A Benchmark for Hierarchical Parameterized Action Markov Decision Process

Dengxian Yang[1], Neil M. Dundon[2], Elizabeth Rizor[2], Scott Grafton[2], Linda Petzold[1]

[1]Department of Computer Science, University of California, Santa Barbara, CA 93106, USA
[2]Department of Psychological and Brain Sciences, University of California, Santa Barbara, CA 93106, USA
dengxian_yang, dundonnm, elizabethrizor, stgrafton, petzold@ucsb.edu

Abstract

Discrete-continuous action combinations are prevalent in everyday activities and play a crucial role in robotic control. Hierarchical decision-making, a vital element of both reinforcement learning and daily life, has been explored through recent studies that investigate these actions within a hierarchical framework and propose various on-policy learning algorithms. However, there remains a notable gap in the literature: no task has yet been designed that integrates hierarchical decision-making with discrete-continuous action combinations. Additionally, the scarcity of data in this domain hinders the development of offline reinforcement learning algorithms.

In this work we introduce a novel task that serves as a comprehensive platform for investigating planning and action controls within a Hierarchical Parameterized Action Markov Decision Process (HPAMDP). This task incorporates elements of spatial planning, fine action control, discrete-continuous action integration, hierarchical planning, and robust reinforcement learning under noisy conditions, creating a rich environment conducive to research across these disciplines. We also provide valuable human data to support this initiative and encourage the community to contribute further, facilitating the advancement of off-policy algorithms. This contribution aims to bridge critical gaps and catalyze future research in this interdisciplinary field.

code and dataset —
https://github.com/DaraYang/Boatdock_v1

Introduction

Recent advancements in reinforcement learning (RL) have transformed the field, enabling the resolution of complex decision-making tasks and demonstrating remarkable performance across various domains such as Atari games (Mnih et al. 2015), robotic control (Schulman et al. 2018; Lillicrap et al. 2019), and autonomous navigation (Kiran et al. 2021). Traditional benchmarks and tasks in RL development have typically focused on either purely discrete action spaces, as seen in Atari games, or continuous action spaces, such as those used in MuJoCo environments. However, many real-world tasks often require planning over a long horizon such as the maze (Botvinick, Niv, and Barto 2009), and intricate control of hybrid discrete and continuous actions, such as robot soccer (Masson, Ranchod, and Konidaris 2016) and MOBA games (Xiong et al. 2018). To address the challenge of hybrid actions over a long horizon, several paradigms have been explored, including Hierarchical Reward Machine (HRM), Task and Motion Planning (TAMP), Hierarchical Reinforcement Learning (HRL) and Parameterized Action Markov Decision Problems (PAMDP) (Parr and Russell 1997; Guo et al. 2023; Masson, Ranchod, and Konidaris 2016).

TAMP and HRM both have hierarchical structures like HRL that decompose tasks into subgoals or hierarchy of controls to maximize cumulative rewards over extended periods. TAMP is widely used in robotic control that focuses on learning planning-execution pairs while satisfying both physical and designed constraints (Guo et al. 2023; Lagriffoul et al. 2018). HRM decomposes tasks into subtasks with separate reward functions and encode objectives as hierarchical finite state machines. It is particularly useful in hybrid action scenarios where discrete and continuous actions operate at different hierarchies and is widely used in applications such as pathfinding in a pre-defined map with finite states (Cong, Liu, and Liu 2023; Furelos-Blanco et al. 2023). Additionally, HRM can be used as an extension to RL algorithms to increase the sample efficiency (Icarte et al. 2022), though classical HRM often requires pre-defind and labeled options (Parr and Russell 1997; Furelos-Blanco et al. 2021). To achieve an end-to-end algorithm in TAMP or HRM, hierarchies of planning, subgoal identification, and hierarchical control for hybrid action spaces are often required (Lo, Zhang, and Stone 2018). Both paradigms also face challenges in online planning and learning tasks with infinite possible states. In this paper, we focus on PAMDP as a robust and scalable approach to solving hybrid action space in an online learning framework without the need of manually defined options. Among many existing benchmark tasks for hierarchical reinforcement learning, most of them involve planning consisting of multiple steps, such as the maze, towers of Hanoi, the kitchen, or a sequence of subtasks, for example, delivering a piece and mail and also picking up a cup of coffee in an office map. However, to the best of our knowledge, there is no hierarchical task that requires the agent to choose the preferred media to perform a task. For example, suppose you have two ways of transportation:

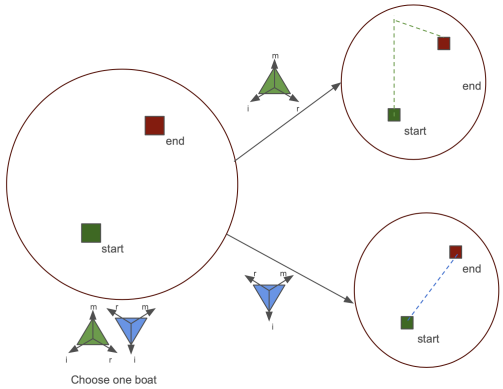

Figure 1: The boat task presents the participant with start and end placement, and asks the participant to choose one controller out of the two. The direction each controller can go and the corresponding key mappings (r, m, i refer to the right-hand ring, middle, and index finger on the m, h, and v keys on the keyboard respectively) are shown here for clarity, but are hidden for participants. Participants or agents need to sail the boat with the chosen controller from the start to the goal with limited gas, without exceeding the circle boundary.

driving or taking the bus. Depending on your skill of driving, bus route and schedule, parking availability, time, and fuel resource cost, you may choose different vehicles when needed to move from one place to another. In order to maximize long-term returns, choosing the right tool to solve the problem under different circumstances is another important aspect of the hierarchy an agent should learn.

When designing RL agents, they are often characterized as model-based (Heess et al. 2015; Clavera et al. 2018) or model-free (Mnih et al. 2015; Fujimoto, Hoof, and Meger 2018) depending on what function they are approximating. While the dichotomy between model-based and model-free algorithms has provided a foundational framework for research in human behavior, neuroscience, and reinforcement learning models, the possibility of frameworks extending beyond simple dichotomy has been widely discussed in the community (Collins and Cockburn 2020; O'Doherty et al. 2021). These discussions are particularly relevant as tasks grow more complex, involving hierarchical structures and a mixture of model-based and model-free paradigms (Botvinick, Niv, and Barto 2009; Lepora and Pezzulo 2015). In order to further the development of agents that are able to solve more and more complicated problems and achieve long-term learning, we are introducing a new benchmark that: (1) demands effective spatial and temporal planning with hybrid discrete and continuous actions from algorithms; (2) incorporates options for hierarchical decision-making to optimize long-horizon returns; (3) includes two sets of controllers to test models' generalizability and (4) provides a dataset from human players to foster the development of offline PAMDP algorithmic solutions.

Background and Related Work

A Markov Decision Process (MDP) is a mathematical framework for modeling decision-making in environments with stochastic outcomes (Sutton 1990). Formally, an MDP is defined as a tuple $\langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$, where $\mathcal{S}$ represents the set of states, $\mathcal{A}$ is the set of actions, $P(s'|s,a)$ is the transition probability from state $s \in \mathcal{S}$ to state $s' \in \mathcal{S}$ when action $a \in \mathcal{A}$ is taken, $R(s,a)$ is the reward function that specifies the immediate reward for taking action a in state s, and $\gamma \in [0,1)$ is the discount factor that controls the importance of future rewards. The goal in an MDP is to find a policy $\pi(a|s)$, which specifies the probability of taking action a in state s, to maximize the expected cumulative reward over time.

A Parameterized Action Markov Decision Process (PAMDP) extends the traditional MDP framework by incorporating parameterized actions, which combine discrete actions with associated continuous parameters. Formally, a PAMDP is defined as a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{X}, P, R, \gamma \rangle$. Here, $\mathcal{S}$ represents the set of states, $\mathcal{A}$ is a finite set of discrete actions, and for each discrete action a, there is a continuous parameter $x_a \subseteq R^{m_a}$. Then the action space can be written as $\bigcup_{a \in A_d} \{(a,x) \mid x \in X_a\}$. Parameterized actions enable tasks with more fine-grained control to maximize expected cumulative reward over time.

To tackle challenges in hybrid action spaces, several algorithmic approaches have been developed. A common strategy involves converting the heterogeneous action space into a homogeneous one, either by discretizing the continuous actions or by using softmax or argmax to select discrete actions (Bester, James, and Konidaris 2019a; Baumann et al. 2018). Although discretization allows for the handling of continuous actions, it suffers from scalability issues due to the exponential increase in the number of discrete actions. Conversely, representing all discrete actions as continuous can complicate the mapping of actions and reduce policy learning efficiency. An alternative approach treats PAMDPs as a hierarchical problem, selecting a discrete action first and subsequently choosing a continuous action using another model (Masson, Ranchod, and Konidaris 2016; Xiong et al. 2018; Fan et al. 2019; Wei, Wicke, and Luke 2018). Some research has explored simultaneous handling of discrete and continuous actions by learning conditional latent embeddings (Li et al. 2022) or learning a joint distribution (Neunert et al. 2020). Another recent work first used a model-based technique to solve PADMPs (Zhang et al. 2024). Despite these advances, a significant gap remains in benchmarks including hierarchical learning and spatial and temporal planning over a long horizon (Bacon, Harb, and Precup 2017; Zhang and Whiteson 2019). Existing platforms such as the 1D action space environment (Masson, Ranchod, and Konidaris 2016; Bester, James, and Konidaris 2019b) lack 2D spatial planning capabilities. Environments like the Goal (Masson, Ranchod, and Konidaris 2016) and HardMove (Li et al. 2022) require spatial planning to move objects to target areas, however, they do not incorporate time or resource constraints, limiting the need for optimized action control in the parameterized action space.

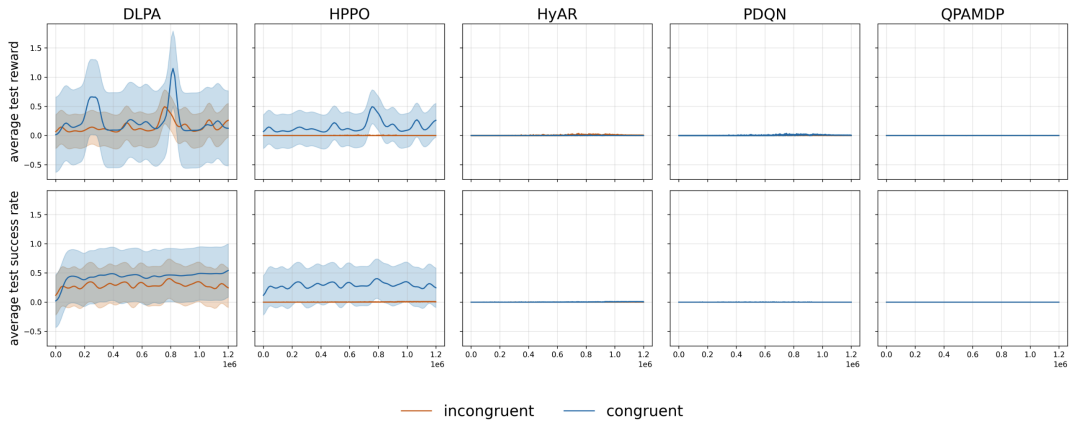

Figure 2: Comparison of algorithms on the Boat environment. The x-axis denotes the number of steps. The first row shows the average test reward over 100 episodes and the second row shows the average test success rate over 100 episodes. The curve and shade denote the mean and one standard deviation over 3 random seeds.

The Boat Task and Dataset

In this work we implemented a human-subject task designed in our previous work (Dundon et al. 2023) in the frame of hierarchical PAMDP, in the OpenAI Gym environment. This task includes a hybrid discrete and continuous action extended from the discrete grid-sail task (Fermin et al. 2016). As shown in Figure1, at the beginning of each episode, the initial state of this episode is presented and the participants must choose between 2 controllers with different underlying directions. There is always one controller whose key mappings are intuitive (congruent boat) and the other with counter-intuitive key mappings (incongruent boat). For example, in Figure1, the green boat is congruent and the blue boat is incongruent. The directions a controller can go are not explicitly shown to participants. They need to learn through trial and error. No indications of total gas tank capacity, or gas usage are available to participants during an episode. At the end of the episode, the participant is presented with an outcome page showing one of the following outcomes: Out of gas, reward = 0, Out of boundary, reward = 0, Approached the target too fast: reward = 0, Success! reward = percentage of remaining gas * 100. Each session contains 6 blocks of 80 episodes. The monetary reward participants receive at the end of the experiment is positively correlated with their overall performance. Therefore, participants are incentivized to improve their fine action control as well as to choose a strategy to optimize their return in the long horizon. A full tank is at most 6 seconds of acceleration time. Given the nature of human control in the presence of delay and noise, on average a human can press buttons 5.45 ± 3.57 times in an episode.

The Gym Environment

We implemented the corresponding task within the OpenAI Gym environment, incorporating flexible design features to allow for hierarchical decision-making. When the hierarchical option is enabled, an agent must choose between two controllers after observing the initial state of each episode. Conversely, when the hierarchical option is disabled, the algorithm can be configured to use only a congruent boat, only an incongruent boat, or a randomly selected boat. Once a controller is chosen, the agent must navigate the boat to the goal using the corresponding buttons.

In our environment, only one button can be pressed at a time. If multiple buttons are pressed simultaneously, the action is ignored. In human-subject experiments, the action executed first is taken into account. Consistent with the human task, an acceleration coefficient is implemented such that the boat accelerates non-linearly as the duration of a button press increases. This design requires both agents and human participants to learn the temporal dynamics and integrate them with spatial cues. Different from the HardMove task introduced by (Li et al. 2022), which required agents to choose among n actuators to toggle on or off and directly decide displacement for each actuator, our task requires agents to learn the underlying mapping between continuous parameters and movement outcomes. Additionally, our task imposes stricter spatial and temporal constraints. Spatial constraints include limited directional options and boundaries that must not be crossed, while temporal constraints arise from the limited fuel supply and acceleration sensitivity. These constraints aim to push agents toward learning optimal behavior under challenging conditions.

Due to the inherent difficulty of the task, reward signals are very sparse. Initial experiments provided rewards only at the end of each episode, with no intermediate feedback to best align with the human subject experiments. Under these conditions, all algorithms failed to learn the task. To mitigate this issue, we introduced a modified reward structure. Agents now receive a small reward proportional to their spatial proximity to the target, multiplied by a scaling factor. A large reward (scaled between 0 and 5) is provided for a successful outcome, and an award of -1 is

provided for failures. This adjustment stabilized training by scaling the original reward range from [0, 100] to [-1, 5]. Each observation in the environment consists of: start and goal (x,y) position, the boundary position represented by the center of the circle and the radius, the current boat choice(0 being incongruent and 1 congruent), the current (x,y) position of the boat, current velocity of the boat, current fuel level(percentage), and current proximity to the goal(percentage).

Data Collection and Processing

A total of 53 right-handed human participants were recruited to participate in this single-session "boatdock" task via both word-of-mouth and the online participant recruitment portal at the University of California, Santa Barbara (UCSB). There were 34 reported female and 19 reported male participants with an average age of 21.9 ± 3.05. Experiments are performed under the relevant guidelines and regulations required and approved by The Institutional Review Board at UCSB (Human Subject Committee protocol:36-21-0405). All procedures were performed in accordance with the Declaration of Helsinki and written consents were obtained from all participants before participation (Dundon et al. 2023).

Experiments

All of the experiments were performed on a single NVIDIA GeForce RTX 4060 Ti GPU. Experiments ranged from 4 hours to 58 hours, depending on the algorithm and implementation. All of the results shown are averaged across 3 random seeds.

In this work, we tested five representative algorithms, summarized below, each embodying a distinct approach to PAMDP. Because there have not been any algorithms that handle hierarchical PAMDP problems, we turned off the boat-selection and trained separate agents for the congruent and incongruent boats.

- QPAMDP (Masson, Ranchod, and Konidaris 2016) alternately learns a value function for discrete actions and a policy function for continuous actions.
- PDQN (Xiong et al. 2018) extends on DQN (Mnih et al. 2015) and DDPG (Lillicrap et al. 2016) to first train the deterministic function that maps states and discrete actions to the continuous parameters and then a value function is fitted to predict the expected reward given the tuple of state and action pairs.
- HPPO (Fan et al. 2019) extends the PPO (Schulman et al. 2017) structure. Two actors with a shared feature extractor of the state space are built to learn the discrete and continuous part of action space separately, and then one critic learns the value function of the current state.
- HyAR (Li et al. 2022) keeps the discrete action in the embedding table, and uses a variational autoencoder (Kingma 2013) conditioned on state s and discrete action a_d to generate the continuous parameter x_a, and then TD3 (Fujimoto, Hoof, and Meger 2018) is used to learn the policy network. This approach takes the discrete action into consideration when generating continuous parameters.
- DLPA (Zhang et al. 2024) is the first model that uses a model-based approach, with inspiration from model predictive control (Garcia, Prett, and Morari 1989). It trains the transition model with H-step loss and designs two reward predictors for return prediction and termination prediction respectively.

As shown in Table1 and Figure2, humans overall have the best performance compared to the algorithms. Among the algorithms, DLPA achieves significantly higher performance in both average reward and success rate. For model-free algorithms, HPPO was able to learn to navigate the congruent boat, while HyAR and PDQN demonstrated only sporadic success. Almost all of the algorithms failed the task when using the incongruent boat. Although DLPA achieved a higher success rate with the incongruent boat compared to other methods, its performance remained substantially lower than with the congruent boat. This finding is particularly intriguing, as it suggested that incongruent key mapping typically thought to increase cost and learning curve for humans, may also influence reinforcement learning agents. Does this imply that the neural network commonly used in these algorithms possesses an inherent bias toward certain default action mappings? Exploring this hypothesis further is planned in our future research.

	congruent	incongruent
human	3.0163 $\pm$0.87	2.9318 $\pm$0.94
DLPA	0.2441 $\pm$0.64	0.1484 $\pm$0.29
HyAR	0.005 $\pm$0.0005	-1.3623 $\pm$0.223
HPPO	0.6720 $\pm$0.16	0.1448 $\pm$0.58
PDQN	0.0073 $\pm$0.003	-1.73 $\pm$0.0002
QPAMDP	-1.6758 $\pm$0.0002	-1.7410 $\pm$0.0001

Table 1: Comparison of average episodic reward on human dataset and algorithms. Mean and standard deviation of evaluation of 1000 episodes after training are reported for algorithms, and human performance is averaged across all stages of learning.

Conclusion and Future Work

This work introduces a novel task for hierarchical PAMDP problems, integrating spatial and temporal planning with hierarchical decision-making. Through evaluations using state-of-the-art algorithms for PAMDP problems, we observed that PAMDP algorithms are not yet capable of achieving human-level performance, even when considering only the action control aspect. Additionally, we identified that challenges related to dexterity, which are difficult for humans, also pose significant obstacles for machine learning agents. We look forward to receiving feedback from the community and aim to expand the dataset to facilitate further research.

Acknowledgement

This work is supported by the UCSB Institute for Collaborative Biotechnologies under Cooperative Agreement W911NF-19-2-0026 from the ArmyResearch Office.

References

Bacon, P.-L.; Harb, J.; and Precup, D. 2017. The option-critic architecture. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Baumann, D.; Zhu, J.-J.; Martius, G.; and Trimpe, S. 2018. Deep reinforcement learning for event-triggered control. In *2018 IEEE Conference on Decision and Control (CDC)*, 943–950. IEEE.

Bester, C. J.; James, S. D.; and Konidaris, G. D. 2019a. Multi-pass q-networks for deep reinforcement learning with parameterised action spaces. *ArXiv preprint ArXiv:1905.04388*.

Bester, C. J.; James, S. D.; and Konidaris, G. D. 2019b. Multi-Pass Q-Networks for Deep Reinforcement Learning with Parameterised Action Spaces. *ArXiv preprint ArXiv:1905.04388*.

Botvinick, M. M.; Niv, Y.; and Barto, A. G. 2009. Hierarchically organized behavior and its neural foundations: A reinforcement learning perspective. *Cognition*, 113(3): 262–280.

Clavera, I.; Rothfuss, J.; Schulman, J.; Fujita, Y.; Asfour, T.; and Abbeel, P. 2018. Model-based reinforcement learning via meta-policy optimization. In *Conference on Robot Learning*, 617–629. PMLR.

Collins, A. G.; and Cockburn, J. 2020. Beyond dichotomies in reinforcement learning. *Nature Reviews Neuroscience*, 21(10): 576–586.

Cong, J.; Liu, Y.; and Liu, C. 2023. Guiding Task Learning by Hierarchical RL with an Experience Replay Mechanism Through Reward Machines. In *Pacific Rim International Conference on Artificial Intelligence*, 164–170. Springer.

Dundon, N. M.; Colas, J. T.; Garrett, N.; Babenko, V.; Rizor, E.; Yang, D.; MacNamara, M.; Petzold, L.; and Grafton, S. T. 2023. Decision heuristics in contexts integrating action selection and execution. *Scientific Reports*, 13(1): 6486.

Fan, Z.; Su, R.; Zhang, W.; and Yu, Y. 2019. Hybrid Actor-Critic Reinforcement Learning in Parameterized Action Space. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, 2279–2285. International Joint Conferences on Artificial Intelligence Organization.

Fermin, A. S.; Yoshida, T.; Yoshimoto, J.; Ito, M.; Tanaka, S. C.; and Doya, K. 2016. Model-based action planning involves cortico-cerebellar and basal ganglia networks. *Scientific Reports*, 6(1): 31378.

Fujimoto, S.; Hoof, H.; and Meger, D. 2018. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, 1587–1596. PMLR.

Furelos-Blanco, D.; Law, M.; Jonsson, A.; Broda, K.; and Russo, A. 2021. Induction and exploitation of subgoal automata for reinforcement learning. *Journal of Artificial Intelligence Research*, 70: 1031–1116.

Furelos-Blanco, D.; Law, M.; Jonsson, A.; Broda, K.; and Russo, A. 2023. Hierarchies of reward machines. In *International Conference on Machine Learning*, 10494–10541. PMLR.

Garcia, C. E.; Prett, D. M.; and Morari, M. 1989. Model predictive control: Theory and practice—A survey. *Automatica*, 25(3): 335–348.

Guo, H.; Wu, F.; Qin, Y.; Li, R.; Li, K.; and Li, K. 2023. Recent trends in task and motion planning for robotics: A survey. *ACM Computing Surveys*, 55(13s): 1–36.

Heess, N.; Wayne, G.; Silver, D.; Lillicrap, T.; Erez, T.; and Tassa, Y. 2015. Learning continuous control policies by stochastic value gradients. *Advances in Neural Information Processing Systems*, 28.

Icarte, R. T.; Klassen, T. Q.; Valenzano, R.; and McIlraith, S. A. 2022. Reward machines: Exploiting reward function structure in reinforcement learning. *Journal of Artificial Intelligence Research*, 73: 173–208.

Kingma, D. P. 2013. Auto-encoding variational bayes. *ArXiv preprint ArXiv:1312.6114*.

Kiran, B. R.; Sobh, I.; Talpaert, V.; Mannion, P.; Al Sallab, A. A.; Yogamani, S.; and Pérez, P. 2021. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(6): 4909–4926.

Lagriffoul, F.; Dantam, N. T.; Garrett, C.; Akbari, A.; Srivastava, S.; and Kavraki, L. E. 2018. Platform-independent benchmarks for task and motion planning. *IEEE Robotics and Automation Letters*, 3(4): 3765–3772.

Lepora, N. F.; and Pezzulo, G. 2015. Embodied choice: how action influences perceptual decision making. *PLoS Computational Biology*, 11(4): e1004110.

Li, B.; Tang, H.; ZHENG, Y.; HAO, J.; Li, P.; Wang, Z.; Meng, Z.; and Wang, L. 2022. HyAR: Addressing Discrete-Continuous Action Reinforcement Learning via Hybrid Action Representation. In *International Conference on Learning Representations*.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2016. Continuous control with deep reinforcement learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2019. Continuous control with deep reinforcement learning. ArXiv:1509.02971.

Lo, S.-Y.; Zhang, S.; and Stone, P. 2018. PETLON: planning efficiently for task-level-optimal navigation. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 220–228.

Masson, W.; Ranchod, P.; and Konidaris, G. 2016. Reinforcement Learning with Parameterized Actions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1).

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533.

Neunert, M.; Abdolmaleki, A.; Wulfmeier, M.; Lampe, T.; Springenberg, T.; Hafner, R.; Romano, F.; Buchli, J.; Heess, N.; and Riedmiller, M. 2020. Continuous-discrete reinforcement learning for hybrid control in robotics. In *Conference on Robot Learning*, 735–751. PMLR.

O'Doherty, J. P.; Lee, S. W.; Tadayonnejad, R.; Cockburn, J.; Iigaya, K.; and Charpentier, C. J. 2021. Why and how the brain weights contributions from a mixture of experts. *Neuroscience & Biobehavioral Reviews*, 123: 14–23.

Parr, R.; and Russell, S. 1997. Reinforcement learning with hierarchies of machines. *Advances in Neural Information Processing Systems*, 10.

Schulman, J.; Moritz, P.; Levine, S.; Jordan, M.; and Abbeel, P. 2018. High-Dimensional Continuous Control Using Generalized Advantage Estimation. ArXiv:1506.02438.

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

Sutton, R. S. 1990. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Machine Learning Proceedings 1990*, 216–224. Elsevier.

Wei, E.; Wicke, D.; and Luke, S. 2018. Hierarchical Approaches for Reinforcement Learning in Parameterized Action Space. In *AAAI Spring Symposia*.

Xiong, J.; Wang, Q.; Yang, Z.; Sun, P.; Han, L.; Zheng, Y.; Fu, H.; Zhang, T.; Liu, J.; and Liu, H. 2018. Parametrized Deep Q-Networks Learning: Reinforcement Learning with Discrete-Continuous Hybrid Action Space. *CoRR*, abs/1810.06394.

Zhang, R.; Fu, H.; Miao, Y.; and Konidaris, G. 2024. Model-based Reinforcement Learning for Parameterized Action Spaces. In Salakhutdinov, R.; Kolter, Z.; Heller, K.; Weller, A.; Oliver, N.; Scarlett, J.; and Berkenkamp, F., eds., *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 58935–58954. PMLR.

Zhang, S.; and Whiteson, S. 2019. DAC: The double actor-critic architecture for learning options. *Advances in Neural Information Processing Systems*, 32.