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

Anonymous authors
Paper under double-blind review

Keywords: Deep RL, Visual RL, Neural Network Design, Plasticity, Normalization

Summary

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 (SAC) 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.

Contribution(s)

1. We identify and analyze severe training instabilities—such as sharp loss landscapes, dormant units, and feature collapse—that are inherent in standard convolutional architectures widely adopted in visual reinforcement learning.
   Context: While previous visual RL research has primarily focused on algorithmic innovations , the underlying neural architectures have largely remained simple, often adopting the standard DrQ-v2 baseline. Our analysis reveals the vulnerabilities and capacity loss associated with these heavily utilized architectures.
2. We propose V-Simba, a novel visual RL architecture that integrates normalization layers, weight regularization, and a distributional critic to stabilize training, alongside large-stride and pointwise convolutions to maintain computational efficiency.
   Context: Inspired by the recent Simba architectures in state-based RL that use principled architectural guidelines to constrain feature and parameter growth , we adapt these concepts to provide a suitable inductive bias specifically tailored for visual domains.
3. We experimentally demonstrate that V-Simba matches or outperforms state-of-the-art visual RL methods across 29 tasks spanning the DeepMind Control Suite, Adroit, and Meta-World benchmarks.
   Context: We evaluate V-Simba against strong algorithmic baselines including DrQ-v2, MR.Q, TD-MPC2, DrM, and TACO , showing that robust architectural design alone can achieve superior sample and compute efficiency without introducing complex algorithmic add-ons.

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

Anonymous authors
Paper under double-blind review

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 (SAC) 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.

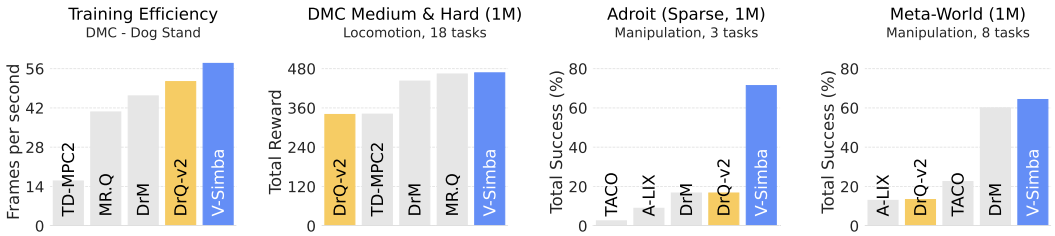


Figure 1: **Benchmark Summary.** We evaluate the effectiveness of V-Simba across 29 visual continuous control tasks spanning multiple domains, with a *single* set of hyperparameters. By incorporating V-Simba into Soft Actor-Critic with data augmentation, it matches or outperforms state-of-the-art visual 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 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 (Yarats et al., 2021b), latent dynamics modeling (Fujimoto et al., 2025; Zheng et al., 2023), world models (Hansen et al., 2023; Hafner et al., 2023),

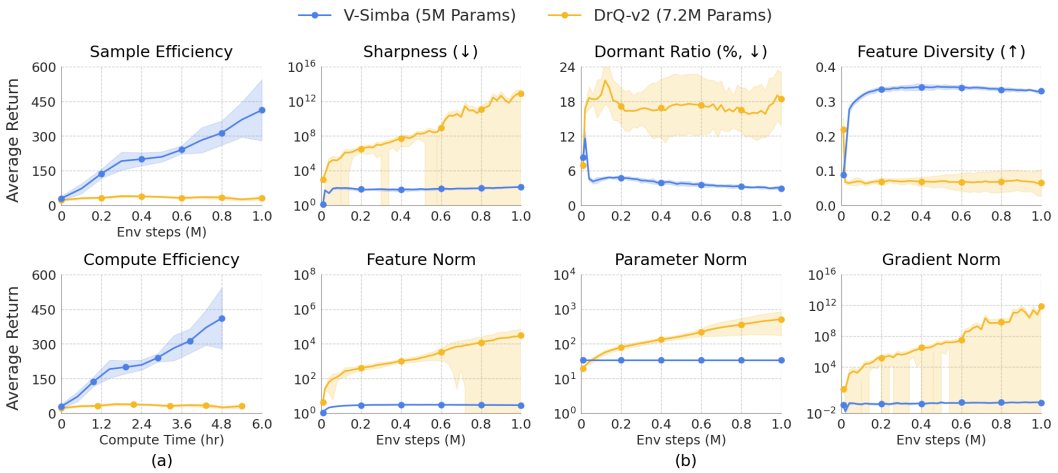


Figure 2: **DrQ-v2 v.s. V-Simba.** Comparison of the DrQ-v2 architecture and our V-Simba in the `Dog Stand` environment. Both are evaluated on Soft Actor-Critic (SAC) with results averaged over 5 seeds. **(a)** V-Simba is substantially more efficient than DrQ-v2 in terms of both sample and compute. **(b)** 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 7.2.

and improved exploration strategies (Burda et al., 2018b; Xu et al., 2023). Yet, despite these advances, the underlying neural architectures have remained relatively simple. A prominent example is DrQ-v2 (Yarats et al., 2021a), which combines the DDPG algorithm (Lillicrap, 2015) with data augmentation (Laskin et al., 2020). Its architecture consists of a shallow convolutional encoder, followed by a large fully connected layer and a single layer normalization layer (Lei Ba et al., 2016) 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 (Xu et al., 2023; Zheng et al., 2023; Cetin et al., 2022; Sukhija et al., 2024) 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 (Foret et al., 2020; Lee et al., 2024a), a high fraction of dormant units representing plasticity loss (Sokar et al., 2023a), and low feature diversity indicating feature collapse (Woo et al., 2023). In contrast, recent advances in state-based RL (Lee et al., 2024b; Bhatt et al., 2024; Lee et al., 2025; Palenicek et al., 2025) demonstrate that carefully designed architectures can effectively mitigate these instabilities. Notably, the Simba series of architectures (Lee et al., 2024b; 2025) introduce 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. In response, 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) (Haarnoja et al., 2018), 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 (Hua et al., 2018) and max-pooling operations (Krizhevsky et al., 2012).

We evaluate V-Simba on three standard benchmarks: DMControl (Tassa et al., 2018), Adroit (Rajeswaran et al., 2017), and Metaworld (Yu et al., 2020) 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 (Fujimoto et al., 2025) and TD-MPC2 (Hansen et al., 2023) in DMControl and outperforming DrM (Xu et al., 2023) and TACO (Zheng et al., 2023) 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 (Ma et al., 2022).

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 (Stooke et al., 2021; Zheng et al., 2023; Schwarzer et al., 2020; 2021; Kim et al., 2022; Lee et al., 2020; Van Hoof et al., 2016; Yu et al., 2022; Gelada et al., 2019; Seo et al., 2022; Ni et al., 2024; Yu et al., 2021; Fujimoto et al., 2021; McInroe et al., 2021; Fujimoto et al., 2025) or separate dynamics models in model-based methods (Hansen et al., 2023; Hafner et al., 2023; Lin et al., 2025; Wu et al., 2023; Ha & Schmidhuber, 2018; Finn et al., 2016; Watter et al., 2015); (2) data augmentation, especially random shifts (Laskin et al., 2020; Kostrikov et al., 2020; Yarats et al., 2021a), for better efficiency and generalization; and (3) exploration methods, including planning (Sekar et al., 2020; Wang et al., 2023), curiosity-driven (Pathak et al., 2017; Burda et al., 2018a; Guo et al., 2022), and information maximization (Sukhija et al., 2024). 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 (Mnih et al., 2015) over a decade ago. While some works have incorporated architectural elements from computer vision—such as ResNet-like architectures in Impala (Espeholt et al., 2018), BBF (Schwarzer et al., 2023), and EfficientZero (Ye et al., 2021), or transformers in DTQN (Esslinger et al., 2022)—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 (Lyle et al., 2023; Ball et al., 2023; Lyle et al., 2024), 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 (Trumpp et al., 2025) or using Mixture-of-Experts (Obando-Ceron et al., 2024; Sokar et al., 2024), no substantial architectural innovations have been sufficiently explored in visual RL (Espeholt et al., 2018; Huang et al., 2025).

3 Preliminary

As a preliminary, we briefly describe the problem setup of visual RL, DrQ-v2 architecture (Yarats et al., 2021a), and the Soft Actor-Critic algorithm (Haarnoja et al., 2018), 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) (Bellman, 1957), defined by the tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$ of state space $\mathcal{S}$, action space $\mathcal{A}$, transition function $P : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{P}(\mathcal{S})$, reward function $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ 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) (Bellman, 1957) 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) (Yarats et al., 2021a) 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 (Lillicrap, 2015) 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 (Lei Ba et al., 2016) in between.

While DrQ-v2 has become the de facto architecture for many recent visual RL algorithms (Xu et al., 2023; Zheng et al., 2023), 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) (Lei Ba et al., 2016) and residual connections (He et al., 2020) 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 (Lyle et al., 2024; Palenicek et al., 2025). Moreover, uneven growth across layers causes inconsistent gradient scales, destabilizing optimization (Lee et al., 2025). 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 (Kumar et al., 2023) which has been empirically shown to stabilize the gradients of embedding layer [1]. For preventing parameter growth, we surprisingly found a simple ℓ_2 weight regularization to be sufficient, as shown in Figure 2.

We further stabilize gradients by employing a distributional critic with KL divergence loss (Bellemare et al., 2017) and reward normalization (Lee et al., 2025). The KL divergence loss is more robust to noisy targets than mean squared error due to its smoother loss landscape (Farebrother et al., 2024), 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 (He et al., 2016), ConvNeXt (Liu et al., 2022), and Vision Transformer (Dosovitskiy et al., 2020). 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 convolutional layers into more cost-effective alternatives. For spatial convolutions, we instead utilize pointwise (1×1 kernel) convolutions (Hua et al., 2018), which operate channel-wise without mixing spatial information, preserving spatial details at a lower cost. When downsampling, instead of 2×2 kernel convolutional layers, which induces significantly less overhead in both training and inference. Empirically, we found that these alterations does not meaningfully alter the learning dynamics and the learning curves.

[1] In practice, we adopt the shift-and-norm strategy introduced in SimbaV2 (Lee et al., 2025) to preserve magnitude information. We use ℓ_2-norm for action inputs however, as when $|\mathcal{A}| = 1$ shift-and-LN always outputs $[-1, 1]$.

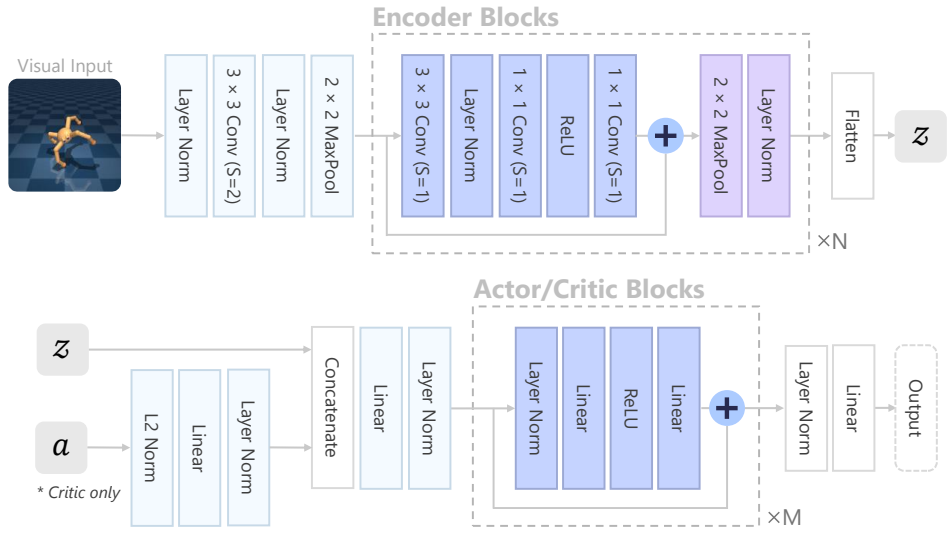


Figure 3: **V-Simba architecture.** We aim to develop an architecture that constrains its feature norm, weight norm, and gradient norm for better stability and generalization. Precisely, we make extensive use of layer normalization and residual connections for stable features and gradients, and incorporate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

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 an alternating sequence of normalization and reduction steps. Specifically, the input passes through an initial LayerNorm, followed by a 3×3 convolution (stride 2), a second LayerNorm, and finally a 2×2 max-pooling layer (stride 2). This results in downsampled and 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 to aggregate spatial features without changing resolution.
2. Feature normalization (LayerNorm) to provide stable features to the subsequent layers.
3. Two pointwise (1×1) convolutions with nonlinearities to refine and filter features, followed by a residual connection for stable gradient flow. Here, we employ an inverted bottleneck with $4\times$ expansion, following ConvNext (Liu et al., 2022) and Simba (Lee et al., 2024b),
4. Downsampling with a 2×2 maxpool layer 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 Lee et al. (2024b).

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

5 Experiments

We now provide an empirical evaluation of V-Simba:

1. **Performance Evaluation** (Sections 5.2), comparing V-Simba against leading visual RL methods to demonstrate its effectiveness across diverse benchmarks.
2. **Ablation Study** (Section 5.3), conducting experiments demonstrating the contribution of each architectural component in V-Simba.

5.1 Experimental Setup

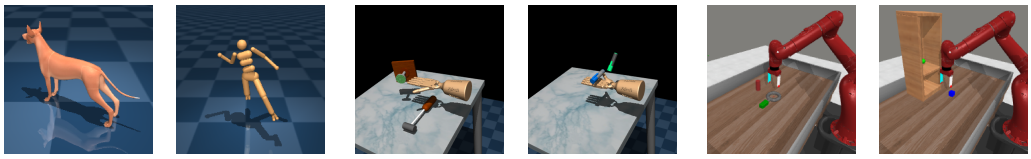
Environment. We consider a total of 29 continuous control tasks spanning 3 benchmarks: DeepMind Control (DMC) Suite (Tassa et al., 2018), Adroit (Rajeswaran et al., 2017), and Meta-World (Yu et al., 2020). Figure 4 shows the visualization of each task. These environments pose diverse challenges, including 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 (Yarats et al., 2021a), A-LIX (Cetin et al., 2022)), advanced exploration (DrM (Xu et al., 2023)), and model-based representation learning (TACO (Zheng et al., 2023), TD-MPC2 (Hansen et al., 2023), MR.Q (Fujimoto et al., 2025)). 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 (Sokar et al., 2023b)-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.

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 6, 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 5 shows that V-Simba performs competitively with MR.Q, though full task success remains elusive. We believe that this



(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 (Tassa et al., 2018), Adroit (Rajeswaran et al., 2017), and Meta-World (Yu et al., 2020).

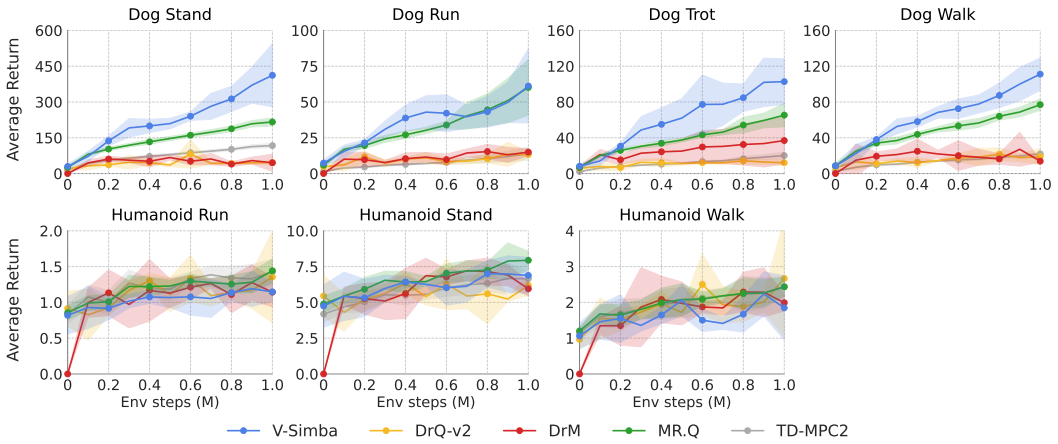


Figure 5: **DeepMind Control Suite - Hard.** Average episode returns on 7 hard-level tasks from DeepMind Control Suite (Tassa et al., 2018). Each curve represents the mean performance across 5 random seeds per algorithm; shaded areas indicate 95% bootstrap confidence intervals.

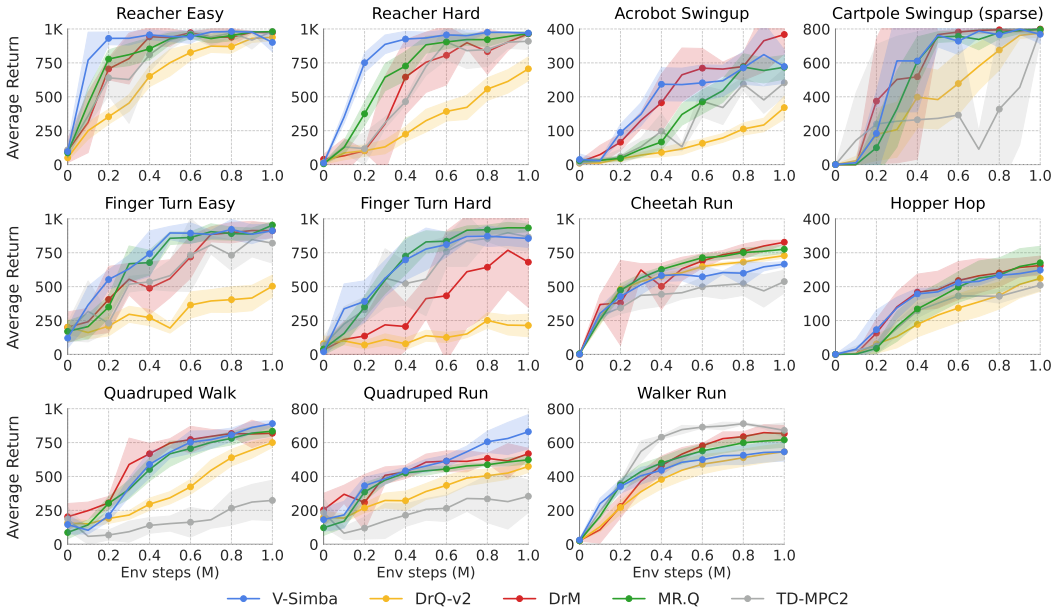


Figure 6: **DeepMind Control Suite - Medium.** Average episode returns on 11 medium-level tasks from DeepMind Control Suite (Tassa et al., 2018). Each curve represents the mean performance across 5 random seeds per algorithm; shaded areas indicate 95% bootstrap confidence intervals.

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.

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`

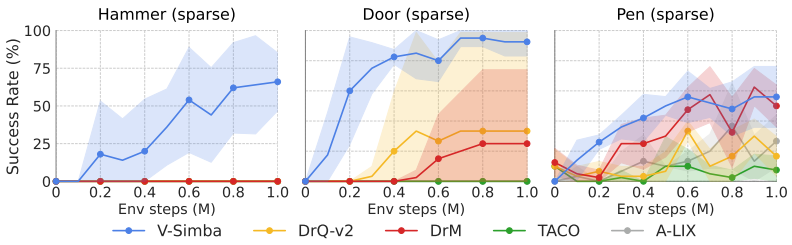


Figure 7: **Adroit - Sparse.** Average success rates on 3 sparse-reward tasks from Adroit (Rajeswaran et al., 2017). Each curve represents the mean performance across 5 random seeds per algorithm; shaded areas indicate 95% bootstrap confidence intervals.

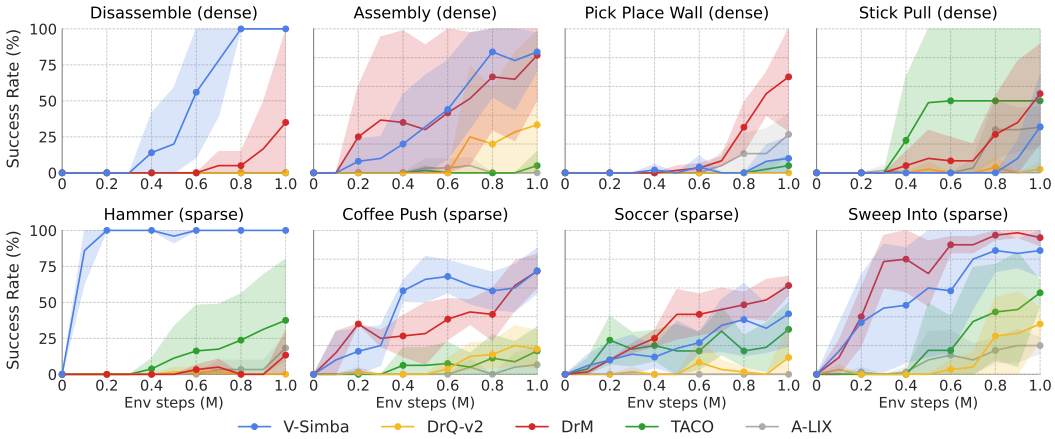


Figure 8: **Meta-World.** Average success rates on 8 tasks from the Meta-World (Yu et al., 2020). Each curve represents the mean performance across 5 random seeds per algorithm; shaded areas indicate 95% bootstrap confidence intervals.

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 Yu et al. (2020), 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.

We first investigate the effect of normalization layers (Table 1.(a)-(c)). **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

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	0.442 ± 0.095	0.694 ± 0.122	0.497 ± 0.167	0.492 ± 0.081
(b) No Input Normalization	0.401 ± 0.098	0.612 ± 0.133	0.642 ± 0.158	0.502 ± 0.083
(c) LN → LN w/o γ, β	0.453 ± 0.073	0.668 ± 0.133	0.586 ± 0.125	0.522 ± 0.063
Residual and Weight Decay				
(d) No Residual Connection	0.453 ± 0.098	0.623 ± 0.178	0.618 ± 0.154	0.526 ± 0.081
(e) No Weight Decay	0.452 ± 0.099	0.591 ± 0.144	0.513 ± 0.171	0.492 ± 0.080
Value Learning				
(f) No Categorical Critic	0.441 ± 0.096	0.591 ± 0.144	0.576 ± 0.167	0.504 ± 0.079
(g) No Reward Scaling	0.439 ± 0.094	0.683 ± 0.144	0.599 ± 0.158	0.519 ± 0.080
V-Simba	0.467 ± 0.075	0.725 ± 0.167	0.668 ± 0.115	0.549 ± 0.060

the bias and scale parameters of LayerNorm. In summary, by removing certain normalization layers or components, the network loses the control over the feature norms, leading to degradation.

Next, we quantify the importance of residual connections and weight decay (Table 1.(d)-(e)). 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 (Teney et al., 2024; Lee et al., 2024b). 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.(f)). Even with categorical loss, leaving no bounds to the reward scales led to similar consequences (Table 1.(g)), 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 (Lee et al., 2024b). 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 (Yarats et al., 2021a). 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 (Makoviychuk et al., 2021; Zakka et al., 2025), 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

Philip J Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient online reinforcement learning with offline data. In *International Conference on Machine Learning*, pp. 1577–1594. PMLR, 2023.

Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *International conference on machine learning*, pp. 449–458. PMLR, 2017.

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

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. URL https://openreview.net/forum?id=PczQtTsTIX.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018a.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018b.

Edoardo Cetin, Philip J Ball, Steve Roberts, and Oya Celiktutan. Stabilizing off-policy deep reinforcement learning from pixels. *arXiv preprint arXiv:2207.00986*, 2022.

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.

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*, pp. 1407–1416. PMLR, 2018.

Kevin Esslinger, Robert Platt, and Christopher Amato. Deep transformer q-networks for partially observable reinforcement learning. *arXiv preprint arXiv:2206.01078*, 2022.

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.

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)*, pp. 512–519. IEEE, 2016.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2018.

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*, pp. 3518–3529. PMLR, 2021.

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.

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*, pp. 2170–2179. PMLR, 2019.

Gene H Golub and Charles F Van Loan. Lanczos methods. *Matrix Computations. Baltimore: Johns Hopkins University Press*, pp. 470–507, 1996.

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.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

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*, pp. 1861–1870. PMLR, 2018.

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

Nicklas Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. *arXiv preprint arXiv:2310.16828*, 2023.

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.

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*, pp. 770–778, 2016.

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*, pp. 984–993, 2018.

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.

Kyungsoo Kim, Jeongsoo Ha, and Yusung Kim. Self-predictive dynamics for generalization of vision-based reinforcement learning. In *IJCAI*, pp. 3150–3156, 2022.

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.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Manoj Kumar, Mostafa Dehghani, and Neil Houlsby. Dual patchnorm. *arXiv preprint arXiv:2302.01327*, 2023.

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.

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.

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, 2024a.

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*, 2024b.

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.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *ArXiv e-prints*, pp. arXiv–1607, 2016.

TP Lillicrap. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

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.

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*, pp. 11976–11986, 2022.

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.

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.

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.

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.

Trevor McInroe, Lukas Schäfer, and Stefano V Albrecht. Learning temporally-consistent representations for data-efficient reinforcement learning. *arXiv preprint arXiv:2110.04935*, 2021.

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.

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.

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.

Daniel Palenicek, Florian Vogt, and Jan Peters. Scaling off-policy reinforcement learning with batch and weight normalization. *arXiv preprint arXiv:2502.07523*, 2025.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pp. 2778–2787. PMLR, 2017.

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.

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.

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.

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*, pp. 30365–30380. PMLR, 2023.

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*, pp. 8583–8592. PMLR, 2020.

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*, pp. 19561–19579. PMLR, 2022.

Ghada Sokar, Rishabh Agarwal, Pablo Samuel Castro, and Utku Evci. The dormant neuron phenomenon in deep reinforcement learning. *arXiv preprint arXiv:2302.12902*, 2023a.

Ghada Sokar, Rishabh Agarwal, Pablo Samuel Castro, and Utku Evci. The dormant neuron phenomenon in deep reinforcement learning. In *International Conference on Machine Learning*, pp. 32145–32168. PMLR, 2023b.

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.

Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling representation learning from reinforcement learning. In *International conference on machine learning*, pp. 9870–9879. PMLR, 2021.

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.

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.

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*, pp. 4786–4796, 2024.

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.

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)*, pp. 3928–3934. IEEE, 2016.

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.

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.

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*, pp. 16133–16142, 2023.

Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. Daydreamer: World models for physical robot learning. In *Conference on robot learning*, pp. 2226–2240. PMLR, 2023.

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.

Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. *arXiv preprint arXiv:2107.09645*, 2021a.

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, pp. 10674–10681, 2021b.

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.

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.

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.

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*, pp. 1094–1100. PMLR, 2020.

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.

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.