ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

Multi-Fidelity Reinforcement Learning with Control Variates

Sami Khairy[1] and Prasanna Balaprakash[2] *

1- Microsoft, Vancouver, BC, Canada

2- Oak Ridge National Laboratory, Oak Ridge, TN, United States

Abstract. In this paper, we investigate reinforcement learning (RL) in multi-fidelity environments and enhance the performance of the agent using cross-correlated data. We introduce a multifidelity estimator based on control variates to reduce the variance in state-action value function estimation. By employing this estimator, we develop a multifidelity Monte Carlo RL (MFMCRL) algorithm that boosts agent learning in high-fidelity settings. Our experiments show that, given a finite high-fidelity sample budget, the MFMCRL agent outperforms an RL agent relying solely on high-fidelity interactions for policy optimization.

1 Introduction

In the computational science and engineering community, multifidelity data refers to data that is derived from diverse sources with varying fidelity levels. Low-fidelity data, which is generally less expensive to produce, serve as an approximation to high-fidelity data [1]. By leveraging cross-correlations between low- and high-fidelity data, numerous applications including black-box optimization, inference, and uncertainty propagation, can be applied to solve new problems that would otherwise be too costly to solve using only high-fidelity data [2].

This work focuses on reinforcement learning (RL) in the presence of multiple environments with different fidelity levels. Although RL has achieved great success in single-fidelity environments [3, 4], it suffers from poor sample complexity. Transfer learning (TL) [5, 6, 7] addresses this issue by transferring policies learned in low-fidelity environments to high-fidelity ones. In contrast, our work explores multifidelity estimation in RL, aiming to improve agent learning without modifying the exploration-exploitation process.

Our main contributions include: (1) investigating a generic multifidelity RL setup with low- and high-fidelity environments, (2) proposing an unbiased reduced-variance multifidelity estimator for the state-action value function using control variates, (3) introducing a multifidelity Monte Carlo RL algorithm, `MFMCRL`, to enhance agent learning in high-fidelity environments, and (4) empirically assessing the proposed `MFMCRL` algorithm's performance in synthetic and neural architecture search (NAS) multifidelity environments. An extended preprint of this work can be found on arXiv:2206.05165.

2 Related Works

[8] presents a policy search algorithm using a crude approximate model $\hat{\mathcal{P}}$ for deterministic Markov decision processes (MDPs). In asymmetric RL [9, 10], agents have

*Sami Khairy's work was conducted while he was affiliated with Argonne National Laboratory.

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

access to full state information during training but only partial observations during testing. Our work involves different MDPs for low- and high-fidelity environments with distinct state spaces, reward functions, and transition functions. Transfer learning (TL) [11, 12] uses parameters from one environment to bootstrap learning in a high-fidelity environment. Multifidelity RL (MFRL) [13, 14] extends TL by allowing agents to switch between environments and use low-fidelity value functions to guide exploration in high-fidelity environments. These approaches assume a known bound on the difference between optimal low- and high-fidelity value functions. In contrast, our work only requires correlation between low- and high-fidelity returns without knowing the correlation a priori. We leverage the cross-correlation to reduce the variance in estimating high-fidelity value functions, complementing existing TL and MFRL techniques [13, 14]. This allows agents to benefit from low-fidelity data to improve performance.

3 Multifidelity estimation in RL

3.1 Problem setup

We consider a multifidelity setup in which the RL agent has access to two environments, Σ^{lo} and Σ^{hi}, modeled by the two MDPs $\mathcal{M}^{\text{lo}} = (\mathcal{S}^{\text{lo}}, \mathcal{A}, \mathcal{P}^{\text{lo}}, \boldsymbol{\beta}^{\text{lo}}, \mathcal{R}^{\text{lo}}, \gamma)$, and $\mathcal{M}^{\text{hi}} = (\mathcal{S}^{\text{hi}}, \mathcal{A}, \mathcal{P}^{\text{hi}}, \boldsymbol{\beta}^{\text{hi}}, \mathcal{R}^{\text{hi}}, \gamma)$, respectively, as shown in Figure 1. We focus on the two-environment case for clarity, yet the proposed methods can be readily generalized to more environments. Σ^{lo} is a low-fidelity environment in which the low-fidelity reward function $\mathcal{R}^{\text{lo}}$ and the low-fidelity **stochastic** dynamics $\mathcal{P}^{\text{lo}}$ are cheap to evaluate/simulate, yet they are potentially inaccurate. On the other hand, Σ^{hi} is a high-fidelity environment in which the high-fidelity reward function $\mathcal{R}^{\text{hi}}$ and the high-fidelity **stochastic** dynamics $\mathcal{P}^{\text{hi}}$ describe the real-world system with the highest accuracy, yet they are expensive to evaluate/simulate [15]. We stress that $(\mathcal{P}^{\text{hi}}, \boldsymbol{\beta}^{\text{hi}}, \mathcal{R}^{\text{hi}})$ and $(\mathcal{P}^{\text{lo}}, \boldsymbol{\beta}^{\text{lo}}, \mathcal{R}^{\text{lo}})$ are **unknown** to the agent, and interaction with the two environments is only through the exchange of states, actions, next states and rewards.

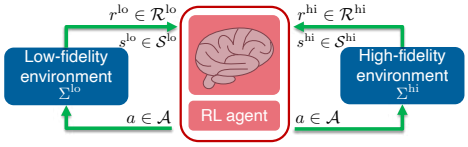


Figure 1: RL with low- and high-fidelity environments. Σ^{lo} is cheap to evaluate but is potentially inaccurate. Σ^{hi} represents the real world with the highest accuracy, yet it is expensive to evaluate.

The action space $\mathcal{A}$ is the same in both environments, yet the state space may differ. It is assumed that the low-fidelity state space is a subset of the high-fidelity state space, $\mathcal{S}^{\text{lo}} \subseteq \mathcal{S}^{\text{hi}}$, and there exists a known mapping $\mathcal{T} : \mathcal{S}^{\text{hi}} \to \mathcal{S}^{\text{lo}}$ as in previous works [11, 13]. High-fidelity environments usually capture more state information than low-fidelity environments do so $\mathcal{T}$ can be a many-to-one map.

Access to the high-fidelity simulator Σ^{hi} is restricted to full episodes $\tau^{\text{hi}} = (s_0^{\text{hi}}, a_0, r_1^{\text{hi}}, s_1^{\text{hi}}, a_1, r_2^{\text{hi}}, s_2^{\text{hi}}, \cdots, s_T^{\text{hi}})$. On the other hand, Σ^{lo} is generative, and simulation can be started at any state-action pair [16]. Using $\mathcal{T}$ and Σ^{lo}, the agent can map a τ^{hi} to $\tau^{\text{lo}} = (\mathcal{T}(s_0^{\text{hi}}), a_0, r_1^{\text{lo}}, \mathcal{T}(s_1^{\text{hi}}), a_1, r_2^{\text{lo}}, \mathcal{T}(s_2^{\text{hi}}), \cdots, \mathcal{T}(s_T^{\text{hi}}))$. In this work we study how to leverage the cheaply accessible low-fidelity trajectories from Σ^{lo}, to learn an optimal π^* with respect to the high-fidelity environment Σ^{hi}.

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

3.2 Multifidelity Monte Carlo RL

The Monte Carlo method to solve the RL problem is based on the idea of averaging sample returns. In the MC method, experience is divided into episodes. At the end of an episode, state-action values are estimated, and the policy is updated. For ease of exposition, we consider a specific state-action pair (s^{hi}, a) in what follows and suppress the dependence on (s^{hi}, a) from the notation to avoid clutter. Consider a sample trajectory τ^{hi} that results from the agent's interaction with the high-fidelity environment starting at $(s_0^{\text{hi}} = s^{\text{hi}}, a_0 = a)$ and following π, that is, $\tau^{\text{hi}} : s_0^{\text{hi}}, a_0, r_1^{\text{hi}}, s_1^{\text{hi}}, a_1, r_2^{\text{hi}}, \cdots, s_T^{\text{hi}}$. Note that $r_{t+1}^{\text{hi}} = \mathcal{R}^{\text{hi}}(s_t^{\text{hi}}, a_t)$. Let $\mathcal{G}^{\text{hi}}$ denote the corresponding long-term discounted return, $\mathcal{G}^{\text{hi}} = \sum_{t=0}^{\infty} \gamma^t r_{t+1}^{\text{hi}}$. The high-fidelity state-action value of the pair (s, a) when the agent follows π is $Q_\pi^{\text{hi}}(s^{\text{hi}}, a) = \mathbb{E}_{\tau^{\text{hi}}}\big[\mathcal{G}^{\text{hi}} | s_0^{\text{hi}} = s^{\text{hi}}, a_0 = a\big]$. Notice that $Q_\pi^{\text{hi}}(s^{\text{hi}}, a)$ is the expectation of random variable (r.v.) $\mathcal{G}^{\text{hi}}$ with respect to the random trajectory τ^{hi}.

By interacting with the environment, the agent can sample only a finite number of trajectories, n. Let $\tau_1^{\text{hi}}, \tau_2^{\text{hi}}, \cdots, \tau_n^{\text{hi}}$ be the n sampled trajectories that starts at the pair (s^{hi}, a). Furthermore, let $\mathcal{G}_1^{\text{hi}}, \mathcal{G}_2^{\text{hi}}, \cdots, \mathcal{G}_n^{\text{hi}}$ be i.i.d. r.vs. that correspond to the long-term discounted returns of the sampled trajectories, $\tau_1^{\text{hi}}, \tau_2^{\text{hi}}, \cdots, \tau_n^{\text{hi}}$, respectively. Notice that $\mathbb{E}_{\tau^{\text{hi}}}[\mathcal{G}_1^{\text{hi}}] = \mathbb{E}_{\tau^{\text{hi}}}[\mathcal{G}_2^{\text{hi}}] = \cdots = \mathbb{E}_{\tau^{\text{hi}}}[\mathcal{G}_n^{\text{hi}}] = Q_\pi^{\text{hi}}(s, a)$. The first-visit MC sample average is $\hat{Q}_{\pi,n}^{\text{hi}}(s^{\text{hi}}, a) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{G}_i^{\text{hi}}$.

Using the low-fidelity generative environment and the method of control variates, we design an unbiased estimator for the expected long-term discounted returns that has a smaller variance than the previous estimator. Let τ_i^{lo} be the ith low-fidelity trajectory that is obtained from τ_i^{hi} by using $\mathcal{T}$ and the generative low-fidelity environment to evaluate $r_{t+1}^{\text{low}} = \mathcal{R}^{\text{lo}}(\mathcal{T}(s_t^{\text{hi}}), a_t)$. Let $\mathcal{G}_i^{\text{lo}}$ be the r.v. which corresponds to the long-term discounted return of τ_i^{lo}. Notice that $\mathcal{G}_i^{\text{hi}}$ and $\mathcal{G}_i^{\text{lo}}$ are correlated r.vs since they come from the same realization of the random process defined over the state-action pairs in this multifidelity setup. Based on those low-fidelity trajectories, the low-fidelity first-visit MC sample average is $\hat{Q}_{\pi,n}^{\text{lo}}(\mathcal{T}(s^{\text{hi}}), a) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{G}_i^{\text{lo}}$ and has a variance of $\text{Var}\Big[\hat{Q}_{\pi,n}^{\text{lo}}(\mathcal{T}(s^{\text{hi}}), a)\Big] = \frac{\sigma_{\text{lo}}^2(\mathcal{T}(s^{\text{hi}}), a)}{n}$, where $\sigma_{\text{lo}}^2(\mathcal{T}(s^{\text{hi}}), a) = \mathbb{E}_{\tau^{\text{lo}}}\Big[\big(\mathcal{G}^{\text{lo}} - Q_\pi^{\text{lo}}(\mathcal{T}(s^{\text{hi}}), a)\big)^2 | s_0 = \mathcal{T}(s^{\text{hi}}), a_0 = a\Big]$ and $Q_\pi^{\text{lo}}(\mathcal{T}(s^{\text{hi}}), a)$ is the true population mean.

Using the method of control variates, we propose the following multifidelity MC estimator:

$$\hat{Q}_{\pi,n}^{\text{MFMC}}(s^{\text{hi}}, a) = \hat{Q}_{\pi,n}^{\text{hi}}(s^{\text{hi}}, a) + \alpha_{s,a}^* \bigg(Q_\pi^{\text{lo}}(\mathcal{T}(s^{\text{hi}}), a) - \hat{Q}_{\pi,n}^{\text{lo}}(\mathcal{T}(s^{\text{hi}}), a) \bigg), \tag{1}$$

where

$$\alpha_{s,a}^* = \frac{\text{Cov}\big[\hat{Q}_{\pi,n}^{\text{hi}}(s^{\text{hi}}, a), \hat{Q}_{\pi,n}^{\text{lo}}(\mathcal{T}(s^{\text{hi}}), a)\big]}{\text{Var}\big[\hat{Q}_{\pi,n}^{\text{lo}}(\mathcal{T}(s^{\text{hi}}), a)\big]}. \tag{2}$$

This is unbiased and has a variance of $\text{Var}\Big[\hat{Q}_{\pi,n}^{\text{MFMC}}(s^{\text{hi}}, a)\Big] = \big(1 - \rho_{s,a}^2\big)\text{Var}\Big[\hat{Q}_{\pi,n}^{\text{hi}}(s^{\text{hi}}, a)\Big]$, where $\rho_{s,a}$ is the correlation coefficient between the low-fidelity and high-fidelity long-term discounted returns. Therefore, the variance in estimating the value of a state-action pair under a policy π can be reduced by a factor of $\big(1 - \rho_{s,a}^2\big)$ when the low-fidelity data is exploited, although the budget of high-fidelity samples remains the same.

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

Notice that $\mathrm{Cov}\big[\hat{Q}^{\mathrm{hi}}_{\pi,n}(s^{\mathrm{hi}},a),\hat{Q}^{\mathrm{lo}}_{\pi,n}(\mathcal{T}(s^{\mathrm{hi}}),a)\big] = \mathrm{Cov}\big[\frac{1}{n}\sum_{i=1}^{n}\mathcal{G}^{\mathrm{hi}}_i,\frac{1}{n}\sum_{i=1}^{n}\mathcal{G}^{\mathrm{lo}}_i\big] = \frac{1}{n}\mathrm{Cov}\big[\mathcal{G}^{\mathrm{hi}}_i,\mathcal{G}^{\mathrm{lo}}_i\big]$, because $\mathcal{G}^{\mathrm{hi}}_i,\mathcal{G}^{\mathrm{lo}}_j$ are independent r.vs. $\forall i\neq j$. Hence, $\mathrm{Cov}\big[\hat{Q}^{\mathrm{hi}}_{\pi,n}(s^{\mathrm{hi}},a),\hat{Q}^{\mathrm{lo}}_{\pi,n}(\mathcal{T}(s^{\mathrm{hi}}),a)\big]$, $\mathrm{Var}\Big[\hat{Q}^{\mathrm{hi}}_{\pi,n}(s^{\mathrm{hi}},a)\Big]$, and $\mathrm{Var}\Big[\hat{Q}^{\mathrm{lo}}_{\pi,n}(\mathcal{T}(s^{\mathrm{hi}}),a)\Big]$ can all be estimated in practice based on the return data samples using the standard unbiased estimators for the variance and covariance.

The reduced-variance estimator of (1) can be used to design a multifidelity Monte Carlo RL algorithm (`MFMCRL`). This algorithm is based on the on-policy first-visit MC control algorithm with ϵ-soft policies [17] but uses the multifidelity estimator (1). In the policy evaluation step the state-action value function is made consistent with the current policy by updating the estimated long-term discounted returns of a state-action pair (s_t,a_t) using the control-variate-based estimator (1). Next, in the policy improvement step, the policy is made ϵ-greedy with respect to the current state-action value function. In each episode, the agent needs to evaluate the policy in the low-fidelity environment to obtain Q^{lo}_{π}. This can be done in practice by collecting a large number of m return samples from the cheap low-fidelity environment and setting $Q^{\mathrm{lo}}_{\pi}(\mathcal{T}(s^{\mathrm{hi}}),a)\approx\hat{Q}^{\mathrm{lo}}_{\pi,m+n}(\mathcal{T}(s^{\mathrm{hi}}),a)$.

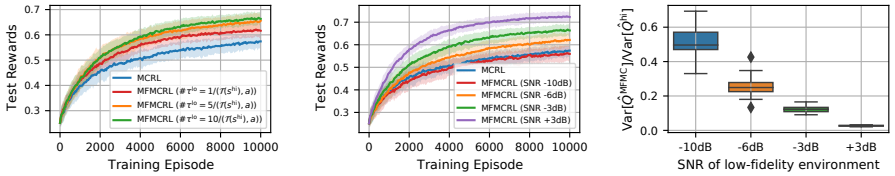
4 Numerical experiments

In this section we empirically evaluate the performance of the proposed `MFMCRL` algorithm on synthetic MDP problems and on a NAS use case.

4.1 Synthetic MDPs

We generate random MDP problems with state space cardinality $|\mathcal{S}|$ and action space cardinality $|\mathcal{A}|$. High-fidelity transition and reward functions, $\mathcal{P}^{\mathrm{hi}}$ and $\mathcal{R}^{\mathrm{hi}}$, are created based on a random process. The corresponding low-fidelity functions, $\mathcal{P}^{\mathrm{low}}$ and $\mathcal{R}^{\mathrm{low}}$, are created by injecting Gaussian noise to meet a desired signal-to-noise (SNR) ratio. Notice that even with infinite samples from the low-fidelity environment, the agent cannot recover the high-fidelity functions, as the low-fidelity functions are corrupted. This reflects real-world scenarios when learning low-fidelity functions from data to train RL agents. We encapsulate the high- and low-fidelity functions in gym-like environments, forming separate high-fidelity and low-fidelity environments. We train an RL agent using the proposed `MFMCRL` algorithm over 10K high-fidelity episodes and compare it to another RL agent (`MCRL`) trained using the standard first-visit MC control algorithm over 10K high-fidelity episodes. We set γ and ϵ to 0.99 and 0.1, respectively. Performance is tested on 200 test episodes every 50 training episodes. We repeat the whole experiment with 36 different random seeds and report the mean and standard deviation of the test episode rewards in Figure 2(a). The proposed `MFMCRL` algorithm outperforms `MCRL` in policy performance, improving as the RL agent collects more low-fidelity samples. In Figure 2(b), we observe that performance improves as SNR increases, with no benefit from multifidelity RL at -10 dB SNR due to weak correlation between low- and high-fidelity environments. In Figure 2(c), we show the mean variance reduction factor, with more variance reduction attained when the low-fidelity environment is less noisy.

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.



(a) Σ^{lo} has an SNR of -3dB (b) $\#\tau^{\text{lo}} = 10/(\mathcal{T}(s^{\text{hi}}), a))$ (c) Variance reduction factor

Figure 2: Test episode rewards for the proposed MFMCRL during training: (a) test episode rewards improve with increasing number of low-fidelity samples ($\#\tau^{\text{lo}}$); (b) test episode rewards improve with less noisy low-fidelity environments; (c) variance reduction factor improves when low- and high-fidelity environments are more correlated.

4.2 NAS

In the Neural Architecture Search (NAS) experiment, we study how multifidelity RL can improve learning in NAS over standard RL. We use the NAS-Bench-201 dataset [18] to construct multifidelity RL environments, and the RL agent is tasked with sequentially configuring nodes of an architecture to maximize total rewards and discover high-performing architectures. Two multifidelity scenarios are constructed with high-fidelity rewards being the validation accuracy at 200 epochs. For low-fidelity environments, we have two cases: (i) validation accuracy at the 10th epoch, and (ii) a smaller search space with validation accuracy at the 10th epoch. We train the proposed MFMCRL and MCRL and report the mean and standard deviation of test episode rewards in Figure 3. Our multifidelity RL framework improves over standard RL, with higher performance gains when low- and high-fidelity environments are more similar (case i).

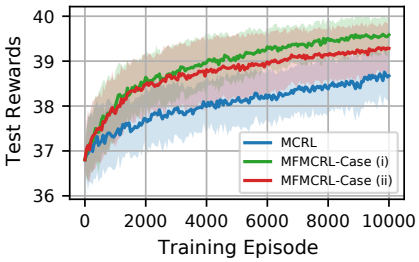


Figure 3: Test episode rewards for the proposed MFMCRL during training on multifidelity NAS environments ($\#\tau^{\text{lo}} = 5/(\mathcal{T}(s^{\text{hi}}), a))$).

5 Conclusion

This paper investigates the RL problem in the presence of low- and high-fidelity environments, aiming to improve agent performance using multifidelity data. We propose a control variates-based multifidelity estimator to reduce variance in state-action value function estimation. A multifidelity Monte Carlo RL algorithm (MFMCRL) is introduced, and empirical evaluations demonstrate that, with a limited budget of high-fidelity data, MFMCRL effectively leverages cross-correlations between low- and high-fidelity data, resulting in superior performance. Future work will explore a control-

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

variate-based multifidelity RL framework with function approximation for continuous state-action space RL problems.

References

[1] Xuhui Meng and George Em Karniadakis. A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse PDE problems. *Journal of Computational Physics*, 401:109020, 2020.

[2] Benjamin Peherstorfer, Karen Willcox, and Max Gunzburger. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. *Siam Review*, 60(3):550–591, 2018.

[3] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on Machine Learning*, pages 1889–1897. PMLR, 2015.

[4] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[5] Matthew E Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(7), 2009.

[6] Yevgen Chebotar, Ankur Handa, Viktor Makoviychuk, Miles Macklin, Jan Issac, Nathan Ratliff, and Dieter Fox. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8973–8979. IEEE, 2019.

[7] Zhuangdi Zhu, Kaixiang Lin, and Jiayu Zhou. Transfer learning in deep reinforcement learning: A survey. *arXiv preprint arXiv:2009.07888*, 2020.

[8] Pieter Abbeel, Morgan Quigley, and Andrew Y Ng. Using inaccurate models in reinforcement learning. In *Proceedings of the 23rd international conference on Machine Learning*, pages 1–8, 2006.

[9] Andrew Warrington, Jonathan W Lavington, Adam Scibior, Mark Schmidt, and Frank Wood. Robust asymmetric learning in pomdps. In *International Conference on Machine Learning*, pages 11013–11023. PMLR, 2021.

[10] Lerrel Pinto, Marcin Andrychowicz, Peter Welinder, Wojciech Zaremba, and Pieter Abbeel. Asymmetric actor critic for image-based robot learning. *arXiv preprint arXiv:1710.06542*, 2017.

[11] Matthew E Taylor, Peter Stone, and Yaxin Liu. Transfer learning via inter-task mappings for temporal difference learning. *Journal of Machine Learning Research*, 8(9), 2007.

[12] Timothy A Mann and Yoonsuck Choe. Directed exploration in reinforcement learning with transferred knowledge. In *European Workshop on Reinforcement Learning*, pages 59–76. PMLR, 2013.

[13] Mark Cutler, Thomas J Walsh, and Jonathan P How. Real-world reinforcement learning via multifidelity simulators. *IEEE Transactions on Robotics*, 31(3):655–671, 2015.

[14] Varun Suryan, Nahush Gondhalekar, and Pratap Tokekar. Multifidelity reinforcement learning with Gaussian processes: model-based and model-free algorithms. *IEEE Robotics & Automation Magazine*, 27(2):117–128, 2020.

[15] M Giselle Fernández-Godino, Chanyoung Park, Nam-Ho Kim, and Raphael T Haftka. Review of multi-fidelity models. *arXiv preprint arXiv:1609.07196*, 2016.

[16] Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *In Proc. 19th International Conference on Machine Learning*. Citeseer, 2002.

[17] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT Press, 2018.

[18] Xuanyi Dong and Yi Yang. NAS-Bench-201: Extending the scope of reproducible neural architecture search. *arXiv preprint arXiv:2001.00326*, 2020.