Growing Q-Networks: Solving Continuous Control Tasks with Adaptive Control Resolution

Anonymous Author(s)
Affiliation
Address
email

Abstract

Recent reinforcement learning approaches have shown surprisingly strong capabilities of bang-bang policies for solving continuous control benchmarks. The underlying coarse action space discretizations often yield favorable exploration characteristics, while final performance does not visibly suffer in the absence of action penalization in line with optimal control theory. In robotics applications, smooth control signals are commonly preferred to reduce system wear and improve energy efficiency, while regularization via action costs can be detrimental to exploration. Our work aims to bridge this performance gap by growing discrete action spaces from coarse to fine control resolution. We take advantage of recent results in decoupled Q-learning to scale our approach to high-dimensional action spaces up to $dim(\mathcal{A}) = 38$. Our work indicates that an adaptive control resolution in combination with value decomposition yields simple critic-only algorithms that enable surprisingly strong performance on continuous control tasks.

1 Introduction

Reinforcement learning for continuous control applications commonly leverages policies parameterized via continuous distributions. Recent works have shown surprisingly strong performance of discrete policies in the actor-critic and critic-only setting [Tang and Agrawal, 2020, Tavakoli et al., 2021, Seyde et al., 2021]. While discrete critic-only methods promise simpler controller designs than their continuous actor-critic counterparts, applications such as robot control tend to favor smooth control signals to maintain stability and prevent system wear [Hodel, 2018]. It has previously been noted that coarse action discretization can provide exploration benefits early during training [Czarnecki et al., 2018, Farquhar et al., 2020], while converged policies should increasingly prioritize controller smoothness [Bohez et al., 2019].

Our work aims to bridge the gap between these two objectives while maintaining algorithm simplicity. We introduce Growing Q-Networks (GQN), a simple discrete critic-only agent that combines the scalability benefits of fully decoupled Q-learning [Seyde et al., 2022b] with the exploration benefits of dynamic control resolution [Czarnecki et al., 2018, Farquhar et al., 2020]. Introducing an adaptive action masking mechanism into a value-decomposed Q-Network, the agent can autonomously decide when to increase control resolution. This approach enhances learning efficiency and balances the exploration-exploitation trade-off more effectively, improving convergence speed and solution smoothness. The primary contributions of this paper are threefold:

- **A framework for adaptive control resolution:** we grow control resolution from coarse to fine within decoupled Q-learning. This reconciles coarse exploration during early training with smooth control at convergence, retaining the scaling properties of decoupled control.

Submitted to the ICML 2024 Workshop on Foundations of Reinforcement Learning and Control. Do not distribute.

- **Insights into the scalability of discretized control:** our research provides valuable insights into overcoming exploration challenges in soft-contrained continuous control settings via simple discrete Q-learning methods, studying applicability in challenging control scenarios.
- **Comprehensive experimental validation:** we validate the effectiveness of our GQN algorithm on a diverse set of continuous control tasks, highlighting the benefits of adaptive control resolution over static DQN variations and recent continuous actor-critic methods.

The remainder of the paper is organized as follows: Section 2 reviews related work, Section 3 introduces preliminaries, Section 4 details the proposed GQN methodology, Section 5 presents experimental results, and Section 6 concludes with a discussion on future research directions.

2 Related Works

In the following, we discuss several key related works grouped by their primary research thrust.

Discretized Control Learning continuous control tasks commonly relies on policies with continuous support, primarily Gaussians with diagonal covariance matrices [Schulman et al., 2017, Haarnoja et al., 2018, Abdolmaleki et al., 2018a, Hafner et al., 2020, Wulfmeier et al., 2020]. Recent works have shown that competitive performance is often attainable via discrete policies [Tavakoli et al., 2018, Neunert et al., 2020, Tang and Agrawal, 2020, Seyde et al., 2022a] with bang-bang control at the extreme [Seyde et al., 2021]. Bang-bang controllers have been extensively investigated in optimal control research [Sonneborn and Van Vleck, 1964, Bellman et al., 1956, LaSalle, 1959, Maurer et al., 2005] as well as early works in reinforcement learning [Waltz and Fu, 1965, Lambert and Levine, 1970, Anderson, 1988], while the extreme switching behavior was often observed to naturally emerge even under continuous policy distributions [Huang et al., 2019, Novati and Koumoutsakos, 2019, Thuruthel et al., 2019]. The direct application of discrete action-space algorithms then harbors potential benefits for reducing model complexity [Metz et al., 2017, Sharma et al., 2017, Tavakoli, 2021, Watkins and Dayan, 1992], although control resolution trade-offs and scalability may require computational overhead [Van de Wiele et al., 2020].

Scalability The scalability of Q-learning approaches has been studied extensively in the context of mitigating coordination challenges and system non-stationarity [Tan, 1993, Claus and Boutilier, 1998, Matignon et al., 2012, Lauer and Riedmiller, 2000, Matignon et al., 2007, Foerster et al., 2017, Busoniu et al., 2006, Böhmer et al., 2019]. Exponential coupling can be avoided by information-sharing [Schneider et al., 1999, Russell and Zimdars, 2003, Yang et al., 2018], composition of local utility functions [Sunehag et al., 2017, Rashid et al., 2018, Son et al., 2019, Wang et al., 2020, Su et al., 2021, Peng et al., 2021], and considering different levels of interaction [Guestrin et al., 2002, Kok and Vlassis, 2006]. Centralization can further be facilitated via high degrees of parameter-sharing [Gupta et al., 2017, Böhmer et al., 2020, Christianos et al., 2021, Van Seijen et al., 2017, Chu and Ye, 2017]). Decoupled control via Q-learning was proposed for Atari [Sharma et al., 2017] and extended to mixing across higher-order action subspaces [Tavakoli et al., 2021], with decoupled bang-bang control displaying strong performance on continuous control tasks [Seyde et al., 2022b]. While coarse discretization can benefit exploration, particularly in the presence of action penalties, it may also reduce steady-state performance. Conversely, fine discretization can exacerbate coordination challenges [Seyde et al., 2022b, Ireland and Montana, 2024]. Here, we consider adapting the control resolution over the course of training to achieve the best of both worlds.

Expanding Action Spaces Smith et al. [2023] present an adaptive policy regularization approach that introduces soft constraints on feasible action regions, growing continuous regions linearly over the course of training with adjustments based on dynamics uncertainty. They focus on learning quadrupedal locomotion on hardware and expand locally around joint angles of a stable initial pose. In discrete action spaces, one can instead leverage iterative resolution refinement. Czarnecki et al. [2018] consider DeepMind Lab navigation tasks [Beattie et al., 2016] with a natively discrete action space that avoids reasoning about system dynamics stability. Their policy-based method formulates a mixture policy optimized under a distillation objective to facilitate knowledge transfer, adjusting the mixing weights via Population Based Training (PBT) [Jaderberg et al., 2017]. Similarly, Synnaeve et al. [2019] consider multi-agent coordination in StarCraft and adjust spatial command resolution via PBT. Farquhar et al. [2020] grow action resolution under a linear growth schedule while showing limited application to simple continuous control tasks, as they enumerate the action space and do not consider decoupled optimization. Beyond control applications, Yang et al. [2023] demonstrate adaptive mesh refinement strategies that reduce the errors in finite element simulations. Their refinement policy recursively adds finer elements, expanding the action space.

Constrained Optimization Reward-optimal bang-bang policies may not be desirable for real-world applications as they can be less energy efficient and increase wear and tear on physical systems, e.g., Hodel [2018]. In the past, this behavior was generally avoided by employing penalty functions as soft constraints at the cost of potentially hindering exploration or enabling reward hacking [Skalse et al., 2022]. The rewards and costs are automatically re-balanced to combat this issue in Bohez et al. [2019]. Similarly, undesirable behaviors are avoided by automatically balancing soft chance constraints with the primary rewards in Roy et al. [2021]. Here, we do not assume access to explicit penalty terms and efficiently learn controllers directly based on environment reward.

3 Preliminaries

We formulate the learning control problem as a Markov Decision Process (MDP) described by the tuple $\{\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma\}$, where $\mathcal{S} \subset \mathbb{R}^N$ and $\mathcal{A} \subset \mathbb{R}^M$ denote the state and action space, respectively, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ the transition distribution, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ the reward function, and $\gamma \in [0, 1)$ the discount factor. Let s_t and a_t denote the state and action at time t, where actions are sampled from policy $\pi(a_t|s_t)$. We define the discounted infinite horizon return as $G_t = \sum_{\tau=t}^{\infty} \gamma^{\tau-t} R(s_\tau, a_\tau)$, where $s_{t+1} \sim \mathcal{T}(\cdot|s_t, a_t)$ and $a_t \sim \pi(\cdot|s_t)$. Our objective is to learn the optimal policy that maximizes the expected infinite horizon return $\mathbb{E}[G_t]$ under unknown dynamics and reward mappings. Conventional algorithms for continuous control settings leverage actor-critic designs with a continuous policy $\pi_\phi(a_t|s_t)$ maximizing expected returns from a value estimator $Q_\theta(s_t, a_t)$ or $V_\theta(s_t)$. Recent studies have shown strong results with simpler methods employing discretized actors [Tang and Agrawal, 2020, Seyde et al., 2021] or critic-only formulations [Tavakoli et al., 2018, 2021, Seyde et al., 2022b]. Here, we focus on the light-weight critic-only setting and increase control resolution over the course of training to bridge the gap between discrete and continuous control.

3.1 Deep Q-Networks

We consider the general framework of Deep Q-Networks (DQN) [Mnih et al., 2013], where the state-action value function $Q_\theta(s_t, a_t)$ is represented by a neural network with parameters θ. The parameters are updated to minimize the temporal-difference (TD) error, where we leverage several performance enhancements based on the Rainbow agent [Hessel et al., 2018]. These include target networks to improve stability in combination with double Q-learning to mitigate overestimation [Mnih et al., 2015, Van Hasselt et al., 2016], prioritized experience replay (PER) to focus sampling on more informative transitions [Schaul et al., 2015], and multi-step returns to improve stability of Bellman backups [Sutton and Barto, 2018]. The resulting objective function is given by

$$\mathcal{L}(\theta) = \sum_{b=1}^{B} L_\delta(y_t - Q_\theta(s_t, a_t)), \tag{1}$$

where action evaluation employs the target $y_t = \sum_{j=0}^{n-1} \gamma^j r(s_{t+j}, a_{t+j}) + \gamma^n Q_{\theta^-}\big(s_{t+n}, a^*_{t+n}\big)$, action selection uses $a^*_{t+1} = \arg\max_a Q_\theta(s_{t+1}, a)$, $L_\delta(\cdot)$ is the Huber loss and the batch size is B. Here, we leverage a target network with parameters Q_{θ^-} to further enhance learning stability.

3.2 Decoupled Q-Networks

Traditional DQN-based agents enumerate the entire action space and do not scale well to high dimensional control problems. Decoupled representations address scalability issues by treating subsets of action dimensions as separate agents and coordinating joint behavior in expectation [Sharma et al., 2017, Sunehag et al., 2017, Rashid et al., 2018, Tavakoli et al., 2021, Seyde et al., 2022b]. The Decoupled Q-Networks (DecQN) agent introduced in Seyde et al. [2022b] employs a complete decomposition with the critic predicting univariate utilities for each action dimension a^j conditioned

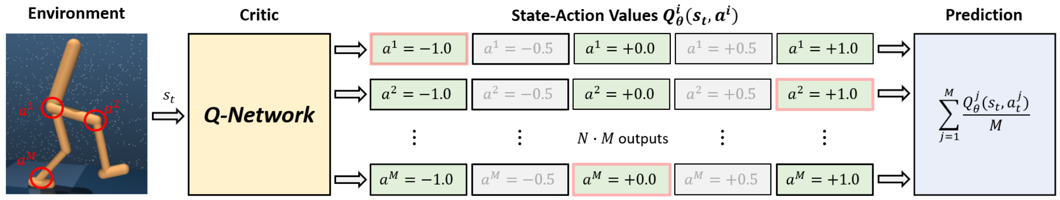

Figure 1: Schematic of a GQN agent with decoupled 5-bin discretization and 3-bin active subspace. The available actions are highlighted in green while the masked actions are depicted in gray. The predicted state-action values $Q(\boldsymbol{s}, a^0, ..., a^M)$ are computed via linear composition of the univariate utilities $Q(\boldsymbol{s}, a^j)$ by selecting one action per dimension (red). We consider a homogeneous discretization across action dimensions for simplicity, heterogeneous discretization are also feasible.

on the global state $\boldsymbol{s}$. The corresponding state-action value function is recovered as

$$Q_\theta(\boldsymbol{s}_t, \boldsymbol{a}_t) = \sum_{j=1}^{M} \frac{Q_\theta^j(\boldsymbol{s}_t, a_t^j)}{M}, \tag{2}$$

where the objective is analogous to Eq. 1, enabling centralized training with decentralized execution.

4 Growing Q-Networks

Discrete control algorithms have demonstrated competitive performance on continuous control benchmarks [Tang and Agrawal, 2020, Tavakoli et al., 2018, Seyde et al., 2021]. One potential benefit of these methods is the intrinsic coarse exploration that can accelerate the generation of informative environment feedback. Robot control applications favor smooth controllers at convergence to limit hardware stress. We aim to bridge the gap between coarse exploration capabilities and smooth control performance while retaining sample-efficient learning. We leverage insights from the growing action space literature [Czarnecki et al., 2018, Farquhar et al., 2020] and consider a decoupled critic that increases its control resolution over the course of training. To this end, we define the discrete action sub-space at iteration g as $\mathcal{A}^g \subset \mathcal{A}$ and modify the TD target to yield

$$y_t = \sum_{j=0}^{n-1} \gamma^j r(s_{t+j}, a_{t+j}) + \gamma^n \sum_{j=1}^{M} \max_{a_{t+1}^j \in \mathcal{A}^g} \frac{Q_{\theta^-}^j(\boldsymbol{s}_{t+n}, a_{t+n}^j)}{M}, \tag{3}$$

where ϵ-greedy action sampling is constrained to $\mathcal{A}^g$. The network architecture accommodates the full discretized action space from the start and constrains the active set via action masking, enabling masked action combinations to profit from information propagation in the shared torso [Van Seijen et al., 2017]. A schematic of a decoupled agent with 5-bin discretization and active 3-bin subspace is provided in Figure 1. In order to deploy such an agent, we require a schedule for when to expand the active action space $\mathcal{A}^g \rightarrow \mathcal{A}^{g+1}$. Here, we consider two simple variations to limit engineering effort. First, we consider a linear schedule that doubles control resolution every $\frac{1}{N+1}$ of training episodes, where N indicates the number of subspaces $\mathcal{A}^g$. Second, we formulate an adaptive schedule based on an upper confidence bound inspired threshold over the moving average returns

$$G_{\textbf{threshold},t} = \left(1.00 - 0.05 \operatorname{sgn} \mu_{\textbf{MA},t-1}^G\right) \mu_{\textbf{MA},t-1}^G + 0.90 \sigma_{\textbf{MA},t-1}^G, \tag{4}$$

where $\mu_{\textbf{MA}}$ and $\sigma_{\textbf{MA}}$ are the moving average mean and standard deviation of the evaluation returns, respectively. The objective underestimates the mean by 5% and expands the action space whenever the current mean return falls below the threshold $\mu_t^G < G_{\textbf{threshold},t}$, signifying performance stagnation. This parameterization can avoid pre-mature expansion when exploring under sparse rewards, but alternative formulations are also applicable. A qualitative example of our approach is provided in Figure 2, where we visualize the state-action value function over the course of training on a pendulum swing-up task. We consider a GQN agent with discretization $2 \rightarrow 9$ (meaning $\{2, 3, 5, 9\}$) and provide learned values for each action bin starting at initialization and adding a row every time the action space is grown (top to bottom). The active bins are framed in green, where we observe the accurate representation of the state-action value function for active bins, while the inactive bins still provide structured output due to the high degree of weight sharing provided by our architecture.

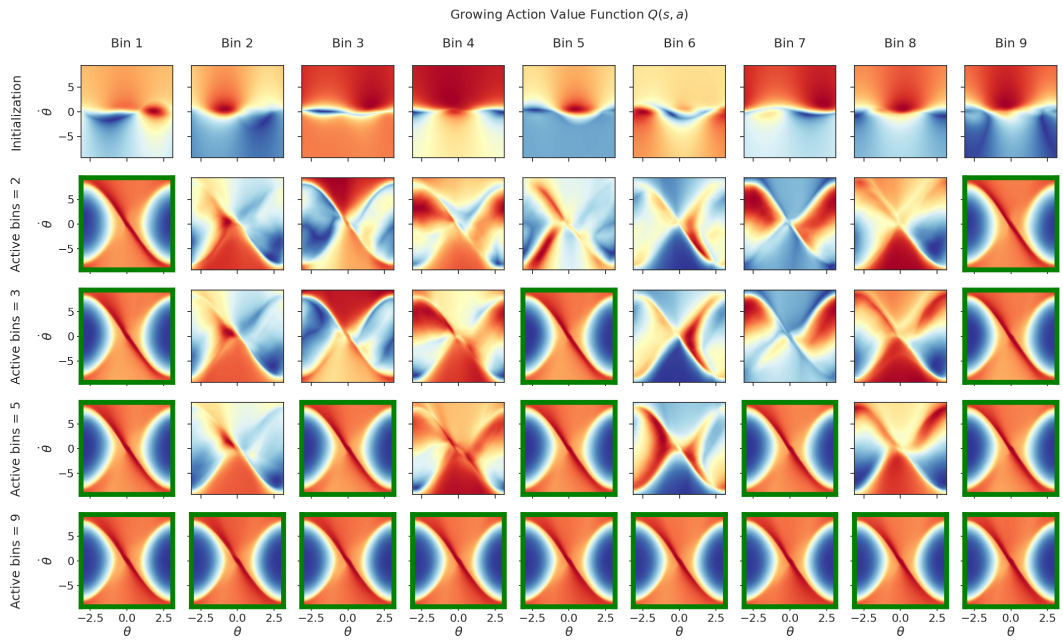

Figure 2: State-action values for a pendulum swing-up task over the course of training (top to bottom). The active bins are outlined in green. The value predictions transition from random at initialization to structured upon activation. Inactive bins profit from the emergent structure within the shared network torso to warm-start their optimization.

In the following section, we provide quantitative results on a range of challenging continuous control tasks. We use the same set of hyperparameters throughout all experiments, unless otherwise indicated, following the general parameterization of Seyde et al. [2022b] with a simple multi-layer perceptron architecture and dimensionality $[512, 512]$. We evaluate mean performance with standard deviation across 4 seeds and 10 evaluation episodes for each task. Our implementation builds on the codebase of Seyde et al. [2022b] and we provide hyperparameter values in Table 1 of the Appendix.

5 Experiments

We evaluate our approach on a selection of tasks from the DeepMind Control Suite [Tunyasuvunakool et al., 2020], MetaWorld [Yu et al., 2020], and MyoSuite [Vittorio et al., 2022]. The former two benchmarks generally do not consider action penalties and have previously been solved with bang-bang control [Seyde et al., 2022b]. Therefore, we focus on action-penalized task variations to encourage smooth control and highlight exploration challenges in the presence of penalty terms.

We first evaluate performance on tasks from the DeepMind Control Suite with action dimensionality up to $dim(\mathcal{A}) = 38$. We consider 2 penalty weights $c_a \in \{0.1, 0.5\}$, such that rewards are computed as $r_t = r_t^o - c_a \sum_{j=1}^{M} {a_t^j}^2 / M$ from original reward r_t^o. We consider GQN agents that grow their action space discretization from 2 to 9 bins in each action dimension, where we evaluate both the linear and adaptive growing schedules discussed in Section 4. We compare performance against the state-of-the-art continuous control D4PG [Barth-Maron et al., 2018] and DMPO [Abdolmaleki et al., 2018b] agents while providing two discrete control DecQN agents with stationary action space discretization of 2 or 9 for reference. The results in Figures 3 and 4 indicate the strong performance of GQN agents, with the adaptive schedule improving upon the linear schedule in terms of convergence rate and variance. Growing control resolution further provides a clear advantage over the stationary DecQN agents both in terms of final performance (vs. DecQN 2) and exploration abilities (vs. DecQN 9). These observations mirror findings by Czarnecki et al. [2018], where coarse control resolution was beneficial for early exploration, a characteristic amplified by action penalties. We further observe strong performance of discrete GQN agents compared to the continuous D4PG and DMPO agents.

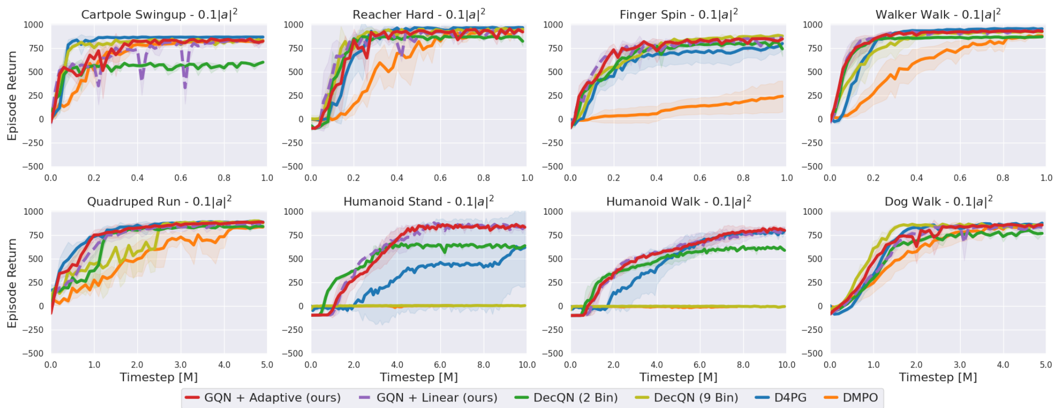

Figure 3: Performance on tasks from the DeepMind Control Suite with action penalty $-0.1|a|^2$. Our GQN agent grows its action space resolution via a $2 \rightarrow 3 \rightarrow 5 \rightarrow 9$ bin sequence, where the linear and adaptive expansion schedules yield similar results. The GQN agent performs competitive to the discrete DecQN as well as the continuous D4PG and DMPO baselines, achieving noticeable improvements on the Humanoid Stand and Walk tasks.

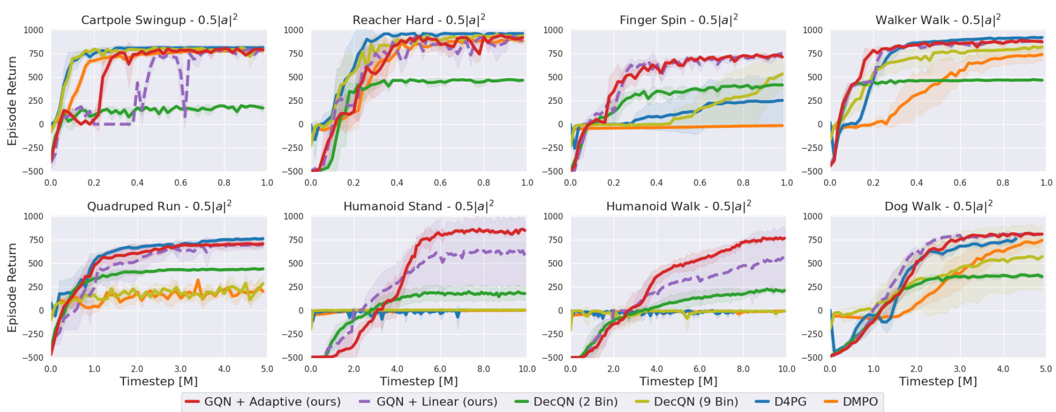

Figure 4: Performance on tasks from the DeepMind Control Suite with action penalty $-0.5|a|^2$. Our GQN agent grows its action space resolution via a $2 \rightarrow 3 \rightarrow 5 \rightarrow 9$ bin sequence, where we observe benefits of the adaptive variant over the linear schedule. GQN yields performance improvements over the discrete DecQN as well as the continuous D4PG and DMPO baselines, with particularly strong deltas on the Humanoid and Finger tasks.

The non-stationary optimization objective inherent to GQN may not be necessary on simpler tasks with limited exploration requirements such as Cartpole Swinup or Reacher Hard, while it significantly improves performance on complex domains such as Humanoid or Dog.

In order to provide additional quantitative motivation for the presence of action penalties, we compare the smoothness of the converged policies in Figure 5. We consider the adaptive GQN agent with action penalties $c_a \in \{0.1, 0.5\}$ and the continuous D4PG agent with action penalty $c_a = 0.5$. The metrics we consider are original non-penalized task performance, R, incurred action penalty, P, action magnitude, $|a|$, instantaneous action change, $|\Delta a|$, and the Fast Fourier Transform (FFT) based smoothness metric from Mysore et al. [2021], SM. All metrics are normalized by the corresponding value achieved by the unconstrained GQN agent with $c_a = 0.0$. The results indicate that increasing the action penalty yields noticeably smoother control signals while only having a minor impact on the original task performance as measured by the unconstrained reward, R. We further find that smoothness of the discrete GQN agent is at least as good as for the continuous D4PG agent on the tasks considered (note that D4PG is unable to solve the Humanoid tasks, $R \approx 0$).

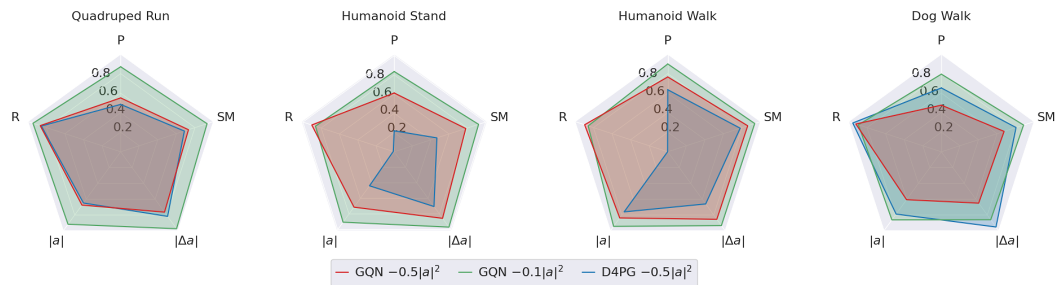

Figure 5: Comparison of control smoothness and reward performance, relative to GQN without action penalties. Increasing the action penalty coefficient yields smoother control while only having a minor impact on the original task performance as measured by unconstrained reward R. The discrete GQN further improves upon the continuous D4PG agent.

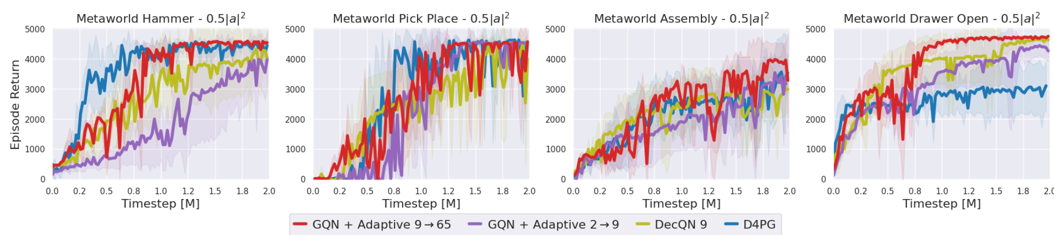

Figure 6: Performance on manipulation tasks from MetaWorld with action penalty $-0.5|a|^2$. These tasks require control at the velocity level and are therefore more challenging to solve with extremely coarse discretization. We therefore investigate the scalability of our GQN agent and consider growing discretizations via a $9 \rightarrow 17 \rightarrow 33 \rightarrow 65$ bin sequence. The resulting policy achieves stable learning and performs competitively with the continuous D4PG baseline while improving on the stationary 9 bins DecQN agent.

Next, we extend our study to velocity-level control tasks for the Sawyer robot in MetaWorld. While acceleration-level control often provides sufficient filtering to interact favorably with highly discretized bang-bang exploration, velocity-level control tends to require more fine-grained inputs. We investigate the scalability of growing action spaces within decoupled Q-learning representations. To this end, we consider GQN agents with $2 \rightarrow 9$ and $9 \rightarrow 65$ (meaning $\{9, 17, 33, 65\}$) discretization as well as a stationary DecQN agent with 9 bins. The results in Figure 6 indicate that initial bang-bang action selection is not well-suited for generating velocity-level actions, with the agent achieving good performance once transitioning to more fine-grained discretization (GQN $2 \rightarrow 9$). Interestingly, considering a larger growing action space with GQN $9 \rightarrow 65$ can surpass the performance of a stationary DecQN 9 agent, despite the non-stationary optimization objective induced by the addition of finer action discretizations over the course of training. The performance of GQN $9 \rightarrow 65$ is furthermore competitive with the continuous D4PG agent on average.

Lastly, we stress-test our approach by considering a selection of tasks from the MyoSuite benchmark. The tasks require control of biomechanical models that aim to be physiologically accurate with $dim(\mathcal{A}) = 39$ and up to $dim(\mathcal{O}) = 115$ and should constrain the applicability of simple decoupled Q-learning approaches such as GQN. Indeed, we find that the agent capacity becomes a limiting factor yielding overestimation errors further exacerbated by the large magnitude reward signals. We therefore extend the network capacity to $[512, 512] \rightarrow [2048, 2048]$ and lower the discount factor $\gamma = 0.99 \rightarrow 0.95$ (alternatively, increasing multi-step returns $3 \rightarrow 5$ worked similarly well). With these parameter adjustments, we observe good performance as measured by task success at the final step of an episode, comparing favorably to the continuous D4PG agent in Figure 7. This further underlines the surprising effectiveness that decoupled discrete control can yield in continuous control settings and the benefit of adaptive control resolution change over the course of training.

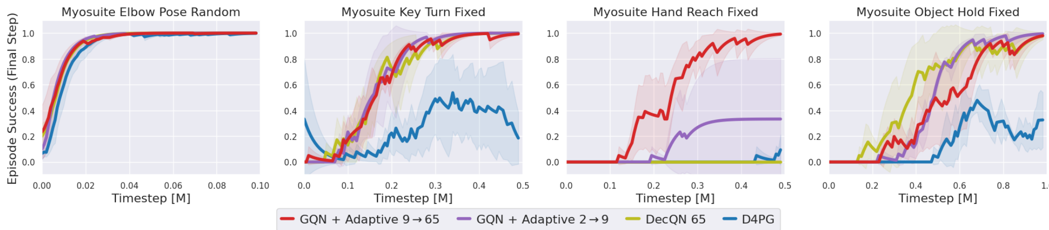

Figure 7: Performance for controlling biomechanical models from the MyoSuite as measured by task success at termination. These continuous control tasks stress test growing decoupled discrete action spaces, due to their dimensionality and inherent complexity. Increasing the network capacity and adjusting the discount factor to mitigate overestimation, we observe strong performance for growing action spaces up to a discretization of 65 bins.

6 Conclusion

This work investigates the application of growing action spaces within decoupled Q-learning to efficiently solve continuous control tasks. Our Growing Q-Networks (GQN) agent leverages a linear value decomposition along actuators to retain scalability in high-dimensional action spaces and adaptively increases control resolution over the course of training. This enables coarse exploration early during training without reduced control smoothness and accuracy at convergence. The resulting agent is robust and performs well even for very fine control resolutions despite inherent non-smoothness in the optimization objective arising at the transition between resolution levels. While GQN as a critic-only method displays very strong performance compared to recent continuous actor-critic methods on the tasks considered, we also investigate scenarios that prove challenging for decoupled discrete controllers as exemplified by velocity-level control of simulated manipulators or applications to control of biomechanical models. Interesting avenues for future work include addressing coordination challenges in increasingly high-dimensional action spaces and mitigating overestimation bias. Generally, GQN provides a simple yet capable agent that efficiently bridges the gap between coarse exploration and solution smoothness through adaptive control resolution refinement.

References

A. Abdolmaleki, J. T. Springenberg, J. Degrave, S. Bohez, Y. Tassa, D. Belov, N. Heess, and M. Riedmiller. Relative entropy regularized policy iteration. *arXiv preprint arXiv:1812.02256*, 2018a.

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

C. W. Anderson. Learning to Control an Inverted Pendulum with Connectionist Networks. In *Proceedings of the American Control Conference (ACC)*, 1988.

G. Barth-Maron, M. W. Hoffman, D. Budden, W. Dabney, D. Horgan, D. Tb, A. Muldal, N. Heess, and T. Lillicrap. Distributed distributional deterministic policy gradients. *arXiv preprint arXiv:1804.08617*, 2018.

C. Beattie, J. Z. Leibo, D. Teplyashin, T. Ward, M. Wainwright, H. Küttler, A. Lefrancq, S. Green, V. Valdés, A. Sadik, et al. Deepmind lab. *arXiv preprint arXiv:1612.03801*, 2016.

R. Bellman, I. Glicksberg, and O. Gross. On the “bang-bang” control problem. *Quarterly of Applied Mathematics*, 14(1), 1956.

S. Bohez, A. Abdolmaleki, M. Neunert, J. Buchli, N. Heess, and R. Hadsell. Value constrained model-free continuous control. *arXiv preprint arXiv:1902.04623*, 2019.

W. Böhmer, T. Rashid, and S. Whiteson. Exploration with unreliable intrinsic reward in multi-agent reinforcement learning. *arXiv preprint arXiv:1906.02138*, 2019.

W. Böhmer, V. Kurin, and S. Whiteson. Deep coordination graphs. In *International Conference on Machine Learning*, pages 980–991. PMLR, 2020.

L. Busoniu, B. De Schutter, and R. Babuska. Decentralized reinforcement learning control of a robotic manipulator. In *2006 9th International Conference on Control, Automation, Robotics and Vision*, pages 1–6. IEEE, 2006.

F. Christianos, G. Papoudakis, M. A. Rahman, and S. V. Albrecht. Scaling multi-agent reinforcement learning with selective parameter sharing. In *International Conference on Machine Learning*, pages 1989–1998. PMLR, 2021.

X. Chu and H. Ye. Parameter sharing deep deterministic policy gradient for cooperative multi-agent reinforcement learning. *arXiv preprint arXiv:1710.00336*, 2017.

C. Claus and C. Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. *AAAI/IAAI*, 1998:2, 1998.

W. Czarnecki, S. Jayakumar, M. Jaderberg, L. Hasenclever, Y. W. Teh, N. Heess, S. Osindero, and R. Pascanu. Mix & match agent curricula for reinforcement learning. In *International Conference on Machine Learning*, pages 1087–1095. PMLR, 2018.

G. Farquhar, L. Gustafson, Z. Lin, S. Whiteson, N. Usunier, and G. Synnaeve. Growing action spaces. In *International Conference on Machine Learning*, pages 3040–3051. PMLR, 2020.

J. Foerster, N. Nardelli, G. Farquhar, T. Afouras, P. H. Torr, P. Kohli, and S. Whiteson. Stabilising experience replay for deep multi-agent reinforcement learning. In *International conference on machine learning*, pages 1146–1155. PMLR, 2017.

C. Guestrin, M. Lagoudakis, and R. Parr. Coordinated reinforcement learning. In *ICML*, volume 2, pages 227–234. Citeseer, 2002.

J. K. Gupta, M. Egorov, and M. Kochenderfer. Cooperative multi-agent control using deep reinforcement learning. In *International conference on autonomous agents and multiagent systems*, pages 66–83. Springer, 2017.

T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

D. Hafner, T. Lillicrap, M. Norouzi, and J. Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.

M. Hessel, J. Modayil, H. Van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

B. J. Hodel. Learning to Operate an Excavator via Policy Optimization. *Procedia Computer Science*, 140, 2018.

S. H. Huang, M. Zambelli, J. Kay, M. F. Martins, Y. Tassa, P. M. Pilarski, and R. Hadsell. Learning Gentle Object Manipulation with Curiosity-Driven Deep Reinforcement Learning. *arXiv:1903.08542*, 2019.

D. Ireland and G. Montana. Revalued: Regularised ensemble value-decomposition for factorisable markov decision processes. *arXiv preprint arXiv:2401.08850*, 2024.

M. Jaderberg, V. Dalibard, S. Osindero, W. M. Czarnecki, J. Donahue, A. Razavi, O. Vinyals, T. Green, I. Dunning, K. Simonyan, et al. Population based training of neural networks. *arXiv preprint arXiv:1711.09846*, 2017.

J. R. Kok and N. Vlassis. Collaborative multiagent reinforcement learning by payoff propagation. *Journal of Machine Learning Research*, 7:1789–1828, 2006.

J. Lambert and M. Levine. A two-stage learning control system. *Trans. on Automatic Control*, 15(3), 1970.

J. P. LaSalle. Time Optimal Control Systems. *Proceedings of the National Academy of Sciences*, 45 (4), 1959.

M. Lauer and M. Riedmiller. An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In *In Proceedings of the Seventeenth International Conference on Machine Learning*. Citeseer, 2000.

L. Matignon, G. J. Laurent, and N. Le Fort-Piat. Hysteretic q-learning: an algorithm for decentralized reinforcement learning in cooperative multi-agent teams. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 64–69. IEEE, 2007.

L. Matignon, G. J. Laurent, and N. Le Fort-Piat. Independent reinforcement learners in cooperative markov games: a survey regarding coordination problems. *The Knowledge Engineering Review*, 27:1–31, 2012.

H. Maurer, C. Büskens, J.-H. R. Kim, and C. Y. Kaya. Optimization methods for the verification of second order sufficient conditions for bang–bang controls. *Optimal Control Applications and Methods*, 26(3), 2005.

L. Metz, J. Ibarz, N. Jaitly, and J. Davidson. Discrete sequential prediction of continuous actions for deep rl. *arXiv preprint arXiv:1705.05035*, 2017.

V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

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

S. Mysore, B. Mabsout, R. Mancuso, and K. Saenko. Regularizing action policies for smooth control with reinforcement learning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1810–1816. IEEE, 2021.

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

G. Novati and P. Koumoutsakos. Remember and Forget for Experience Replay. In *International Conference on Machine Learning (ICML)*, 2019.

B. Peng, T. Rashid, C. Schroeder de Witt, P.-A. Kamienny, P. Torr, W. Böhmer, and S. Whiteson. Facmac: Factored multi-agent centralised policy gradients. *Advances in Neural Information Processing Systems*, 34, 2021.

T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, and S. Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 4295–4304. PMLR, 2018.

J. Roy, R. Girgis, J. Romoff, P.-L. Bacon, and C. Pal. Direct behavior specification via constrained reinforcement learning. *arXiv preprint arXiv:2112.12228*, 2021.

S. J. Russell and A. Zimdars. Q-decomposition for reinforcement learning agents. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 656–663, 2003.

T. Schaul, J. Quan, I. Antonoglou, and D. Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

J. G. Schneider, W.-K. Wong, A. W. Moore, and M. A. Riedmiller. Distributed value functions. In *ICML*, 1999.

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

T. Seyde, I. Gilitschenski, W. Schwarting, B. Stellato, M. Riedmiller, M. Wulfmeier, and D. Rus. Is bang-bang control all you need? solving continuous control with bernoulli policies. *Advances in Neural Information Processing Systems*, 34, 2021.

T. Seyde, W. Schwarting, I. Gilitschenski, M. Wulfmeier, and D. Rus. Strength through diversity: Robust behavior learning via mixture policies. In *Conference on Robot Learning*, pages 1144–1155. PMLR, 2022a.

T. Seyde, P. Werner, W. Schwarting, I. Gilitschenski, M. Riedmiller, D. Rus, and M. Wulfmeier. Solving continuous control via q-learning. In *The Eleventh International Conference on Learning Representations*, 2022b.

S. Sharma, A. Suresh, R. Ramesh, and B. Ravindran. Learning to factor policies and action-value functions: Factored action space representations for deep reinforcement learning. *arXiv preprint arXiv:1705.07269*, 2017.

J. Skalse, N. Howe, D. Krasheninnikov, and D. Krueger. Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems*, 35:9460–9471, 2022.

L. Smith, Y. Cao, and S. Levine. Grow your limits: Continuous improvement with real-world rl for robotic locomotion. *arXiv preprint arXiv:2310.17634*, 2023.

K. Son, D. Kim, W. J. Kang, D. E. Hostallero, and Y. Yi. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 5887–5896. PMLR, 2019.

L. M. Sonneborn and F. S. Van Vleck. The Bang-Bang Principle for Linear Control Systems. *Journal of the Society for Industrial and Applied Mathematics Series A Control*, 2(2), 1964.

J. Su, S. Adams, and P. A. Beling. Value-decomposition multi-agent actor-critics. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11352–11360, 2021.

P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*, 2017.

R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. 2018.

G. Synnaeve, J. Gehring, Z. Lin, D. Haziza, N. Usunier, D. Rothermel, V. Mella, D. Ju, N. Carion, L. Gustafson, et al. Growing up together: Structured exploration for large action spaces. 2019.

M. Tan. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth international conference on machine learning*, pages 330–337, 1993.

Y. Tang and S. Agrawal. Discretizing continuous action space for on-policy optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5981–5988, 2020.

A. Tavakoli. *On structural and temporal credit assignment in reinforcement learning*. PhD thesis, Imperial College London, 2021.

A. Tavakoli, F. Pardo, and P. Kormushev. Action branching architectures for deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

A. Tavakoli, M. Fatemi, and P. Kormushev. Learning to represent action values as a hypergraph on the action vertices. In *International Conference on Learning Representations*, 2021.

T. G. Thuruthel, E. Falotico, F. Renda, and C. Laschi. Model-Based Reinforcement Learning for Closed-Loop Dynamic Control of Soft Robotic Manipulators. *IEEE T-RO*, 35(1), 2019.

S. Tunyasuvunakool, A. Muldal, Y. Doron, S. Liu, S. Bohez, J. Merel, T. Erez, T. Lillicrap, N. Heess, and Y. Tassa. dm_control: Software and tasks for continuous control. *Software Impacts*, 6:100022, 2020.

T. Van de Wiele, D. Warde-Farley, A. Mnih, and V. Mnih. Q-learning in enormous action spaces via amortized approximate maximization. *arXiv preprint arXiv:2001.08116*, 2020.

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

H. Van Seijen, M. Fatemi, J. Romoff, R. Laroche, T. Barnes, and J. Tsang. Hybrid reward architecture for reinforcement learning. *Advances in Neural Information Processing Systems*, 30, 2017.

C. Vittorio, W. Huawei, D. Guillaume, S. Massimo, and K. Vikash. Myosuite – a contact-rich simulation suite for musculoskeletal motor control. `https://github.com/myohub/myosuite`, 2022. URL `https://arxiv.org/abs/2205.13600`.

M. Waltz and K. Fu. A heuristic approach to reinforcement learning control systems. *IEEE TACON*, 10(4), 1965.

Y. Wang, B. Han, T. Wang, H. Dong, and C. Zhang. Dop: Off-policy multi-agent decomposed policy gradients. In *International Conference on Learning Representations*, 2020.

C. J. Watkins and P. Dayan. Q-learning. *Machine learning*, 8:279–292, 1992.

M. Wulfmeier, A. Abdolmaleki, R. Hafner, J. T. Springenberg, M. Neunert, N. Siegel, T. Hertweck, T. Lampe, N. Heess, and M. Riedmiller. Compositional Transfer in Hierarchical Reinforcement Learning. In *Robotics: Science and Systems (RSS)*, 2020.

J. Yang, T. Dzanic, B. Petersen, J. Kudo, K. Mittal, V. Tomov, J.-S. Camier, T. Zhao, H. Zha, T. Kolev, et al. Reinforcement learning for adaptive mesh refinement. In *International Conference on Artificial Intelligence and Statistics*, pages 5997–6014. PMLR, 2023.

Y. Yang, R. Luo, M. Li, M. Zhou, W. Zhang, and J. Wang. Mean field multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 5571–5580. PMLR, 2018.

T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning*, pages 1094–1100. PMLR, 2020.

A Hyperparameters

Throughout our experiments, we use the hyperparameter values in Table 1 unless otherwise indicated.

Table 1: GQN hyperparameters.

Parameter	Value
Optimizer	Adam
Learning rate	1×10^{-4}
n-step returns	3
Action repeat	1
Discount γ	0.99
Batch size	256
Gradient clipping	40
Target update period	100
Imp. sampling exponent	0.2
Priority exponent	0.6
Exploration ϵ	0.1