DrM: Mastering Visual Reinforcement Learning through Dormant Ratio Minimization

Guowei Xu[1*†] **Ruijie Zheng**[2*] **Yongyuan Liang**[2*] **Xiyao Wang**[2]
Zhecheng Yuan[1] **Tianying Ji**[1] **Yu Luo**[1] **Xiaoyu Liu**[2] **Jiaxin Yuan**[2] **Pu Hua**[1]
Shuzhen Li[1] **Yanjie Ze**[34] **Hal Daumé III**[2] **Furong Huang**[2] **Huazhe Xu**[145†]
[1] Tsinghua University [2] University of Maryland, College Park
[3] Shanghai Jiao Tong University [4] Shanghai Qi Zhi Institute [5] Shanghai AI Lab

Abstract

Visual reinforcement learning (RL) has shown promise in continuous control tasks. Despite its progress, current algorithms are still unsatisfactory in virtually every aspect of the performance such as sample efficiency, asymptotic performance, and their robustness to the choice of random seeds. In this paper, we identify a major shortcoming in existing visual RL methods that is the agents often exhibit sustained inactivity during early training, thereby limiting their ability to explore effectively. Expanding upon this crucial observation, we additionally unveil a significant correlation between the agents' inclination towards motorically inactive exploration and the absence of neuronal activity within their policy networks. To quantify this inactivity, we adopt dormant ratio (Sokar et al., 2023) as a metric to measure inactivity in the RL agent's network. Empirically, we also recognize that the dormant ratio can act as a standalone indicator of an agent's activity level, regardless of the received reward signals. Leveraging the aforementioned insights, we introduce `DrM`, a method that uses three core mechanisms to guide agents' exploration-exploitation trade-offs by actively minimizing the dormant ratio. Experiments demonstrate that `DrM` achieves significant improvements in sample efficiency and asymptotic performance with no broken seeds (76 seeds in total) across three continuous control benchmark environments, including DeepMind Control Suite, MetaWorld, and Adroit. Most importantly, `DrM` is the first model-free algorithm that consistently solves tasks in both the Dog and Manipulator domains from the DeepMind Control Suite as well as three dexterous hand manipulation tasks without demonstrations in Adroit, all based on pixel observations. [1]

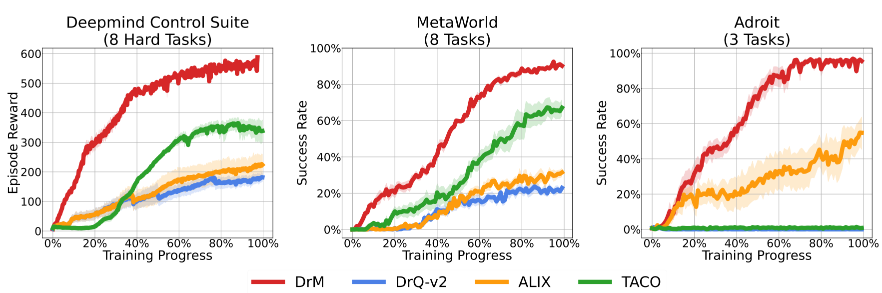


Figure 1: Success rate and episode reward as a function of training progress for each of the three domains that we consider (Deepmind Control Suite, MetaWorld, Adroit). All results are averaged over 4 random seeds, and the shaded region stands for standard deviation across different random seeds.

*Equal contribution.
†Corresponding author. {xgw23@mails, huazhe_xu@mail}.tsinghua.edu.cn
[1]Please refer to https://drm-rl.github.io/ for experiment videos and benchmark results.

1 INTRODUCTION

Visual deep reinforcement learning (RL) agents that tackle complex continuous control tasks using high-dimensional pixels are crucial. Recent progress has been made through the incorporation of data augmentation (Yarats et al., 2022; 2021; Laskin et al., 2020a), self-supervised representation learning (Zheng et al., 2023; Laskin et al., 2020b; Stooke et al., 2021; Schwarzer et al., 2021; D'Oro et al., 2023), regularization of the temporal difference update (Cetin et al., 2022), and high update-to-data (UTD) ratio (Hiraoka et al., 2022). Nonetheless, the sample efficiency exhibited by these RL agents remains unsatisfactory. To be more specific, visual RL's inability first appears in the face of complex kinematics and a high number of degrees of freedom (DoFs), such as the Dog and Humanoid tasks in the DeepMind Control Suite (Tassa et al., 2018) or dexterous hand manipulation tasks in Adroit (Rajeswaran et al., 2018) without demonstrations. Second, the current leading visual RL agents might get stuck in the local optimum during the learning process under different initial random seeds. The inability to deal with complex systems and the presence of broken random seeds combined pose significant challenges to deploying visual RL agents in real-world applications.

In this paper, we examine the behaviors of visual RL agents at different stages of training. Intriguingly, a recurrent issue we identify by observing the learning agents' behavior is that the agents frequently become motorically inactive during the initial phases of training, hindering the effective exploration of useful behaviors. When the agent is experiencing motor inactivity, we find that the policy neural network also possesses a high rate of inactive neurons, which is defined as dormant neurons (Sokar et al., 2023) in the literature. As the training progresses, the agents' acquisition of new skills is usually accompanied by a decline in the portion of dormant neurons i.e., dormant ratio. Hence, we hypothesize and empirically verify that the dormant ratio acts as an inherent gauge of an agent's activity level, irrespective of the external rewards it receives. Such a connection opens up a new path for balancing between exploration and exploitation in RL agents. Remarkably, this pattern of inactivity in motor skills and neurons mirrors the arousal theory (Harrison & W, 2015; Güzel et al., 2020) in neuroscience, which states that an optimal neural network activity level is essential for enhancing attention, memory, and learning efficiency.

Based on this observation and insight, we propose to train visual RL agents with **D**ormant **r**atio **M**inimization (`DrM`). `DrM` introduces three simple mechanisms to effectively balance between exploration and exploitation while lowering the dormant ratio: a periodical neural network weight perturbation mechanism, a dormant-ratio-based exploration scheduler, and a dormant-ratio-based exploitation mechanism extended from Chen et al. (2021a). Consequently, the agent could emphasize exploration when the dormant ratio is high and shift its focus to exploitation when the dormant ratio is low. `DrM` is easy to implement, computationally efficient, and empirically sample efficient.

`DrM` is evaluated across three different domains, Deepmind Control Suite (Tassa et al., 2018), MetaWorld (Yu et al., 2019), and Adroit (Rajeswaran et al., 2018), including 19 tasks within the realm of locomotion control, tabletop manipulation, and dexterous hand manipulation. Most notably, `DrM` is the **first documented model-free algorithm** that reliably solves complex dog and manipulator tasks, as well as demonstration-free Adroit dexterous hand manipulation tasks from pixels. Furthermore, compared with previous state-of-the-art model-free algorithms, `DrM` is significantly more sample efficient, especially on tasks with sparse rewards. To be precise, our technique requires 70%, 45%, and 60% fewer samples to match the peak asymptotic performance seen in the three baseline methods on the Deepmind Control suite, MetaWorld, and Adroit, respectively. Moreover, in terms of asymptotic performance, our method exhibits improvements of 65%, 35%, and 75% over the best-performing baseline on the Deepmind Control suite, MetaWorld, and Adroit, respectively.

Below, we summarize our key contributions:

1. Through systematic examinations of the dormant ratio within agents performing continuous control tasks, we establish a crucial insight that a decline in this dormant ratio is an early indicator of successful skill acquisition, even before the increase of reward.
2. We introduce a mechanism that periodically perturbs the model weights of the agent, effectively reducing the dormant ratio and hence accelerating skill acquisition.
3. We additionally design a dormant-ratio-based self-adaptive exploration-exploitation scheduler that ensures the agent explores when the dormant ratio is low and exploits its past success when the dormant ratio is low.

4. Extensive experiments on Deepmind Control Suite, MetaWorld, and Adroit show that `DrM` is particularly adept at handling tasks with sparse rewards or complex dynamics, achieving state-of-the-art performance against current leading visual RL baselines. `DrM` is the first model-free RL algorithm that can reliably solve complex tasks such as Dog, and Manipulator, as well as demonstration-free Adroit dexterous hand manipulation tasks directly from pixels.

2 PRELIMINARY

Visual reinforcement learning. In visual RL (Kaelbling et al., 1998), the landscape is characterized by the inherent challenge of partial observability when dealing with image inputs, which prompts us to approach the problem as a Partially Observable Markov Decision Process (POMDP) (Bellman, 1957), encapsulated within the tuple $\langle \mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$. Here, $\mathcal{S}$ is the state space, $\mathcal{O}$ is the observation space and $\mathcal{A}$ stands for the action space. $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ defines the state transition kernel, where $\Delta(\mathcal{S})$ is a distribution over the state space. $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ denotes the reward function and $\gamma \in [0, 1)$ represents the discount factor. Starting from an initial state $s_0 \in \mathcal{S}$, the overarching objective within this framework is to discover an optimal policy $\pi^* : \mathcal{S} \to \Delta(\mathcal{A})$ that maximizes the expected cumulative return, formulated as $\mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r_t]$.

Dormant Ratio of Neural Network The notion of dormant neurons, as originally introduced in Sokar et al. (2023), identifies neurons that have become nearly inactive, displaying minimal activation levels. This concept plays an important role in analyzing neural network behavior since networks used in online RL tend to lose their expressive ability.

Definition 2.1. *(Sokar et al., 2023) Consider a linear layer l with N^l neurons in total. Given an input distribution $\mathcal{D}$, let $h_i^l(x)$ denote the output of neuron i in layer l under input $x \in \mathcal{D}$. The **score** of a neuron i is:*

$$s_i^l = \frac{\mathbb{E}_{x \in \mathcal{D}} |h_i^l(x)|}{\frac{1}{N^l} \sum_{k \in l} \mathbb{E}_{x \in \mathcal{D}} |h_k^l(x)|} \tag{1}$$

Then we define a neuron i in layer l to be τ-dormant if $s_i^l \leq \tau$.

Definition 2.2. *For a linear layer l, we denote the number of τ-dormant neurons as H_τ^l. The **τ-dormant ratio** of a neural network ϕ can be formally defined as follows:*

$$\beta_\tau = \sum_{l \in \phi} H_\tau^l / \sum_{l \in \phi} N^l \tag{2}$$

3 METHOD

In this section, we begin by discussing a key empirical observation: there is a connection between the sharp reduction of an agent's dormant ratio and the agent's skill acquisition in visual continuous control tasks. This is detailed in Section 3.1. Building on top of this crucial insight, in Section 3.2, we introduce our proposed algorithm `DrM` . In particular, we come up with three simple yet effective mechanisms in `DrM` such that they aim to not only reduce the agent's dormant ratio but also utilize the calculated dormant ratio to strike a balance between exploration and exploitation.

3.1 KEY INSIGHT: DORMANT RATIO AND BEHAVIORAL VARIETY

While previous works Lyle et al. (2022); Sokar et al. (2023) have highlighted that the actor/critic network of RL agents tends to lose expressivity during training, our empirical study offers a unique perspective on visual reinforcement learning for continuous control tasks: the dormant ratio and the agent's behavioral variety are correlated.

To illustrate this, we choose DrQ-v2, a leading model-free RL algorithm that learns directly from pixel observations. In Figure 2, we display the dormant ratio of an agent's policy network, alongside the behaviors learned by the agent during its training on the Hopper Hop task from DeepMind Control Suite as an example. Interestingly, as depicted in this figure, we notice that **a sharp decline in the dormant ratio of an agent's policy network serves as an intrinsic indicator of the agent executing meaningful actions for exploration.** Namely, when the dormant ratio is high, the agent becomes immobilized and struggles to make meaningful movements. However, as this ratio decreases,

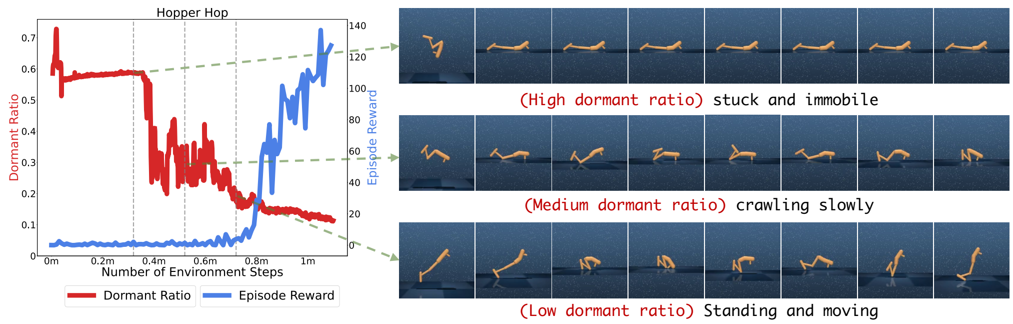


Figure 2: (Dormant ratio of a DrQ-v2 agent trained on Hopper Hop task during the first 1M frames): Interestingly, we find that with a declining dormant ratio, the agent incrementally acquires action capabilities. Even though the reward stays minimal during this phase, the dormant ratio provides a more insightful gauge of the agent's initial learning progress than the reward does.

we observe a clear progression in the agent's mobility, as demonstrated in the figure: starting with crawling, advancing to standing, and ultimately, hopping. We refer the readers to Appendix A for more visualizations of the dormant ratio.

Based on these empirical observations, we conclude that the decline in the dormant ratio is closely linked to the agent's initiation of meaningful actions, marking a departure from its prior monotonous or random behaviors. Interestingly, this shift can happen without a corresponding rise in the agent's rewards. This suggests that the dormant ratio acts as an intrinsic metric, influenced more by the diversity and relevance of the agent's behaviors than by its received rewards, which underscores the value of the dormant ratio as a meaningful metric for understanding the behaviors of visual RL agents.

Motivated by this insight, we aim to utilize dormant ratio as a pivotol tool for balancing exploration and exploitation. Many existing strategies adjust exploration noise based on static factors such as task complexity and training stage. Nonetheless, an agent's performance can fluctuate across tasks and with different initializations, making adjustments based solely on these static factors less efficient and often mandating exntensive, task-specific fine-tuning of hyperparameters. In contrast, customizing exploration noise according to the agent's current performance offers a more flexible and effective approach. While an intuitive approach would be to rely on reward signals, this strategy brings up the following challenges: 1) Reward values definitions vary across different tasks and domains, necessitating domain-specific knowledge for interpretation and hyperparameter tuning. 2) Even within a specific task, rewards might not indicate the agent's underlying learning phase. As depicted in Figure 2, an agent can attain similar rewards regardless of whether it has mastered motion or remains stagnant.

In light of this, the dormant ratio emerges as a more effective metric for adjusting the exploration and exploitation tradeoff, as it faithfully reflects the dynamic changes in the agent's behavior. Our design of `DrM` follows this simple intuition: a higher dormant ratio suggests the need for increased exploration, whereas a lower ratio calls for exploitation. As the dormant ratio captures the intrinsic characteristics of an agent's policy and behaviors, `DrM` is demonstrated to be effective across diverse tasks and domains with minimal hyperparameter tuning required.

3.2 DrM : Visual Reinforcement Learning through **D**ormant **R**atio **M**inimization

As shown in the previous subsection, it is essential for a visual RL agent to actively reduce its dormant ratio, thereby enabling it to explore the environment through purposeful actions. Driven by this insight, we introduce the three mechanisms of our proposed `DrM` algorithm in detail.

Dormant-ratio-guided perturbation. The goal of this mechanism is to perturb the model weights when the RL agent's network displays a high dormant ratio, losing its expressivity. Here, we utilize the perturbation reset method (D'Oro et al., 2023; Ash & Adams, 2020) that employs soft resets, a process that interpolates all the agent's parameters between their prior values and randomly initialized values. This can be expressed with the following equation:

$$\theta_t = \alpha\theta_{t-1} + (1-\alpha)\phi, \phi \sim \text{initializer} \tag{3}$$

Here, α is referred to as the *perturb factor*, θ_{t-1} indicates the network weights before the reset, θ_t is the network weight after the reset, and ϕ is randomly initialized weights. One reset is generally conducted every 20000 frames of interaction with the environment. The value of α is controlled by the dormant ratio β: $\alpha = clip(1 - k\beta, \alpha_{min}, \alpha_{max})$, where k is the perturb rate.

Awaken exploration scheduler. We aim to emphasize exploration with a large exploration noise when the dormant ratio is high, and reduce the exploration noise when the dormant ratio is low. Thus, rather than utilizing the linear decay of exploration noise variance in the original DrQ-v2, we introduce a dormant-ratio-based ***awaken exploration scheduler***.

Specifically, let $\hat{\beta}$ denote a low dormant ratio threshold. We define the agent as "awakened" when its dormant ratio is below $\hat{\beta}$. Let t_0 be the number of timesteps until the agent becomes "awakened" from the start of training. The standard deviation of the exploration noise, $\sigma(t)$, is then defined as:

$$\sigma(t) = \begin{cases} \max\left\{\frac{1}{1+exp(-(\beta-\hat{\beta})/T)},\quad \sigma_{\text{linear}}(t-t_0)\right\} & \text{if awakened} \\ \frac{1}{1+exp(-(\beta-\hat{\beta})/T)} & \text{otherwise} \end{cases} \tag{4}$$

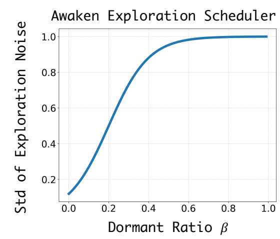


Figure 3: Visualization of the awaken exploration scheduler as a function of the dormant ratio β

Here, T is the exploration temperature hyperparameter. $\sigma_{\text{linear}}(\cdot)$ is the linear schedule of exploration noise defined in DrQ-v2. We visualize the ***awaken exploration scheduler*** in Figure 3 as a function of the dormant ratio. Initially, when the dormant ratio is high, we would like to give the agent a big exploration noise to encourage effective exploration of the environment. As training progresses and the dormant ratio decreases to a relatively low level (below the threshold $\hat{\beta}$), this indicates that the agent should transition from exploration to exploitation.

Dormant-ratio-guided exploitation. When we take exploitation into account, our goal is to focus on the exploitation of past successes when the dormant ratio is low. To achieve this, we've introduced an effective operator that skillfully balances exploitation and exploration, guided by the dormant ratio. In Q-learning algorithms, the agent's critic is trained to approximate the target value $r(s,a) + \gamma \arg\max_{a' \in \mathcal{A}} Q(s', a')$. For continuous control tasks using actor-critic algorithms, the critic aims to approximate $r(s,a) + \gamma Q(s', \pi(s'))$ instead due to the continuous nature of the action space. Consequently, in Ji et al. (2023), it demonstrates that value underestimation occurs as the agent starts to acquire skills, given that π is sub-optimal. To address this, it proposes to approximate a high expectile of Q values with a state value function V using expectile regression and combining it with Q, making the new target value

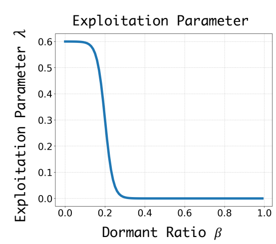


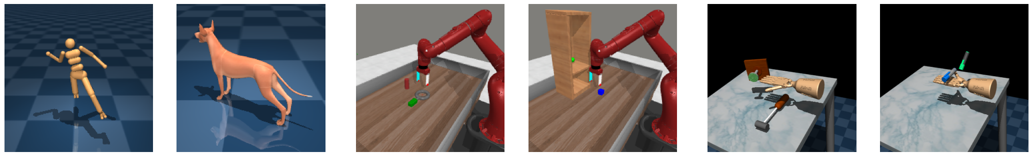
Figure 4: Visualization of exploitation hyperparameter as a function of the dormant ratio β

$$r(s,a) + \gamma[\lambda V(s') + (1-\lambda)Q(s', \pi(s'))], \lambda \in [0,1] \tag{5}$$

Here, λ serves as the *exploitation hyperparameter*. Higher values of λ focus more on exploiting past successes through the fitted V function, the value of the best actions in that state. We introduce a dormant-ratio-guided exploitation technique λ, which is now defined as a function of the dormant ratio β:

$$\lambda(\beta) = \frac{\overline{\lambda}}{1 + \exp((\beta - \hat{\beta})/T')} \tag{6}$$

Here, $\overline{\lambda}$ is the maximum exploitation hyperparameter, and T' is the exploitation temperature hyperparameter. β and $\hat{\beta}$ represent the dormant ratio and its threshold, as previously defined. In Figure 4, we plot the exploitation hyperparameter λ as a function of the dormant ratio β. When the agent's dormant ratio exceeds the threshold $\hat{\beta}$, a lower λ is selected to emphasize exploration. Conversely, when the dormant ratio is low, indicating the agent can perform meaningful actions, a higher λ is chosen to prioritize exploitation.



(a) DeepMind Control Suite **(b)** MetaWorld **(c)** Adroit

Figure 5: Three visual continuous control benchmarks to evaluate our proposed algorithm: DeepMind Control Suite, MetaWorld, and Adroit.

4 EXPERIMENT

4.1 BENCHMARK EVALUATION

In this section, we evaluate `DrM` on three visual continuous control benchmarks for both locomotion and robotic manipulation: DeepMind Control Suite (Tassa et al., 2018), MetaWorld (Yu et al., 2019), and Adroit (Rajeswaran et al., 2018). These environments feature rich visual elements such as textures and shading, necessitate fine-grained control due to complex geometry, and introduce additional challenges such as sparse rewards and high-dimensional action spaces that previous visual RL algorithms such as DrQv2 (Yarats et al., 2022) have been unable to solve.

Baselines. We compare our algorithm with the three strongest existing model-free visual RL algorithms: **DrQ-v2** (Yarats et al., 2022), **ALIX** (Cetin et al., 2022), and **TACO** (Zheng et al., 2023). **ALIX** and **TACO** build upon **DrQ-v2**. **ALIX** adds an adaptive regularization to the encoder's gradients to stabilize temporal difference learning from visual encoders. **TACO** incorporates an auxiliary temporal action-driven contrastive learning objective to simultaneously learn state and action representations.

DeepMind control suite. For Deepmind Control Suite, we evaluate `DrM` on eight hardest tasks from the Humanoid, Dog, and Manipulator domain, as well as Acrobot Swingup Sparse. The Manipulator domain is particularly challenging due to its sparse reward structure and the long horizon required for skill acquisition, while Humanoid and Dog tasks feature intricate kinematics, skinning weights, collision geometry, as well as muscle and tendon attachment points. This complexity makes these domains extremely difficult for algorithms to learn to control effectively. Following the experimental procedure described by Yarats et al. (2022), we evaluate `DrM` and all baseline algorithms over 30 million frames of online interaction, while Acrobot Swingup Sparse was run for 6 million frames. Intriguingly, in four dog tasks, we observe that existing baselines encounter a sudden performance decline for some random seeds. We have confirmed this is not due to the checkpoint loading mechanisms, and in contrast, `DrM` does not exhibit this issue in any of the four tasks. As shown in Figure 6, we note that `DrM` is the **first documented model-free visual RL algorithm** that is capable of solving both Dog and Manipulator domains in the DeepMind Control Suite using pixel observations.

MetaWorld. As shown in Figure 7, we evaluate `DrM` and baselines on eight challenging tasks including 4 very hard tasks with dense rewards following prior works and 4 medium tasks with sparse success signals. Consistently across the spectrum of tasks within MetaWorld, our method outperforms other visual RL baselines, which demonstrates the significantly improved sample efficiency of `DrM` . Especially in more challenging scenarios featuring only sparse task completion rewards, existing visual RL baselines struggle to find a good policy, while `DrM` shines by achieving success rates on par with those using dense reward signals. This underscores the remarkable advantages brought by dormant-ratio-based exploration when dealing with tasks with sparse rewards.

Adroit. In Figure 8, we also evaluate `DrM` on the Adroit domain, focusing on three dexterous hand manipulation tasks: Hammer, Door, and Pen, which requires controlling a robotic hand with 24 degrees of freedom. For additional task details, we refer readers to Rajeswaran et al. (2018). Given the task's high-dimensional action space and intricate physics, previous reinforcement learning algorithms have faced significant challenges, especially when learning from pixel observations. Notably, `DrM` is the **first documented model-free visual RL algorithm** that is capable of reliably solving tasks in the Adroit domain **without expert demonstrations**.

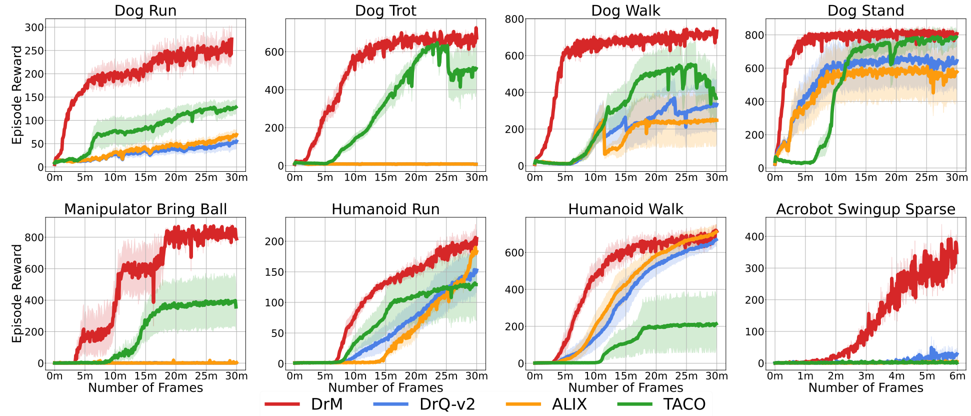


Figure 6: Performance of `DrM` against baseline algorithms **DrQ-v2**, **ALIX**, and **TACO** on Deepmind Control suite. All results are averaged over 4 random seeds, and the shaded region stands for standard deviation across different random seeds.

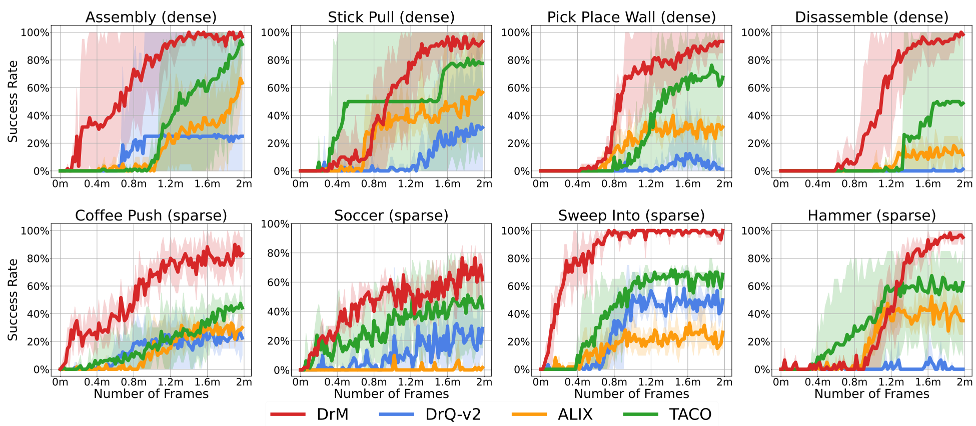


Figure 7: Success rate for `DrM` and baselines on MetaWorld including 4 very hard tasks with dense rewards and 4 medium tasks with spare rewards. All results are aggregated over 4 random seeds, with shaded areas representing the standard deviation across seeds. Notably, our method demonstrates significantly higher sample efficiency, especially in tasks with sparse rewards.

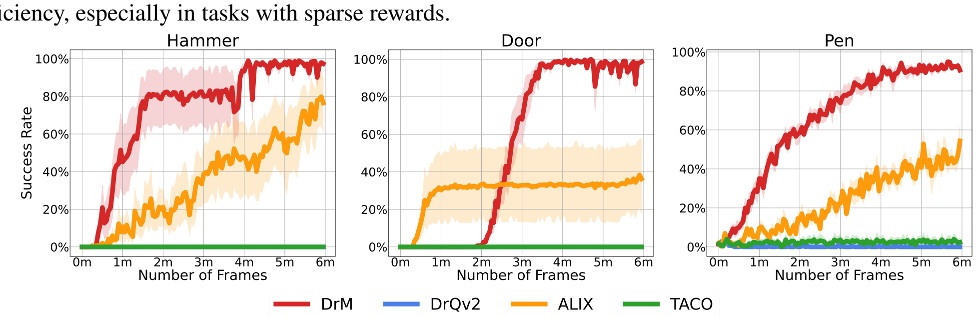


Figure 8: Performance of `DrM` against baseline algorithms **DrQ-v2**, **ALIX**, and **TACO** on Adroit. All results are averaged over 4 random seeds, and the shaded region stands for standard deviation across different random seeds.

4.2 Dormant ratio analysis

In this section, we conduct a detailed analysis and comparison of the dormant ratio changes during the training process of `DrM` and DrQ-v2. We carry out experiments in three visual DMC environments, namely Acrobot swingup sparse, Humanoid run, and Manipulator bring ball. The experimental results are shown in Figure 9. From this figure, we observe that as training progresses, the dormant ratio of `DrM` rapidly decreases, indicating that our method effectively minimizes the dormant ratio. This also explains why our approach exhibits high sample efficiency and performance. In contrast, the dormant

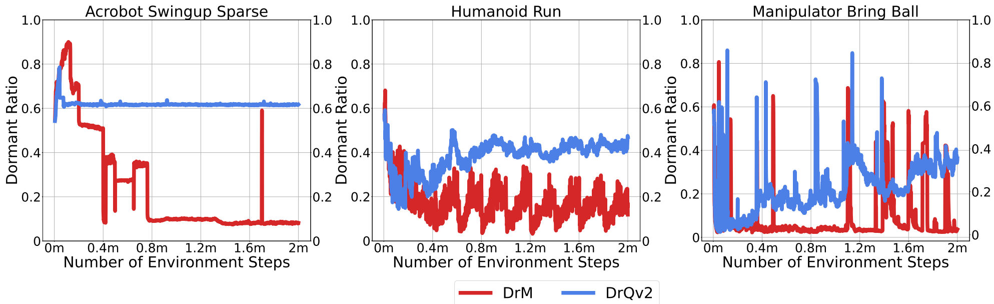


Figure 9: Dormant ratio of our method `DrM` and DrQv2 on three visual DMC tasks during learning.

ratio of DrQ-v2 remains constant in the acrobot task and even gradually increases in the other two tasks during training, which likewise elucidates DrQ-v2's incompetency in complex control tasks.

4.3 Ablation Study

We conduct ablation studies on the Adroit environment to evaluate the contribution of each component to our method, i.e., dormant-ratio-guided perturbation, awaken exploration, and dormant-ratio-guided exploitation. The experiment results are shown in Figure 10. From the results, we find that all three components are necessary to achieve the best results. We observe that after removing the dormant-ratio-guided exploitation (DrM w/o Drg Exploitation), the final success rate decreased by nearly 20%, while eliminating either the dormant-ratio-guided perturbation (DrM w/o Drg Perturbation) or the awaken exploration (DrM w/o Awaken Exploration) lead to a decline of close to 40%, highlighting the importance of each component. In our ablated version without dormant-ratio-guided perturbation, the model shows a slight improvement in the initial phase. However, the agent only converges to a suboptimal policy, reaching a success rate of just about 40%. This is likely due to the fact that without the awaken exploration, the agent lacks sufficient exploration, making it easy to get stuck in a sub-optimal policy. Additionally, when removing the dormant-ratio-guided exploitation component, the agent lacks the ability to exploit its past success, and there fore exhibits a significantly slower learning curve.

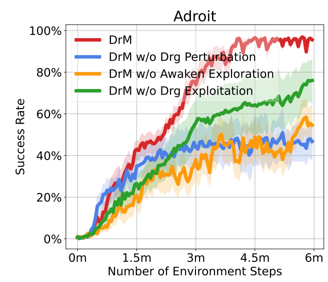


Figure 10: Ablation study of different proposed components in `DrM` being removed. Drg represents Dormant-ratio-guided.

5 Related Work

Visual reinforcement learning. Visual reinforcement learning (RL) faces substantial challenges when training agents to make decisions based on pixel observations. Within this domain, two primary categories of approaches have emerged: model-based and model-free methods. Model-based methods (Hansen et al., 2022; Hafner et al., 2020; 2021; 2019; Lee et al., 2020; Hafner et al., 2023) accelerate visual RL by learning world models of the environment. On the other hand, model-free methods have made significant strides in improving data efficiency. These advancements include auxiliary losses, such as the contrastive objective in CURL (Laskin et al., 2020b), ATC (Stooke et al., 2021) for state representations, TACO (Zheng et al., 2023) for learning state and action representations through mutual information, and self-prediction representations in SPR (Schwarzer et al., 2021) and SR-SPR (D'Oro et al., 2023). Data augmentation techniques, exemplified by RAD (Laskin et al., 2020a), DrQ (Yarats et al., 2021), and its enhanced version DrQv2 (Yarats et al., 2022), have been instrumental in enabling robust learning directly from pixel data, effectively bridging the gap between state-based and image-based RL. Additionally, regularization methods such as A-LIX (Cetin et al., 2022) have been introduced to mitigate catastrophic self-overfitting by providing adaptive regularization to convolutional features. Furthermore, strategies such as scaling network sizes (Schwarzer et al., 2023), high update-to-data (UTD) ratios (D'Oro et al., 2023) and ensemble Q (Chen et al., 2021b; Hiraoka et al., 2022) have been explored to enhance sample efficiency in visual RL. TD-MPC (Hansen et al., 2022) merges the advantages of model-based and model-free methods through temporal difference learning. V-MPO (Song et al., 2020b), an on-policy adaptation of MPO (Song et al., 2020a), exhibits high asymptotic performance on challenging pixel-control tasks (Tassa et al., 2018). These various techniques collectively represent the state-of-the-art in visual RL, addressing the multifaceted challenges associated with decision-making from raw visual input. However, our proposed framework differs in that we address the sample efficiency challenge from the perspective of dormant ratio. We propose more effective `DrM` that achieves superior performance than prior model-free baselines.

Loss of expressivity of deep RL. In deep RL, there is a growing body of evidence suggesting that neural networks tend to lose their capacity and expressiveness for fitting new targets over time and ultimately harm their final performance. To alleviate this issue, Lyle et al. (2022) and Kumar et al. (2021) primarily focus on adjusting the learned feature values. Nikishin et al. (2022) shed light on the primacy bias when training on early data, which can impede further learning progress. Their proposal involves periodic parameter reinitialization for the last few layers while keeping the replay buffer unchanged. Lyle et al. (2023) aims to identify that the loss of plasticity is fundamentally influenced by the curvature of the loss landscape. Additionally, the dormant neuron phenomenon, as demonstrated by Sokar et al. (2023) prompts the development of ReDo, a method aimed at reducing dormant neurons and preserving network expressivity during training. Nikishin et al. (2023) introduces plasticity injection, a minimalistic intervention that temporarily freezes the current network and leverages newly initialized weights to facilitate continuous learning. These diverse approaches collectively address the issue of expressivity loss in deep RL, offering insights and methods to enhance computational efficiency and continual learning capabilities in deep RL algorithms. In our paper, we leverage the dormant ratio to gain valuable insights and interpretability into agent behavior in visual RL. We introduce a novel perturbation technique and exploration strategy based on the dormant ratio for addressing visual continuous control tasks.

Exploration in RL. Efficient exploration remains a substantial challenge in online RL, particularly in high-dimensional environments with sparse rewards. Based on different key ideas and principles, exploration strategies can be classified into two major categories. The first category is uncertainty-oriented exploration (Jin et al., 2020; Ménard et al., 2021a;b; Kaufmann et al., 2021; Wang et al., 2023), which often employs techniques such as the upper confidence bound (UCB) (Auer, 2002) to capture value estimate uncertainty to guide exploration. Another category is intrinsic motivation-oriented exploration, which encourages agents to explore by maximizing intrinsic rewards. These rewards are often based on prediction errors (Houthooft et al., 2016; Pathak et al., 2017; Burda et al., 2019; Sekar et al., 2020; Badia et al., 2020) or count-based state novelty (Bellemare et al., 2016; Tang et al., 2017; Ostrovski et al., 2017), motivating the agent to visit states with high prediction errors or the unexplored states. A close idea is exploration by maximizing state entropy as an intrinsic reward (Lee et al., 2019; Hazan et al., 2019; Mutti et al., 2022; Yang & Spaan, 2023). Exploration methods have proven effective in enhancing sample efficiency in vision-based RL. RE3(Seo et al., 2021) utilizes a fixed random encoder to obtain a stable state entropy estimate, along with a value-conditional extension proposed in Kim et al. (2023). MADE (Zhang et al., 2021) introduces an adaptive regularization that maximizes deviation from explored regions, while BEE (Chen et al., 2021a) leverages past successes to capitalize on fortuitous circumstances. Closely relevant techniques involve injecting noise into action (Wawrzynski, 2015; Lillicrap et al., 2016) or parameter spaces (Rückstieß et al., 2010; Sehnke et al., 2010; Fortunato et al., 2018; Plappert et al., 2018). Furthermore, strategies that dynamically adjust exploration noise based on factors like agent performance, environmental complexity, and training stage have shown promise in Amos et al. (2021); Yarats et al. (2022). Our method distinguishes itself by directly perturbing the model weights of the agent to reduce the dormant ratio and design a dormant-ratio-guide exploration technique to improve exploration efficiency.

6 Conclusion

In this paper, we introduce a highly efficient online RL algorithm that integrates existing deep reinforcement learning methodologies. This algorithm not only stabilizes the training process but also resolves complex visual control tasks that previous models failed to tackle, setting a new benchmark in both sample and time efficiency. In addition, we conduct an in-depth exploration of the dormant ratio's interpretability and its potential applications, using it to dynamically adjust hyperparameters. This potentially provides a novel approach for the community to determine the RL agents' training progress and their suitable hyperparameters. Looking ahead, we perceive two main avenues for future RL exploration research. Firstly, the dormant ratio's interpretability is a captivating aspect, and subsequent research could delve into why it has a significant correlation with the diversity and significance of an agent's action, from a theoretical standpoint. Secondly, as the dormant ratio delivers a more precise depiction of an agent's early learning outcomes compared to rewards, it could be used in unsupervised RL. We are confident that the dormant ratio's value extends beyond our current understanding, and that its strategic application could greatly enhance the performance of visual reinforcement algorithms.

REFERENCES

Brandon Amos, Samuel Stanton, Denis Yarats, and Andrew Gordon Wilson. On the model-based stochastic value gradient for continuous reinforcement learning. In *L4DC*, volume 144 of *Proceedings of Machine Learning Research*, pp. 6–20. PMLR, 2021. 9

Jordan Ash and Ryan P Adams. On warm-starting neural network training. *Advances in neural information processing systems*, 33:3884–3894, 2020. 4

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *J. Mach. Learn. Res.*, 3: 397–422, 2002. 9

Adrià Puigdomènech Badia, Pablo Sprechmann, Alex Vitvitskyi, Zhaohan Daniel Guo, Bilal Piot, Steven Kapturowski, Olivier Tieleman, Martín Arjovsky, Alexander Pritzel, Andrew Bolt, and Charles Blundell. Never give up: Learning directed exploration strategies. In *ICLR*. OpenReview.net, 2020. 9

Marc G. Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Rémi Munos. Unifying count-based exploration and intrinsic motivation. In *NIPS*, pp. 1471–1479, 2016. 9

Richard Bellman. A markovian decision process. *Journal of mathematics and mechanics*, pp. 679–684, 1957. 3

Yuri Burda, Harrison Edwards, Amos J. Storkey, and Oleg Klimov. Exploration by random network distillation. In *ICLR (Poster)*. OpenReview.net, 2019. 9

Edoardo Cetin, Philip J. Ball, Stephen J. Roberts, and Oya Çeliktutan. Stabilizing off-policy deep reinforcement learning from pixels. In *ICML*, volume 162 of *Proceedings of Machine Learning Research*, pp. 2784–2810. PMLR, 2022. 2, 6, 8

Annie S. Chen, HyunJi Nam, Suraj Nair, and Chelsea Finn. Batch exploration with examples for scalable robotic reinforcement learning. *IEEE Robotics Autom. Lett.*, 6(3):4401–4408, 2021a. 2, 9

Xinyue Chen, Che Wang, Zijian Zhou, and Keith W. Ross. Randomized ensembled double q-learning: Learning fast without a model. In *ICLR*. OpenReview.net, 2021b. 8

Pierluca D'Oro, Max Schwarzer, Evgenii Nikishin, Pierre-Luc Bacon, Marc G. Bellemare, and Aaron C. Courville. Sample-efficient reinforcement learning by breaking the replay ratio barrier. In *ICLR*. OpenReview.net, 2023. 2, 4, 8

Meire Fortunato, Mohammad Gheshlaghi Azar, Bilal Piot, Jacob Menick, Matteo Hessel, Ian Osband, Alex Graves, Volodymyr Mnih, Rémi Munos, Demis Hassabis, Olivier Pietquin, Charles Blundell, and Shane Legg. Noisy networks for exploration. In *ICLR (Poster)*. OpenReview.net, 2018. 9

Özlem Güzel, Ilker Sahin, and Chris Ryan. Push-motivation-based emotional arousal: A research study in a coastal destination. *Journal of Destination Marketing & Management*, 16:100428, 2020. 2

Danijar Hafner, Timothy P. Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2555–2565. PMLR, 2019. 8

Danijar Hafner, Timothy P. Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *ICLR*. OpenReview.net, 2020. 8

Danijar Hafner, Timothy P. Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *ICLR*. OpenReview.net, 2021. 8

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023. 8

Nicklas Hansen, Hao Su, and Xiaolong Wang. Temporal difference learning for model predictive control. In *ICML*, volume 162 of *Proceedings of Machine Learning Research*, pp. 8387–8406. PMLR, 2022. 8

Harrison and David W. Arousal theory. *Brain Asymmetry and Neural Systems: Foundations in Clinical Neuroscience and Neuropsychology*, pp. 427–435, 2015. 2

Elad Hazan, Sham M. Kakade, Karan Singh, and Abby Van Soest. Provably efficient maximum entropy exploration. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2681–2691. PMLR, 2019. 9

Takuya Hiraoka, Takahisa Imagawa, Taisei Hashimoto, Takashi Onishi, and Yoshimasa Tsuruoka. Dropout q-functions for doubly efficient reinforcement learning. In *ICLR*. OpenReview.net, 2022. 2, 8

Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. VIME: variational information maximizing exploration. In *NIPS*, pp. 1109–1117, 2016. 9

Tianying Ji, Yu Luo, Fuchun Sun, Xianyuan Zhan, Jianwei Zhang, and Huazhe Xu. Seizing serendipity: Exploiting the value of past success in off-policy actor-critic. *arXiv preprint arXiv:2306.02865*, 2023. 5

Chi Jin, Akshay Krishnamurthy, Max Simchowitz, and Tiancheng Yu. Reward-free exploration for reinforcement learning. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4870–4879. PMLR, 2020. 9

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998. 3

Emilie Kaufmann, Pierre Ménard, Omar Darwiche Domingues, Anders Jonsson, Edouard Leurent, and Michal Valko. Adaptive reward-free exploration. In *ALT*, volume 132 of *Proceedings of Machine Learning Research*, pp. 865–891. PMLR, 2021. 9

Dongyoung Kim, Jinwoo Shin, Pieter Abbeel, and Younggyo Seo. Accelerating reinforcement learning with value-conditional state entropy exploration. *arXiv preprint arXiv:2305.19476*, 2023. 9

Aviral Kumar, Rishabh Agarwal, Dibya Ghosh, and Sergey Levine. Implicit under-parameterization inhibits data-efficient deep reinforcement learning. In *ICLR*. OpenReview.net, 2021. 9

Michael Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. In *NeurIPS*, 2020a. 2, 8

Michael Laskin, Aravind Srinivas, and Pieter Abbeel. CURL: contrastive unsupervised representations for reinforcement learning. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5639–5650. PMLR, 2020b. 2, 8

Alex X. Lee, Anusha Nagabandi, Pieter Abbeel, and Sergey Levine. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. In *NeurIPS*, 2020. 8

Lisa Lee, Benjamin Eysenbach, Emilio Parisotto, Eric Xing, Sergey Levine, and Ruslan Salakhutdinov. Efficient exploration via state marginal matching. *arXiv preprint arXiv:1906.05274*, 2019. 9

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *ICLR (Poster)*, 2016. 9

Clare Lyle, Mark Rowland, and Will Dabney. Understanding and preventing capacity loss in reinforcement learning. In *ICLR*. OpenReview.net, 2022. 3, 9

Clare Lyle, Zeyu Zheng, Evgenii Nikishin, Bernardo Ávila Pires, Razvan Pascanu, and Will Dabney. Understanding plasticity in neural networks. In *ICML*, volume 202 of *Proceedings of Machine Learning Research*, pp. 23190–23211. PMLR, 2023. 9

Pierre Ménard, Omar Darwiche Domingues, Anders Jonsson, Emilie Kaufmann, Edouard Leurent, and Michal Valko. Fast active learning for pure exploration in reinforcement learning. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7599–7608. PMLR, 2021a. 9

Pierre Ménard, Omar Darwiche Domingues, Xuedong Shang, and Michal Valko. UCB momentum q-learning: Correcting the bias without forgetting. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7609–7618. PMLR, 2021b. 9

Mirco Mutti, Riccardo De Santi, and Marcello Restelli. The importance of non-markovianity in maximum state entropy exploration. In *ICML*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16223–16239. PMLR, 2022. 9

Evgenii Nikishin, Max Schwarzer, Pierluca D'Oro, Pierre-Luc Bacon, and Aaron C. Courville. The primacy bias in deep reinforcement learning. In *ICML*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16828–16847. PMLR, 2022. 9

Evgenii Nikishin, Junhyuk Oh, Georg Ostrovski, Clare Lyle, Razvan Pascanu, Will Dabney, and André Barreto. Deep reinforcement learning with plasticity injection. *arXiv preprint arXiv:2305.15555*, 2023. 9

Georg Ostrovski, Marc G. Bellemare, Aäron van den Oord, and Rémi Munos. Count-based exploration with neural density models. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2721–2730. PMLR, 2017. 9

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2778–2787. PMLR, 2017. 9

Matthias Plappert, Rein Houthooft, Prafulla Dhariwal, Szymon Sidor, Richard Y. Chen, Xi Chen, Tamim Asfour, Pieter Abbeel, and Marcin Andrychowicz. Parameter space noise for exploration. In *ICLR (Poster)*. OpenReview.net, 2018. 9

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. In *Robotics: Science and Systems*, 2018. 2, 6

Thomas Rückstieß, Frank Sehnke, Tom Schaul, Daan Wierstra, Yi Sun, and Jürgen Schmidhuber. Exploring parameter space in reinforcement learning. *Paladyn J. Behav. Robotics*, 1(1):14–24, 2010. 9

Max Schwarzer, Ankesh Anand, Rishab Goel, R. Devon Hjelm, Aaron C. Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. In *ICLR*. OpenReview.net, 2021. 2, 8

Max Schwarzer, Johan Samir Obando-Ceron, Aaron C. Courville, Marc G. Bellemare, Rishabh Agarwal, and Pablo Samuel Castro. Bigger, better, faster: Human-level atari with human-level efficiency. In *ICML*, volume 202 of *Proceedings of Machine Learning Research*, pp. 30365–30380. PMLR, 2023. 8

Frank Sehnke, Christian Osendorfer, Thomas Rückstieß, Alex Graves, Jan Peters, and Jürgen Schmidhuber. Parameter-exploring policy gradients. *Neural Networks*, 23(4):551–559, 2010. 9

Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8583–8592. PMLR, 2020. 9

Younggyo Seo, Lili Chen, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. State entropy maximization with random encoders for efficient exploration. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9443–9454. PMLR, 2021. 9

Ghada Sokar, Rishabh Agarwal, Pablo Samuel Castro, and Utku Evci. The dormant neuron phenomenon in deep reinforcement learning. In *ICML*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32145–32168. PMLR, 2023. 1, 2, 3, 9

H. Francis Song, Abbas Abdolmaleki, Jost Tobias Springenberg, Aidan Clark, Hubert Soyer, Jack W. Rae, Seb Noury, Arun Ahuja, Siqi Liu, Dhruva Tirumala, Nicolas Heess, Dan Belov, Martin A. Riedmiller, and Matthew M. Botvinick. V-MPO: on-policy maximum a posteriori policy optimization for discrete and continuous control. In *ICLR*. OpenReview.net, 2020a. 9

H. Francis Song, Abbas Abdolmaleki, Jost Tobias Springenberg, Aidan Clark, Hubert Soyer, Jack W. Rae, Seb Noury, Arun Ahuja, Siqi Liu, Dhruva Tirumala, Nicolas Heess, Dan Belov, Martin A. Riedmiller, and Matthew M. Botvinick. V-MPO: on-policy maximum a posteriori policy optimization for discrete and continuous control. In *ICLR*. OpenReview.net, 2020b. 8

Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling representation learning from reinforcement learning. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9870–9879. PMLR, 2021. 2, 8

Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. #exploration: A study of count-based exploration for deep reinforcement learning. In *NIPS*, pp. 2753–2762, 2017. 9

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018. 2, 6, 9

Xiyao Wang, Ruijie Zheng, Yanchao Sun, Ruonan Jia, Wichayaporn Wongkamjan, Huazhe Xu, and Furong Huang. Coplanner: Plan to roll out conservatively but to explore optimistically for model-based rl, 2023. 9

Pawel Wawrzynski. Control policy with autocorrelated noise in reinforcement learning for robotics. *International Journal of Machine Learning and Computing*, 5(2):91, 2015. 9

Qisong Yang and Matthijs T. J. Spaan. CEM: constrained entropy maximization for task-agnostic safe exploration. In *AAAI*, pp. 10798–10806. AAAI Press, 2023. 9

Denis Yarats, Ilya Kostrikov, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *ICLR*. OpenReview.net, 2021. 2, 8

Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. In *ICLR*. OpenReview.net, 2022. 2, 6, 8, 9

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *CoRL*, volume 100 of *Proceedings of Machine Learning Research*, pp. 1094–1100. PMLR, 2019. 2, 6

Tianjun Zhang, Paria Rashidinejad, Jiantao Jiao, Yuandong Tian, Joseph E. Gonzalez, and Stuart Russell. MADE: exploration via maximizing deviation from explored regions. In *NeurIPS*, pp. 9663–9680, 2021. 9

Ruijie Zheng, Xiyao Wang, Yanchao Sun, Shuang Ma, Jieyu Zhao, Huazhe Xu, Hal Daumé III, and Furong Huang. `TACO`: Temporal latent action-driven contrastive loss for visual reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=ezCsMOy1w9`. 2, 6, 8

A MORE VISUALIZATION RESULTS OF DORMANT RATIO

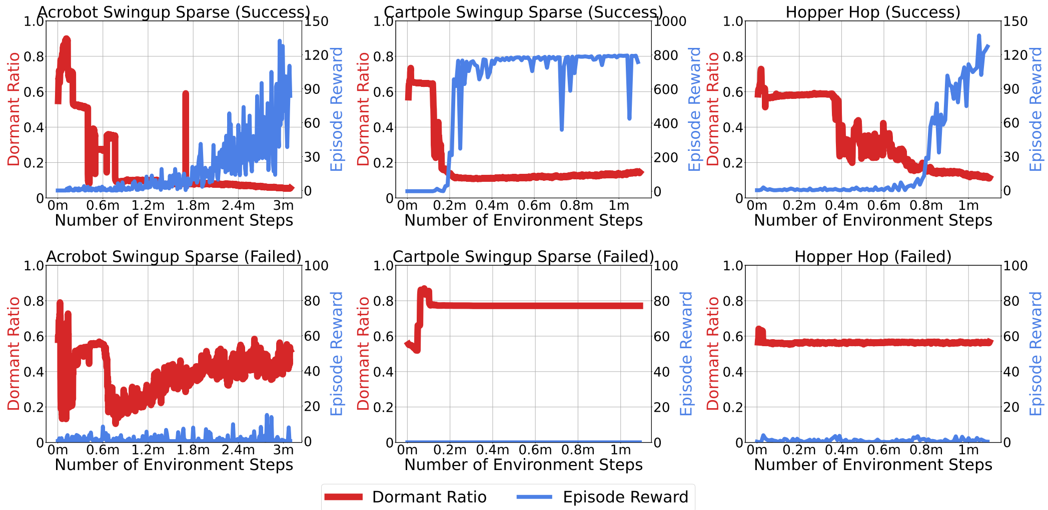


Figure 11: Analysis of the dormant ratio in successful vs. broken seeds reveals distinct behavior patterns. In a successful seed, a decreasing dormant ratio allows the agent to effectively explore the environment and learn skills. Conversely, in a broken seed, the agent becomes immobile and fails to discover meaningful motions.

B TIME EFFICIENCY OF DrM

To assess the algorithms' speed, we measure their frames per second (**FPS**) on the same DeepMind Control Suite task, Dog Walk, using an identical Nvidia RTX A5000 GPU. As Figure 12 shows, while achieving significant sample efficiency and asymptotic performance, DrM only slightly compromises wall-clock time compared to **DrQ-v2**. Compared with two other baselines, DrM is roughly as time-efficient as **ALIX** and about three times faster than **TACO**, which needs a batch size four times larger than that of **DrQ-v2** to compute its temporal contrastive loss.

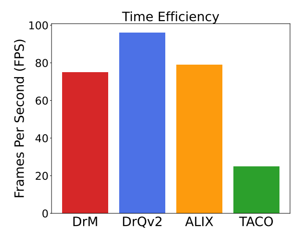


Figure 12: Comparison of time-efficiency

C IMPLEMENTATION DETAILS

In this section, we describe the implementation details of `DrM` . We have built `DrM` upon the publicly available source code of DrQ-v2. Subsequently, we present the pseudo-code outlining our approach.

C.1 DORMANT RATIO CALCULATION

In this subsection, we demonstrate how the dormant ratio is calculated in Algorithm 1.

Algorithm 1 Dormant Ratio Calculation

procedure CAL_DORMANT_RATIO(model, inputs, τ-dormant threshold)
    Initialize counters: total_neurons, dormant_neurons
    Operate a forward propagation: model(inputs)
    for each module in model **do**
        if module is Linear Layer **then**
            Compute average output of neurons
            Identify dormant neurons based on τ-dormant threshold
            Update total_neurons and dormant_neurons
        end if
    end for
    return dormant ratio = dormant_neurons/total_neurons
end procedure

C.2 DORMANT-RATIO-GUIDED PERTURBATION

In this subsection, we demonstrate how the perturbation is performed based on the dormant ratio in Algorithm 2.

Algorithm 2 Dormant-ratio-guided Perturbation

procedure PERTURB(network, optimizer, perturb_factor)
    Create a deep copy of the network and store in new_net
    Initialize weights of new_net
    for each layer and parameter in the network **do**
        if layer is Linear Layer **then**
            Compute noise as: new_net $\times$ $(1 -$ perturb_factor$)$
            Update parameter with: net $\times$ perturb_factor $+$ noise
        end if
    end for
    Reset the state of the optimizer
    return updated network, optimizer
end procedure

C.3 AWAKEN EXPLORATION SCHEDULER

In this subsection, we demonstrate how the awaken exploration scheduler is performed in Algorithm 3.

Algorithm 3 Awaken Exploration Scheduler

```
Initialize awaken_step to None
function STDDEV(step)
    if awaken_step is None then
        return dormant_stddev
    else
        linear_stddev = linear_schedule(step − awaken_step)
        return max(dormant_stddev, linear_stddev)
    end if
end function
function UPDATE_AWAKEN_STEP(step)
    if awaken_step is None and dormant_ratio < dormant_ratio_threshold then
        awaken_step ← step
    end if
end function
```

C.4 DORMANT-RATIO-GUIDED EXPLOITATION

In this subsection, we demonstrate how the dormant-ratio-guided exploitation is performed in Algorithm 4.

Algorithm 4 Dormant-ratio-guided exploitation

```
function UPDATE_VALUE_NETWORK(obs, action)
    Q1, Q2 = critic(obs, action)
    Q = min(Q1, Q2)
    V = V_net(obs)
    error = V − Q
    sign = { 1   if error > 0
           { 0   otherwise
    weight = (1 − sign)expectile + sign(1 − expectile)
    value_loss = mean(weight × error^2)
    Update value network using value_loss
end function
function CAL_TARGET_Q(next_obs, reward, discount)
    action distribution = actor(next_obs, awaken exploration scheduler)
    Sample next_action from the distribution with clipping
    target_Q1, target_Q2 = critic_target(next_obs, next_action)
    target_V_explore = min(target_Q1, target_Q2)
    target_V_exploit = V_net(next_obs)
    target_V = λ × target_V_exploit + (1 − λ) × target_V_explore
    target_Q = reward + (discount × target_V)
    return target_Q
end function
```

D Hyperparameters

We summarize all the hyperparameters in Table 1.

Parameter	Setting
Replay buffer capacity	10^6
Action repeat	2
Seed frames	4000
Exploration steps	2000
n-step returns	3
Mini-batch size	256
Discount γ	0.99
Optimizer	Adam
Learning rate	8×10^{-5} (DeepMind Control Suite) 10^{-4} (MetaWorld & Adroit)
Agent update frequency	2
Soft update rate	0.01
Features dim.	100 (Humanoid & Dog) 50 (Others)
Hidden dim.	1024
τ-Dormant ratio	0.025
Dormant ratio threshold $\hat{\beta}$	0.2
Minimum perturb factor α_{min}	0.2
Maximum perturb factor α_{max}	0.6 (Dog [Stand, Walk, Run], Humanoid Run, Coffee Push & Soccer) 0.9 (Others)
Perturb rate k	2
Perturb frames	200000
Linear exploration stddev. clip	0.3
Linear exploration stddev. schedule	linear(1.0, 0.1, 2000000) (DeepMind Control Suite) linear(1.0,0.1,300000) (MetaWorld & Adroit)
Awaken exploration temperature T	0.1
Target exploitation parameter $\hat{\lambda}$	0.6
Exploitation temperature T'	0.02
Exploitation expectile	0.7 (Adroit) 0.9 (Others)

Table 1: A default set of hyper-parameters used in our experiments.