrishnan, M., Downs, L., Ibarz, J., Pastor, P., Konolige, K., et al. Using simulation and domain adaptation to improve efficiency of deep robotic grasping. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4243–4250. IEEE, 2018.

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

Chebotar, Y., Handa, A., Makoviychuk, V., Macklin, M., Issac, J., Ratliff, N., and Fox, D. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8973–8979. IEEE, 2019.

Chua, K., Calandra, R., McAllister, R., and Levine, S. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, pp. 4754–4765, 2018.

Clavera, I., Nagabandi, A., Fearing, R. S., Abbeel, P., Levine, S., and Finn, C. Learning to adapt: Meta-learning for model-based control. *arXiv preprint arXiv:1803.11347*, 3, 2018.

Cortes, C. and Mohri, M. Domain adaptation and sample bias correction theory and algorithm for regression. *Theoretical Computer Science*, 519:103–126, 2014.

Cutler, M., Walsh, T. J., and How, J. P. Reinforcement learning with multi-fidelity simulators. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3888–3895. IEEE, 2014.

Dann, C., Neumann, G., Peters, J., et al. Policy evaluation with temporal differences: A survey and comparison. *Journal of Machine Learning Research*, 15:809–883, 2014.

Dayan, P. and Hinton, G. E. Using expectation-maximization for reinforcement learning. *Neural Computation*, 9(2):271–278, 1997.

Deisenroth, M. and Rasmussen, C. E. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pp. 465–472, 2011.

Dudík, M., Langford, J., and Li, L. Doubly robust policy evaluation and learning. *arXiv preprint arXiv:1103.4601*, 2011.

Eysenbach, B., Gu, S., Ibarz, J., and Levine, S. Leave no trace: Learning to reset for safe and autonomous reinforcement learning. *arXiv preprint arXiv:1711.06782*, 2017.

Farchy, A., Barrett, S., MacAlpine, P., and Stone, P. Humanoid robots learning to walk faster: From the real world to simulation and back. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pp. 39–46, 2013.

Feldbaum, A. Dual control theory. i. *Avtomatika i Telemekhanika*, 21(9):1240–1249, 1960.

Fernando, B., Habrard, A., Sebban, M., and Tuytelaars, T. Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the IEEE international conference on computer vision*, pp. 2960–2967, 2013.

Finn, C. and Levine, S. Deep visual foresight for planning robot motion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2786–2793. IEEE, 2017.

Fujimoto, S., Meger, D., and Precup, D. Off-policy deep reinforcement learning without exploration. *arXiv preprint arXiv:1812.02900*, 2018.

Gamrian, S. and Goldberg, Y. Transfer learning for related reinforcement learning tasks via image-to-image translation. *arXiv preprint arXiv:1806.07377*, 2018.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

Guadarrama, S., Korattikara, A., Ramirez, O., Castro, P., Holly, E., Fishman, S., Wang, K., Gonina, E., Harris, C., Vanhoucke, V., et al. Tf-agents: A library for reinforcement learning in tensorflow, 2018.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., and Davidson, J. Learning latent dynamics for planning from pixels. *arXiv preprint arXiv:1811.04551*, 2018.

Higgins, I., Pal, A., Rusu, A., Matthey, L., Burgess, C., Pritzel, A., Botvinick, M., Blundell, C., and Lerchner, A. Darla: Improving zero-shot transfer in reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1480–1490. JMLR. org, 2017.

Hoffman, J., Wang, D., Yu, F., and Darrell, T. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint arXiv:1612.02649*, 2016.

Huszár, F. Variational inference using implicit distributions. *arXiv preprint arXiv:1702.08235*, 2017.

Janner, M., Fu, J., Zhang, M., and Levine, S. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems*, pp. 12498–12509, 2019.

Jaques, N., Gu, S., Bahdanau, D., Hernández-Lobato, J. M., Turner, R. E., and Eck, D. Sequence tutor: Conservative fine-tuning of sequence generation models with kl-control. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1645–1654. JMLR. org, 2017.

Jaques, N., Ghandeharioun, A., Shen, J. H., Ferguson, C., Lapedriza, A., Jones, N., Gu, S., and Picard, R. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456*, 2019.

Kanamori, T., Hido, S., and Sugiyama, M. A least-squares approach to direct importance estimation. *Journal of Machine Learning Research*, 10(Jul):1391–1445, 2009.

Kappen, H. J. Path integrals and symmetry breaking for optimal control theory. *Journal of statistical mechanics: theory and experiment*, 2005(11):P11011, 2005.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Koller, D. and Friedman, N. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Kouw, W. M. and Loog, M. A review of domain adaptation without target labels. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

Levine, S. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

Levine, S. and Koltun, V. Variational policy search via trajectory optimization. In *Advances in neural information processing systems*, pp. 207–215, 2013.

Levine, S., Pastor, P., Krizhevsky, A., Ibarz, J., and Quillen, D. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research*, 37(4-5):421–436, 2018.

Lipton, Z. C., Wang, Y.-X., and Smola, A. Detecting and correcting for label shift with black box predictors. *arXiv preprint arXiv:1802.03916*, 2018.

Ljung, L. System identification. *Wiley encyclopedia of electrical and electronics engineering*, pp. 1–19, 1999.

Madrigal, A. C. Waymo built a secret world for self-driving cars, Dec 2018. URL https://www.theatlantic.com/technology/archive/2017/08/inside-waymos-secret-testing-and-simulation-facilities/537648/.

Matsakis, L. Amazon has a history of bear repellent accidents, Dec 2018. URL https://www.wired.com/story/amazon-first-bear-repellent-accident/.

Mihatsch, O. and Neuneier, R. Risk-sensitive reinforcement learning. *Machine learning*, 49(2-3):267–290, 2002.

Mohamed, S. and Lakshminarayanan, B. Learning in implicit generative models. *arXiv preprint arXiv:1610.03483*, 2016.

Munos, R., Stepleton, T., Harutyunyan, A., and Bellemare, M. Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 1054–1062, 2016.

Peng, X. B., Andrychowicz, M., Zaremba, W., and Abbeel, P. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 1–8. IEEE, 2018.

Polydoros, A. S. and Nalpantidis, L. Survey of model-based reinforcement learning: Applications on robotics. *Journal of Intelligent & Robotic Systems*, 86(2):153–173, 2017.

Raibert, M. The best robots on four legs with marc raibert (boston dynamics), Apr 2019. URL https://youtu.be/tAhxi8WldCU?t=485.

Rajeswaran, A., Ghotra, S., Ravindran, B., and Levine, S. Epopt: Learning robust neural network policies using model ensembles. *arXiv preprint arXiv:1610.01283*, 2016.

Rawlik, K., Toussaint, M., and Vijayakumar, S. On stochastic optimal control and reinforcement learning by approximate inference. In *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

Ross, S. and Bagnell, J. A. Agnostic system identification for model-based reinforcement learning. *arXiv preprint arXiv:1203.1007*, 2012.

Sadeghi, F. and Levine, S. Cad2rl: Real single-image flight without a single real image. *arXiv preprint arXiv:1611.04201*, 2016.

Sastry, S. S. and Isidori, A. Adaptive control of linearizable systems. *IEEE Transactions on Automatic Control*, 34(11):1123–1131, 1989.

Schroecker, Y. and Isbell, C. Universal value density estimation for imitation learning and goal-conditioned reinforcement learning. *arXiv preprint arXiv:2002.06473*, 2020.

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.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Sønderby, C. K., Caballero, J., Theis, L., Shi, W., and Huszár, F. Amortised map inference for image super-resolution. *arXiv preprint arXiv:1610.04490*, 2016.

Song, H. F., Abdolmaleki, A., Springenberg, J. T., Clark, A., Soyer, H., Rae, J. W., Noury, S., Ahuja, A., Liu, S., Tirumala, D., et al. V-mpo: On-policy maximum a posteriori policy optimization for discrete and continuous control. *arXiv preprint arXiv:1909.12238*, 2019.

Sugiyama, M. and Müller, K.-R. Input-dependent estimation of generalization error under covariate shift. *Statistics & Decisions*, 23(4/2005):249–279, 2005a.

Sugiyama, M. and Müller, K.-R. Model selection under covariate shift. In *International Conference on Artificial Neural Networks*, pp. 235–240. Springer, 2005b.

Sugiyama, M., Krauledat, M., and MÃžller, K.-R. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(May):985–1005, 2007.

Sugiyama, M., Nakajima, S., Kashima, H., Buenau, P. V., and Kawanabe, M. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in neural information processing systems*, pp. 1433–1440, 2008.

Sugiyama, M., Takeuchi, I., Suzuki, T., Kanamori, T., Hachiya, H., and Okanohara, D. Conditional density estimation via least-squares density ratio estimation. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 781–788, 2010.

Sutton, R. S. Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4):160–163, 1991.

Tamar, A., Xu, H., and Mannor, S. Scaling up robust mdps by reinforcement learning. *arXiv preprint arXiv:1306.6189*, 2013.

Tan, J., Xie, Z., Boots, B., and Liu, C. K. Simulation-based design of dynamic controllers for humanoid balancing. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2729–2736. IEEE, 2016.

Tanaskovic, M., Fagiano, L., Smith, R., Goulart, P., and Morari, M. Adaptive model predictive control for constrained linear systems. In *2013 European Control Conference (ECC)*, pp. 382–387. IEEE, 2013.

Thananjeyan, B., Balakrishna, A., Rosolia, U., Li, F., McAllister, R., Gonzalez, J. E., Levine, S., Borrelli, F., and Goldberg, K. Safety augmented value estimation from demonstrations (saved): Safe deep model-based rl for sparse cost robotic tasks. *IEEE Robotics and Automation Letters*, 5(2):3612–3619, 2020.

Theodorou, E., Buchli, J., and Schaal, S. A generalized path integral control approach to reinforcement learning. *journal of machine learning research*, 11(Nov):3137–3181, 2010.

Tiao, L. C., Bonilla, E. V., and Ramos, F. Cycle-consistent adversarial learning as approximate bayesian inference. *arXiv preprint arXiv:1806.01771*, 2018.

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., and Abbeel, P. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 23–30. IEEE, 2017.

Todorov, E. Linearly-solvable markov decision problems. In *Advances in neural information processing systems*, pp. 1369–1376, 2007.

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

Uehara, M., Sato, I., Suzuki, M., Nakayama, K., and Matsuo, Y. Generative adversarial nets from a density ratio estimation perspective. *arXiv preprint arXiv:1610.02920*, 2016.

Vemula, A., Oza, Y., Bagnell, J. A., and Likhachev, M. Planning and execution using inaccurate models with provable guarantees. *arXiv preprint arXiv:2003.04394*, 2020.

Wang, T., Bao, X., Clavera, I., Hoang, J., Wen, Y., Langlois, E., Zhang, S., Zhang, G., Abbeel, P., and Ba, J. Benchmarking model-based reinforcement learning. *arXiv preprint arXiv:1907.02057*, 2019.

Werbos, P. J. Neural networks for control and system identification. In *Proceedings of the 28th IEEE Conference on Decision and Control,*, pp. 260–265. IEEE, 1989.

White, M. Unifying task specification in reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3742–3750. JMLR. org, 2017.

Williams, G., Aldrich, A., and Theodorou, E. Model predictive path integral control using covariance variable importance sampling. *arXiv preprint arXiv:1509.01149*, 2015.

Wittenmark, B. Adaptive dual control methods: An overview. In *Adaptive Systems in Control and Signal Processing 1995*, pp. 67–72. Elsevier, 1995.

Wulfmeier, M., Bewley, A., and Posner, I. Addressing appearance change in outdoor robotics with adversarial domain adaptation. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1551–1558. IEEE, 2017.

Yu, W., Tan, J., Liu, C. K., and Turk, G. Preparing for the unknown: Learning a universal policy with online system identification. *arXiv preprint arXiv:1702.02453*, 2017.

Yu, Y. and Szepesvári, C. Analysis of kernel mean matching under covariate shift. *arXiv preprint arXiv:1206.4650*, 2012.

Zadrozny, B. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 114, 2004.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017a.

Zhu, S., Kimmel, A., Bekris, K. E., and Boularias, A. Fast model identification via physics engines for data-efficient policy search. *arXiv preprint arXiv:1710.08893*, 2017b.

Ziebart, B. D. *Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy*. PhD thesis, Carnegie Mellon University, 2010.