UNIVERSITY OF CALIFORNIA SAN DIEGO

Democratizing Visual Robot Learning

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Computer Science (Computer Engineering)

by

Abdulaziz Almuzairee

Committee in charge:

Professor Henrik I. Christensen, Chair
Professor Nikolay Atanasov
Professor Julian McAuley
Professor Michael Yip

2026

The Dissertation of Abdulaziz Almuzairee is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2026

DEDICATION

To my dear family and friends, you make it all worthwhile.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I am beyond grateful to my research advisor, Professor Henrik I. Christensen, for his unwavering support and continued guidance, pushing me to grow as a researcher during my PhD journey. Our meetings have always left me with renewed determination and optimism, and his advice has been invaluable in navigating my research path. I am honored for having him as my research advisor.

I would like to extend my sincere gratitude to my committee members, Professor Nikolay Atanasov, Professor Julian McAuley, and Professor Michael Yip for their invaluable time and their insightful feedback on my PhD thesis.

I would like to thank my co-authors: Nicklas Hansen, Rohan Patil, Dwait Bhatt, Shubhankar Borse, Marvin Klingner, Varun Ravi Kumar, Hong Cai, Senthil Yogamani, and Fatih Porikli. Without their support, this work would not have been possible.

I would like to thank my lab members: Jing-yan Liao, Jiaming Hu, Seth Farrell, Yang Jiao, Chenghao Li, Rohan Patil, Yiding Qiu, Julian Raheema, Luobin Wang, Zihan Zhang, Hengyuan Zhang, Anwesan Pal, Akanimoh Adeleye, Christopher D'Ambrosia, and David Paz. They made this journey lighter, brighter, and richer.

I would like to express my heartfelt thanks to my supervisors during my Qualcomm internship Steve Han and Hong Cai for their guidance and support during my internship.

I would like to thank both Professor Hao Su and Jiayuan Gu for their support and time during our joint work.

I would like to sincerely thank Professor Patrick Virtue from Carnegie Melon University for his support and advice.

Chapter 2 and Appendix A, in full, are a reprint of the material as it appears from "A Recipe for Data Augmentation in Visual Reinforcement Learning" by Abdulaziz Almuzairee, Nicklas Hansen, and Henrik I. Christensen. In Reinforcement Learning Journal, 2024. The dissertation author was the primary investigator and author of this paper.

Chapter 3 and Appendix B, in full, are a reprint of the material as it appears from

”Merging and Disentangling Views in Visual Reinforcement Learning for Robotic Manipulation” by Abdulaziz Almuzairee, Rohan Patil, Dwait Bhatt, and Henrik I. Christensen. In Conference on Robot Learning, 2025. The dissertation author was the primary investigator and author of this paper.

Chapter 4, in full, is a reprint of the material as it appears from ”Squint: Fast Visual Reinforcement Learning for Sim-to-Real Robotics” by Abdulaziz Almuzairee and Henrik I. Christensen. Under Review, 2026. The dissertation author was the primary investigator and author of this paper.

VITA

2019	Bachelor of Science in Computer Engineering, San Diego State University
2019–2021	Research Assistant, Carnegie Melon University
2020	Master of Science in Electrical and Computer Engineering, Carnegie Melon University
2022–2026	Research Scientist, Cognitive Robotics Lab, University of California San Diego
2022	Deep Learning Research Intern, Qualcomm Inc, San Diego
2025	Teaching Assistant, Department of Computer Science and Engineering University of California San Diego
2026	Doctor of Philosophy in Computer Science (Computer Engineering), University of California San Diego

PUBLICATIONS

Abdulaziz Almuzairee and Henrik I. Christensen. "Squint: Fast Visual Reinforcement Learning for Sim-to-Real Robotics." *Under Submission*, 2026.

Abdulaziz Almuzairee, Rohan Patil, Dwait Bhatt, and Henrik I. Christensen. "Merging and Disentangling Views in Visual Reinforcement Learning for Robotic Manipulation." Conference on Robot Learning. PMLR, 2025.

Abdulaziz Almuzairee, Nicklas Hansen, and Henrik I Christensen. "A Recipe for Unbounded Data Augmentation in Visual Reinforcement Learning." Reinforcement Learning Journal, vol. 1, 2024, pp. 130–157.

Shubhankar Borse, Marvin Klingner, Varun Ravi, Hong Cai, **Abdulaziz Almuzairee**, Senthil Yogamani, and Fatih Porikli. "X-Align++: cross-modal cross-view alignment for Bird's-eye-view segmentation." Machine Vision and Applications 34.4 (2023).

Shubhankar Borse, Marvin Klingner, Varun Ravi Kumar, Hong Cai, **Abdulaziz Almuzairee**, Senthil Yogamani, and Fatih Porikli. "X-align: Cross-modal cross-view alignment for bird's-eye-view segmentation." Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2023.

ABSTRACT OF THE DISSERTATION

Democratizing Visual Robot Learning

by

Abdulaziz Almuzairee

Doctor of Philosophy in Computer Science (Computer Engineering)

University of California San Diego, 2026

Professor Henrik I. Christensen, Chair

Visual reinforcement learning holds great promise for training robot controllers end-to-end from image inputs, offering capabilities that classical controllers fail to attain. However, scaling visual reinforcement learning to diverse backgrounds, setups and tasks remains nontrivial. These limitations are not due to a lack of data alone, but due to current visual reinforcement learning methods being too brittle and inefficient for widespread adoption. To further democratize visual robot learning, we tackle three core limitations of visual reinforcement learning for continuous control: poor visual generalization, poor visual robustness, and poor visual training efficiency.

We first address poor visual generalization by introducing SADA (Stabilized Actor-Critic under Data Augmentation), a training recipe that augments the agent's replay buffer with large

image datasets and visual transformations, enabling invariance to visual perturbations without destabilizing training. Second, we tackle visual robustness under multi-camera and multi-sensor setups with MAD (Merge and Disentangle), a lightweight method that maximizes training efficiency with multiple inputs while remaining robust to sensor failures at deployment. Third, we improve training efficiency with Squint, a novel off-policy visual reinforcement learning algorithm that achieves faster wall-time training than prior methods through parallel simulation, resolution squinting, a tuned updated-to-data ratio, a distributional critic, layer normalization, and an optimized implementation. Squint succeeds in training deployable visual reinforcement learning policies to our real SO-101 robot setup in minutes. Together, these three contributions lower the barrier to entry for visual robot learning, making visual reinforcement learning agents more practical and accessible for the broader research community.

Chapter 1

Introduction

Reinvention is at the heart of humanity's essence, as it marks different ages in history. The most imminent one is that of artificial intelligence and robotics. Both artificial intelligence and robotics hold the promise of reforming our lives, improving its quality, and advancing humanity. While the goal is clear, the path towards these advancements is unclear. Towards this endeavor, many research trajectories have been pursued and formulated. This thesis focuses on *training intelligent agents that learn from experience*, namely reinforcement learning (RL) [Kaelbling et al., 1996, Sutton and Barto, 2018], *to control robots from visual inputs*. Through trial and error, these agents can learn to control robots, imbuing them with unprecedented capabilities and accuracy that classical controllers fail to attain.

While earlier formulations of reinforcement learning are tabular [Watkins and Dayan, 1992], recent advances in reinforcement learning have shown it to work with deep neural networks, enabling the age of deep reinforcement learning [Mnih et al., 2013]. In Mnih et al. [2013], agents were trained to play Atari from scratch in simulation. Given input images, a neural network would be trained to output discrete actions, that maximize an action-reward function, referred to as the Q-function [Watkins and Dayan, 1992, Mnih et al., 2013, 2015], in a goal to maximize the final score in the each Atari level. While visual reinforcement learning started with Atari for discrete control, its implications soon followed into robotics for continuous control [Levine et al., 2016, Pinto and Gupta, 2016, Kalashnikov et al., 2018]. Learning robot controllers end-to-end purely

from vision allows lightweight deployment to visual tasks without the need of auxiliary models to extract task-specific representations. In return, this also allows agents to focus on extracting features that are useful for the downstream task. These two advantages hold the promise of allowing us to train superhuman agents that can control robots to excel at tasks that we define.

While this proves powerful for training an agent to control a robot on a single task, scaling this agent to allow it to generalize to different backgrounds, setups, and tasks is nontrivial. This difficulty is not due to a lack of data alone, but because current visual reinforcement learning methods are too brittle and inefficient for widespread adoption.

To *democratize visual robot learning*, the barrier of entry for this field needs to be lower than ever, and the agents need to be easily trainable, deployable and robust. Since this dissertation focuses on visual robot learning through the lens of visual reinforcement learning, we focus on tackling the major limitations of visual reinforcement learning for continuous control. Visual reinforcement learning agents suffer from (1) poor visual generalization: easily overfitting to input images, making them fail with the slightest visual perturbations (2) poor visual robustness: failing when some redundant cameras or sensors fail, even though enough information is present. (3) slow training times: taking long times to train policies on task due to the use of visual encoders.

Firstly, we *improve visual generalization of visual reinforcement learning agents* through the use of data augmentation. While overfitting stems from the limited images that an agent sees during training on a single task, we augment the data that the agent learns from by incorporating large image datasets and visual transformations. During training, reinforcement learning agents keep a replay buffer of all their experiences, from which they sample from continuously to keep updating their weights. By augmenting the data in this replay buffer, we allow the agent to see diverse possibilities of images that map to the same task specific state, thereby learning to be invariant to different visual perturbations. However, augmenting the replay buffer naively hurts training efficiency and risks divergence in training. We introduce a training recipe for applying augmentation in Chapter 2, named SADA (Stabilized Actor critic under Data Augmentation)

[Almuzairee et al., 2024], that produces visual robot agents that can generalize to diverse visual scenarios.

Secondly, we show that we can *produce robust visual reinforcement learning agents* under multi sensor and multi camera setups. When training under multi camera or multi sensor setups, reinforcement learning algorithms are dependent on all the inputs, even redundant ones. This renders them fragile whenever a single sensor malfunctions or is not available on deployment. Furthermore, a common paradigm in training is that we would like to maximize learning efficiency by using all available cameras and sensors during training, but still allowing lightweight deployment, with the minimal set of sensors or cameras needed. To achieve this, we introduce a method named MAD (Merge and Disentangle) [Almuzairee et al., 2025] in Chapter 3. MAD allows RL agents to maximize training efficiency with multiple input cameras, while still being robust enough to work with any single camera on deployment. Most importantly, MAD incorporates lightweight changes that can be applied to any reinforcement learning algorithm.

Third, we highlight how we can *speed up training times and improve training efficiency when training visual reinforcement learning agents*. The use of visual encoders, encoding and storing of input images, combined with non-stationary targets that are present in reinforcement learning frameworks, introduce many challenges for training fast reinforcement learning agents. This slows down research iteration, and makes it difficult for practitioners to adopt and build on these algorithms. Most previous visual reinforcement learning methods show that on-policy methods, namely ones that waste samples by only using training samples produced by the current policies, can be faster to train in terms of wall-time than off-policy methods, namely ones that can reuse samples from any policy. This is mainly due to the use of fast parallel simulators, as on-policy methods are easy to scale. In Chapter 3, we introduce a novel off-policy visual reinforcement learning algorithm, named Squint [Almuzairee and Christensen, 2026], that achieves faster wall-time training speed than prior visual reinforcement learning algorithms. Squint achieves this via parallel simulation, a distributional critic, resolution squinting, layer normalization, a tuned update-to-data ratio, and an optimized implementation. We train an

SO-101 robot arm using Squint for 15 minutes and deploy it zero-shot to our real robot setup, with most tasks converging in under 6 minutes.

To summarize, the rest of this thesis is organized as follows: we discuss improving visual generalization in Chapter 2, improving visual robustness in Chapter 3, and improving visual efficiency for reducing training times in Chapter 4. Finally, we discuss concluding remarks and future directions in Chapter 5.

Chapter 2

Visual Generalization

Visual Reinforcement Learning (RL) has a myriad of real-world applications [Mnih et al., 2013, Levine et al., 2016, Pinto and Gupta, 2016, Kalashnikov et al., 2018, Berner et al., 2019, Vinyals et al., 2019], and visual Q-learning algorithms are especially enticing because of their potential for high data-efficiency. However, they are very prone to overfitting on their training distribution due to the combination of flexible representation, high-dimensional data, and limited visual diversity in training environments [Peng et al., 2018, Cobbe et al., 2019, Julian et al., 2020].

Data augmentation is a widely used technique for learning visual invariances in supervised learning [Noroozi and Favaro, 2016, Tian et al., 2019, Chen et al., 2020], but has been found to cause training instabilities when applied to visual RL [Lee et al., 2019a, Laskin et al., 2020, Hansen and Wang, 2021]. Prior work, SVEA [Hansen et al., 2021b], found that a more selective application of data augmentation in the critic update of actor-critic algorithms [Lillicrap et al., 2015, Haarnoja et al., 2018] improved training stability significantly. The actor (*policy*) – which shares its visual backbone with the critic (Q*-function*) – is then optimized solely from unaugmented observations. By sharing their visual backbone, the actor indirectly benefits from the learned invariances.

In this work, we revisit the data augmentation recipe proposed in SVEA, and discover an *assumption* that limits its practicality to augmentations of a photometric (color or light altering)

nature. SVEA *assumes* that an encoder's output embedding can become fully invariant to input augmentations. If an encoder's output is fully invariant to input augmentations, then an actor, only trained on unaugmented observations, can become robust to input augmentations indirectly through a shared actor-critic encoder. However, this leads to two key failure cases: *(i)* the output of a convolutional neural network (CNN) encoder can not become invariant to input *geometric* augmentations *e.g.*, rotation or translation; (see Figure 2.1) and *(ii)* the encoder and critic are trained end-to-end, thus, part of the invariance may be off-loaded to the critic regardless of the type of augmentation.

To address these limitations, we propose **SADA**: **S**tabilized **A**ctor-Critic under **D**ata **A**ugmentation, a generalized data augmentation recipe that supports a wide variety of augmentations. Instead of only augmenting critic inputs, SADA augments *both* actor and critic inputs, but does so carefully to avoid training instabilities: *(1)* in actor updates, only the policy input is augmented while the Q-function input is left unaugmented, *(2)* in critic updates, only the online Q-function input is augmented while the target Q-function input is unaugmented, and *(3)* we jointly optimize components on both augmented and unaugmented data. Importantly, SADA requires no additional forward passes, losses, or parameters.

To stress-test our method, we propose DMC-GB2, an extension of the DeepMind Control Suite Generalization Benchmark [Hansen and Wang, 2021] that encompasses a wider and more challenging collection of test environments than existing benchmarks. We benchmark methods across DMC-GB2, tasks from Meta-World [Yu et al., 2020], and the Distracting Control Suite [Stone et al., 2021], and find that SADA greatly improves training stability and generalization of RL agents under a diverse set of augmentations.

2.1 Prior Work on Data Augmentation for Visual RL

The practice of learning visual invariances by augmenting data is ubiquitous in machine learning literature, and has been studied extensively in the context of supervised and self-

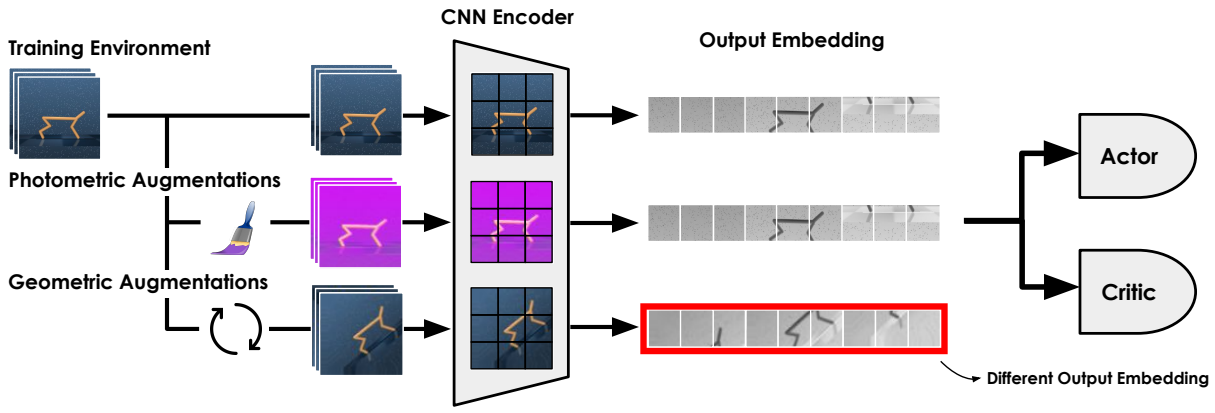


Figure 2.1. **Augmentation Effect on CNN Output.** We illustrate how the output embedding of a trained CNN changes wrt. image augmentations. The output of unaugmented and photometrically augmented images are identical due to the ability of a CNN to learn color invariances. However, the output of a CNN is generally not invariant to *geometric* augmentations (*e.g.*, rotation).

supervised learning algorithms for computer vision problems [Noroozi and Favaro, 2016, Wu et al., 2018, Oord et al., 2018, Tian et al., 2019, Chen et al., 2020, He et al., 2022]. More recently, use of augmentation has also been popularized in the context of visual RL. However, there is mounting evidence that much of the wisdom and practices developed in other areas (*e.g.* computer vision) do not translate to RL problems, presumably due to differences in learning objectives, datasets, and network architectures used. For example, while machine learning literature commonly considers a *fixed* dataset, RL algorithms are often trained on a non-stationary data distribution (replay buffer) that changes throughout training, and incoming data is typically a function of the current (behavioral) policy. As a result, RL datasets are often small and have limited diversity. This section provides an overview of prior work that leverages data augmentation to improve *data-efficiency* and *generalization*.

Improving *data-efficiency* with data augmentation. Much of the existing literature on visual RL leverages *weak* data augmentation (*e.g.* random crop or image shift) as a regularizer when data is limited, *i.e.*, when data-efficiency is critical [Srinivas et al., 2020, Laskin et al., 2020, Kostrikov et al., 2020, Stooke et al., 2020, Yarats et al., 2021b, Hansen et al., 2023], without particular emphasis on generalization or robustness to changes in the environment. For

example, seminal works RAD [Laskin et al., 2020] and DrQ [Kostrikov et al., 2020] demonstrate that randomly cropping or shifting images, respectively, by a few pixels greatly improves data-efficiency and training stability of Q-learning algorithms – even when agents are trained and tested in the same environment. However, Laskin et al. [2020] simultaneously find that other types of augmentation (rotation, random convolution, masking) lead to training instabilities and a substantial *decrease* in data-efficiency.

Improving *generalization* with data augmentation. Visual generalization is a challenging but increasingly important problem in RL due to its limited data diversity. Multiple prior works aim to improve the training stability and generalization of RL algorithms by, *e.g.*, proposing new types of augmentation [Lee et al., 2019a, Wang et al., 2020, Hansen and Wang, 2021, Hansen et al., 2021b, Zhang and Guo, 2021, Wang et al., 2023], or introducing new (auxiliary) objectives [Raileanu et al., 2020, Hansen et al., 2021a, Wang et al., 2021, Fan et al., 2021, Yuan et al., 2022a, Yang et al., 2024]. For example, Lee et al. [2019a] augment high-frequency content in observations using random convolution, Hansen and Wang [2021] randomly overlay observations with out-of-domain images, and Yang et al. [2024] adapt to camera changes at test-time using an auxiliary self-supervised objective and augmented data. Perhaps most importantly, SVEA [Hansen et al., 2021b] investigate *why* strong augmentations (such as those used in the aforementioned works) often destabilize training in an RL context, and propose an alternative method of applying augmentations that mitigate these instabilities. Our work builds upon SVEA and demonstrates that – while SVEA is robust to *photometric* augmentations – it largely fails when applied to (equally important) *geometric* augmentations.

We recommend the survey by Kirk et al. [2023] for a more comprehensive overview of prior work.

2.2 Background & Definitions

Visual Reinforcement Learning (RL) formulates interaction between an agent and its environment as a Partially Observable Markov Decision Process (POMDP) [Kaelbling et al., 1998]. A POMDP can be formalized as a tuple $(\mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{T}, R, \gamma)$, where $\mathcal{S}$ is an unobservable state space, $\mathbf{o} \in \mathcal{O}$ are observations from the environment (*e.g.*, RGB images), $\mathbf{a} \in \mathcal{A}$ are actions, $\mathcal{T}\colon \mathcal{S} \times \mathcal{A} \mapsto \mathcal{S}$ is a transition function, r is a task reward from a reward function $R\colon \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$, and γ is a discount factor. Throughout this work, we approximate the unobservable states $\mathbf{s} \in \mathcal{S}$ by defining observations as a stack of the three most recent RGB frames $\mathbf{o}_t \doteq \{\mathbf{x}_t, \mathbf{x}_{t-1}, \mathbf{x}_{t-2}\}$ for frames $\mathbf{x}_{t:t-2}$ at time t [Mnih et al., 2013]. The goal is then to learn a policy $\pi\colon \mathcal{O} \mapsto \mathcal{A}$ such that the discounted sum of rewards $\mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$ is maximized (in expectation) when following the policy π.

Q-Learning algorithms developed for visual RL generally estimate the optimal state-action value function $Q^*\colon \mathcal{O} \times \mathcal{A} \mapsto \mathbb{R}$ with a neural network (denoted the *critic*). This is achieved by dynamic programming using the single-step Bellman error $Q(\mathbf{o}_t, \mathbf{a}_t) - y_t$ where y_t is the temporal difference (TD) target $y_t \doteq r_t + \gamma Q(\mathbf{o}_{t+1}, \mathbf{a}_{t+1}),\ \mathbf{a}_{t+1} \sim \pi(\cdot|\mathbf{o}_{t+1})$. In practice, the Q-network used to compute y_t is usually chosen to be an exponential moving average of the Q-function being learned [Lillicrap et al., 2015, Haarnoja et al., 2018]. The policy π is obtained by taking the action $\mathbf{a}_t \approx \arg\max_{\mathbf{a}_t} Q(\mathbf{o}_t, \mathbf{a}_t)\ \forall \mathbf{o}_t$ in the current dataset (replay buffer), which is typically estimated by training a separate *actor* network when $\mathcal{A}$ is continuous. These two components – the actor and the critic – are iteratively updated by collecting data in the environment, appending it to a replay buffer $\mathcal{D}$, and optimizing Q, π with the following objectives using stochastic gradient descent:

$$\mathcal{L}_Q(\mathcal{D}) = \mathbb{E}_{(\mathbf{o}_t, \mathbf{a}_t, r_t, \mathbf{o}_{t+1}) \sim \mathcal{D}}\left[\|Q(\mathbf{o}_t, \mathbf{a}_t) - y_t\|_2\right] \quad \text{(critic)} \tag{2.1}$$

$$\mathcal{L}_\pi(\mathcal{D}) = \mathbb{E}_{\mathbf{o}_t \sim \mathcal{D}}\left[-Q(\mathbf{o}_t, \pi(\mathbf{o}_t))\right] \qquad \text{(actor)} \tag{2.2}$$

where gradients of the first objective are computed wrt. Q only, and gradients of the second objective are computed wrt. π only. When learning from images, observations are commonly encoded using a shared convolutional encoder f such that Q, π are redefined as $Q(f(\mathbf{o}_t), \mathbf{a})$ and $\pi(f(\mathbf{o}_t))$, with f only being updated by the critic objective $\mathcal{L}_Q$. Due to the recurrent and self-referential nature of Equations 2.1-2.2, Q-learning algorithms are often more data-efficient than other algorithm classes, but are very prone to training instabilities – especially when data augmentation is applied to observations.

Image transformations. Throughout this work, we classify image transformations into two types: *photometric* and *geometric* transformations. Photometric transformations alter image color and lighting properties while preserving the spatial arrangement of pixels (*e.g.* random convolution, image overlay). Geometric transformations alter the spatial arrangement of pixels while keeping image color and lighting properties intact (*e.g.* rotation, shift). We visualize examples of photometric and geometric transformations in Figure 2.1.

2.3 Stabilized Actor-Critic Learning under Data Augmentation

We revisit common wisdom and practices when applying data augmentation in Q-learning algorithms, and discover that prior work makes an assumption that only holds for augmentations that are photometric in nature. We propose **SADA**: **S**tabilized **A**ctor-Critic Learning under **D**ata **A**ugmentation, a generalized recipe for data augmentation that significantly improves the performance of a wider variety of augmentations. We start by outlining the assumptions of prior work, its implications, and then present our proposed solution.

2.3.1 Shortcomings of Prior Work

Naive augmentation, where all inputs are indiscriminately augmented, has been shown to lead policies to suboptimal convergence [Raileanu et al., 2020, Hansen et al., 2021b]. Unlike supervised learning, the application of augmentation in RL can lead to a conflict of task objective,

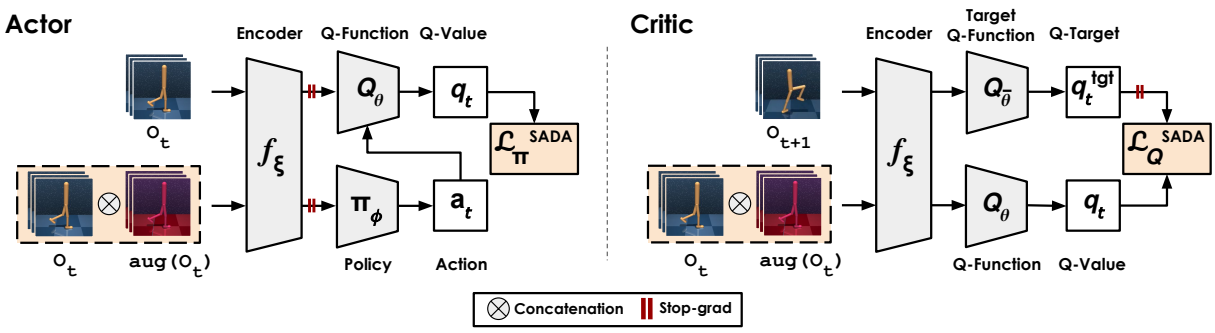


Figure 2.2. **Our approach.** Overview of SADA applied to a generic actor-critic algorithm. We highlight our algorithmic contributions in yellow. SADA *selectively* applies augmentations to the actor and critic inputs, and modifies the learning objectives accordingly.

conflict of learning objective, or increased variance that exacerbates instabilities within actor-critic frameworks.

To stabilize actor-critic learning under *strong* applications of data augmentation, SVEA [Hansen et al., 2021b] selectively applies augmentations in the critic updates, without any application of augmentation in the actor updates. The actor – optimized purely from unaugmented observations – becomes robust to augmentations indirectly, through the use of a shared actor-critic encoder. By using this formulation, SVEA *assumes* that the encoder can output embeddings that are invariant to input augmentations, such that an actor can indirectly become robust to input augmentations. This assumption leads to two key failure cases: *(i)* the output embedding of a CNN encoder can not become invariant to input *geometric* augmentations, *(ii)* even with *photometric* augmentations, part of the robustness could be off-loaded to the critic.

We provide a motivating example for key failure case (i) in Figure 2.1 and show that geometric transformations will always induce changes in a CNN's output embedding. Therefore, an actor not directly trained on these changed output embeddings will not become robust to these geometric transformations. As for key failure case (ii), a CNN can learn to output embeddings that are invariant to input photometric augmentations. However, the objective is formulated such that the output of the critic is robust to input image augmentations, indicating that if either the encoder or the critic is robust, the objective will be satisfied. Therefore, some of the photometric

resistance might be contained within the critic, rendering the actor weaker against photometric transformations.

2.3.2 Our Proposed Recipe

To mitigate shortcomings of previous works, the actor needs to *directly* train on the augmented stream. However, naively training the actor on the augmented stream exacerbates training instabilities. Each image augmentation applied adds a more complex distribution for the agent to learn compared to the original training distribution. To overcome this complexity, we introduce SADA, a general framework for stabilizing actor-critic agents under *strong* applications of data augmentation.

In the actor's update, we elect to use asymmetric observation inputs to the policy and Q-function [Pinto et al., 2017]. Specifically, we allow the policy to observe *both* the augmented and unaugmented streams, while the Q-function estimates the Q-value observing *only* the unaugmented stream. Since the Q-value estimates of both the augmented and unaugmented streams should be identical, we allow the Q-function to exploit only the unaugmented stream (easier distribution) in making accurate Q-value estimates. Given an observation $\mathbf{o}_t$, replay buffer $\mathcal{D}$, and an encoder f_ξ, the actor objective for a generic actor critic thus becomes:

$$\mathcal{L}_{\pi_\phi}^{\textbf{SADA}}(\mathcal{D}) = \mathbb{E}_{\mathbf{o}_t \sim \mathcal{D}}\left[-Q_\theta(\mathbf{m}_t, \pi_\phi(\mathbf{p}_t))\right] \qquad \text{(actor)} \tag{2.3}$$

where $\mathbf{p}_t = f_\xi([\mathbf{o}_t, \mathbf{o}_t^{\text{aug}}]_N)$, $\mathbf{m}_t = f_\xi([\mathbf{o}_t, \mathbf{o}_t]_N)$ and $\mathbf{o}_t^{\text{aug}} = \text{aug}(\mathbf{o}_t, v_t), v_t \sim \mathcal{V}$. We use $[\cdot]_N$ to denote concatenation for batch size of dimensionality N where $\mathbf{o}_t, \mathbf{o}_t^{\text{aug}} \in \mathbb{R}^{N \times C \times H \times W}$. We use aug() as the augmentation operator where we stochastically sample from the augmentation distribution $\mathcal{V}$ and apply it to the input observation.

In the critic update, we apply a similar asymmetric observation setup with the Q-value and Q-target estimates. We allow the online Q-function, Q_θ, to estimate the Q-value observing both the augmented and unaugmented streams, while the target Q-function, $Q_{\bar{\theta}}$, estimates the

Q-targets observing only the unaugmented stream. Since the Q-target estimates of both the augmented and unaugmented streams should be identical, this reduces the variance in Q-target estimates and allows the target Q-function to exploit the unaugmented stream (easier distribution) in making accurate Q-target estimates. The Q-target estimate, q_t^{tgt}, is unaltered while the critic objective, $\mathcal{L}_{Q_\theta}^{\textbf{SADA}}$, is changed such that:

$$q_t^{tgt} = r(\mathbf{o}_t, \mathbf{a_t}) + \gamma \text{max}_{\mathbf{a}'_\mathbf{t}} Q_{\bar{\theta}}(f_\xi(\mathbf{o}_{t+1}), \mathbf{a}') \tag{2.4}$$

$$\mathcal{L}_{Q_\theta}^{\textbf{SADA}}(\mathcal{D}) = \mathbb{E}_{(\mathbf{o}_t, \mathbf{a}_t, r_t, \mathbf{o}_{t+1}) \sim \mathcal{D}} \left[\| Q_\theta(\mathbf{p}_t, \mathbf{a}_t) - \mathbf{y}_t \|_2 \right] \quad \text{(critic)} \tag{2.5}$$

where $\mathbf{p}_t = f_\xi([\mathbf{o}_t, \mathbf{o}_t^{\text{aug}}]_N)$, and $\mathbf{y}_t = \left[q_t^{tgt}, q_t^{tgt}\right]_N$. An overview of our algorithm is provided in Figure 2.2. A detailed SAC-based formulation is provided in Appendix A.1.3, and pseudocode is provided in Appendix A.1.4.

2.4 Experiments

We evaluate our method and baselines across 11 visual RL tasks from the DMControl [Tassa et al., 2018] and Meta-World-v2 [Yu et al., 2020] benchmarks and 12 test distributions from our proposed DMControl - Generalization Benchmark 2 (DMC-GB2). We additionally evaluate on the Distracting Control Suite [Stone et al., 2021] and provide the results in Appendix A.2.2. All methods are trained under *strong* augmentations in the training environments and evaluated in a zero-shot manner on their respective test distributions. See Figure 2.3 for a visualization of DMC-GB2 test environments. The full DMControl and Meta-World task lists are provided in Appendix A.1.2. Concretely, we aim to answer the following questions through experimentation:

- **Robustness.** How does SADA compare to baselines in terms of *overall* augmentation robustness? In terms of *geometric* vs *photometric* robustness?
- **Analysis.** Why do baselines fail to display *geometric* robustness, and how does SADA solve the problem? How do each of the SADA *design choices* affect results?

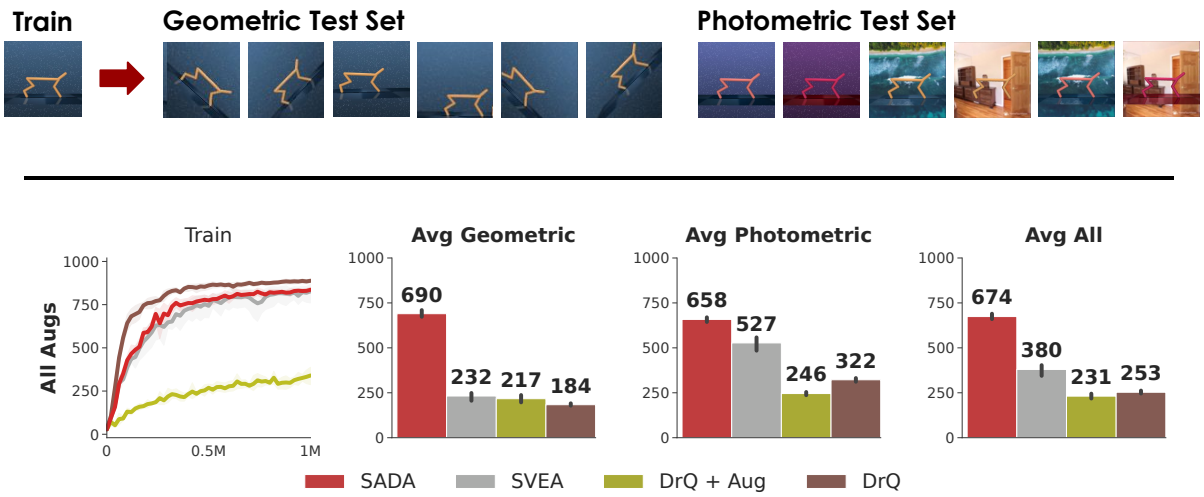


Figure 2.3. **Overall Robustness.** *(Top)* Samples from the DMC-GB2 test distributions, divided into geometric and photometric test sets. *(Bottom)* Episode reward on DMC-GB2 when trained under all (geometric and photometric) augmentations, averaged across all DMControl tasks. Mean and 95% CI over 5 seeds.

- **Generality.** Can SADA be *readily applied* to other RL backbones and benchmarks?

Setup. We build on DrQ [Laskin et al., 2020] as our backbone algorithm, and use a fixed set of hyperparameters across all tasks and environments. All agents are trained for one million environments steps and use stacks of the three most recent RGB frames ($3 \times \mathbb{R}^{(\mathbf{3\times84\times84})}$) as observations. A full list of hyperparameters and training details is provided in Appendix A.1.

Test environments. As our experiments will reveal, previous methods largely fail to generalize to geometric changes, which has gone unnoticed due to existing benchmarks predominantly evaluating photometric robustness. Therefore, we propose the Deepmind Control Suite Generalization Benchmark 2 (DMC-GB2), an extension of DMC-GB [Hansen and Wang, 2021] to encompass a wider collection of photometric and geometric test distributions. In DMC-GB2, we provide geometric and photometric test sets. The geometric test set considers two types of geometric distributions – rotations and shifts – both individually and jointly, and at varying intensities categorized as *easy* and *hard* environments. The photometric test set considers a complementary setup for two types of photometric distributions – colors and videos. Detailed visualizations of all 12 DMC-GB2 test distributions is provided in Appendix A.3.2.

Data augmentation. We apply a *weak* augmentation of random shifting to all our inputs as conducted in our DrQ baseline, and consider it unaugmented. We further employ a set of *strong* augmentations, taking into account both *geometric* and *photometric* transformations. For geometric augmentations, we use random shift [Laskin et al., 2020], random rotation, and a combination consisting of random rotation followed by random shift. For photometric augmentations we use random convolution [Lee et al., 2019a], random overlay [Hansen and Wang, 2021], and a combination consisting of random convolution followed by random overlay. We sample an augmentation from this set of six *strong* augmentations for each input sample in all our experiments unless stated otherwise. Detailed visualizations of all augmentations is provided in Appendix A.3.1.

Baselines. We benchmark our method against the following well-established baselines. 1) **DrQ** Laskin et al. [2020], a visual based Soft Actor Critic baseline that uses random shifts as the default augmentation for all inputs. 2) **DrQ + Aug**, a variant of DrQ implemented with a naive application of *strong* augmentations. 3) **SVEA** Hansen et al. [2021b], which builds on DrQ with a selective application of augmentation in the Q-function to increase robustness under *strong* augmentations.

2.4.1 Results & Discussion

Robustness. For measuring the *overall* robustness, we train all methods under all augmentations (geometric and photometric augmentations) and evaluate them on all DMC-GB2 test sets (geometric and photometric test sets). As our empirical results indicate in Figure 2.3, SADA's *overall* robustness surpasses the baselines in all DMC-GB2 test sets by a large margin (77%), all while attaining a similar sample efficiency to its unaugmented DrQ baseline on the training environment.

To analyze the *geometric* vs *photometric* robustness of each method, we conduct another experiment where we train under each set of augmentations separately. We train under strong geometric augmentations and evaluate on the geometric test set, and follow a complementary

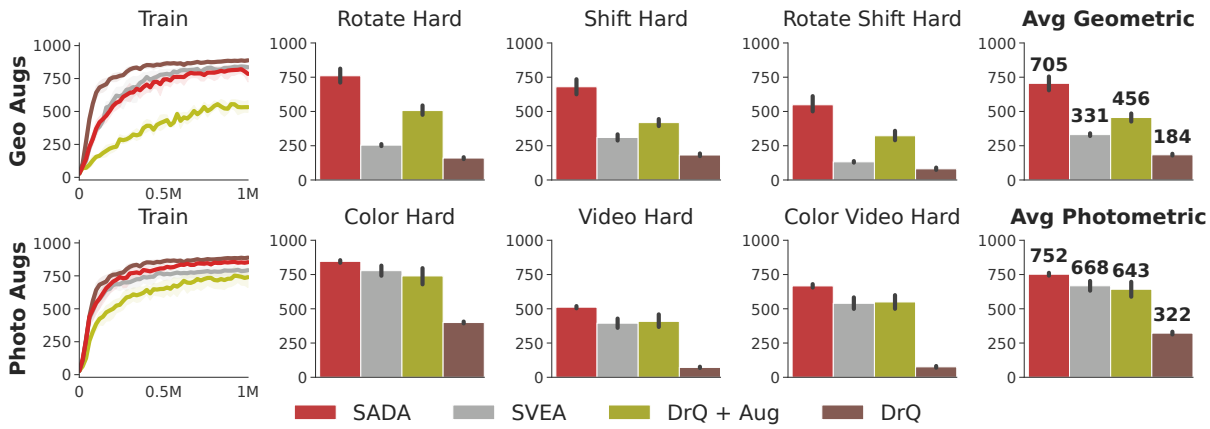


Figure 2.4. **Geometric vs Photometric Robustness.** Episode reward averaged over all DMControl tasks. *(Top)* Trained under geometric augmentations and evaluated on DMC-GB2 geometric test set. *(Bottom)* Trained under photometric augmentations and evaluated on DMC-GB2 photometric test set. All hard levels visualized. Mean and 95% CI over 5 random seeds.

setup under strong photometric augmentations with the photometric test set. We visualize the results in Figure 2.4 along with all the individual hard level intensities in our test suite. SADA *consistently* shows superior robustness, outperforming baselines in all separate test sets and individual levels while achieving similar training sample efficiency to its unaugmented DrQ baseline. Extended results and per-task breakdowns are provided in Appendix A.4.

Analysis. While baselines show varying degrees of photometric robustness, they *fail to display geometric robustness* in Figures 2.3 and 2.4. For the DrQ baseline, geometric transformations are out of its training distribution. With naive application of data augmentation, DrQ + Aug achieves poor training sample efficiency. To achieve high training sample efficiency, SVEA selectively applies augmentation in the critic update. Nevertheless, this performance does not translate to the geometric test distributions due to key failure case (i), the output embedding of a CNN can not become invariant to input geometric transformations. This failure case can only be resolved with an actor *directly* trained on the input augmentations. When the actor is *directly* trained on the input augmentations using SADA's objective, the agent is able to achieve high training sample efficiency that effectively translates to the geometric test distributions.

Even in terms of photometric robustness, SADA surpasses all baselines, including SVEA. This is mainly due to failure case (ii) of SVEA's assumption, where some of the augmentation robustness is contained within the critic and not the encoder. This can also be resolved by training the actor *directly* on the input augmentations using SADA's objective.

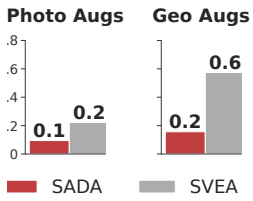


Figure 2.5. **Actor Prediction Variance.** Actor prediction variance between augmented and unaugmented observations. ↓ Lower is better.

To *quantitatively* assess the augmentation robustness of converged SADA and SVEA agents, we measure the variance of actions predicted on the augmented observations with respect to the unaugmented observations in Figure 2.5. Despite being trained on all augmentations, SVEA's action predictions have high variance when observing geometric augmentations as opposed to photometric augmentations, confirming SVEA's shortcomings. For a *qualitative* assessment, we utilize T-SNE to visualize the encoder's output embedding before being fed into the actor and the critic in Appendix A.2.3. Our findings reveal that photometric distributions can share the same space in the latent embedding as the original training distribution, while geometric distributions are distant in the latent space and seem to have little overlap with the training distribution, necessitating the need to *directly* train the actor on them.

We ablate each of our *design choices*, evaluating methods under all augmentations on all DMC-GB2 test sets, and show results in Figure 2.6. Naively applying augmentation to the actor or the critic, as displayed in SADA (Naive Actor Aug) and SADA (Naive Critic Aug) respectively, leads to deteriorated performance. As for SADA (No Critic Aug), we only apply augmentation to the actor using SADA's objective without any application of augmentation to the critic. SADA (No Critic Aug) displays impressive geometric robustness and training sample efficiency, but lacks in photometric robustness. If a user is only interested in geometric robustness, SADA (No Critic Aug) provides commendable geometric robustness. Overall, each of our design choices play a key role in establishing the superiority of SADA in all applications of data augmentation.

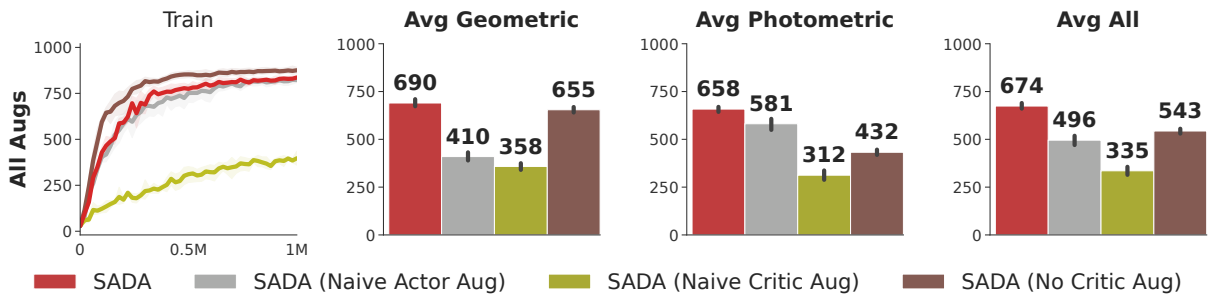


Figure 2.6. **Ablations.** Episode reward on DMC-GB2 when trained under all augmentations, averaged across all DMControl tasks. SADA (Naive Actor Aug) and SADA (Naive Critic Aug) correspond to naive application of augmentation to the actor and the critic respectively. SADA (No Critic Aug) corresponds to applying augmentation only to the actor without any application of augmentation to the critic. More details in Appendix A.2.1. Mean and 95% CI over 5 random seeds.

Generality. To demonstrate the generality of our approach, we swap our DrQ backbone with TD-MPC2 [Hansen et al., 2022, 2024], a state-of-the-art model-based RL algorithm; results are shown in Figure 2.7. We observe that SADA similarly improves training stability and generalization of TD-MPC2 on DMC-GB2.

We further evaluate our DrQ-based SADA on our Meta-World setup (see Appendix A.3.3), and showcase the results in Figure 2.8. Even on Meta-World, SADA surpasses all other baselines in terms of success rate, all while achieving similar training sample efficiency to its unaugmented DrQ baseline. This asserts that SADA can be *readily applied* to diverse tasks, environments, and backbones, and can be used a generic data augmentation strategy for modern visual based reinforcement learning.

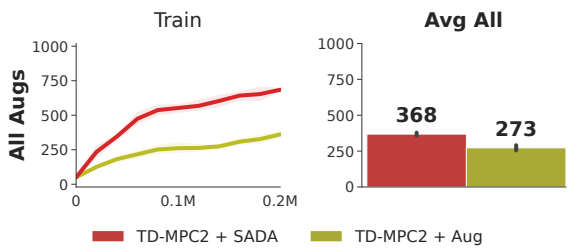


Figure 2.7. **TD-MPC2 Baseline.** Episode reward on DMC-GB2 when trained under all augmentations with a TD-MPC2 backbone, averaged across all DMControl tasks. Mean and 95% CI over 5 seeds.

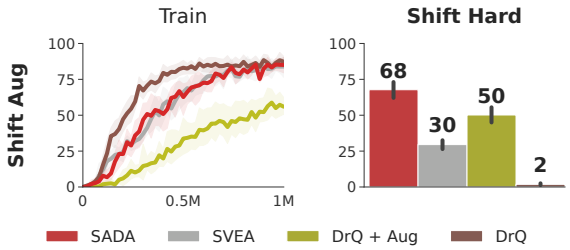


Figure 2.8. **Meta-World.** Success rate (%) on Shift Hard (Meta-World) distribution when trained under *strong* shift augmentation only, averaged across all Meta-World tasks. Mean and 95% CI for 5 random seeds.

2.5 Summary

Throughout this work, we give an overview of data augmentation within visual RL, highlighting the shortcomings of previous work, its implications, and presenting SADA, a generic data augmentation recipe for modern visual based reinforcement learning. We empirically prove SADA's superiority to previous methods and provide a deep analysis of its effectiveness. Concurrently, we curated a comprehensive visual generalization benchmark, DMC-GB2, which we make publicly available at https://aalmuzairee.github.io/SADA, with the aim of furthering research efforts within visual RL.

Chapter 2 and Appendix A, in full, are a reprint of the material as it appears from "A Recipe for Data Augmentation in Visual Reinforcement Learning" by Abdulaziz Almuzairee, Nicklas Hansen, and Henrik I. Christensen. In Reinforcement Learning Journal, 2024. The dissertation author was the primary investigator and author of this paper.

Chapter 3

Visual Robustness

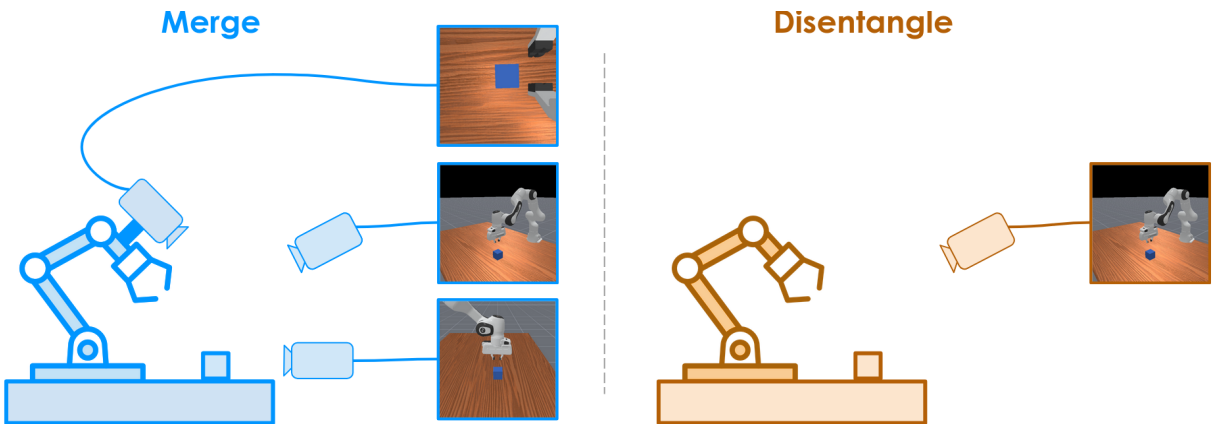


Figure 3.1. **Merge And Disentangle (MAD).** We introduce a method that can merge multiple camera views during training to learn better representations, while simultaneously disentangling the camera view representations, such that the policy can function with any singular view input during deployment.

Visual reinforcement learning (RL) has demonstrated remarkable progress, achieving superhuman performance on Atari games [Mnih et al., 2013] and successfully tackling complex real-world applications from robotic manipulation to autonomous flight [Levine et al., 2016, Pinto and Gupta, 2016, Kalashnikov et al., 2018, Kaufmann et al., 2023]. However, the fundamental limitation of learning from 2D observations constrains these systems' understanding of the 3D world, particularly in robotics where depth perception and control are differentiators.

Multi-view approaches offer promising solutions for visual RL, especially when combined

with data efficient Q-learning algorithms. They can *merge* multiple views to learn better representations, overcome occlusions, and achieve higher sample efficiency [Hsu et al., 2022]. However, they come with a practical challenge: conditioning policies on multi-view representations can be burdensome to deploy and render them fragile in the case of malfunctioning sensors. To deploy lightweight policies that are robust to a reduction in available camera views, policies must be carefully *disentangled* during training [Dunion and Albrecht, 2024].

Aligning these two directions, we propose a method to **M**erge **A**nd **D**isentangle views (**MAD**), thereby leveraging multi-view representations for *improved sample efficiency*, while being *robust to a reduction of camera views* for lightweight deployment. To accomplish that, MAD processes each camera view input individually through a single shared CNN encoder. Then, all singular view features are merged through feature summation to create a merged multi-view feature representation. This merged feature representation is passed down to the downstream actor (*policy*) and critic (*Q-function*) for learning. However, to properly disentangle view features, the downstream actor and critic need to also train on the singular view features. Naively training on both merged feature representations and all their singular view representations decreases sample efficiency and destabilizes learning, as they can be viewed as multiple different states by the downstream actor and critic networks. Therefore, we build upon SADA [Almuzairee et al., 2024], a framework for applying data augmentation to visual RL agents that stabilizes both the actor and the critic under data augmentation by *selectively* augmenting their inputs. We modify the RL loss objectives, such that we train on each merged multi-view feature, and apply all its singular view features as augmentations to it. By using this formulation, we are able to increase sample efficiency while ensuring robustness to a reduction in input camera views. Most importantly, we achieve this without requiring ordered input views, auxiliary losses, extra forward passes, or additional learnable parameters.

Methods that merge and disentangle are vital when deploying robots to new environments or designing benchmarks, where determining optimal camera placement and ensuring task observability is non-trivial. By learning to merge information from multiple views while

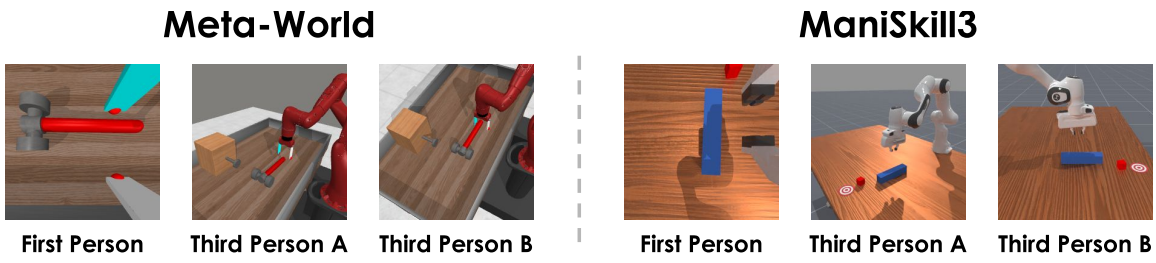


Figure 3.2. **Environment Setup.** Our environment setup for robotic manipulation where we use three input camera views as inputs, namely First Person, Third Person A, and Third Person B across two visual RL benchmarks: *(Left)* Meta-World *(Right)* ManiSkill3. Extended visuals in Appendix B.5.

maintaining the ability to generalize to individual perspectives, policies can leverage the complementary nature of different viewpoints, benefiting from the increased sample efficiency and ensuring robustness. This process of merging multiple camera views and simultaneously disentangling individual views to create robust policies is illustrated in Figure 3.1.

We evaluate our method on 20 visual RL tasks from Meta-World [Yu et al., 2020] and ManiSkill3 [Tao et al., 2024] benchmarks, and find that MAD is able to achieve higher sample efficiency than baselines while being robust to a reduction in available camera views.

3.1 Related Works

Data Augmentation in Visual Reinforcement Learning. To improve the visual generalization of models, data augmentation has been a commonly employed strategy in supervised and self-supervised learning for computer vision tasks [Noroozi and Favaro, 2016, Wu et al., 2018, Oord et al., 2018, Tian et al., 2019, Chen et al., 2020, He et al., 2022]. Due to its limited data diversity, visual reinforcement learning is especially prone to overfitting on visual inputs. Recent works have found that applying data augmentation with input image transformations such as random crops or random shifts regularizes the learning and increases sample efficiency in RL agents [Srinivas et al., 2020, Laskin et al., 2020, Kostrikov et al., 2020, Stooke et al., 2020, Yarats et al., 2021b,a]. In contrast, [Laskin et al., 2020, Raileanu et al., 2020]

find that stronger image transformations such as rotation, random convolution, and masking lead to training instabilities and a decrease in data efficiency. To mitigate this, many works focus on improving training stability under stronger data transformations [Raileanu et al., 2020, Hansen et al., 2021b, Wang et al., 2021, Fan et al., 2021, Yuan et al., 2022a, Grooten et al., 2023, Yang et al., 2024, Almuzairee et al., 2024] and increasing visual generalization by proposing new types of transformations [Lee et al., 2019b, Wang et al., 2020, Hansen and Wang, 2021, Zhang and Guo, 2021, Huang et al., 2022b, Wang et al., 2023, Teoh et al., 2024]. Most importantly, Almuzairee et al. [2024] propose a generic recipe for applying data augmentation, named SADA, which allows RL agents to generalize to many types of stronger image transformations, without sacrificing training sample efficiency. Our work builds on SADA and shows that it can be extended to feature level data augmentation within our framework.

Robot Manipulation from Multiple Views. While robots can learn from singular camera views, increasing the available cameras and using merged representations have been shown to improve performance in robotic manipulation. [Akinola et al., 2020, Hsu et al., 2022, Seo et al., 2023b, Gervet et al., 2023, Goyal et al., 2024, Qian et al., 2024, Yuan et al., 2024]. These approaches leverage multiple camera perspectives to mitigate occlusions and improve representations, leading to enhanced task performance [Szot et al., 2021, Hsu et al., 2022]. Most notably, Hsu et al. [2022] show that a first-person view outperforms a third-person view when considering only one single view. However, due to the first-person view's limited observability, fusing it with a third person camera improves learning, as long as the third person view is regularized with a variational information bottleneck. They show that their method, VIB, outperforms singular and combined views on tabletop robot manipulation. Jangir et al. [2022] achieve similar conclusions, where fusing first and third person views with cross attention achieves better real-world transfer than their singular view counterparts. Although there is merit in using multiple views, performance degradation is significant when a view that was available during training is not present during inference, especially if the views were not disentangled properly during training [Dunion and Albrecht, 2024].

Merging and Disentangling Features. Research in computer vision has extensively explored feature merging [Su et al., 2015, Yao et al., 2018, Borse et al., 2023, Li et al., 2024] and feature disentanglement [Oord et al., 2018, Khosla et al., 2020, Oquab et al., 2023] across multiple camera views and sensor modalities. Inspired by these efforts, multiple works in visual RL have explored feature merging [Li et al., 2019, Yang et al., 2022, Jangir et al., 2022, Hsu et al., 2022] and feature disentanglement [Eysenbach et al., 2018, Srinivas et al., 2020, Park et al., 2023]. However, policies trained on multiple feature inputs often suffer significant performance degradation when any feature inputs become unavailable [Vasco et al., 2021, Dunion and Albrecht, 2024]. Multiple works have researched this problem of feature disentanglement for reinforcement learning. Hu et al. [2024] explore the use of scaffolding through extra privileged features at training time, and disentangling them to achieve better test time performance than their unscaffolded counterparts. Skand et al. [2024] introduce a multi sensor feature encoder for RL polices that disentangles features through dropout, yielding robust policies. Recently, Dunion and Albrecht [2024] addressed feature disentanglement in the multi-camera view RL setup for robotic manipulation. They introduced MVD, an actor-critic method that *disentangles* view representations into shared and private components through contrastive losses, enabling policies to maintain performance despite a reduction in input camera views. While MVD focuses exclusively on disentangling views, our approach focuses on both merging and disentangling views: we *merge* multi-view features to improve sample efficiency and we train on both merged multi-view features and singular-view features to ensure *disentanglement* in the case of a singular camera view input.

3.2 Preliminaries

Visual Reinforcement Learning formulates the interaction between the agent and the environment as a Partially Observable Markov Decision Process (POMDP) [Kaelbling et al., 1998], defined by a tuple of $\langle \mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{T}, R, \gamma \rangle$. In a POMDP, $\mathcal{S} \subseteq \mathbb{R}^d$ is an unobservable state

space by the agent, $o \in \mathcal{O}$ is the observation space, $a \in \mathcal{A}$ the action space, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the state-to-state transition probability function, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function and γ is the discount factor. To better approximate the current state $s_t \in \mathcal{S}$, we follow Mnih et al. [2013] in defining observations as a stack of three prior consecutive RGB frames at time t. The goal is to learn a policy $\pi : \mathcal{O} \to \mathcal{A}$ that maximizes the expected discounted sum of rewards $\mathbf{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r_t]$ where $r_t = R(s_t, a_t)$.

Data Regularized Q-Learning (DrQ) [Kostrikov et al., 2020] is an actor critic algorithm, based on Soft Actor Critic [Haarnoja et al., 2018] commonly used for continuous control in visual RL. DrQ concurrently learns a Q-function Q_θ (critic) and a policy π_ϕ (actor) with a seperate neural network for each. The Q-function Q_θ aims to estimate the optimal state-action value function Q^*: $\mathcal{O} \times \mathcal{A} \mapsto \mathbb{R}$ by minimizing the one step Bellman Residual $\mathcal{L}_{Q_\theta}(\mathcal{D}) = \mathbb{E}_{(\mathbf{o}_t, \mathbf{a}_t, r_t, \mathbf{o}_{t+1}) \sim \mathcal{D}}[(Q_\theta(\mathbf{o}_t, \mathbf{a}_t) - r_t - \gamma Q_{\bar{\theta}}(\mathbf{o}_{t+1}, \mathbf{a}'))^2]$ where $\mathcal{D}$ is the replay buffer, $\mathbf{a}'$ is an action sampled from the current policy $\mathbf{a}' \sim \pi_\phi(\cdot|\mathbf{o}_{t+1})$, and $Q_{\bar{\theta}}$ represents an exponential moving average of the weights from Q_θ [Watkins and Dayan, 1992, Lillicrap et al., 2015, Haarnoja et al., 2018]. The policy π_ϕ is a stochastic policy with temperature alpha α that aims to maximize entropy and Q-values. For simplicity, we abstract the entropy objective and define a generic actor loss to be $\mathcal{L}_{\pi_\phi}(\mathcal{D}) = \mathbb{E}_{\mathbf{o}_t \sim \mathcal{D}}\left[-Q_\theta(\mathbf{o}_t, \pi_\phi(\mathbf{o}_t))\right]$. Both the critic and actor losses are updated iteratively using stochastic gradient descent in an aim to maximize the expected discounted sum of rewards. DrQ also applies small random shifts to all input images sampled from the replay buffer which improves learning sample efficiency [Kostrikov et al., 2020, Yarats et al., 2021a].

3.3 Merge And Disentangle Views in Visual Reinforcement Learning

Building on prior work in multi-view reinforcement learning, we found it to pursue two distinct directions: one focusing on merging views to improve sample efficiency, while the other focusing on disentangling views to create policies that are robust to view reduction. To unify these

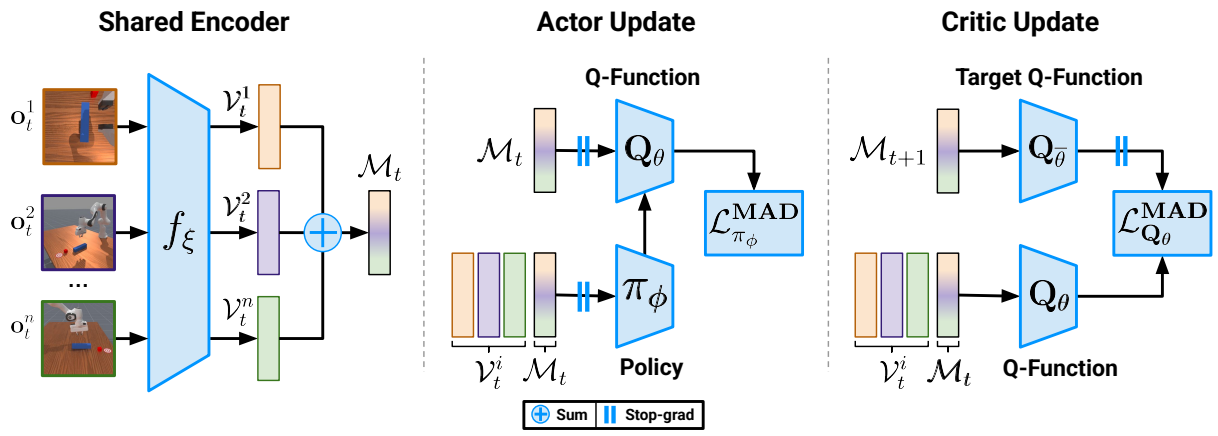


Figure 3.3. **MAD framework.** Update diagram of a generic visual actor-critic model with our modifications. MAD *merges* camera views through feature summation and *disentangles* camera views by selectively augmenting inputs to the downstream actor and critic with all the singular view features. The agent is trained end-to-end with our defined MAD loss functions. *(Left)*: Single Shared CNN Encoder. *(Middle)*: Actor Update diagram *(Right)*: Critic Update diagram.

complementary approaches in pursuit of both increased sample efficiency and policy robustness, we propose **MAD**: **M**erge **A**nd **D**isentangle views for visual reinforcement learning. Our method merges views through feature summation, and simultaneously disentangles views by training the downstream actor and critic on both the merged and singular view features. However, *naively* training on both merged and singular view features deteriorates policy performance. Therefore, we elect to train using the merged features, and apply singular view features as augmentations to the inputs of the downstream actor and critic networks. We start by defining our merge module, followed by our application of feature-level data augmentation, and finally our augmentation strategy.

3.3.1 Merge Module

Given a multi-view input $\mathbf{o}_t^m = \mathbf{o}_t^1, \mathbf{o}_t^2, ..., \mathbf{o}_t^n$ consisting of n views at time t, where each $\mathbf{o}_t^i$ represents a single view, each view is passed separately through a shared CNN encoder f_ξ. The output features of a single view input is then defined to be $\mathcal{V}_t^i = f_\xi(\mathbf{o}_t^i)$. After encoding all singular views $\mathcal{V}_t^1, \mathcal{V}_t^2, ..., \mathcal{V}_t^n$, the features are merged through summation such that the combined multi-view representation becomes: $\mathcal{M}_t = \sum_i^n \mathcal{V}_t^i$.

Feature summation is chosen as the merging method for two reasons: 1) To make the downstream actor and critic robust to a reduction in views, they need to be trained on both merged features $\mathcal{M}_t$ and singular-view features $\mathcal{V}_t^i$. With feature summation, both $\mathcal{M}_t$ and $\mathcal{V}_t^i$ have equal dimensionality, and thus no architectural changes need to be made to accommodate different feature dimensions in the case of missing input views. 2) Feature summation preserves the magnitude of different view features, such that the downstream actor and critic have a signal of how many views are inputted. As our experiments will show, most merging methods in visual RL perform similarly, and so the choice of the merging method comes down to the properties desired within a certain framework.

3.3.2 Feature-level Data Augmentation

Feature-level augmentation in MAD differs from traditional RL data augmentation techniques. While conventional approaches apply data augmentation by modifying input images through random cropping and color jittering, or altering input states via amplitude scaling and Gaussian noise injection [Laskin et al., 2020], MAD introduces augmentation at the feature level—specifically between the image encoder and the downstream actor and critic networks. The process begins by encoding each camera view image into its corresponding singular-view feature representation $\mathcal{V}_t^i$, then combining these features through summation to produce the merged representation $\mathcal{M}_t$. Standard multi-view algorithms would only pass this merged representation $\mathcal{M}_t$ to the downstream components. However, to strengthen the robustness of the downstream actor and critic to missing input camera views, MAD augments the downstream component inputs with all the singular-view feature representations $\mathcal{V}_t^i$.

3.3.3 Augmentation Strategy

To generalize to all singular view inputs $\mathbf{o}_t^i$, MAD applies their corresponding features V_t^i as feature-level augmentations to the downstream actor and critic. However, naive application of data augmentation in visual RL has been shown to degrade policy performance and stability

[Laskin et al., 2020, Raileanu et al., 2020, Almuzairee et al., 2024]. Later works established ways to stabilize actor-critic learning under data augmentation [Hansen et al., 2021b, Almuzairee et al., 2024]. We follow SADA [Almuzairee et al., 2024] in their recipe for applying data augmentation, where given an encoder f_ξ, actor π_ϕ, and critic Q_θ, data augmentation is *selectively* applied to the actor and critic inputs such that: *(1)* In the critic update, the target Q-value is predicted *purely* from the unaugmented stream, while the online Q-value is predicted from *both* the augmented and unaugmented streams. *(2)* In the actor update, the Q-value is predicted *purely* from unaugmented stream while the policy action is predicted from *both* the augmented and unaugmented streams. By predicting the learning targets in both actor and critic updates from the unaugmented stream, the variance in the targets is reduced, thereby stabilizing the learning objective under data augmentation.

We build on this loss formulation with some modifications. Given n input camera views, n single-view features $\mathcal{V}_t^i = f_\xi(\mathbf{o}_t^i)$ for $i \in \{1, ..., n\}$, and a multi-view feature representation $\mathcal{M}_t = \sum_{i=1}^{n} \mathcal{V}_t^i$, the MAD update loss function for generic actor becomes:

$$\mathcal{L}_{\pi_\phi}^{\textbf{UnAug}}(\mathcal{D}) = \mathbb{E}_{\mathbf{o}_t^m \sim \mathcal{D}} \left[-Q_\theta(\mathcal{M}_t, \pi_\phi(\mathcal{M}_t)) \right]$$

$$\mathcal{L}_{\pi_\phi}^{\textbf{Aug}}(\mathcal{D}, i) = \mathbb{E}_{\mathbf{o}_t^m \sim \mathcal{D}} \left[-Q_\theta(\mathcal{M}_t, \pi_\phi(\mathcal{V}_t^i)) \right]$$

$$\mathcal{L}_{\pi_\phi}^{\textbf{MAD}}(\mathcal{D}) = \alpha * \mathcal{L}_{\pi_\phi}^{\textbf{UnAug}}(\mathcal{D}) \; + \; (1-\alpha) * \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_{\pi_\phi}^{\textbf{Aug}}(\mathcal{D}, i) \quad \text{(actor)} \tag{3.1}$$

where α is a hyperparameter that we add to the SADA loss. This α hyperparameter weighs the unaugmented and augmented streams for more fine grained control over the learning, similar to Hansen et al. [2021b]. Setting this α hyperparameter to 0.5 recovers the original SADA

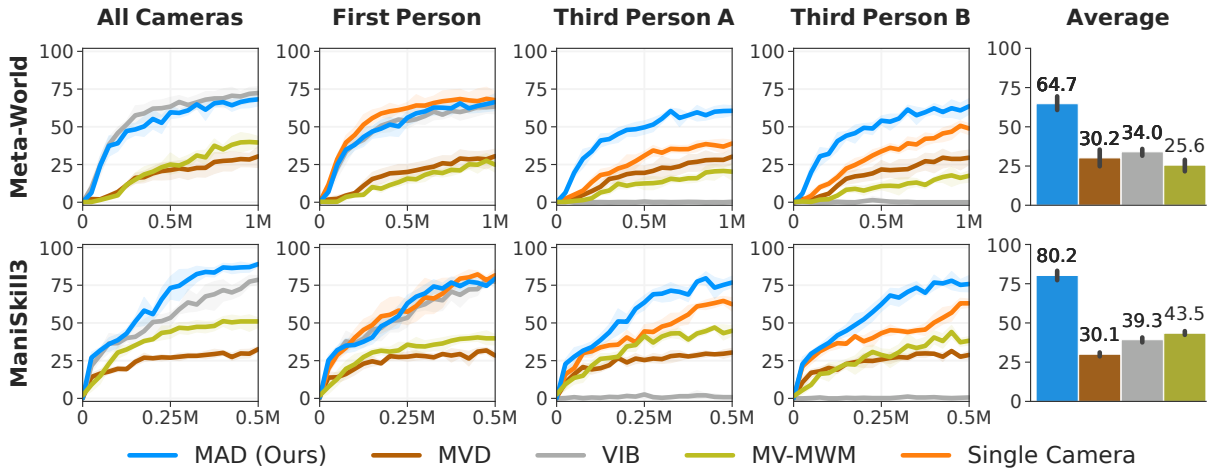


Figure 3.4. **Overall Robustness.** Success rate as a function of environment steps, averaged over all *(Top)* 15 Meta-World and *(Bottom)* 5 ManiSkill3 visual RL tasks. Methods are trained on all three camera views and evaluated on all and singular camera views with the final average displayed on the far right. Mean and 95% CI over 5 random seeds.

loss. On the other hand, the MAD update loss function for a generic critic becomes:

$$\mathcal{L}_{Q_\theta}^{\textbf{UnAug}}(\mathcal{D}) = \mathbb{E}_{(\mathbf{o}_t^m, \mathbf{a}_t, r_t, \mathbf{o}_{t+1}^m) \sim \mathcal{D}} \Big[\big(Q_\theta(\mathcal{M}_t, \mathbf{a}_t) \; - r_t - \gamma Q_{\bar{\theta}}(\mathcal{M}_{t+1}, \mathbf{a}') \big)^2 \Big]$$

$$\mathcal{L}_{Q_\theta}^{\textbf{Aug}}(\mathcal{D}, i) = \mathbb{E}_{(\mathbf{o}_t^m, \mathbf{a}_t, r_t, \mathbf{o}_{t+1}^m) \sim \mathcal{D}} \Big[\big(Q_\theta(\mathcal{V}_t^i, \mathbf{a}_t) \; - r_t - \gamma Q_{\bar{\theta}}(\mathcal{M}_{t+1}, \mathbf{a}') \big)^2 \Big]$$

$$\mathcal{L}_{Q_\theta}^{\textbf{MAD}}(\mathcal{D}) = \alpha * \mathcal{L}_{Q_\theta}^{\textbf{UnAug}}(\mathcal{D}) \; + \; (1-\alpha) * \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_{Q_\theta}^{\textbf{Aug}}(\mathcal{D}, i) \quad \text{(critic)} \tag{3.2}$$

A higher value of α would increase the weight of the unaugmented objective, while a lower α would increase the weight of the augmented objective. Using this formulation, and after tuning α, MAD is able to train on both the merged features and singular view features with minimal loss to training sample efficiency. A detailed diagram of our method update is provided in Figure 3.3.

3.4 Experiments

We benchmark our method and baselines on a total of 20 visual RL tasks, consisting of 15 Meta-World v2 tasks [Yu et al., 2020] and 5 ManiSkill3 tasks [Tao et al., 2024], focusing on tabletop robot manipulation tasks. Both environments are set up with three camera views,

consisting of a first-person camera attached to the robot arm, and two third-person cameras with reasonable observability to the task at hand. See Figure 3.2 for visualizations of our camera setup in each environment. The full task list is defined in B.1.3, and detailed visuals of all tasks can be found in B.5. Through experimentation, our aim is to answer the following questions:

- **Robustness.** How does MAD compare to baselines in terms of sample efficiency? Does MAD function with a reduction in camera views?
- **Analysis.** Why do baselines fail to achieve similar sample efficiency to MAD? How much does each component contribute within MAD? How do different merge methods compare?
- **Adaptability.** How well does MAD adapt with occluded views? Can MAD be adapted to other modalities?

Implementation. We use DrQ [Kostrikov et al., 2020], a visual based Soft Actor Critic algorithm [Haarnoja et al., 2018], as the backbone algorithm for our method across both environments, with detailed hyper parameters defined in Appendix B.1.1. To process all input camera views, we use a single shared DQN CNN encoder [Mnih et al., 2015]. For our observations, we pass in a stack of three most recent RGB images such that input observations for each view are of shape $(3 \times \mathbb{R}^{(\mathbf{3}\times\mathbf{84}\times\mathbf{84})})$. We further pass in proprioceptive robot state input to the actor and critic, similar to prior works [Hsu et al., 2022, Dunion and Albrecht, 2024]. When learning from images, DrQ regularizes its learning by applying small random shift transformations to input images, which has been shown to improve performance. We apply identical random shifts to all input camera views at each timestep.

Environments. We evaluate on 15 Meta-World [Yu et al., 2020] tasks spanning medium, hard, and very hard difficulties, as defined in Seo et al. [2023a], and 5 tasks from ManiSkill3 [Tao et al., 2024]. For each of the two benchmarks, we use a fixed camera setup consisting of a first person camera view and two third person camera views. When evaluating, we report the mean success rate of the agent across 20 episodes.

Baselines. We compare MAD against the following strong baselines. 1) **MVD** [Dunion and Albrecht, 2024], a contrastive learning method that learns to *disentangle* multi-view image inputs by encoding a shared and private representation for each view. The shared representation is pushed to align with other views while the private representation is pushed to differ. Through combined contrastive losses, they are able to learn representations that are robust to a reduction in views. 2) **VIB** [Hsu et al., 2022], a variational information bottleneck approach that *merges* first person and third person views with a variational information bottleneck objective applied only on third person views to provide complementary information to the first person view. 3) **MV-MWM** [Seo et al., 2023b], a multi-view masked autoencoder that learns representations across multiple views while using a world model for planning. While MV-MWM uses expert demonstrations to bootstrap its learning, we train it without any expert demonstrations for a fair comparison with our method and baselines. 4) **Single Camera** DrQ [Kostrikov et al., 2020], where DrQ agents are trained only on a single view and evaluated on that same single view.

3.4.1 Empirical Results

Robustness. To quantify the *sample efficiency* of MAD, we train all methods on 20 visual RL tasks from Meta-World and ManiSkill3, using all three cameras as input, and evaluate the methods on all cameras views combined, and individual, reporting the combined, individual and average success rates in Figure 3.4. As our results indicate, MAD demonstrates superior performance to baselines on Meta-World by (30%) success rate over 15 tasks, and on ManiSkill3 by (36%) over 5 tasks. These improvements in success rates within the same number of environment steps as baseline methods directly reflect the enhanced sample efficiency of MAD.

With a *reduction in camera views*, MAD achieves stronger robustness on the Third Person A and Third Person B singular views than all the baselines, including the Single Camera baseline that was trained solely on these third person views. This indicates that MAD was able to leverage better representations from All Cameras to achieve higher success rates on these third person views. On the First Person View, MAD outperforms most baselines and achieves similar success

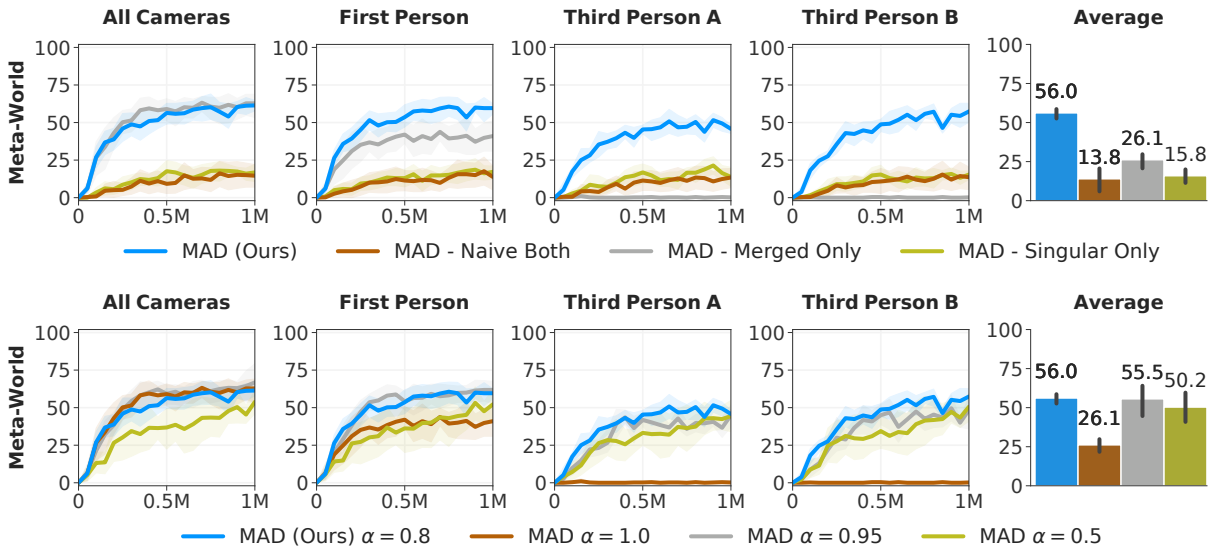


Figure 3.5. **Ablations.** Success rate as a function of environment steps, averaged over 5 Meta-World hard tasks. *(Top)* Component Ablations. *(Bottom)* Alpha Ablations. Methods trained on all camera views and evaluated on all and singular camera views. Mean and 95% CI over 5 random seeds.

rates as the Single Camera and VIB baselines. Overall, MAD is able to achieve consistent success rates whether All Cameras are present, or when only singular camera view inputs are available, indicating a robustness to a *reduction in camera views*. Detailed graphs of all results can be found in Appendix B.4. Extended experiments on increasing views (5 views), and varying input view resolutions are provided in Appendix B.2.

Analysis. Based on our experiments, baselines *fail to achieve similar sample efficiency and robustness to MAD* over the two benchmarks. Our *disentanglement* baseline, MVD, successfully disentangles representations such that it is robust to a reduction in views, but it struggles with poor sample efficiency. This limitation stems from its design approach: rather than leveraging all input views simultaneously, MVD randomly selects individual views to encode into shared and private representations. While this strategy effectively promotes disentanglement, it deliberately avoids creating representations dependent on all views combined — a trade-off that preserves robustness at the cost of efficiency. On the other hand, our *merge* baseline, VIB, appears highly dependent on the first person view, since it applies a variational information bottleneck only

on third person views. Therefore, it achieves high sample efficiency only when the first person view is available, and it fails otherwise. Our third baseline, MV-MWM, is able to disentangle views effectively. However, it uses a heavy auxiliary objective of learning masked multi view representations that hinders its learning speed when no expert demonstrations are provided. In contrast to MVD, MAD is able to leverage all input views combined to increase its learning speed, while augmenting with singular view features to disentangle representations. In contrast to VIB, MAD is agnostic to input perspectives, allowing it to function even without first person view inputs. In contrast to MV-MWM, MAD uses no heavy auxiliary objective that requires expert demonstrations to increase its learning speed. Overall, MAD is able to achieve higher sample efficiency and robustness than MVD, VIB, and MV-MWM.

To measure the *contribution of each component within MAD*, we conduct ablations over 5 hard tasks from Meta-World and plot the results in Figure 3.5. We first ablate different components. In (*MAD - Naive Both*), we train our DrQ baseline naively on both merged and singular view features without using the *MAD loss formulations*. In (*MAD - Merged Only*), we train our DrQ baseline on the merged multi view features only without any *disentanglement*, and in (*MAD - Singular Only*) we train our DrQ baseline on all the singular view features without *merging* them. MAD outperforms all these alternative setups by (29%) success rate, indicating the necessity of each component within our formulation. We further ablate different α values for the loss, where we find $\alpha = 0.8$ to outperform other alpha values, including the original SADA formulation of $\alpha = 0.5$. In summary, each of our design choices appears to be essential for the superior performance of MAD.

Furthermore, we *compare different merge methods* with our merge module. We train DrQ agents on 5 hard tasks from Meta-World using different merge methods for multi view inputs. These merge techniques consist of early, middle, and late merging. For early merging we employ: 1) Frame Stack, where multiple views are stacked across the channel dimension before being passed to the CNN encoder [Mnih et al., 2013, 2015]. For middle merging we include: 2) Concat, where singular views are passed through the encoder and their features are then concatenated

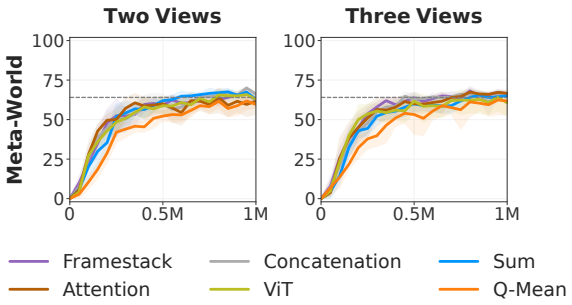


Figure 3.6. **Multi-view Merging.** Success rate as a function of environment steps averaged over 5 Meta-World hard visual RL tasks. Methods were trained and evaluated on *(Left)* two camera views - First and Third A - *(Right)* three camera views. Dashed line indicates highest performance of singular view baselines. Mean and 95% CI over 5 random seeds.

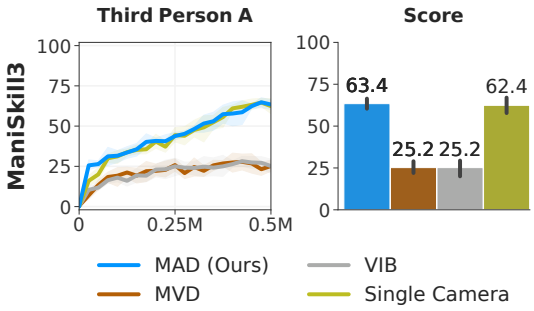


Figure 3.7. **Occlusion Adaptability.** Success rate as a function of environment steps, averaged over 5 ManiSkill3 tasks. Methods are trained on all three views, with two views fully occluded and only Third Person A being observable. Mean and 95% CI over 5 random seeds.

[Hsu et al., 2022] 3) Sum (Ours), where singular view features are summed instead of being concatenated [Lancaster et al., 2024], 4) Attention, where cross attention is used between singular view features [Jangir et al., 2022], 5) ViT, where singular view features are passed through a ViT layer to leverage attention across views [Seo et al., 2023b]. For late merging we add: 6) Q-mean, where we predict a separate Q-value for each view and then average across all views to get the final Q-value [Akinola et al., 2020, Jang et al., 2023]. We train these merge strategies using our DrQ baseline on two views and on three views and display the results in Figure 3.6. The results indicate that all merge methods achieve similar sample efficiency on two and three views. The only merging method that slightly under performs is Q-Mean, which indicates that late merging might not be as effective as other merging strategies for multi view RL.

Adaptability. To test the *adaptability of MAD to occluded or partial views*, we alter the angles of two cameras to face uninformative views and align only one camera, namely Third Person A, to face the robot arm and task. We train all methods using this setup on 5 ManiSkill3 tasks and report the results in Figure 3.7. Our method, MAD, is able to solve the tasks with similar success rate (63%) as the Single Camera (62%), despite it being overloaded with uninformative views. Compared to MAD, neither the MVD nor the VIB baselines seem to

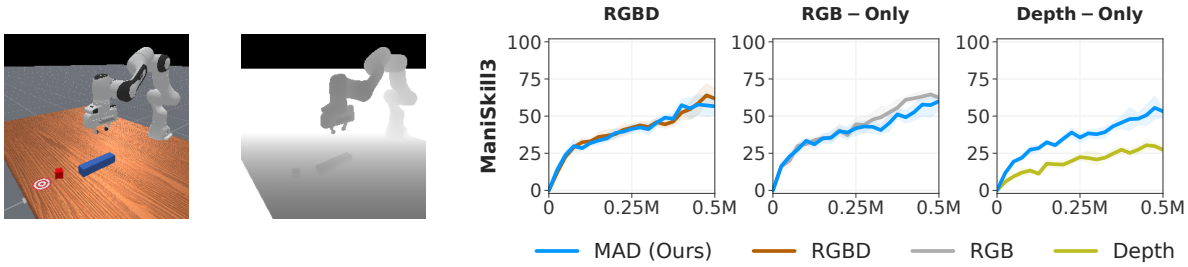


Figure 3.8. **Modality Adaptability.** Success rate as a function of environment steps averaged over 5 ManiSkill3 tasks. *(Left)* RGB and Depth images from the PokeCube Third Person A view. *(Right)* Methods are trained on one camera view (Third Person A) but with different modalities, and evaluated accordingly. Mean and 95% CI over 5 random seeds.

achieve similar sample efficiency. For the MVD baseline, the three views don't have a useful shared representation, since only one view is observant of the task, and thus it fails to solve the tasks effectively. On the other hand, the VIB baseline fails to solve the tasks due to its heavy dependency on the first person camera view.

We further evaluate whether *MAD can be adapted to different modalities* such as depth and RGB as opposed to different camera views, and display the results in Figure 3.8. We train MAD and DrQ agents on 5 ManiSkill3 tasks, with only one camera view input. We pass in RGBD inputs to MAD, and different combinations of RGB and Depth to DrQ agents to measure the baseline performance on that modality. While RGBD input is usually merged by adding depth as an extra channel to the CNN input, we merge it in MAD by summing its features after encoding them separately, for consistency with the MAD framework. Based on the results, MAD is able to achieve similar sample efficiency to the RGBD baseline, while generalizing to both the RGB-Only and Depth-Only modalities. Most surprisingly, MAD is more sample efficient than the Depth baseline on the Depth-Only evaluation, mainly due to it leveraging more informative representations from the RGB inputs. This can prove to be an effective way to train deployable depth-only policies from RGBD inputs.

3.5 Limitations and Conclusion

A limitation of the current study is the absence of real-world experimental validation. Future research could address this gap by implementing MAD on physical robots through simulation-to-real transfer methods [Lin et al., 2025], demonstration-bootstrapped methods [Hu et al., 2023], or enhanced sample efficient visual Q-learning methods [Xu et al., 2023, Hansen et al., 2024, Fujimoto et al., 2025]. Furthermore, generalizing to novel unseen viewpoints in both MAD, and visual RL in general, remains challenging. Future research is needed to expand on the potential of MAD for tackling novel view points, especially with the aid of generative foundational models [Van Hoorick et al., 2024], to deploy robust robot policies to the real world.

To conclude, throughout this work, we identified that the two directions in visual reinforcement learning of: *(1)* merging views and *(2)* disentangling views, can be complementary. We proposed our framework, Merge And Disentangle (**MAD**), which bridges these two directions, benefiting from the merits of each. Concretely in MAD, multi-view inputs are merged for higher sample efficiency from multi-view representations, while singular view features are selectively applied as feature-level augmentations to the downstream components to ensure disentanglement. This produces policies that are robust to a reduction of available cameras, whether in training or deployment. We heavily benchmarked MAD on 20 visual RL tasks from Meta-World and ManiSkill3 against multiple baselines, showcasing its superiority in terms of sample efficiency and robustness. Furthermore, we showed that the application of MAD could be versatile, whether to multiple camera views, or multiple input modalities. Bringing this work forward, we hope this can aid in democratizing visual reinforcement learning, making it more accessible, and serve as a strong baseline for future work in multi view reinforcement learning.

Chapter 3 and Appendix B, in full, are a reprint of the material as it appears from ”Merging and Disentangling Views in Visual Reinforcement Learning for Robotic Manipulation” by Abdulaziz Almuzairee, Rohan Patil, Dwait Bhatt, and Henrik I. Christensen. In Conference on Robot Learning, 2025. The dissertation author was the primary investigator and author of this

paper.

Chapter 4

Visual Efficiency

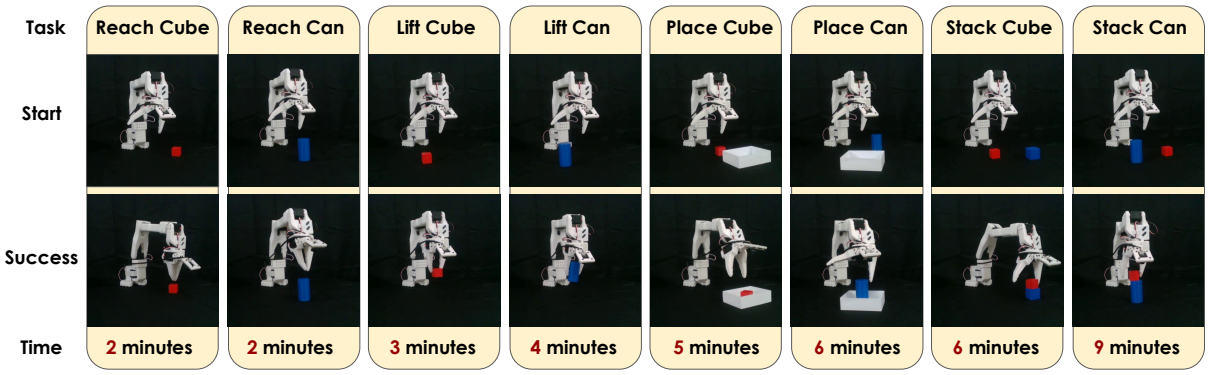


Figure 4.1. **Fast Sim-to-Real Robotics.** We setup 8 visual tasks on the 5 DoF SO-101 Robotics Arm in ManiSkill3 and train our method, *Squint*, purely from images and proprioceptive state on these tasks for 15 minutes. After 15 minutes, we take the final checkpoint, and deploy it zero-shot to our real robot setup. We show visuals of the real robot for each of the tasks above, along with an image of the "Start" frame, the final "Success" frame, and the "Time" it takes our agent to converge in simulation for each task on a single 3090 RTX GPU.

Visual reinforcement learning (RL) [Mnih et al., 2013] offers a compelling paradigm for robotics — enabling deployment onto real robots with camera inputs and requiring no task-specific instrumentation [Kalashnikov et al., 2018, Zhuang et al., 2023, Kaufmann et al., 2023]. Despite this appeal, training visuomotor policies remains notoriously expensive, demanding millions of environment interactions and substantial compute.

Over the past few years, researchers have made strides in reducing the number of environment interactions needed, often referred to as sample efficiency, by improving off-policy

algorithms. These off-policy algorithms, such as SAC [Haarnoja et al., 2018], TD3 [Fujimoto et al., 2018], and TD-MPC2 [Hansen et al., 2024], reuse experiences through a replay buffer. Yet, these improvements come with computational costs that increase training wall-time of these agents in simulation. This slows research iteration. On the other hand, on-policy methods like Proximal Policy Optimization [Schulman et al., 2017] offer poor sample efficiency, without any experience reuse, but can parallelize trivially across thousands of environments on modern GPUs, achieving faster wall-times than their off-policy counterparts. This has led to on-policy dominance in GPU-accelerated simulators and PPO to be the de facto standard for sim-to-real robotics in the past decade [Heess et al., 2017, Hwangbo et al., 2019].

While sample efficiency and wall-time are both important axes for reinforcement learning, recent works like FastTD3 and FastSAC [Seo et al., 2025b,a] explore an intermediate axis, one where off-policy algorithms can be tuned to maximize wall-time rather than sample efficiency, achieving faster wall times than their on-policy counterparts in parallelized simulations. These works made significant progress in scaling off-policy learning for state-based control. Yet, extending these works to visual reinforcement learning remains nontrivial: high-dimensional input images complicate training dynamics, storing them in replay buffers strains memory, and encoding them through convolutional networks adds computational overhead.

Motivated by these challenges, we introduce our method, *Squint*, a visual based Soft Actor Critic method, that through careful image preprocessing, architectural design choices, and hyperparameter selection, is able to leverage parallel environments and experience reuse effectively, achieving faster wall-clock training time than both prior visual off-policy and on-policy methods, and solving visual robotic tasks in minutes.

To train Squint, we leverage ManiSkill3 [Tao et al., 2024], a fast-GPU simulator, with powerful batched rendering capabilities, and create a digital twin of our real robot setup. Our real robot setup consists of a 5 DoF SO-101 Robot Arm [Cadene et al., 2024]. We build 8 distinct tasks, which we term the *SO-101 Task set*, and use them as a testbed for investigating scalable visual reinforcement learning. We train our method and baselines on the SO-101 Task set for 15

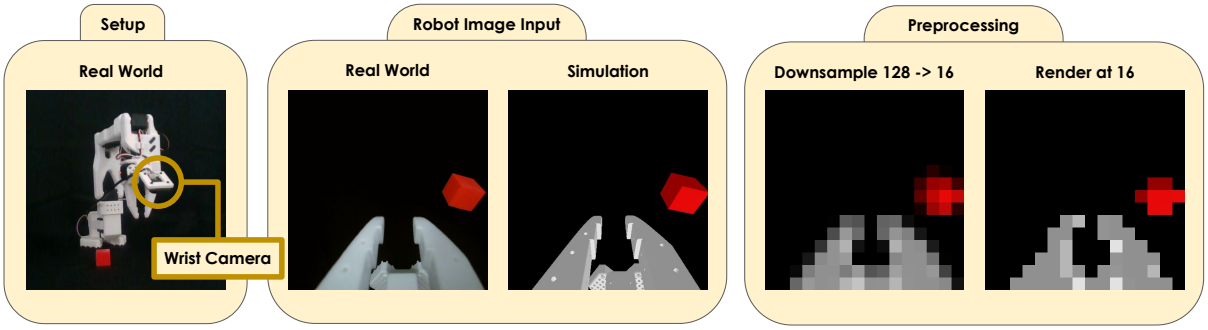


Figure 4.2. **Robot Setup.** *(Left)* Setup of the real robot from an external camera. *(Middle)* The input camera image for the robot in both the real world and simulation. We only use the wrist camera as the input image for the robot. *(Right)* We display different preprocessing effects, downsampling from a higher resolution vs rendering at the target resolution directly.

minutes, and deploy them to our real robot setup, which can be seen in Figure 4.1 and Figure 4.2.

Our key contributions are **(1)** Squint: A fast off-policy visual actor critic method that achieves faster wall-clock training than prior visual RL methods, with extensive experiments justifying each design choice. **(2)** SO-101 Task Set: eight manipulation tasks in ManiSkill3 with heavy domain randomization for sim-to-real transfer. **(3)** Real-world validation: Policies trained in 15 minutes on a single RTX 3090, deployed zero-shot to the real world SO-101 hardware.

4.1 Related Work

Parallel Simulators for Robotics. Simulators provide a fast way for reinforcement learning agents to learn in a safe, constrained environments [Bellemare et al., 2013, Collins et al., 2021]. Early parallelization in CPU allowed agent to train faster in terms of wall time [Heess et al., 2017, Tassa et al., 2018, Kalashnikov et al., 2018, Yu et al., 2020, Zhu et al., 2020]. With the advent of fast GPU-based parallelization, these simulators became faster, and allowed the community to iterate faster on training and deployment. [Liang et al., 2018, Makoviychuk et al., 2021, Mu et al., 2021, Gu et al., 2023, Tao et al., 2024, Sferrazza et al., 2024, Zakka et al., 2025] Most notably, ManiSkill3 [Tao et al., 2024] provides fast GPU-based batched rendering, which allows researchers to train fast visual based reinforcement learning agents for deployment. Due

to its realism, and ease of use, we use ManiSkill3 as our main simulator for this work. Despite its realism, domain randomization [Tobin et al., 2017] is necessary to bridge the sim-to-real gap. Thus, we apply domain randomization to our task set to allow better sim to real transfer.

Visual Reinforcement Learning. Learning from vision has driven the start of deep RL in the recent decade [Mnih et al., 2013, 2015]. Building on it, many works explored different methods to improve off-policy training sample efficiency, including: autoencoder architectures [Finn et al., 2015, Yarats et al., 2019], contrastive learning representations [Srinivas et al., 2020], data regularized representations [Laskin et al., 2020, Yarats et al., 2021a], and model-based representations [Hafner et al., 2023, Hansen et al., 2024, Fujimoto et al., 2025]. Perhaps most notably, DrQ-v2 [Yarats et al., 2021a] is widely used as a solid baseline for visual reinforcement learning sample efficiency. Unfortunately, this sample efficiency comes at the cost of training wall-time, which on-policy PPO [Schulman et al., 2017] exceeds at in scale. While most prior visual off policy learning algorithms focus on sample efficiency, we follow PQL[Li et al., 2023], PQN [Gallici et al., 2024], FastTD3 [Seo et al., 2025b] and FastSAC [Seo et al., 2025a] in focusing on training wall-time, and introduce our algorithm, Squint, to excel at this axis in the visual RL domain.

4.2 Background

Visual Reinforcement Learning formulates the interaction between the agent and the environment as a Partially Observable Markov Decision Process (POMDP) [Kaelbling et al., 1998], defined by a tuple of $\langle \mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{T}, R, \gamma \rangle$. In a POMDP, $s \in \mathcal{S}$ is the state space, $o \in \mathcal{O}$ is the observation space, $a \in \mathcal{A}$ the action space, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the state-to-state transition probability function, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function and γ is the discount factor. At each time step t, the agent receives as input: a single RGB image as an observation $o \in \mathcal{O}$ and *optionally* the proprioceptive state $s^{proprio} \in \mathcal{S}$, and is queried to select an action a . The goal is to learn a policy $\pi : \mathcal{O} \to \mathcal{A}$ that maximizes the expected discounted sum of rewards

$\mathbf{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r_t]$ where $r_t = R(s_t, a_t)$.

Soft Actor Critic (SAC) [Haarnoja et al., 2018] is an off-policy actor critic algorithm, commonly used for continuous control. SAC concurrently learns a Q-function Q_θ (critic) and a policy π_ϕ (actor) with a separate neural network for each. The Q-function Q_θ aims to estimate the optimal soft state-action value function by minimizing the one step Bellman Residual:

$$y = r_t + \gamma\Big(Q_{\bar{\theta}}(\mathbf{s}_{t+1}, \tilde{\mathbf{a}}_{t+1}) - \alpha \log \pi_\phi(\tilde{\mathbf{a}}_{t+1}|\mathbf{s}_{t+1})\Big) \tag{4.1}$$

$$\mathcal{L}_{Q_\theta}(\mathcal{D}) = \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1}) \sim \mathcal{D}} \left[\left(Q_\theta(\mathbf{s}_t, \mathbf{a}_t) - y \right)^2 \right] \tag{4.2}$$

where $\mathcal{D}$ is the replay buffer, $\tilde{\mathbf{a}}_{t+1} \sim \pi_\phi(\cdot|\mathbf{s}_{t+1})$, $Q_{\bar{\theta}}$ represents an exponential moving average of the weights from Q_θ, and α is the temperature parameter [Watkins and Dayan, 1992, Lillicrap et al., 2015, Haarnoja et al., 2018]. SAC also employs clipped double-Q learning [Hasselt et al., 2016, Fujimoto et al., 2018] which we omit from our notation for simplicity. The policy π_ϕ is a stochastic policy that aims to maximize both entropy and Q-values:

$$\mathcal{L}_{\pi_\phi}(\mathcal{D}) = \mathbb{E}_{\mathbf{s}_t \sim \mathcal{D}} \left[\alpha \log \pi_\phi(\tilde{\mathbf{a}}_t|\mathbf{s}_t) - Q_\theta(\mathbf{s}_t, \tilde{\mathbf{a}}_t) \right] \tag{4.3}$$

where $\tilde{\mathbf{a}}_t \sim \pi_\phi(\cdot|\mathbf{s}_t)$. The temperature parameter α can either be fixed or learned by minimizing against a target entropy $\mathcal{H}^{\text{target}}$ [Haarnoja et al., 2018]. Both the critic and actor losses are updated iteratively using stochastic gradient descent in an aim to maximize the expected discounted sum of rewards.

When learning from images, the SAC agent does not have access to state $\mathbf{s}_t$ but rather observation $\mathbf{o}_t$. We follow DrQ and DrQ-v2 [Kostrikov et al., 2020, Yarats et al., 2021a] in encoding $\mathbf{o}_t$ with a single shared CNN encoder f_ψ, such that all previous state inputs $\mathbf{s}_t$ can be replaced with $[f_\psi(\mathbf{o}_t)]$ or $[f_\psi(\mathbf{o}_t), s_t^{proprio}]$ when proprioceptive state is available. The encoder f_ψ is trained end-to-end by the critic gradients.

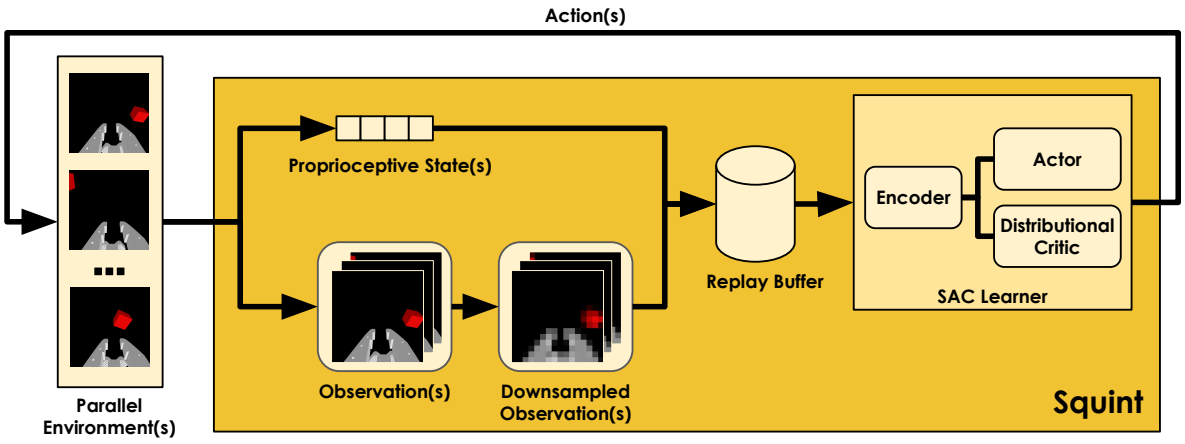


Figure 4.3. **Method Diagram.** Squint is a fast off-policy actor-critic visual reinforcement learning algorithm that accelerates training speed by leveraging parallel environments and downsampling input image observations.

4.3 Method

We present *Squint*, a visual Soft Actor Critic [Haarnoja et al., 2018] method that trains successfully within minutes through: parallel simulation, a distributional critic [Bellemare et al., 2017], resolution squinting, layer normalization [Ba et al., 2016], tuned update-to-data ratio, and an optimized implementation. These trained policies transfer zero-shot to a real-world SO-101 robotic arm. We present the flow of our training pipeline in Figure 4.3. We further show the impact of each design choice in Figure 4.4 and discuss them below:

Update-to-Data Ratio. Tuning the update-to-data (UTD) ratio has long been a tradeoff between sample efficiency and training wall time [D'Oro et al., 2023]. For minimizing training wall time, prior work has shown that one needs to use a very large number of parallel environments with a very low number of updates [Shukla, 2025, Seo et al., 2025b,a], creating a low UTD ratio. In Figure 4.4, we study the impact of these choices, and find that a large number of parallel environments (1024) along with a large number of updates (256) can reduce training time significantly. These findings indicate that while very low UTD ratios (<0.06) work for humanoids [Seo et al., 2025b,a], manipulation domains seem to benefit from larger UTD ratios, seemingly (0.25) in our task set.

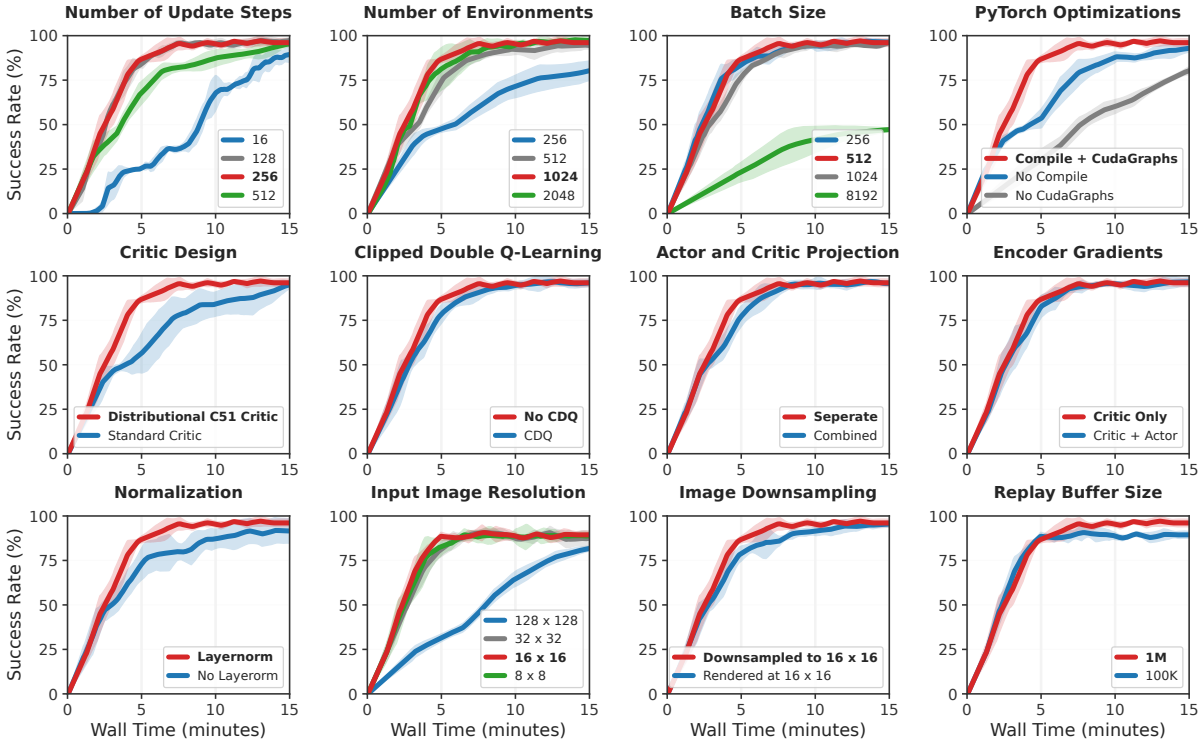


Figure 4.4. **Design Choices.** We experiment with the most critical design choices that allow our agents to be trained while maximizing wall-time for sim-to-real robotics deployment. Mean Success Rate and 95% CI over 5 seeds for all 8 tasks.

Batch Size. While larger batch sizes have been shown to improve training efficiency in Seo et al. [2025b,a], we find that using a lower batch size of 512 provides the same stable training efficiency, while minimizing wall time for every update.

Pytorch Optimizations. Our codebase builds on LeanRL and CleanRL [Huang et al., 2022a]. We add support for both PyTorch compile and cudagraphs [Paszke et al., 2019], which leverage kernel fusion and reduced CPU launch overhead to significantly accelerate the training loop. We further add AMP bfloat16 support for the update loops, reducing training time with convolutional neural networks. Combined, these optimizations give more than a 5x speedup in training speed.

Critic Design Choices. We adopt a Distributional C51 Critic [Bellemare et al., 2017], similar to [Li et al., 2023, Seo et al., 2025b,a], where we minimize cross entropy loss rather than

the mean square error on TD-targets. We find that the choice of a distributional critic greatly improves wall time speed, even though it requires more computation than a standard Critic. We further investigate the effect of Clipped Double Q-learning (CDQ) [Fujimoto et al., 2018] and find a slight improvement in using the average rather than CDQ.

Encoder Design Choices. We leverage a small two layer CNN encoder, and share it between the actor and critic. The encoder is only updated by the critic's TD-loss. After encoding the images, we pass them through separate one layer projections for each of the actor and critic. We adopt these design choices following prior state-of-the-art visual RL methods [Kostrikov et al., 2020, Yarats et al., 2021a], though our ablations reveal they yield negligible performance gains on our task set.

Normalization. All our linear layers are followed by Layer Normalization layers [Ba et al., 2016], which we find to improve training speed, similar to the findings of Ball et al. [2023], Nauman et al. [2024], Seo et al. [2025a].

Input Image Resolution. We compare training agents with larger and lower input image resolutions, and find them to achieve identical sample efficiency, but varying wall time training speed. We only show the wall time axis in Figure 4.4. Our findings show that the smaller the resolution, the faster the wall time, and thus we use a 16 x 16 input image resolution.

Squinting. We compare rendering at 16 x 16 vs rendering at a high resolution of 128 x 128 and then area downsampling to 16 x 16, which we refer to as *squinting*. We find that downsampling improves performance on our task set. This downsampling provides natural anti-aliasing and preserves scene structure, which we hypothesize aids sim-to-real transfer. We further show a visual comparison in Figure 4.2.

Replay Buffer Size. Due to our use of squinting and low image sizes of 16 x 16, we are able to scale replay buffer size to easily fit on the GPU for faster training speed. We compare using 100K vs 1M replay buffer storage size, and find that the 1M improves asymptotic method success rate on our task set by 7%. While we use a simple FIFO buffer, we leave it to future work to use more advanced buffer sampling techniques [Schaul et al., 2015, Hou et al., 2017]

that could further improve wall time.

Simulation Control Frequency. To facilitate faster training times, we train with 10Hz control frequency using a PD Joint Position Target Delta controller. We use large position offset limits, to allow the agent to move faster in simulation.

Real-World Robot Transfer. To transfer trained agents to the real robot, without compromising safety, we scale all actions down by 0.15, and triple the control frequency to 30Hz. While this constitutes a mismatch between simulation and real world control frequency, we find that a higher control frequency in the real world provides faster recovery control for the robot, and provides smoother trajectories.

4.4 Experiments

We setup 8 distinct tasks in ManiSkill3 [Tao et al., 2024] simulator with the 5-DoF SO-101 robotic arm [Cadene et al., 2024], and train our method and baselines on each of these tasks for 15 minutes over 5 seeds. We only use the wrist camera image and proprioceptive state as input. We then deploy the best seed from each method on each task, and evaluate it zero-shot on our real world robot setup. Our real robot setup is shown in Figure 4.2.

Task Design. We design the tasks to be digital twins of the real world setup, thereby reducing the visual gap. We use the wrist camera for learning, as it has been shown to provide better representations than third person cameras in robot manipulation [Hsu et al., 2022, Almuzairee et al., 2025]. For each task in our task set, the objects are randomly placed on the table in a 0.2m x 0.2m xy box around the gripper's starting xy position. The tasks are: *(1) Reach Cube*: robot reaches 2 cm above the cube. *(2) Reach Can*: robot reaches 2 cm above the can. *(3) Lift Cube*: robot lifts the cube and moves to a predefined rest position. *(4) Lift Can*: robot lifts the can and moves to a predefined rest position. *(5) Place Cube*: robot lifts the cube and places it into the container. *(6) Place Can*: robot lifts the can and places it into the container. *(7) Stack Cube*: robot lifts the red cube and stacks it on top of the blue cube. *(8) Stack Can*: robot lifts the cube

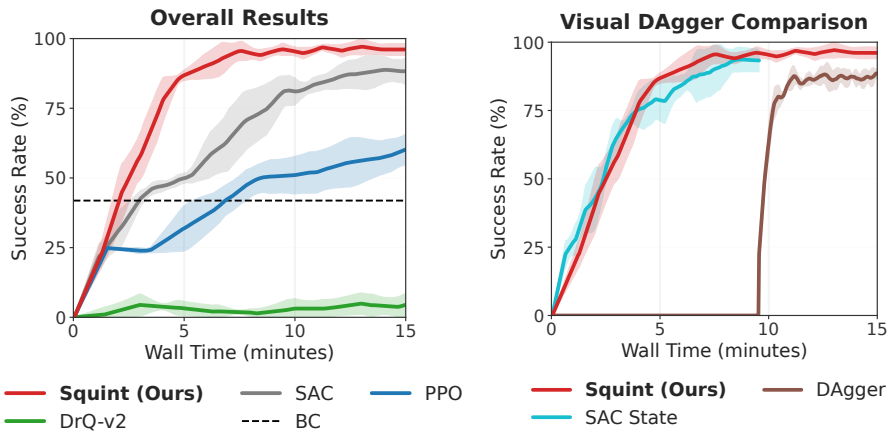


Figure 4.5. **Overall Results.** *(Left)* Comparison with visual RL baselines. *(Right)* Comparison with State-to-Visual DAgger, where we include the time it takes to train an SAC state expert into the comparison. Mean Success Rate and 95% CI over 5 seeds on all tasks.

and stacks it on top of the can. We show the start and success position of each task in Figure 4.1.

Domain Randomization Setup. Even though we design simulation tasks to be similar to the real world setup, domain randomization is necessary to be able to bridge the gap between the simulation and the real world. We apply visual domain randomization consisting of small random position, rotation, and FOV perturbations to the wrist camera, as well as lighting and color jitter alterations. We apply physical domain randomization where we randomize object sizes, object frictions and gripper closing speed. Furthermore, we apply additive isotropic Gaussian noise $(\sigma = 5)$ to joint positions in the proprioceptive state to ensure robustness to any joint position mismatches between simulation and real world setups. We *emphasize* that Squint is able to achieve fast training speed despite these aggressive randomizations.

Baselines. We compare our method, Squint, against the following competitive baselines: (1) **SAC**: an optimized implementation of Soft Actor Critic [Haarnoja et al., 2018] for parallel simulators, with Squint's tuned hyper parameters, but without Squint's architectural differences. (2) **PPO**: an optimized implementation of Proximal Policy Optimization [Schulman et al., 2017], the de facto standard for visual sim-to-real robotics due to its fast wall time training speed with parallel simulators. (3) **DrQ-v2**: Data Regularized Q-Learning, the sample efficiency standard for off-policy visual learning [Yarats et al., 2021a]. We note that we kept DrQ-v2 sequential

with one environment, rather than parallelized, to stay true to its original implementation, and compare our method with the standard for off-policy visual learning. (4) **BC**: Behaviour Cloning [Pomerleau, 1988, Atkeson and Schaal, 1997] from expert demonstrations, commonly used for distilling visual agents. (5) **DAgger**: Dataset Aggregation [Ross et al., 2011], a stronger imitation learning baseline than Behaviour Cloning, commonly used to train visual agents when there is access to the environment and the expert agent. We specifically focus on State-to-Visual DAgger [Ross et al., 2011, Mu et al., 2025], where one trains an expert SAC state based agent and distills it into a visual actor for deployment.

4.5 Results

Overall Results. We display the *overall average* plot in Figure 4.5 and all *detailed results* in Figure 4.6, showcasing per-task breakdowns and the tables for both simulation and real world success rates. Our method, Squint, surpasses all baselines in terms of training wall-time speed, and achieves an average success rate of 96.1% over all tasks at the end of 15 minutes of training. On the real world results, our method achieves 91.3% success rate averaged over 80 trials for 8 tasks, surpassing baselines on the real robot setup.

Table 4.1. Real World Success - Compared with Visual DAgger.

	Squint (Ours)	**Visual DAgger**
Total	73/80	53/80
Average	**91.3%**	66.3%

Comparison with Visual DAgger. We further compare our method with Visual DAgger agents that are distilled from SAC state experts and display the results in Figure 4.5. We include the time it takes to train SAC state experts as done in [Mu et al., 2025]. We find that Squint achieves similar training speed as the SAC state agent, and achieves higher success rate at the end of training than the State-to-Visual DAgger agent on our task set. Furthermore, we deploy

DAgger onto our real robot setup, showcasing the results in Table 4.1, and we find that Squint outperforms DAgger's success rate by 25% in the real world.

Table 4.2. Simulation Success Rates (%)

Task	Squint (Ours)	SAC	PPO	DrQ-v2	BC
Reach Cube	$100.0_{\pm 0.0}$	$99.4_{\pm 1.1}$	$88.0_{\pm 10.3}$	$12.3_{\pm 8.3}$	$61.2_{\pm 9.2}$
Reach Can	$100.0_{\pm 0.0}$	$99.1_{\pm 1.8}$	$96.6_{\pm 2.4}$	$23.6_{\pm 17.6}$	$56.2_{\pm 12.5}$
Lift Cube	$99.8_{\pm 0.4}$	$98.1_{\pm 2.4}$	$92.8_{\pm 4.5}$	$0.0_{\pm 0.0}$	$46.2_{\pm 3.1}$
Lift Can	$98.8_{\pm 2.5}$	$95.4_{\pm 3.5}$	$96.2_{\pm 4.7}$	$0.0_{\pm 0.0}$	$53.8_{\pm 12.9}$
Place Cube	$97.5_{\pm 3.1}$	$99.1_{\pm 1.7}$	$29.0_{\pm 33.8}$	$0.0_{\pm 0.0}$	$38.8_{\pm 8.3}$
Place Can	$96.3_{\pm 3.0}$	$98.6_{\pm 1.8}$	$71.1_{\pm 18.3}$	$0.0_{\pm 0.0}$	$28.8_{\pm 8.5}$
Stack Cube	$95.0_{\pm 6.1}$	$97.7_{\pm 1.6}$	$4.9_{\pm 9.7}$	$0.0_{\pm 0.0}$	$26.2_{\pm 9.2}$
Stack Can	$81.2_{\pm 8.8}$	$18.7_{\pm 23.2}$	$3.0_{\pm 6.1}$	$0.0_{\pm 0.0}$	$23.8_{\pm 13.3}$
All Tasks	**96.1**$_{\pm 1.6}$	$88.3_{\pm 3.4}$	$60.2_{\pm 4.0}$	$4.5_{\pm 2.9}$	$41.9_{\pm 4.2}$

Table 4.3. Real-World Success Rates

Task	Squint (Ours)	SAC	PPO	DrQ-v2	BC
Reach Cube	10/10	10/10	7/10	4/10	8/10
Reach Can	10/10	10/10	10/10	4/10	7/10
Lift Cube	10/10	10/10	6/10	0/10	6/10
Lift Can	9/10	9/10	7/10	0/10	6/10
Place Cube	10/10	8/10	9/10	0/10	7/10
Place Can	10/10	6/10	4/10	0/10	0/10
Stack Cube	8/10	6/10	3/10	0/10	2/10
Stack Can	6/10	6/10	4/10	0/10	2/10
Total	73/80	65/80	50/80	8/80	38/80
All Tasks	**91.3%**	81.3%	62.5%	10.0%	47.5%

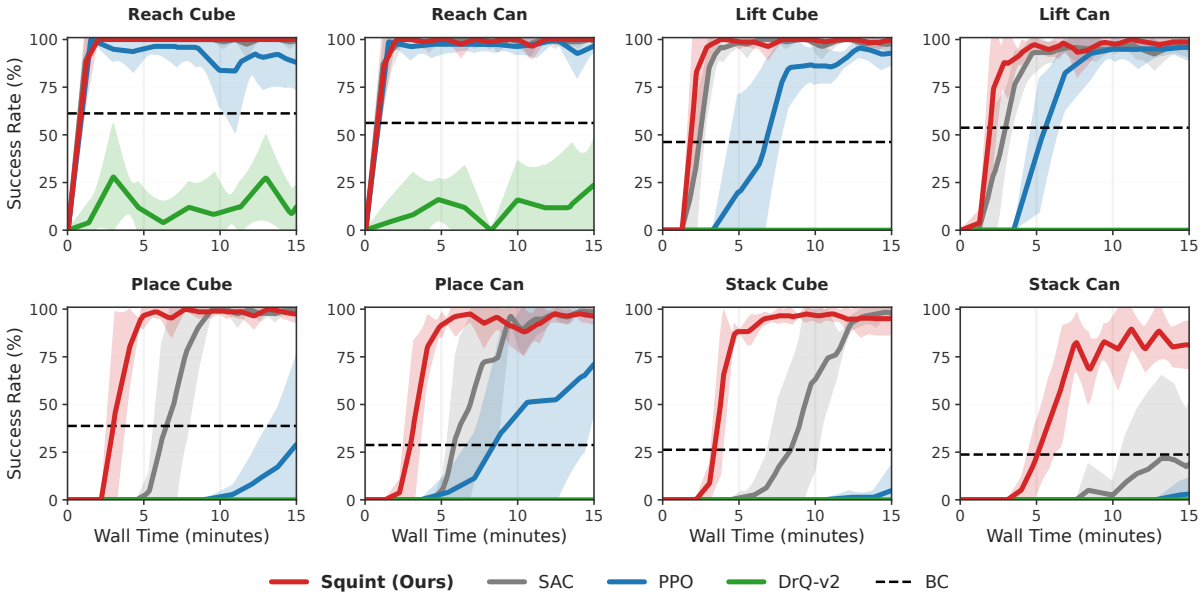
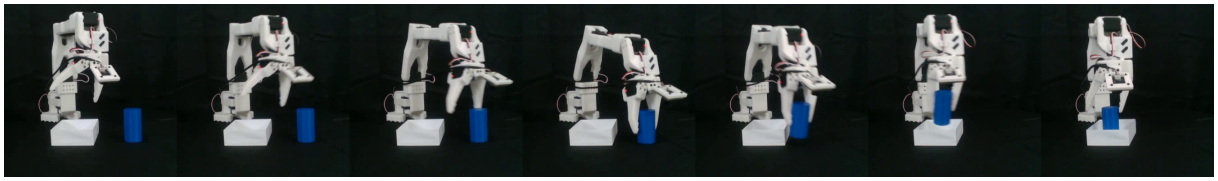
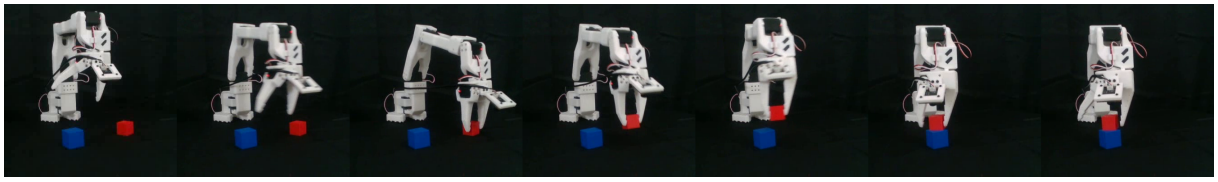


Figure 4.6. **Detailed Results.** We train all methods for 15 minutes on all 8 tasks, and plot the results above. Mean Success Rates and 95% CI over 5 seeds. *(Top Left)* Simulation success rate table of all methods after 15 minutes. *(Top Right)* Real world success rate table over 10 trials for each task. *(Bottom)* Per-task success rate plots in simulation during training.

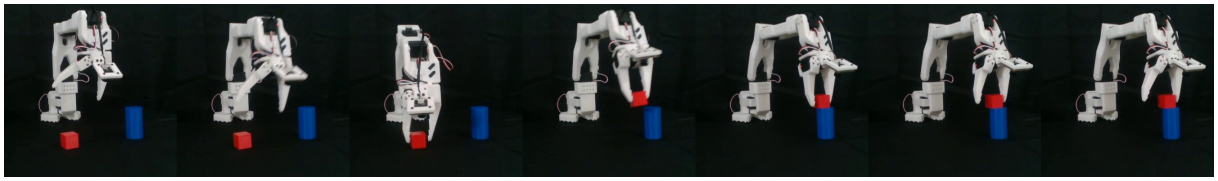
4.6 Discussion



(a) Place Can task.



(b) Stack Cube task.



(c) Stack Can task.

Figure 4.7. **Qualitative Examples.** We show timesteps from left to right, and demonstrate successful deployment on our real world robot.

Quantitative Analysis. Based on simulation and real world results in Figure 4.6, we find that the real world results closely align with the simulation results. This indicates that the simulated environments provide a good test bed for measuring success in the real world. Our SAC baseline outperforms the other baselines, but falls short of Squint, due to its architectural limitations. Our PPO baseline doesn't learn as fast as off-policy methods on the harder tasks of our task set, mainly due to the reuse of data that off-policy algorithms can leverage. Our DrQ-v2 baseline fails to learn, primarily due to it learning from one environment only with heavy randomization, and doesn't benefit from the speed of parallelization. Our BC baseline learns, but is limited in performance due to the state distribution mismatch that DAgger can alleviate. While DAgger learns better, we find that it still falls short of Squint. Both BC and

Table 4.4. Real World Success - Visual Robustness.

	Squint (Ours)	**Squint - No Color Jitter**
Total	73/80	58/80
Average	**91.3%**	72.5%

DAgger share a key limitation in our setting, which is learning from a state-based agent. Our task setup uses wrist cameras, which require active vision, and a different exploration movement than an all-seeing state agent would take. Overall, Squint shows promise as it accelerates wall-time speed of training deployable visual agents to the real world.

Qualitative Analysis. We display sample trajectories of Squint on the real SO-101 robot in Figure 4.7. The three trajectories consist of the three hardest tasks: PlaceCan, StackCube and StackCan.

Visual Robustness Analysis. Despite building a digital twin of the environment in simulation, visual reinforcement learning agents are brittle to changes in visuals that they haven't encountered before [Almuzairee et al., 2024]. We found that even slight lighting changes in the real robot setup were degrading the performance of our deployed agent. We ablate removing color jitter from our environment pipeline and evaluate it on the real robot setup. We display our results in Table 4.4, where we find that our agent's real world performance decreases by 18% on all tasks without the application of color jitter. We leave it to future work to integrate more visual robustness into the training pipeline.

Algorithm 1. Squint: Pseudocode (distributional critic is omitted for simplicity)

Initialize encoder f_ψ, actor π_ϕ, two critics $Q_{\theta_1}, Q_{\theta_2}$, entropy temperature α, replay buffer $\mathcal{B}$

Initialize target critics $Q_{\bar{\theta}_1}, Q_{\bar{\theta}_2}$ with $\bar{\theta}_1 \leftarrow \theta_1$ and $\bar{\theta}_2 \leftarrow \theta_2$

for each environment step t **do**

  Observe $(o_t^{\text{rgb}}, s_t^{\text{proprio}})$ and downsample image $\tilde{o}_t = \textsc{Downsample}(o_t^{\text{rgb}})$

  Sample $a_t \sim \pi_\phi(f_\psi(\tilde{o}_t), s_t^{\text{proprio}})$ and take action a_t

  Observe next $(o_{t+1}^{\text{rgb}}, s_{t+1}^{\text{proprio}})$ and reward r_t

  Store transition $\tau = (\tilde{o}_t, s_t^{\text{proprio}}, a_t, r_t, \textsc{Downsample}(o_{t+1}^{\text{rgb}}), s_{t+1}^{\text{proprio}})$ in $\mathcal{B} \leftarrow \mathcal{B} \cup \{\tau\}$

  for $j = 1$ **to** `num_updates` **do**

    Sample mini-batch $B = \{\tau_k\}_{k=1}^{|B|}$ from $\mathcal{B}$

    Encode observations: $z_t \leftarrow f_\psi(\tilde{o}_t), \quad z_{t+1} \leftarrow f_\psi(\tilde{o}_{t+1})$

    Compute target Q-value via average:

    $\tilde{a}_{t+1} \sim \pi_\phi(\cdot | z_{t+1}, s_{t+1}^{\text{proprio}})$

    $y = r_t + \frac{\gamma}{2} \sum_{i=1}^{2} \Big(Q_{\bar{\theta}_i}(z_{t+1}, s_{t+1}^{\text{proprio}}, \tilde{a}_{t+1}) - \alpha \log \pi_\phi(\tilde{a}_{t+1} | z_{t+1}, s_{t+1}^{\text{proprio}}) \Big)$

    Update critic and encoder:

    $(\psi, \theta_i) \leftarrow (\psi, \theta_i) - \nabla_{(\psi,\theta_i)} \frac{1}{|B|} \sum_{\tau_k \in B} \Big(Q_{\theta_i}(z_t, s_t^{\text{proprio}}, a_t) - y \Big)^2$ for $i \in \{1, 2\}$

    Update entropy temperature:

    $\alpha \leftarrow \alpha - \nabla_\alpha \frac{1}{|B|} \sum_{\tau_k \in B} (\mathcal{H}^{\texttt{target}} - \mathcal{H}(z_t, s_t^{\text{proprio}})) \cdot \alpha$

    if $j \,\%$ `policy_freq` $== 0$ **then**

      Update actor:

      $z_t^{\text{sg}} \leftarrow \textsc{stopgrad}(z_t)$

      $\tilde{a}_t \sim \pi_\phi(\cdot | z_t^{\text{sg}}, s_t^{\text{proprio}})$

      $\phi \leftarrow \phi + \nabla_\phi \frac{1}{2|B|} \sum_{\tau_k \in B} \sum_{i=1}^{2} \Big(Q_{\theta_i}(z_t^{\text{sg}}, s_t^{\text{proprio}}, \tilde{a}_t) - \alpha \log \pi_\phi(\tilde{a}_t | z_t^{\text{sg}}, s_t^{\text{proprio}}) \Big)$

    end if

    Update target critic $\bar{\theta}_i \leftarrow \rho \bar{\theta}_i + (1 - \rho)\theta_i$ for $i \in \{1, 2\}$

  end for

end for

4.7 Conclusion

Limitations and Opportunities. Building on this work, there are several limitations and promising directions that warrant further investigation:

- **Visual Robustness.** Visual reinforcement learning agents remains brittle to visual changes. While most prior works in off-policy learning focus on improving visual robustness under the sample efficiency axis, optimizing visual robustness under the wall-time axis presents an equally compelling objective—whether through visual augmentations [Hansen et al., 2021b, Almuzairee et al., 2024, Teoh et al., 2024, Ma et al., 2025], pretrained encoders [Yuan et al., 2022b, Wang et al., 2023, Brown and Berseth, 2026], or auxiliary representation learning objectives [Batra and Sukhatme, 2024, Echchahed and Castro, 2025].

- **Sample Efficiency.** Numerous advances in off-policy learning have yielded significant gains in sample efficiency [Hansen et al., 2024, Lee et al., 2024, 2025, Fujimoto et al., 2025, Castanyer et al., 2025, Obando-Ceron et al., 2025, Chang et al., 2026]. Incorporating these methods into Squint while maximizing wall-time efficiency is a promising direction.

- **Gripper Design.** Despite the 5 DoF robot's remarkable capability for its price point, sim-to-real transfer for can grasping was challenging due to tipping and insufficient gripper friction. More adhesive [Glick et al., 2018] or alternative grip designs could mitigate this issue.

- **Generalization.** While this work addresses single-task visual RL, natural extensions include multi-task [Espeholt et al., 2018, Xu et al., 2024, Nauman et al., 2025], multi-view [Yuan et al., 2024, Almuzairee et al., 2025, Li et al., 2025, Kim et al., 2025], or multi-agent [Yu et al., 2022] settings as a step towards more generalist agents.

- **Privileged Training.** We formulated Squint as a symmetric visual based reinforcement learning agent for simplicity. Extending Squint to an asymmetric pipeline, where the critic

could leverage privileged information [Pinto et al., 2017, Hu et al., 2024], could further speed up the learning.

- **Dataset Diversity.** Our work extensively evaluated the SO-101 task set in simulation and the real world. Evaluating and improving Squint on larger and diverse visual benchmarks is another promising direction [Tassa et al., 2018, Yu et al., 2020, Hansen et al., 2025].

- **Imitation Learning Limitations.** Imitation learning agents achieve suboptimal policies when given an active vision observation space and asked to imitate a state based agent. Future work could introduce imitation learning agents that address this limitation [Hu et al., 2024, Song et al., 2025].

- **Sim and Real Co-training.** While we train Squint policies purely from simulation, co-training with both parallel simulation and real world demonstrations could prove a powerful way to bootstrap the learning [Maddukuri et al., 2025].

Summary. Fast GPU-based parallel simulators have dramatically lowered the entry barrier for robot learning. In this work, we integrate the 5-DoF SO-101 robotic arm – a low-cost platform – into ManiSkill3, establishing 8 benchmark tasks with extensive domain randomization for sim-to-real transfer. We introduce Squint, an off-policy visual reinforcement learning algorithm that, through careful architectural design, hyperparameter tuning and image preprocessing, achieves superior wall-time efficiency and success rates compared to existing baselines on our proposed SO-101 Task Set. Squint trains on a single RTX 3090 GPU in under 15 minutes, producing policies ready for real-world deployment. We hope this work accelerates iteration cycles in robot learning, makes the field more accessible to researchers, and provides a strong foundation for future advances in visual reinforcement learning.

Chapter 4, in full, is a reprint of the material as it appears from ”Squint: Fast Visual Reinforcement Learning for Sim-to-Real Robotics” by Abdulaziz Almuzairee and Henrik I. Christensen. Under Review, 2026. The dissertation author was the primary investigator and

author of this paper.

Chapter 5

Conclusion

Throughout this thesis, we introduced the motivation behind advancing visual robot learning. In its current form, visual robot learning, particularly through visual reinforcement learning, suffers from poor visual generalization, poor visual robustness, and poor training efficiency that makes it difficult for researchers to adopt it for training deployable robot policies. To democratize visual robot learning, we proposed multiple algorithmic changes that can alleviate these challenges.

In Chapter 2, we introduced a recipe named SADA [Almuzairee et al., 2024], for visual reinforcement learning agents, that can improve visual generalization of visual RL agents to different visual backgrounds. SADA achieves this through a selective application of data augmentation, through diverse image transformations and large image datasets, that allows visual policies to generalize to different photometric and geometric variations of the backgrounds.

In Chapter 3, we proposed a method named MAD [Almuzairee et al., 2025], for visual reinforcement learning agents, that improves the robustness of agents in multi camera setups. Specifically, MAD combines multiple camera or multiple sensors during training to improve training efficiency while carefully disentangling features, such that the policy is robust to failing cameras during deployment. This allows MAD to run with even single camera setups during deployment.

In Chapter 4, we showed how we can improve training efficiency and speed up wall time

training with a novel off-policy visual RL method named Squint [Almuzairee and Christensen, 2026]. Squint achieves this via parallel simulation, resolution squinting, a distributional critic, a tuned update-to-data-ratio, layernorm, and an optimized implementation. In a few minutes, Squint is able to train successful visual RL policies that deploy zero-shot to our real robot setup.

While these lower the barrier of entry for visual robot learning, there are many exciting future directions that can be explored:

- Combining the three algorithms into one comprehensive algorithm would be a natural extension, which would introduce a large method with many beneficial properties for general purpose policies.
- Furthermore, scaling to multi task settings would be an interesting future direction, as multi task settings allow an agent to share latent features across embeddings, and transfer skills across different setups.
- Also, starting from an offline reinforcement learning setup or demonstrations, and then converting to an online visual reinforcement learning setup for finetuning can prove a powerful way for training general purpose visual robot learning policies.

While we mention a few future directions here, we note that there are many other unmentioned interesting directions. By introducing these algorithms, we hope that this further democratizes visual robot learning, making it easier for practitioners to build upon and iterate, and allowing researchers to easily train deployable robust robot policies to the real world.

Appendix A

Visual Generalization: Extended Material

A.1 Setup and Implementation

A.1.1 Hyper-parameters

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

Parameter	Setting
Replay buffer capacity	10^6
Action repeat	2
Frame stack	3
Seed frames	4000
Exploration steps	2000
Mini-batch size	256
Discount γ	0.99
Optimizer	Adam
Learning rate	5×10^{-4}
Agent update frequency	2
Critic Q-function soft-update rate τ	0.01
Features dim.	50
Hidden dim.	1024
Actor log stddev bounds	$[-10, 2]$
Init temperature	0.1
Strong Augmentations	Max Random Shift Pixels: 16×16 Max Random Rotation Degrees: $180°$ Random Overlay Alpha: 0.5

A.1.2 Training and Evaluation Setup

DMControl. Each episode length is set to 1000 environment steps. We train all models for 1M environment steps, evaluating on the training environment every 20,000 environment steps for 10 episodes. Post training, we evaluate trained agents on each level of our test suite for 100 episodes and report our mean episode reward. We consider six tasks defined below:

Table A.2. **DMControl Task List**. Task observations are rgb frames of dimensionality $\mathbb{R}^{(\mathbf{3}\times\mathbf{84}\times\mathbf{84})}$. We use frame stacking of the three most recent rgb frames such that the observation dimensionality becomes $\mathbb{R}^{(\mathbf{9}\times\mathbf{84}\times\mathbf{84})}$. Task difficulty is based on the difficulty classification defined in Yarats et al. [2021b].

Tasks	Action Dim	Difficulty
Walker Walk	6	Easy
Walker Stand	6	Easy
Cheetah Run	6	Medium
Finger Spin	2	Easy
Cartpole Swingup	1	Easy
Cup Catch	2	Easy

Meta-World. Each episode length is set to 200 environment steps. We train all models for 1M environment steps. Every 20,000 environment steps, we evaluate for 50 episodes and report the mean success rate. At the end of training we evaluate on the test environments for 50 episodes as well, and report the mean success rate. We use the same camera setup as Seo et al. [2023a] and consider five tasks defined below:

Table A.3. **Meta-World Task List**. Task observations are rgb frames of dimensionality $\mathbb{R}^{(\mathbf{3}\times\mathbf{84}\times\mathbf{84})}$. We use frame stacking of the three most recent rgb frames such that the observation dimensionality becomes $\mathbb{R}^{(\mathbf{9}\times\mathbf{84}\times\mathbf{84})}$. Task difficulty is based on the difficulty classification defined in Seo et al. [2023a].

Tasks	Action Dim	Difficulty
Door Open	4	Easy
Peg Unplug Side	4	Easy
Sweep Into	4	Medium
Basketball	4	Medium
Push	4	Hard

A.1.3 SAC Based Formulation

In the following section, we formulate our objective in the context of our base algorithm, Soft Actor Critic [Haarnoja et al., 2018], but we stress that these changes are applicable to *any* actor critic framework. The actor update objective for SAC with a learned temperature α thus becomes:

$$\mathcal{L}_{\pi_\phi}^{\textbf{SADA}}(\mathcal{D}) = \mathbb{E}_{\mathbf{o}_t \sim \mathcal{D}}[D_{KL}(\pi_\phi(\cdot|\mathbf{p}_t))||\exp\{\frac{1}{\alpha}Q_\theta(\mathbf{m}_t, \cdot)\})]. \tag{A.1}$$

$$\mathcal{L}_{\alpha}^{\textbf{SADA}}(\mathcal{D}) = \mathbb{E}_{\substack{\mathbf{o}_t \sim \mathcal{D} \\ \mathbf{a_t} \sim \pi_\phi(\cdot|\mathbf{p}_t)}}[-\alpha \log \pi_\phi(\mathbf{a_t}|\mathbf{p}_t) - \alpha\bar{\mathcal{H}}], \tag{A.2}$$

where $\mathbf{p}_t = f_\xi([\mathbf{o}_t, \mathbf{o}_t^{\text{aug}}]_{\text{N}}),\ \mathbf{m}_t = f_\xi([\mathbf{o}_t, \mathbf{o}_t]_{\text{N}}),$ and $\mathbf{o}_t^{\text{aug}} = \text{aug}(\mathbf{o}_t, v_t), v_t \sim V$. We use f_ξ to denote the CNN encoder, and $[\cdot]_{\text{N}}$ to denote concatenation for batch size of dimensionality N where $\mathbf{o}_t, \mathbf{o}_t^{\text{aug}} \in \mathbb{R}^{\text{N}\times\text{C}\times\text{H}\times\text{W}}$. We use aug() as the augmentation operator where we stochastically sample from the augmentation distribution $\mathcal{V}$ and apply it to the input.papers/sada/

On the critic's side, the critic's target prediction is unaltered:

$$q_t^{tgt} = r(\mathbf{o}_t, \mathbf{a_t}) + \gamma \text{max}_{\mathbf{a}'_\mathbf{t}} Q_{\bar{\theta}}(f_\xi(\mathbf{o}_{t+1}), \mathbf{a}') \tag{A.3}$$

while the critic's objective is changed to become:

$$\mathcal{L}_{Q_\theta}^{\textbf{SADA}}(\mathcal{D}) = \mathbb{E}_{\mathbf{o}_t, \mathbf{a_t}, \mathbf{r_t}, \mathbf{o}_{t+1} \sim \mathcal{D}}\Big[\big|\big|Q_\theta(\mathbf{p}_t, \mathbf{a_t}) - \mathbf{y}_t\big|\big|_2\Big] \tag{A.4}$$

where $\mathbf{p}_t = f_\xi([\mathbf{o}_t, \mathbf{o}_t^{\text{aug}}]_{\text{N}}),$ and $\mathbf{y}_t = \big[q_t^{tgt}, q_t^{tgt}\big]_{\text{N}}$.

A.1.4 Pseudocode

Algorithm 2. Generic **SADA** Visual Actor Critic Algorithm
(► naïve augmentation, ► our modifications)

f_ξ, π_ϕ, Q_θ: encoder, policy, and Q-function respectively
T, η, $\mathcal{D}$, τ: training steps, learning rate, data replay buffer, target update rate
$\mathrm{aug}, \mathcal{V}$: choice of strong image augmentation, augmentation distribution

for each timestep $t = 1..T$ **do**
  $\mathbf{a_t} \sim \pi(\cdot|\mathbf{o}_t)$
  $\mathbf{o}_{t+1} \sim p(\cdot|\mathbf{o}_t, \mathbf{a_t})$
  $\mathcal{D} \leftarrow \mathcal{D} \cup (\mathbf{o}_t, \mathbf{a_t}, r(\mathbf{o}_t, \mathbf{a_t}), \mathbf{o}_{t+1})$
  UpdateCritic($\mathcal{D}$)
  UpdateActor($\mathcal{D}$)
procedure UpdateCritic($\mathcal{D}$)
  $\{\mathbf{o}_i, \mathbf{a}_i, r(\mathbf{o}_i, \mathbf{a}_i), \mathbf{o}_{i+1} \mid i = 1...N\} \sim \mathcal{D}$ ▷ Sample batch of transitions
  $\mathbf{o}_i,\ \mathbf{o}_{i+1} = \mathrm{aug}(\mathbf{o}_i, \nu_i),\ \mathrm{aug}(\mathbf{o}_{i+1}, \nu_{i'})\ \nu_i, \nu_{i'} \sim \mathcal{V}$
  $q_i^{tgt} = r(\mathbf{o}_i, \mathbf{a_i}) + \gamma \mathrm{max}_{\mathbf{a_i'}} Q_{\overline{\theta}}(f_\xi(\mathbf{o}_{i+1}), \mathbf{a}')$ ▷ Compute Q-target
  $\mathbf{o}_i^{\mathrm{aug}} = \mathrm{aug}(\mathbf{o}_i, \nu_i),\ \nu_i \sim \mathcal{V}$ ► Apply stochastic data augmentation
  $\mathbf{p}_i = [\mathbf{o}_i, \mathbf{o}_i^{\mathrm{aug}}]_{\mathrm{N}},\ \mathbf{y}_i = \left[q_i^{tgt}, q_i^{tgt}\right]_{\mathrm{N}}$ ► Pack data streams
  $\theta \longleftarrow \theta - \eta \nabla_\theta \mathcal{L}_{Q_\theta}^{\mathbf{SADA}}(\mathbf{p}_i, \mathbf{y}_i)$ ► Update Q-function and encoder
  $\overline{\theta} \longleftarrow (1-\tau)\overline{\theta} + \tau\theta$ ▷ Update target Q-function weights
end procedure
procedure UpdateActor($\mathcal{D}$)
  $\{\mathbf{o}_i, \mathbf{a}_i, r(\mathbf{o}_i, \mathbf{a}_i), \mathbf{o}_{i+1} \mid i = 1...N\} \sim \mathcal{D}$ ▷ Sample batch of transitions
  $\mathbf{o}_i = \mathrm{aug}(\mathbf{o}_i, \nu_i),\ \nu_i \sim \mathcal{V}$
  $\mathbf{o}_i^{\mathrm{aug}} = \mathrm{aug}(\mathbf{o}_i, \nu_i),\ \nu_i \sim \mathcal{V}$ ► Apply stochastic data augmentation
  $\mathbf{p}_i = [\mathbf{o}_i, \mathbf{o}_i^{\mathrm{aug}}]_{\mathrm{N}},\ \mathbf{m}_i = [\mathbf{o}_i, \mathbf{o}_i]_{\mathrm{N}}$ ► Pack data streams
  $\phi \longleftarrow \phi - \eta \nabla_\phi \mathcal{L}_{\pi_\phi}^{\mathbf{SADA}}(\mathbf{p}_i, \mathbf{m}_i)$ ► Update policy
end procedure

A.2 Extended Analysis

A.2.1 Ablations

We ablate all our design choices and show the specific modifications in Figure A.1. We refer to SADA's application of augmentation as selective, where not all inputs are augmented. We use 'naive' to refer to a naive application of augmentation, where all inputs are augmented. We use - to denote no application of augmentation.

Method	Actor Aug	Critic Aug	Avg Geometric	Avg Photometric	Avg All
SADA	Selective	Selective	**690**	**658**	**674**
SADA (Naive Actor Aug)	Naive	Selective	410	581	496
SADA (Naive Critic Aug)	Selective	Naive	358	312	335
SADA (No Critic Aug)	Selective	-	655	432	543
SVEA	-	Selective	232	527	380
DrQ + Aug	Naive	Naive	217	246	231
DrQ	-	-	184	322	253

Figure A.1. **Ablations.** Episode reward on DMC-GB2. Methods trained under all augmentations and averaged across all DMControl tasks. Mean and 95% CI for 5 random seeds.

A.2.2 Distracting Control Suite Results

We train all methods in the DMControl training environments under all strong augmentations, and evaluate them in a zero-shot manner on the Distracting Control Suite. The results are shown below in Figure A.2, where SADA outperforms all baselines using the current augmentations.

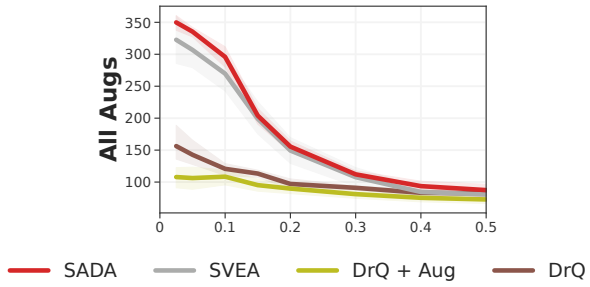


Figure A.2. **Distracting Control Suite.** Episode reward on Distracting Control Suite. Methods trained under all augmentations and averaged across all DMControl tasks. Mean and 95% CI for 5 random seeds.

A.2.3 T-SNE Visualization

We visualize the T-SNE projection of converged SADA and SVEA agents in Figure A.3. Analyzing the graph, we notice a general trend where photometric distributions largely overlap with the training distribution, while geometric distributions seem distant and have little overlap with the training distribution. This asserts the fact presented in Figure 2.1, that the CNN encoder can align the photometric augmentations with the training distribution, such that their latent space is similar. On the other hand, geometric augmentations induce changes in the encoder's output embedding that force it to be placed in seperate latent space.

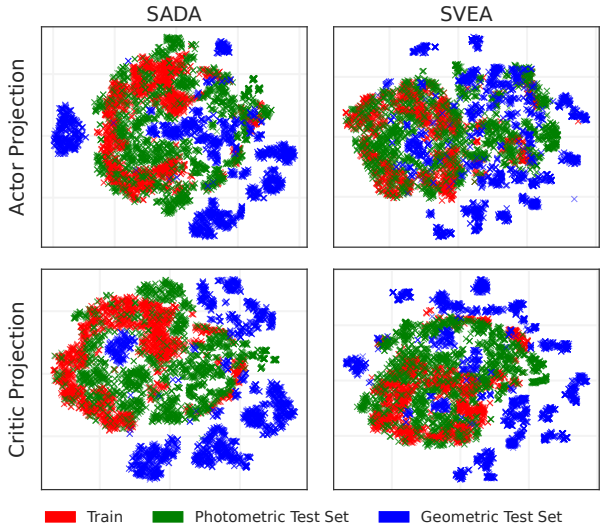


Figure A.3. **T-SNE Embeddings.** We use T-SNE to visualize the projections of converged SADA and SVEA agents trained under all augmentations in the Walker Walk task.

A.3 Visuals

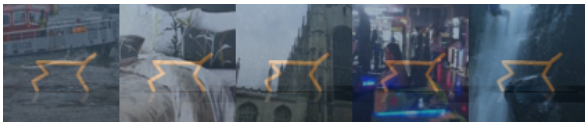
A.3.1 Augmentations

(a) No augmentation (cheetah)

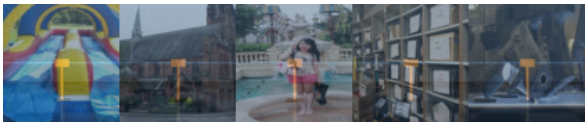
(b) No augmentation (cartpole)

(c) Random convolution (cheetah)

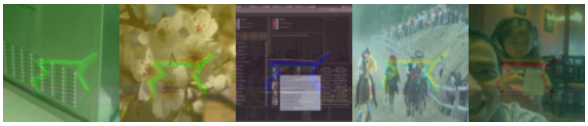
(d) Random convolution (cartpole)



(e) Random overlay (cheetah)

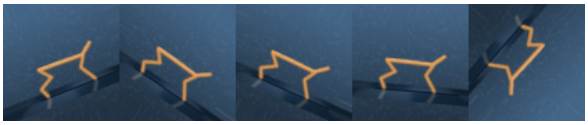


(f) Random overlay (cartpole)



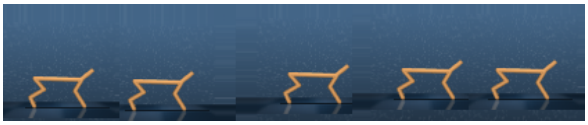
(g) Random convolution and overlay (cheetah)

(h) Random convolution and overlay (cartpole)



(i) Random rotate (cheetah)

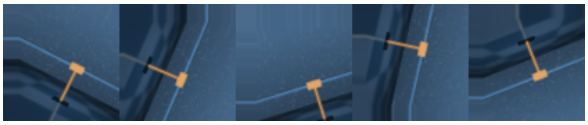
(j) Random rotate (cartpole)



(k) Random shift (cheetah)

(l) Random shift (cartpole)

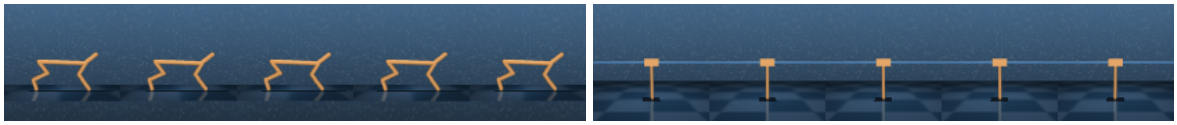
(m) Random rotate and shift (cheetah)



(n) Random rotate and shift (cartpole)

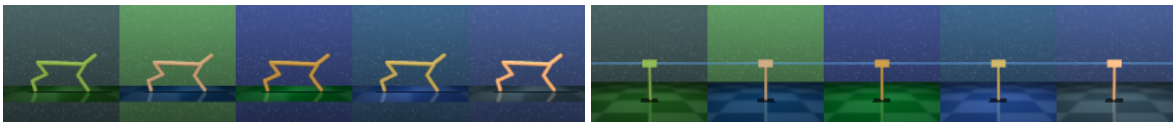
Figure A.4. **Data augmentation**. Visualizations of data augmentations applied in this study. Left column contains samples from the *Cheetah Run* task, and right column contains samples from the *Cartpole Swingup* task. Sets (c)-(h) constitute of photometric augmentations while sets (i)-(n) constitute of geometric augmentations.

A.3.2 DeepMind Control Suite



(a) Training environment (cheetah) *(b)* Training environment (cartpole)

Figure A.5. **DMControl Train environment.** *(Left) Cheetah Run* task. *(Right) Cartpole Swingup* task.

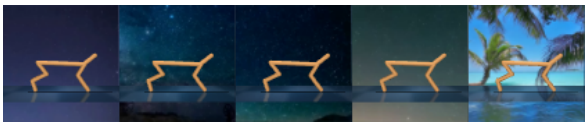


(a) `color_easy` environment (cheetah)

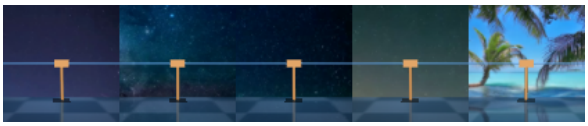
(b) `color_easy` environment (cartpole)

(c) `color_hard` environment (cheetah)

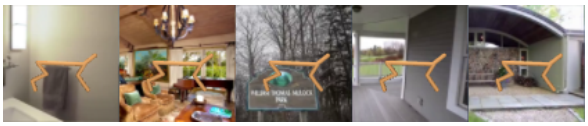
(d) `color_hard` environment (cartpole)



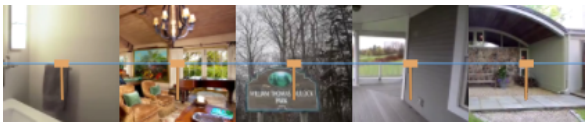
(e) `video_easy` environment (cheetah)



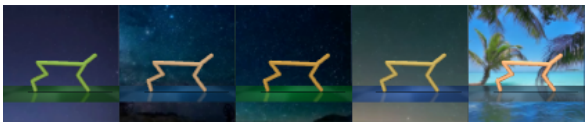
(f) `video_easy` environment (cartpole)



(g) `video_hard` environment (cheetah)

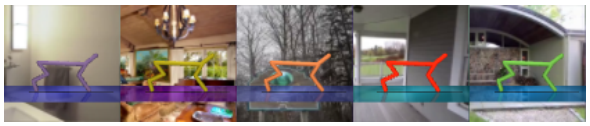


(h) `video_hard` environment (cartpole)



(i) `color_video_easy` environment (cheetah)

(j) `color_video_easy` environment (cartpole)



(k) `color_video_hard` environment (cheetah)

(l) `color_video_hard` environment (cartpole)

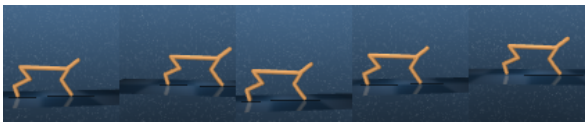
Figure A.6. **DMC-GB2 Photometric Test Set.** Visualizations from the 6 photometric test distributions in DMC-GB2.

(a) `rotate_easy` environment (cheetah)

(b) `rotate_easy` environment (cartpole)

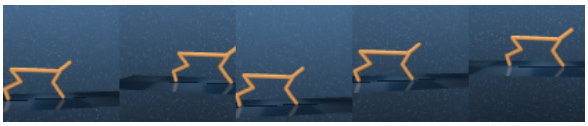
(c) `rotate_hard` environment (cheetah)

(d) `rotate_hard` environment (cartpole)

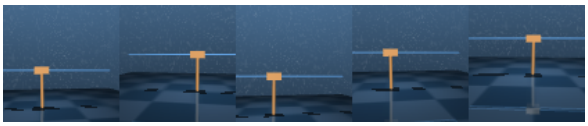


(e) `shift_easy` environment (cheetah)

(f) `shift_easy` environment (cartpole)

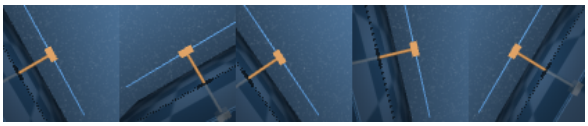


(g) `shift_hard` environment (cheetah)



(h) `shift_hard` environment (cartpole)

(i) `rotate_shift_easy` environment (cheetah)



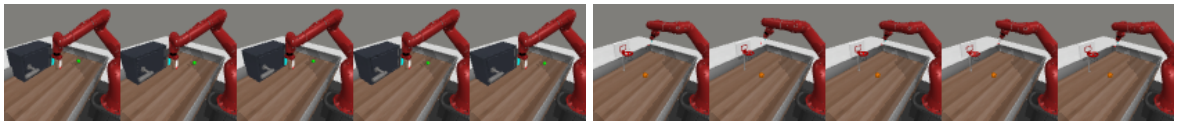
(j) `rotate_shift_easy` environment (cartpole)

(k) `rotate_shift_hard` environment (cheetah)

(l) `rotate_shift_easy` environment (cartpole).

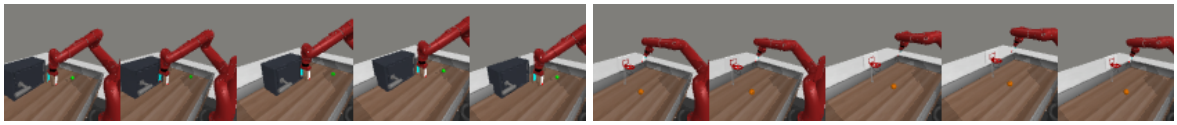
Figure A.7. **DMC-GB2 Geometric Test Set.** Visualizations from the 6 geometric test distributions in DMC-GB2.

A.3.3 Meta-World



(a) Training environment (door-open) *(b)* Training environment (basketball)

Figure A.8. **Meta-World Train environment.** *(Left) Door Open* task. *(Right) Basketball* task.



(a) `shift_hard` environment (door-open) *(b)* `shift_hard` environment (basketball)

Figure A.9. **Meta-World Test environment.** Geometric test distribution in Meta-World.

A.4 Extended Results

A.4.1 DeepMind Control Suite Results

a) **Rotate Easy**

	DrQ	DrQ + Aug	SVEA	SADA
Walker Walk	232±23	166±33	278±21	**808±90** *
Walker Stand	408±24	329±113	505±24	**958±6** *
Cheetah Run	89±10	84±44	127±26	**302±57** *
Finger Spin	116±39	618±80	148±15	**870±152** *
Cartpole Swingup	228±29	219±17	295±23	**743±56** *
Cup Catch	409±45	111±40	408±150	**909±30** *

b) **Rotate Hard**

	DrQ	DrQ + Aug	SVEA	SADA
Walker Walk	133±11	147±22	154±7	**799±89** *
Walker Stand	268±11	288±79	330±20	**960±9** *
Cheetah Run	46±3	86±48	72±16	**290±80** *
Finger Spin	59±20	603±116	75±7	**862±149** *
Cartpole Swingup	178±15	211±19	219±9	**746±57** *
Cup Catch	277±38	107±46	241±76	**908±39** *

c) **Shift Easy**

	DrQ	DrQ + Aug	SVEA	SADA
Walker Walk	63±8	153±32	288±36	**824±95** *
Walker Stand	299±73	307±89	656±53	**962±5** *
Cheetah Run	35±7	104±45	90±28	**348±27** *
Finger Spin	287±84	772±23	386±47	**903±152**
Cartpole Swingup	274±43	212±20	421±80	**798±33** *
Cup Catch	884±77	128±60	771±353	**947±15**

d) **Shift Hard**

	DrQ	DrQ + Aug	SVEA	SADA
Walker Walk	36±2	93±25	58±8	**641±139** *
Walker Stand	161±12	251±59	228±30	**870±38** *
Cheetah Run	11±4	54±30	23±15	**284±26** *
Finger Spin	3±2	573±38	13±15	**802±112** *
Cartpole Swingup	206±31	189±29	284±53	**719±59** *
Cup Catch	676±91	131±50	674±284	**871±62**

e) **Rotate Shift Easy**

	DrQ	DrQ + Aug	SVEA	SADA
Walker Walk	43±5	107±32	102±13	**663±140** *
Walker Stand	196±27	280±83	327±19	**897±30** *
Cheetah Run	12±6	50±24	25±9	**231±44** *
Finger Spin	2±2	381±83	3±2	**732±93** *
Cartpole Swingup	139±28	189±12	195±14	**644±71** *
Cup Catch	353±93	131±64	369±147	**815±70** *

f) **Rotate Shift Hard**

	DrQ	DrQ + Aug	SVEA	SADA
Walker Walk	34±3	57±11	38±4	**307±70** *
Walker Stand	147±10	191±35	180±19	**652±78** *
Cheetah Run	6±2	19±13	13±6	**131±18** *
Finger Spin	1±0	155±61	0±0	**476±46** *
Cartpole Swingup	111±16	172±12	149±12	**497±33** *
Cup Catch	189±52	144±80	204±72	**668±102** *

g) **Color Easy**

	DrQ	DrQ + Aug	SVEA	SADA
Walker Walk	582±47	228±48	755±55	**837±70** *
Walker Stand	826±39	333±103	900±47	**965±10** *
Cheetah Run	**341±42** *	88±39	203±89	252±69
Finger Spin	795±61	693±74	**924±33**	895±162
Cartpole Swingup	696±54	230±28	542±104	**704±33**
Cup Catch	833±37	139±62	821±322	**969±5**

h) **Color Hard**

	DrQ	DrQ + Aug	SVEA	SADA
Walker Walk	265±41	238±44	667±51	**825±72** *
Walker Stand	527±65	355±121	861±60	**963±7** *
Cheetah Run	178±25	87±35	133±73	**239±75**
Finger Spin	466±73	661±76	802±108	**868±150**
Cartpole Swingup	441±43	240±22	478±101	**716±34** *
Cup Catch	520±68	157±66	779±320	**961±11**

i) **Video Easy**

	DrQ	DrQ + Aug	SVEA	SADA
Walker Walk	390±56	132±33	788±103	**791±56**
Walker Stand	603±41	279±63	**945±13**	923±45
Cheetah Run	75±52	49±9	102±56	**121±59**
Finger Spin	441±39	654±88	774±137	**875±157**
Cartpole Swingup	375±54	204±34	427±85	**524±49** *
Cup Catch	523±21	150±45	736±303	**934±23**

j) **Video Hard**

	DrQ	DrQ + Aug	SVEA	SADA
Walker Walk	36±5	166±29	264±57	**270±31**
Walker Stand	154±17	225±47	429±95	**702±65** *
Cheetah Run	25±16	75±20	28±8	**82±20**
Finger Spin	7±4	234±29	263±123	**566±118** *
Cartpole Swingup	98±21	154±26	259±32	**363±31** *
Cup Catch	111±31	152±55	416±252	**662±43** *

k) **Color Video Easy**

	DrQ	DrQ + Aug	SVEA	SADA
Walker Walk	208±49	219±36	681±44	**791±59** *
Walker Stand	487±28	330±105	852±36	**945±15** *
Cheetah Run	60±36	64±16	100±58	**153±64**
Finger Spin	310±30	653±74	705±147	**850±150**
Cartpole Swingup	327±43	217±23	427±86	**570±38** *
Cup Catch	447±61	160±54	716±318	**931±36**

l) **Color Video Hard**

	DrQ	DrQ + Aug	SVEA	SADA
Walker Walk	42±10	215±37	421±67	**686±61** *
Walker Stand	170±17	288±84	659±69	**906±30** *
Cheetah Run	26±17	82±23	44±24	**99±43**
Finger Spin	2±2	365±52	307±139	**633±106** *
Cartpole Swingup	94±17	166±30	294±45	**426±39** *
Cup Catch	122±48	163±77	484±291	**697±37**

Figure A.10. **DMC-GB2 Overall Robustness Results.** Episode Reward. Methods trained under all (geometric and photometric) augmentations and evaluated on the all DMC-GB2 Test Sets. Mean and Stddev over 5 random seeds. Highest scores in bold. Asterisk (*) indicates that the method is statistically significantly greater than all compared methods with 95% confidence.

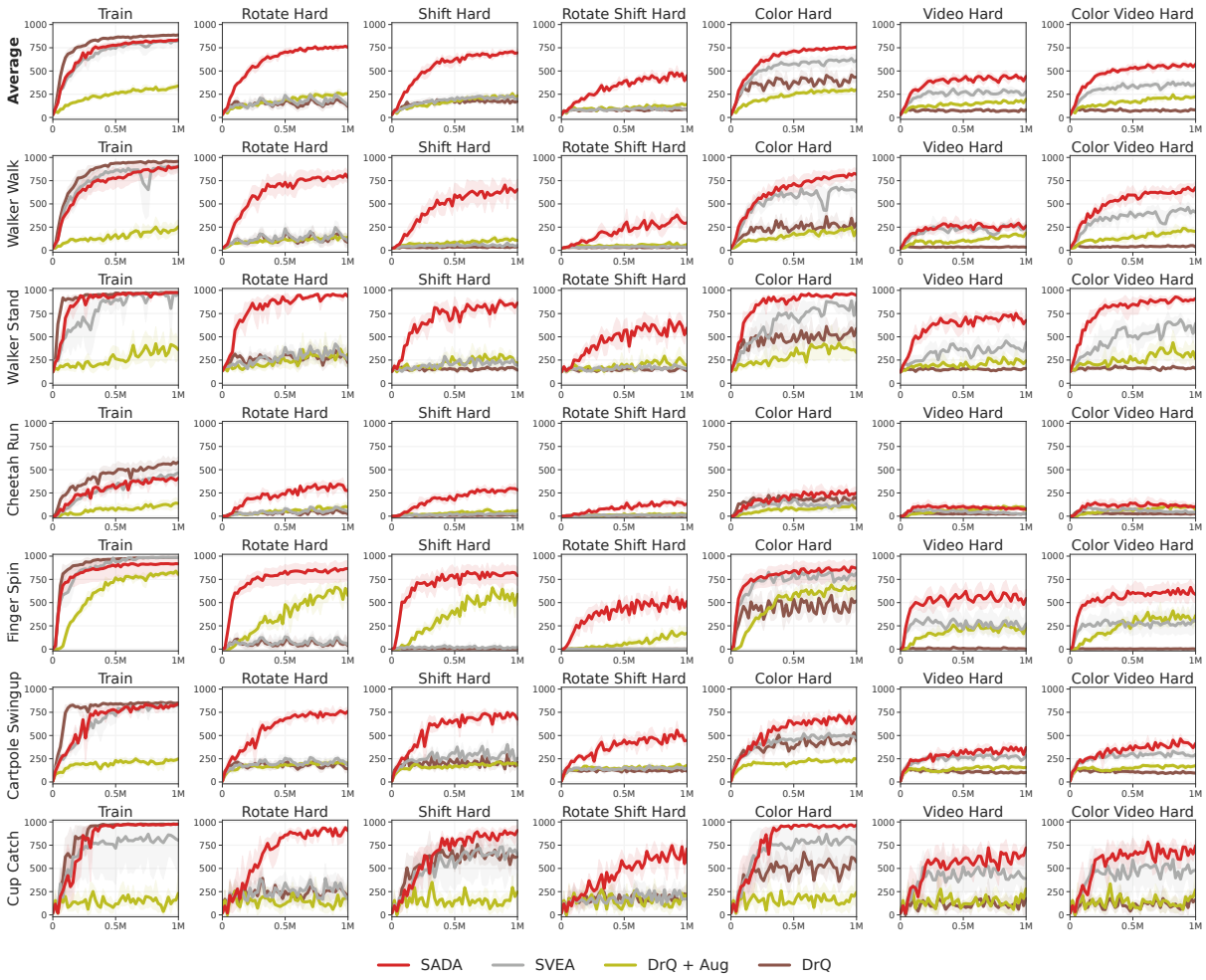


Figure A.11. **DMC-GB2 Overall Robustness Graphs.** Episode Reward. Methods trained under all (geometric and photometric) augmentations and evaluated on all DMC-GB2 Test Sets. Hard levels visualized. Mean and 95% CI over 5 random seeds.

a) **Rotate Easy**

	DrQ	DrQ + Aug	SVEA	SADA
Walker Walk	232±23	431±41	426±37	**728±369**
Walker Stand	408±24	809±159	629±60	**968±5** *
Cheetah Run	89±10	180±33	147±31	**420±76** *
Finger Spin	116±39	829±26	257±58	**885±150**
Cartpole Swingup	228±29	304±77	422±23	**801±54** *
Cup Catch	409±45	600±167	618±86	**803±350**

b) **Rotate Hard**

	DrQ	DrQ + Aug	SVEA	SADA
Walker Walk	133±11	416±45	228±25	**729±367**
Walker Stand	268±11	777±167	406±46	**961±9** *
Cheetah Run	46±3	168±42	86±26	**415±76** *
Finger Spin	59±20	820±28	128±25	**862±158**
Cartpole Swingup	178±15	289±63	280±11	**798±62** *
Cup Catch	277±38	569±173	397±94	**797±352**

c) **Shift Easy**

	DrQ	DrQ + Aug	SVEA	SADA
Walker Walk	63±8	415±61	692±67	**740±374**
Walker Stand	299±73	822±166	765±98	**946±15**
Cheetah Run	35±7	179±22	133±18	**413±72** *
Finger Spin	287±84	678±142	460±85	**899±136** *
Cartpole Swingup	274±43	288±29	564±89	**767±57** *
Cup Catch	884±77	695±137	**940±43**	811±343

d) **Shift Hard**

	DrQ	DrQ + Aug	SVEA	SADA
Walker Walk	36±2	304±83	154±46	**636±324** *
Walker Stand	161±12	671±167	387±70	**897±31** *
Cheetah Run	11±4	129±14	52±18	**344±43** *
Finger Spin	3±2	588±204	90±48	**781±147**
Cartpole Swingup	206±31	216±23	318±33	**634±104** *
Cup Catch	676±91	604±188	**859±77**	790±348

e) **Rotate Shift Easy**

	DrQ	DrQ + Aug	SVEA	SADA
Walker Walk	43±5	316±72	228±17	**678±345** *
Walker Stand	196±27	705±196	489±92	**934±22** *
Cheetah Run	12±6	146±14	56±16	**331±28** *
Finger Spin	2±2	683±169	52±39	**802±147**
Cartpole Swingup	139±28	257±51	269±37	**742±58** *
Cup Catch	353±93	586±156	589±61	**788±347**

f) **Rotate Shift Hard**

	DrQ	DrQ + Aug	SVEA	SADA
Walker Walk	34±3	166±75	62±12	**356±183** *
Walker Stand	147±10	484±189	222±33	**791±41** *
Cheetah Run	6±2	78±13	20±6	**180±57** *
Finger Spin	1±0	513±201	1±1	**663±193**
Cartpole Swingup	111±16	183±28	162±14	**553±94** *
Cup Catch	189±52	512±159	327±49	**749±333**

Figure A.12. **DMC-GB2 Geometric Test Set Results.** Episode Reward. Methods trained under geometric augmentations and evaluated on DMC-GB2 Geometric Test Set. Mean and Stddev over 5 random seeds. Highest scores in bold. Asterisk (*) indicates that the method is statistically significantly greater than all compared methods with 95% confidence.

a) **Color Easy**

	DrQ	DrQ + Aug	SVEA	SADA
Walker Walk	582±47	911±34	841±126	**947±26**
Walker Stand	826±39	964±7	815±341	**975±4**
Cheetah Run	341±42	274±34	348±71	**368±54**
Finger Spin	795±61	948±51	910±142	**983±2**
Cartpole Swingup	696±54	626±152	**843±16**	842±19
Cup Catch	833±37	713±353	**976±2**	973±3

b) **Color Hard**

	DrQ	DrQ + Aug	SVEA	SADA
Walker Walk	265±41	907±31	834±127	**946±23**
Walker Stand	527±65	963±10	813±343	**974±2**
Cheetah Run	178±25	273±37	333±60	**362±48**
Finger Spin	466±73	944±53	882±132	**980±3**
Cartpole Swingup	441±43	627±149	833±15	**843±17**
Cup Catch	520±68	722±339	**974±2**	972±4

c) **Video Easy**

	DrQ	DrQ + Aug	SVEA	SADA
Walker Walk	390±56	885±41	824±143	**936±24**
Walker Stand	603±41	964±5	813±339	**972±2**
Cheetah Run	75±52	264±41	298±40	**340±50**
Finger Spin	441±39	923±41	879±140	**972±4**
Cartpole Swingup	375±54	533±157	**770±44**	749±74
Cup Catch	523±21	690±355	947±16	**961±5**

d) **Video Hard**

	DrQ	DrQ + Aug	SVEA	SADA
Walker Walk	36±5	255±49	243±91	**329±20**
Walker Stand	154±17	669±79	533±203	**692±40**
Cheetah Run	25±16	**151±38**	105±54	91±27
Finger Spin	7±4	600±100	436±106	**735±44** *
Cartpole Swingup	98±21	257±41	387±51	**407±81**
Cup Catch	111±31	518±256	664±48	**816±70** *

e) **Color Video Easy**

	DrQ	DrQ + Aug	SVEA	SADA
Walker Walk	208±49	879±42	817±140	**935±24**
Walker Stand	487±28	963±6	811±341	**970±4**
Cheetah Run	60±36	263±48	294±27	**331±57**
Finger Spin	310±30	920±42	866±137	**972±4**
Cartpole Swingup	327±43	528±154	**761±44**	748±64
Cup Catch	447±61	697±353	944±17	**959±8**

f) **Color Video Hard**

	DrQ	DrQ + Aug	SVEA	SADA
Walker Walk	42±10	639±69	600±150	**736±68**
Walker Stand	170±17	889±48	730±315	**920±22**
Cheetah Run	26±17	**216±53**	153±66	187±63
Finger Spin	2±2	684±82	500±151	**815±25** *
Cartpole Swingup	94±17	300±57	464±63	**469±80**
Cup Catch	122±48	570±300	792±63	**873±43** *

Figure A.13. **DMC-GB2 Photometric Test Set Results.** Episode Reward. Methods trained under photometric augmentations and evaluated on DMC-GB2 Photometric Test Set. Mean and Stddev over 5 random seeds. Highest scores in bold. Asterisk (*) indicates that the method is statistically significantly greater than all compared methods with 95% confidence.

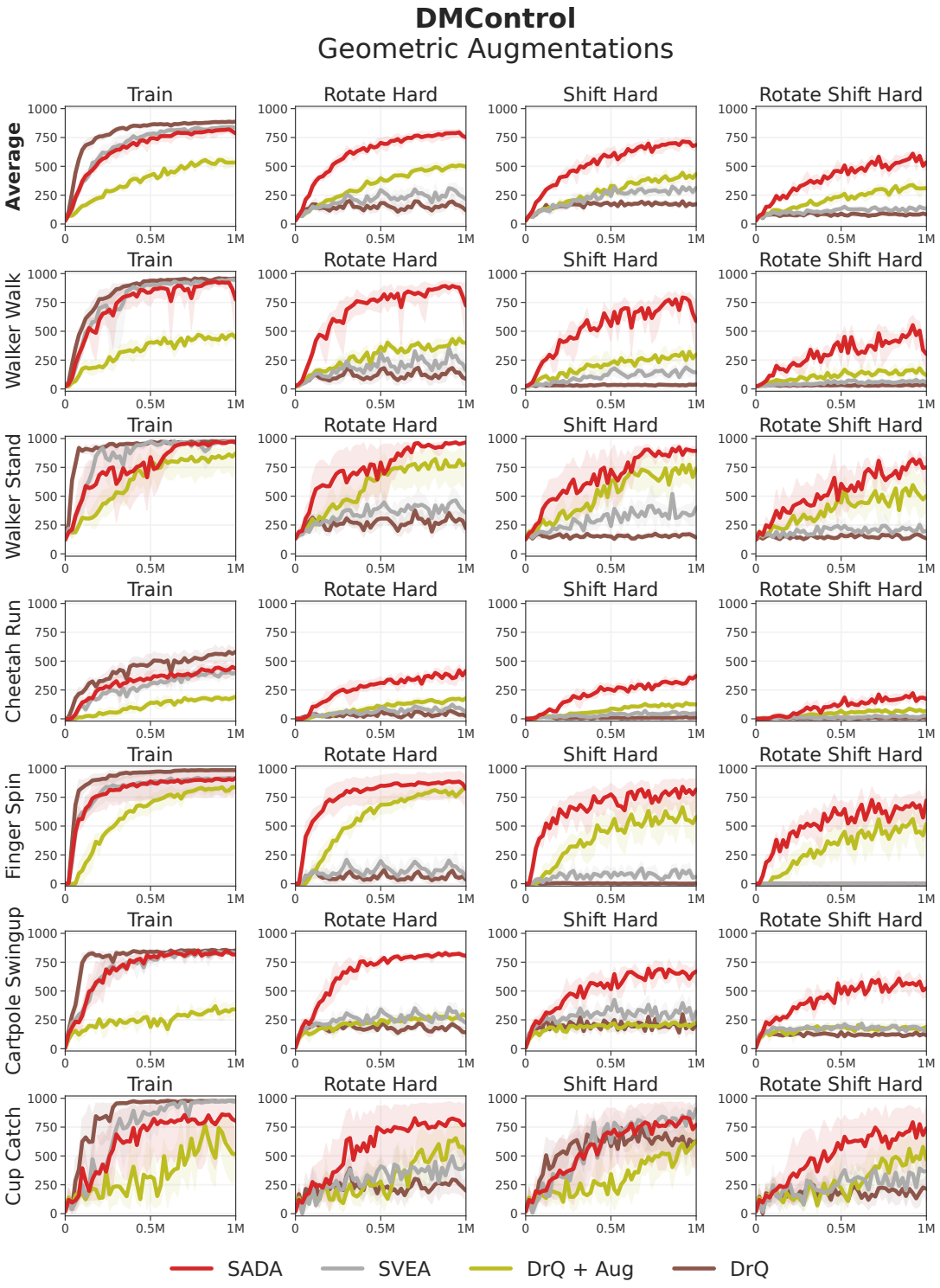


Figure A.14. **DMC-GB2 Geometric Test Set Graphs.** Episode Reward. Methods trained under geometric augmentations and evaluated on DMC-GB2 Geometric Test Set. Hard levels visualized. Mean and 95% CI over 5 random seeds.

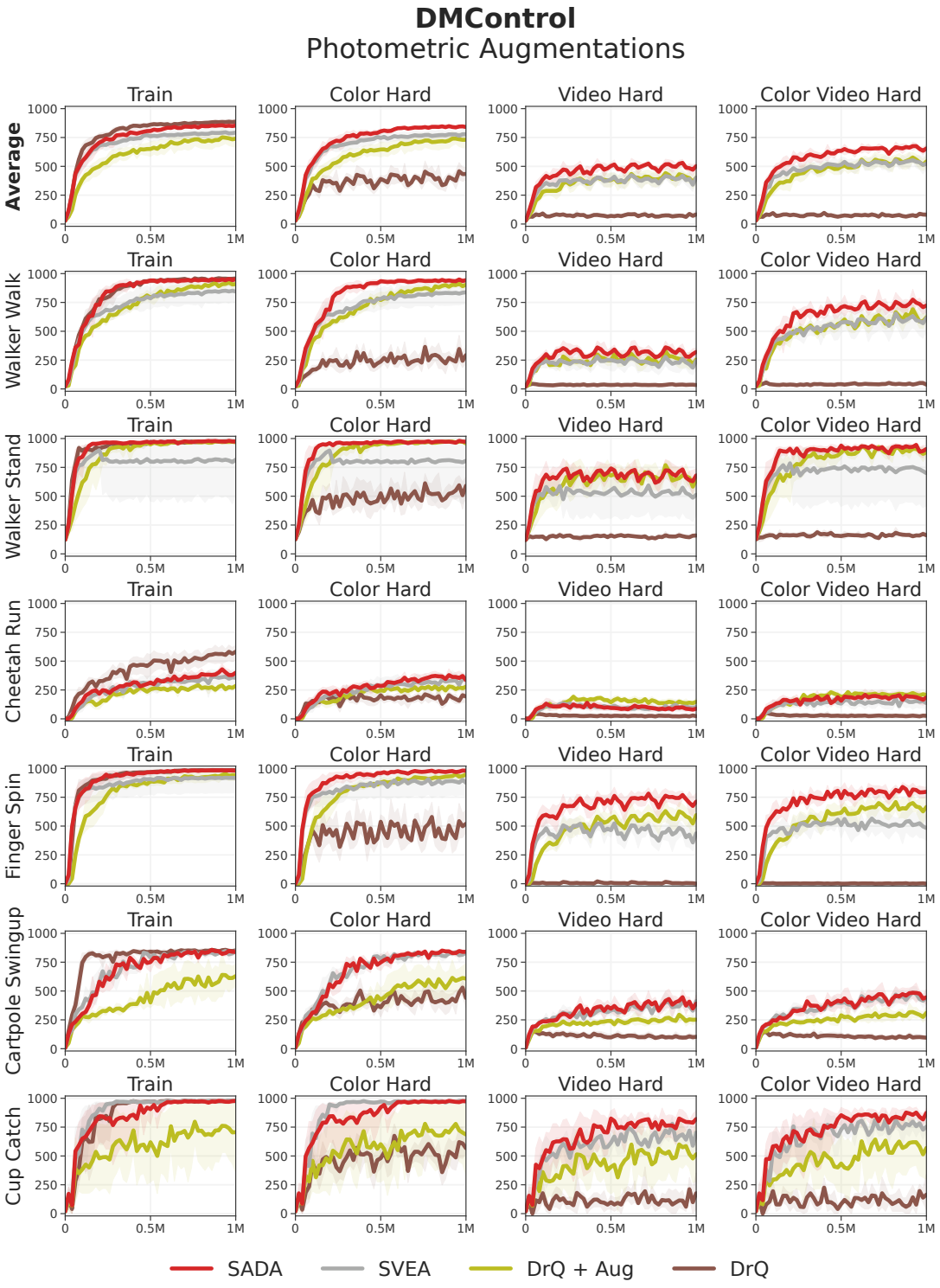


Figure A.15. **DMC-GB2 Photometric Test Set Graphs.** Episode Reward. Methods trained under photometric augmentations and evaluated on DMC-GB2 Photometric Test Set. Hard levels visualized. Mean and 95% CI over 5 random seeds.

A.4.2 Meta-World Results

Shift Hard (Meta-World)

	DrQ	DrQ + Aug	SVEA	SADA
Door Open	2±2	51±12	28±7	**59±9**
Peg Unplug Side	2±1	33±27	32±13	**70±18** *
Sweep Into	3±2	**76±9**	42±8	74±8
Basketball	0±0	48±31	18±16	**75±16**
Push	2±2	43±23	28±4	**61±16**

Figure A.16. **Meta-World Results.** Success rate (%). Trained under strong shift augmentation only. Evaluated on Meta-World Shift Hard. Mean and Stddev of 5 random seeds. Highest scores in bold. Asterisk (*) indicates that the method is statistically significantly greater than all compared methods with 95% confidence.

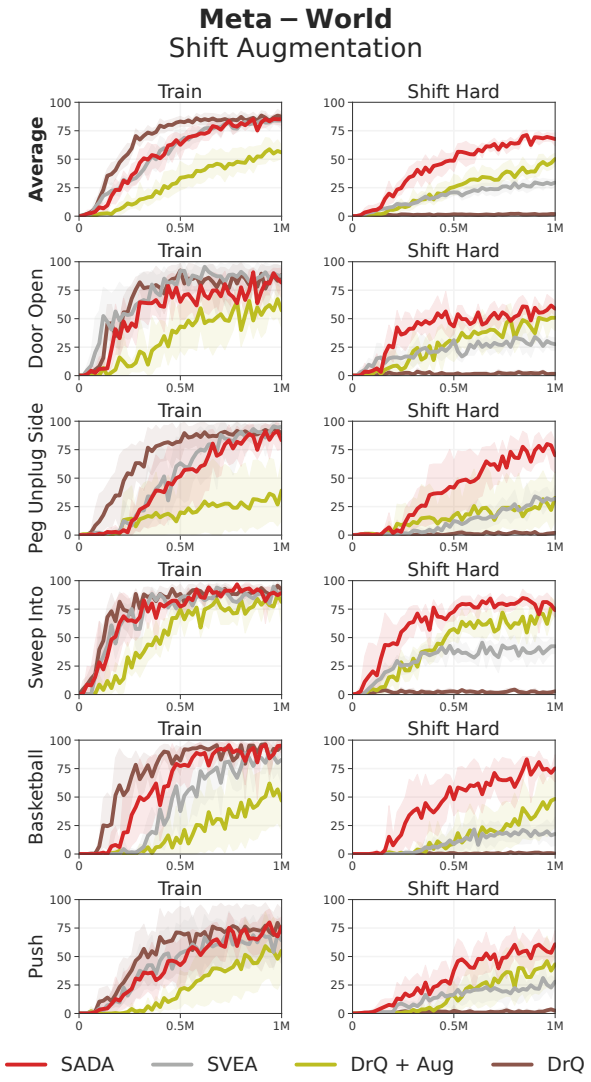


Figure A.17. **Meta-World Graphs.** Success rate (%). Trained under Shift Augmentation, Evaluated on Meta-World Shift Hard. Mean and 95% CI of 5 random seeds.

A.5 Statistical Significance Testing

We conduct statistical significance testing for all our experiments and provide it below. Given two methods $\mathcal{A}$ and $\mathcal{B}$, we use a one tailed Welch t-test to determine the statistical significance and formulate the following hypotheses:

$$\textbf{Null Hypothesis} \quad \mathcal{H}_o\text{: } \mathcal{A} \leq \mathcal{B} \tag{A.5}$$

$$\textbf{Alternative Hypothesis} \quad \mathcal{H}_a\text{: } \mathcal{A} > \mathcal{B} \tag{A.6}$$

Using an alpha value of 0.05 (95% confidence), all p-values greater than 0.05 indicate that the null hypothesis cannot be rejected and that the expected mean of $\mathcal{A}$ is statistically significantly **less than or equal to** the expected mean of $\mathcal{B}$. On the other hand, all p-values less than 0.05 indicate that we should reject the null hypothesis in favor of the alternative hypothesis, indicating that the expected mean of $\mathcal{A}$ is statistically significantly **greater than** the expected mean of $\mathcal{B}$. To control for multiple pairwise comparisons, we apply the Holm-Bonferroni method, where we sort the p-values in ascending order, and compare them with their adjusted alpha values (0.0167, 0.025, 0.05) respectively. Using the Holm-Bonferroni method, there is only a 5% chance of rejecting at least one true null hypothesis (i.e., making a Type I error) from the three hypotheses in every comparison.

We provide per-task statistical significance testing results in the tables in Appendix A.4. We also provide the overall category statistical significance testing results below.

A.5.1 Overall Category Results:

For all the *overall category* results of the experiments conducted throughout this paper, there is sufficient evidence (with 95% confidence) that the mean performance of SADA is statistically significantly greater than all of the baselines.

In the overall category statistical significance testing, we provide both the **p** and **t** values

for the Welch t-test results. **p** denotes the p-value which represents the probability of observing the data or more extreme data under the assumption that the null hypothesis is true. **t** denotes the test statistic which is a standardized measure of the difference between two group means, adjusted for the variability within the groups, used to assess the significance of the observed difference.

Overall Robustness

Method $\mathcal{A}$		**Method $\mathcal{B}$**		
		SVEA	**DrQ + Aug**	**DrQ**
SADA	**Avg Geometric**	p=4.0×10^{-9}, t=27.21	p=8.1×10^{-10}, t=30.62	p=4.0×10^{-8}, t=45.78
	Avg Photometric	p=1.2×10^{-3}, t=5.77	p=1.4×10^{-10}, t=47.60	p=2.9×10^{-10}, t=36.17
	Avg All	p=3.4×10^{-6}, t=15.42	p=1.1×10^{-10}, t=38.84	p=1.9×10^{-9}, t=44.27

Figure A.18. **Overall Robustness.** Statistical Significance Measurement using Welch t-test on the episode reward on DMC-GB2. Methods trained under all augmentations and averaged across all DMControl tasks. Mean over 5 random seeds.

Geometric vs Photometric Robustness

Method $\mathcal{A}$		**Method $\mathcal{B}$**		
		SVEA	**DrQ + Aug**	**DrQ**
SADA	**Avg Geometric**	p=6.2×10^{-5}, t=12.96	p=8.8×10^{-5}, t=7.53	p=2.0×10^{-5}, t=18.28
	Avg Photometric	p=7.3×10^{-3}, t=3.80	p=1.4×10^{-2}, t=3.29	p=1.3×10^{-11}, t=50.40

Figure A.19. **Geometric vs Photometric Robustness.** Statistical Significance Measurement using Welch t-test on the episode reward on DMC-GB2. Methods were trained under geometric augmentations and evaluated on the geometric test set, and trained under photometric augmentations and evaluated on the photometric test set, averaged across all DMControl tasks. Mean over 5 random seeds.

Ablations

Method $\mathcal{A}$		Method $\mathcal{B}$		
		SADA (Naive Actor Aug)	**SADA (Naive Critic Aug)**	**SADA (No Critic Aug)**
SADA	**Avg Geometric**	p=1.3×10^{-7}, t=16.88	p=6.3×10^{-9}, t=23.43	p=1.6×10^{-2}, t=2.61
	Avg Photometric	p=3.2×10^{-3}, t=4.30	p=2.0×10^{-7}, t=22.86	p=2.3×10^{-8}, t=20.58
	Avg All	p=1.1×10^{-5}, t=10.89	p=1.6×10^{-8}, t=24.05	p=1.3×10^{-6}, t=12.01

Figure A.20. **Ablations.** Statistical Significance Measurement using Welch t-test on the episode reward on DMC-GB2. Methods trained under all augmentations and averaged across all DMControl tasks. Mean over 5 random seeds.

TD-MPC2 Baseline

Method $\mathcal{A}$		Method $\mathcal{B}$
		TD-MPC2 + Aug
TD-MPC2 + SADA	**Avg All**	p=8.1×10^{-5}, t=7.90

Figure A.21. **TD-MPC2 Baseline.** Statistical Significance Measurement using Welch t-test on the episode reward on DMC-GB2. Trained under all augmentations with a TD-MPC2 backbone, averaged across all DMControl tasks. Mean over 5 random seeds.

Meta-World

Method $\mathcal{A}$		Method $\mathcal{B}$		
		SVEA	**DrQ + Aug**	**DrQ**
SADA	**Shift Hard**	p=1.5×10^{-5}, t=10.26	p=2.2×10^{-3}, t=3.90	p=1.4×10^{-5}, t=20.45

Figure A.22. **Meta-World.** Statistical Significance Measurement using Welch t-test on the success rate (%) on Shift Hard (Meta-World) distribution. Trained under strong shift augmentation only, averaged across all Meta-World tasks. Mean over 5 random seeds.

Appendix B

Visual Robustness: Extended Material

B.1 Experiment Setup

B.1.1 Hyper-parameters

Table B.1. The default set of hyper-parameters used in our experiments.

Parameter	Setting
Replay buffer capacity	500,000
Image Size	(3, 84, 84)
Frame stack	3
Exploration steps	2000
Mini-batch size	256
Optimizer	Adam
Learning rate	5×10^{-4}
Agent update frequency	2
Critic Q-function soft-update rate τ	0.01
Features dim.	50
Hidden dim.	1024
Actor log stddev bounds	$[-10, 2]$
Init temperature	0.1
MAD Alpha α	0.8
Discount γ	(Meta-World) 0.99
	(ManiSkill3) 0.8

B.1.2 Baseline Implementations

For all baselines, we reimplement them on top of our DrQ baseline for a fair comparison. The MV-MWM baseline however was kept using its model-based baseline, as DrQ is model-free. Nevertheless, MV-MWM was tuned accordingly on both benchmarks.

B.1.3 Environment Setups

Meta-World. We evaluate on 15 tasks where 5 tasks are chosen from each of medium, hard, and very hard difficulties, defined in Seo et al. [2023a]. Every 25,000 environment steps, we evaluate for 20 episodes and report the mean success rate.

MetaWorld-v2 Tasks

Task	Difficulty	Action Dim	Proprioceptive State Dim
Basketball	Medium	4	4
Hammer	Medium	4	4
Peg Insert Side	Medium	4	4
Soccer	Medium	4	4
Sweep Into	Medium	4	4
Assembly	Hard	4	4
Hand Insert	Hard	4	4
Pick Out Of Hole	Hard	4	4
Pick Place	Hard	4	4
Push	Hard	4	4
Shelf Place	Very Hard	4	4
Disassemble	Very Hard	4	4
Stick Pull	Very Hard	4	4
Stick Push	Very Hard	4	4
Pick Place Wall	Very Hard	4	4

ManiSkill3. We evaluate on 5 visual tasks from ManiSkill3. Every 20,000 environment steps, we evaluate for 20 episodes and report the mean success rate. We alter the default camera setups of the tasks to accommodate our setup. For both PushCube and PokeCube tasks, we alter the state to include the goal position of the cube to keep consistent with other tasks in the ManiSkill3 environments.

ManiSkill3 Tasks

Tasks	Difficulty	Action Dim	Proprioceptive State Dim
Push Cube	-	4	28
Pull Cube	-	4	28
Pick Cube	-	4	29
Place Sphere	-	4	29
Poke Cube	-	4	28

Environment Settings

	Environment Steps	Episode Length	Action Repeat	Metric	Number of Eval Episodes
Meta-World	1M	200	2	Success	20
ManiSkill3	0.5M	50	1	Success	20

B.2 Camera Scalability Experiments

B.2.1 Increased Camera Views

We further evaluate how our method would scale with larger number of views. We setup ManiSkill3 with five views as outlined below and evaluate our method and baselines on 5 ManiSkill3 tasks. As shown below, MAD outperforms all baselines by (48%) success rate when increased to 5 views.

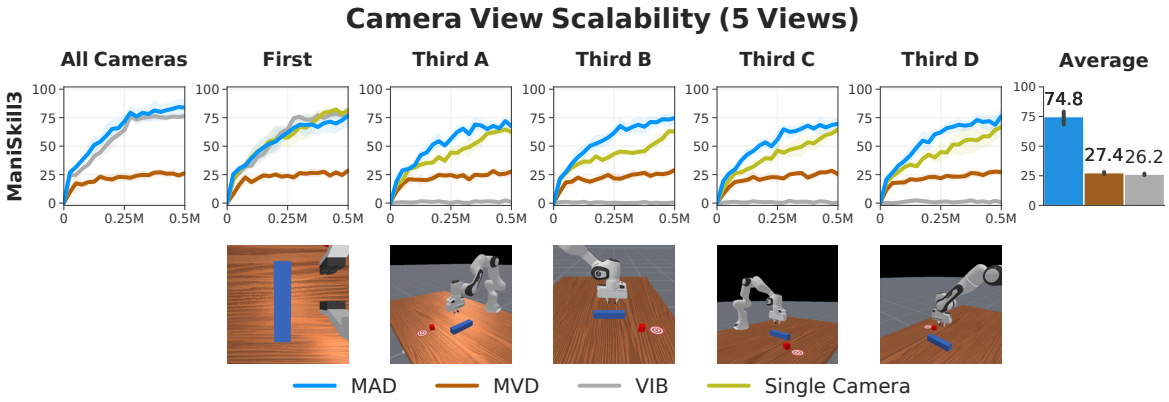


Figure B.1. **View Robustness.** Success rate as a function of environment steps, averaged over 5 ManiSkill3 visual RL tasks. Methods are trained on all five camera views and evaluated on all and singular camera views with the final average displayed on the far right. Mean and 95% CI over 5 random seeds.

B.2.2 Different Input Camera View Sizes

We also evaluate the case where we have multiple camera inputs with different resolutions. If we have multiple cameras with different resolutions, some options to process them are:

(a) As a preprocessing step, resize all images to a fixed size.

(b) Use separate CNN encoders for each image size. This would mean that the input views need to be ordered, such that it is possible to assign each input view to its corresponding CNN encoder.

We opt for option (a) due to its simplicity. Given a First Person View of size (64x64), a Third Person A View of size (96x96) and a Third Person B View of size (84x84), we resize

all input images to (84x84) and train MAD on 5 ManiSkill3 tasks to empirically validate our suggestions. We display the results in the figure below, where we find that MAD is able to support different sizes of camera views with a similar performance to our original setup.

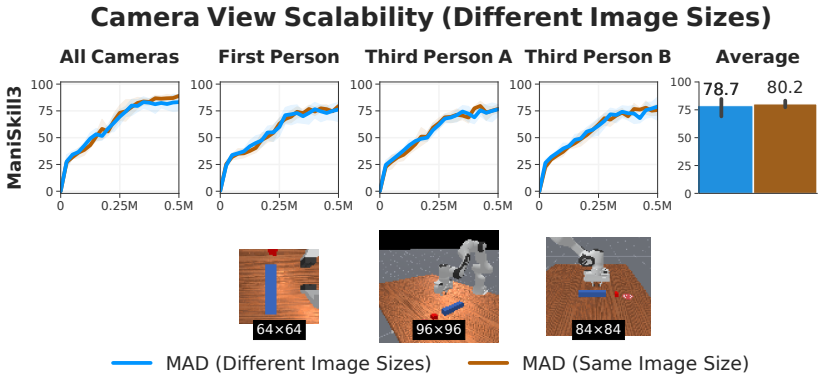


Figure B.2. **View Robustness.** Success rate as a function of environment steps, averaged over 5 ManiSkill3 visual RL tasks. Methods are trained on all three camera views and evaluated on all and singular camera views with the final average displayed on the far right. Mean and 95% CI over 5 random seeds.

B.3 Extended Analysis

B.3.1 Best Camera View for Learning

Is there a single camera view that consistently outperforms other camera views? Based on our experiments, *there is a singular view that outperforms other singular views*, and that being the First Person view. This observation aligns with findings from Hsu et al. [2022]. Due to the first person camera's attachment to the robot arm, and active perception system, it provides the best representations for robot learning. However, its attachment to the robot arm can be a double edged sword, as it can suffer from limited observability that prevents it from solving the task. We shed light on the PokeCube task from ManiSkill3 in Figure B.3, where the First Person view isn't sufficient to solve the task as it has limited observability. Nevertheless, MAD is able to leverage useful information from other third person views and achieve high success rate on All Cameras. In the case of a reduction in views, MAD can still maintain the highest possible success rate on the First Person view given its limited observability.

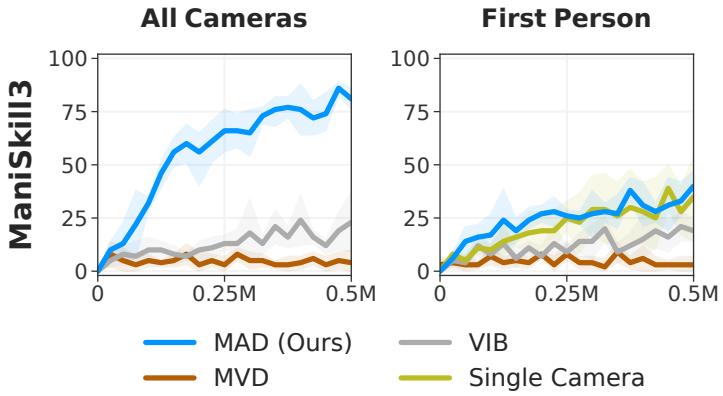


Figure B.3. **First Person Camera View Failure.** Success rate as a function of environment steps on the ManiSkill3 PokeCube task. Methods are trained on all three views and evaluated an all and singular views separately. Mean and 95% CI over 5 random seeds.

B.4 Extended Results

B.4.1 Overall Robustness

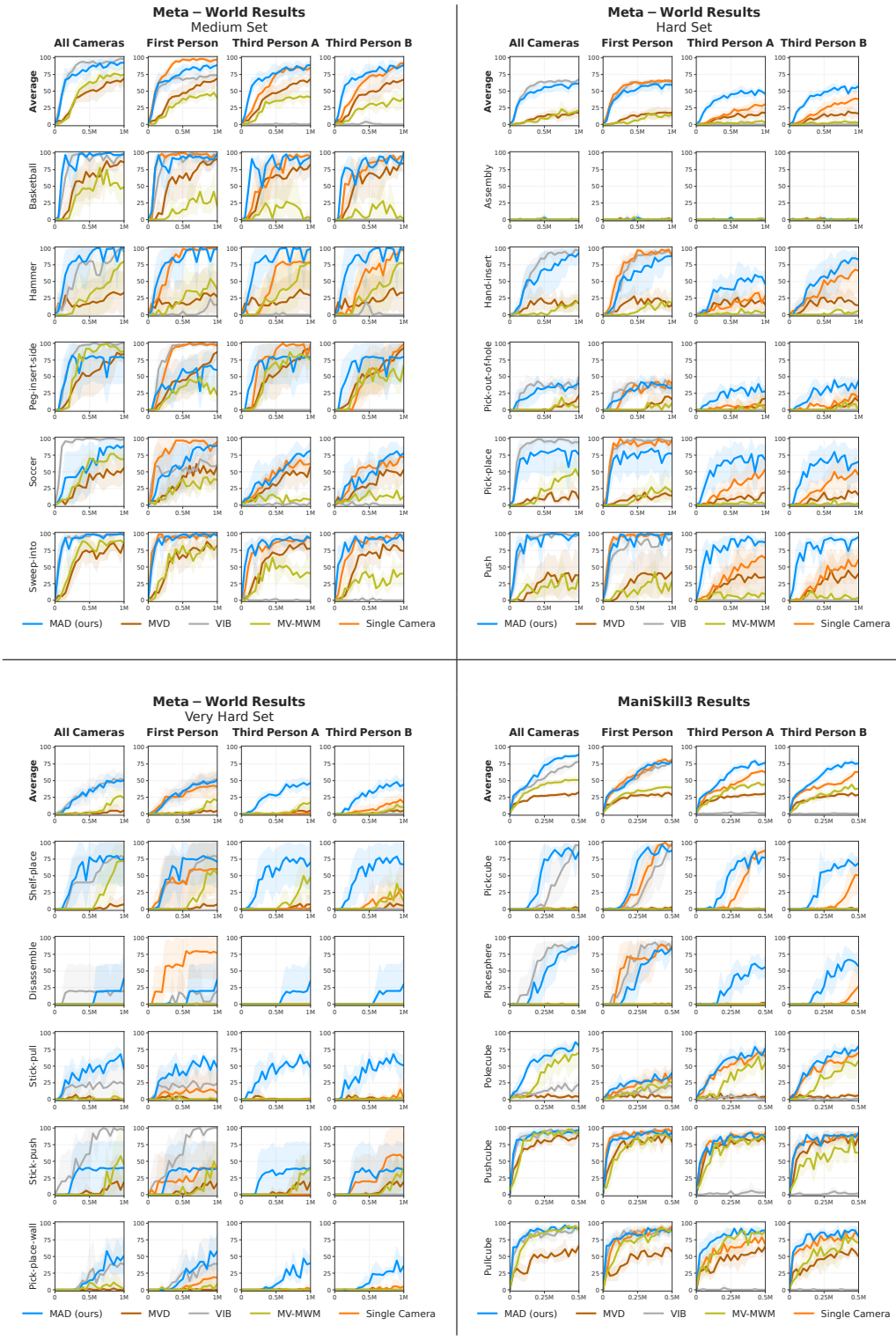


Figure B.4. **Overall Robustness.** Success rate as a function of environment steps, averaged over 15 Meta-World and 5 ManiSkill3 visual RL tasks. *(Top Left)* Meta-World Medium. *(Top Right)* Meta-World Hard. *(Bottom Left)* Meta-World Very Hard. *(Bottom Right)* ManiSkill3. Methods are trained on all three camera views and evaluated on all and singular camera views. Mean and 95% CI over 5 random seeds. In reference to Figure 3.4 in the main text.

B.4.2 Ablations

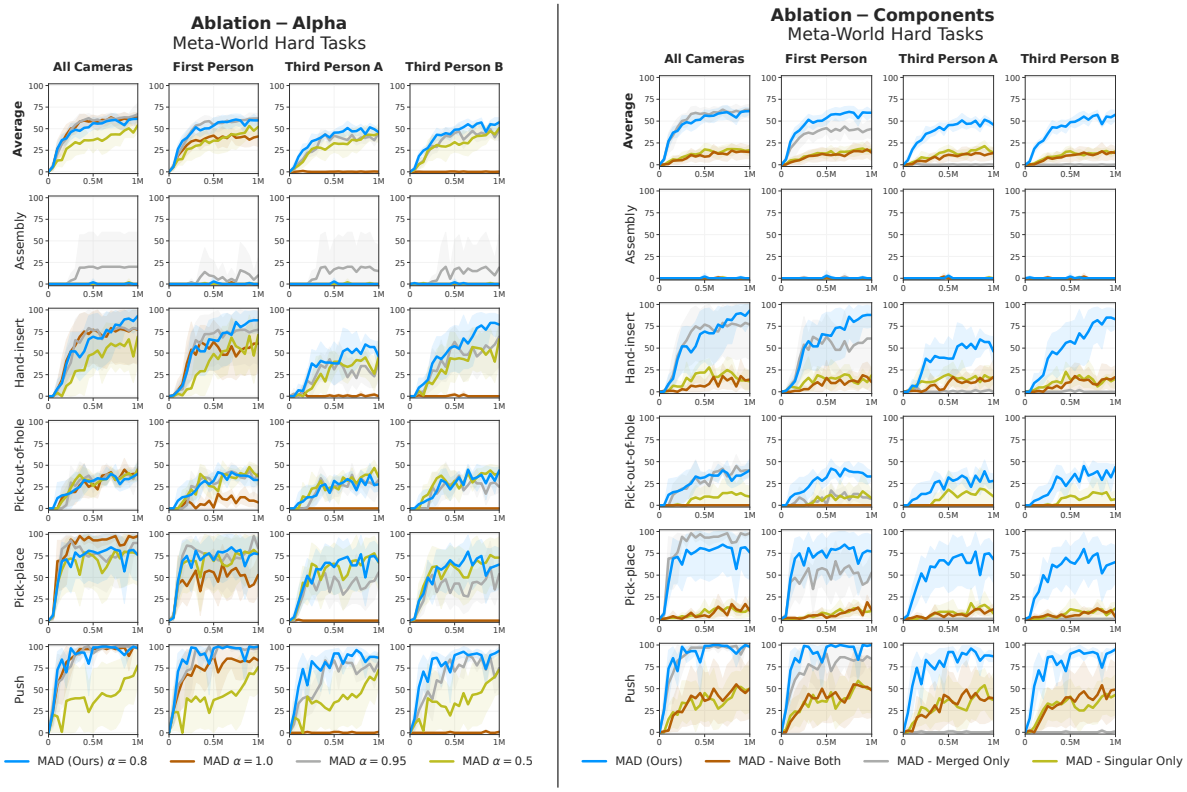


Figure B.5. **Ablations.** Success rate as a function of environment steps, averaged over 5 Meta-World hard visual RL tasks. *(Left)* Alpha Ablations. *(Right)* Component Ablations. Methods are trained on all three camera views and evaluated on all and singular camera views. Mean and 95% CI over 5 random seeds. In reference to Figure 3.5 in the main text.

B.4.3 Multi-View Merging

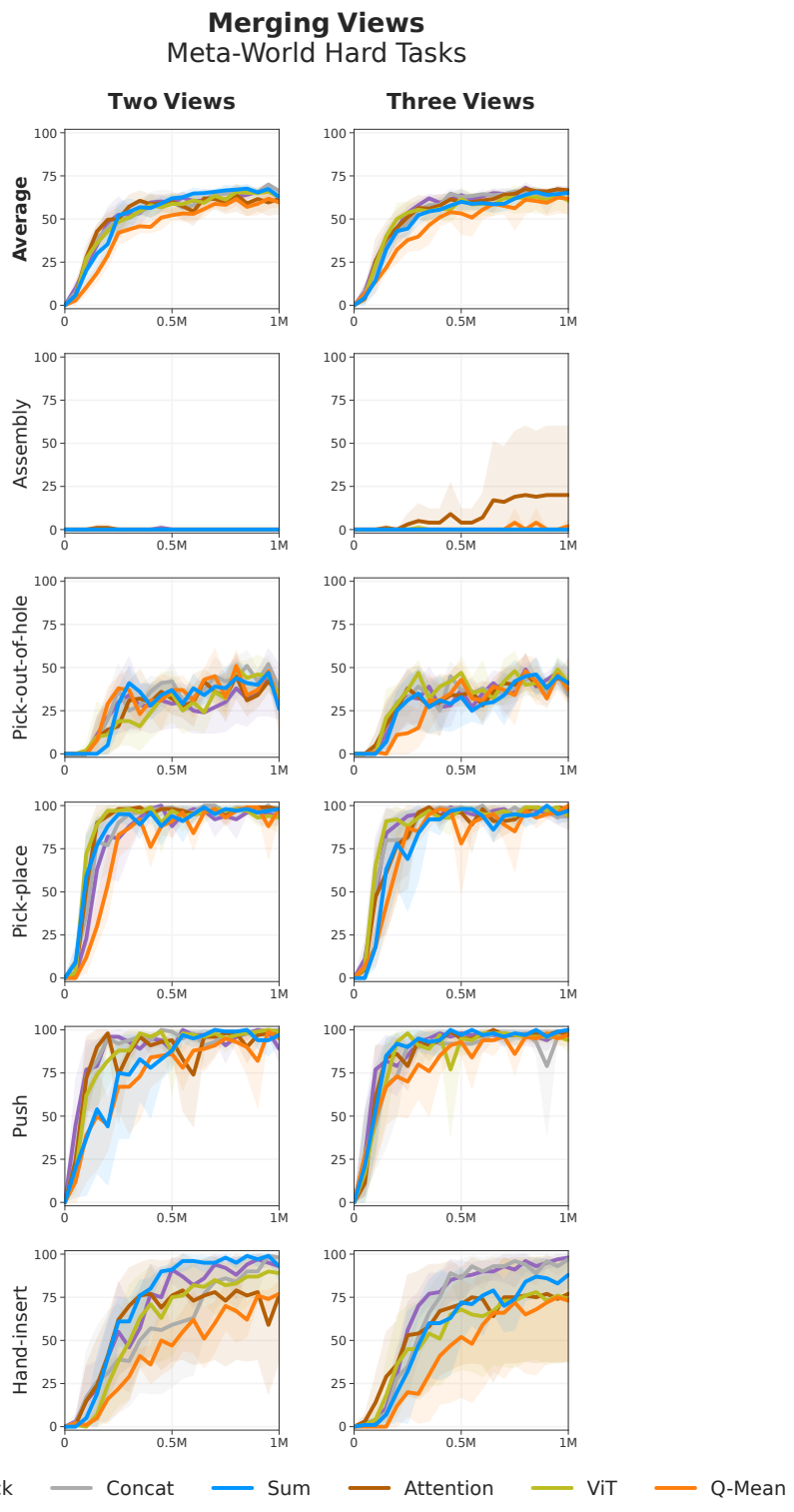
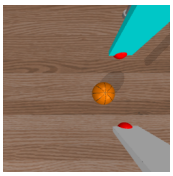
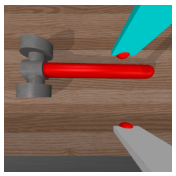
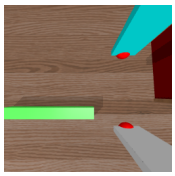
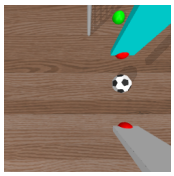
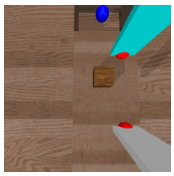
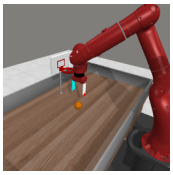
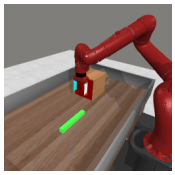
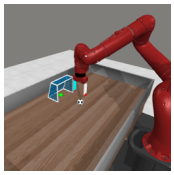
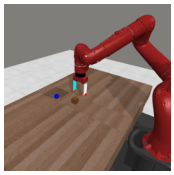
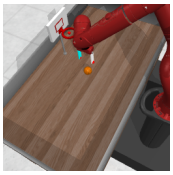
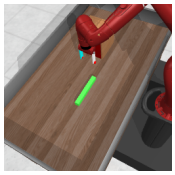
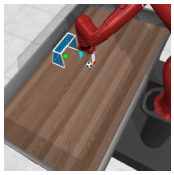
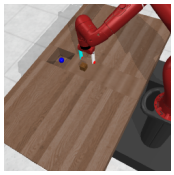


Figure B.6. **Multi-view Merging.** Success rate as a function of environment steps, averaged over 5 Meta-World hard visual RL tasks. Methods trained and evaluated on *(Left)* two camera views - First and Third A - *(Right)* three camera views. Mean and 95% CI over 5 random seeds. In reference to Figure 3.6 in the main text.

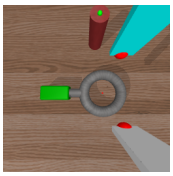
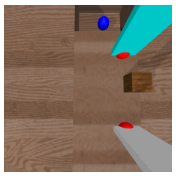
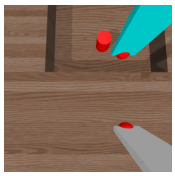
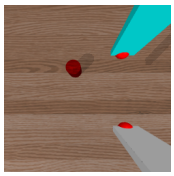
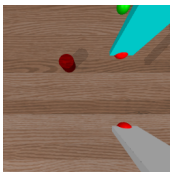
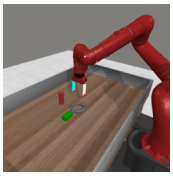
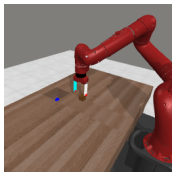
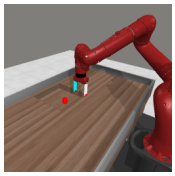
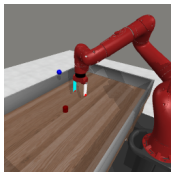
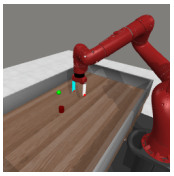
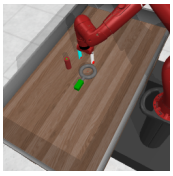
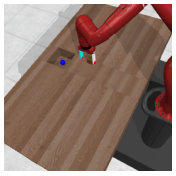
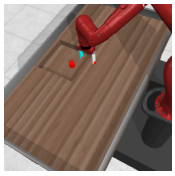
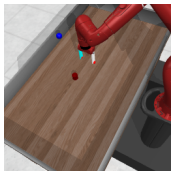
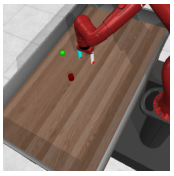
B.5 Visuals

B.5.1 Meta-World

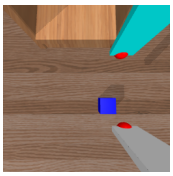
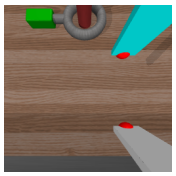
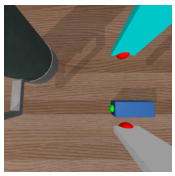
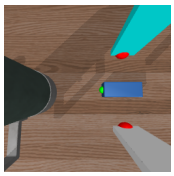
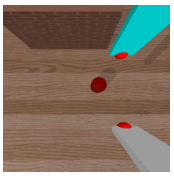
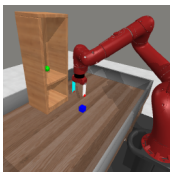
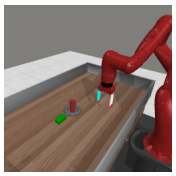
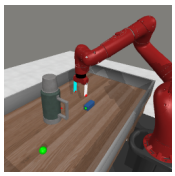
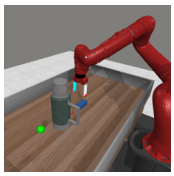
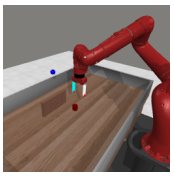
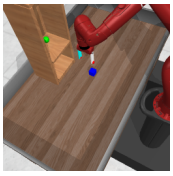
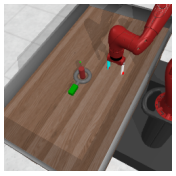
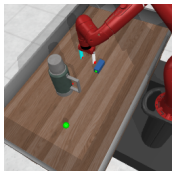
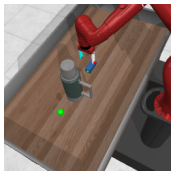
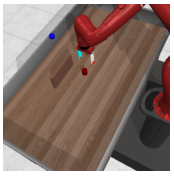
Medium Tasks

View	Basketball	Hammer	Peg Insert Side	Soccer	Sweep Into
First View					
Third View A					
Third View B					

Hard Tasks

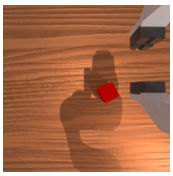
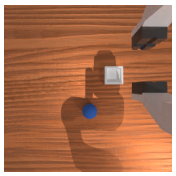
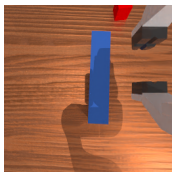
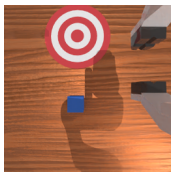
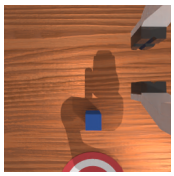
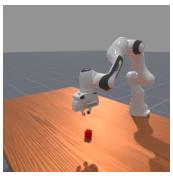
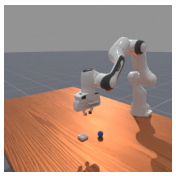
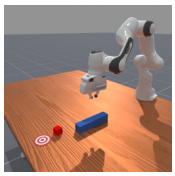
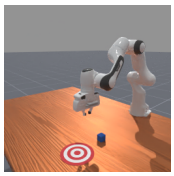
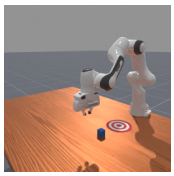
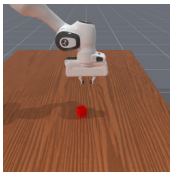
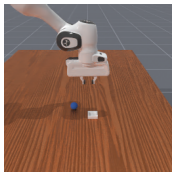
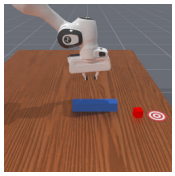
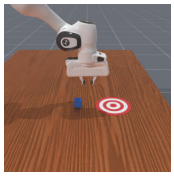
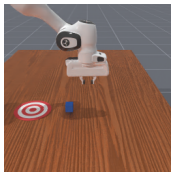
View	Assembly	Hand Insert	Pick Out Of Hole	Pick Place	Push
First View					
Third View A					
Third View B					

Very Hard Tasks

View	Shelf Place	Disassemble	Stick Pull	Stick Push	Pick Place Wall
First View					
Third View A					
Third View B					

B.5.2 ManiSkill3

ManiSkill3 Tasks

View	Pick Cube	Place Sphere	Poke Cube	Push Cube	Pull Cube
First View					
Third View A					
Third View B					

Bibliography

Iretiayo Akinola, Jacob Varley, and Dmitry Kalashnikov. Learning precise 3d manipulation from multiple uncalibrated cameras. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4616–4622. IEEE, 2020.

Abdulaziz Almuzairee and Henrik I Christensen. Squint: Fast visual reinforcement learning for sim-to-real robotics. *arXiv preprint arXiv:2602.21203*, 2026.

Abdulaziz Almuzairee, Nicklas Hansen, and Henrik I Christensen. A recipe for unbounded data augmentation in visual reinforcement learning. *Reinforcement Learning Journal*, 1:130–157, 2024.

Abdulaziz Almuzairee, Rohan Prashant Patil, Dwait Bhatt, and Henrik I Christensen. Merging and disentangling views in visual reinforcement learning for robotic manipulation. In *Conference on Robot Learning*, pages 981–1003. PMLR, 2025.

Christopher G Atkeson and Stefan Schaal. Robot learning from demonstration. In *ICML*, volume 97, pages 12–20, 1997.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

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.

Sumeet Batra and Gaurav S Sukhatme. Zero shot generalization of vision-based rl without data augmentation. In *Forty-second International Conference on Machine Learning*, 2024.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of artificial intelligence research*, 47:253–279, 2013.

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.

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemyslaw Debiak, Christy Dennison, David Farhi, Quirin Fischer, et al. Dota 2 with large scale deep reinforcement learning. *ArXiv*, abs/1912.06680, 2019.

Shubhankar Borse, Marvin Klingner, Varun Ravi Kumar, Hong Cai, Abdulaziz Almuzairee, Senthil Yogamani, and Fatih Porikli. X-align: Cross-modal cross-view alignment for bird's-eye-view segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3287–3297, January 2023.

Alexandre Brown and Glen Berseth. Segdac: Improving visual reinforcement learning by extracting dynamic object-centric representations from pretrained vision models, 2026. URL https://arxiv.org/abs/2508.09325.

Remi Cadene, Simon Alibert, Alexander Soare, Quentin Gallouedec, Adil Zouitine, Steven Palma, Pepijn Kooijmans, Michel Aractingi, Mustafa Shukor, Dana Aubakirova, Martino Russi, Francesco Capuano, Caroline Pascal, Jade Choghari, Jess Moss, and Thomas Wolf. Lerobot: State-of-the-art machine learning for real-world robotics in pytorch. https://github.com/huggingface/lerobot, 2024.

Roger Creus Castanyer, Johan Obando-Ceron, Lu Li, Pierre-Luc Bacon, Glen Berseth, Aaron Courville, and Pablo Samuel Castro. Stable gradients for stable learning at scale in deep reinforcement learning. *arXiv preprint arXiv:2506.15544*, 2025.

Wei-Di Chang, Mikael Henaff, Brandon Amos, Gregory Dudek, and Scott Fujimoto. The surprising difficulty of search in model-based reinforcement learning. *arXiv preprint arXiv:2601.21306*, 2026.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. Quantifying generalization in reinforcement learning. In *International conference on machine learning*, pages 1282–1289. PMLR, 2019.

Jack Collins, Shelvin Chand, Anthony Vanderkop, and David Howard. A review of physics simulators for robotic applications. *IEEE Access*, 9:51416–51431, 2021.

Pierluca D'Oro, Max Schwarzer, Evgenii Nikishin, Pierre-Luc Bacon, Marc G Bellemare, and Aaron Courville. Sample-efficient reinforcement learning by breaking the replay ratio barrier. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=OpC-9aBBVJe.

Mhairi Dunion and Stefano V Albrecht. Multi-view disentanglement for reinforcement learning

with multiple cameras. *arXiv preprint arXiv:2404.14064*, 2024.

Ayoub Echchahed and Pablo Samuel Castro. A survey of state representation learning for deep reinforcement learning. *arXiv preprint arXiv:2506.17518*, 2025.

Lasse Espeholt, Hubert Soyer, R. Munos, K. Simonyan, V. Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. *ArXiv*, abs/1802.01561, 2018.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

Linxi Fan, Guanzhi Wang, De-An Huang, Zhiding Yu, Li Fei-Fei, Yuke Zhu, and Animashree Anandkumar. Secant: Self-expert cloning for zero-shot generalization of visual policies. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 3088–3099. PMLR, 18–24 Jul 2021.

Chelsea Finn, Xin Yu Tan, Yan Duan, Trevor Darrell, Sergey Levine, and Pieter Abbeel. Learning visual feature spaces for robotic manipulation with deep spatial autoencoders. *arXiv preprint arXiv:1509.06113*, 25(2), 2015.

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.

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.

Matteo Gallici, Mattie Fellows, Benjamin Ellis, Bartomeu Pou, Ivan Masmitja, Jakob Nicolaus Foerster, and Mario Martin. Simplifying deep temporal difference learning. *arXiv preprint arXiv:2407.04811*, 2024.

Theophile Gervet, Zhou Xian, Nikolaos Gkanatsios, and Katerina Fragkiadaki. Act3d: Infinite resolution action detection transformer for robotic manipulation. *arXiv preprint arXiv:2306.17817*, 2023.

Paul Glick, Srinivasan A Suresh, Donald Ruffatto, Mark Cutkosky, Michael T Tolley, and Aaron Parness. A soft robotic gripper with gecko-inspired adhesive. *IEEE Robotics and Automation Letters*, 3(2):903–910, 2018.

Ankit Goyal, Valts Blukis, Jie Xu, Yijie Guo, Yu-Wei Chao, and Dieter Fox. Rvt-2: Learning precise manipulation from few demonstrations. *arXiv preprint arXiv:2406.08545*, 2024.

Bram Grooten, Tristan Tomilin, Gautham Vasan, Matthew E Taylor, A Rupam Mahmood, Meng Fang, Mykola Pechenizkiy, and Decebal Constantin Mocanu. Madi: Learning to mask distractions for generalization in visual deep reinforcement learning. *arXiv preprint arXiv:2312.15339*, 2023.

Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, et al. Maniskill2: A unified benchmark for generalizable manipulation skills. *arXiv preprint arXiv:2302.04659*, 2023.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

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

Nicklas Hansen and Xiaolong Wang. Generalization in reinforcement learning by soft data augmentation. In *International Conference on Robotics and Automation*, 2021.

Nicklas Hansen, Rishabh Jangir, Yu Sun, Guillem Alenyà, Pieter Abbeel, Alexei A. Efros, Lerrel Pinto, and Xiaolong Wang. Self-supervised policy adaptation during deployment. In *International Conference on Learning Representations*, 2021a.

Nicklas Hansen, Hao Su, and Xiaolong Wang. Stabilizing deep q-learning with convnets and vision transformers under data augmentation. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2021b.

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. In *International Conference on Machine Learning (ICML)*, 2022.

Nicklas Hansen, Zhecheng Yuan, Yanjie Ze, Tongzhou Mu, Aravind Rajeswaran, Hao Su, Huazhe Xu, and Xiaolong Wang. On pre-training for visuo-motor control: Revisiting a learning-from-scratch baseline. In *International Conference on Machine Learning (ICML)*, 2023.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations (ICLR)*, 2024.

Nicklas Hansen, Hao Su, and Xiaolong Wang. Learning massively multitask world models for continuous control. *arXiv preprint arXiv:2511.19584*, 2025.

H. V. Hasselt, A. Guez, and D. Silver. Deep reinforcement learning with double q-learning. In *Aaai*, 2016.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

Nicolas Heess, Dhruva Tb, Srinivasan Sriram, Jay Lemmon, Josh Merel, Greg Wayne, Yuval Tassa, Tom Erez, Ziyu Wang, SM Eslami, et al. Emergence of locomotion behaviours in rich environments. *arXiv preprint arXiv:1707.02286*, 2017.

Yuenan Hou, Lifeng Liu, Qing Wei, Xudong Xu, and Chunlin Chen. A novel ddpg method with prioritized experience replay. In *2017 IEEE international conference on systems, man, and cybernetics (SMC)*, pages 316–321. IEEE, 2017.

Kyle Hsu, Moo Jin Kim, Rafael Rafailov, Jiajun Wu, and Chelsea Finn. Vision-based manipulators need to also see from their hands. *arXiv preprint arXiv:2203.12677*, 2022.

Edward S Hu, James Springer, Oleh Rybkin, and Dinesh Jayaraman. Privileged sensing scaffolds reinforcement learning. *arXiv preprint arXiv:2405.14853*, 2024.

Hengyuan Hu, Suvir Mirchandani, and Dorsa Sadigh. Imitation bootstrapped reinforcement learning. *arXiv preprint arXiv:2311.02198*, 2023.

Shengyi Huang, Rousslan Fernand Julien Dossa, Chang Ye, Jeff Braga, Dipam Chakraborty, Kinal Mehta, and João G.M. Araújo. Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms. *Journal of Machine Learning Research*, 23(274): 1–18, 2022a. URL http://jmlr.org/papers/v23/21-1342.html.

Yangru Huang, Peixi Peng, Yifan Zhao, Guangyao Chen, and Yonghong Tian. Spectrum random masking for generalization in image-based reinforcement learning. *Advances in Neural Information Processing Systems*, 35:20393–20406, 2022b.

Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26):eaau5872, 2019.

Seongwon Jang, Hyemi Jeong, and Hyunseok Yang. Murm: utilization of multi-views for goal-conditioned reinforcement learning in robotic manipulation. *Robotics*, 12(4):119, 2023.

Rishabh Jangir, Nicklas Hansen, Sambaran Ghosal, Mohit Jain, and Xiaolong Wang. Look closer: Bridging egocentric and third-person views with transformers for robotic manipulation. *IEEE Robotics and Automation Letters*, 7(2):3046–3053, 2022.

Ryan C. Julian, Benjamin Swanson, Gaurav S. Sukhatme, Sergey Levine, Chelsea Finn, and Karol Hausman. Efficient adaptation for end-to-end vision-based robotic manipulation. *ArXiv*, abs/2004.10190, 2020.

Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 1998.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*, 2018.

Elia Kaufmann, Leonard Bauersfeld, Antonio Loquercio, Matthias Müller, Vladlen Koltun, and Davide Scaramuzza. Champion-level drone racing using deep reinforcement learning. *Nat.*, 620(7976):982–987, 2023. doi: 10.1038/S41586-023-06419-4. URL https://doi.org/10.1038/s41586-023-06419-4.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

Sunghwan Kim, Woojeh Chung, Zhirui Dai, Dwait Bhatt, Arth Shukla, Hao Su, Yulun Tian, and Nikolay Atanasov. Seeing the bigger picture: 3d latent mapping for mobile manipulation policy learning. *arXiv preprint arXiv:2510.03885*, 2025.

Robert Kirk, Amy Zhang, Edward Grefenstette, and Tim Rocktäschel. A survey of zero-shot generalisation in deep reinforcement learning. *Journal of Artificial Intelligence Research*, 76: 201–264, 2023.

Ilya Kostrikov, Denis Yarats, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *International Conference on Learning Representations*, 2020.

Patrick Lancaster, Nicklas Hansen, Aravind Rajeswaran, and Vikash Kumar. Modem-v2: Visuomotor world models for real-world robot manipulation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7530–7537. IEEE, 2024.

Michael Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. *arXiv preprint arXiv:2004.14990*, 2020.

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.

Hojoon Lee, Youngdo Lee, Takuma Seno, Donghu Kim, Peter Stone, and Jaegul Choo. Hyperspher-

ical normalization for scalable deep reinforcement learning. *arXiv preprint arXiv:2502.15280*, 2025.

Kimin Lee, Kibok Lee, Jinwoo Shin, and Honglak Lee. A simple randomization technique for generalization in deep reinforcement learning. *ArXiv*, abs/1910.05396, 2019a.

Su Young Lee, Choi Sungik, and Sae-Young Chung. Sample-efficient deep reinforcement learning via episodic backward update. *Advances in neural information processing systems*, 32, 2019b.

Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

Minne Li, Lisheng Wu, Jun Wang, and Haitham Bou Ammar. Multi-view reinforcement learning. *Advances in neural information processing systems*, 32, 2019.

Zechu Li, Tao Chen, Zhang-Wei Hong, Anurag Ajay, and Pulkit Agrawal. Parallel q-learning: Scaling off-policy reinforcement learning under massively parallel simulation. In *International Conference on Machine Learning*, pages 19440–19459. PMLR, 2023.

Zheng Li, Pei Qu, Yufei Jia, Shihui Zhou, Haizhou Ge, Jiahang Cao, Jinni Zhou, Guyue Zhou, and Jun Ma. Manivid-3d: Generalizable view-invariant reinforcement learning for robotic manipulation via disentangled 3d representations. *arXiv preprint arXiv:2509.11125*, 2025.

Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from lidar-camera via spatiotemporal transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Jacky Liang, Viktor Makoviychuk, Ankur Handa, Nuttapong Chentanez, Miles Macklin, and Dieter Fox. Gpu-accelerated robotic simulation for distributed reinforcement learning. In *Conference on Robot Learning*, pages 270–282. PMLR, 2018.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Toru Lin, Kartik Sachdev, Linxi Fan, Jitendra Malik, and Yuke Zhu. Sim-to-real reinforcement learning for vision-based dexterous manipulation on humanoids. *arXiv preprint arXiv:2502.20396*, 2025.

Guozheng Ma, Zhen Wang, Zhecheng Yuan, Xueqian Wang, Bo Yuan, and Dacheng Tao. A comprehensive survey of data augmentation in visual reinforcement learning. *International Journal of Computer Vision*, 133(10):7368–7405, 2025.

Abhiram Maddukuri, Zhenyu Jiang, Lawrence Yunliang Chen, Soroush Nasiriany, Yuqi

Xie, Yu Fang, Wenqi Huang, Zu Wang, Zhenjia Xu, Nikita Chernyadev, et al. Sim-and-real co-training: A simple recipe for vision-based robotic manipulation. *arXiv preprint arXiv:2503.24361*, 2025.

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.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

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.

Tongzhou Mu, Zhan Ling, Fanbo Xiang, Derek Yang, Xuanlin Li, Stone Tao, Zhiao Huang, Zhiwei Jia, and Hao Su. Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations. *arXiv preprint arXiv:2107.14483*, 2021.

Tongzhou Mu, Zhaoyang Li, Stanis law Wiktor Strzelecki, Xiu Yuan, Yunchao Yao, Litian Liang, and Hao Su. When should we prefer state-to-visual dagger over visual reinforcement learning? *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(14):14637–14645, 2025.

Michal Nauman, Mateusz Ostaszewski, Krzysztof Jankowski, Piotr Mi loś, and Marek Cygan. Bigger, regularized, optimistic: scaling for compute and sample efficient continuous control. *Advances in neural information processing systems*, 37:113038–113071, 2024.

Michal Nauman, Marek Cygan, Carmelo Sferrazza, Aviral Kumar, and Pieter Abbeel. Bigger, regularized, categorical: High-capacity value functions are efficient multi-task learners. *arXiv preprint arXiv:2505.23150*, 2025.

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016.

Johan Obando-Ceron, Walter Mayor, Samuel Lavoie, Scott Fujimoto, Aaron Courville, and Pablo Samuel Castro. Simplicial embeddings improve sample efficiency in actor-critic agents. *arXiv preprint arXiv:2510.13704*, 2025.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov,

Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Seohong Park, Oleh Rybkin, and Sergey Levine. Metra: Scalable unsupervised rl with metric-aware abstraction. *arXiv preprint arXiv:2310.08887*, 2023.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018.

Lerrel Pinto and Abhinav Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 3406–3413. IEEE, 2016.

Lerrel Pinto, Marcin Andrychowicz, Peter Welinder, Wojciech Zaremba, and Pieter Abbeel. Asymmetric actor critic for image-based robot learning. *arXiv preprint arXiv:1710.06542*, 2017.

Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

Shengyi Qian, Kaichun Mo, Valts Blukis, David F Fouhey, Dieter Fox, and Ankit Goyal. 3d-mvp: 3d multiview pretraining for robotic manipulation. *arXiv preprint arXiv:2406.18158*, 2024.

Roberta Raileanu, M. Goldstein, Denis Yarats, Ilya Kostrikov, and R. Fergus. Automatic data augmentation for generalization in deep reinforcement learning. *ArXiv*, abs/2006.12862, 2020.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Younggyo Seo, Danijar Hafner, Hao Liu, Fangchen Liu, Stephen James, Kimin Lee, and Pieter

Abbeel. Masked world models for visual control. In *Conference on Robot Learning*, pages 1332–1344. PMLR, 2023a.

Younggyo Seo, Junsu Kim, Stephen James, Kimin Lee, Jinwoo Shin, and Pieter Abbeel. Multi-view masked world models for visual robotic manipulation. In *International Conference on Machine Learning*, pages 30613–30632. PMLR, 2023b.

Younggyo Seo, Carmelo Sferrazza, Juyue Chen, Guanya Shi, Rocky Duan, and Pieter Abbeel. Learning sim-to-real humanoid locomotion in 15 minutes. *arXiv preprint arXiv:2512.01996*, 2025a.

Younggyo Seo, Carmelo Sferrazza, Haoran Geng, Michal Nauman, Zhao-Heng Yin, and Pieter Abbeel. Fasttd3: Simple, fast, and capable reinforcement learning for humanoid control. *arXiv preprint arXiv:2505.22642*, 2025b.

Carmelo Sferrazza, Dun-Ming Huang, Xingyu Lin, Youngwoon Lee, and Pieter Abbeel. Humanoidbench: Simulated humanoid benchmark for whole-body locomotion and manipulation. *arXiv preprint arXiv:2403.10506*, 2024.

Arth Shukla. Speeding up sac with massively parallel simulation. *https://arthshukla.substack.com*, Mar 2025. URL https://arthshukla.substack.com/p/speeding-up-sac-with-massively-parallel.

Skand Skand, Bikram Pandit, Chanho Kim, Li Fuxin, and Stefan Lee. Simple masked training strategies yield control policies that are robust to sensor failure. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=AsbyZRdqPv.

Yuda Song, Dhruv Rohatgi, Aarti Singh, and J Andrew Bagnell. To distill or decide? understanding the algorithmic trade-off in partially observable reinforcement learning. *arXiv preprint arXiv:2510.03207*, 2025.

Aravind Srinivas, Michael Laskin, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. *arXiv preprint arXiv:2004.04136*, 2020.

Austin Stone, Oscar Ramirez, Kurt Konolige, and Rico Jonschkowski. The distracting control suite–a challenging benchmark for reinforcement learning from pixels. *arXiv preprint arXiv:2101.02722*, 2021.

Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling representation learning from reinforcement learning. *ArXiv*, abs/2004.1499, 2020.

Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. *Advances in neural information processing systems*, 34:251–266, 2021.

Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse-kai Chan, et al. Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai. *arXiv preprint arXiv:2410.00425*, 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.

Eugene Teoh, Sumit Patidar, Xiao Ma, and Stephen James. Green screen augmentation enables scene generalisation in robotic manipulation. *arXiv preprint arXiv:2407.07868*, 2024.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017.

Basile Van Hoorick, Rundi Wu, Ege Ozguroglu, Kyle Sargent, Ruoshi Liu, Pavel Tokmakov, Achal Dave, Changxi Zheng, and Carl Vondrick. Generative camera dolly: Extreme monocular dynamic novel view synthesis. In *European Conference on Computer Vision*, pages 313–331. Springer, 2024.

Miguel Vasco, Hang Yin, Francisco S Melo, and Ana Paiva. How to sense the world: Leveraging hierarchy in multimodal perception for robust reinforcement learning agents. *arXiv preprint arXiv:2110.03608*, 2021.

Oriol Vinyals, I. Babuschkin, Wojciech Czarnecki, Michaël Mathieu, Andrew Dudzik, J. Chung, D. Choi, Richard Powell, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, pages 1–5, 2019.

Kaixin Wang, Bingyi Kang, Jie Shao, and Jiashi Feng. Improving generalization in reinforcement learning with mixture regularization. *Advances in Neural Information Processing Systems*, 33: 7968–7978, 2020.

Xudong Wang, Long Lian, and Stella X. Yu. Unsupervised visual attention and invariance for reinforcement learning. *ArXiv*, abs/2104.02921, 2021.

Ziyu Wang, Yanjie Ze, Yifei Sun, Zhecheng Yuan, and Huazhe Xu. Generalizable visual reinforcement learning with segment anything model. *arXiv preprint arXiv:2312.17116*, 2023.

Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8:279–292, 1992.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018.

Charles Xu, Qiyang Li, Jianlan Luo, and Sergey Levine. Rldg: Robotic generalist policy distillation via reinforcement learning. *arXiv preprint arXiv:2412.09858*, 2024.

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.

Huanhuan Yang, Dianxi Shi, Guojun Xie, Yingxuan Peng, Yi Zhang, Yantai Yang, and Shaowu Yang. Self-supervised representations for multi-view reinforcement learning. In *The 38th Conference on Uncertainty in Artificial Intelligence*, 2022.

Sizhe Yang, Yanjie Ze, and Huazhe Xu. Movie: Visual model-based policy adaptation for view generalization. *Advances in Neural Information Processing Systems*, 36, 2024.

Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European conference on computer vision (ECCV)*, pages 767–783, 2018.

Denis Yarats, Amy Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and Rob Fergus. Improving sample efficiency in model-free reinforcement learning from images, 2019.

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, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Reinforcement learning with prototypical representations. *arXiv preprint arXiv:2102.11271*, 2021b.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in neural information processing systems*, 35:24611–24624, 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*, pages 1094–1100. PMLR, 2020.

Zhecheng Yuan, Guozheng Ma, Yao Mu, Bo Xia, Bo Yuan, Xueqian Wang, Ping Luo, and Huazhe Xu. Don't touch what matters: Task-aware lipschitz data augmentation for visual reinforcement learning. *arXiv preprint arXiv:2202.09982*, 2022a.

Zhecheng Yuan, Zhengrong Xue, Bo Yuan, Xueqian Wang, Yi Wu, Yang Gao, and Huazhe Xu. Pre-trained image encoder for generalizable visual reinforcement learning. *Advances in Neural Information Processing Systems*, 35:13022–13037, 2022b.

Zhecheng Yuan, Tianming Wei, Shuiqi Cheng, Gu Zhang, Yuanpei Chen, and Huazhe Xu. Learning to manipulate anywhere: A visual generalizable framework for reinforcement learning. *arXiv preprint arXiv:2407.15815*, 2024.

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.

Hanping Zhang and Yuhong Guo. Generalization of reinforcement learning with policy-aware adversarial data augmentation. *arXiv preprint arXiv:2106.15587*, 2021.

Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020.

Ziwen Zhuang, Zipeng Fu, Jianren Wang, Christopher Atkeson, Soeren Schwertfeger, Chelsea Finn, and Hang Zhao. Robot parkour learning. *arXiv preprint arXiv:2309.05665*, 2023.