CTDE^2: Continuous Training Discrete Execution

Nicolas Rowies
Vrije Universiteit Brussel
Brussels, Belgium
nicolas.anthony.rowies@vub.be
nicolasrowies@gmail.com

Florent Delgrange
Vrije Universiteit Brussel & Flanders Make
Brussels, Belgium
florent.delgrange@vub.be

Ann Nowé
Vrije Universiteit Brussel
Brussels, Belgium
ann.nowe@vub.be

Diederik M. Roijers
Gemeente Amsterdam
Amsterdam, Netherlands
Vrije Universiteit Brussel
Brussels, Belgium
diederik.roijers@vub.be

ABSTRACT

Multi-agent reinforcement learning is challenging, especially when combined with continuous action spaces. To tackle this challenge, we introduce the Continuous Training with Discrete Execution (CTDE^2) paradigm. The key insight behind this is that at execution time, the agents do not typically require the full infinite action space, but can rely on the smaller set of actions strictly required by the optimal policy. Furthermore, in practice agents often will only need a compact learned set of discrete actions (i.e. codebook). As such, we can learn which codebook is needed, and make both policy execution and learning much faster. In this paper, we propose the MQ-LAN algorithm which operationalizes CTDE^2 by replacing exhaustive sampling of the action space with a learnable codebook of discrete action embeddings. The candidate actions can dynamically migrate across the continuous value landscape via gradient ascent. Empirical results confirm that treating the discretization as a learnable parameter allows MQ-LAN to achieve strong computational efficiency and superior returns compared to rigid discrete baselines, Actor-Critic methods, and Q-functionals. Based on these results, we believe that CTDE^2 is a highly scalable and efficient pathway to tackle continuous multi-agent RL.

KEYWORDS

Multi-Agent reinforcement learning, Continuous reinforcement learning, Continuous control

1 INTRODUCTION

Value-based Multi-Agent Reinforcement Learning (MARL) has established state-of-the-art sample efficiency and performance in complex coordination tasks, an advancement largely driven by the Centralized Training with Decentralized Execution (CTDE) paradigm [14, 18]. However, extending these decentralized execution mechanisms to physical real-world domains (which inherently require precise continuous actuation, such as fluid robotic locomotion or dynamic reservoir control [3]) exposes a critical computational bottleneck. While discrete value-based MARL relies on highly efficient linear-time arg max enumerations over factorized utilities or local advantage functions to derive greedy policies during execution [2, 18], translating this operation to continuous action spaces requires solving a non-convex global optimization problem over a complex joint value function at every single timestep. This optimization burden renders continuous value-based MARL practically intractable for real-time, high-dimensional multi-agent execution.

To bypass the continuous optimization bottleneck, literature often relies on Actor-Critic architectures (e.g., MAPPO [29], MASAC [8]), which amortize action-selection costs but are more prone to suffer from high-variance, sample inefficiency, and stale action-value estimates [5, 12]. Recent algorithmic extensions attempt to mitigate these limitations through advanced representation learning [10], scalable natural gradients [9, 28], or sequence-conditioned critics [27]. However, these approaches rely on sophisticated architectural interventions rather than directly addressing the continuous maximization problem. Conversely, preserving value-based methods in continuous domains demands severe compromises. For instance, CAQL [20] relies on expensive iterative optimization, NAF [6] restricts Q-functions to uni-modal representations [13], and highly expressive Q-functionals [13] require intractable sampling densities to locate true maxima [19]. Consequently, a persistent trade-off remains between algorithmic expressivity, sample efficiency, and execution tractability.

In this paper, we resolve this dichotomy by challenging the fundamental assumption that continuous execution requires a uniform, infinite action space. We present the *best action reusability hypothesis*, demonstrating empirically that optimal multi-agent policies exhibit profound action selection overlap across varying states. Rather than utilizing the full continuous spectrum, agents naturally gravitate toward highly specialized, localized action clusters dictated by their emergent roles. Building upon this insight, we introduce a novel paradigm: **Continuous Training with Discrete Execution (CTDE^2)**. Unlike heuristic approaches such as Bang-Bang discretization [21] that rigidly bin the action space a priori at the boundaries, CTDE^2 treats the discretization itself as a dynamically learnable parameter. During the continuous training phase, agents discover and continually refine an individualized optimal discrete action set (i.e., a codebook) which is then exclusively utilized during decentralized execution.

Proc. of the Adaptive and Learning Agents Workshop (ALA 2026), Aydeniz, Delgrange, Mohammedalamen, Yang (eds.), May 25 – 26, 2026, Paphos, Cyprus, https://alaworkshop2026.github.io/. 2026.

To operationalize the CTDE^2 paradigm, we propose the MovingQ algorithm. MovingQ maintains a persistent codebook of action embeddings that dynamically migrate across the continuous value landscape via gradient ascent, effectively seeking out high-value coordinates without restricting the underlying value function's expressivity. We integrate this mechanism into the highly scalable Local Advantage Networks (LAN) [2] framework to construct MovingQ-LAN (MQ-LAN). Because the execution phase evaluates only a fixed codebook of highly overlapping optimal actions, MQ-LAN successfully reduces the intractable continuous global optimization problem to a simple, highly efficient discrete enumeration. Our empirical evaluations demonstrate that MQ-LAN substantially outperforms continuous Actor-Critic baselines, value-based continuous Q-functionals, and rigid discrete baselines, achieving superior Interquartile Mean (IQM) returns. By eliminating the necessity for exhaustive continuous maximization, this work establishes CTDE^2 as a highly efficient, scalable pathway for high-dimensional continuous multi-agent reinforcement learning.

2 BACKGROUND

2.1 Decentralized POMDPs & CTDE

We ground our problem in the formalism of a Markov Decision Process (MDP) [11], defined by the tuple $\langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$. At each timestep t, the agent observes a state $s_t \in \mathcal{S}$, selects an action $a_t \in \mathcal{A}$, and receives a reward $r_t = R(s_t, a_t)$. The environment transitions to a new state s_{t+1} according to the probability distribution $P(s_{t+1}|s_t, a_t)$. The agent's goal is to learn a policy $\pi(a|s)$ that maximizes the expected cumulative discounted return $G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$.

In multi-agent scenarios with partial observability, this extends to a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) [15], defined as $\langle \mathcal{S}, \mathcal{A}, P, R, \Omega, O, \gamma \rangle$. Here, agents do not observe the global state s_t directly. Instead, each agent i receives a local observation $o_t^{(i)} \in \Omega$ drawn from the observation function $O(\cdot \mid s_t, a_{t-1})$. Because the local observation is insufficient to infer the full state, agents typically rely on action-observation histories $\tau_t^{(i)} = (o_0^{(i)}, a_0^{(i)}, \ldots, o_t^{(i)})$ to approximate the state and coordinate their actions.

2.2 Local Advantage Networks (LAN)

To tackle the Dec-POMDP in a value-based framework, we adopt Local Advantage Networks (LAN) [2]. LAN operates on the principle of learning a decentralized best-response policy without constructing a full joint Q-function. Instead of estimating the expected cumulative return for the joint actions of all agents in a given state (and following the current policy accordingly), LAN completely avoids the exponential complexity of learning a Q-function over the joint action space. Specifically, it decomposes the value into a shared centralized value function $V^\pi(s, \tau)$, which depends on the global state s and the joint observation-action history $\tau = (\tau^{(1)}, \ldots, \tau^{(N)})$ of all agents, and individual local advantage functions $A^{\pi^{(i)}}(\tau^{(i)}, a^{(i)})$, which depend only on the local history $\tau^{(i)}$ and actions $a^{(i)}$ of the specific agent i. The local Q-value proxy is defined as:

$$\tilde{Q}^{\pi}_{(i)}(s, \tau, a^{(i)}) = V^\pi(s, \tau) + A^{\pi^{(i)}}(\tau^{(i)}, a^{(i)}) \tag{1}$$

We select LAN as our foundation because it offers high scalability. Unlike QMIX [18] or QPLEX [26], which utilize mixing networks that grow in complexity with the number of agents, LAN's centralized critic is discarded during execution, leaving agents with lightweight, independent advantage networks suitable for decentralized execution.

2.3 Value-Based Continuous Action RL

Extending value-based methods to continuous action spaces $\mathcal{A} \in \mathbb{R}^d$ introduces the maximization optimization problem. In discrete spaces, deriving a deterministic greedy policy, which maps a state directly to an optimal action via $\arg\max_a Q(s, a)$, is a simple enumeration. In continuous spaces, this becomes a global optimization problem over a potentially non-convex surface, which is computationally intractable to solve at every timestep.

2.3.1 Normalized Advantage Function (NAF). NAF [6] restricts the advantage function to a quadratic form,

$$A(s, a) = -\frac{1}{2}(a - \mu(s))^T M(s)(a - \mu(s)),$$

where $\mu(s)$ represents the analytically derived optimal action for state s, and $M(s)$ is a state-dependent positive-definite square matrix. This ensures the global maximum is analytically available at $\mu(s)$. However, this restriction limits the agent to representing uni-modal value functions, which is often insufficient for complex multi-agent coordination where the value landscape may be multi-modal [13].

2.3.2 Continuous Action Q-Learning (CAQL). CAQL [20] avoids structural restrictions by approximating $Q(s, a)$ with deep neural network critic. To perform the maximization, it solves a Mixed-Integer Program (MIP) or uses iterative gradient-based optimization during the update step. While expressive, the computational cost of solving an optimization problem at every inference step is prohibitive for real-time control.

2.3.3 Q-functionals. This approach models the Q-function as a linear combination of state-dependent coefficients and a basis function expansion over the action space (e.g., polynomials) [13]. This allows the maximization to be approximated via efficient matrix multiplication over a batch of sampled actions ("maximization by sampling"). However, high-rank bases required for complex tasks create nuanced value landscapes that necessitate intractable sampling densities to locate the true maximum.

2.3.4 Multi-Agent Continuous Control. The challenges of continuous value-based learning are further amplified in Multi-Agent Reinforcement Learning (MARL). While Actor-Critic architectures dominate this space, tracing the literature reveals limitations in using policy networks for continuous action selection. Standard deterministic methods, train a policy network to estimate the maximum-valued action [24]. In doing so, the network acts as an amortized estimator of the Q-function to bypass expensive computations. However, this forces the agent to optimize actions based on a potentially stale critic.

Furthermore, stochastic policy gradient methods generally parameterize a simple distribution, such as a Gaussian, for action selection [8]. Because these policies are either deterministic or

drawn from a tight distribution around a single maximum, it is structurally unlikely for the network to sample multiple high-value actions if they are spread far apart in the action space. Consequently, methods relying on Gaussian parameterizations inherently fail to represent the multi-modal nature of complex optimal policies [23].

Recent work has proposed Mixed Q-functionals (MQF) [4]. MQF extends the Q-functional architecture to the multi-agent setting by allowing agents to learn individual functional representations that are subsequently mixed via a centralized network. This enables the efficient, simultaneous evaluation of joint actions without an exhaustive search. Other approaches, such as Continuous QMIX (CQMIX) [17], attempt to retain the mixing network structure by replacing the analytical maximization with iterative optimization methods like the Cross-Entropy Method (CEM) during execution. However, these methods often face a steep trade-off between the precision of the continuous maximization and the computational latency required for real-time multi-agent coordination.

3 OTHER RELATED WORK

Neural Discrete Representation Learning. Our approach to learning a discrete codebook has conceptual similarity to models like the Vector Quantized-Variational AutoEncoder (VQ-VAE) [25]. In representation learning, VQ-VAE models continuous data by mapping latent encodings to the nearest element in a learnable, discrete codebook, optimized via a vector quantization objective. We transpose this powerful concept into the reinforcement learning domain. Rather than quantizing data representations, our approach effectively discretizes the continuous action space itself into a dynamic, learnable codebook of optimal candidate actions that are continually updated via gradient ascent to maximize expected returns.

Action Reuse and Heuristic Search. In multi-agent settings, precisely maximizing joint payoffs becomes computationally prohibitive as the agent count scales. Approaches like Reusing Iterative Local Search (RILS) [16] address this in discrete domains by replacing exact global maximization with a rapid local search. RILS exploits temporal locality by "warm-starting" its search using the best joint assignment from the previous timestep, recognizing that optimal actions rarely shift abruptly. RILS serves as a discrete conceptual precursor to our CTDE^2 paradigm. However, while RILS applies action reuse merely as a local, temporal initialization heuristic, our work fundamentally scales this principle to continuous domains. It persistently maintains and globally refines a reusable codebook of continuous action embeddings across the entire learning process, bypassing the need for exhaustive search during execution.

4 THE CTDE^2 PARADIGM & MOVINGQ

As established in Section 2.3, existing continuous value-based architectures struggle with a strict trade-off between expressivity and tractability. Approximating highly expressive, unconstrained value functions often requires solving computationally prohibitive global optimization problems at every inference step [4, 13, 20], whereas restrictive structural constraints limit the modeling of multi-modal returns [6, 13]. To resolve this bottleneck, we introduce **the Continuous Training with Discrete Execution (CTDE^2)** paradigm and design the **MovingQ** algorithm (Section 4.3) explicitly to bypass exhaustive continuous optimization during both training and execution.

4.1 Hypothesis: Best Action Reusability

We first propose the best action reusability hypothesis. We hypothesize that for a wide class of continuous control problems, optimal policies exhibit a strong overlap in action selection across varying states. Rather than utilizing the full continuous action space uniformly, the optimal policy frequently reuses a concentrated subset of overlapping actions. Formally, given two randomly sampled states s_1 and s_2 from the environment's state distribution ρ, the probability that their optimal actions $a^*(s)$ are nearly identical (within a small margin ϵ) is significantly higher than if these actions were selected purely at random from a uniform distribution over the continuous action space $\mathcal{A}$:

$$\mathbb{P}_{s_1,s_2\sim\rho}(||a^*(s_1) - a^*(s_2)||_2 < \epsilon) \gg \mathbb{P}_{a_1,a_2\sim U(\mathcal{A})}(||a_1 - a_2||_2 < \epsilon). \tag{2}$$

In other words, the best actions for certain states are highly likely to also be the best actions for other states under an optimal policy.

Empirical observation of agent behavior (specifically when learning with the Q-functional LAN architecture [13], (see also Figure 10) strongly supports this hypothesis (illustrated in Figure 1). Specifically, the marginalized spatial distribution of utilized actions over states for the learned policy in the Waterworld environment (see Section 5.1), reveals that optimal actions are highly concentrated rather than uniformly distributed globally. Agent actions naturally gravitate toward specific high-value regions of the action space.

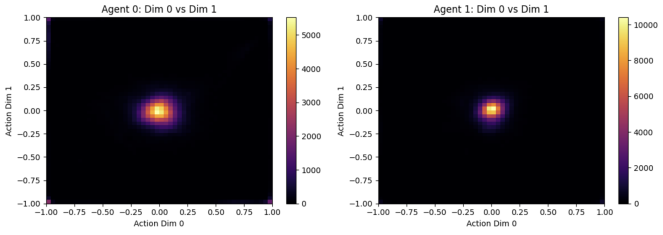


Figure 1: Action distribution of an optimal policy marginalized over states in the Waterworld environment.

We note that, as can be seen in Figure 1, the marginalized spatial distribution of optimal actions exhibits noticeable heterogeneity across agents. While both agents display concentrated action distribution near the origin, their specific spatial profiles diverge. Agent 1 exhibits a highly dense, tightly clustered distribution, whereas Agent 0 maintains a relatively broader, yet still localized, action profile. This variance highlights a critical insight: role-based specialization dictates that depending on their emergent roles, each agent restricts itself to a specific, but sometimes different specialized subset of candidate actions.

To contextualize, Waterworld simulates swarm behavior where pursuer agents coordinate to encircle targets and avoid poison using continuous 2D thrust vectors. Here, Agent 1 adopts a "follower" or "blocker" role via localized, small maneuvers, whereas Agent 0 acts as a highly mobile "chaser" utilizing a distinct action subset.

Ultimately, this demonstrates that different emergent roles exploit distinct regions of the continuous action space, strongly reinforcing the best action reusability hypothesis across diverse policies.

4.2 Continuous Training with Discrete Execution (CTDE^2)

Based on the empirical observations supporting the best action reusability hypothesis, we recognize that optimal actions are not uniformly distributed globally. To leverage this inherent reusability of actions across different states, we developed the Continuous Training with Discrete Execution (CTDE^2) paradigm.

The core philosophy of CTDE^2 is to explicitly leverage this hypothesis by finding the best overlapping candidate actions shared across different states, thereby decoupling the complexity of continuous learning from the computational constraints of execution. This paradigm allows the agent to learn the optimal discrete action set (i.e., the codebook) and the optimal policy simultaneously. This is achieved through a continuous training process where a finite set of action embeddings is allowed to dynamically change over the training process, enabling the discrete codebook to explore over the continuous action space while training. As the critic learns the underlying value function, it utilizes its evolving beliefs to actively select and guide these action embeddings towards increasingly better, high-value coordinates.

A critical nuance of this simultaneous learning process is the explicit optimization for specialist rather than generalist actions in our codebook. Formally, let $U = \{u_1, u_2, ..., u_K\}$ be the codebook of K action embeddings, and let $a^*(s) = \arg\max_{a \in \mathcal{A}} Q^*(s, a)$ represent the true optimal continuous action(s) for state s. The objective of the codebook is to minimize the expected projection error over the state distribution induced by the optimal policy, ρ_{π^*} :

$$\mathcal{L}(U) = \mathbb{E}_{s \sim \rho_{\pi^*}} \left[\min_{u_k \in U} ||a^*(s) - u_k||_2^2 \right] \tag{3}$$

Because $a^*(s)$ is a true value and inaccessible during training, we treat this equation mainly as a conceptual framework which serves as groundwork for algorithms trying to implement this approach. Implementing a solution to this conceptual objective depends entirely on the specific algorithm and its individual approach.

When the allowed number of action slots K is sufficiently large, the codebook can provide a pure specialist ($u_k \approx a^*(s)$) for nearly every critical state. However, because K is strictly constrained to ensure computational efficiency at execution, a perfect mapping for every state is impossible. Consequently, the mechanism gracefully degrades into localized averaging. Instead of defaulting to a single generalist candidate action that yields safely sub-optimal return everywhere, each action embedding u_k converges to the local average of the marginalized states that optimally select it. Specifically, if $\mathcal{S}_k \subset \mathcal{S}$ represents the localized subset of states where u_k is the best available action in the codebook, the embedding naturally approximates $u_k \approx \mathbb{E}_{s \sim \mathcal{S}_k}[a^*(s)]$. By prioritizing localized peak performance over broad adequacy, the agent ensures it remains as specialized as possible within the constraints of K.

Allowing action embeddings to change preserves the continuous nature of the training process, ensuring the agent can map complex environments and explore freely without restricting the action space's expressivity. However, once a good policy is found and training concludes, these action embeddings settle into a fixed, optimized codebook. Consequently, at execution time, the agent transitions to operate purely as a discrete model. By evaluating this fixed codebook of highly overlapping optimal actions, execution is reduced to a simple, highly efficient enumeration, successfully bypassing the computational latency of exhaustive continuous optimization.

4.3 The MovingQ Mechanism

To implement the CTDE^2 paradigm, we require a mechanism that can effectively discover and refine the codebook of action embeddings. We propose MovingQ, an algorithm that addresses the optimization bottleneck by using the gradient ascent and evolutionary techniques to approximate the best policy with the best codebook.

MovingQ maintains a persistent codebook (i.e., a learnable discrete action set) matrix $U \in \mathbb{R}^{K \times d}$ of K discrete action embeddings. These actions are treated as learnable parameters. MovingQ executes a multi-step process to simultaneously learn the value function via a critic network and optimally position this codebook for maximal expressivity preservation. This creates a highly coupled dual-learning dynamic: while the critic is trained to evaluate state-action pairs and identify strong policies, the action embeddings actively migrate across this evolving value landscape. The training process initially relies on high action embedding mobility to broadly identify high-value action regions, eventually decaying this mobility to fine-tune their positions locally and crystallize the most effective codebook. Consequently, the algorithm concurrently optimizes both the agent's expected return and the specific action expressivity required to execute that policy.

4.3.1 Gradient Ascent. To optimize the actions while ensuring they remain within the valid boundaries of the environment's action space, we distinguish between the internal parametrization and the executed action. Let $\tilde{u}_k$ represent the unbounded, learnable action embedding coordinates, and let ϕ be a bounding activation function (such as Tanh). The actual continuous action executed by the agent is therefore the projected value $u_k = \phi(\tilde{u}_k)$. To perform the optimization, we freeze the critic parameters and backpropagate the gradient of the advantage function directly into the unconstrained action embedding coordinates $\tilde{u}_k$

$$\tilde{u}_k \leftarrow \tilde{u}_k + \alpha \cdot \nabla_{\tilde{u}_k} A(s, \phi(\tilde{u}_k)) \tag{4}$$

Given a particular state s, we effectively move our action embeddings higher on the slope dictated by the critic's value function. This increases the expected return of this specific action for the state s, by modifying the continuous action it represents.

4.3.2 Vectorized Codebook Management. To maximize codebook efficiency and prevent expressivity collapse, we employ multiple regulatory mechanisms:

(1) **Competitive Learning:** Only the Top-K action embeddings (those yielding the highest expected return) are updated. This refines the best actions into specialists and avoids sub-optimal updates that would degrade their utility elsewhere.
(2) **Pruning via Proximity & Starvation:** To prevent collapse into shared local optima, we prune redundant embeddings whose Euclidean distance falls below a threshold

($||u_i - u_j|| < \sigma$), retaining the one with higher historical utility. A starvation mechanism also prunes rarely selected "dead units" to free up computational slots.

(3) **Maximin Resets:** Pruned embeddings are re-initialized in the largest unexplored voids of the action space. By maximizing the minimum distance to the active codebook, we actively drive exploration.

(4) **Freezing Action Embeddings:** Following a reset, new embeddings are placed in regions with inaccurate critic estimates. To prevent misguided gradient ascent, they are temporarily frozen. This allows them to generate exploration data while remaining locked until the critic reduces its prediction error.

In essence, MovingQ transforms global optimization bottleneck into a dynamic, codebook-based ecosystem that fluidly transitions from broad exploration to precise exploitation. Initially, the action embeddings serve a foundational exploratory purpose: acting as active explorers, they generate the diverse data necessary for the critic to accurately evaluate the expected return of different state-action pairs. This mobility is governed by a dedicated ζ-exploration parameter that controls their gradient step size. As the critic refines its value estimations and the ζ parameter gradually decays, this movement naturally shifts from global exploration to localized fine-tuning. Consequently, the embeddings gracefully transition from wide-ranging exploration to settling strategically into their final, optimized coordinates. Guided by the algorithm's competitive and evolutionary strategies, the discrete action embeddings ultimately crystallizes into the highly effective codebook of the learned policy.

4.4 MovingQ-LAN Architecture

To construct the final Multi-Agent architecture, we integrate the proposed MovingQ algorithm into the Local Advantage Networks (LAN) framework [2]. In MovingQ-LAN (MQ-LAN), the agent's local advantage network preserves the core structural attributes of the original LAN, such as utilizing a Gated Recurrent Unit (GRU) to process the action-observation history and using a replay buffer for batch training.

Because we use MovingQ's framework, we adapt the replay buffer such that the learned actions are the exact continuous actions used at the moment of the reward, rather than linking states to the newly updated candidate actions. The architecture utilizes a single, weight-sharing model which takes a critic-only approach. Instead of outputting a scalar value for a single action, the network maintains a persistent codebook of action embeddings, giving an expected return for every candidate action in the codebook simultaneously. It then updates the Top-K candidate actions based on the local critic. The rest of the architecture, such as the centralized critic, remains consistent with the original LAN structure [2]. To be more precise, the changing part of the architecture on the illustration 2 can be observed on the right below part.

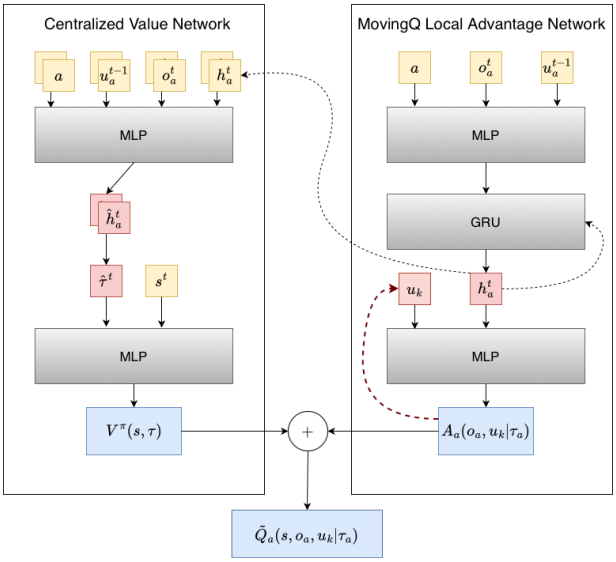


Figure 2: Overview of the MovingQ-LAN architecture

Depending on the environment and the homogeneity of the agents, the action embedding codebook can be instantiated either per-agent or shared globally across the team. A per-agent codebook allows individual agents to cultivate highly specialized, distinct subsets of candidate actions tailored to their specific emergent roles. Conversely, a shared codebook requires all agent to draw from a common pool of candidate actions. While this shared approach can accelerate learning in symmetric, highly homogeneous environments by aggregating gradient updates into a single unified codebook, we generally advocate for the per-agent instantiation. As observed previously (Section 4.1), agents tend to have different usages of the continuous action space, therefore maintaining individualized codebooks empowers each agent to optimally compress the action space and tailor its candidate actions to the unique strategic requirements of its own optimal policy.

5 EXPERIMENTS

To rigorously evaluate the proposed continuous adaptations, we benchmark them on environments from the SISL suite within the PettingZoo library [7, 22]. These environments require tight multi-agent coordination and precise continuous actuation. The source code of the algorithms and experiments presented in this work is available in a public repository [1].

5.1 Environments

The Multiwalker environment [22] is a popular cooperative, physics-based task where two bipedal walkers must coordinate to balance and transport a package across a uneven terrain. The agents share a collective reward based on the package's forward displacement. The action space is continuous $\mathbb{R}^4$, representing the torque applied to the two joints in each of the agent's two legs.

Waterworld [22] is a continuous simulation of swarm behavior modeling particles in a fluid environment, where we deploy two

[1] https://github.com/nico1740/Adapting_LAN_for_Continuous_MARL.git

pursuer agents and one evader. The team of agents must coordinate to encircle and consume moving food targets while actively avoiding poison. The action space is continuous $\mathbb{R}^2$, representing a thrust vector that allows for smooth movement through the fluid.

5.2 Algorithm Implementation and Experimental Settings

All algorithms are implemented using the PyTorch framework with the software environment standardized on Python 3.11.3 and PyTorch 2.3.0. Experimental evaluations are conducted on a cluster utilizing Slurm for job scheduling and resource management. The workload is distributed across a heterogeneous set of AMD EPYC nodes, specifically Zen 5 Turing and Zen 4 Genoa-X processors. Each individual training job is typically allocated 16 CPU cores to ensure stable and efficient processing.

To evaluate the effectiveness of our proposed continuous action methodologies, we compared four distinct algorithms:

(1) **Bang-Bang LAN:** This variant represents the most extreme form of discretization. Based on the paper [21], we restrict the agent's choices solely to the boundaries of the continuous action space. For an action dimension d, the action set consists only of the permutations of the extreme values $\{-1, 1\}^d$. This approach tests whether simple "Bang-Bang" control is sufficient for solving the given multi-agent tasks.

(2) **MAPPO:** As a strong, standard baseline for on-policy multi-agent reinforcement learning in continuous spaces, we use the official MAPPO implementation.[2] The code was adapted from its standard GitHub repository to interface correctly with our specific environments while maintaining its core hyperparameters and training procedures.

(3) **QF-LAN (Q-functional LAN):** This architecture models the advantage function as a continuous surface using a Legendre polynomial basis [13]. The rank chosen was 5, in both cases, the maximization step approximated the global optimum by evaluating 1.000 random samples.

(4) **MQ-LAN (MovingQ LAN):** This variant employs a quantization approach to handle continuous actions. We configured the algorithm with $K = 100$ discrete action slots to approximate the continuous dimension. Furthermore, to manage potential conflicts when multiple agents attempt to select similar quantized actions, we incorporated a "clash" parameter set to 0.05, defining the sensitivity threshold for action proximity.

5.3 Results and Discussion

To rigorously evaluate the proposed MQ-LAN architecture, we present aggregate metrics across both environments, including sample efficiency curves, performance profiles, and point estimates (Interquartile Mean, Median, and Mean normalized returns) alongside 95% stratified bootstrap confidence intervals [1].

The empirical results deliver a comprehensive validation of the MQ-LAN architecture and the overarching CTDE[2] paradigm. As illustrated in Figure 3, MQ-LAN consistently dominates all aggregate statistical measures. Specifically, MQ-LAN achieves an Interquartile Mean (IQM) return of approximately 0.85, marking a substantial improvement over the standard on-policy MAPPO algorithm (0.60) and the rigidly discretized Bang-Bang LAN baseline (0.55). Moreover, MQ-LAN yields the narrowest Optimality Gap, underscoring its reliability in consistently converging to the best-performing policies compared to alternative architectures.

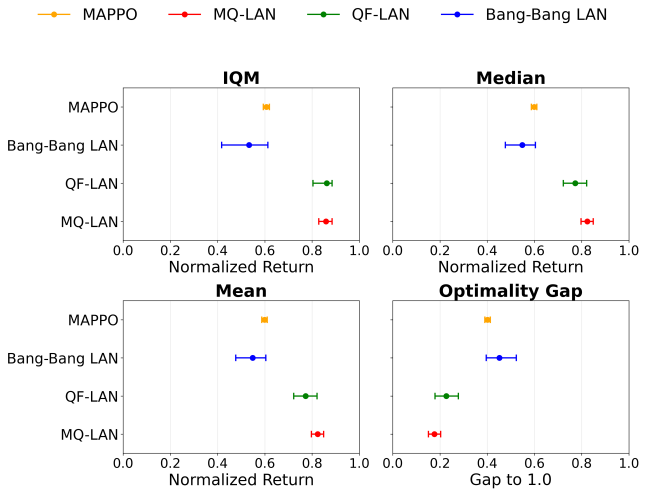


Figure 3: Points Estimates

We further illustrate the training dynamics and algorithmic robustness in Figure 4. Although MQ-LAN exhibits a brief performance dip during early training phases it quickly recovers, demonstrating a steep performance trajectory that ultimately surpasses the sample efficiency of both MAPPO and QF-LAN. The Performance Profiles corroborate this resilience, showing that MQ-LAN sustains a higher fraction of successful runs across increasingly stringent return thresholds.

The statistical significance of these performance gains is detailed in Figure 5, where MQ-LAN consistently demonstrates a strong probability of improvement over each baseline algorithm. Compared to Bang-Bang LAN, MQ-LAN shows a clear advantage with a probability of improvement of approximately 0.60. While QF-LAN represents a robust value based approach, it is nevertheless surpassed by MQ-LAN. The high probability that MQ-LAN outperforms QF-LAN (approximately 0.71) solidifies the premise that dynamic, learnable discretization provides a more effective mechanism for continuous action selection than exhaustively sampling the advantage function. Finally, MQ-LAN exhibits near absolute dominance over MAPPO, achieving a probability of improvement of approximately 1.0, further emphasizing the efficacy of the proposed method in this continuous action setting.

Task-specific analyses of the Multiwalker and Waterworld environments (detailed in Appendix Figures 8 and 9) further emphasize the versatility of MQ-LAN. In the Waterworld environment, which demands highly precise coordination for swarm encirclement, MQ-LAN approaches optimal performance while the Bang-Bang LAN baseline fails completely. This divergence indicates that extreme

[2]https://github.com/marlbenchmark/on-policy

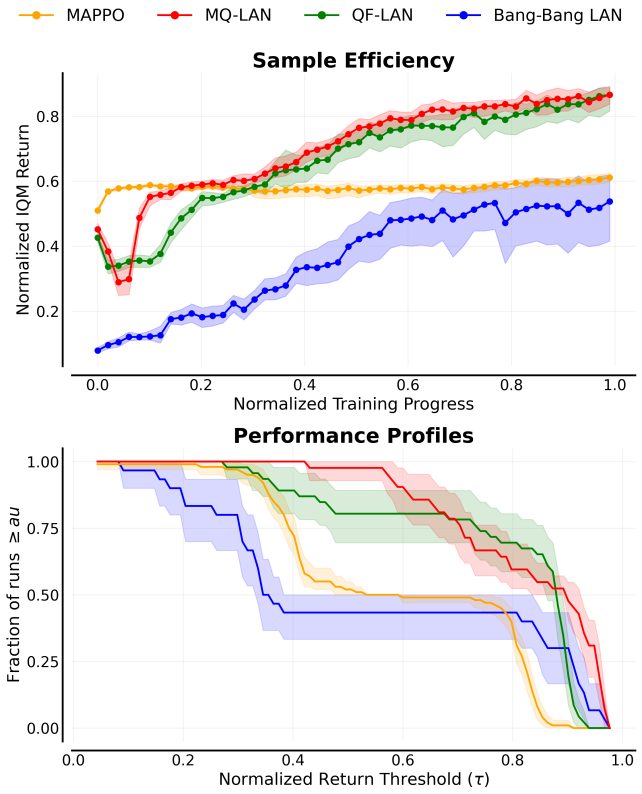


Figure 4: Sample Efficiency and Performance Profiles

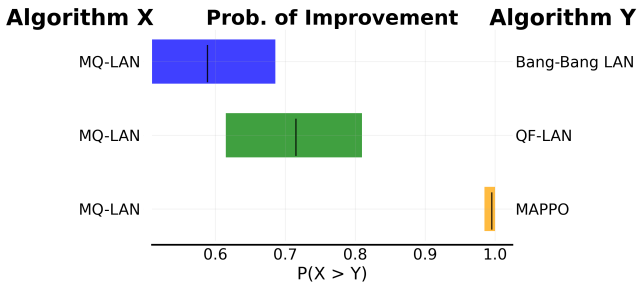


Figure 5: Probabilistic Comparison of Improvement

boundary actions are critically insufficient for nuanced thrust control, whereas the dynamic codebook of MQ-LAN successfully captures essential high-value regions of the continuous action space.

Conversely, in the physics-based Multiwalker task, while MQ-LAN demonstrates a highly stable learning trajectory, it is closely outperformed by QF-LAN and significantly beaten by Bang-Bang LAN. We hypothesize that because the Multiwalker environment requires more extreme actions for fast-paced package delivery, it benefits from the simple discretization of Bang-Bang control. This contrast highlights that while some environments necessitate highly nuanced actions, others perform exceptionally well with extreme boundary values. By dynamically adapting to these varying requirements, MQ-LAN positions itself as a robust solution capable of autonomously discovering the most effective action codebook for the given task. Together, these findings validate the CTDE^2 paradigm as a highly scalable and computationally efficient approach for high-dimensional, continuous multi-agent reinforcement learning.

5.4 MQ-LAN's Learned Codebooks

To validate the effectiveness of the MovingQ mechanism, we visually analyze the final spatial distribution of the action embeddings learned by the MQ-LAN agents in the Waterworld environment. By comparing these generated codebooks against the marginalized empirical state-action distribution (Figure 1), we can determine if the algorithm successfully learned an environmentally relevant codebook, as predicted by the best action reusability hypothesis.

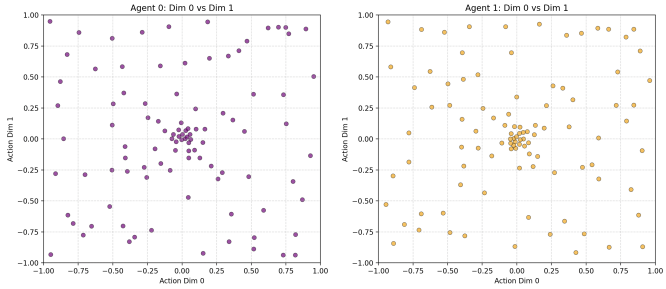


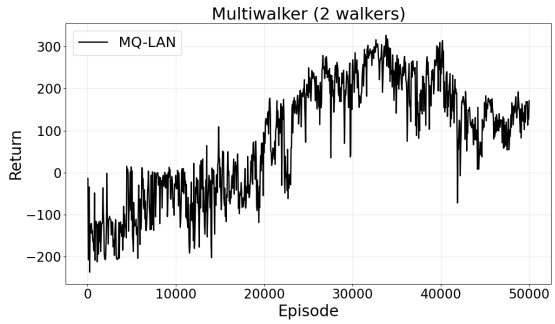
Figure 6: Spatial distribution of the action embeddings learned by the MQ-LAN agents in the Waterworld environment.

Examining the scatter plots in Figure 6 of the learned codebooks reveals a striking structural alignment with the empirical baseline. The MQ-LAN candidate actions do not form a rigid or evenly spaced grid. Instead, a highly dense cluster of action embeddings has dynamically migrated directly to the origin (0, 0), perfectly capturing the swarm agent's frequent need for idle states or minute, precise adjustments within the fluid environment. This directly mirrors the massive, bright concentration of optimal actions found at the center of the empirical heatmap.

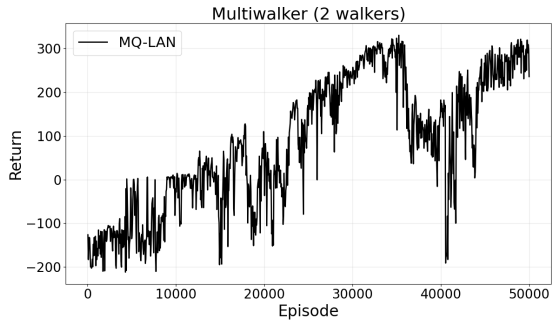
Furthermore, the algorithm's diversity mechanisms have efficiently distributed the remaining computational budget outward. The peripheral action embeddings are strategically positioned, with several finding their way to the extreme corners and boundaries to accommodate necessary maximum-thrust evasive or pursuing maneuvers. The spatial configuration confirms that the CTDE^2 paradigm successfully drives agents to abandon unhelpful continuous regions, crystallizing a specialized, high-density action codebook exactly where the learned optimal policy's value landscape demands it.

5.5 Catastrophic Collapse & Future Improvements

While MQ-LAN discovers highly efficient policies, empirical evaluations reveal a vulnerability to a catastrophic collapse phenomenon. In certain runs, agents converge to near-optimal policies before experiencing a sudden, severe performance degradation from which recovery is inconsistent (Figures 7a–7b).



(a)



(b)

Figure 7: Individual MQ-LAN runs suffering from catastrophic collapse in the Multiwalker environments.

We attribute this instability to MovingQ's hard gradient updates rather than the underlying CTDE^2 paradigm. Abruptly shifting established action embeddings deprives the agent of the specific codebook responsible for its prior success, forcing the critic to drastically re-adapt its value estimations. While high action mobility benefits early exploration, it destabilizes the delicate fine-tuning required in later training stages.

To validate this hypothesis, we evaluated a simplified, evolution-based variant of MQ-LAN. By entirely disabling gradient updates, this variant relies exclusively on random spawning and a starvation mechanism to prune uncompetitive embeddings. Across all experimental runs, this approach successfully prevented catastrophic collapse. While it did not reach the absolute peak performance of the standard gradient-driven MQ-LAN, it confirms that hard parameter migration is the primary driver of the observed instability. The evolution-based algorithm exhibits a notably smaller standard deviation, directly reflecting the absence of sudden performance drops (see Appendix 11 for full learning trajectories). Consequently, these findings highlight evolutionary stabilization as a highly promising direction for future refinements of the CTDE^2 paradigm.

6 CONCLUSION

The transition of value-based Multi-Agent Reinforcement Learning (MARL) to continuous action spaces has historically been hindered by the maximization bottleneck, forcing a trade-off between computational tractability and model expressivity. While existing literature attempts to bridge this gap, expressive architecture inevitably necessitate computationally expensive sampling densities [4, 13] or complex online optimization [20]. However, we observed that agents operating in continuous domains do not uniformly exploit the infinite action space; rather, they naturally gravitate toward a specific, localized set of action regions dictated by their emergent roles. We empirically confirmed the *best action reusability* hypothesis by analyzing the marginalized spatial distribution of optimal actions, which revealed highly dense, specialized clusters rather than broad continuous usage.

Based on this observation, we formulated the Continuous Training with Discrete Execution (CTDE^2) paradigm, a foundational approach posting that an optimal continuous policy can be arbitrarily closely approximated by a compact, reusable codebook of discrete actions. CTDE^2 decouples the complexity of continuous learning from execution by treating the discretization of the action space as a learnable parameter, allowing agents to simultaneously learn the optimal discrete action set and the optimal policy.

Finally, we operationalized this paradigm through the MovingQ algorithm, integrating it into the Local Advantage Networks (LAN) [2] framework to create MQ-LAN. By replacing exhaustive continuous sampling with a dynamic population of action embeddings that migrate via gradient ascent, MQ-LAN bypasses the traditional computational burden. Empirical evaluations in the Multiwalker and Waterworld environment confirm that MQ-LAN achieves superior computational efficiency and higher final returns compared to rigid discrete baselines, standard Actor-Critic methods, and Q-functionals (QF-LAN). Ultimately, this research establishes CTDE^2 as a highly scalable, efficient, and precise pathway for high-dimensional continuous multi-agent control.

6.1 Future Work

To address the vulnerabilities identified in MQ-LAN, notably the catastrophic collapse anomaly, and to further expand the CTDE^2 paradigm, promising research directions include:

(1) **Evolutionary Stabilization:** Catastrophic collapse could be mitigated by augmenting gradient ascent with an evolutionary approach: anchoring original action embeddings while their copies explore the gradient, and relying on the existing starvation mechanism to prune uncompetitive actions.
(2) **Discrete Set of Localized Q-functionals:** A hybrid approach could maintain a discrete codebook of local Q-functions rather than individual discrete action embeddings. While this slightly increases computational cost at inference, it grants significantly higher continuous precision within critical, high-value regions of the action space.

ACKNOWLEDGMENTS

The basis for this work was the MSc thesis of NR [19]. DR was funded by the European Union's Horizon Europe Research and Innovation Programme, under Grant Agreement number 101120406 (PEER). This work was realized under F. Delgrange's VUB OZR mandate (VUB-OZR4417) and was supported by the "DESCARTES" iBOF project. The paper reflects only the authors' view and the EC is not responsible for any use that may be made of the information it contains. We acknowledge Raphaël Avalos for making the original LAN code available to us.

REFERENCES

[1] Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C. Courville, and Marc G. Bellemare. 2021. Deep Reinforcement Learning at the Edge of the Statistical Precipice. (2021), 29304–29320. https://proceedings.neurips.cc/paper/2021/hash/f514cec81cb148559cf475e7426eed5e-Abstract.html

[2] Raphaël Avalos, Mathieu Reymond, Ann Nowé, and Diederik M. Roijers. 2023. Local Advantage Networks for Multi-Agent Reinforcement Learning in Dec-POMDPs. *Trans. Mach. Learn. Res.* 2023 (2023). https://openreview.net/forum?id=adpKzWQunW

[3] Andrea Castelletti, Stefano Galelli, Marcello Restelli, and Rodolfo Soncini-Sessa. 2010. Tree-based reinforcement learning for optimal water reservoir operation. *Water Resources Research* 46, 9 (2010).

[4] Yasin Findik and Seyed Reza Ahmadzadeh. 2024. Mixed Q-Functionals: Advancing Value-Based Methods in Cooperative MARL with Continuous Action Domains. *CoRR* abs/2402.07752 (2024). https://doi.org/10.48550/ARXIV.2402.07752 arXiv:2402.07752

[5] Scott Fujimoto, Herke van Hoof, and David Meger. 2018. Addressing Function Approximation Error in Actor-Critic Methods. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018 (Proceedings of Machine Learning Research)*, Jennifer G. Dy and Andreas Krause (Eds.). PMLR, 1582–1591. http://proceedings.mlr.press/v80/fujimoto18a.html

[6] Shixiang Gu, Timothy Lillicrap, Ilya Sutskever, and Sergey Levine. 2016. Continuous deep q-learning with model-based acceleration. In *International Conference on Machine Learning*. PMLR, 2829–2838.

[7] Jayesh K. Gupta, Maxim Egorov, and Mykel Kochenderfer. 2017. Cooperative Multi-Agent Control Using Deep Reinforcement Learning. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. Springer, 66–83.

[8] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018 (Proceedings of Machine Learning Research)*, Jennifer G. Dy and Andreas Krause (Eds.). PMLR, 1856–1865. http://proceedings.mlr.press/v80/haarnoja18b.html

[9] Jakub Grudzien Kuba, Ruiqing Chen, Muning Wen, Ying Wen, Fanglei Sun, Jun Wang, and Yaodong Yang. 2022. Trust Region Policy Optimisation in Multi-Agent Reinforcement Learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net. https://openreview.net/forum?id=EcGGFkNTxdJ

[10] Michael Laskin, Aravind Srinivas, and Pieter Abbeel. 2020. CURL: Contrastive Unsupervised Representations for Reinforcement Learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event (Proceedings of Machine Learning Research)*. PMLR, 5639–5650. http://proceedings.mlr.press/v119/laskin20a.html

[11] Michael L. Littman. 1994. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the 11th International Conference on Machine Learning (ICML)*. Morgan Kaufmann, 157–163.

[12] Samuel Lobel, Sreehari Rammohan, Bowen He, Shangqun Yu, and George Konidaris. 2023. Q-functionals for Value-Based Continuous Control. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, Brian Williams, Yiling Chen, and Jennifer Neville (Eds.). AAAI Press, 8932–8939. https://doi.org/10.1609/AAAI.V37I7.26073

[13] Samuel Lobel, Sreehari Rammohan, Bowen He, Shangqun Yu, and George Konidaris. 2023. Q-functionals for Value-Based Continuous Control. (2023), 8932–8939. https://doi.org/10.1609/AAAI.V37I7.26073

[14] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in neural information processing systems*, Vol. 30.

[15] Frans A. Oliehoek and Christopher Amato. 2016. *A Concise Introduction to Decentralized POMDPs*. Springer. https://doi.org/10.1007/978-3-319-28929-8

[16] Ramon Petri, Eugenio Bargiacchi, Huib Aldewereld, and Diederik M. Roijers. 2021. Heuristic Coordination in Cooperative Multi-Agent Reinforcement Learning. In *Proceedings of the 33rd Benelux Conference on Artificial Intelligence and the 30th Belgian-Dutch Conference on Machine Learning (BNAIC/BeneLearn)*. 90–104. https://bnaic2021.uni.lu/program/

[17] Tabish Rashid, Bei Peng, and Shimon Whiteson. 2021. Continuous Coordination: Extending QMIX to Continuous Action Spaces. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[18] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder de Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2018. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. In *International Conference on Machine Learning (ICML)*. PMLR, 4295–4304.

[19] Nicolas Rowies. 2026. *Adapting Local Advantage Networks for Continuous Multi-Agent Control: Continuous Training with Discrete Execution & Q-functionals*. Master's thesis. Vrije Universiteit Brussel (VUB).

[20] Moonkyung Ryu, Yinlam Chow, Ross Anderson, Christian Tjandraatmadja, and Craig Boutilier. 2020. CAQL: Continuous Action Q-Learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. https://openreview.net/forum?id=BkxXe0Etwr

[21] Tim Seyde, Igor Gilitschenski, Wilko Schwarting, Bartolomeo Stellato, Martin A. Riedmiller, Markus Wulfmeier, and Daniela Rus. 2021. Is Bang-Bang Control All You Need? Solving Continuous Control with Bernoulli Policies. (2021), 27209–27221. https://proceedings.neurips.cc/paper/2021/hash/e46be61f0050f9cc3a98d5d2192cb0eb-Abstract.html

[22] J. K. Terry, Benjamin Black, Nathaniel Grammel, Mario Jayakumar, Ananth Hari, Ryan Sullivan, Luis S. Santos, Clemens Dieffendahl, Caroline Horsch, Rodrigo Perez-Vicente, Niall L. Williams, Yashas Lokesh, and Praveen Ravi. 2021. PettingZoo: Gym for Multi-Agent Reinforcement Learning. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (Eds.). 15032–15043. https://proceedings.neurips.cc/paper_files/paper/2021/hash/803f7c4c3ff61b71be53a0c803bfb57f-Abstract.html

[23] Chen Tessler, Guy Tennenholtz, and Shie Mannor. 2019. Distributional Policy Optimization: An Alternative Approach for Continuous Control. (2019), 1350–1360. https://proceedings.neurips.cc/paper/2019/hash/72da7fd6d1302c0a159f6436d01e9eb0-Abstract.html

[24] Alexander Pritzel Nicolas Heess Tom Erez Yuval Tassa David Silver Daan Wierstra Timothy P. Lillicrap, Jonathan J. Hunt. 2016. Continuous control with deep reinforcement learning. *ICLR* (2016).

[25] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2017. Neural Discrete Representation Learning. *CoRR* abs/1711.00937. arXiv:1711.00937 http://arxiv.org/abs/1711.00937

[26] Jianhao Wang, Zongqian Ren, Terry Liu, Jack Gelfand, and Shimon Whiteson. 2021. QPLEX: Duplex Dueling Multi-Agent Q-Learning. In *International Conference on Learning Representations (ICLR)*.

[27] Muning Wen, Jakub Grudzien Kuba, Runji Lin, Weinan Zhang, Ying Wen, Jun Wang, and Yaodong Yang. 2022. Multi-Agent Reinforcement Learning is a Sequence Modeling Problem. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (Eds.). http://papers.nips.cc/paper_files/paper/2022/hash/69413f87e5a34897cd010ca698097d0a-Abstract-Conference.html

[28] Yuhuai Wu, Elman Mansimov, Roger B. Grosse, Shun Liao, and Jimmy Ba. 2017. Scalable trust-region method for deep reinforcement learning using Kronecker-factored approximation. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 5279–5288. https://proceedings.neurips.cc/paper/2017/hash/361440528766bbaaaa1901845cf4152b-Abstract.html

[29] Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre Bayen, and Yi Wu. 2021. The surprising effectiveness of ppo in cooperative, multi-agent games. *arXiv preprint arXiv:2103.01955* (2021).

APPENDIX

Extra Figures

Figures 8 and 9 depict sample efficiency plots (median with 25–75 confidence interval) environment-wise.

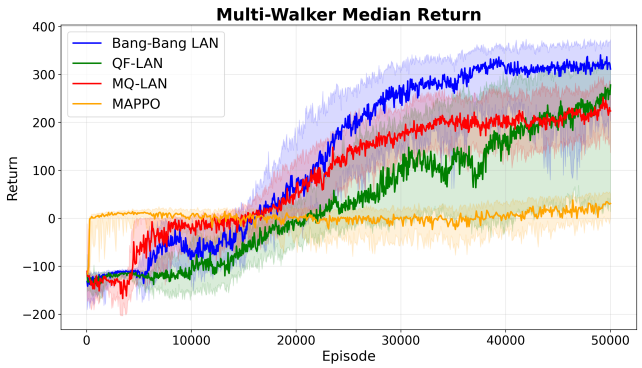


Figure 8: Comparative learning curves on the Multiwalker environment (Median)

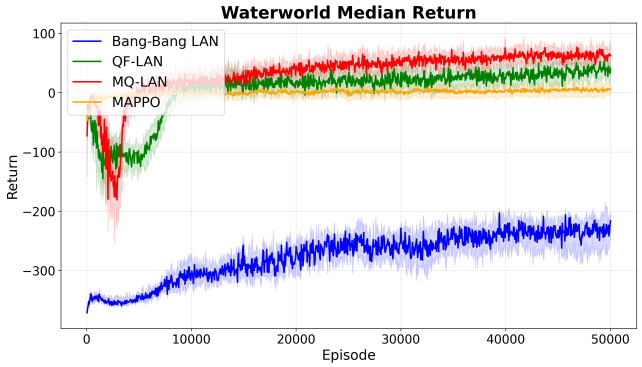


Figure 9: Comparative learning curves on the Waterworld environment (Median)

Figure 11 illustrates the comparative learning trajectories of the standard gradient-driven MQ-LAN and the simplified, evolution-based variant over the course of training. The shaded regions denote the standard deviation across multiple independent experimental runs.

Notably, the evolution-only method displays a significantly tighter variance and maintains a stable lower bound throughout training. By relying exclusively on random spawning and a starvation mechanism, the evolutionary variant entirely avoids the sharp, catastrophic drops in reward that characterize the baseline model's instability. While the peak reward achieved is marginally lower than that of the gradient-updated model under optimal conditions, the absence of sudden performance degradation visually corroborates the hypothesis that hard parameter migration is the primary catalyst for catastrophic collapse in the standard architecture.

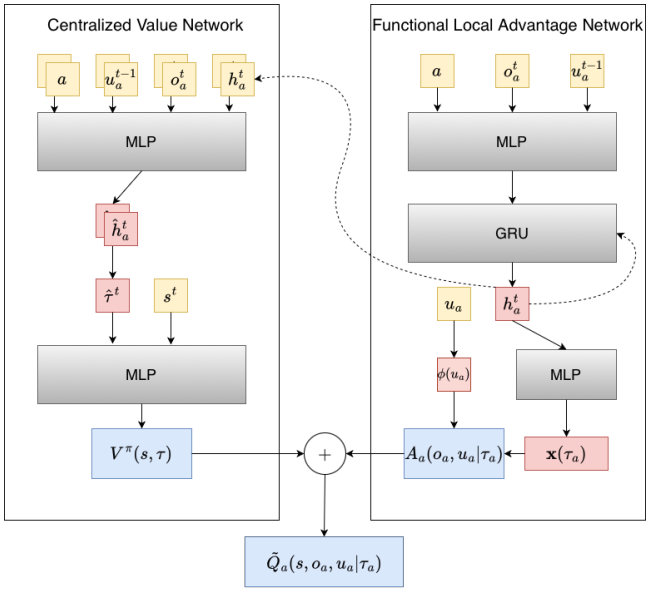


Figure 10: Overview of the QF-LAN architecture

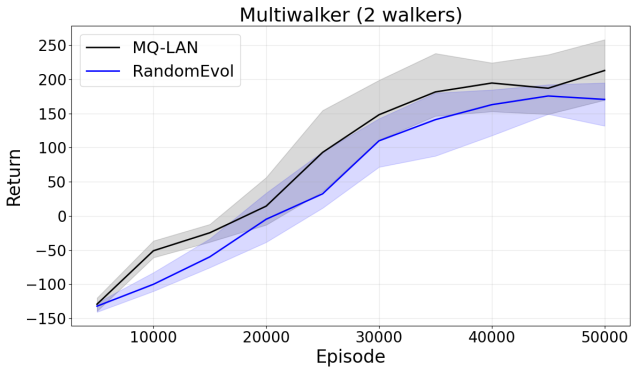


Figure 11: Comparative learning curves of MQ-LAN and a Naive Random Evolution algorithm to try to solve the Catastrophic Collapse anomalies.

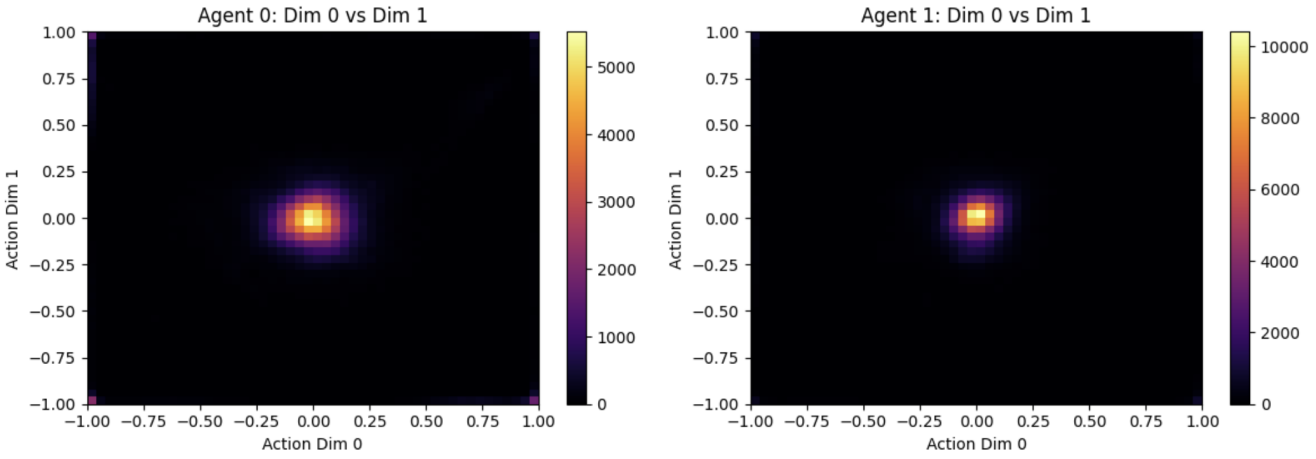


Figure 12: Action distribution of an optimal policy marginalized over states in the Waterworld environment. (Larger size)

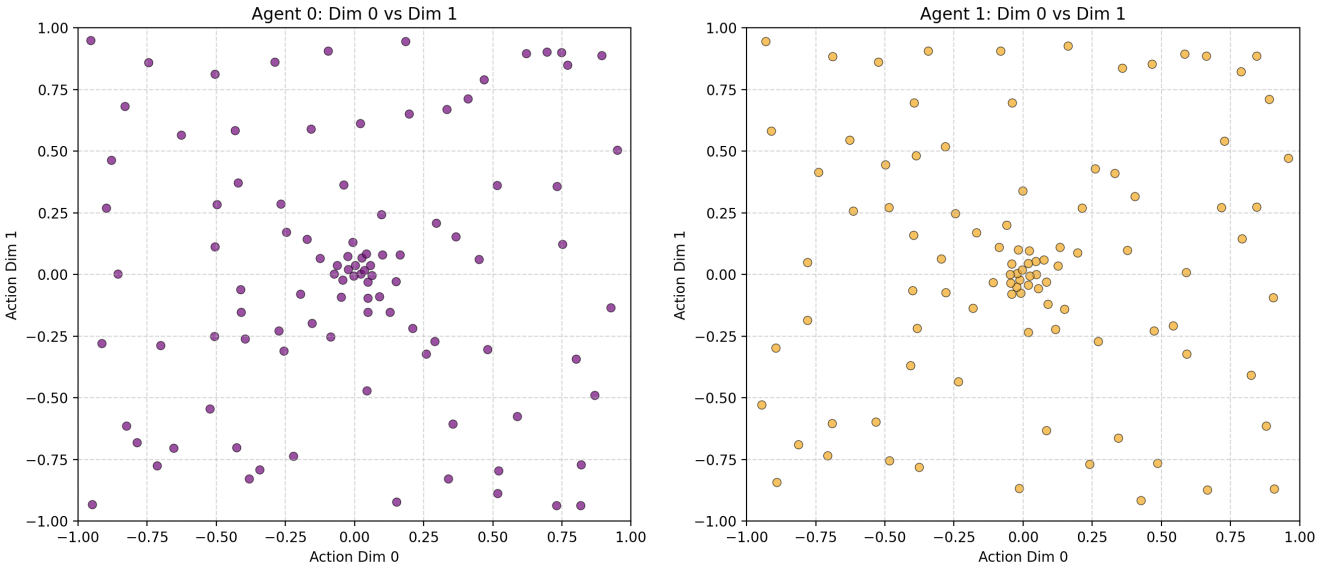


Figure 13: Spatial distribution of the action embeddings learned by the MQ-LAN agents in the Waterworld environment. (Larger size)