# CHDP: Cooperative Hybrid Diffusion Policies for Reinforcement Learning in Parameterized Action Space

**Bingyi Liu[1,2], Jinbo He[1], Haiyong Shi[1], Enshu Wang[3*], Weizhen Han[1*], Jingxiang Hao[3], Peixi Wang[1], Zhuangzhuang Zhang[4]**

[1]School of Computer Science and Artificial Intelligence, Wuhan University of Technology
[2]Hubei Key Laboratory of Transportation Internet of Things (Wuhan University of Technology)
[3]School of Cyber Science and Engineering, Wuhan University
[4]Department of Computer Science, City University of Hong Kong
{byliu, jinbohe, shihaiyong, hanweizhen, wpx20101220}@whut.edu.cn, {wanges17, haojingxiang}@whu.edu.cn, zhuangzhuangzhang@cityu.edu.hk

## Abstract

Hybrid action space, which combines discrete choices and continuous parameters, is prevalent in domains such as robot control and game AI. However, efficiently modeling and optimizing hybrid discrete-continuous action space remains a fundamental challenge, mainly due to limited policy expressiveness and poor scalability in high-dimensional settings. To address this challenge, we view the hybrid action space problem as a fully cooperative game and propose a **Cooperative Hybrid Diffusion Policies (CHDP)** framework to solve it. CHDP employs two cooperative agents that leverage a discrete and a continuous diffusion policy, respectively. The continuous policy is conditioned on the discrete action's representation, explicitly modeling the dependency between them. This cooperative design allows the diffusion policies to leverage their expressiveness to capture complex distributions in their respective action spaces. To mitigate the update conflicts arising from simultaneous policy updates in this cooperative setting, we employ a sequential update scheme that fosters co-adaptation. Moreover, to improve scalability when learning in high-dimensional discrete action space, we construct a codebook that embeds the action space into a low-dimensional latent space. This mapping enables the discrete policy to learn in a compact, structured space. Finally, we design a Q-function-based guidance mechanism to align the codebook's embeddings with the discrete policy's representation during training. On challenging hybrid action benchmarks, CHDP outperforms the state-of-the-art method by up to 19.3% in success rate.

## 1 Introduction

Deep Reinforcement Learning (DRL) has excelled in domains with either purely discrete or continuous actions (Duan et al. 2021, 2025; Liu et al. 2024a). However, extending these successes to real-world scenarios requires tackling hybrid action space. Such a space combines categorical choices (e.g., selecting a specific tool in robotic manipulation, choosing a movement mode in autonomous driving) with fine-grained continuous adjustments (e.g., regulating force magnitude, tuning speed parameters).

Existing works (Masson, Ranchod, and Konidaris 2016; Massaroli et al. 2020) face two primary challenges in tackling these hybrid action tasks. The first is that current policies lack the expressiveness to handle the multi-modality of many hybrid-action tasks. This is because such multi-modality, where multiple action-pairs can be equally effective, conflicts with the unimodal architectures (e.g., Gaussian or deterministic) on which most methods are built (Fan et al. 2019; Hausknecht and Stone 2016). For example, a soccer goal can be scored with a left- or right-foot shot—distinct actions, each with unique continuous parameters (e.g., force, angle) (Masson, Ranchod, and Konidaris 2016). Consequently, such policies are forced to either average solutions into an ineffective compromise or collapse to a single mode, resulting in limited expressiveness and suboptimal performance (Jain, Akhound-Sadegh, and Ravanbakhsh 2024; Wang, Jin, and Montana 2025; Huang et al. 2023).

Second, existing methods (Xiong et al. 2018; Fan et al. 2019) face a scalability limitation, as they lack a strategy to overcome the combinatorial explosion in high-dimensional hybrid action space (Li et al. 2022). For instance, in non-prehensile manipulation, this challenge arises from selecting among numerous discrete contact points, each defined by its own set of continuous parameters (Zhou et al. 2023; Le et al. 2025). The difficulty of exploring such vast hybrid space leads to sample inefficiency, rendering most approaches impractical for these tasks (Zhang et al. 2024). Addressing these dual challenges of policy expressiveness and limited scalability is therefore crucial for unlocking DRL's potential in complex hybrid-action environments.

To address these challenges, we propose **Cooperative Hybrid Diffusion Policies (CHDP)**, a multi-agent reinforcement learning (MARL) framework that views the hybrid-action problem as a fully cooperative game between two agents. CHDP consists of two cooperative agents that leverage a discrete and a continuous diffusion policy, respectively. These agents collaborate to generate a final hybrid action aimed at maximizing a shared objective. This collaboration is implemented sequentially, where the continuous policy is conditioned on the discrete action representation to explicitly model the dependency between them. This design leverages the expressiveness of diffusion policies to capture

*Corresponding author.

complex, multi-modal distributions across their respective action spaces. However, this cooperative setting can lead to conflicts when policies are updated simultaneously, a challenge empirically observed in MARL (Lowe et al. 2017; Yu et al. 2022). Our framework employs a sequential policy update training scheme inspired by prior works (Wang et al. 2023; Liu et al. 2024c; Zhong et al. 2024). This mechanism prevents the optimization of one policy from inadvertently impairing the other, thus fostering co-adaptation and allowing CHDP to solve complex hybrid-action tasks. Moreover, to improve scalability when learning in high-dimensional discrete action space, we construct a codebook that embeds the high-dimensional discrete action space into a low-dimensional latent space. The discrete policy generates a continuous latent vector, which is then quantized by mapping it to its nearest neighbor in the embedding codebook. This quantization process confines the discrete policy's learning to a compact, structured space, thereby mitigating the curse of dimensionality. Subsequently, the selected embedding serves as a semantic condition to guide the continuous policy. Finally, we design a shared, Q-function-based guidance mechanism to align the codebook's embeddings with the latent representations generated by the discrete policy during training.

Our main contributions are summarized as follows:

- We propose CHDP, a MARL framework that solves the hybrid action space problem by fully leveraging two cooperative agents with diffusion policies. This design unleashes the expressiveness of the diffusion policies to capture multi-modal distributions in hybrid action space.
- We propose two key mechanisms: a sequential update scheme to mitigate policy conflicts and a Q-guided codebook to address scalability limitations through a compact action representation.
- CHDP demonstrates state-of-the-art (SOTA) performance on challenging hybrid action benchmarks, improving upon the prior SOTA method by up to $19.3\%$.

## 2 Related Work

**The Challenge of Limited Scalability.** While Deep Reinforcement Learning (DRL) has become the dominant approach for the hybrid action space problem (Hausknecht and Stone 2016; Fan et al. 2019), its various architectural explorations have consistently faced challenges with scalability and stability. Early attempts to homogenize hybrid action space, by either discretizing continuous actions or converting discrete ones into continuous, proved ineffective due to the curse of dimensionality and the creation of poorly structured policy functions, respectively (Li et al. 2022). Another line of research, centered on the Parameterized DQN (PDQN) (Xiong et al. 2018) suffers from critical redundancy and scalability issues. HACMan (Zhou et al. 2023) employs a similar architecture for non-prehensile manipulation, inheriting similar challenges. Meanwhile, hierarchical approaches like the Hierarchical Hybrid Q-Network (HHQN) (Fu et al. 2019) are plagued by severe non-stationarity (Wang et al. 2021). Collectively, these works lack an architecture that can scale effectively in domains with hybrid action space.

**The Limitation of Policy Expressiveness.** Running parallel to the scalability issue is the limitation of policy expressiveness. This trade-off is evident in recent methods: while HyAR (Li et al. 2022) offers a scalable latent-space framework, its expressiveness is constrained by its underlying deterministic policy and is often hampered by significant training instability (Liu et al. 2024b). Conversely, HyDo (Le et al. 2025) employs an expressive continuous diffusion policy, but its underlying architecture inherits the scalability limitations of methods like PDQN. Other works have focused solely on enhancing expressiveness through methods like energy-based models (Jain, Akhound-Sadegh, and Ravanbakhsh 2024), reparameterization in RPG (Huang et al. 2023), GMMs in LOM (Wang, Jin, and Montana 2025), and Diffusion Q-learning (Wang, Hunt, and Zhou 2023). However, these methods lack the necessary expressiveness to capture the multi-modality for both discrete and continuous action spaces.

In summary, the methods discussed above suffer from poor scalability and limited expressiveness. We propose CHDP, a novel framework that bridges this gap by combining the scalability of a Q-guided codebook with the rich expressiveness of diffusion policies.

## 3 Preliminaries

### 3.1 Parameterized Action MDP (PAMDP)

In this paper, we distinguish between two types of timesteps. We use superscripts $i \in \{1, \dots, N\}$ to denote the diffusion process timestep, and subscripts $t \in \{1, \dots, T\}$ to denote the reinforcement learning trajectory timestep.

The problem of controlling systems with hybrid actions is formalized as a Parameterized Action Markov Decision Process (PAMDP) (Hausknecht and Stone 2016). A PAMDP is defined by a tuple $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$, consisting of a state space $\mathcal{S}$, a hybrid action space $\mathcal{A}$, a transition function $P : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$, a reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and a discount factor $\gamma \in [0, 1)$. The key feature of a PAMDP lies in its hybrid action space, where an action $\mathbf{a} \in \mathcal{A}$ is a pair $\mathbf{a} = (a^d, a^c)$. This pair comprises a discrete action $a^d \in \mathcal{A}_d$ and a vector of continuous parameters $a^c$ from the discrete action-dependent set $\mathcal{A}_c(a^d)$.

To solve the PAMDP, we approach it from a MARL perspective. We decompose the single agent's decision-making process into a cooperative task executed by two agents. A discrete agent, guided by its policy $\pi_{\theta_d}(e_t|s_t)$, first selects a representation $e_t$. This representation is then used to derive both the discrete action $a_t^d$ and its corresponding embedding, $e_k$. Subsequently, a continuous agent determines the corresponding parameters $a_t^c$ via its policy $\pi_{\theta_c}(a_t^c|s_t, e_k)$, which is conditioned on both the state and the discrete action's embedding. Here, $\theta_d$ and $\theta_c$ are the learnable parameters of their respective policies. The objective is to learn a joint policy $\pi = (\pi_{\theta_d}, \pi_{\theta_c})$ that maximizes the expected discounted return $\mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r(s_t, \mathbf{a}_t)\right]$. Correspondingly, the action-value (or Q-value) for the joint policy $\pi$ is defined

as $Q_\phi^\pi(s_t, e_t, a_t^c) = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r(s_{t+k}, \mathbf{a}_{t+k}) | s_t, e_t, a_t^c\right]$, where $\phi$ denotes the parameters of the Q-function.

### 3.2 Diffusion Models and DQL

**Diffusion Models for Generative Learning.** Our framework builds upon diffusion models (Ho, Jain, and Abbeel 2020; Sohl-Dickstein et al. 2015; Song and Ermon 2019), a class of powerful generative models. A diffusion model learns to generate data by reversing a fixed forward process that gradually adds Gaussian noise to data $a_0$ over $N$ steps, transforming it into pure noise $a_N \sim \mathcal{N}(0, I)$.

The core of the model is to learn the reverse process. This is achieved by training a neural network, $\epsilon_\theta(a_i, s, i)$, to predict the noise that was added at each step $i$. Once trained, this network can be used to iteratively denoise a random noise vector to generate a new data sample $a_0$. Each step of this reverse process is performed as follows:

$$a_{i-1} = \frac{1}{\sqrt{\alpha_i}}\left(a_i - \frac{1-\alpha_i}{\sqrt{1-\bar{\alpha}_i}}\epsilon_\theta(a_i, s, i)\right) + \sqrt{\beta_i}z, \quad (1)$$

where $\alpha_i, \bar{\alpha}_i, \beta_i$ are hyperparameters from a predefined noise schedule, and $z \sim \mathcal{N}(0, I)$. The network $\epsilon_\theta$ is trained by the diffusion loss:

$$\mathcal{L}_d(\theta) = \mathbb{E}_{s,a_0,\epsilon,i}\left[\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_i}a_0 + \sqrt{1-\bar{\alpha}_i}\epsilon, s, i)\|^2\right]. \quad (2)$$

**Diffusion Q-Learning (DQL).** Diffusion Q-Learning (DQL) (Wang, Hunt, and Zhou 2023) adapts this generative framework to serve as an expressive policy in reinforcement learning. Instead of merely modeling the data distribution, a DQL policy must generate actions that maximize long-term returns. To achieve this, DQL augments the standard diffusion loss $\mathcal{L}_d(\theta)$ with a policy improvement term, $\mathcal{L}_q(\theta)$, which guides the policy towards high-value actions.

The overall training objective for a DQL policy is a weighted sum of these two components:

$$\mathcal{L}(\theta) = \mathcal{L}_d(\theta) + \alpha \cdot \mathcal{L}_q(\theta). \quad (3)$$

The policy improvement term $\mathcal{L}_q(\theta)$ is defined as the negative expected Q-value of the generated actions:

$$\mathcal{L}_q(\theta) = -\mathbb{E}_{s\sim\mathcal{D}, a_0\sim\pi_\theta}[Q_\phi(s, a_0)]. \quad (4)$$

DQL sets $\alpha = \eta / \mathbb{E}_{s,a\sim\mathcal{D}}[|Q_\phi(s, a)|]$, where $\eta$ is a hyperparameter that balances the two loss terms.

## 4 Cooperative Hybrid Diffusion Policies

### 4.1 Overview of the CHDP Framework

The proposed CHDP, depicted in Figure 1, features a novel cooperative framework with three core components. CHDP consists of two specialized agents: a discrete agent using policy $\pi_{\theta_d}$ for selecting discrete actions, and a continuous agent using policy $\pi_{\theta_c}$ for specifying the corresponding continuous parameters. In addition, a sequential update mechanism ensures co-adaptation by mitigating policy conflicts. Inspired by VQ-VAE (Van Den Oord, Vinyals et al. 2017), we construct a learnable Q-guided codebook, $\mathbf{E}_\zeta \in \mathbb{R}^{K\times d_e}$, which embeds the high-dimensional discrete action space into a low-dimensional latent space. Comprising $K$ learnable codewords ($e_k \in \mathbb{R}^{d_e}$), where $K$ is the cardinality of the discrete action space, the codebook also serves as the semantic bridge between the discrete and continuous policies.

At each timestep, the action generation process unfolds as a structured, sequential collaboration. Given the current state $s$, the discrete policy, $\pi_{\theta_d}$, first generates a latent representation $e$ by iteratively denoising an initial Gaussian noise vector $e_N \sim \mathcal{N}(0, I)$. Each sampling step is defined as:

$$e_{i-1} = \frac{1}{\sqrt{\alpha_i}}\left(e_i - \frac{1-\alpha_i}{\sqrt{1-\bar{\alpha}_i}}\epsilon_{\theta_d}(e_i, s, i)\right) + \sqrt{\beta_i}z, \quad (5)$$

where $z \sim \mathcal{N}(0, I)$ is a noise vector and $\epsilon_{\theta_d}$ denotes the noise prediction network for the discrete policy. The final output, $e = e_0$, then undergoes a Vector Quantization (VQ) (Linde, Buzo, and Gray 2003) step, where it is mapped to the nearest entry in the codebook. This yields the index $k$ of the nearest codeword, which is set as the discrete action ($a^d = k$), and the corresponding codeword vector $e_k$, which is retrieved to condition the continuous policy. The specific mechanism for this codebook will be detailed in Section 4.3.

Subsequently, the continuous policy $\pi_{\theta_c}$ generates the continuous action $a^c$. Critically, its reverse diffusion process is conditioned on both the state $s$ and the codeword $e_k$ from the first stage:

$$a_{i-1}^c = \frac{1}{\sqrt{\alpha_i}}\left(a_i^c - \frac{1-\alpha_i}{\sqrt{1-\bar{\alpha}_i}}\epsilon_{\theta_c}(a_i^c, s, e_k, i)\right) + \sqrt{\beta_i}z, \quad (6)$$

where $\epsilon_{\theta_c}$ is the noise prediction network. This directly infuses the semantic information of the discrete choice into the continuous action generation, ensuring the two components are coherent.

### 4.2 Sequential Update Scheme

CHDP employs a sequential update scheme for policy updates, a strategy inspired by Heterogeneous-Agent Reinforcement Learning (HARL) (Liu et al. 2024c; Zhong et al. 2024). As illustrated in Figure 1(Right), this scheme decomposes the joint optimization into a turn-based sequence.

**Step 1: Discrete Policy Update.** The training process begins with the discrete policy. CHDP first updates the discrete policy $\pi_{\theta_d}$, which provides a fixed, stable target for the subsequent update of the continuous policy. This update is performed by minimizing the objective function $\mathcal{L}(\theta_d)$ with the goal of improving the generated latent representation $e \sim \pi_{\theta_d}(\cdot|s)$, which is defined as:

$$\mathcal{L}(\theta_d) = \mathcal{L}_d(\theta_d) - \alpha \cdot \mathbb{E}_{s,a^c\sim\mathcal{D}, e\sim\pi_{\theta_d}}[Q_\phi(s, e, a^c)], \quad (7)$$

where $\mathcal{L}_d(\theta_d)$ is the standard diffusion loss from DQL, as defined in Eq. (2), which regularizes the policy using data from the replay buffer. The second term, $\mathcal{L}_q(\theta_d)$, is the policy improvement objective that guides the latent representation $e$ towards higher Q-values. The continuous action $a^c$ is sampled from the replay buffer $\mathcal{D}$, and this fixed value serves as a stable target for evaluating the latent representation $e$. This initial step provides a stable foundation for the subsequent optimization of the continuous policy and codebook.

**Step 2: Continuous Policy and Codebook Update.** In this second step, the newly updated discrete policy, $\pi'_{\theta_d}$, generates a latent representation $e' \sim \pi'_{\theta_d}(\cdot|s)$. This vector is

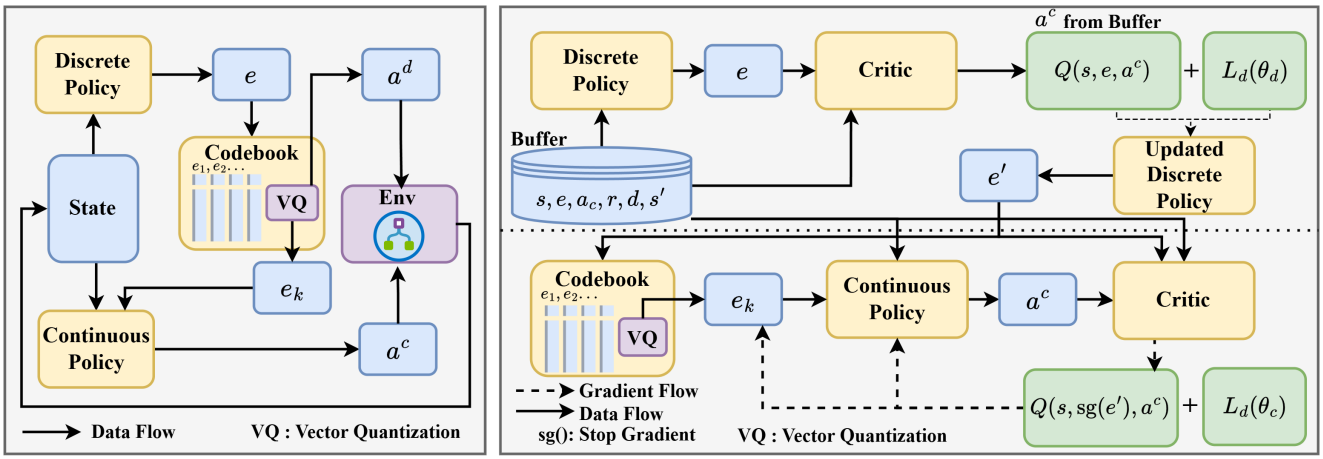



Figure 1: An overview of our CHDP framework. (Left) Inference: The discrete policy's latent representation is quantized by a codebook to yield a discrete action and relevant codeword. This codeword conditions the continuous policy for generating the final continuous action. (Right) Training: Guided by a shared Q-function, the discrete policy is updated first. Subsequently, conditioned on the output of this updated discrete policy, the continuous policy and the codebook are jointly optimized.

quantized to select a codeword $e_k$ from the codebook $\mathbf{E}_\zeta$, which in turn conditions the continuous policy to produce an action $a^c \sim \pi_{\theta_c}(\cdot|s, e_{a^d})$. The continuous policy parameters, $\theta_c$, and the codebook parameters, $\zeta$, are then jointly optimized by minimizing their shared objective $\mathcal{L}(\theta_c, \zeta)$:

$$\mathcal{L}(\theta_c, \zeta) = \mathcal{L}_d(\theta_c) + \alpha \cdot \mathcal{L}_q(\theta_c, \zeta), \tag{8}$$

where the first term, $\mathcal{L}_d(\theta_c)$, is a diffusion loss analogous to the one for the discrete policy, as defined in Eq. (2).

The second term, $\mathcal{L}_q(\theta_c, \zeta)$, is the policy improvement term that guides the continuous policy towards generating actions that yield higher Q-values. It is defined as:

$$\mathcal{L}_q(\theta_c, \zeta) = -\mathbb{E}_{s\sim\mathcal{D}, e'\sim\pi'_{\theta_d}, a^c\sim\pi_{\theta_c}}[Q_\phi(s, \text{sg}(e'), a^c)]. \tag{9}$$

The design of this objective is crucial for managing the gradient flow. Gradients from the Q-function update the continuous policy $\pi_{\theta_c}$ based on the output of the updated discrete policy, rather than the outdated discrete action from the replay buffer. Meanwhile, a stop-gradient operator $\text{sg}(\cdot)$ on the discrete representation prevents any gradient from flowing back to the discrete policy $\pi'_{\theta_d}$, thereby resolving potential conflicts in the policy updates.

Concurrently, the critics are updated by minimizing the Mean-Squared Bellman Error (MSBE). To ensure stability, we adopt the Double Q-learning paradigm (Van Hasselt, Guez, and Silver 2016). The loss for each critic $Q_{\phi_i}$ is:

$$\mathcal{L}(\phi_i) = \mathbb{E}_{(s_t, e_t, a_t^c, r_t, s_{t+1})\sim\mathcal{D}}\left[\left(Q_{\phi_i}(s_t, e_t, a_t^c) - y_t\right)^2\right], \tag{10}$$

where the target value $y_t$ is computed using the next action $\mathbf{a}_{t+1}$, which is sampled from the target policies $\pi'_{\theta'_d}$ and $\pi'_{\theta'_c}$. The target value is then defined as:

$$y_t = r_t + \gamma \min_{j=1,2} Q'_{\phi'_j}(s_{t+1}, e_{t+1}, a^c_{t+1}). \tag{11}$$

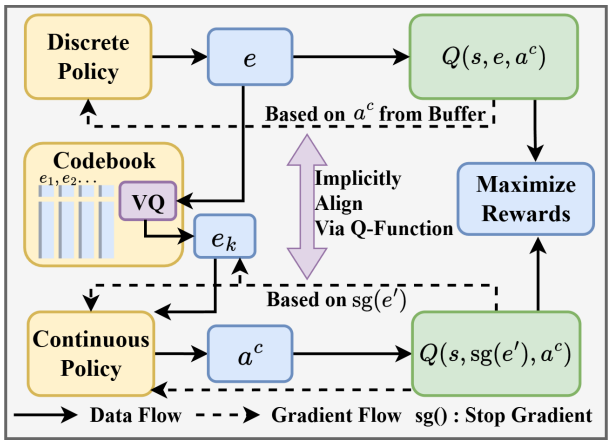



Figure 2: The process of aligning latent representations and codewords.

### 4.3 Downstream Q-guided Codebook

CHDP constructs a learnable codebook $\mathbf{E}_\zeta$ to represent the discrete action space. Instead of learning via reconstruction, the value of a codeword is measured by its functional impact on the downstream task, that is, its effectiveness in maximizing the Q-value. This is enforced by a deliberately asymmetric gradient flow, where the shared Q-function guides both the discrete policy's latent output and the selected codeword through distinct optimization pathways, forcing them to implicitly align.

The aligning process, as illustrated in Figure 2, begins with the discrete policy $\pi_{\theta_d}$ generating a latent representation $e \sim \pi_{\theta_d}(\cdot|s)$. This vector is then quantized to its nearest codeword $e_k$ in the codebook $\mathbf{E}_\zeta$ via a VQ operation:

$$k = \underset{k}{\text{argmin}} \|e - e_k\|^2. \tag{12}$$

The selected codeword $e_k$ is subsequently used to condition the continuous policy $\pi_{\theta_c}$, which generates the final continuous parameters.

The codebook's parameters, $\zeta$, are optimized via gradients originating from the continuous policy's improvement term $\mathcal{L}_q(\theta_c, \zeta)$. Since the continuous action $a^c$ is conditioned on the chosen codeword $e_k$, the final Q-value is an implicit function of this codeword. Consequently, during backpropagation, gradients from the Q-value propagate through the continuous action $a^c$ back to the specific codeword $e_k$ used in its generation. This downstream signal steers the codeword's embedding toward a representation that supports higher-value actions.

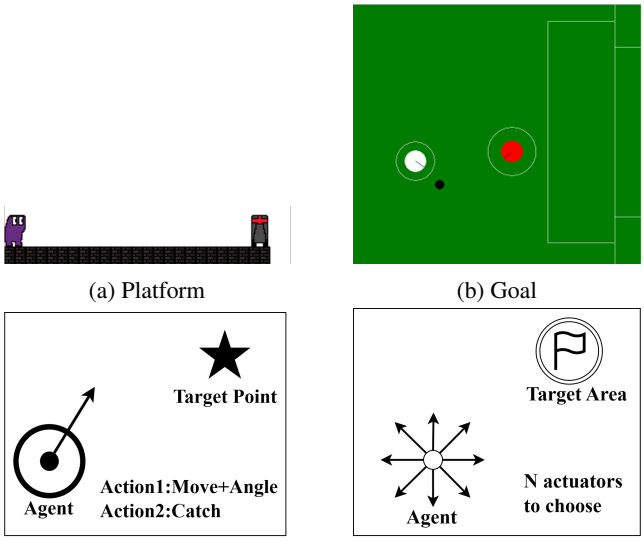
In contrast, the discrete policy, which generates the latent vector $e$, is optimized using the same Q-function, $Q(s, e, a^c)$, as defined in Eq. (7). Crucially, for this update, the continuous action $a^c$ is a detached constant sampled from the replay buffer $\mathcal{D}$, severing the gradient path through it. In this way, both the codebook embeddings $e_k$ and the discrete policy's outputs $e$ are guided by the same ultimate measure of utility, the Q-value, forcing them to implicitly align within the shared latent space. This training paradigm ensures the semantic meaning of each discrete action's embedding is forged by its downstream utility of maximizing the Q-value, rather than by reconstruction. This end-to-end process produces a task-aware, semantically potent representation of the discrete action space, ensuring a synergistic coupling between the policies. As a result, this integrated design provides a principled and scalable mechanism for representation learning that obviates the need for pre-training.

# 5 Experiments

We evaluate CHDP on various hybrid action environments against representative prior algorithms. Then, a detailed ablation study is conducted to verify the contribution of each component in CHDP. Moreover, we provide a qualitative experiment to show the expressiveness of the diffusion policy.

**Benchmarks:** We evaluated CHDP on eight standard Parameterized Action MDP (PAMDP) benchmarks: Platform and Goal (Masson, Ranchod, and Konidaris 2016), Catch Point (Fan et al. 2019), Hard Goal, and four variations of Hard Move (Li et al. 2022). These environments are identical to those used to evaluate HyAR, which represents the SOTA algorithm for these tasks. Visualizations of these environments are shown in Figure 3.

**Baselines:** We evaluate CHDP against several SOTA baselines for hybrid action space. Our primary comparison is against HyAR (Li et al. 2022), the current leading method. To ensure a fair and direct comparison, we adopt the same set of TD3-based baselines used in HyAR. These baselines extend prior methods using the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm (Fujimoto, Hoof, and Meger 2018) and include: PDQN-TD3 (an extension of PDQN (Xiong et al. 2018)), PA-TD3 (from PADDPG (Hausknecht and Stone 2016)), and HHQN-TD3 (from HHQN (Fu et al. 2019)). Additionally, we include the PPO-based HPPO (Fan et al. 2019) for broader coverage.




(a) Platform (b) Goal

(c) Catch Point (d) Hard Move

Figure 3: Visualizations of the tested environments.

## 5.1 Performance Evaluation

We conducted a comprehensive empirical study to evaluate the performance of our proposed method, CHDP, against a suite of strong baseline algorithms across eight standard PAMDP benchmarks. The mean success rates at the end of training are presented in Table 1, and the learning dynamics are shown in Figure 4. The results clearly demonstrate that CHDP achieves SOTA performance, consistently outperforming all baselines by showcasing superior policy expressiveness, scalability, and sample efficiency.

**Analysis of Policy Expressiveness.** CHDP demonstrates significant performance gains in hybrid action environments that demand highly expressive policies to handle multi-modal action distributions. This is most evident in the `Hard Goal` task, where CHDP achieves a success rate of $79.5\%$, substantially surpassing the strongest baseline, HyAR-TD3, which scored only $60.2\%$. This advantage extends across other challenging tasks. As shown in Table 1, CHDP also secures SOTA performance in the `Platform` ($99.7\%$) and `Catch Point` ($93.8\%$) environments. This underscores the advantage of CHDP's diffusion-based policy in capturing complex, multi-modal strategies.

**Analysis of Scalability in High-Dimensional Space.** The `Hard Move` suite of tasks is designed to test scalability as the size of the discrete action space increases. In these environments, a discrete action involves selecting an on/off state for each of $n$ actuators, creating a combinatorial space of size $2^n$. Each of these discrete choices is paired with its own set of continuous parameters (e.g., for $n = 4$, there are $2^4 = 16$ discrete actions, each with its own corresponding set of continuous parameters). As shown in Table 1, CHDP consistently outperforms all competing methods across all difficulty levels, from $n = 4$ to $n = 10$. Notably, CHDP maintains a success rate exceeding $90\%$ for $n$ up to 8 ($2^8 = 256$ options), a scale where other methods

| ENV | HPPO | PATD3 | PDQN-TD3 | HHQN-TD3 | HyAR-TD3 | CHDP(ours) |
|---|---|---|---|---|---|---|
| Goal | $0.0 \pm 0.0$ | $0.0 \pm 0.0$ | $71.4 \pm 4.3$ | $0.0 \pm 0.0$ | $77.3 \pm 4.2$ | $\mathbf{80.9 \pm 4.9}$ |
| Hard Goal | $0.0 \pm 0.0$ | $43.0 \pm 10.0$ | $0.0 \pm 0.0$ | $1.2 \pm 0.7$ | $60.2 \pm 5.0$ | $\mathbf{79.5 \pm 5.0}$ |
| Platform | $66.3 \pm 0.9$ | $95.1 \pm 3.6$ | $96.7 \pm 4.1$ | $56.7 \pm 29.4$ | $96.6 \pm 2.2$ | $\mathbf{99.7 \pm 0.2}$ |
| Catch Point | $55.7 \pm 5.2$ | $86.7 \pm 8.3$ | $89.8 \pm 3.2$ | $23.7 \pm 5.5$ | $86.6 \pm 0.9$ | $\mathbf{93.8 \pm 0.6}$ |
| Hard Move (n=4) | $3.3 \pm 1.3$ | $63.9 \pm 20.5$ | $79.7 \pm 4.8$ | $81.8 \pm 3.5$ | $91.4 \pm 2.4$ | $\mathbf{94.2 \pm 1.7}$ |
| Hard Move (n=6) | $2.5 \pm 0.4$ | $9.8 \pm 5.6$ | $31.1 \pm 8.1$ | $47.1 \pm 27.6$ | $92.3 \pm 0.6$ | $\mathbf{93.9 \pm 1.0}$ |
| Hard Move (n=8) | $2.3 \pm 0.9$ | $4.6 \pm 2.1$ | $6.6 \pm 2.5$ | $18.8 \pm 8.9$ | $88.3 \pm 1.9$ | $\mathbf{90.6 \pm 2.2}$ |
| Hard Move (n=10) | $3.4 \pm 0.5$ | $10.3 \pm 2.0$ | $3.3 \pm 0.5$ | $11.3 \pm 6.3$ | $69.0 \pm 5.6$ | $\mathbf{79.8 \pm 5.4}$ |

Table 1: Mean success rates ($\pm$ one standard deviation) over 5 independent trials, where the result for each trial is itself the average of the success rates from its final 5 evaluations.

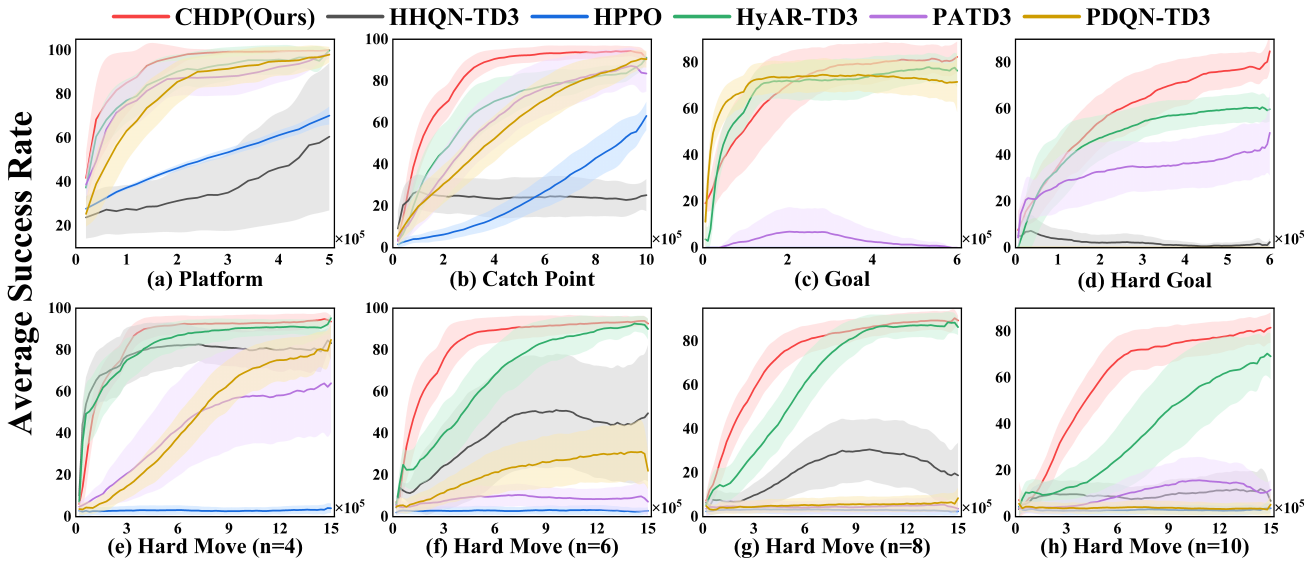



Figure 4: Comparisons of algorithms in different environments. The x- and y-axis represent the environment steps ($\times 10^5$) and the average success rate, respectively. The solid curve and the shaded region denote the mean and standard deviation over 5 independent runs. All curves are smoothed for visual clarity.

degrade significantly. This validates the excellent scalability of our Q-guided codebook approach.

**Analysis of Convergence and Sample Efficiency.** The learning curves in Figure 4 demonstrate CHDP's superior sample efficiency. Compared to all baselines, CHDP's learning curve (red) consistently exhibits a steeper ascent, indicating that it learns effective policies with substantially fewer environmental interactions. This advantage is particularly pronounced in `Platform` (a), `Catch Point` (b), and the `Hard Move` suite (e-h), where CHDP not only converges to a higher final performance but also does so more rapidly. Such high sample efficiency not only validates the effectiveness of our approach but is also a crucial factor for its practical viability in complex, resource-intensive tasks.

## 5.2 Ablation Study

To quantitatively validate our design, we conducted an ablation study to isolate the roles of our components: the expressive diffusion policies, the Q-guided codebook, and the sequential update scheme. The results on the `Hard Goal` and `Hard Move` ($n = 6$) environments are presented in Table 2. They demonstrate that each component is indispensable, addressing a distinct challenge within the framework.

**The Ablation of Diffusion Policy.** To verify our central hypothesis that policy expressivity is crucial, we created a variant where the diffusion policies were replaced by deterministic ones. This change led to a dramatic performance drop on both `Hard Goal` and `Hard Move`. Although this deterministic variant still achieved a 51.3% success rate in `Hard Goal`—outperforming several baselines—this result is far from SOTA. This starkly demonstrates that even with our cooperative framework, a policy constrained by expressiveness is insufficient. The success of the full CHDP model is therefore directly attributable to the diffusion policy's ability to capture complex, multi-modal action distributions, validating our primary motivation.

**The Ablation of Codebook.** The contribution of the Q-guided codebook is most evident in environments with a

| Method | Hard Goal | Hard Move |
|---|---|---|
| **CHDP (Full Model)** | **75.9 ± 3.7** | **93.9 ± 1.0** |
| w/o Diffusion Policy | 51.3 ± 10.2 | 45.1 ± 19.5 |
| w/o Codebook | 71.0 ± 6.0 | 11.1 ± 6.9 |
| w/o Sequential Update | 32.8 ± 4.3 | 89.4 ± 3.5 |
| w/o Both | 31.5 ± 16.4 | 10.7 ± 5.1 |

Table 2: Ablation of CHDP components. Values are mean success rates (± std.) over 5 independent runs. The score for each run is the average of its final 5 evaluations. The first row uses default parameters: Diffusion steps $N = 15$, $\eta = 5$ and $d_e = 8$, distinct from the configuration in Table 1. The 'w/o Both' variant removes both the Codebook and the Sequential Update scheme. Best results are in **bold**.

high-dimensional discrete action space. As illustrated in the `Hard Move` ($n = 6$) environment, which features $2^6$ discrete actions, replacing the codebook with a simple `argmax` selection over raw outputs caused performance to collapse catastrophically from 93.9% to 11.1%. This result confirms the codebook's critical role in ensuring scalability. By embedding the high-dimensional space into a compact latent space, the codebook provides a structured, low-dimensional representation for the discrete policy to learn in.

**The Ablation of Sequential Update.** To evaluate our sequential update scheme's role in mitigating optimization conflicts, we tested an ablation variant that updates both policies concurrently using data from the replay buffer, a mechanism identical to MADDPG (Lowe et al. 2017). The concurrent update is detrimental in the more complex `Hard Goal` environment, where performance collapses from 75.9% to 32.8%. We hypothesize this is because this task's challenge is policy coordination, making the prevention of conflicting updates critical. In contrast, the performance drop is minimal in the `Hard Move` environment (from 93.9% to 89.4%), where the main bottleneck is its high-dimensional action space. This confirms that our sequential update scheme is a vital component for ensuring co-adaptation, especially in coordination-intensive tasks.

**The Ablation of Both.** The 'w/o Both' variant, which lacks both the codebook and the sequential update mechanism, exhibits a near-total performance failure in both environments (31.5% and 10.7%). This result provides evidence that CHDP's success stems from a synergistic interplay between its components, confirming that both the codebook and sequential update are indispensable.

### 5.3 Qualitative Analysis of Expressive Policy

To assess our policy's multi-modality, we designed a targeted experiment. In the `Hard Move (n=6)` environment, we placed the target area just above the agent's fixed starting position, which renders the task solvable in a single step. This setting isolates our analysis from the confounding effects of a multi-step trajectory. Our analysis then hinges on two key metrics: the `Base Direction`, representing the resultant force vector from one of $2^6 = 64$ possible actuator combinations, and the `Continuous Action`, a scalar value to control the intensity of the final acceleration.

| Base Direction | Frequency | Continuous Action |
|---|---|---|
| **CHDP (Ours)** | | |
| $(-0.5, -0.866)$ | 79.0% | $-0.897 \pm 0.089$ |
| $(0.0, -0.866)$ | 17.0% | $-0.703 \pm 0.309$ |
| $(0.75, 0.443)$ | 4.0% | $0.945 \pm 0.070$ |
| **HyAR** | | |
| $(0.75, 0.433)$ | 100.00% | $0.445 \pm 0.000$ |

Table 3: Action distribution statistics across 100 trials.

Our analysis compares CHDP against the deterministic baseline HyAR. The Gaussian baseline, HPPO, is not included as it failed to solve the task. As detailed in Table 3, HyAR's policy collapses into a single, rigid mode, consistently selecting the same actuator combination. This results in a fixed `Base Direction` and an unvarying `Continuous Action` with zero variance, demonstrating its inability to capture the task's multi-modality. In contrast, CHDP leverages its expressiveness to discover a multi-modal policy, employing at least three distinct actuator combinations with frequencies of 79.0%, 17.0%, and 4.0%. Most notably, CHDP's policy demonstrates a sophisticated understanding of the task's dynamics by mastering different strategies. For instance, its primary approach (79.0% frequency) involves pairing a southwest-pointing `Base Direction`, (-0.5, -0.866), with a strong negative `Continuous Action` to effectively invert the force vector and reach the objective. In stark contrast, it simultaneously discovers a more intuitive, northeast-pointing strategy, (0.75, 0.443), coupled with a positive `Continuous Action`. This ability to identify and exploit multiple, non-intuitive, and even opposing pathways provides compelling evidence of CHDP's superior expressiveness.

## 6 Conclusion

In this paper, we have proposed CHDP, a novel framework that addresses hybrid action space problems using two cooperating agents, each driven by a diffusion policy. The cooperative design lies in modeling the complex dependencies between actions by conditioning the continuous policy on the discrete action's representation. To ensure both stability and efficiency, CHDP further incorporates a sequential update scheme to mitigate update conflicts and a Q-guided codebook to create a compact yet expressive representation for the discrete action space. Experimental results demonstrate that CHDP achieves SOTA performance on challenging benchmarks, outperforming the prior leading method by up to 19.3%. We further designed a qualitative experiment to evaluate the framework's expressiveness and scalability. In the `Hard Move` ($n = 6$) benchmark, CHDP successfully learned at least three distinct strategies, despite its large discrete action space of 64 ($2^6$) options.

## Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 62272357 and 62302326; in part by Wuhan Science and Technology Joint Project for Building a Strong Transportation Country under Grant 2024-2-7; in part by The State Key Laboratory of Integrated Services Networks under Grant ISN25-09, and in part by the Wuhan Science and Technology Project for Key Research and Development No. 2024050702030090.

## References

Duan, J.; Guan, Y.; Li, S. E.; Ren, Y.; Sun, Q.; and Cheng, B. 2021. Distributional soft actor-critic: Off-policy reinforcement learning for addressing value estimation errors. *IEEE transactions on neural networks and learning systems (TNNLS)*, 33(11): 6584–6598.

Duan, J.; Wang, W.; Xiao, L.; Gao, J.; Li, S. E.; Liu, C.; Zhang, Y.-Q.; Cheng, B.; and Li, K. 2025. Distributional soft actor-critic with three refinements. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Fan, Z.; Su, R.; Zhang, W.; and Yu, Y. 2019. Hybrid actor-critic reinforcement learning in parameterized action space. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, 2279–2285.

Fu, H.; Tang, H.; Hao, J.; Lei, Z.; Chen, Y.; and Fan, C. 2019. Deep multi-agent reinforcement learning with discrete-continuous hybrid action spaces. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, 2329–2335.

Fujimoto, S.; Hoof, H.; and Meger, D. 2018. Addressing function approximation error in actor-critic methods. In *International conference on machine learning (ICML)*, 1587–1596. PMLR.

Hausknecht, M. J.; and Stone, P. 2016. Deep Reinforcement Learning in Parameterized Action Space. In Bengio, Y.; and LeCun, Y., eds., *4th International Conference on Learning Representations (ICLR), 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems (NeurIPS)*, 33: 6840–6851.

Huang, Z.; Liang, L.; Ling, Z.; Li, X.; Gan, C.; and Su, H. 2023. Reparameterized policy learning for multimodal trajectory optimization. In *International Conference on Machine Learning (ICML)*, 13957–13975. PMLR.

Jain, V.; Akhound-Sadegh, T.; and Ravanbakhsh, S. 2024. Sampling from Energy-based Policies using Diffusion. In *Reinforcement Learning Conference (RLC)*.

Le, H.; Hoang, T.; Gabriel, M.; Neumann, G.; and Vien, N. A. 2025. Enhancing Exploration With Diffusion Policies in Hybrid Off-Policy RL: Application to Non-Prehensile Manipulation. *IEEE Robotics and Automation Letters (RA-L)*, 10(6): 6143–6150.

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 (ICLR)*.

Linde, Y.; Buzo, A.; and Gray, R. 2003. An algorithm for vector quantizer design. *IEEE Transactions on communications*, 28(1): 84–95.

Liu, B.; Hao, J.; Wang, E.; Jia, D.; Han, W.; Wu, L.; and Xiong, S. 2024a. Multi-agent reinforcement learning based resource allocation for efficient message dissemination in c-v2x networks. In *2024 IEEE/ACM 32nd International Symposium on Quality of Service (IWQoS)*, 1–10. IEEE.

Liu, J.; Hao, X.; ZHENG, Y.; Hu, Y.; Fan, C.; Lv, T.; Hu, Z.; et al. 2024b. Hybrid CtrlFormer: learning adaptive search space partition for hybrid action control via transformer-based Monte Carlo tree search. In *The 40th Conference on Uncertainty in Artificial Intelligence (UAI)*.

Liu, J.; Zhong, Y.; Hu, S.; Fu, H.; FU, Q.; Chang, X.; and Yang, Y. 2024c. Maximum Entropy Heterogeneous-Agent Reinforcement Learning. In *The Twelfth International Conference on Learning Representations (ICLR)*.

Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; and Mordatch, I. 2017. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. *Neural Information Processing Systems (NeurIPS)*.

Massaroli, S.; Poli, M.; Bakhtiyarov, S.; Yamashita, A.; Asama, H.; and Park, J. 2020. Neural ordinary differential equation value networks for parametrized action spaces. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*.

Masson, W.; Ranchod, P.; and Konidaris, G. 2016. Reinforcement learning with parameterized actions. In *Proceedings of the AAAI conference on artificial intelligence (AAAI)*, volume 30.

Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning (ICML)*, 2256–2265. pmlr.

Song, Y.; and Ermon, S. 2019. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems (NeurIPS)*, 32.

Van Den Oord, A.; Vinyals, O.; et al. 2017. Neural discrete representation learning. *Advances in neural information processing systems (NeurIPS)*, 30.

Van Hasselt, H.; Guez, A.; and Silver, D. 2016. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence (AAAI)*, volume 30.

Wang, M.; Jin, Y.; and Montana, G. 2025. Learning on One Mode: Addressing Multi-modality in Offline Reinforcement Learning. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Wang, R.; Yu, R.; An, B.; and Rabinovich, Z. 2021. I2hrl: Interactive influence-based hierarchical reinforcement learning. In *Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence (IJCAI)*, 3131–3138.

Wang, X.; Tian, Z.; Wan, Z.; Wen, Y.; Wang, J.; and Zhang, W. 2023. Order Matters: Agent-by-agent Policy Optimization. In *The Eleventh International Conference on Learning Representations, (ICLR) 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Wang, Z.; Hunt, J. J.; and Zhou, M. 2023. Diffusion Policies as an Expressive Policy Class for Offline Reinforcement Learning. In *The Eleventh International Conference on Learning Representations (ICLR)*.

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.

Yu, C.; Velu, A.; Vinitsky, E.; Gao, J.; Wang, Y.; Bayen, A.; and Wu, Y. 2022. The surprising effectiveness of PPO in cooperative multi-agent games. In *Proceedings of the 36th International Conference on Neural Information Processing Systems (NeurIPS)*, 24611–24624.

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 (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, 58935–58954. PMLR.

Zhong, Y.; Kuba, J. G.; Feng, X.; Hu, S.; Ji, J.; and Yang, Y. 2024. Heterogeneous-Agent Reinforcement Learning. *Journal of Machine Learning Research (JMLR)*, 25(32): 1–67.

Zhou, W.; Jiang, B.; Yang, F.; Paxton, C.; and Held, D. 2023. HACMan: Learning Hybrid Actor-Critic Maps for 6D Non-Prehensile Manipulation. In Tan, J.; Toussaint, M.; and Darvish, K., eds., *Conference on Robot Learning, CoRL 2023, 6-9 November 2023, Atlanta, GA, USA*, volume 229 of *Proceedings of Machine Learning Research*, 241–265. PMLR.