Control First, Robustness Next: Decoupled Representation Learning for Visual RL Generalization

Chanyong Heo[1] **Hyelyn Jeong**[1] **Jongchan Park**[1]
Seungjun Oh[1] Yusung Kim[1,2]

Abstract

Visual reinforcement learning (RL) is promising for real-world problems such as autonomous driving, robotic locomotion, and manipulation, but remains highly vulnerable to visual distribution shifts between training and deployment. Existing methods improve robustness through Q-consistency regularization, masking, and auxiliary objectives, yet most of them follow a coupled representation learning paradigm in which control-oriented and robustness-oriented representation learning are jointly optimized within the same training framework. We argue that this coupled structure can introduce harmful interference between control learning and robustness learning, degrading original-environment control performance and limiting generalization under visual shifts. To address this issue, we propose **Separate Then Align Representations (STAR)**, which reformulates visual RL generalization as a problem of decoupled representation learning. Specifically, we first learn a control-relevant reference representation through online RL under weak augmentation, and then perform an offline post-representation alignment stage that maps strongly augmented inputs back to this reference representation. Experiments on RL-ViGen benchmarks spanning DMControl and Robosuite show that **STAR** achieves strong generalization under visual distribution shifts while better preserving performance in the original environment than prior coupled approaches. Our project page is available at https://mark-98.github.io/star-project-page/

1 Introduction

Visual reinforcement learning (visual RL) has emerged as a promising approach for learning control policies directly from image observations, with applications in autonomous driving, robotic locomotion, and manipulation [1]. However, visual RL policies often suffer from severe performance degradation when the visual distribution changes between training and evaluation, such as changes in background, lighting, object materials, or textures [2, 3]. Since such visual variations commonly occur in real-world environments even when the underlying task remains unchanged, robust generalization under visual distribution shift is essential for the practical deployment of visual RL.

To improve visual robustness, prior work has explored data augmentation, Q-consistency regularization, saliency- or masking-based methods, and various auxiliary objectives [4, 5, 3, 6, 7, 8]. Although these methods differ in their specific designs, they commonly optimize robustness-oriented signals together with policy and value learning within the same online training process [9, 10, 11]. We refer to this as a coupled representation learning paradigm, where control-oriented representation learning and robustness-oriented representation learning are co-optimized and directly shape the same learned representation. Even when weakly augmented observations are used for the RL objective and strongly

[1]Department of Artificial Intelligence, Sungkyunkwan University, Suwon, Republic of Korea.
[2]Department of Computer Science and Engineering, Sungkyunkwan University, Suwon, Republic of Korea.
Correspondence to: Yusung Kim (yskim525@skku.edu).

augmented observations are used for consistency or auxiliary objectives, the resulting representation is still influenced by both control and robustness signals during policy optimization.

We argue that this coupling is a central bottleneck in robust visual RL. Strong perturbations and robustness objectives introduced during policy optimization may suppress not only nuisance visual variations but also task-relevant cues required for accurate policy and value estimation. As a result, objectives intended to improve visual robustness can interfere with the formation of control-relevant representations, destabilizing RL training and degrading performance even in the original environment. This issue is particularly critical in manipulation tasks, where subtle geometric structures, object relations, and contact-relevant visual cues can significantly affect action values [1, 12]. Thus, the key challenge is not merely to design stronger augmentations or additional auxiliary losses, but to determine how control learning and robustness learning should be structured.

While most prior methods optimize control learning and robustness learning jointly, related work such as SECANT [13] has explored staged robustification after initial policy learning. However, it improves robustness through additional online policy-distillation training. In contrast, we aim to robustify a trained control policy purely at the representation level, without additional environment interaction or actor-critic retraining.

To this end, we propose **Separate Then Align Representations (STAR)**, a decoupled representation learning framework for visual RL generalization. STAR first learns a control-relevant representation through standard online RL under weak augmentation, without applying strong perturbation-based robustness objectives during policy optimization. It then freezes the learned control encoder as a fixed reference and trains a robust encoder offline to map strongly augmented observations back to this reference representation space. In this way, STAR turns robustification from a moving-target alignment problem during online RL into a fixed-target representation recovery problem after control learning. Because the robustness objective is defined over latent representations rather than policy-specific action outputs, STAR can serve as a modular post-hoc robustification method for pretrained image-based control models, including visual RL agents and imitation-based visuomotor policies such as Diffusion Policy.

We evaluate the proposed framework on DMControl and Robosuite environments in the RL-ViGen benchmark [14]. Our results show that the proposed decoupled framework better preserves original-environment performance while improving robustness under visual distribution shifts. We further demonstrate that the same representation-level alignment principle can improve the visual robustness of diffusion policies trained from limited demonstration data.

Our contributions are summarized as follows:

- We empirically demonstrate that coupling robustness representation learning with online control learning is associated with increased representation drift and TD residuals, indicating instability in both representations and critic learning.
- We propose STAR, a decoupled visual RL framework that separates control learning from robustness learning by performing offline alignment of strongly perturbed observations to a fixed control-relevant reference representation learned through online RL.
- We demonstrate across DMControl and Robosuite that STAR improves visual generalization while better preserving original-environment performance, and further validate its extensibility to pretrained imitation-based visuomotor policies such as Diffusion Policy.

2 Background

2.1 Visual reinforcement learning

We consider a control problem defined by an underlying Markov decision process [15] $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P(s_{t+1} \mid s_t, a_t)$ is the transition probability, $r(s_t, a_t)$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor. The goal is to learn a policy π that maximizes the expected discounted return $J(\pi) = \mathbb{E}_\pi[\sum_{t=0}^{T} \gamma^t r(s_t, a_t)]$.

In visual reinforcement learning, the agent does not observe the underlying state s_t directly. Instead, it receives an image observation, where the same control state may appear differently due to background, lighting, texture, or camera noise. In this work, we consider the setting where policy performance

degrades under visual distribution shifts even though the underlying control dynamics remain nearly unchanged. This happens because visually different observations of the same control state can be encoded into inconsistent representations. Our goal is therefore to learn control-consistent representations that remain stable under such visual changes.

2.2 Soft actor-critic training

Our visual RL backbone follows the DrQ training scheme, which is based on Soft Actor-Critic (SAC) with weak image augmentation [5, 16]. Given a latent representation $z_t = E(o_t)$, the critic is trained using the soft Bellman target

$$y_t = r_t + \gamma \left(Q_{\bar{\theta}}(z_{t+1}, a_{t+1}) - \alpha \log \pi_\phi(a_{t+1} \mid z_{t+1}) \right), \qquad a_{t+1} \sim \pi_\phi(\cdot \mid z_{t+1}), \tag{1}$$

and the critic loss is

$$\mathcal{L}_{\text{critic}} = \mathbb{E}_{(o_t, a_t, r_t, o_{t+1}) \sim \mathcal{B}} \left[\left(Q_\theta(z_t, a_t) - y_t \right)^2 \right]. \tag{2}$$

The actor is trained with the entropy-regularized objective

$$\mathcal{L}_{\text{actor}} = \mathbb{E}_{o_t \sim \mathcal{B},\, a_t \sim \pi_\phi(\cdot \mid z_t)} \left[\alpha \log \pi_\phi(a_t \mid z_t) - Q_\theta(z_t, a_t) \right]. \tag{3}$$

2.3 Weak and strong visual augmentations

We use two types of visual augmentations. A *weak augmentation* f_{weak} denotes random shift, a standard augmentation in visual RL that preserves most task-relevant visual content [5]. A *strong augmentation* f_{strong} denotes random overlay, which blends the observation with an external image [17] and introduces larger appearance changes in background, texture, and color [3]. Specifically, given an observation o and an external overlay image b, random overlay constructs

$$\tilde{o} = (1 - \alpha) o + \alpha b, \qquad \alpha \sim \text{Uniform}(0, 1). \tag{4}$$

In our framework, f_{weak} is used during online control learning, while f_{strong} is used to construct perturbed observations for post-representation alignment.

3 Motivation

Many visual RL generalization methods jointly optimize control learning and robustness-oriented representation learning. However, the agent must learn both what is relevant for control and what should remain invariant to visual perturbations from scratch. This joint learning can be challenging because the two objectives are optimized while influencing the same representation. In actor-critic RL, the control representation is formed together with the critic through bootstrapped updates, while robustness objectives simultaneously encourage invariance through consistency, regularization, or auxiliary losses [6, 7, 9, 10, 11].

Following standard visual RL practice, the visual encoder is updated through the critic loss and kept frozen during actor updates. Therefore, the learned representation is shaped primarily by the critic-side objective [5, 6].

Representation and TD-learning changes under coupled learning Figure 1 compares source-domain performance and learning dynamics under coupled robustness learning. We measure *representation shift* as the change in the latent embedding of the same observations over training, and *normalized TD error* as the mismatch between the current Q-value and its Bellman target relative to the target magnitude. DrQ [5] achieves higher average source-domain return than SVEA [6], SMG [9], and SimGRL [10] on both DMC and Robosuite.

The coupled methods also exhibit larger representation shifts and normalized TD errors than DrQ. On DMC, the two quantities reach 1.24–$2.05\times$ and 1.34–$1.65\times$ the DrQ levels, respectively; on Robosuite, they increase up to $4.54\times$ and $5.57\times$. Thus, the differences under coupled learning are reflected not only in the learned representation but also in TD learning.

A key difficulty is that robustness is optimized while the control representation is still evolving through online RL. Representations induced by strongly augmented observations must therefore adapt both to visual perturbations and to a changing control representation. This motivates a simple principle: *control first, robustness next*. STAR first establishes the representation needed for control and then learns robustness relative to this fixed representation.

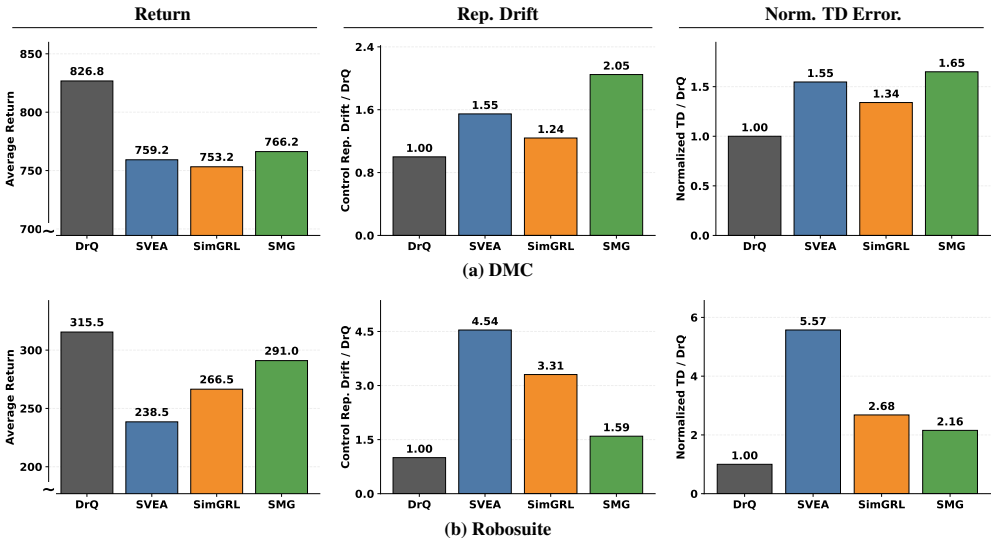


Figure 1: From left to right: in-distribution return, representation shift, and normalized TD Error. Representation shift and normalized TD Error are reported relative to the corresponding DrQ value ($= 1.0$).

4 Method

STAR consists of two stages. Let E_C denote the encoder that produces the latent representation used for control. In Stage 1, E_C is trained through DrQ-style online RL with an auxiliary forward-dynamics objective to learn a control-relevant reference representation.

After Stage 1, the reference encoder and downstream controller are frozen. In Stage 2, a robust encoder E_R, initialized from E_C, and an alignment module g_ϕ are trained offline to recover the fixed reference representation from strongly augmented observations. Only the robust visual pathway is updated during this stage.

We refer to this Stage-2 procedure as *Post-Representation Alignment* (PRA), while STAR denotes the complete two-stage framework.

4.1 Stage 1: control-oriented reference learning

In Stage 1, we train a visual RL agent in the source environment following the DrQ-style [5] training pipeline. We do not apply strong perturbation-based robustness objectives during this stage. The purpose of Stage 1 is to learn a stable control representation that can later serve as the reference for post-representation alignment.

Let E_C denote the encoder, including the visual encoder and projection module. Given an observation o_t, the encoder produces a latent representation

$$z_t^C = E_C(o_t),$$

which is used by both the actor and critic for policy and value learning.

In addition to the standard actor-critic losses, we add a lightweight forward dynamics objective to encourage the latent representation to capture action-conditioned transition information given a transition (o_t, a_t, o_{t+1}), the dynamics model predicts the next latent representation:

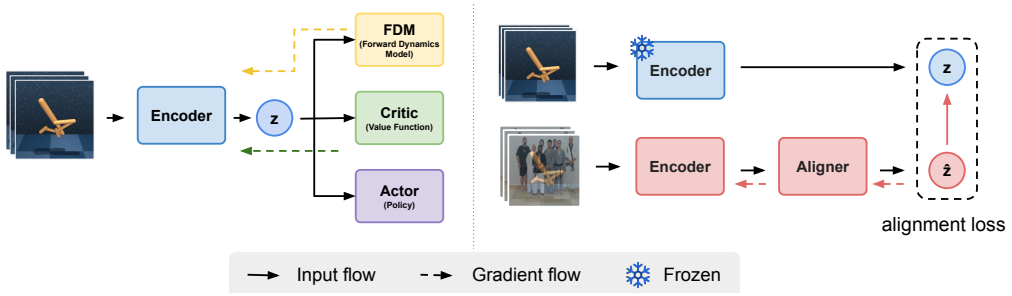
$$\mathcal{L}_{\text{dyn}} = \mathbb{E}_{\mathcal{B}} \left[\left\| F_\psi(z_t^C, a_t) - \text{sg}[z_{t+1}^C] \right\|_2^2 \right], \tag{5}$$

where $z_{t+1}^C = E_C(o_{t+1})$ and $\text{sg}[\cdot]$ denotes stop-gradient.

The Stage 1 representation objective is

$$\mathcal{L}_{\text{ctrl-rep}} = \mathcal{L}_{\text{critic}} + \lambda \mathcal{L}_{\text{dyn}}. \tag{6}$$

After Stage 1, E_C is frozen, and its output z_t^C is used as the target reference latent.



(a) Stage 1: online control-oriented learning. (b) Stage 2: offline post-representation alignment.

Figure 2: Overview of the proposed two-stage framework. Stage 1 learns the reference latent z using forward dynamics and critic objectives to capture transition- and value-relevant information. Stage 2 freezes the pretrained actor and critic and aligns the shifted latent $\hat{z}$ with z.

4.2 Stage 2: offline post-representation alignment

After online RL-based control learning, we perform robustification without any additional environment interaction. We use only observations from the replay buffer $\mathcal{D}$ collected during Stage 1. The control encoder E_C is frozen and serves as the reference encoder, while a robust encoder E_R is initialized from E_C and updated during alignment.

For each observation $o_t \sim \mathcal{D}$, we construct a weakly augmented view $o_t^{\mathrm{w}} = f_{\mathrm{weak}}(o_t)$ and a strongly perturbed view $\tilde{o}_t = f_{\mathrm{strong}}(o_t)$. The weak augmentation uses the same random shift as in Stage 1, while the strong augmentation introduces diverse visual perturbations. We obtain a reference representation from the weak view using the frozen control encoder E_C, and an aligned robust representation from the strongly perturbed view using E_R and g_ϕ:

$$z_t^{\mathrm{ref}} = \mathrm{sg}[E_C(o_t^{\mathrm{w}})]\,, \quad z_t^R = g_\phi(E_R(\tilde{o}_t))\,, \tag{7}$$

where $\mathrm{sg}[\cdot]$ denotes stop-gradient and g_ϕ is a lightweight alignment module. We train E_R and g_ϕ to align the robust representation with the fixed control-relevant reference by minimizing

$$\mathcal{L}_{\mathrm{align}} = \mathbb{E}_{o_t \sim \mathcal{D}}\left[\left\| z_t^R - z_t^{\mathrm{ref}} \right\|_2^2\right]. \tag{8}$$

For the strong augmentation f_{strong}, we use the random overlay defined in Eq. (4). Since the mixing coefficient α is sampled for each observation; the robust encoder is exposed to a range of perturbation strengths, from mild appearance changes to severe visual corruptions. This avoids relying on a single fixed overlay ratio and encourages the robust encoder to align perturbed observations of varying difficulty to the same reference latent.

During Stage 2, only the robust encoder E_R and the alignment module g_ϕ are updated. The reference encoder E_C, actor, critic, and optional forward dynamics module remain fixed. Thus, robustification is performed purely at the representation level: strongly perturbed observations are mapped back to the fixed control-relevant latent learned during Stage 1, without policy or critic retraining.

4.3 Inference with the Fixed Controller

As illustrated in Figure 2(b), Stage 2 trains the robust encoder E_R and alignment module g_ϕ to recover the latent representation produced by the Stage-1 reference encoder E_C from strongly augmented observations. At evaluation time, the robust pathway $g_\phi \circ E_R$ replaces E_C, while the learned controller remains fixed.

At inference, an observation o_t is mapped by the robust pathway to the aligned representation $z_t^R = g_\phi(E_R(o_t))$, which is passed directly to the frozen controller. The reference encoder E_C is used only to construct alignment targets during Stage 2 and is not required at inference.

5 Experiments

5.1 Evaluation questions

We evaluate the proposed *decoupled representation learning* framework from three complementary perspectives. Specifically, our experiments are designed to answer the following questions.

Q1. Effectiveness of decoupled representation learning. Can the proposed decoupled representation learning framework improve visual robustness under distribution shifts while preserving performance in the original environment?

Q2. Performance on contact-rich manipulation. Does the proposed method achieve stronger performance than prior robust visual RL baselines on contact-rich manipulation tasks?

Q3. Roles of Dynamics Supervision and Stage-2 Alignment. What are the respective contributions of Stage-1 dynamics supervision and Stage-2 post-representation alignment?

Q4. Policy-class extensibility. Can the proposed method be applied not only to pretrained visual RL policies but also to other policy classes, including imitation-based policies such as Diffusion Policy?

Together, these questions allow us to assess whether the proposed decoupled representation learning framework can effectively improve visual robustness, remain competitive on contact-rich manipulation tasks, and extend to diverse policy classes.

5.2 Experimental setup

5.2.1 Baselines

We compare **STAR** with representative robust visual RL baselines. **DrQ** [5] is the vanilla baseline, using random shift without an explicit robustness objective. **SVEA** [6], **SGQN** [7], **SMG** [9], and **SimGRL** [10] introduce robustness-oriented objectives during online RL, such as value consistency, saliency guidance, reconstruction or feature consistency, and strong visual augmentation. We also compare with **SECANT** [13], a staged robustification baseline that first trains a teacher policy and then learns a robust student policy in a second stage.

5.2.2 Benchmarks and metrics

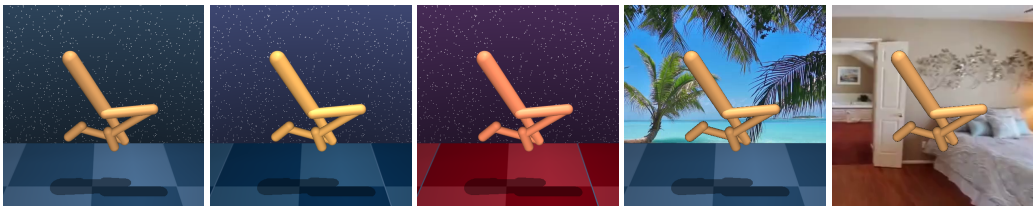
We evaluate **STAR** on the RL-ViGen benchmark [14], which tests visual generalization under distribution shifts such as changes in background, color, texture, lighting, and object appearance. We consider two environment groups: DMC-GB, based on the DeepMind Control Suite [18], and Robosuite manipulation tasks [19].

For online RL experiments, we evaluate on DMC-GB tasks including *walker walk*, *cheetah run*, *finger spin*, and *cartpole swingup*, as well as Robosuite tasks including *Door* and *TwoArmPegInHole*. These policies are trained in the original environment and evaluated zero-shot under visual distribution shifts. We report episodic return as the evaluation metric.

To evaluate the extensibility of representation-level alignment beyond online RL policies, we also consider pretrained imitation-based visuomotor policies. Specifically, we train Diffusion Policy models [20] from expert demonstrations on two Robosuite tasks, *PickPlaceCan* and *NutAssemblySquare*, and apply our offline alignment objective to their visual representations. For these diffusion policy experiments, we report the task success rate under visual distribution shifts.

5.3 Q1: Effectiveness of decoupled representation learning

Overall, **STAR** not only achieves the best performance across all four DMC-GB tasks in the Original setting, but also shows strong zero-shot generalization under visual distribution shifts. Across the shifted settings, it achieves the best or competitive performance in most cases. The gains are particularly clear in tasks where **STAR** learns a stronger source-domain policy. For example, in *Walker Walk* and *Cheetah Run*, **STAR** shows relatively large improvements in the Original setting, and this advantage often carries over to shifted environments such as Color Easy, Color Hard, and



(a) Training (b) Color-easy (c) Color-hard (d) Video-easy (e) Video-hard

Figure 3: Examples of training and evaluation observations for Walker Walk in RL-ViGen DMC-GB. Color Easy and Color Hard change the scene appearance. Video Easy and Video Hard introduce video backgrounds; Video Hard also removes the floor and the walker's shadow.

Video Easy. This suggests that learning a stronger control-relevant reference representation in Stage 1 can provide a better target for subsequent representation alignment.

While **STAR** is not uniformly best under every shift, it remains highly competitive. In a few cases, such as *Finger Spin* and *Cartpole SwingUp* under Video Easy or *Cheetah Run* under Video Hard, it is slightly behind the best baseline. Overall, however, **STAR** preserves strong Original performance while achieving robust zero-shot generalization across diverse DMC-GB shifts.

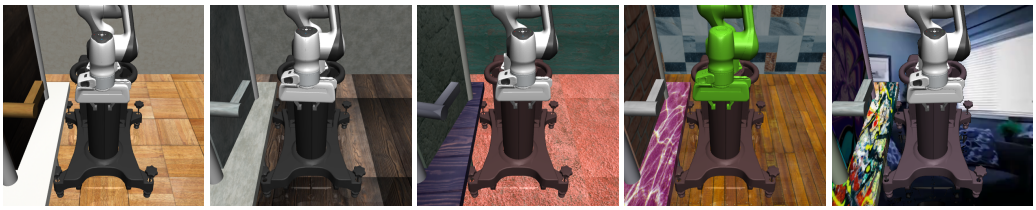
Table 1: Performance comparison on RL-ViGen DMC-GB. We report mean episodic return $\pm$ standard deviation over four seeds. The best result in each task–setting pair is shown in bold.

Setting	Task	DrQ	SVEA	SGQN	SMG	SimGRL	SECANT	**STAR**	Δ
Original	Walker	932±19	851±78	895±34	877±93	870±78	904±21	**958±6**	+26 (3%)
	Cheetah	524±127	354±32	284±21	361±22	310±38	460±84	**615±52**	+91 (17%)
	Finger	983±2	968±24	984±1	979±3	972±22	956±21	**985±1**	+1 (1%)
	Cartpole	868±12	864±12	861±7	848±7	861±12	860±12	**872±11**	+4 (1%)
Color Easy	Walker	772±145	821±44	856±62	873±44	799±101	875±45	**939±15**	+64 (7%)
	Cheetah	378±47	313±46	253±30	358±11	302±41	423±75	**567±64**	+144 (34%)
	Finger	950±34	888±178	921±29	971±5	972±8	946±27	**975±10**	+3 (1%)
	Cartpole	770±58	828±48	788±43	843±11	860±16	858±13	**869±15**	+9 (1%)
Color Hard	Walker	615±87	781±35	718±53	806±49	682±92	800±48	**880±43**	+74 (9%)
	Cheetah	251±38	299±39	229±18	337±13	305±31	388±70	**493±91**	+105 (27%)
	Finger	813±109	879±183	853±46	912±15	907±13	**939±15**	907±41	−32 (3%)
	Cartpole	551±129	802±51	743±33	811±9	812±27	809±17	**824±34**	+12 (1%)
Video Easy	Walker	709±76	802±80	860±35	863±89	847±81	803±38	**950±8**	+87 (10%)
	Cheetah	268±62	277±25	242±3	333±33	302±42	264±76	**510±87**	+177 (53%)
	Finger	538±59	724±20	946±22	966±17	947±24	672±43	**980±7**	+14 (1%)
	Cartpole	424±49	811±17	769±60	787±18	**839±16**	721±49	837±40	−2 (1%)
Video Hard	Walker	119±56	417±96	732±56	802±103	795±97	331±34	**924±18**	+122 (15%)
	Cheetah	60±6	104±36	189±21	**310±17**	303±46	77±26	271±67	−39 (13%)
	Finger	7±4	242±40	781±93	925±39	927±17	174±28	**977±8**	+50 (5%)
	Cartpole	133±30	325±57	644±46	405±115	746±69	317±39	**828±21**	+82 (11%)
Original avg.		826.8	759.3	756.0	766.3	753.3	795.0	**857.5**	+30.7 (4%)
Shifted avg.		459.9	582.1	657.8	706.4	709.1	587.3	**795.7**	+86.6 (12%)

5.4 Q2: Performance on contact-rich manipulation

We next evaluate whether **STAR** remains effective in contact-rich manipulation tasks, where robust perception must be combined with precise object interaction. We use two Robosuite tasks from RL-ViGen, *Door* and *TwoArmPegInHole*, under a 500K online interaction budget. Compared with DMC-GB, these tasks involve more structured scene geometry and contact-sensitive control, making them a challenging testbed for visual robustness.

Table 2 reports the results. The proposed method achieves the best performance on both tasks in the Original setting, indicating that decoupled representation learning preserves strong source-domain



(a) Training (b) Easy (c) Medium (d) Hard (e) Video

Figure 4: Examples of training and evaluation observations for RL-ViGen Robosuite tasks. The panels use the Original, Easy, Medium, Hard, and Video labels adopted in this paper; their mapping to the benchmark settings is described in the text.

control performance even in manipulation environments. Under visual distribution shifts, it also obtains the best performance in 7 out of 8 shifted settings. The gains are especially large on *Door*, where our method consistently outperforms all baselines across Easy, Medium, Hard, and Video shifts.

On *TwoArmPegInHole*, the proposed method also shows strong robustness, achieving the best performance in Easy, Medium, and Video settings. The only exception is the Hard setting, where SECANT performs slightly better. Nevertheless, our method remains competitive in this case and substantially outperforms the other coupled robustness baselines.

Overall, these results show that the proposed framework is not limited to locomotion or simple continuous-control tasks. By preserving a strong control-relevant representation and aligning perturbed observations to it, our method provides robust zero-shot generalization in contact-rich manipulation environments.

Table 2: Performance comparison on RL-ViGen Robosuite manipulation tasks. We report mean episodic return ± standard deviation over four seeds. The best result in each task–setting pair is shown in bold. TwoArm denotes TwoArmPegInHole.

Setting	Task	DrQ	SVEA	SGQN	SMG	SimGRL	SECANT	**STAR**	Δ
Original	TwoArm	399±84	303±189	282±43	300±92	334±127	415±95	**473±6**	+58 (14%)
	Door	232±221	174±188	279±151	282±172	199±250	253±200	**455±32**	+173 (61%)
Easy	TwoArm	259±45	282±165	281±40	297±82	317±110	397±76	**461±13**	+64 (16%)
	Door	48±33	144±153	180±86	237±172	165±209	250±184	**440±34**	+190 (76%)
Medium	TwoArm	163±26	259±133	273±27	287±83	296±77	370±76	**397±22**	+27 (7%)
	Door	12±8	118±118	146±80	129±89	86±103	225±171	**442±38**	+217 (96%)
Hard	TwoArm	149±10	214±92	251±14	257±68	201±42	**357±79**	335±57	−22 (6%)
	Door	10±16	62±81	119±93	138±142	46±69	167±166	**202±145**	+35 (21%)
Video	TwoArm	143±24	152±20	205±4	168±13	265±87	243±25	**424±9**	+159 (60%)
	Door	2±2	40±39	100±53	51±23	24±19	47±28	**343±41**	+243 (243%)
Original avg.		315.5	238.5	280.5	291.0	266.5	334.0	**464.0**	+130.0 (39%)
Shifted avg.		98.3	158.9	194.4	195.5	175.0	257.0	**380.5**	+123.5 (48%)

5.5 Q3: Roles of Dynamics Supervision and Stage-2 Alignment

We conduct an ablation study to examine the roles of Stage-1 dynamics supervision (Dyn) and Stage-2 post-representation alignment (S2). Table 3 reveals a clear separation between their contributions. Stage 2 provides the primary gain in visual robustness. Without dynamics supervision, adding S2 increases the shifted average from 399.9 to 773.4 on DMC and from 96.0 to 262.0 on Robosuite, while leaving Original performance nearly unchanged. This shows that post-representation alignment remains effective even when the Stage-1 representation is learned without dynamics supervision.

Dynamics supervision plays a complementary role. Without S2, adding Dyn has little effect on shifted performance (399.9 vs. 399.3 on DMC and 96.0 vs. 103.8 on Robosuite), but substantially improves

Table 3: Matched ablation of Stage-1 dynamics supervision (Dyn) and Stage-2 post-representation alignment (S2). Values are average episodic returns; Shifted averages over all appearance-shift conditions in each benchmark.

		w/o S2		w/ S2	
Benchmark	Evaluation	−Dyn	+Dyn	−Dyn	+Dyn
DMC	Original	852.5	**858.0**	850.3	857.5
	Shifted	399.9	399.3	773.4	**795.7**
Robosuite	Original	319.5	**464.5**	315.5	464.0
	Shifted	96.0	103.8	262.0	**380.5**

Original performance on Robosuite from 319.5 to 464.5. When combined with S2, Dyn further increases the shifted average from 773.4 to 795.7 on DMC and from 262.0 to 380.5 on Robosuite.

These results separate the two roles in STAR: Stage 1 establishes a control-relevant reference, while Stage 2 provides visual robustness by recovering this fixed reference from strongly augmented observations. Dynamics supervision improves the learned control reference, particularly on Robosuite, whereas Stage 2 is the direct mechanism for post-hoc robustification. Full task-level results are provided in Appendix B.1.

5.6 Q4: Policy-class extensibility

The third question asks whether the proposed framework can extend beyond actor–critic-based visual RL policies to other policy classes. To this end, we apply post-representation alignment to a pretrained Diffusion Policy [20]. This experiment tests whether robustness can be improved by aligning visual representations alone, without retraining the diffusion-based action generation module.

We evaluate this setting under visual distribution shifts constructed by modifying scene appearance, such as textures, object appearance, and background visuals, while keeping the underlying task unchanged. Details of these visual modifications are provided in Appendix A.6.

Concretely, we initialize the robust visual encoder from the pretrained Diffusion Policy encoder and apply the same offline representation alignment objective.

After alignment, the robust encoder module is connected to the pretrained diffusion action module for evaluation. The results in Table 4 show that post-representation alignment improves the robustness of Diffusion Policy under visual distribution shifts.

These findings support our broader claim that robustness learning can be separated from policy learning. Since the action generation module is kept fixed, the improvement comes from aligning perturbed observations to the latent interface expected by the pretrained policy, suggesting that the proposed framework can serve as a general post-representation robustification mechanism for visuomotor policies.

Table 4: Success rates (%) of Diffusion Policy with and without post-representation alignment (PRA). Easy, Medium, and Hard denote increasing visual-shift severity.

Task	Method	Original	Easy	Medium	Hard
Can	DP	$\mathbf{100.0 \pm 0.0}$	39.3 ± 4.1	22.0 ± 4.3	14.0 ± 3.3
	DP + PRA	99.3 ± 0.9	$\mathbf{72.0 \pm 3.3}$	$\mathbf{62.0 \pm 4.3}$	$\mathbf{41.3 \pm 7.4}$
Square	DP	96.0 ± 1.6	43.3 ± 3.4	28.0 ± 1.6	15.3 ± 2.5
	DP + PRA	$\mathbf{98.0 \pm 0.0}$	$\mathbf{76.0 \pm 2.8}$	$\mathbf{74.7 \pm 5.2}$	$\mathbf{67.3 \pm 0.9}$

6 Related work

Representation learning for visual RL. Visual RL must learn control from high-dimensional image observations while improving both sample efficiency and visual generalization. Prior work

improves visual representations through auxiliary objectives, including contrastive learning methods [21, 22], dynamics-based prediction methods [23, 24, 25], and broader self-supervised objectives such as reconstruction, future prediction, or latent world modeling [26, 27]. Another line focuses on robustness to visual shifts in background, color, texture, lighting, and object appearance through augmentation, consistency regularization, masking, saliency guidance, representation separation, or auxiliary losses [5, 6, 7, 9, 10, 11]. Despite their differences, these methods generally couple representation learning, robustness, and policy/value optimization within the same training loop.

SECANT [13] is the closest prior work because it also separates policy learning and robustification into two stages. However, SECANT trains a student policy through DAgger-style [28] online imitation under strong augmentation, whereas **STAR** freezes the Stage 1 control representation and performs purely offline post-representation alignment, without rollouts and actor–critic retraining.

7 Discussion

Our ablations show that STAR's two stages serve complementary roles: Stage-1 establishes a strong control-relevant representation, while Stage-2 recovers it under visual perturbations to provide the main robustness gain. This separation enables lightweight post-hoc robustification without additional environment interaction or downstream controller retraining, and applies to independently pretrained DrQ and Diffusion Policy. These results suggest a path toward broader pretrained visuomotor policies, including Vision-Language-Action models.

Limitations. Despite the promising results, our framework has several limitations. First, the effectiveness of Stage 2 depends on the quality of the control-relevant representation learned in Stage 1. If the Stage 1 policy learns a suboptimal representation, the alignment stage may inherit this limitation rather than correct it. Second, when applied to imitation-based policies such as Diffusion Policy, post-representation alignment improves shifted-environment performance but can slightly degrade original-environment performance, suggesting a remaining trade-off between robustness and preserving the original policy behavior. Finally, following prior visual generalization studies, our scope assumes that visual shifts mainly affect the observation appearance, while the underlying control problem remains unchanged. If the environment dynamics or task semantics change, additional techniques such as domain randomization or domain adaptation may be required.

Future Work. Future work can improve both the formation and recovery of control-relevant representations. Stronger control-oriented objectives may provide better Stage-1 references, while control-aware alignment could selectively recover features or subspaces that matter most for downstream control rather than matching the entire latent uniformly. Extending this principle to broader visual shifts and larger pretrained visuomotor and Vision-Language-Action models is a promising direction.

Acknowledgements

This work was supported in part by the National Research Foundation of Korea (NRF) (Grant No. RS-2024-00348376); the Institute of Information & Communications Technology Planning & Evaluation (IITP) (Grant Nos. RS-2024-00438686, RS-2022-II221045, RS-2025-02218768, and RS-2019-II190421); the National IT Industry Promotion Agency (NIPA) (Grant No. RS-2026-25621569); and the Industrial Technology Innovation Program (Grant No. RS-2025-25448266), with funding from the Korean government through the Ministry of Science and ICT (MSIT), the Ministry of Trade, Industry and Energy (MOTIE), and the Korea Evaluation Institute of Industrial Technology (KEIT).

References

[1] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40, 2016.

[2] Xingyou Song, Yiding Jiang, Stephen Tu, Yilun Du, and Behnam Neyshabur. Observational overfitting in reinforcement learning. In *International Conference on Learning Representations*, 2020.

[3] Nicklas Hansen and Xiaolong Wang. Generalization in reinforcement learning by soft data augmentation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13611–13617. IEEE, 2021.

[4] Misha Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. *Advances in neural information processing systems*, 33:19884–19895, 2020.

[5] Denis Yarats, Ilya Kostrikov, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International Conference on Learning Representations*, 2021.

[6] Nicklas Hansen, Hao Su, and Xiaolong Wang. Stabilizing deep q-learning with convnets and vision transformers under data augmentation. *Advances in neural information processing systems*, 34:3680–3693, 2021.

[7] David Bertoin, Adil Zouitine, Mehdi Zouitine, and Emmanuel Rachelson. Look where you look! saliency-guided q-networks for generalization in visual reinforcement learning. *Advances in neural information processing systems*, 35:30693–30706, 2022.

[8] 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.

[9] Di Zhang, Bowen Lv, Hai Zhang, Feifan Yang, Junqiao Zhao, Hang Yu, Chang Huang, Hongtu Zhou, Chen Ye, and Changjun Jiang. Focus on what matters: Separated models for visual-based rl generalization. *Advances in Neural Information Processing Systems*, 37:116960–116986, 2024.

[10] Wonil Song, Hyesong Choi, Kwanghoon Sohn, and Dongbo Min. A simple framework for generalization in visual rl under dynamic scene perturbations. *Advances in Neural Information Processing Systems*, 37:121790–121826, 2024.

[11] Yun Zhou, Yuqiang Wu, and Chunyu Tan. Pqda: Policy-aligned q-consistency meets decoupled augmentation for generalizable visual rl. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 29080–29088, 2026.

[12] 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.

[13] Linxi (Jim) Fan, Guanzhi Wang, De-An Huang, Zhiding Yu, Li Fei-Fei, Yuke Zhu, and Anima Anandkumar. Secant: Self-expert cloning for zero-shot generalization of visual policies. In *International Conference on Machine Learning*, 2021.

[14] Zhecheng Yuan, Sizhe Yang, Pu Hua, Can Chang, Kaizhe Hu, and Huazhe Xu. Rl-vigen: A reinforcement learning benchmark for visual generalization. *Advances in Neural Information Processing Systems*, 36:6720–6747, 2023.

[15] Richard S. Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: a framework for temporal abstraction in reinforcement learning. *Artif. Intell.*, 112(1–2):181–211, August 1999.

[16] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. Pmlr, 2018.

[17] Bolei Zhou, Àgata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40:1452–1464, 2018.

[18] 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.

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

[20] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

[21] A. Srinivas, Michael Laskin, and P. Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. *ArXiv*, abs/2004.04136, 2020.

[22] Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Ruslan Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[23] Kyungsoo Kim, Jeongsoo Ha, and Yusung Kim. Self-predictive dynamics for generalization of vision-based reinforcement learning. In Lud De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 3150–3156. International Joint Conferences on Artificial Intelligence Organization, 7 2022. Main Track.

[24] Zhaohan Daniel Guo, Bernardo Avila Pires, Bilal Piot, Jean-Bastien Grill, Florent Altché, Remi Munos, and Mohammad Gheshlaghi Azar. Bootstrap latent-predictive representations for multitask reinforcement learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3875–3886. PMLR, 13–18 Jul 2020.

[25] Jeongsoo Ha, KyungSoo Kim, and Yusung Kim. Dream to generalize: Zero-shot model-based reinforcement learning for unseen visual distractions. *ArXiv*, abs/2506.05419, 2023.

[26] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

[27] Tao Yu, Zhizheng Zhang, Cuiling Lan, Yan Lu, and Zhibo Chen. Mask-based latent reconstruction for reinforcement learning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[28] Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 627–635, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR.

[29] Abdulaziz Almuzairee, Nicklas Hansen, and Henrik I Christensen. A recipe for unbounded data augmentation in visual reinforcement learning. In *Reinforcement Learning Conference*, 2024.

A Experimental Protocol and Implementation

This appendix separates implementation details from the evidence for the two-stage design. We first describe the visual RL and Diffusion Policy protocols. Appendix B then examines the contributions of Stage-1 dynamics supervision and offline PRA, including its application to pretrained DrQ. Appendix C studies alignment capacity, Stage-2 objectives, and action consistency. Appendix D reports the geometric-shift extension. Recorded computational cost is included with the visual RL protocol in Appendix A.4.

A.1 Visual RL training protocol

The visual RL experiments separate control learning from offline representation alignment. Stage 1 trains the control reference for 500K environment steps using 84×84 RGB observations with three stacked frames. Stage 2 performs 100K offline gradient updates using observations retained in the Stage-1 replay buffer, without collecting additional environment interactions. The reference encoder and downstream control modules remain fixed during PRA; only the robust visual pathway is updated.

Table 5 summarizes the observation, optimization, and budget settings. The distinction between environment steps and gradient updates is important: the Stage-2 budget is additional computation on existing data, not an extension of the interaction budget. The visual RL experiments use an NVIDIA RTX 4090 GPU with $24\,\text{GB}$ of memory.

Table 5: Visual RL configuration. Stage 1 uses environment interaction; Stage 2 uses only offline gradient updates on the retained replay buffer.

Hyperparameter	**Value**
Observations and control	
Observation size	84×84
Frame stack	3
Discount factor γ	0.99
Batch size	256
Embedding size	DMC-GB: 100; Robosuite: 256
Action repeat	Cartpole SwingUp: 8; Walker Walk and Cheetah Run: 4; Finger Spin: 2; Robosuite: 1
Training budgets	
Stage-1 interaction budget	500K
Stage-2 update budget	100K
Replay-buffer capacity	500K
Optimization	
Actor optimizer	Adam, $\beta_1 = 0.9$, $\beta_2 = 0.999$; DMC-GB: $\text{lr} = 10^{-4}$; Robosuite: $\text{lr} = 5 \times 10^{-5}$
Critic optimizer	Adam, $\beta_1 = 0.9$, $\beta_2 = 0.999$; DMC-GB: $\text{lr} = 10^{-4}$; Robosuite: $\text{lr} = 5 \times 10^{-5}$
Dynamics loss weight λ	DMC-GB: 5; Robosuite: 1
Stage-2 alignment optimizer	Adam, $\text{lr} = 10^{-4}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$

A.2 Paired views and the alignment module

For each retained observation, a single random spatial shift produces the weak view. The reference branch processes this view, while the robust branch processes the same view after an additional appearance perturbation. The two branches therefore share their weak spatial transform: the additional appearance perturbation, rather than an independently sampled spatial shift, distinguishes the paired inputs.

The reference encoder remains fixed throughout Stage 2. Resampling a weak view changes the input to this encoder but does not update the reference mapping. In the default visual RL configuration, a

four-layer alignment module maps robust-encoder features to the latent interface of the pretrained controller. The depth comparison in Appendix C.1 examines this architectural choice.

A.3 Visual generalization settings

Figures 3 and 4 illustrate the training and evaluation appearances in RL-ViGen [14]. Original denotes the training appearance. On DMC-GB, Color Easy and Color Hard alter scene appearance, whereas Video Easy and Video Hard introduce dynamic video backgrounds. Video Hard additionally removes the floor and the agent's shadow. These settings distinguish changes to scene appearance from changes that also introduce dynamic visual content.

For Robosuite, Easy denotes a change in scene ID. The Medium, Hard, and Video labels used in this paper correspond to the original RL-ViGen Easy, Medium, and Hard settings, respectively. This is a change in naming only; the underlying benchmark configurations are unchanged. Unless otherwise stated, the visual RL tables report Original separately, and their shifted averages exclude Original and give equal weight to each reported task–shift pair.

A.4 Training time

Table 6 reports recorded approximate training times on DMC using an NVIDIA RTX 4090 GPU. STAR comprises 11.5 hours for 500K Stage-1 environment steps and 1.5 hours for a 100K-update offline Stage-2 profiling run, for a total of 13 hours. The profiled Stage-2 update count matches the visual RL budget in Table 5.

The offline stage therefore adds approximately 13.0% to the recorded Stage-1 training time ($1.5/11.5$), without increasing the environment-interaction budget. This separates the computational cost of representation alignment from the cost of collecting more control experience. The table provides configuration-specific timing context; it is not a hardware-independent runtime guarantee and does not measure the Diffusion Policy alignment schedule.

Table 6: Recorded approximate training time on DMC using an NVIDIA RTX 4090, in hours. STAR comprises 11.5 hours for 500K Stage-1 environment steps and 1.5 hours for a 100K-update offline Stage-2 profiling run.

Method	Training time (hours)
DrQ	8
SVEA	11.5
SGQN	14
SimGRL	10
SMG	18
STAR	13 (11.5 + 1.5)

A.5 Diffusion Policy protocol

We also apply PRA to pretrained Diffusion Policy [20] to examine the same separation between visual adaptation and fixed action generation in a demonstration-based policy. Offline alignment uses the retained Robosuite Proficient-Human dataset, comprising 200 episodes in the reported configuration. Inputs contain third-person and wrist views, with an independent three-layer MLP adapter of hidden dimension 512 for each view. This configuration is distinct from the four-layer alignment module used in visual RL.

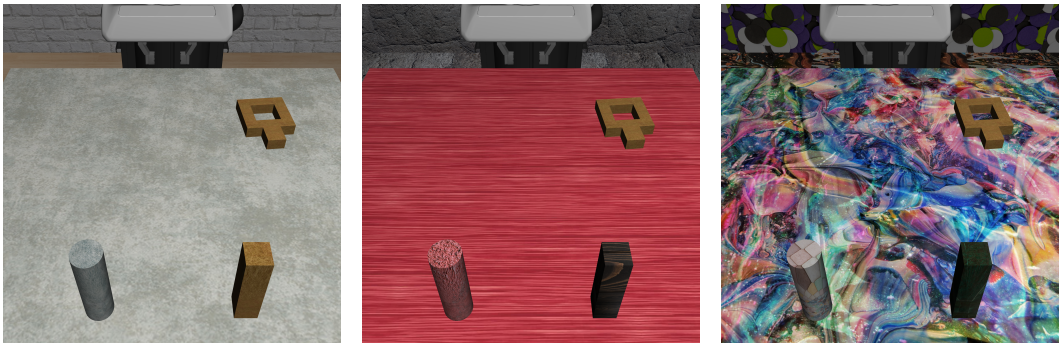
Only the robust encoders and adapters are updated. The diffusion denoising and action-generation module remains fixed throughout the 20-epoch alignment schedule in Table 7. Overlay images and mixing coefficients are resampled per observation during offline training. At inference, the aligned visual features enter the pretrained diffusion module at its visual-feature interface, preserving the normalization placement of the pretrained pathway. No new environment interaction is required for alignment.

Table 7: Diffusion Policy alignment configuration. The pretrained diffusion module remains fixed throughout the 20-epoch offline schedule.

Component	Configuration
Data and visual interface	
Dataset	Robosuite Proficient-Human
Demonstrations	200 episodes
Observation	84×84 RGB images
Frame stack	2
Views	Third-person and wrist
Adapters	Independent three-layer MLP per view
Adapter hidden dimension	512
Offline optimization	
Alignment epochs	20
Optimizer	AdamW
Learning rate	10^{-4}
Weight decay	10^{-6}
Batch size	64
Updated and frozen modules	
Updated modules	Robust encoders and adapters
Frozen modules	Diffusion denoising and action module

A.6 Diffusion Policy visual shifts

We evaluate Diffusion Policy on Can (PickPlaceCan) and Square (NutAssemblySquare). Can requires grasping and placing a can, whereas Square requires placing a nut onto a square peg. Original denotes the training appearance. Easy changes background factors, such as table and wall textures or colors; Medium additionally changes task-related visual elements; and Hard introduces stronger appearance changes. Figure 5 illustrates the Square settings. These labels describe the Diffusion Policy evaluation protocol and should be read separately from the Robosuite visual RL naming convention above.



(a) Square Easy (b) Square Medium (c) Square Hard

Figure 5: Square visual shifts for Diffusion Policy. Easy changes background appearance, Medium additionally modifies task-related visual elements, and Hard introduces stronger appearance changes.

B Controlled Results

The following experiments address two questions: how Stage-1 dynamics supervision and Stage-2 PRA contribute within a matched architecture, and whether PRA can also improve an independently pretrained RL agent. The matched ablation separates the two stages, while the DrQ experiment tests post-hoc application beyond the STAR training configuration.

B.1 Matched dynamics and alignment ablations

Tables 8 and 9 provide the task-level results underlying Table 3. The Stage-1 comparison varies forward-dynamics supervision (Dyn) while holding the STAR architecture, learning-rate configuration, optimizer, interaction budget, and seed set fixed. For each Stage-1 checkpoint, we evaluate both its original visual pathway and the pathway obtained after offline PRA (S2) using the retained replay buffer. Thus, the four columns distinguish the effect of learning a reference with dynamics supervision from the effect of subsequently aligning the visual pathway.

Table 8: Matched dynamics and PRA ablation on DMC-GB: mean episodic return $\pm$ standard deviation over four seeds. Dyn denotes Stage-1 forward-dynamics supervision; S2 denotes offline PRA. Original averages cover four tasks; shifted averages cover all 16 task–shift pairs. Walker, Cheetah, Finger, and Cartpole abbreviate Walker Walk, Cheetah Run, Finger Spin, and Cartpole SwingUp, respectively.

Setting	Task	$-$Dyn, $-$S2	$+$Dyn, $-$S2	$-$Dyn, $+$S2	$+$Dyn, $+$S2
Original	Walker	898±61	950±13	899±61	958±6
	Cheetah	665±32	624±36	651±40	615±52
	Finger	983±4	986±1	984±4	985±1
	Cartpole	864±9	872±10	867±11	872±11
Video Easy	Walker	521±86	753±58	886±67	950±8
	Cheetah	261±25	107±72	492±25	510±87
	Finger	631±25	506±13	967±18	980±7
	Cartpole	370±55	461±116	817±37	837±40
Video Hard	Walker	44±5	29±6	849±51	924±18
	Cheetah	29±10	17±11	219±43	271±67
	Finger	28±22	17±17	983±3	977±8
	Cartpole	112±37	85±19	791±17	828±21
Color Easy	Walker	608±23	597±100	857±56	939±15
	Cheetah	315±86	324±85	565±60	567±64
	Finger	953±34	917±50	968±26	975±10
	Cartpole	600±63	738±83	856±5	869±15
Color Hard	Walker	412±71	300±46	842±52	880±43
	Cheetah	192±62	215±50	509±44	493±91
	Finger	844±78	827±50	948±44	907±41
	Cartpole	478±53	496±101	825±30	824±34
Original avg.		852.5	858.0	850.3	857.5
Shifted avg.		399.9	399.3	773.4	795.7

DMC-GB: most of the shifted-return improvement follows PRA. In Table 8, adding dynamics supervision without PRA changes the original average from 852.5 to 858.0, while the shifted average remains nearly unchanged (399.9 versus 399.3). Applying PRA increases the shifted average to 773.4 without dynamics supervision and to 795.7 with it. The corresponding original averages change only slightly, from 852.5 to 850.3 and from 858.0 to 857.5. The improvement is particularly visible under Video Hard: for Walker, the dynamics-supervised reference scores 29 ± 6 before PRA and 924 ± 18 after it. Dynamics supervision alone is therefore not sufficient for robustness to the evaluated appearance shifts.

Table 9: Matched dynamics and PRA ablation on Robosuite: mean episodic return ± standard deviation. Dyn denotes Stage-1 forward-dynamics supervision; S2 denotes offline PRA. Original averages cover both tasks; shifted averages cover all eight task–shift pairs.

Setting	Task	$-\text{Dyn}, -\text{S2}$	$+\text{Dyn}, -\text{S2}$	$-\text{Dyn}, +\text{S2}$	$+\text{Dyn}, +\text{S2}$
Original	TwoArmPegInHole	349 ± 175	464 ± 18	344 ± 177	473 ± 6
	Door	290 ± 140	465 ± 19	287 ± 142	455 ± 32
Easy	TwoArmPegInHole	226 ± 75	267 ± 18	345 ± 169	461 ± 13
	Door	77 ± 35	122 ± 66	276 ± 152	440 ± 34
Medium	TwoArmPegInHole	149 ± 27	145 ± 18	325 ± 154	397 ± 22
	Door	26 ± 21	21 ± 18	253 ± 132	442 ± 38
Hard	TwoArmPegInHole	136 ± 16	136 ± 11	275 ± 138	335 ± 57
	Door	6 ± 4	4 ± 2	89 ± 111	202 ± 145
Video	TwoArmPegInHole	133 ± 22	134 ± 19	345 ± 159	424 ± 9
	Door	15 ± 14	1 ± 1	188 ± 188	343 ± 41
Original avg.		319.5	464.5	315.5	464.0
Shifted avg.		96.0	103.8	262.0	380.5

Robosuite: reference quality and visual robustness play distinct roles. Table 9 shows a larger original-domain benefit from dynamics supervision: the average increases from 319.5 to 464.5. Without PRA, however, the shifted average increases only from 96.0 to 103.8. PRA raises these shifted averages to 262.0 and 380.5, respectively. With both components, the original average is 464.0, close to the 464.5 reference value. The combined configuration has the highest aggregate shifted mean, but it does not eliminate every difficult case; for example, Door Hard remains at 202 ± 145.

Interpretation. The aggregate results distinguish the two contributions rather than treating them as interchangeable. Dynamics supervision improves the control reference most clearly on Robosuite, whereas PRA improves shifted performance for references trained both with and without that supervision. Their combination has the highest shifted average on both benchmarks, although it is not best in every individual DMC-GB setting. These comparisons support the two-stage design without implying that Stage-1 dynamics supervision is a prerequisite for PRA.

B.2 Applying PRA to pretrained DrQ

The matched ablation above varies components within STAR. Here, we ask a separate question: can offline PRA improve an existing DrQ agent that was pretrained without Stage-1 dynamics supervision? The reference encoder and downstream controller remain frozen, while only the robust encoder and alignment module are updated. This evaluates a change to the visual pathway, not additional optimization of the pretrained controller. Table 10 reports the aggregate returns.

Table 10: Post-hoc PRA on pretrained DrQ. Values are mean episodic returns averaged over the tasks and settings of each benchmark. Orig. denotes the training appearance; Shift. excludes Original. DMC and Robo denote DMC-GB and Robosuite, respectively.

Setting	DrQ	DrQ+PRA
DMC Orig.	**826.8**	821.5
DMC Shift.	459.9	**749.8**
Robo Orig.	**315.5**	312.0
Robo Shift.	98.3	**267.1**

PRA increases the average shifted return from 459.9 to 749.8 on DMC-GB and from 98.3 to 267.1 on Robosuite, corresponding to relative increases of 63.0% and 171.7%. Original-domain averages decrease slightly, from 826.8 to 821.5 and from 315.5 to 312.0, respectively. Thus, source

performance is approximately, rather than exactly, preserved. The results support applying PRA to the tested pretrained DrQ agents without retraining their controllers or collecting new environment interactions.

C Alignment Design and Diagnostics

Having separated the contributions of the two stages, we next examine how to configure the alignment stage. The depth ablation addresses alignment-module capacity, the objective ablation tests additional supervision beyond latent alignment, and the action-consistency diagnostic measures output sensitivity to appearance at fixed states. These analyses address different aspects of the method and are reported separately below.

C.1 Alignment-module depth

We compare the configurations labeled w/o AM, 2-layer, 4-layer, and 8-layer in Tables 11 and 12, where AM denotes the alignment module. This architectural comparison should be distinguished from removing Stage 2 in Appendix B.1: the label w/o AM refers to the absence of the module, not to the $-\mathrm{S2}$ column of the matched ablation.

Table 11: Alignment-module depth on DMC-GB: mean episodic return $\pm$ standard deviation over four seeds. AM denotes the alignment module; w/o AM denotes the configuration without this module. The four-layer module is the visual RL default. Bold indicates the highest mean within each task–setting pair, including ties, not statistical significance.

Setting	Task	w/o AM	2-layer	4-layer	8-layer
Original	Walker Walk	950±6	949±11	**958±6**	953±6
	Cheetah Run	600±40	610±56	**615±52**	579±87
	Finger Spin	983±1	984±1	**985±1**	**985±0**
	Cartpole SwingUp	870±10	**872±9**	**872±11**	869±8
Color Easy	Walker Walk	894±39	938±11	939±15	**945±14**
	Cheetah Run	517±88	541±78	**567±64**	496±103
	Finger Spin	968±11	970±17	**975±10**	970±11
	Cartpole SwingUp	868±10	**871±8**	869±15	869±8
Color Hard	Walker Walk	715±75	846±77	880±43	**904±33**
	Cheetah Run	455±85	**493±63**	**493±91**	457±98
	Finger Spin	881±40	897±41	907±41	**910±30**
	Cartpole SwingUp	807±38	819±26	**824±34**	**824±25**
Video Easy	Walker Walk	901±16	946±8	**950±8**	947±11
	Cheetah Run	441±66	459±82	**510±87**	457±129
	Finger Spin	946±12	978±10	**980±7**	979±6
	Cartpole SwingUp	780±28	**838±33**	837±40	835±26
Video Hard	Walker Walk	786±7	885±50	**924±18**	909±28
	Cheetah Run	259±48	**273±71**	271±67	255±70
	Finger Spin	917±23	972±8	**977±8**	973±9
	Cartpole SwingUp	732±63	813±47	**828±21**	800±53

DMC-GB: four layers give the highest aggregate shifted mean. Averaging the reported means over the 16 shifted task–setting pairs gives 741.7, 783.7, 795.7, and 783.1 for w/o AM, 2-layer, 4-layer, and 8-layer, respectively. These are unweighted averages computed from Table 11, not separate measurements. The four-layer configuration improves over w/o AM most clearly in settings such as Walker Video Hard (924 versus 786). Eight layers can help individual cases, including Walker Color Hard (904 versus 880 with four layers), but do not improve the overall shifted mean.

Table 12: Alignment-module depth on Robosuite: mean episodic return ± standard deviation over four seeds. AM denotes the alignment module; w/o AM denotes the configuration without this module. The four-layer module is the visual RL default. Bold indicates the highest mean within each task–setting pair, including ties, not statistical significance.

Setting	Task	w/o AM	2-layer	4-layer	8-layer
Original	TwoArmPegInHole	342±80	467±15	**473±6**	**473±5**
	Door	453±29	448±43	**455±32**	452±37
Easy	TwoArmPegInHole	305±69	441±17	461±13	**463±10**
	Door	299±109	428±43	**440±34**	418±58
Medium	TwoArmPegInHole	235±48	370±17	397±22	**421±13**
	Door	254±143	424±32	**442±38**	383±73
Hard	TwoArmPegInHole	169±18	291±78	335±57	**353±48**
	Door	34±40	184±130	**202±145**	181±170
Video	TwoArmPegInHole	175±51	395±35	**424±9**	422±41
	Door	136±112	309±67	**343±41**	253±91

Robosuite: additional depth has task-dependent effects. The analogous averages over the eight shifted pairs are 200.9, 355.3, 380.5, and 361.8, computed from the reported means in Table 12. Eight layers improve several TwoArmPegInHole settings, but four layers obtain higher means on all four shifted Door settings. For example, Door Video scores 343 ± 41 with four layers and 253 ± 91 with eight. Removing the module also reduces original-domain TwoArmPegInHole performance (342 ± 80 versus 473 ± 6), so its effect is not restricted to shifted inputs.

Default configuration. We use four layers for visual RL because this depth has the highest aggregate shifted mean on both benchmarks and also performs well on Original. The results do not support a monotonic improvement with depth or uniform superiority in every setting. In particular, the reported means and standard deviations do not establish that four layers are statistically superior to every alternative. The three-layer, per-view adapters used for Diffusion Policy remain a separate configuration described in Appendix A.5.

C.2 Auxiliary objectives in Stage 2

We next test whether adding task-related objectives improves upon latent alignment alone. Tables 13 and 14 compare PRA with the addition of forward-dynamics prediction (FW), Q-value consistency (Q), or replay-buffer action reconstruction (Recon.). Every variant retains latent alignment, the robust encoder, and the alignment module, while the reference encoder and downstream controller remain fixed.

Two distinctions are important for interpreting this comparison. First, FW is added during Stage 2; Stage-1 dynamics supervision is held fixed across variants. Second, PRA+Recon. reconstructs the action actually stored with each observation in the Stage-1 replay buffer, not an action newly generated by querying the frozen policy. This experiment therefore tests auxiliary losses added to PRA, rather than replacing latent supervision with action reconstruction.

Table 13: Stage-2 auxiliary objectives on DMC-GB: mean episodic return ± standard deviation over four seeds. All variants retain PRA. FW, Q, and Recon. add forward-dynamics prediction, Q-value consistency, and replay-buffer action reconstruction, respectively. Stage-1 dynamics supervision is held fixed. Shifted averages exclude Original. Bold indicates the highest mean, including ties.

Setting	Task	PRA	PRA+FW	PRA+Q	PRA+Recon.
Original	Walker	**958±6**	956±4	398±339	943±15
	Cheetah	615±52	**621±35**	237±133	497±283
	Finger	**985±1**	**985±1**	890±116	900±52
	Cartpole	872±11	870±11	607±5	**873±10**
Color Easy	Walker	939±15	**953±7**	300±257	925±24
	Cheetah	**567±64**	546±86	176±94	405±240
	Finger	975±10	**977±4**	868±137	852±44
	Cartpole	869±15	868±14	581±1	**873±10**
Color Hard	Walker	**880±43**	867±58	289±259	813±50
	Cheetah	493±91	**497±61**	138±76	363±182
	Finger	907±41	**909±31**	803±110	784±18
	Cartpole	**824±34**	822±33	507±19	817±38
Video Easy	Walker	950±8	**952±3**	344±300	921±18
	Cheetah	**510±87**	482±140	124±56	347±186
	Finger	**980±7**	979±9	867±113	895±46
	Cartpole	837±40	**840±36**	328±61	**840±29**
Video Hard	Walker	**924±18**	923±11	274±209	884±20
	Cheetah	271±67	**279±72**	117±67	173±111
	Finger	**977±8**	971±15	853±101	865±61
	Cartpole	828±21	**850±20**	234±97	820±65
Original avg.		857.5	**858.0**	533.0	803.3
Shifted avg.		**795.7**	794.7	425.2	723.6

DMC-GB: forward prediction changes little in the aggregate. PRA and PRA+FW have similar shifted averages (795.7 and 794.7) and original averages (857.5 and 858.0). Individual settings favor different variants, but the additional forward-prediction objective does not improve the overall shifted mean. In contrast, PRA+Q and PRA+Recon. reduce the shifted average to 425.2 and 723.6, and the original average to 533.0 and 803.3. The degradation in these configurations is therefore not confined to visual-shift evaluation.

Table 14: Stage-2 auxiliary objectives on Robosuite: mean episodic return ± standard deviation over four seeds. All variants retain PRA. FW, Q, and Recon. add forward-dynamics prediction, Q-value consistency, and replay-buffer action reconstruction, respectively. Stage-1 dynamics supervision is held fixed. Shifted averages exclude Original. TwoArm denotes TwoArmPegInHole. Bold indicates the highest mean.

Setting	Task	PRA	PRA+FW	PRA+Q	PRA+Recon.
Original	TwoArm	**473±6**	**473±6**	343±80	457±25
	Door	**455±32**	447±43	209±201	391±125
Easy	TwoArm	461±13	**464±12**	305±69	436±28
	Door	**440±34**	424±62	203±199	337±98
Medium	TwoArm	397±22	**415±21**	235±48	400±34
	Door	**442±38**	412±42	166±186	274±106
Hard	TwoArm	335±57	**358±61**	170±18	345±25
	Door	202±145	**214±120**	50±55	91±52
Video	TwoArm	424±9	**435±11**	176±50	378±20
	Door	**343±41**	335±30	107±124	239±79
Original avg.		**464.0**	460.0	276.0	424.0
Shifted avg.		380.5	**382.1**	176.5	312.5

Robosuite: small aggregate gains from FW conceal task-level trade-offs. PRA+FW slightly increases the shifted average from 380.5 to 382.1, while reducing the original average from 464.0 to 460.0. For example, TwoArmPegInHole Hard improves from 335 to 358, whereas Door Medium decreases from 442 to 412. PRA+Q and PRA+Recon. again obtain lower shifted averages (176.5 and 312.5) and lower original averages (276.0 and 424.0) than PRA alone.

Choice of Stage-2 objective. Across the two benchmarks, adding FW does not provide a consistent aggregate improvement, while Q-consistency and replay-buffer action reconstruction lower both original and shifted averages in the tested configurations. We therefore use latent alignment alone as the default Stage-2 objective. This conclusion is specific to the evaluated losses and configurations; the tables do not identify the optimization mechanism behind the degradation or rule out other auxiliary-loss designs.

C.3 Action consistency under visual shifts

Episodic returns measure closed-loop performance, but they do not directly isolate how appearance changes affect the action at a fixed state. Following the action-variance diagnostic [29], we therefore measure action variation across Door Medium appearances while holding the underlying state fixed. The reported variances are 0.3002 for SMG, 0.2496 for SimGRL, and 0.1718 for STAR. STAR reduces this diagnostic by 0.0778, or 31.2%, relative to SimGRL.

This result indicates lower action sensitivity to the evaluated appearance changes. It does not by itself establish action correctness or long-horizon task success, and the comparison is between complete methods rather than an isolated PRA ablation. We therefore use it as a complementary diagnostic, with episodic returns providing the closed-loop evaluation.

D Extension to a Geometric Shift Family

D.1 Shift-aware geometric augmentation

The appearance-shift experiments use random overlay during Stage 2. Here, we examine whether changing the augmentation can address a specified geometric-shift family while keeping the learned control reference fixed. We evaluate the Rotation Easy/Hard and Shift Easy/Hard configurations ported from DMC-GB2 [29] to RL-ViGen [14].

Starting from the same Stage-1 control reference, STAR(G) replaces the appearance overlay of STAR(O) with geometric augmentation matched to the evaluation transformation families and ranges.

The reference encoder and downstream controller remain fixed. Augmentation design uses knowledge of these families and ranges, but no observations or interactions from the geometric test distributions. The experiment is therefore shift-aware: it does not evaluate an augmentation selected without knowledge of the target transformation family.

Table 15: Geometric-shift results: mean episodic return ± standard deviation. O and G denote random-overlay and geometric augmentation, respectively. SVEA (O) uses Original scores from Table 1 and geometric-shift scores from our evaluation. SVEA† (G) is an external reference from SADA [29], with different training settings and five seeds; our evaluations use four seeds. Averages are rounded to integers. Bold indicates the highest reported mean; the external reference is not a matched comparison.

Setting	Task	DrQ	SVEA (O)	SVEA† (G)	STAR (G)
Original	Walker	932±19	851±78	—	**945±18**
	Cheetah	524±127	354±32	—	**594±39**
	Finger	**983±2**	968±24	—	981±3
	Cartpole	**868±12**	864±12	—	855±9
Rot. Easy	Walker	217±21	212.5±19.6	426±37	**942±16**
	Cheetah	91±10	68.1±8.4	147±31	**584±45**
	Finger	69±46	102.7±7.3	257±58	**964±33**
	Cartpole	253±13	254.7±12.7	422±23	**812±29**
Rot. Hard	Walker	132±11	129.9±12.4	228±25	**944±9**
	Cheetah	49±10	41.7±2.4	86±26	**591±46**
	Finger	36±24	51.7±3.0	128±25	**960±39**
	Cartpole	183±18	186.3±4.9	280±11	**808±57**
Shift Easy	Walker	108±10	111.3±23.8	692±67	**926±31**
	Cheetah	28±10	31.4±7.8	133±18	**579±37**
	Finger	23±34	17.3±9.5	460±85	**928±48**
	Cartpole	318±21	359.5±42.1	564±89	**725±35**
Shift Hard	Walker	52±6	77.6±9.2	154±46	**880±30**
	Cheetah	11±4	23.9±10.1	52±18	**500±34**
	Finger	1±0	1.1±0.8	90±48	**766±73**
	Cartpole	239±14	256.1±21.0	318±33	**554±40**
Original avg.		827	759	—	**844**
Shifted avg.		113	120	277	**779**

Results and remaining limitations. Table 15 reports a shifted average of 779 for STAR(G), compared with 113 for DrQ and 120 for SVEA(O). STAR(G) has the highest reported mean in each geometric task–shift pair, while its original average is 844. Robustness is nevertheless incomplete under the stronger translations. For example, Shift Hard reduces Cartpole from 855 on Original to 554, and Finger from 981 to 766. These residual drops are important when interpreting the aggregate improvement.

Scope of the comparison. SVEA†(G), with a shifted average of 277, is an external reference from the geometric-augmentation experiment in SADA. It uses different training settings and five seeds, whereas our results use four seeds, so it should not be read as a matched comparison. The experiment supports using offline alignment with a specified geometric augmentation family while retaining the controller; it does not establish robustness to arbitrary unseen geometric transformations.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly include the claims made in the paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations and future work are discussed in the Section 7.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [No]

Justification:

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper reports the experimental setup, evaluation protocol, and implementation details needed to reproduce the main results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The paper includes experiments requiring code.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental setting is presented in the paper in Appendix A.1

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper reports the mean and standard deviation of the results in the tables

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper includes experiments compute resources.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research was conducted in accordance with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not involve data or models with high misuse risk

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Creators or original owners of assets used in the paper, are properly credited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs were not used as part of the core methodology, experiments, or analysis in this research.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.