V-Simba: Unleashing the Architectural Potential of RL in Visual Continuous Control

Donghu Kim[1*] **Youngdo Lee**[1] **Hojoon Lee**[1] **Johan Obando-Ceron**[2,3] **Byungkun Lee**[1]

Aaron Courville[2,3,5] **Pablo Samuel Castro**[2,3] **Jaegul Choo**[1] **Clare Lyle**[4]

[1]KAIST [2]Mila – Québec AI Institute [3]Université de Montréal
[4]Google DeepMind [5]Canada CIFAR AI Chair

Abstract

Improving sample efficiency remains a core challenge in reinforcement learning (RL), especially in real-world settings like robotics, where data collection is costly. This challenge is pronounced in visual RL, where high-dimensional inputs often obscure learning signals. While prior work in visual RL has focused on algorithmic solutions, such as better dynamics models or exploration strategies, recent advances in state-based RL show that architectural design alone can lead to significant gains in sample efficiency. This raises an important question: *Can these architectural principles transfer to visual RL?* In response, we introduce **V-Simba**, a simple yet effective visual RL architecture inspired by the Simba architecture from state-based RL. Built on top of Soft Actor-Critic with data augmentation, V-Simba modifies the architecture by adding normalization layers to stabilize training and using pointwise convolutions to reduce computation. Despite its simplicity, V-Simba matches or outperforms the state-of-the-art methods across DMC, Adroit, and Meta-World benchmarks, while being more computationally efficient than DrQ-v2.

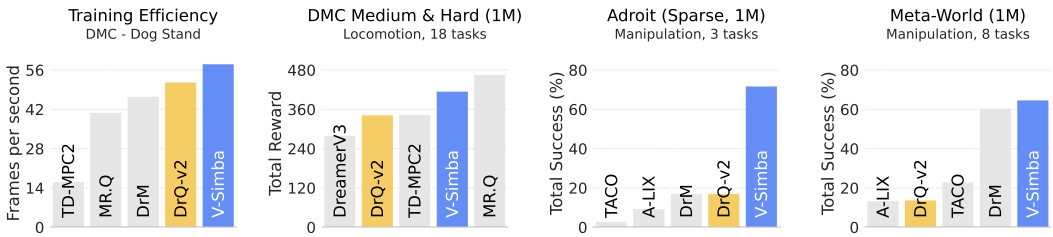


Figure 1: **Benchmark Summary.** We evaluate the effectiveness of V-Simba across 29 visual continuous control tasks spanning multiple domains, with a *single* set of hyperparameters. By incorporating V-Simba into Soft Actor-Critic with data augmentation, it matches or outperforms state-of-the-art visual RL methods, demonstrating better sample and compute efficiency.

1 Introduction

Deep reinforcement learning (RL) has long been a prominent approach for solving continuous control tasks. However, RL typically relies on an extensive amount of trial-and-error within the environment, which is often expensive in terms of time, compute, and real-world constraints. This issue is exacerbated in visual RL, where agents must learn from high-dimensional, noisy, and

*Corresponding Author

Preprint.

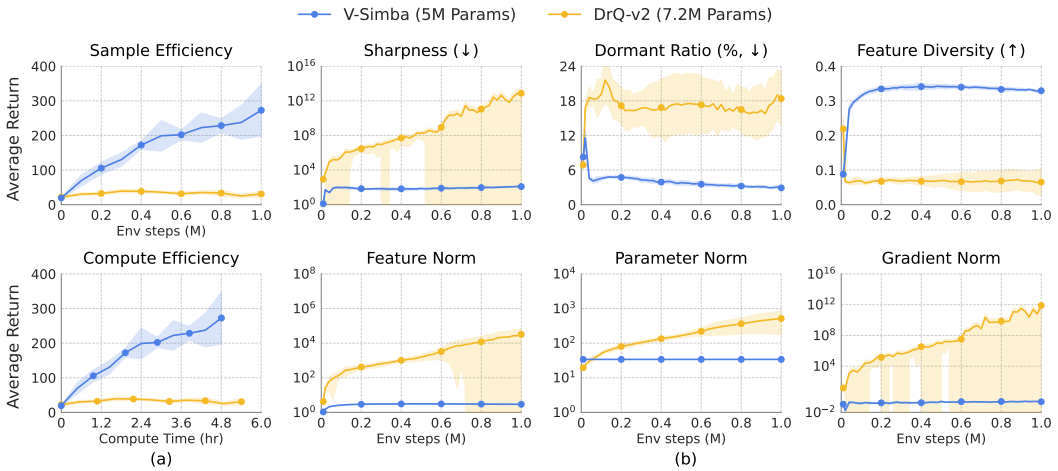


Figure 2: **DrQ-v2 v.s. V-Simba.** Comparison of the DrQ-v2 architecture and our V-Simba in the `Dog Stand` environment. Both are evaluated on Soft Actor-Critic (SAC) with results averaged over 5 seeds. **(a)** V-Simba is substantially more efficient than DrQ-v2 in terms of both sample and compute. **(b)** Unlike DrQ-v2, V-Simba has stable learning dynamics, indicated by smooth loss landscape, low dormant ratio, high feature diversity, and well-controlled feature, parameter, and gradient norms. Detailed explanations of each metric are provided in Appendix A.2.

often partially observable image inputs. Consequently, improving sample efficiency (i.e., learning effectively from limited interaction data) has become a key research topic in visual RL.

To improve sample efficiency, recent work has largely concentrated on algorithmic innovations, including enhanced representation learning [73], latent dynamics modeling [14, 79], world models [23, 22], and improved exploration strategies [5, 71]. Yet, despite these advances, the underlying neural architectures have remained relatively simple. A prominent example is DrQ-v2 [72], which combines the DDPG algorithm [38] with data augmentation [32]. Its architecture consists of a shallow convolutional encoder, followed by a large fully connected layer and a single layer normalization layer [37] in-between. Due to its simplicity and strong empirical performance, DrQ-v2 has become the de facto standard in visual RL, and many state-of-the-art methods [71, 79, 7, 62] adopt DrQ-v2's architecture with minimal modifications.

However, our analysis reveals that this commonly adopted architecture suffers from severe training instabilities. As shown in the top row of Figure 2.(b), DrQ-v2 exhibits sharp loss landscapes that correlate with poor generalization [13, 34], a high fraction of dormant units representing plasticity loss [58], and low feature diversity indicating feature collapse [69]. In contrast, recent advances in state-based RL [35, 4, 36, 50] demonstrate that carefully designed architectures can effectively mitigate these instabilities. Notably, the Simba series [35, 36] introduces principled architectural guidelines that stabilize training by constraining the growth of features, weights, and gradients through targeted normalization and regularization.

While effective in state-based tasks, the Simba architecture lacks a suitable inductive bias for visual data. We propose **V-Simba**, a simple yet effective architecture for visual continuous control, as a means of applying the underlying principles from the Simba model series to visual RL domains. Built on top of Soft Actor-Critic (SAC) [21], V-Simba incorporates three core architectural components: (1) normalization layers (LN) to control feature norms, (2) ℓ_2 weight regularization to limit parameter growth, and (3) a distributional critic with reward normalization to stabilize gradients. To ensure computational efficiency, V-Simba applies early downsampling via large-stride convolutions and makes extensive use of lightweight pointwise convolutions [26].

We evaluate V-Simba on three standard benchmarks: DMControl [63], Adroit [52], and Metaworld [77] using a single set of hyperparameters across all tasks. Despite its simplicity, V-Simba consistently outperforms DrQ-v2, while reducing both model size (7.2M $\rightarrow$ 5.0M) and training time (5.4 $\rightarrow$ 4.8 hours for 1M DMControl steps). Moreover, V-Simba is competitive with leading vision-based methods, matching or surpassing MR.Q [14] and TD-MPC2 [23] in DMControl and outperforming DrM [71] and TACO [79] in dexterous manipulation tasks.

V-Simba is intended to offer a strong, stable, and efficient architectural foundation for advancing visual continuous control. We hope our work highlights the untapped potential of principled architecture design within the visual RL community.

2 Related Work

Learning solely from high-dimensional visual observations poses significant challenges in RL. Due to the partially observable nature of the observation space (Section 3.1), visual RL agents suffer from poor sample efficiency and large generalization gaps compared to their state-based counterparts [43].

Algorithmic approaches for visual RL have primarily focused on: (1) representation learning via auxiliary tasks predicting future latent states, either as auxiliary losses in model-free methods [61, 79, 53, 55, 29, 33, 66, 76, 17, 57, 48, 75, 16, 45, 14] or separate dynamics models in model-based methods [23, 22, 39, 70, 20, 12, 68]; (2) data augmentation, especially random shifts [32, 30, 72], for better efficiency and generalization; and (3) exploration methods, including planning [56, 67], curiosity-driven [51, 6, 19], and information maximization [62]. These algorithmic innovations have driven rapid progress in visual RL, leading to continuous improvements in sample efficiency.

Architectural design for visual RL has received comparatively little attention compared to algorithmic innovations. The field has largely maintained shallow convolutional neural network (CNN) architectures similar to the one established by DQN [46] over a decade ago. While some works have incorporated architectural elements from computer vision—such as ResNet-like architectures in Impala [9], BBF [54], and EfficientZero [74], or transformers in DTQN [10]—these modifications were often introduced alongside complex algorithmic methods. This entanglement has obscured the true contribution of architectural design to performance improvements. Recent studies have identified the benefits of normalization techniques [42, 1, 41], but these have generally been applied to conventional CNN encoders with minimal architectural modifications. To the best of our knowledge, aside from a few isolated attempts such as adding global average pooling [65] or using Mixture-of-Experts [49, 60], no substantial architectural innovations have been sufficiently explored in visual RL [9, 27].

This paper explores the untapped potential of neural architecture design for visual RL. Building on recent architectural successes in state-based RL [28, 47, 35, 36], we show that architectural changes alone can yield significant performance improvements without complex algorithmic modifications.

3 Preliminary

As a preliminary, we briefly describe the problem setup of visual RL, DrQ-v2 architecture [72], and the Soft Actor-Critic algorithm [21], as these form the foundation for our proposed architecture.

3.1 Visual Reinforcement Learning

Reinforcement learning (RL) is typically formulated as a Markov Decision Process (MDP) [3], defined by the tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$ of state space $\mathcal{S}$, action space $\mathcal{A}$, transition function $P : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$, reward function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and discount factor $\gamma \in [0, 1)$. From an initial state $s_0 \in \mathcal{S}$, the objective is to find an optimal policy $\pi^* : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ that maximizes the expected discounted return $\mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)]$. Visual RL is a subclass of this problem where the agent does not have access to the true state $s \in \mathcal{S}$, but instead receives high-dimensional pixel observations $o \in \mathcal{O}$ of the system. Since these observations may not fully capture the true state, the problem is modeled as a Partially Observable Markov Decision Process (POMDP) [3] represented by the tuple $(\mathcal{S}, \mathcal{O}, \mathcal{A}, P, R, \gamma)$ where $\mathcal{O}$ denotes the observation space.

3.2 Data-regularized Q-learning

Data-regularized Q-learning (DrQ-v2) [72] is a model-free RL algorithm that has emerged as a strong baseline in visual RL due to its simplicity, efficiency, and competitive performance. It builds upon the Deep Deterministic Policy Gradient (DDPG) algorithm [38] by incorporating two key modifications: (1) extensive use of data augmentation via random shift transformations, and (2) target Q-function stabilization through exponential moving average (EMA) updates.

At its core, DrQ-v2 improves sample efficiency in off-policy learning by generating augmented views of each observation, thereby increasing data diversity. This augmentation acts as a regularizer, mitigating overfitting to specific visual patterns. Despite its empirical effectiveness, DrQ-v2 employs a notably lightweight architecture: a shallow convolutional encoder followed by an MLP-based prediction head, with a single normalization layer [37] in between.

While DrQ-v2 has become the de facto architecture for many recent visual RL algorithms [71, 79], its architectural simplicity leaves room for improvement in stability and representational capacity.

3.3 Soft Actor-Critic (SAC)

Soft Actor-Critic (SAC) is a prominent off-policy algorithm for continuous control. It aims to maximize both expected cumulative reward and policy entropy, where $\tau = (o, a, r, o')$ represents a transition tuple. SAC comprises a stochastic policy $\pi_\theta(a|o)$, a Q-function $Q_\phi(o, a)$, and an entropy coefficient α that balances reward maximization and entropy regularization. The policy network is optimized to maximize the expected return while encouraging exploration through entropy. This objective is formalized as:

$$\mathcal{L}_\pi = \mathbb{E}_{\bar{a}\sim\pi_\theta}\left[\alpha \log \pi_\theta(\bar{a}|o) - Q_\phi(o, \bar{a})\right]. \tag{1}$$

The Q-function $Q_\phi(o, a)$ is trained to minimize the Bellman residual:

$$\mathcal{L}_Q = \left(Q_\phi(o, a) - \left(r + \gamma Q_{\bar{\phi}}(o', a') - \alpha \log \pi_\theta(a'|o')\right)\right)^2, \tag{2}$$

where $a' \sim \pi_\theta(\cdot|o')$, and $Q_{\bar{\phi}}$ represents the target Q-network updated via an exponential moving average of ϕ.

4 Method

V-Simba leverages architectural design from state-based RL to stabilize optimization dynamics and improve computational efficiency in visual RL. Our design follows two core principles: (1) stabilizing optimization (Section 4.1), and (2) maintaining computational efficiency (Section 4.2). The final architecture builds on these principles (Section 4.3).

4.1 Design Philosophy I: Stabilizing Optimization

As shown in Figure 2, DrQ-v2 suffers from unstable optimization during training. While layer normalization (LN) [37] and residual connections [24] effectively stabilize supervised learning, visual RL methods often underuse them—DrQ-v2, for instance, employs only a single normalization layer without residuals. We incorporate both components to improve stability.

However when adding LayerNorm, one must consider its relationship with the gradient. Concretely, LayerNorm introduces scale invariance: for any scalar $c > 0$ and weight matrix W,

$$\text{Norm}(cWx) = \text{Norm}(Wx), \tag{3}$$

which causes gradients to scale inversely with parameter magnitude:

$$\nabla_W \text{Norm}(cWx) = \frac{1}{c}\nabla_W \text{Norm}(Wx). \tag{4}$$

As parameter norms grow during training, gradients diminish, reducing learning ability [41, 50]. Moreover, uneven growth across layers causes inconsistent gradient scales, destabilizing optimization [36]. This highlights the importance of controlling weight and gradient norms, in addition to the feature norm. Thus, we employ the following design choices to achieve stable norms.

We first opt LayerNorm as the forefront layer of both encoder and critic module, in order to control the norm of not only their intermediate features but also their inputs. While unusual for convolutional networks, this resembles the Dual PatchNorm design [31] which has been empirically shown to stabilize the gradients of embedding layer [2]. For preventing parameter growth, we surprisingly found a simple ℓ_2 weight regularization to be sufficient, as shown in Figure 2.

[2]In practice, we adopt the shift-and-norm strategy introduced in SimbaV2 [36] to preserve magnitude information. We use ℓ_2-norm for action inputs however, as when $|\mathcal{A}| = 1$ shift-and-LN always outputs $[-1, 1]$.

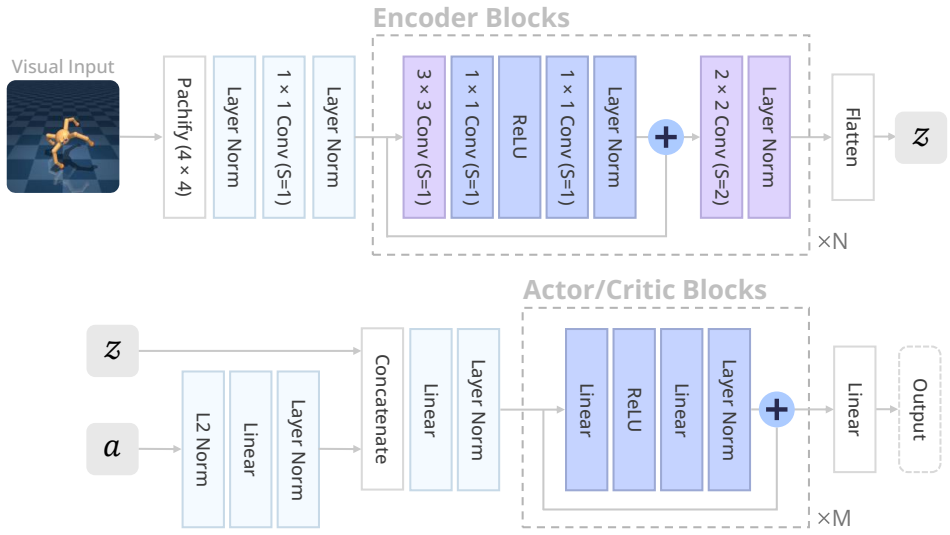


Figure 3: **V-Simba architecture.** We aim to develop an architecture that constrains its feature norm, weight norm, and gradient norm for better stability and generalization. Precisely, we make extensive use of layer normalization and residual connections for stable features and gradients, and incorporate point-wise convolutions in tandem with spatial convolutions for computational efficiency.

Finally, we further stabilize gradients by employing a distributional critic with KL divergence loss [2] and reward normalization [36]. The KL divergence loss is more robust to noisy targets than mean squared error due to its smoother loss landscape [11], while reward normalization ensures consistent learning signals despite varying reward scales.

Specifically, reward normalization maintains unit variance in expected returns. Given reward r_t at time t, we track the discounted return:

$$G_t \leftarrow \gamma G_{t-1} + r_t \tag{5}$$

with G_t re-initialized to 0 at the start of each episode. Let $\sigma^2_{t,G}$ denotes the running variance of G_t. Each reward is then scaled as:

$$\bar{r}_t \leftarrow \frac{r_t}{\sqrt{\sigma^2_{t,G} + \epsilon}}, \tag{6}$$

4.2 Design Philosophy II: Maintaining Computational Efficiency

Adding normalization layers and regularizations increases training cost, so reducing computation is crucial. We find that most of DrQ-v2's computation cost comes from early convolutional layers processing high-resolution inputs. We apply early downsampling via large-stride convolutions, a common practice in ResNet [25], ConvNeXt [40], and Vision Transformer [8]. This results in an early reduction in spatial resolution and in turn, the computational cost of subsequent convolution layers.

We further cut computation by replacing many spatial convolutions with pointwise (1×1 kernel) convolutions [26], which operate channel-wise without mixing spatial information, preserving spatial details at a lower cost.

4.3 V-Simba Architecture

Building on our design principles, we now detail the V-Simba architecture (Figure 3).

Image Preprocessing. The input $o \in \mathbb{R}^{84\times84\times9}$ is a stack of the last three RGB frames. We first apply a patchify operation (with 4×4 patches) to reduce the spatial resolution. The flattened patches are then processed through LayerNorm, a pointwise (1×1) convolution with 32 output channels and another LayerNorm, resulting in normalized features $f_0 \in \mathbb{R}^{21\times21\times32}$.

Encoder. The encoder consists of two sequential blocks transforming and downsampling features: $f_0 \xrightarrow{\text{Block}_1} f_1 \xrightarrow{\text{Block}_2} f_2$, where $f_1 \in \mathbb{R}^{10\times10\times32}$, $f_2 \in \mathbb{R}^{5\times5\times32}$.

Each encoder block processes input f_i as follows:

1. A 3×3 convolution (stride 1, padding 1) to aggregate spatial features without changing resolution.
2. Two pointwise (1×1) convolutions with nonlinearities to refine and filter features. Here, we employ an inverted bottleneck with $4\times$ expansion, following ConvNext [40] and Simba [35].
3. Feature normalization to align magnitudes between residual and nonlinear paths, followed by a residual connection for stable gradient flow.
4. Downsampling with a 2×2 convolution with stride 2 to reduce spatial resolution.
5. Applying LayerNorm to normalize the features before passing to the next block.

After the second block, f_2 is flattened into the latent state vector $z \in \mathbb{R}^{800}$.

Predictor. The latent vector z feeds into separate actor and critic heads, each followed by a linear layer and LayerNorm. For the critic, actions are separately embedded and concatenated with image embedding. The actor and critic embeddings have dimensions $z_\pi \in \mathbb{R}^{128}$, $z_Q \in \mathbb{R}^{512}$ following [35].

Each embedding passes through residual nonlinear blocks: one block for the actor and two for the critic. Finally, the actor output passes through a linear layer with $\tanh$ activation, while the critic output passes through a linear layer modeling the Q-value distribution.

5 Experiments

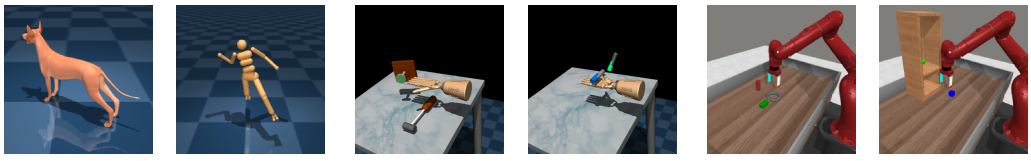
We now provide an empirical evaluation of V-Simba:

1. **Performance Evaluation** (Sections 5.2). Compare V-Simba against leading visual RL methods to demonstrate its effectiveness across diverse benchmarks.
2. **Ablation Study** (Section 5.3.) Conduct ablation studies on architectural component of V-Simba.

5.1 Experimental Setup

Environment. We consider a total of 29 continuous control tasks spanning 3 benchmarks: DeepMind Control (DMC) Suite [63], Adroit [52], and Meta-World [77]. Figure 4 shows the visualization of each task. These environments pose diverse challenges, including high-dimensional action spaces, sparse rewards, and complex dexterous manipulation, often under rich visual observations with shading and textures. Consequently, to solve the tasks, prior visual RL methods typically require either large volumes of frames or privileged information such as low-level robot states.

Baselines. In experiments, we compare V-Simba against a diverse set of state-of-the-art visual RL methods exemplifying three key algorithmic strategies: data and model regularization (DrQ-v2 [72], A-LIX [7]), advanced exploration (DrM [71]), and model-based representation learning (TACO [79], TD-MPC2 [23], MR.Q [14]). Notably, A-LIX, TACO, and DrM build upon DrQ-v2 (see Section 3.2): A-LIX stabilizes training by adaptively regularizing the encoder's gradients; TACO leverages a latent dynamics loss for richer representations; and DrM integrates dormant ratio [59]-guided mechanisms that balance exploration-exploitation dynamically. While these variants benefit from task-specific hyperparameter tuning, our method uses the *same* hyperparameters across all tasks. Whenever possible, we report original paper results; otherwise, we run the authors' official implementations.



(a) DeepMind Control Suite (b) Adroit (c) Meta-World

Figure 4: **Environment Visualization.** We evaluate our V-Simba on 3 visual continuous control benchmarks: DeepMind Control Suite [63], Adroit [52], and Meta-World [77].

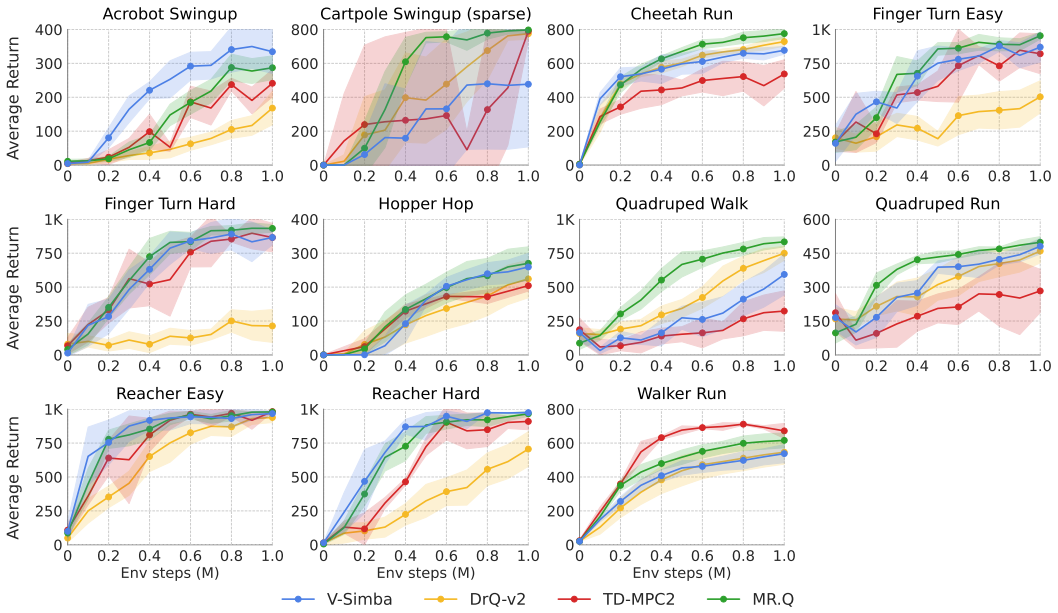


Figure 5: **DeepMind Control Suite - Medium.** Average episode returns on 11 medium-difficulty tasks from DeepMind Control Suite [63]. Each curve represents the mean performance across 3-5 random seeds per algorithm, with shaded areas indicating 95% bootstrap confidence intervals.

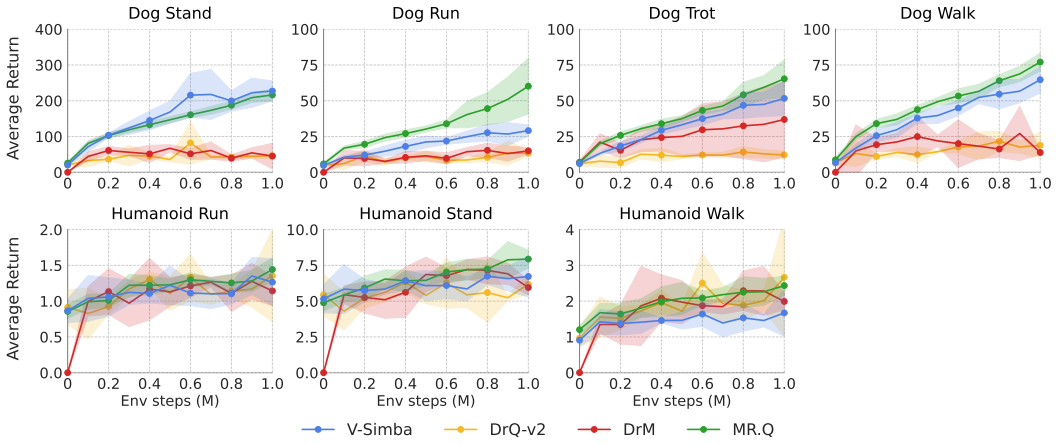


Figure 6: **DeepMind Control Suite - Hard.** Average episode returns on 7 hard-level tasks from DeepMind Control Suite [63]. Each curve represents the mean performance across 3-5 random seeds per algorithm, with shaded areas indicating 95% bootstrap confidence intervals.

5.2 Performance Evaluation

DMC Medium. We begin by evaluating V-Simba on DMC Medium, consisting of 11 mid-difficulty tasks from DMC. As shown in Figure 5, our base algorithm, DrQ-v2, falls behind model-based methods such as TD-MPC2 and MR.Q. However, simply replacing DrQ-v2's neural network with our proposed architecture, V-Simba, yields substantial performance gains. As a result, V-Simba surpasses TD-MPC2 and achieves results competitive with leading algorithm, MR.Q, highlighting the impact of architectural improvements.

DMC Hard. We further assess V-Simba on DMC Hard, a set of 7 high-difficulty tasks in DMC, characterized by complex kinematics and high-dimensional control. Figure 6 shows that V-Simba performs competitively with MR.Q, though full task success remains elusive. We believe that this observation suggests that concurrent advances in both algorithm design and architectural representation are needed in visual RL, to close the gap with state-based performance.

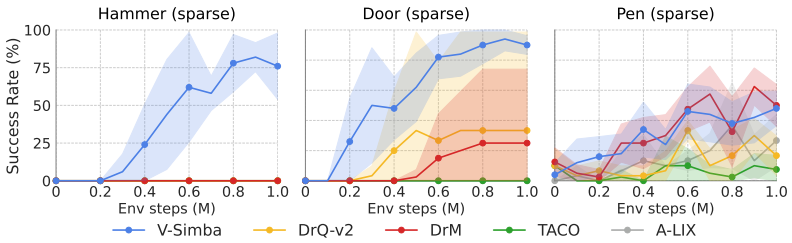


Figure 7: **Adroit - Sparse.** Average success rates on 3 sparse-reward tasks from Adroit [52]. Results for V-Simba are averaged over 5 random seeds, while baselines use 3 random seeds. Shaded regions indicate 95% bootstrap confidence intervals.

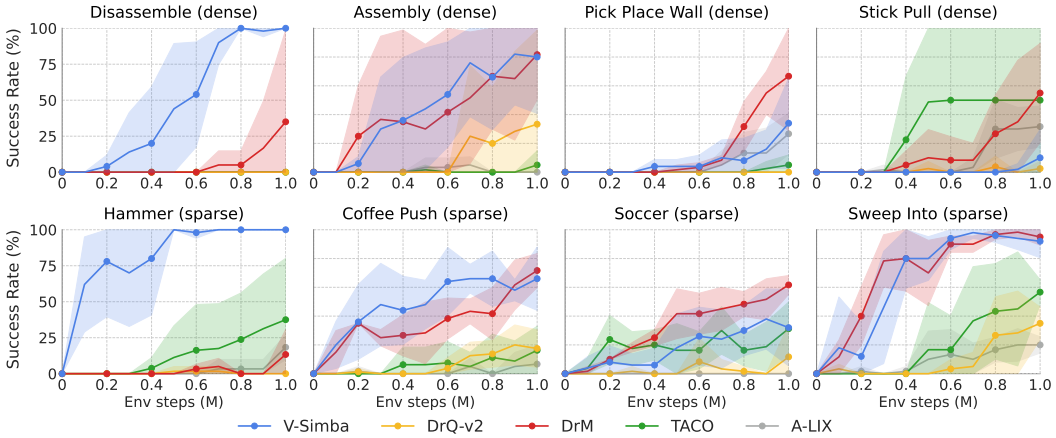


Figure 8: **Meta-World.** Average success rates on 8 tasks from the Meta-World [77]. Results for V-Simba are averaged over 5 random seeds, while baselines use 4 random seeds. Shaded regions indicate 95% bootstrap confidence intervals.

Adroit - Sparse. Moving to more intricate scenarios, we evaluate V-Simba on Adroit under the challenging *sparse-reward* setting. In this domain, the agent must control a dexterous hand-arm system to perform complex manipulation such as opening a door or using tools like a hammer. These tasks pose significant challenges for visual RL, often requiring over 5 million environment frames and access to privileged robot state inputs for successful learning. The comparison results are shown in Figure 7. V-Simba reliably solves or approaches solving all tasks using only 1 million frames. In contrast, DrM–the previous state-of-the-art method–fails to learn meaningful behavior, despite leveraging privileged state vectors. Notably, V-Simba is the only method to solve `Hammer` with 1 million environment steps. These results underscore V-Simba's strong sample-efficiency and effectiveness in high-dimensional visual control settings.

Meta-World. We also benchmark V-Simba on Meta-World, which demands precise object manipulation. We consider 4 medium-difficulty tasks: `Coffee Push`, `Soccer`, `Sweep Into`, and `Hammer`, and 4 high-difficulty tasks: `Assembly`, `Stick Pull`, `Pick Place Wall`, and `Disassemble`. For the medium tasks, we adopt a sparse-reward setting by replacing the ground-truth reward functions with binary success signals, following [77], to increase task difficulty. As shown in Figure 8, while DrQ-v2 struggles to learn in most tasks, V-Simba significantly improves over DrQ-v2 and matches or surpasses leading baselines, demonstrating superior sample efficiency. A notable performance improvement can be seen in the `Disassemble` and `Hammer` task, where V-Simba was able to consistently achieve almost perfect success rate, whereas prior works have failed in few trials.

5.3 Ablation Study

To assess the impact of each component on V-Simba's overall performance, we evaluate variants that remove or modify one component at a time. The results are reported in Table 1.

Table 1: **Ablation Study.** We exclude or modify each component in V-Simba and report their final performance on each benchmark, averaged over 3 random seeds. Each cell is highlighted base on their relative percentile difference to V-Simba, namely: positive (> 0.01), mildly negative $[-0.05, -0.01)$, damaging $[-0.1, -0.05)$, and catastrophic $[-1.0, -0.1)$.

Ablation	**DMC (18)** Return (1k)	**Adroit (3)** Success Rate	**MetaWorld (8)** Success Rate	**All (29)** -
Normalization Layers				
(a) No Normalization Layers	0.418 $\pm$ 0.095	0.756 $\pm$ 0.122	0.533 $\pm$ 0.167	0.484 $\pm$ 0.081
(b) No Input Normalization	0.379 $\pm$ 0.098	0.667 $\pm$ 0.133	0.688 $\pm$ 0.158	0.494 $\pm$ 0.083
(c) LN $\rightarrow$ LN w/o γ, β	0.428 $\pm$ 0.073	0.727 $\pm$ 0.133	0.628 $\pm$ 0.125	0.514 $\pm$ 0.063
(d) LN $\rightarrow$ Zero-Center L2 Norm	0.412 $\pm$ 0.074	0.713 $\pm$ 0.107	0.642 $\pm$ 0.130	0.507 $\pm$ 0.063
Residual and Weight Decay				
(e) No Residual Connection	0.428 $\pm$ 0.098	0.678 $\pm$ 0.178	0.662 $\pm$ 0.154	0.518 $\pm$ 0.081
(f) No Weight Decay	0.427 $\pm$ 0.099	0.644 $\pm$ 0.144	0.550 $\pm$ 0.171	0.484 $\pm$ 0.080
Value Learning				
(g) No Categorical Critic	0.417 $\pm$ 0.096	0.644 $\pm$ 0.144	0.617 $\pm$ 0.167	0.496 $\pm$ 0.079
(h) No Reward Scaling	0.415 $\pm$ 0.094	0.744 $\pm$ 0.144	0.642 $\pm$ 0.158	0.511 $\pm$ 0.080
V-Simba	0.441 $\pm$ 0.075	0.789 $\pm$ 0.167	0.715 $\pm$ 0.115	0.540 $\pm$ 0.060

We first investigate the effect of normalization layers (Table 1.(a)-(d)). **No Normalization Layers** removes LayerNorm entirely from the network, whereas **No Input Normalization** only removes two LayerNorms: for image and action inputs in encoder and critic respectively. **LN w/o** γ, β removes the bias and scale parameters of LayerNorm. In **Zero-Center L2 Norm**, we normalize the L2-norm of features instead of standard deviation. By removing certain layers or components of normalization, the network loses the control over the feature norms, leading to degradation in performance.

Next, we quantify the importance of residual connections and weight decay (Table 1.(e)-(f)). Both residual connection and weight decay, along with their well-known benefits, are also known to bias the network towards simple solutions for improved robustness [64, 35]. Removing such components led to visible drop in performance, similar to removing normalization layers.

Finally, categorical critic and reward scaling are critical components, as they reformulate the regression problem into a categorical prediction, giving a much more stable gradient and learning dynamics. Reverting back to regression loss led to diminished performance (Table 1.(g)). Even with categorical loss, leaving no bounds to the reward scales led to similar consequences (Table 1.(h)), highlighting the importance of assuring the Q-values to stay in a certain range.

6 Lessons and Opportunities

In this work, we introduce V-Simba, a simple yet effective neural network architecture for visual continuous control, inspired by the Simba architecture from state-based RL [35]. By combining feature normalization, weight regularization, and a distributional critic, V-Simba achieves superior performance over prior visual RL methods across multiple benchmarks with minimal algorithmic changes. Additionally, V-Simba reduces computational cost by integrating early downsampling through large-stride convolution and pointwise convolution layers, enabling faster training than DrQ-v2 [72]. We believe our work does not oppose the current trend of adopting model-based learning or exploration strategies; rather, it offers a complementary approach that can be integrated with subsequent studies.

Moreover, in recent years, reinforcement learning for robotic control has gained increased attention. However, limited sample efficiency remains a significant barrier to real-world adoption. While simulators provide valuable virtual environments [44, 78], rendering high-resolution images with complex object interactions is still computationally expensive and difficult to parallelize. This underscores the importance of improving sample efficiency. V-Simba offers a lightweight architectural solution using well-established components that are easy to integrate into existing algorithms. Its simplicity allows practitioners to adopt and extend it with minimal overhead. We hope V-Simba serves as an architectural foundation to accelerate progress in the robotics community.

References

[1] Philip J Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient online reinforcement learning with offline data. In *International Conference on Machine Learning*, pages 1577–1594. PMLR, 2023. (Cited on page 3)

[2] Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *International conference on machine learning*, pages 449–458. PMLR, 2017. (Cited on page 5)

[3] Richard Bellman. A markovian decision process. *Journal of mathematics and mechanics*, pages 679–684, 1957. (Cited on page 3)

[4] Aditya Bhatt, Daniel Palenicek, Boris Belousov, Max Argus, Artemij Amiranashvili, Thomas Brox, and Jan Peters. Crossq: Batch normalization in deep reinforcement learning for greater sample efficiency and simplicity. In *The Twelfth International Conference on Learning Representations*, 2024. (Cited on page 2)

[5] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018. (Cited on page 2)

[6] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018. (Cited on page 3)

[7] Edoardo Cetin, Philip J Ball, Steve Roberts, and Oya Celiktutan. Stabilizing off-policy deep reinforcement learning from pixels. *arXiv preprint arXiv:2207.00986*, 2022. (Cited on page 2, 6, 20)

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. (Cited on page 5)

[9] Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International conference on machine learning*, pages 1407–1416. PMLR, 2018. (Cited on page 3, 17)

[10] Kevin Esslinger, Robert Platt, and Christopher Amato. Deep transformer q-networks for partially observable reinforcement learning. *arXiv preprint arXiv:2206.01078*, 2022. (Cited on page 3)

[11] Jesse Farebrother, Jordi Orbay, Quan Vuong, Adrien Ali Taïga, Yevgen Chebotar, Ted Xiao, Alex Irpan, Sergey Levine, Pablo Samuel Castro, Aleksandra Faust, et al. Stop regressing: Training value functions via classification for scalable deep rl. *arXiv preprint arXiv:2403.03950*, 2024. (Cited on page 5)

[12] Chelsea Finn, Xin Yu Tan, Yan Duan, Trevor Darrell, Sergey Levine, and Pieter Abbeel. Deep spatial autoencoders for visuomotor learning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 512–519. IEEE, 2016. (Cited on page 3)

[13] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020. (Cited on page 2, 17)

[14] Scott Fujimoto, Pierluca D'Oro, Amy Zhang, Yuandong Tian, and Michael Rabbat. Towards general-purpose model-free reinforcement learning. *arXiv preprint arXiv:2501.16142*, 2025. (Cited on page 2, 3, 6, 17, 20)

[15] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018. (Cited on page 17)

[16] Scott Fujimoto, David Meger, and Doina Precup. A deep reinforcement learning approach to marginalized importance sampling with the successor representation. In *International Conference on Machine Learning*, pages 3518–3529. PMLR, 2021. (Cited on page 3)

[17] Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G Bellemare. Deepmdp: Learning continuous latent space models for representation learning. In *International conference on machine learning*, pages 2170–2179. PMLR, 2019. (Cited on page 3)

[18] Gene H Golub and Charles F Van Loan. Lanczos methods. *Matrix Computations. Baltimore: Johns Hopkins University Press*, pages 470–507, 1996. (Cited on page 17)

[19] Zhaohan Guo, Shantanu Thakoor, Miruna Pîslar, Bernardo Avila Pires, Florent Altché, Corentin Tallec, Alaa Saade, Daniele Calandriello, Jean-Bastien Grill, Yunhao Tang, et al. Byol-explore: Exploration by bootstrapped prediction. *Advances in neural information processing systems*, 35:31855–31870, 2022. (Cited on page 3)

[20] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018. (Cited on page 3)

[21] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018. (Cited on page 2, 3, 17)

[22] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023. (Cited on page 2, 3, 20)

[23] Nicklas Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. *arXiv preprint arXiv:2310.16828*, 2023. (Cited on page 2, 3, 6, 19, 20)

[24] Fengxiang He, Tongliang Liu, and Dacheng Tao. Why resnet works? residuals generalize. *IEEE transactions on neural networks and learning systems*, 31(12):5349–5362, 2020. (Cited on page 4)

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. (Cited on page 5)

[26] Binh-Son Hua, Minh-Khoi Tran, and Sai-Kit Yeung. Pointwise convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 984–993, 2018. (Cited on page 2, 5)

[27] Suning Huang, Zheyu Zhang, Tianhai Liang, Yihan Xu, Zhehao Kou, Chenhao Lu, Guowei Xu, Zhengrong Xue, and Huazhe Xu. Mentor: Mixture-of-experts network with task-oriented perturbation for visual reinforcement learning. *Proc. the International Conference on Machine Learning (ICML)*, 2025. (Cited on page 3)

[28] Marcel Hussing, Claas A Voelcker, Igor Gilitschenski, Amir-massoud Farahmand, and Eric Eaton. Dissecting deep rl with high update ratios: Combatting value divergence. In *Reinforcement Learning Conference*, 2024. (Cited on page 3)

[29] Kyungsoo Kim, Jeongsoo Ha, and Yusung Kim. Self-predictive dynamics for generalization of vision-based reinforcement learning. In *IJCAI*, pages 3150–3156, 2022. (Cited on page 3)

[30] Ilya Kostrikov, Denis Yarats, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*, 2020. (Cited on page 3)

[31] Manoj Kumar, Mostafa Dehghani, and Neil Houlsby. Dual patchnorm. *arXiv preprint arXiv:2302.01327*, 2023. (Cited on page 4)

[32] Misha Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. *Advances in neural information processing systems*, 33:19884–19895, 2020. (Cited on page 2, 3)

[33] Alex X Lee, Anusha Nagabandi, Pieter Abbeel, and Sergey Levine. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. *Advances in Neural Information Processing Systems*, 33:741–752, 2020. (Cited on page 3)

[34] Hojoon Lee, Hanseul Cho, Hyunseung Kim, Daehoon Gwak, Joonkee Kim, Jaegul Choo, Se-Young Yun, and Chulhee Yun. Plastic: Improving input and label plasticity for sample efficient reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024. (Cited on page 2, 17)

[35] Hojoon Lee, Dongyoon Hwang, Donghu Kim, Hyunseung Kim, Jun Jet Tai, Kaushik Subramanian, Peter R Wurman, Jaegul Choo, Peter Stone, and Takuma Seno. Simba: Simplicity bias for scaling up parameters in deep reinforcement learning. *arXiv preprint arXiv:2410.09754*, 2024. (Cited on page 2, 3, 6, 9, 17)

[36] Hojoon Lee, Youngdo Lee, Takuma Seno, Donghu Kim, Peter Stone, and Jaegul Choo. Hyperspherical normalization for scalable deep reinforcement learning. *arXiv preprint arXiv:2502.15280*, 2025. (Cited on page 2, 3, 4, 5, 17)

[37] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *ArXiv e-prints*, pages arXiv–1607, 2016. (Cited on page 2, 4)

[38] TP Lillicrap. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015. (Cited on page 2, 3, 17)

[39] Haotian Lin, Pengcheng Wang, Jeff Schneider, and Guanya Shi. Td-m (pc) ^{2}: Improving temporal difference mpc through policy constraint. *arXiv preprint arXiv:2502.03550*, 2025. (Cited on page 3)

[40] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022. (Cited on page 5, 6)

[41] Clare Lyle, Zeyu Zheng, Khimya Khetarpal, James Martens, Hado van Hasselt, Razvan Pascanu, and Will Dabney. Normalization and effective learning rates in reinforcement learning. *arXiv preprint arXiv:2407.01800*, 2024. (Cited on page 3, 4)

[42] Clare Lyle, Zeyu Zheng, Evgenii Nikishin, Bernardo Avila Pires, Razvan Pascanu, and Will Dabney. Understanding plasticity in neural networks. *Proc. the International Conference on Machine Learning (ICML)*, 2023. (Cited on page 3)

[43] Guozheng Ma, Zhen Wang, Zhecheng Yuan, Xueqian Wang, Bo Yuan, and Dacheng Tao. A comprehensive survey of data augmentation in visual reinforcement learning. *arXiv preprint arXiv:2210.04561*, 2022. (Cited on page 3)

[44] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021. (Cited on page 9)

[45] Trevor McInroe, Lukas Schäfer, and Stefano V Albrecht. Learning temporally-consistent representations for data-efficient reinforcement learning. *arXiv preprint arXiv:2110.04935*, 2021. (Cited on page 3)

[46] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015. (Cited on page 3)

[47] Michal Nauman, Mateusz Ostaszewski, Krzysztof Jankowski, Piotr Miłoś, and Marek Cygan. Bigger, regularized, optimistic: scaling for compute and sample-efficient continuous control. *arXiv preprint arXiv:2405.16158*, 2024. (Cited on page 3)

[48] Tianwei Ni, Benjamin Eysenbach, Erfan Seyedsalehi, Michel Ma, Clement Gehring, Aditya Mahajan, and Pierre-Luc Bacon. Bridging state and history representations: Understanding self-predictive rl. *arXiv preprint arXiv:2401.08898*, 2024. (Cited on page 3)

[49] Johan Obando-Ceron, Ghada Sokar, Timon Willi, Clare Lyle, Jesse Farebrother, Jakob Foerster, Gintare Karolina Dziugaite, Doina Precup, and Pablo Samuel Castro. Mixtures of experts unlock parameter scaling for deep rl. *arXiv preprint arXiv:2402.08609*, 2024. (Cited on page 3)

[50] Daniel Palenicek, Florian Vogt, and Jan Peters. Scaling off-policy reinforcement learning with batch and weight normalization. *arXiv preprint arXiv:2502.07523*, 2025. (Cited on page 2, 4, 17)

[51] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pages 2778–2787. PMLR, 2017. (Cited on page 3)

[52] 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. *arXiv preprint arXiv:1709.10087*, 2017. (Cited on page 2, 6, 8, 21)

[53] Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. *arXiv preprint arXiv:2007.05929*, 2020. (Cited on page 3)

[54] Max Schwarzer, Johan Samir Obando Ceron, Aaron Courville, Marc G Bellemare, Rishabh Agarwal, and Pablo Samuel Castro. Bigger, better, faster: Human-level atari with human-level efficiency. In *International Conference on Machine Learning*, pages 30365–30380. PMLR, 2023. (Cited on page 3)

[55] Max Schwarzer, Nitarshan Rajkumar, Michael Noukhovitch, Ankesh Anand, Laurent Charlin, R Devon Hjelm, Philip Bachman, and Aaron C Courville. Pretraining representations for data-efficient reinforcement learning. *Advances in Neural Information Processing Systems*, 34:12686–12699, 2021. (Cited on page 3)

[56] Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In *International conference on machine learning*, pages 8583–8592. PMLR, 2020. (Cited on page 3)

[57] Younggyo Seo, Kimin Lee, Stephen L James, and Pieter Abbeel. Reinforcement learning with action-free pre-training from videos. In *International Conference on Machine Learning*, pages 19561–19579. PMLR, 2022. (Cited on page 3)

[58] Ghada Sokar, Rishabh Agarwal, Pablo Samuel Castro, and Utku Evci. The dormant neuron phenomenon in deep reinforcement learning. *arXiv preprint arXiv:2302.12902*, 2023. (Cited on page 2)

[59] Ghada Sokar, Rishabh Agarwal, Pablo Samuel Castro, and Utku Evci. The dormant neuron phenomenon in deep reinforcement learning. In *International Conference on Machine Learning*, pages 32145–32168. PMLR, 2023. (Cited on page 6, 17)

[60] Ghada Sokar, Johan Obando-Ceron, Aaron Courville, Hugo Larochelle, and Pablo Samuel Castro. Don't flatten, tokenize! unlocking the key to softmoe's efficacy in deep rl. *arXiv preprint arXiv:2410.01930*, 2024. (Cited on page 3)

[61] Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling representation learning from reinforcement learning. In *International conference on machine learning*, pages 9870–9879. PMLR, 2021. (Cited on page 3)

[62] Bhavya Sukhija, Stelian Coros, Andreas Krause, Pieter Abbeel, and Carmelo Sferrazza. Maxinforl: Boosting exploration in reinforcement learning through information gain maximization. *arXiv preprint arXiv:2412.12098*, 2024. (Cited on page 2, 3)

[63] 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. (Cited on page 2, 6, 7, 21)

[64] Damien Teney, Armand Mihai Nicolicioiu, Valentin Hartmann, and Ehsan Abbasnejad. Neural redshift: Random networks are not random functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4786–4796, 2024. (Cited on page 9)

[65] Raphael Trumpp, Ansgar Schäfftlein, Mirco Theile, and Marco Caccamo. Impoola: The power of average pooling for image-based deep reinforcement learning. *arXiv preprint arXiv:2503.05546*, 2025. (Cited on page 3)

[66] Herke Van Hoof, Nutan Chen, Maximilian Karl, Patrick Van Der Smagt, and Jan Peters. Stable reinforcement learning with autoencoders for tactile and visual data. In *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 3928–3934. IEEE, 2016. (Cited on page 3)

[67] 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. *arXiv preprint arXiv:2310.07220*, 2023. (Cited on page 3)

[68] Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. *Advances in neural information processing systems*, 28, 2015. (Cited on page 3)

[69] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16133–16142, 2023. (Cited on page 2, 17)

[70] Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. Daydreamer: World models for physical robot learning. In *Conference on robot learning*, pages 2226–2240. PMLR, 2023. (Cited on page 3)

[71] Guowei Xu, Ruijie Zheng, Yongyuan Liang, Xiyao Wang, Zhecheng Yuan, Tianying Ji, Yu Luo, Xiaoyu Liu, Jiaxin Yuan, Pu Hua, et al. Drm: Mastering visual reinforcement learning through dormant ratio minimization. *arXiv preprint arXiv:2310.19668*, 2023. (Cited on page 2, 4, 6, 20, 21)

[72] Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. *arXiv preprint arXiv:2107.09645*, 2021. (Cited on page 2, 3, 6, 9, 17, 20)

[73] Denis Yarats, Amy Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and Rob Fergus. Improving sample efficiency in model-free reinforcement learning from images. In *Proceedings of the aaai conference on artificial intelligence*, volume 35, pages 10674–10681, 2021. (Cited on page 2)

[74] Weirui Ye, Shaohuai Liu, Thanard Kurutach, Pieter Abbeel, and Yang Gao. Mastering atari games with limited data. *Advances in neural information processing systems*, 34:25476–25488, 2021. (Cited on page 3)

[75] Tao Yu, Cuiling Lan, Wenjun Zeng, Mingxiao Feng, Zhizheng Zhang, and Zhibo Chen. Playvirtual: Augmenting cycle-consistent virtual trajectories for reinforcement learning. *Advances in Neural Information Processing Systems*, 34:5276–5289, 2021. (Cited on page 3)

[76] Tao Yu, Zhizheng Zhang, Cuiling Lan, Yan Lu, and Zhibo Chen. Mask-based latent reconstruction for reinforcement learning. *Advances in Neural Information Processing Systems*, 35:25117–25131, 2022. (Cited on page 3)

[77] 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 *Conference on robot learning*, pages 1094–1100. PMLR, 2020. (Cited on page 2, 6, 8, 21)

[78] Kevin Zakka, Baruch Tabanpour, Qiayuan Liao, Mustafa Haiderbhai, Samuel Holt, Jing Yuan Luo, Arthur Allshire, Erik Frey, Koushil Sreenath, Lueder A Kahrs, et al. Mujoco playground. *arXiv preprint arXiv:2502.08844*, 2025. (Cited on page 9)

[79] 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. *Advances in Neural Information Processing Systems*, 36:48203–48225, 2023. (Cited on page 2, 3, 4, 6, 20)

Appendix

Table of Contents

A Extended Metric Analysis

This section provides the details and extended results of the metric analysis presented in Figure 2.

A.1 Setup

Our goal is to analyze and compare neural network architectures in visual RL in terms of learning dynamics and stability. We use two baseline algorithms—DDPG [38] and SAC [21]—both with data augmentation [72]. We then evaluate four neural architectures proposed by DrQ-v2 [72], IMPALA [9], MR.Q [14] and our proposed V-Simba on each algorithm. For IMPALA, we only employ the encoder with residual blocks, combined with DrQ-v2 predictors. For MR.Q, we exclude the dynamics learning components and learn the encoder and critic end-to-end (dubbed 'no MR' in their ablation experiments).

We follow the original paper for any architecture-specific hyperparameters such as number of layers, hidden dimension, and the use of clipped double Q-learning (CDQ) [15]. Otherwise, we use the same set of hyperparameters for all experiments. We measure the metrics (Section A.2) every 10,000 update steps (20,000 environment steps), using a mini-batch of size 256.

A.2 Metrics

We employ the following metrics for analysis:

Sharpness of the loss landscape. Sharpness is often considered indicative of a neural network's ability to generalize. In reinforcement learning, the underlying data distribution is inherently non-stationary, making consistent generalization crucial. We quantify sharpness by the largest eigenvalue of the Hessian matrix ($\lambda_{\max}(\nabla^2 \mathcal{L})$) [34, 13], which can be approximated using the Lanczos algorithm [18].

Dormant ratio. A neuron is said to be inactive or *dormant* when its absolute activation value tends to be small compared to the layer's average. Formally, the i-th neuron of layer ℓ is τ-dormant if $s_i^\ell = \frac{\mathbb{E}_{x \in D}|h_i^\ell(x)|}{\frac{1}{H^\ell}\sum_{k \in h}\mathbb{E}_{x \in D}|h_i^\ell(x)|} \leq \tau$, where h_i^ℓ are the activation values of layer ℓ [59]. A high proportion of dormant neurons implies that the network's decisions rely heavily on only a few neurons, indicating capacity loss. We use $\tau = 0.1$ in our analysis.

Feature diversity. While maximizing feature diversity itself might not be crucial for RL, preventing feature collapse is critical, as it reduces the network's capacity and hampers learning capability. Inspired by ConvNext-v2 [69], we measure the average cosine distance between the samples within a batch: $\frac{1}{B^2}\sum_i^B \sum_j^B \frac{1-cos(X_i, X_j)}{2}$, where B is the batch size, and $X \in \mathbb{R}^{B \times D}$ is the feature matrix.

Norms. Prior works in state-based RL have shown that controlling the growth of features, weights and gradient norms can stabilize the learning process and thus performance [35, 36, 50]. We investigate whether the same argument could be made for visual RL as well. Following [36], we define the *effective* norm of a set of vector and matrix using dimension-based weights $w_i(z) = \frac{\dim(z_i)}{\sum_{j=1}^N \dim(z_j)}$, which captures dimensional contributions across vectors and matrices. For example, for a neural network's parameter set $\theta = \{\theta_i\}_{i=1}^N$, the effective parameter norm is defined as $\|\theta\|_{\text{eff}}^2 \triangleq \sum_{i=1}^N w_i(\theta)\|\theta_i\|_2$ where $\|\cdot\|_2^2$ denotes the standard ℓ_2-norm (or Frobenius norm $\|\cdot\|_F$ for matrices).

A.3 Results

We visualize the results for DDPG in Figure 9, and SAC in Figure 10. DrQ-v2 and IMPALA architecture exhibit significant instability across both algorithms, showing high sharpness and dormant ratio, low feature diversity, and exploding feature, parameter and gradient norms. Collectively, these issues hinder the learning process and their capacity to learn meaningful behaviors.

Meanwhile, MR.Q maintains better stability by incorporating numerous normalization layers into its design. Notably, MR.Q achieves lower dormant ratios and higher feature diversity compared to V-Simba, highlighting the importance of normalization layers in stabilizing learning dynamics. Despite the strengths, MR.Q still experiences relatively high sharpness and norm magnitudes, although their growth is better controlled.

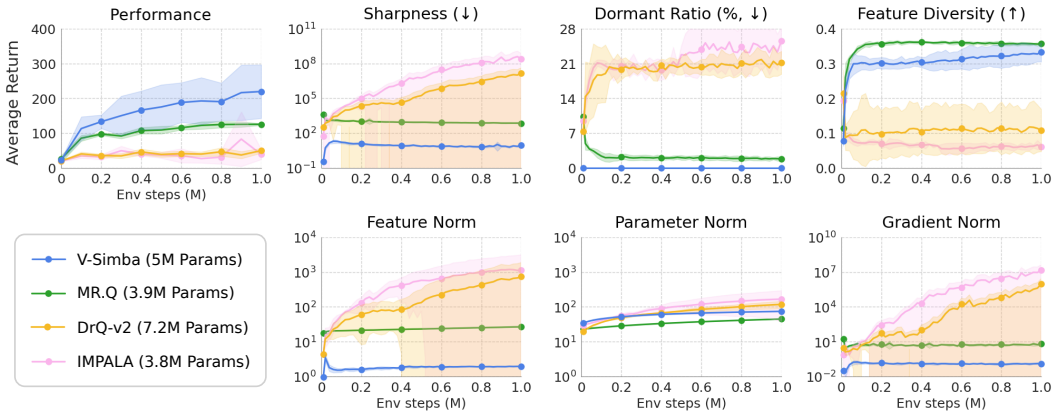


Figure 9: **Comparison of Neural Architectures under DDPG.** We evaluate and compare the neural network architectures proposed by DrQ-v2, IMPALA, MR.Q and our V-Simba, using DDPG with data augmentation in the `Dog Stand` environment. V-Simba maintains greater stability throughout training and outperforms other baselines.

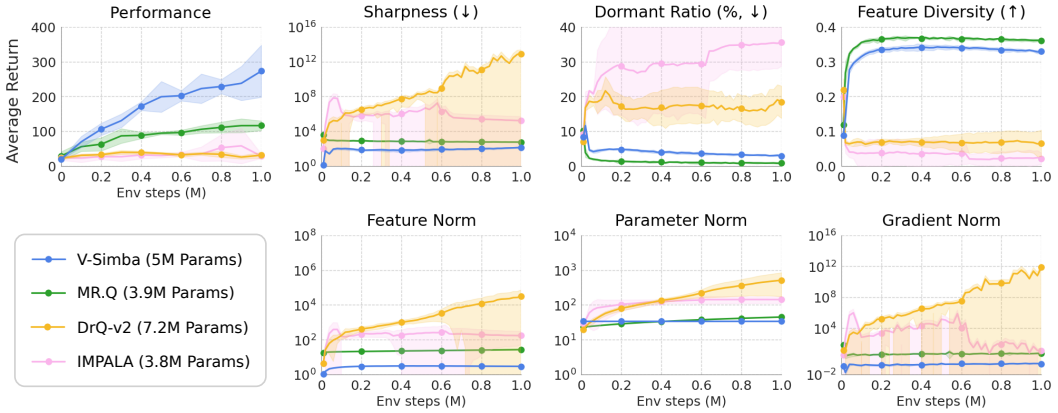


Figure 10: **Comparison of Neural Architectures under SAC.** We evaluate and compare the neural network architectures proposed by DrQ-v2, IMPALA, MR.Q and our V-Simba, using SAC with data augmentation in the `Dog Stand` environment. V-Simba maintains greater stability throughout training and outperforms other baselines.

Finally, our proposed V-Simba further enhances stability by rigorously controlling the norm scales. This leads to smoother loss landscape, low dormant ratio, high feature diversity, and overall superior performance compared to all other evaluated methods.

B Broader Impacts

Our proposed V-Simba architecture contributes to the broader reinforcement learning (RL) community by serving as addition evidence that a principled architectural design can significantly enhance sample efficiency: particularly in a visually complex continuous control tasks. The improved efficiency of V-Simba may facilitate the deployment of RL methods in real-world robotic systems, thereby reducing associated data collection costs and computational resource demands. However, this advancement also carries potential risks, including increased automation in sensitive areas, such as autonomous robotics or surveillance systems, where unintended behaviors might pose safety and ethical concerns. Moreover, our architecture's simplicity and computational efficiency may lower entry barriers, democratizing access to advanced visual RL methods but simultaneously increasing the responsibility to ensure proper safeguards, rigorous testing, and ethical considerations are incorporated into their deployment.

C Hyperparameters

Table 2 lists the hyperparameters consistently used across all experiments presented in this paper.

Table 2: **Hyperparameters Table.** We use the consistent hyperparameters across all benchmarks, which are listed below. The discount factor γ is automatically determined by heuristics from [23].

	Hyperparameter	**Notation**	**Value**
Common	Discount factor	γ	Heuristic [23]
	Replay buffer capacity	-	1M
	Buffer sampling	-	Uniform
	Batch size	-	256
	Observation Shape	$\lvert\mathcal{O}\rvert$	$3 \times 84 \times 84$
	Update-to-data (UTD) ratio	-	1
	TD steps	n	3
Encoder	Number of blocks	L	2
	Hidden dimension (channels)	d_h	32
Predictor - Actor	Number of blocks	L	1
	Hidden dimension	d_h	128
	Initial temperature	α_0	$1\mathrm{e}{-2}$
	Target entropy	$\mathbb{H}^*$	$\lvert\mathcal{A}\rvert/2$
Predictor - Critic	Number of blocks	L	2
	Hidden dimension	d_h	512
	Number of atoms	n_{atoms}	101
	Target critic momentum	τ	$5\mathrm{e}{-3}$
	Clipped double Q	-	No
	Action embedding dimension	d_a	128
Output	Number of return bins	n_{atoms}	101
	Support of return	$[G_{\min}, G_{\max}]$	$[-5, 5]$
	Reward scaler epsilon	ϵ	$1\mathrm{e}{-8}$
Optimizer	Optimizer	-	Adam
	Optimizer momentum	(β_1, β_2)	(0.9, 0.999)
	Weight Decay	-	$1\mathrm{e}{-2}$
	Learning rate	η	$1\mathrm{e}{-4}$

D Compute Resources

We mainly use RTX3090 GPUs in our experiments, which takes approximately 4.8 hours to finish a single seed experiment with V-Simba. We have optimized the replay buffer to be more memory-friendly, requiring around 11GB of RAM memory for each experiment.

E Baselines

DrQ-v2 [72]. Data-regularized Q-learning (DrQ-v2) incorporates data augmentation via random shift transformations into DDPG to avoid overfitting of visual encoder to specific visual patterns. We provide a detailed explanation of the algorithm in Section 3.2. Results for DMC Medium and Meta-World were obtained from [72] and [71], respectively. We run the official repository (`https://github.com/facebookresearch/drqv2`) over 3 random seeds for Adroit, and 5 random seeds for DMC Hard results.

A-LIX [7]. Adaptive Local Signal Mixing (A-LIX) modifies the convolutional layer of DrQ-v2 by performing bilinear interpolation with weights derived from random shifts, regularizing gradients and reducing overfitting. Meta-World results are from [71], which are averaged over 4 random seeds. We obtained Adroit results by running the official repository (`https://github.com/Aladoro/Stabilizing-Off-Policy-RL`) over 3 seeds.

TACO [79]. Temporal Action-driven Contrastive Learning (TACO) jointly learns state and action representations introducing contrastive learning to DrQ-v2, which promotes to generalize its knowledge more effectively across diverse state-action pairs, enhancing the sample efficiency of RL algorithms. Meta-World results are from [71], which are averaged over 4 random seeds. We obtained Adroit results by running the official repository (`https://github.com/FrankZheng2022/TACO`) over 3 seeds.

DrM [71]. Dormant Ratio Minimization (DrM) extends DrQ-v2 with three mechanisms that reduce the agent's dormant ratio and leverage it to balance exploration and exploitation. Meta-World results are from [71], which are averaged over 4 random seeds. We obtained Adroit results by running the official repository (`https://github.com/XuGW-Kevin/DrM`) over 3 seeds.

DreamerV3 [22]. DreamerV3 builds a latent world model by encoding the observation into a compact latent space for long-horizon behavior and value learning. DMC results are from [14], reproduced with the official codebase (`https://github.com/danijar/dreamerv3`) over 10 seeds.

TD-MPC2 [23]. TD-MPC2 learns a decoder-free world model via multi-task dynamics prediction and performs latent-space planning. DMC results are from [14], reproduced with the official codebase (`https://github.com/nicklashansen/tdmpc2`) over 10 seeds.

MR.Q [14]. Model-based Representations for Q-learning (MR.Q) is a model-free algorithm that leverages model-based auxiliary tasks to learn rich actor-critic representations. DMC results are from [14], averaged over 10 seeds.

F Environment Details

This section describes the benchmark environments used in our evaluation. A complete list of tasks, including state and action dimensions, is provided at the end of the section. Although state vectors are not used during training, we report them to reflect task difficulty. Table 3 details episode length, frame stack, action repeat, total environment steps, and performance metrics. Figure 4 shows visualizations of each environment.

F.1 DeepMind Control Suite

DeepMind Control Suite [63, DMC] is a standard benchmark for continuous control benchmarks with varying levels of complexity. Tasks in this benchmark range from simple low-dimensional ($\mathcal{S} \in \mathbb{R}^3$, $\mathcal{A} \in \mathbb{R}^1$) to highly complex continuous control ($\mathcal{S} \in \mathbb{R}^{223}$, $\mathcal{A} \in \mathbb{R}^{38}$). We evaluate 18 tasks, grouped into DMC Medium and DMC Hard. DMC Easy tasks are excluded due to their low difficulty. Full task lists appear in Tables 4 and 5.

F.2 Adroit

Adroit [52] comprises dexterous manipulation tasks involving in-hand manipulation, tool use, and articulated object control, performed using a 24 degree-of-freedom (DoF) anthropomorphic Shadow Hand. Due to the tasks' complexity, the state-of-the-art method DrM [71] uses a privileged robot sensor vector alongside image observations (see official code: `https://github.com/XuGW-Kevin/DrM`). In contrast, we do not use privileged information. To increase difficulty, we also benchmark under sparse reward settings. Full task list is provided in Table 6.

F.3 Meta-World

Meta-World [77] consists of 50 diverse robotic manipulation tasks using a simulated 7-DoF Sawyer arm in a tabletop setting. Following [71], we select 8 tasks spanning object interaction, tool use, and precise motion control to cover a range of manipulation challenges. For the easier half, we replace the ground-truth dense reward function with a binary task completion signal (i.e. a *sparse task completion reward*) and mark them as `sparse`. For details on the success metric, we refer the reader to [77]. Full task list is provided in Table 7.

Table 3: **Environment details.** We list the episode length, frame stack, action repeat for each domain, total environment steps, and performance metrics used for benchmarking.

	DMC	**Adroit**	**Meta-World**
Frame stack		3	
Action repeat		2	
Episode length	$1,000$	100-200	500
Total env. steps		1M	
Performance metric	Average Return	Average Success	Average Success

Table 4: **DMC Medium Task List.** We evaluate 11 tasks from DMC Medium benchmark, listed below. Performance for each task is reported at 1M environment steps.

Task	State dim $\lvert\mathcal{S}\rvert$	Action dim $\lvert\mathcal{A}\rvert$
acrobot-swingup	6	1
cartpole-swingup-sparse	5	1
cheetah-run	17	6
finger-turn-easy	12	2
finger-turn-hard	12	2
hopper-hop	15	4
quadruped-run	78	12
quadruped-walk	78	12
reacher-easy	6	2
reacher-hard	6	2
walker-run	24	6

Table 5: **DMC Hard Task List.** We evaluate 7 tasks from DMC Hard benchmark, listed below. Performance for each task is reported at 1M environment steps.

Task	State dim $\lvert\mathcal{S}\rvert$	Action dim $\lvert\mathcal{A}\rvert$
dog-run	223	38
dog-trot	223	38
dog-stand	223	38
dog-walk	223	38
humanoid-run	67	24
humanoid-stand	67	24
humanoid-walk	67	24

Table 6: **Adroit Task List.** We evaluate 3 tasks from Adroit benchmark, listed below. Performance for each task is reported at 1M environment steps.

Task	State dim $\lvert\mathcal{S}\rvert$	Action dim $\lvert\mathcal{A}\rvert$
door-v0-sparse	39	28
hammer-v0-sparse	46	26
pen-v0-sparse	45	24

Table 7: **Meta-World Task List.** We evaluate 8 tasks from Meta-World benchmark, listed below. Performance for each task is reported at 1M environment steps.

Task	State dim $\lvert\mathcal{S}\rvert$	Action dim $\lvert\mathcal{A}\rvert$
assembly	39	4
disassemble	39	4
pick-place-wall	39	4
stick-pull	39	4
coffee-push-sparse	39	4
hammer-sparse	39	4
soccer-sparse	39	4
sweep-into-sparse	39	4