Merging and Disentangling Views in Visual Reinforcement Learning for Robotic Manipulation

Abdulaziz Almuzairee **Rohan Patil** **Dwait Bhatt** **Henrik I. Christensen**
UC San Diego

aalmuzairee.github.io/mad

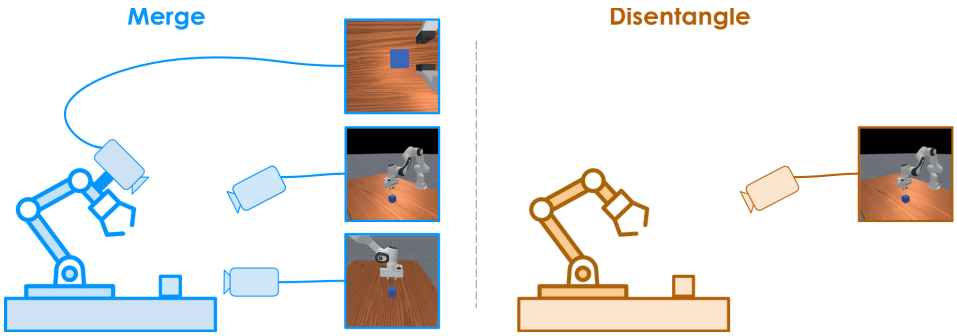


Figure 1. **Merge And Disentangle.** We introduce a method that can merge multiple camera views during training to learn better representations, while simultaneously disentangling the camera view representations, such that the policy can function with any singular view input during deployment.

Abstract:
Vision is well-known for its use in robotic manipulation, especially using visual servoing. Due to the 3D nature of the world, using multiple camera views and merging them creates better representations for Q-learning and in turn, trains more sample efficient policies. Nevertheless, these multi-view policies are sensitive to failing cameras and can be burdensome to deploy. To mitigate these issues, we introduce a **M**erge **A**nd **D**isentanglement (**MAD**) algorithm that efficiently merges views to increase sample efficiency while simultaneously disentangling views by augmenting multi-view feature inputs with single-view features. This produces robust policies and allows lightweight deployment. We demonstrate the efficiency and robustness of our approach using Meta-World and ManiSkill3.

Keywords: Visual Reinforcement Learning, Multi View Robot Learning

1 Introduction

Visual reinforcement learning (RL) has demonstrated remarkable progress, achieving superhuman performance on Atari games [1] and successfully tackling complex real-world applications from robotic manipulation to autonomous flight [2, 3, 4, 5]. 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 [6]. 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 [7].

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

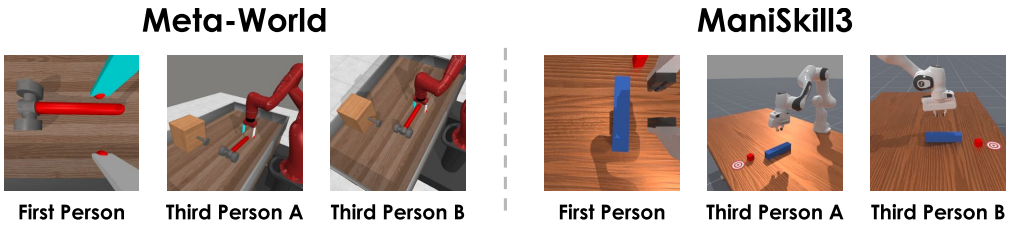


Figure 2. **Environment Setup.** Our environment setup for robotic manipulation where we use three input camera views as inputs, namely First Person, Third Person A, and Third Person B across two visual RL benchmarks: *(Left)* Meta-World *(Right)* ManiSkill3. Extended visuals in Appendix E.

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 [8], 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 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 1.

We evaluate our method on 20 visual RL tasks from Meta-World [9] and ManiSkill3 [10] benchmarks, and find that MAD is able to achieve higher sample efficiency than baselines while being robust to a reduction in available camera views.

2 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 [11, 12, 13, 14, 15, 16]. 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 [17, 18, 19, 20, 21, 22]. In contrast, [18, 23] 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 [23, 24, 25, 26, 27, 28, 29, 8] and increasing visual generalization by proposing new types of transformations [30, 31, 32, 33, 34, 35, 36]. Most importantly, Almuzairee et al. [8] 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 sam-

ple 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. [37, 6, 38, 39, 40, 41, 42]. These approaches leverage multiple camera perspectives to mitigate occlusions and improve representations, leading to enhanced task performance [43, 6]. Most notably, Hsu et al. [6] 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. [44] 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 [7].

Merging and Disentangling Features. Research in computer vision has extensively explored feature merging [45, 46, 47, 48] and feature disentanglement [13, 49, 50] across multiple camera views and sensor modalities. Inspired by these efforts, multiple works in visual RL have explored feature merging [51, 52, 44, 6] and feature disentanglement [53, 17, 54]. However, policies trained on multiple feature inputs often suffer significant performance degradation when any feature inputs become unavailable [55, 7]. Multiple works have researched this problem of feature disentanglement for reinforcement learning. Hu et al. [56] 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. [57] introduce a multi sensor feature encoder for RL polices that disentangles features through dropout, yielding robust policies. Recently, Dunion and Albrecht [7] 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 Preliminaries

Visual Reinforcement Learning formulates the interaction between the agent and the environment as a Partially Observable Markov Decision Process (POMDP) [58], 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. [1] 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) [19] is an actor critic algorithm, based on Soft Actor Critic [59] 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 $\mathcal{L}_{Q_\theta}(\mathcal{D}) = \mathbb{E}_{(\mathbf{o}_t, \mathbf{a}_t, r_t, \mathbf{o}_{t+1}) \sim \mathcal{D}}[(Q_\theta(\mathbf{o}_t, \mathbf{a}_t) - r_t - \gamma Q_{\overline{\theta}}(\mathbf{o}_{t+1}, \mathbf{a}'))^2]$ 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 $Q_{\overline{\theta}}$ represents an exponential moving average of the weights from Q_θ [60, 61, 59]. 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 $\mathcal{L}_{\pi_\phi}(\mathcal{D}) = \mathbb{E}_{\mathbf{o}_t \sim \mathcal{D}}[-Q_\theta(\mathbf{o}_t, \pi_\phi(\mathbf{o}_t))]$. Both the critic and actor losses are updated iteratively using stochastic gradient descent in an aim

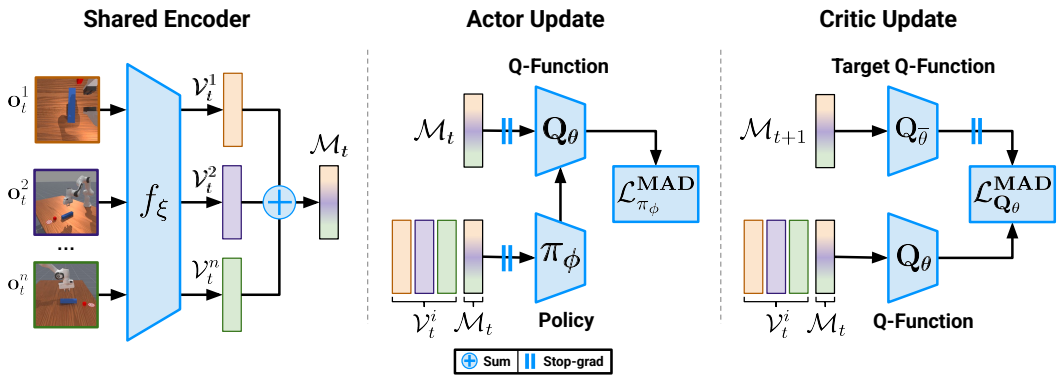


Figure 3. **Our framework.** Update diagram of a generic visual actor-critic model with our modifications. MAD *merges* camera views through feature summation and *disentangles* camera views by selectively augmenting inputs to the downstream actor and critic with all the singular view features. The agent is trained end-to-end with our defined MAD loss functions. *(Left)*: Single Shared CNN Encoder. *(Middle)*: Actor Update diagram *(Right)*: Critic Update diagram.

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 [19, 22].

4 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 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.

4.1 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.

4.2 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 [18], 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 cam-

era 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 $\mathcal{V}_t^i$.

4.3 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 [18, 23, 8]. Later works established ways to stabilize actor-critic learning under data augmentation [24, 8]. We follow SADA [8] 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:

$$\mathcal{L}_{\pi_\phi}^{\textbf{UnAug}}(\mathcal{D}) = \mathbb{E}_{\mathbf{o}_t^m \sim \mathcal{D}} \left[-Q_\theta(\mathcal{M}_t, \pi_\phi(\mathcal{M}_t)) \right]$$
$$\mathcal{L}_{\pi_\phi}^{\textbf{Aug}}(\mathcal{D}, i) = \mathbb{E}_{\mathbf{o}_t^m \sim \mathcal{D}} \left[-Q_\theta(\mathcal{M}_t, \pi_\phi(\mathcal{V}_t^i)) \right]$$
$$\mathcal{L}_{\pi_\phi}^{\textbf{MAD}}(\mathcal{D}) = \alpha * \mathcal{L}_{\pi_\phi}^{\textbf{UnAug}}(\mathcal{D}) \ + \ (1-\alpha) * \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_{\pi_\phi}^{\textbf{Aug}}(\mathcal{D}, i) \quad \text{(actor)} \tag{1}$$

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 [24]. Setting this α hyperparameter to 0.5 recovers the original SADA loss. On the other hand, the MAD update loss function for a generic critic becomes:

$$\mathcal{L}_{Q_\theta}^{\textbf{UnAug}}(\mathcal{D}) = \mathbb{E}_{(\mathbf{o}_t^m, \mathbf{a}_t, r_t, \mathbf{o}_{t+1}^m) \sim \mathcal{D}} \left[\left(Q_\theta(\mathcal{M}_t, \mathbf{a}_t) \ - r_t - \gamma Q_{\overline{\theta}}(\mathcal{M}_{t+1}, \mathbf{a}') \right)^2 \right]$$
$$\mathcal{L}_{Q_\theta}^{\textbf{Aug}}(\mathcal{D}, i) = \mathbb{E}_{(\mathbf{o}_t^m, \mathbf{a}_t, r_t, \mathbf{o}_{t+1}^m) \sim \mathcal{D}} \left[\left(Q_\theta(\mathcal{V}_t^i, \mathbf{a}_t) \ - r_t - \gamma Q_{\overline{\theta}}(\mathcal{M}_{t+1}, \mathbf{a}') \right)^2 \right]$$
$$\mathcal{L}_{Q_\theta}^{\textbf{MAD}}(\mathcal{D}) = \alpha * \mathcal{L}_{Q_\theta}^{\textbf{UnAug}}(\mathcal{D}) \ + \ (1-\alpha) * \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_{Q_\theta}^{\textbf{Aug}}(\mathcal{D}, i) \quad \text{(critic)} \tag{2}$$

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.

5 Experiments

We benchmark our method and baselines on a total of 20 visual RL tasks, consisting of 15 Meta-World v2 tasks [9] and 5 ManiSkill3 tasks [10], 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 2 for visualizations of our camera setup in each environment. The full task list is defined in A.3, and detailed visuals of all tasks can be found in E. 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?

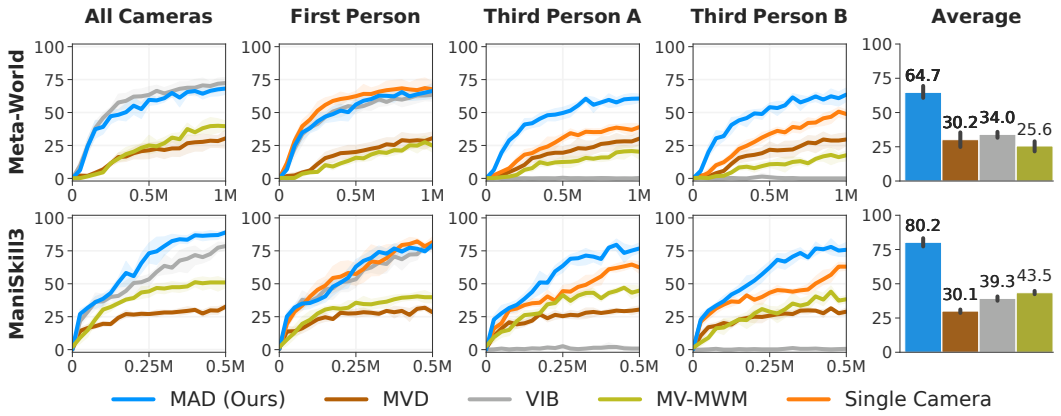


Figure 4. **Overall Robustness.** Success rate as a function of environment steps, averaged over all *(Top)* 15 Meta-World and *(Bottom)* 5 ManiSkill3 visual RL tasks. Methods are trained on all three camera views and evaluated on all and singular camera views with the final average displayed on the far right. Mean and 95% CI over 5 random seeds.

- **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.** Can MAD be adapted to other modalities?

Implementation. We use DrQ [19], a visual based Soft Actor Critic algorithm [59], as the backbone algorithm for our method across both environments, with detailed hyper parameters defined in Appendix A.1. To process all input camera views, we use a single shared DQN CNN encoder [62]. 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 [6, 7]. 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 [9] tasks spanning medium, hard, and very hard difficulties, as defined in [63], and 5 tasks from ManiSkill3 [10]. 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** [7], 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** [6], 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** [38], 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 [19], where DrQ agents are trained only on a single view and evaluated on that same single view.

5.1 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 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

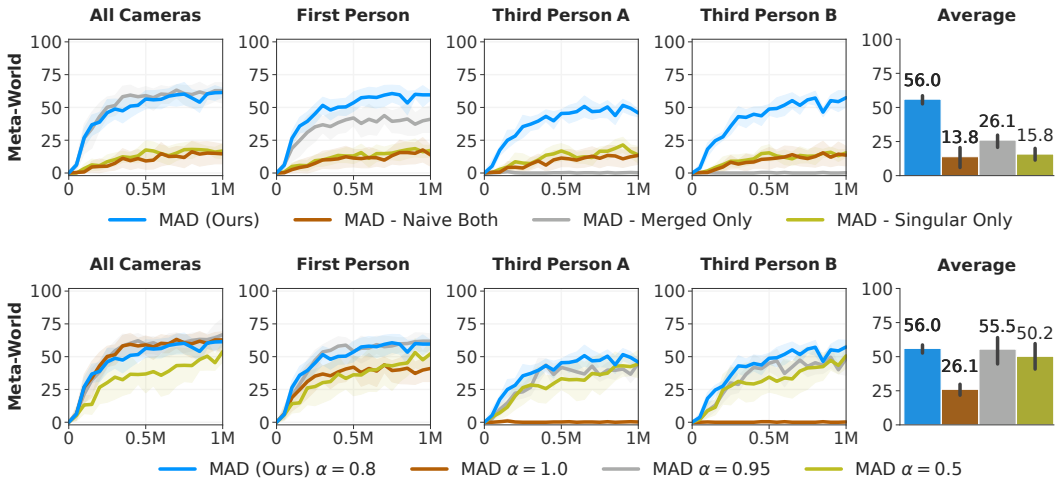


Figure 5. **Ablations.** Success rate as a function of environment steps, averaged over 5 Meta-World hard tasks. *(Top)* Component Ablations. *(Bottom)* Alpha Ablations. Methods trained on all camera views and evaluated on all and singular camera views. Mean and 95% CI over 5 random seeds.

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 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 D. Extended experiments on increasing views (5 views), and varying input view resolutions are provided in Appendix B.

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 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

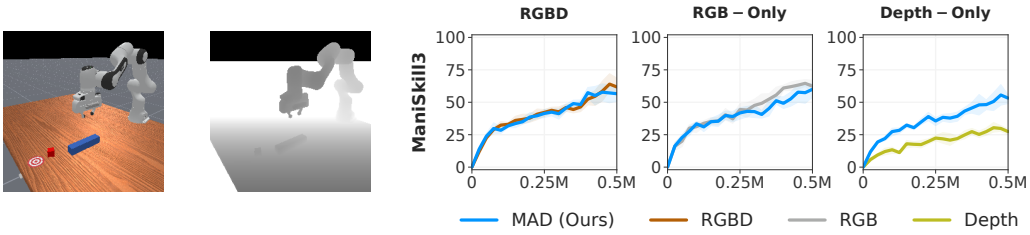


Figure 6. **Modality Adaptability.** Success rate as a function of environment steps averaged over 5 ManiSkill3 tasks. *(Left)* RGB and Depth images from the PokeCube Third Person A view. *(Right)* Methods are trained on one camera view (Third Person A) but with different modalities, and evaluated accordingly. Mean and 95% CI over 5 random seeds.

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 empirically *compare our choice of merging using summation with other alternatives* in Appendix C, and find that all merging methods achieve similar sample efficiency on two and three views.

Adaptability. We further evaluate whether *MAD can be adapted to different modalities* such as depth and RGB as opposed to different camera views, and display results in Figure 6. 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 achieves 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.

6 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 [64], demonstration-bootstrapped methods [65], or enhanced sample efficient visual Q-learning methods [66, 67, 68]. 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 [69], 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, and serve as a strong baseline for future work in multi view reinforcement learning.

Acknowledgments

This research has in part been sponsored by Kuwait University through a visiting fellow program and by ARL DCIST. The support is gratefully acknowledged.

References

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

[2] S. Levine, C. Finn, T. Darrell, and P. Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

[3] L. Pinto and A. 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.

[4] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, et al. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*, 2018.

[5] E. Kaufmann, L. Bauersfeld, A. Loquercio, M. Müller, V. Koltun, and D. 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.

[6] K. Hsu, M. J. Kim, R. Rafailov, J. Wu, and C. Finn. Vision-based manipulators need to also see from their hands. *arXiv preprint arXiv:2203.12677*, 2022.

[7] M. Dunion and S. V. Albrecht. Multi-view disentanglement for reinforcement learning with multiple cameras. *arXiv preprint arXiv:2404.14064*, 2024.

[8] A. Almuzairee, N. Hansen, and H. I. Christensen. A recipe for unbounded data augmentation in visual reinforcement learning. *arXiv preprint arXiv:2405.17416*, 2024.

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

[10] S. Tao, F. Xiang, A. Shukla, Y. Qin, X. Hinrichsen, X. Yuan, C. Bao, X. Lin, Y. Liu, T.-k. Chan, et al. Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai. *arXiv preprint arXiv:2410.00425*, 2024.

[11] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016.

[12] Z. Wu, Y. Xiong, S. X. Yu, and D. 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.

[13] A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[14] Y. Tian, D. Krishnan, and P. Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

[15] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[16] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[17] A. Srinivas, M. Laskin, and P. Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. *arXiv preprint arXiv:2004.04136*, 2020.

[18] M. Laskin, K. Lee, A. Stooke, L. Pinto, P. Abbeel, and A. Srinivas. Reinforcement learning with augmented data. *arXiv preprint arXiv:2004.14990*, 2020.

[19] I. Kostrikov, D. Yarats, and R. Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *International Conference on Learning Representations*, 2020.

[20] A. Stooke, K. Lee, P. Abbeel, and M. Laskin. Decoupling representation learning from reinforcement learning. *ArXiv*, abs/2004.1499, 2020.

[21] D. Yarats, R. Fergus, A. Lazaric, and L. Pinto. Reinforcement learning with prototypical representations. *arXiv preprint arXiv:2102.11271*, 2021.

[22] D. Yarats, R. Fergus, A. Lazaric, and L. Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. *arXiv preprint arXiv:2107.09645*, 2021.

[23] R. Raileanu, M. Goldstein, D. Yarats, I. Kostrikov, and R. Fergus. Automatic data augmentation for generalization in deep reinforcement learning. *ArXiv*, abs/2006.12862, 2020.

[24] N. Hansen, H. Su, and X. Wang. Stabilizing deep q-learning with convnets and vision transformers under data augmentation. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[25] X. Wang, L. Lian, and S. X. Yu. Unsupervised visual attention and invariance for reinforcement learning. *ArXiv*, abs/2104.02921, 2021.

[26] L. Fan, G. Wang, D.-A. Huang, Z. Yu, L. Fei-Fei, Y. Zhu, and A. Anandkumar. Secant: Self-expert cloning for zero-shot generalization of visual policies. In M. Meila and T. 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.

[27] Z. Yuan, G. Ma, Y. Mu, B. Xia, B. Yuan, X. Wang, P. Luo, and H. Xu. Don't touch what matters: Task-aware lipschitz data augmentation for visual reinforcement learning. *arXiv preprint arXiv:2202.09982*, 2022.

[28] B. Grooten, T. Tomilin, G. Vasan, M. E. Taylor, A. R. Mahmood, M. Fang, M. Pechenizkiy, and D. C. Mocanu. Madi: Learning to mask distractions for generalization in visual deep reinforcement learning. *arXiv preprint arXiv:2312.15339*, 2023.

[29] S. Yang, Y. Ze, and H. Xu. Movie: Visual model-based policy adaptation for view generalization. *Advances in Neural Information Processing Systems*, 36, 2024.

[30] S. Y. Lee, C. Sungik, and S.-Y. Chung. Sample-efficient deep reinforcement learning via episodic backward update. *Advances in neural information processing systems*, 32, 2019.

[31] K. Wang, B. Kang, J. Shao, and J. Feng. Improving generalization in reinforcement learning with mixture regularization. *Advances in Neural Information Processing Systems*, 33:7968–7978, 2020.

[32] N. Hansen and X. Wang. Generalization in reinforcement learning by soft data augmentation. In *International Conference on Robotics and Automation*, 2021.

[33] H. Zhang and Y. Guo. Generalization of reinforcement learning with policy-aware adversarial data augmentation. *arXiv preprint arXiv:2106.15587*, 2021.

[34] Y. Huang, P. Peng, Y. Zhao, G. Chen, and Y. Tian. Spectrum random masking for generalization in image-based reinforcement learning. *Advances in Neural Information Processing Systems*, 35:20393–20406, 2022.

[35] Z. Wang, Y. Ze, Y. Sun, Z. Yuan, and H. Xu. Generalizable visual reinforcement learning with segment anything model. *arXiv preprint arXiv:2312.17116*, 2023.

[36] E. Teoh, S. Patidar, X. Ma, and S. James. Green screen augmentation enables scene generalisation in robotic manipulation. *arXiv preprint arXiv:2407.07868*, 2024.

[37] I. Akinola, J. Varley, and D. Kalashnikov. Learning precise 3d manipulation from multiple uncalibrated cameras. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4616–4622. IEEE, 2020.

[38] Y. Seo, J. Kim, S. James, K. Lee, J. Shin, and P. Abbeel. Multi-view masked world models for visual robotic manipulation. In *International Conference on Machine Learning*, pages 30613–30632. PMLR, 2023.

[39] T. Gervet, Z. Xian, N. Gkanatsios, and K. Fragkiadaki. Act3d: Infinite resolution action detection transformer for robotic manipulation. *arXiv preprint arXiv:2306.17817*, 2023.

[40] A. Goyal, V. Blukis, J. Xu, Y. Guo, Y.-W. Chao, and D. Fox. Rvt-2: Learning precise manipulation from few demonstrations. *arXiv preprint arXiv:2406.08545*, 2024.

[41] S. Qian, K. Mo, V. Blukis, D. F. Fouhey, D. Fox, and A. Goyal. 3d-mvp: 3d multiview pretraining for robotic manipulation. *arXiv preprint arXiv:2406.18158*, 2024.

[42] Z. Yuan, T. Wei, S. Cheng, G. Zhang, Y. Chen, and H. Xu. Learning to manipulate anywhere: A visual generalizable framework for reinforcement learning. *arXiv preprint arXiv:2407.15815*, 2024.

[43] A. Szot, A. Clegg, E. Undersander, E. Wijmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. S. Chaplot, O. Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. *Advances in neural information processing systems*, 34:251–266, 2021.

[44] R. Jangir, N. Hansen, S. Ghosal, M. Jain, and X. Wang. Look closer: Bridging egocentric and third-person views with transformers for robotic manipulation. *IEEE Robotics and Automation Letters*, 7(2):3046–3053, 2022.

[45] H. Su, S. Maji, E. Kalogerakis, and E. 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.

[46] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European conference on computer vision (ECCV)*, pages 767–783, 2018.

[47] S. Borse, M. Klingner, V. R. Kumar, H. Cai, A. Almuzairee, S. Yogamani, and F. 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.

[48] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Q. Yu, and J. Dai. Bevformer: Learning bird's-eye-view representation from lidar-camera via spatiotemporal transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[49] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

[50] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[51] M. Li, L. Wu, J. Wang, and H. Bou Ammar. Multi-view reinforcement learning. *Advances in neural information processing systems*, 32, 2019.

[52] H. Yang, D. Shi, G. Xie, Y. Peng, Y. Zhang, Y. Yang, and S. Yang. Self-supervised representations for multi-view reinforcement learning. In *The 38th Conference on Uncertainty in Artificial Intelligence*, 2022.

[53] B. Eysenbach, A. Gupta, J. Ibarz, and S. Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

[54] S. Park, O. Rybkin, and S. Levine. Metra: Scalable unsupervised rl with metric-aware abstraction. *arXiv preprint arXiv:2310.08887*, 2023.

[55] M. Vasco, H. Yin, F. S. Melo, and A. Paiva. How to sense the world: Leveraging hierarchy in multimodal perception for robust reinforcement learning agents. *arXiv preprint arXiv:2110.03608*, 2021.

[56] E. S. Hu, J. Springer, O. Rybkin, and D. Jayaraman. Privileged sensing scaffolds reinforcement learning. *arXiv preprint arXiv:2405.14853*, 2024.

[57] S. Skand, B. Pandit, C. Kim, L. Fuxin, and S. 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.

[58] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 1998.

[59] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

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

[61] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[62] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[63] Y. Seo, D. Hafner, H. Liu, F. Liu, S. James, K. Lee, and P. Abbeel. Masked world models for visual control. In *Conference on Robot Learning*, pages 1332–1344. PMLR, 2023.

[64] T. Lin, K. Sachdev, L. Fan, J. Malik, and Y. Zhu. Sim-to-real reinforcement learning for vision-based dexterous manipulation on humanoids. *arXiv preprint arXiv:2502.20396*, 2025.

[65] H. Hu, S. Mirchandani, and D. Sadigh. Imitation bootstrapped reinforcement learning. *arXiv preprint arXiv:2311.02198*, 2023.

[66] G. Xu, R. Zheng, Y. Liang, X. Wang, Z. Yuan, T. Ji, Y. Luo, X. Liu, J. Yuan, P. Hua, et al. Drm: Mastering visual reinforcement learning through dormant ratio minimization. *arXiv preprint arXiv:2310.19668*, 2023.

[67] N. Hansen, H. Su, and X. Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations (ICLR)*, 2024.

[68] S. Fujimoto, P. D'Oro, A. Zhang, Y. Tian, and M. Rabbat. Towards general-purpose model-free reinforcement learning. *arXiv preprint arXiv:2501.16142*, 2025.

[69] B. Van Hoorick, R. Wu, E. Ozguroglu, K. Sargent, R. Liu, P. Tokmakov, A. Dave, C. Zheng, and C. Vondrick. Generative camera dolly: Extreme monocular dynamic novel view synthesis. In *European Conference on Computer Vision*, pages 313–331. Springer, 2024.

[70] P. Lancaster, N. Hansen, A. Rajeswaran, and V. Kumar. Modem-v2: Visuo-motor world models for real-world robot manipulation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7530–7537. IEEE, 2024.

[71] S. Jang, H. Jeong, and H. Yang. Murm: utilization of multi-views for goal-conditioned reinforcement learning in robotic manipulation. *Robotics*, 12(4):119, 2023.

A Experiment Setup

A.1 Hyper-parameters

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

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

A.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.

A.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. [63]. 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 Camera Scalability Experiments

B.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.

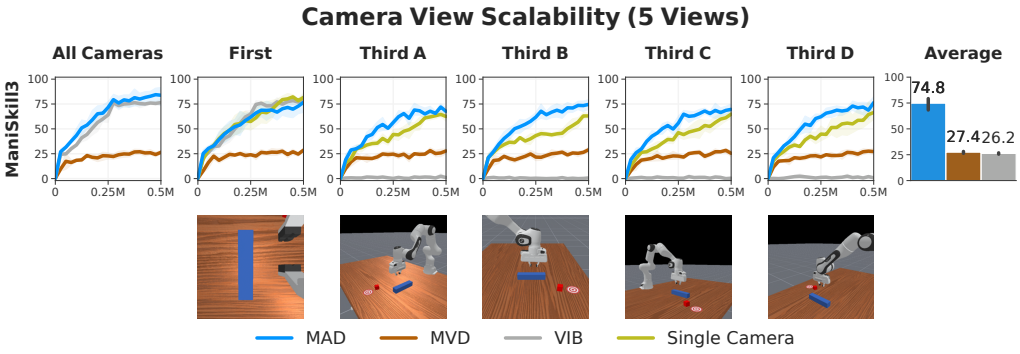


Figure 7. **View Robustness.** Success rate as a function of environment steps, averaged over 5 ManiSkill3 visual RL tasks. Methods are trained on all five camera views and evaluated on all and singular camera views with the final average displayed on the far right. Mean and 95% CI over 5 random seeds.

B.2 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.

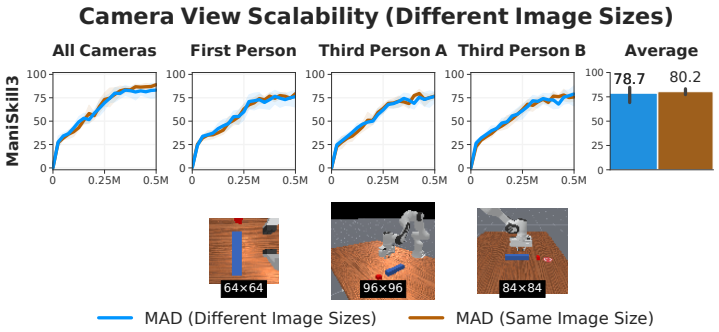


Figure 8. **View Robustness.** Success rate as a function of environment steps, averaged over 5 ManiSkill3 visual RL tasks. Methods are trained on all three camera views and evaluated on all and singular camera views with the final average displayed on the far right. Mean and 95% CI over 5 random seeds.

C Extended Analysis

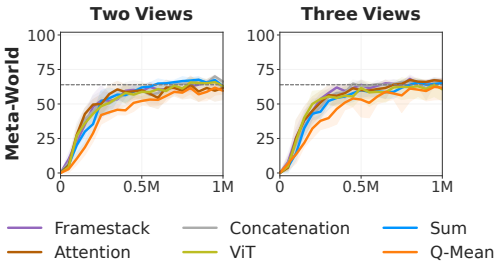


Figure 9. **Multi-view Merging.** Success rate as a function of environment steps averaged over 5 Meta-World hard visual RL tasks. Methods were trained and evaluated on *(Left)* two camera views - First and Third A - *(Right)* three camera views. Dashed line indicates highest performance of singular view baselines. Mean and 95% CI over 5 random seeds.

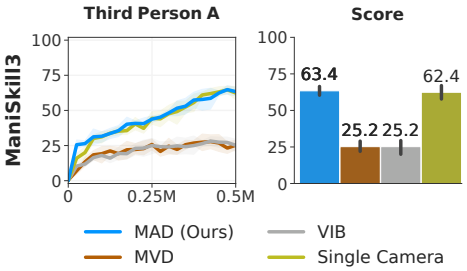


Figure 10. **Occlusion Adaptability.** Success rate as a function of environment steps, averaged over 5 ManiSkill3 tasks. Methods are trained on all three views, with two views fully occluded and only Third Person A being observable. Mean and 95% CI over 5 random seeds.

How do different merge methods compare in visual RL? 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 [1, 62]. For middle merging we include: 2) Concat, where singular views are passed through the encoder and their features are then concatenated [6] 3) Sum (Ours), where singular view features are summed instead of being concatenated [70], 4) Attention, where cross attention is used between singular view features [44], 5) ViT, where singular view features are passed through a ViT layer to leverage attention across views [38]. 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 [37, 71]. We train these merge strategies using our DrQ baseline on two views and on three views and display the results in Figure 9. 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.

How well does MAD adapt with occluded views? 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 10. 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 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.

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. [6]. 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 11, 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.

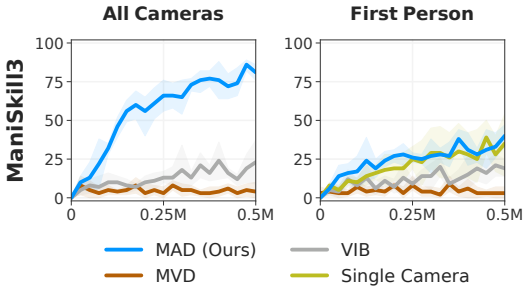


Figure 11. **First Person Camera View Failure.** Success rate as a function of environment steps on the ManiSkill3 PokeCube task. Methods are trained on all three views and evaluated an all and singular views separately. Mean and 95% CI over 5 random seeds.

D Extended Results

D.1 Overall Robustness

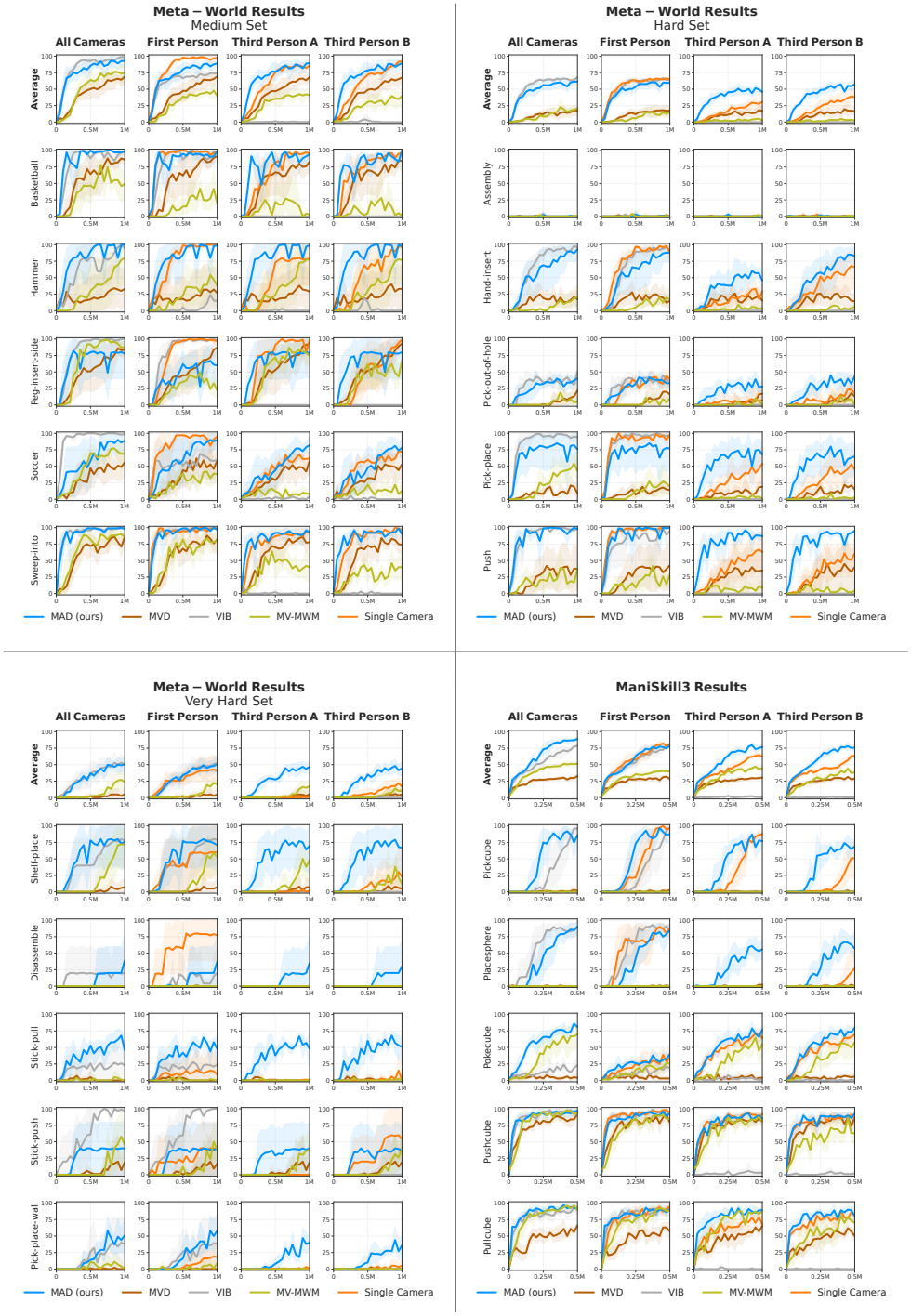


Figure 12. **Overall Robustness.** Success rate as a function of environment steps, averaged over 15 Meta-World and 5 ManiSkill3 visual RL tasks. *(Top Left)* Meta-World Medium. *(Top Right)* Meta-World Hard. *(Bottom Left)* Meta-World Very Hard. *(Bottom Right)* ManiSkill3. Methods are trained on all three camera views and evaluated on all and singular camera views. Mean and 95% CI over 5 random seeds. In reference to Figure 4 in the main text.

D.2 Ablations

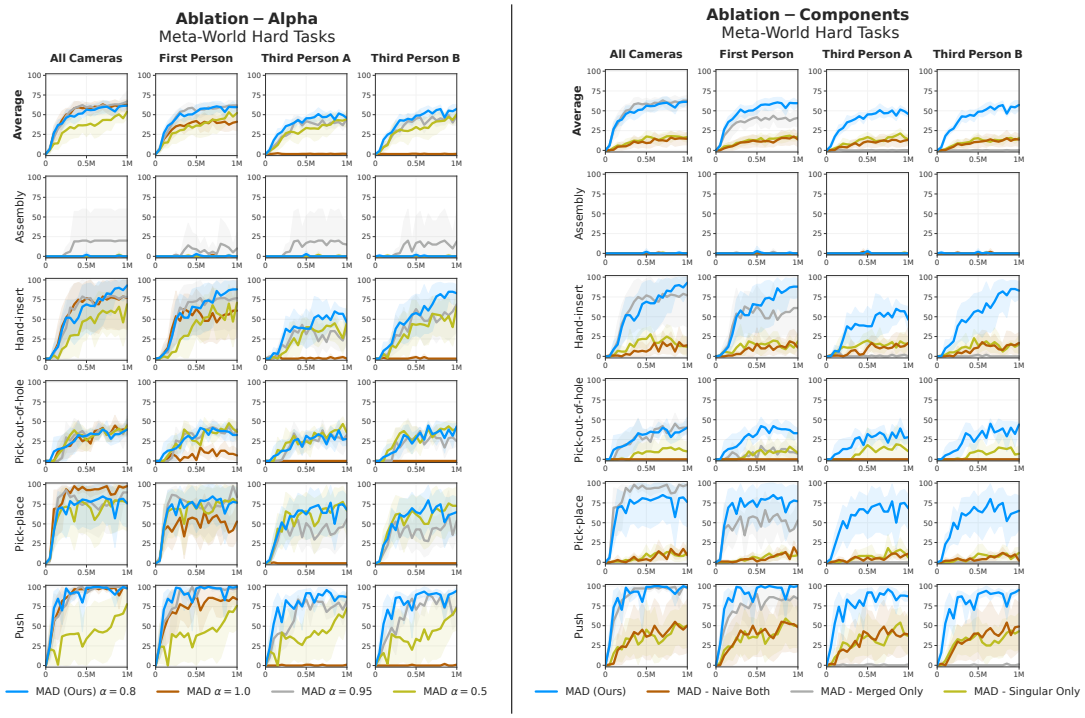


Figure 13. **Ablations.** Success rate as a function of environment steps, averaged over 5 Meta-World hard visual RL tasks. *(Left)* Alpha Ablations. *(Right)* Component Ablations. Methods are trained on all three camera views and evaluated on all and singular camera views. Mean and 95% CI over 5 random seeds. In reference to Figure 5 in the main text.

D.3 Multi-View Merging

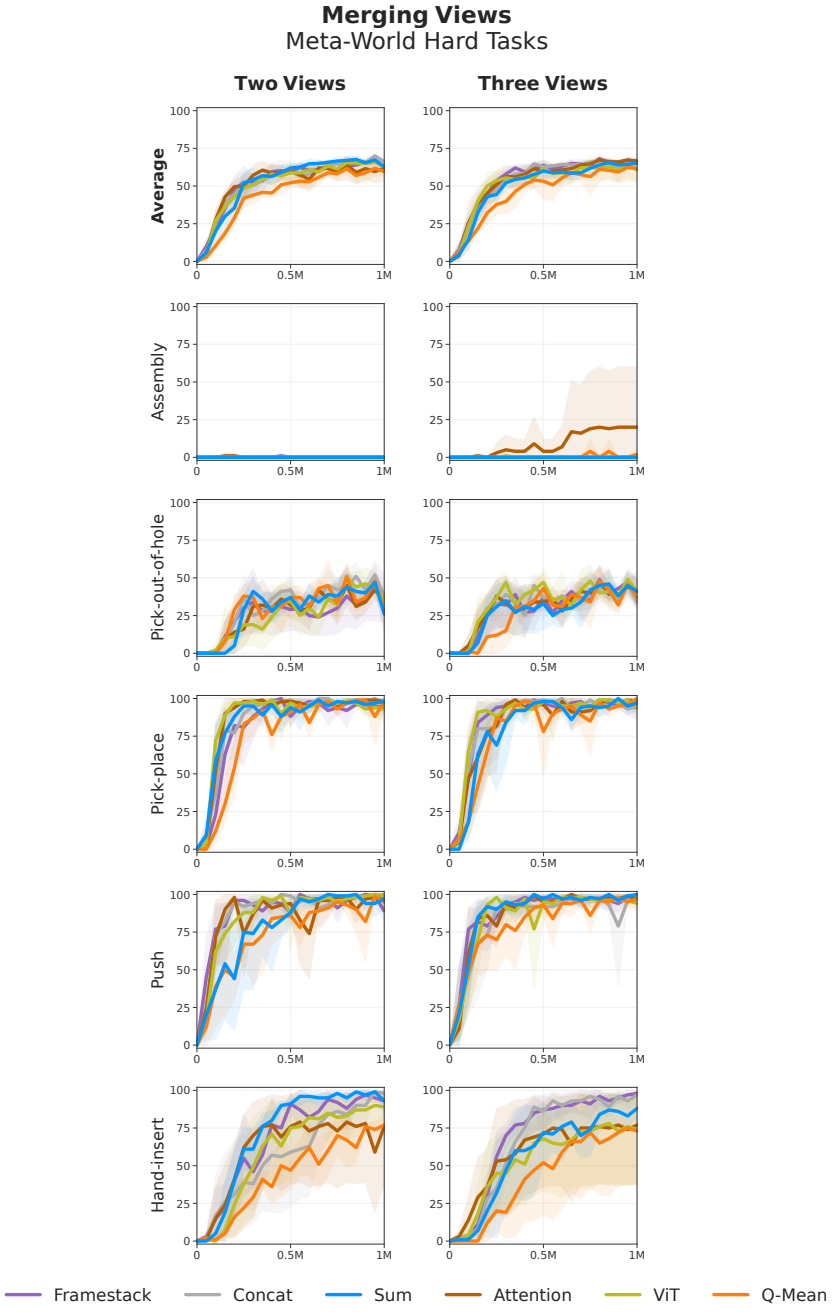
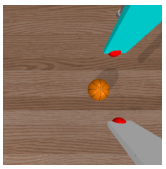
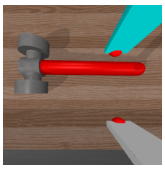
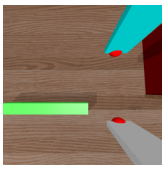
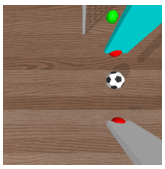
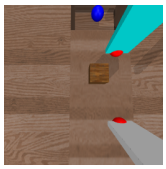
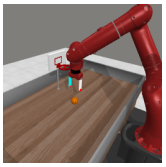
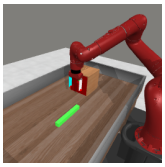
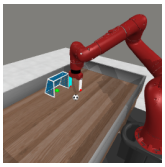
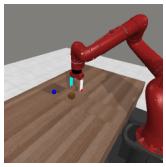
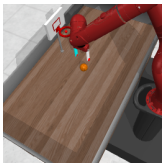
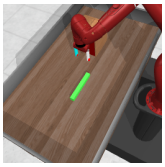
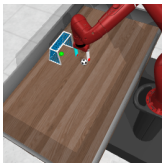
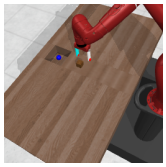


Figure 14. **Multi-view Merging.** Success rate as a function of environment steps, averaged over 5 Meta-World hard visual RL tasks. Methods trained and evaluated on *(Left)* two camera views - First and Third A - *(Right)* three camera views. Mean and 95% CI over 5 random seeds. In reference to Figure 9 in the main text.

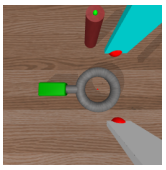
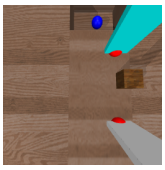
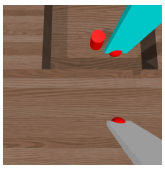
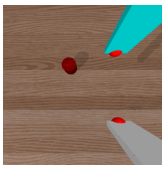
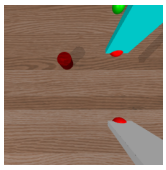
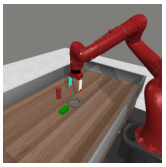
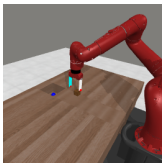
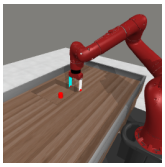
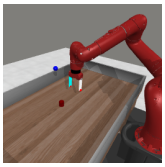
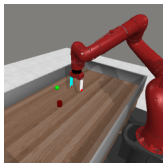
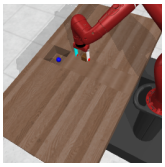
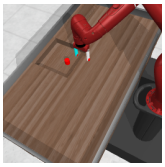
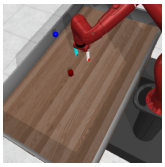
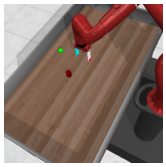
E Visuals

E.1 Meta-World

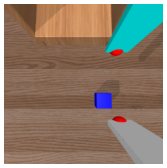
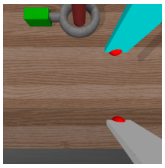
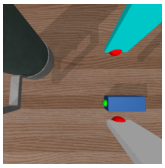
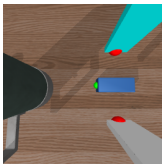
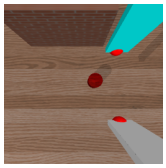
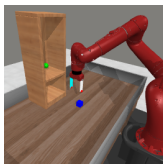
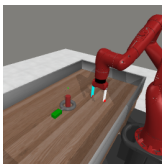
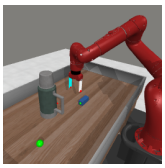
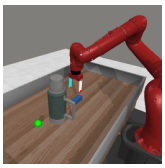
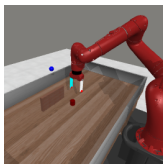
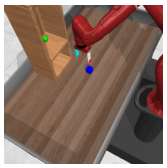
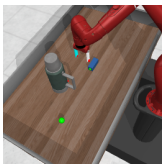
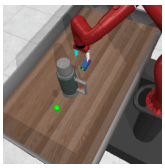
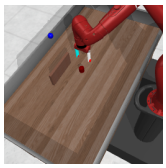
Medium Tasks

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

Hard Tasks

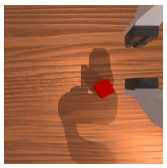
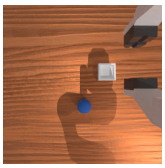
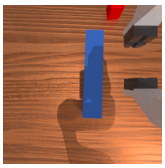
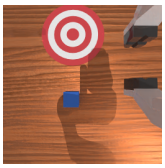
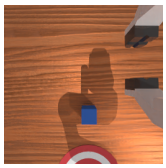
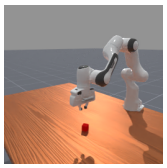
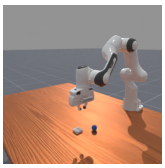
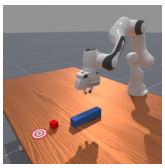
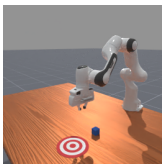
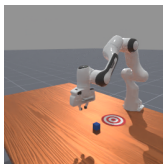
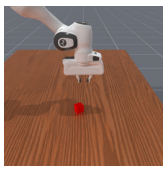
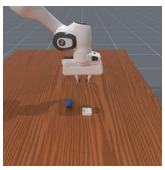
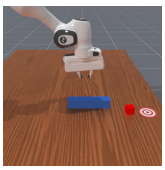
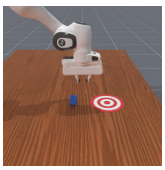
View	Assembly	Hand Insert	Pick Out Of Hole	Pick Place	Push
First View					
Third View A					
Third View B					

Very Hard Tasks

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

E.2 ManiSkill3

ManiSkill3 Tasks

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