Article | Received 5 September 2025; Revised 23 February 2026; Accepted 23 February 2026; Published 30 April 2026
https://doi.org/10.55092/rl20260012

Improving sample efficiency in visual reinforcement learning under data augmentation



Ziru Wang, Wanli Jiang, Qian Kou, Ru Peng and Xuguang Lan*

National Key Laboratory of Human-Machine Hybrid Augmented Intelligence, Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, China

* Correspondence author; E-mail: xglan@mail.xjtu.edu.cn.

Highlights:

- We investigate the inefficiencies of data augmentation in reinforcement learning (RL), focusing on increased complexity in representation learning and increased uncertainty in value estimation.
- We introduce the Feature Consistency and Self-value Distillation (FCSD) framework to address these challenges. We propose Feature Consistency (FC) to aid RL in extracting effective features, and introduce Self-value Distillation (SD) to stabilize the value function learning, thereby enhancing sample efficiency.
- Our experimental results confirm that FCSD enhances sample efficiency and overall performance in both off-policy and on-policy visual RL methods under data augmentation without the need for auxiliary tasks or additional learnable parameters.

Abstract: Visual reinforcement learning (RL) relies on high-dimensional visual inputs but often suffers from low sample efficiency. While data augmentation enhances input diversity, it can inadvertently hinder policy learning by increasing visual complexity and reducing consistency. In this paper, we systematically analyze the efficiency bottlenecks associated with data augmentation and identify two key challenges: increased representational complexity and uncertainty in value estimation. To tackle these issues, we propose a novel framework, Feature Consistency and Self-value Distillation (FCSD). FCSD enforces feature consistency across augmented views and stabilizes value learning by distilling from its own past value network. We evaluate FCSD on several challenging RL benchmarks, including DMControl, ProcGen and Atari. Experimental results suggest that FCSD improves sample efficiency by mitigating augmentation-induced instability. Compared to several representative augmentation-based methods, FCSD demonstrates competitive or superior performance. Moreover, FCSD integrates seamlessly with both on-policy and off-policy RL algorithms, and demonstrates consistently robust and generalizable performance across diverse tasks and domains.

Keywords: visual reinforcement learning; sample efficiency; data augmentation



Copyright©2026 by the authors. Published by ELSP. This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium provided the original work is properly cited.

1. Introduction

Visual reinforcement learning (RL) has shown remarkable potential in a wide range of decision-making problems by learning directly from high-dimensional visual inputs [1–5]. Compared to traditional RL, which typically operates on low-dimensional or structured state inputs [6,7], learning from raw pixels introduces additional challenges in representation learning and policy optimization. These challenges often result in lower sample efficiency and make practical deployment more difficult [8]. In response, many visual RL approaches adopt data augmentation strategies to diversify training samples without increasing the number of environment interactions [9–12].

Data augmentation is widely integrated into visual RL methods at multiple levels, including pixel-level transformations of input observations [13], feature-level auxiliary learning tasks [14–16], and output-level regularization of value or policy predictions [17]. Although these techniques can yield significant performance improvements, their effectiveness heavily depends on the complexity of the augmentations used and often requires extensive training [18,19]. Meanwhile, modern visual deep learning increasingly relies on large-scale models trained on massive datasets, leading to practical constraints such as limited computational budgets and fewer training epochs [20]. Under these resource-constrained settings, conventional augmentation approaches may fail to deliver their full potential, thus highlighting the necessity to reconsider how data augmentation should be effectively integrated into visual RL training frameworks.

To better understand these limitations, we conduct a systematic analysis and identify two major challenges limiting the effectiveness of data augmentation in visual RL: (1) Increased Complexity in Representation Learning: Data augmentation introduces visual variability, complicating the learning of stable and consistent feature representations. Although augmentation enriches training diversity, these transformations can inadvertently introduce irrelevant visual features or noise, making it harder for models to extract robust semantic information [19,21,22]. Our experimental analysis empirically confirms that augmentation-induced variability significantly increases the complexity of the learned representations, hindering efficient feature learning. (2) Increased Uncertainty in Value Estimation: Data augmentation creates visually distinct variations of identical states, leading to inconsistent Q-value estimates across these augmented views. The recursive structure of the Bellman equation exacerbates such inconsistencies, introducing instability into the training process and consequently reducing convergence speed and sample efficiency [23]. Our experimental results illustrate that augmentation-induced variance leads to increased uncertainty in value estimation, necessitating additional environment interactions and computational resources.

Recent approaches attempt to address these two challenges separately, either by enhancing representation quality [24–26] or improving value estimation stability [27–29]. For instance, methods that incorporate auxiliary self-supervised tasks or pretrained visual encoders aim to improve the robustness of learned representations. However, these components are typically decoupled from the policy optimization objective and may introduce conflicts in the optimization process [30,31]. On the other hand, periodic distillation methods stabilize the value targets across temporal or ensemble models [27–29], but typically involve additional networks or alternating training phases, leading to greater

implementation complexity and computational overhead. Despite these efforts, a comprehensive solution that simultaneously addresses both representational complexity and value estimation instability, while remaining lightweight and easy to implement, remains largely unexplored, motivating the development of our proposed approach.

To this end, we propose a lightweight yet effective framework (FCSD)—Feature Consistency and Self-value Distillation—designed to simultaneously mitigate the representational complexity and value estimation instability introduced by data augmentation. FCSD enforces feature invariance across augmented views and stabilizes value learning by aligning current value predictions with temporally smoothed targets derived from past network. Notably, FCSD is model-agnostic and can be seamlessly integrated into a wide range of RL algorithms without requiring any architectural modifications or additional parameters.

(1) Feature Consistency (FC): to improve the efficiency of representation learning, we design FC to encourage invariance across different augmented views. Specifically, two augmented batches are generated from the same raw observation, and FC minimizes the discrepancy between their feature representations. As shown in Figure 1, the features trained with FC exhibit fewer non-zero singular values compared to the baseline, indicating that FC suppresses high-variance directions and promotes the extraction of stable, task-relevant features, thereby improving learning efficiency.

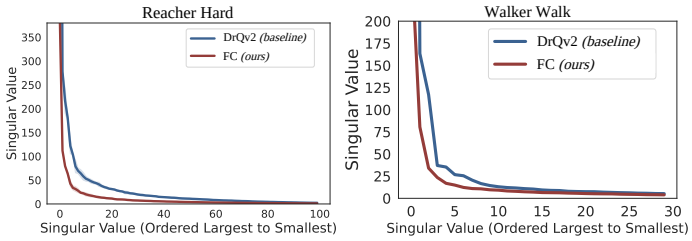


Figure 1. FC flattens the representations of the baseline, DrQv2 [4]. DrQv2 is a popular and effective RL algorithm that directly applies data augmentation. This shows that FC helps capture essential features, thereby improving sample efficiency.

(2) Self-value Distillation (SD): to address value estimation instability, we introduce SD as an auxiliary regularization term added to the standard temporal-difference (TD) loss. SD aligns the current value predictions with a smoothed target obtained via exponential moving average of past value networks. As shown in Figure 2, SD significantly reduces the variance of value predictions, mitigates numerical divergence during training, and maintains consistency in value estimates under different augmentations, thus enhancing both training stability and sample efficiency.

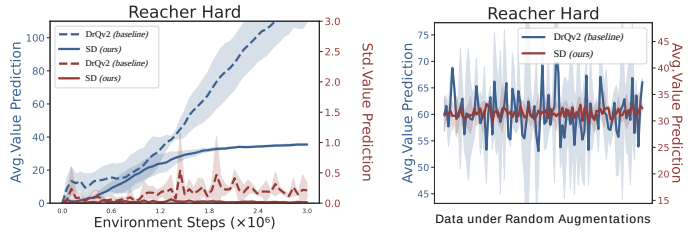


Figure 2. *Left*: Compared to the baseline, DrQv2, SD reduces the variance of value estimation and moderates the upward trend. *Right*: SD mitigates the high variability of DrQv2 under different data augmentations. These demonstrate that SD enhances the stability of the value function learning process, thereby improving sample efficiency.

We systematically evaluate the effectiveness of FCSD from the following perspectives. In terms of sample efficiency, FCSD significantly accelerates policy learning on the DMControl [32] and achieves higher cumulative rewards under the same number of environment interactions. Regarding algorithmic compatibility, we show that FCSD can be seamlessly integrated into both on-policy and off-policy reinforcement learning algorithms, achieving notable or competitive gains across several visual RL benchmarks, including ProcGen [33] and Atari 100k [34].

2. Related work

2.1. Data augmentation in visual reinforcement learning

In visual reinforcement learning, data augmentation strategies have been proposed to regularize different components of the learning process—ranging from raw input observations (pixel-level) [4,13,35], intermediate feature representations (feature-level) [25,36,37], to policy or value outputs (output-level) [17,38]. These methods are jointly applied during training to improve robustness and generalization. Pixel-level augmentation directly applies transformations to raw visual inputs to enrich data diversity. Methods such as RAD [35] and DrQ [39] demonstrate the effectiveness of geometric augmentations like random crop, translation, and flip. DrQv2 [4] further integrates such pixel-space augmentations into an off-policy actor-critic framework. UCB-DrAC [14] automates the selection of augmentation operations, and CycAug [13] alternates among multiple augmentations to balance diversity and stability. Feature-level augmentation introduces auxiliary objectives to improve representation learning. CURL [25] adopts contrastive loss with a momentum encoder to encourage feature discrimination across augmented views. RAIR [37] aligns features using cosine similarity. SIM [36] applies cross-correlation regularization in projection space, translating the self-supervised loss into intrinsic rewards. However, these approaches often operate independently of value learning, and may not directly address instability in value estimation. Output-level augmentation focuses on stabilizing value and policy outputs directly. For example, MixReg [38], inspired by MixUp [40], interpolates observations and their associated value targets, effectively regularizing the output predictions and enforcing smoothness in the learned policy or value function. Hu *et al.* [17] propose a variance-suppression regularization strategy on policy and Q-value outputs to improve consistency under augmentation. While these methods improve performance from different perspectives, most address representation and value estimation challenges in isolation, neglecting their coupled effect under data augmentation. In contrast, our proposed FCSD jointly promotes feature consistency across augmented views and stabilizes value estimation through self-distillation, offering a lightweight augmentation strategy that integrates seamlessly into existing RL algorithms.

2.2. Sample-efficient visual reinforcement learning

Visual RL aims to learn compact state representations from high-dimensional observations while optimizing task-specific policies, a process that is inherently sample-intensive [8]. Enhancing sample efficiency is crucial for accelerating learning and improving performance. Strategies like utilizing self-supervised learning tasks are employed to bolster representation learning, incorporating auxiliary tasks such as pixel or latent reconstruction [8,24], future prediction [41,42], and contrastive learning

for instance discrimination [15,16] or temporal discrimination [43,44]. These are proven to facilitate learning by promoting relevant feature extraction. Additionally, pretraining visual encoders on related tasks proves effective in boosting the initial performance on downstream tasks [26,45,46]. Model-based techniques, which construct predictive models of the environment, contribute to more efficient policy planning [47–49]. Moreover, recent studies delve into the combination of data augmentation and various auxiliary tasks to boost sample efficiency in visual RL [24,25]. While these approaches can enhance learning, they introduce additional computational costs and complexities. Selecting suitable auxiliary tasks is crucial, as suboptimal choices can lead to gradient interference, complicating the learning process [30,31]. Our research departs from existing methods by thoroughly examining how data augmentation can introduce inefficiencies in visual RL. We focus particularly on the impact of these inefficiencies on representation learning and value estimation, enhancing sample efficiency without the need for extensive task-specific adjustments.

3. Preliminaries

3.1. Reinforcement learning

We adopt the standard formulation of RL under the Markov decision process (MDP) formalism [50]. Here, an agent observes a state s from a space S, selects an action a from a space A, and receives a reward r with probability $p_R(r|s,a)$. The environment then transitions into a new state s' with probability $p_{S'}(s'|s,a)$, and this interaction continues. The goal of the agent is to learn a policy π that maximizes the expected discounted sum of rewards $\mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right]$, where t is the time step in the interaction process and $\gamma \in [0,1)$ is a discount factor.

Many RL algorithms use temporal difference (TD) updates. For instance, methods such as Soft Actor-Critic (SAC) [51], Deep Deterministic Policy Gradient (DDPG) [52], and DrQv2 [4] train a neural network to approximate the Q-value. The Q-value of a state-action pair $Q^\pi(s,a)$ is approximated by minimizing the error between the target $T(s,a)$ and its prediction $Q_\theta(s,a)$. The Q target $T(s,a)$ is computed using Bellman's equation:

$$T(s,a) = r + \gamma \bar{Q}(s',a'), \tag{1}$$

where s' and r are sampled from the environment given (s,a), and a' is sampled from the current policy given s'. In some deep RL algorithms, $\bar{Q}$ is a delayed copy of the Q-network, referred to as the target network. The parameters θ of the Q-network are optimized using stochastic gradient descent to minimize the loss function:

$$\mathscr{L}_{RL}(\theta) = \left\| Q_\theta(s,a) - T(s,a) \right\|^2. \tag{2}$$

Training the value network with this loss function involves regression on non-stationary targets, which can complicate the convergence behavior. Previous studies have suggested that this uncertainty is a source of instability [27,52].

Due to space constraints, our analysis here primarily focus on Q-value methods, though the principles discussed can apply to other algorithms as well.

3.2. Pitfalls of data augmentation in visual reinforcement learning

In this section, we elucidate the two primary reasons for limited training performance when data augmentations are naively applied in visual RL: increased complexity of representation learning and increased uncertainty in value estimation. We conduct various experiments on the DMControl suite and analyze the training sample efficiency from two main perspectives: the similarity of features between different paired augmentations and the differences in values across different augmentations. We use a randomly initialized DrQv2 agent as the base model and integrate different data augmentation techniques, including cropping, translation and rotation.

3.2.1. Data augmentation increases the difficulty of representation learning

In visual RL, agents need to extract effective features from visual inputs to make decisions. Data augmentation techniques, such as rotation, scaling, cropping, and color changes, are widely used to increase the diversity of training data. These augmentations help models generalize better by exposing them to a broader range of scenarios. However, the increased variability in input data requires the model to identify stable feature representations [19]. Significant augmentations can make it challenging for the model to extract consistent and reliable features, as they introduce additional complexity to the learning process. This added complexity has a dual effect: it enhances the model's robustness by forcing it to learn more generalizable features, but it can also introduce noise or irrelevant features [21,22], complicating the learning process. This noise can interfere with the model's ability to recognize the inherent characteristics of objects, thereby affecting training efficiency and overall model performance.

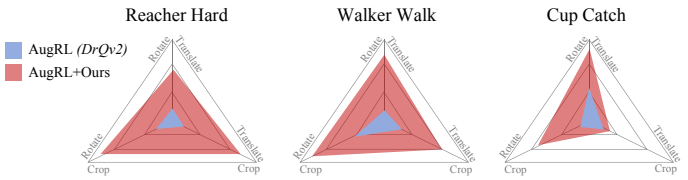
As shown in Figure 3a, our experiments demonstrate that the feature similarity between different augmented pairs is low, adding further complexity to the representation learning process. Low similarity indicates that the model struggles to maintain consistent feature representations across different augmented versions of the same input. However, after applying our proposed method, feature similarity improves significantly. This improvement suggests that our strategy effectively mitigates the difficulties introduced by data augmentation, allowing the model to learn more stable and consistent feature representations. Consequently, this leads to enhanced training sample efficiency.

3.2.2. Data augmentation increases the uncertainty of value estimation

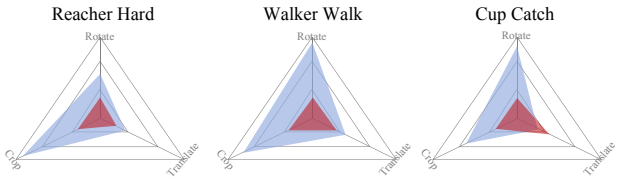
Applying data augmentation introduces increased uncertainty into value estimation, as a single underlying state may appear in diverse forms due to various visual or perceptual transformations. This visual inconsistency causes the model to assign different values to essentially the same state. Given the recursive nature of the Bellman equation (see Equation (1)), such inconsistencies propagate through successive value updates, amplifying estimation errors across the entire value function. For example, in Q-learning, noise introduced by augmentation affects the target of one state-action pair, which in turn impacts the updates of others—leading to a compounding error effect throughout training. This phenomenon presents two major challenges in reinforcement learning: First, the training process becomes unstable, as the targets used for updates are inconsistent across augmentations. This destabilization slows convergence, as the model must constantly adapt to fluctuating and sometimes contradictory value signals. As a

result, the policy learning process becomes less efficient, requiring more steps to reach optimal behavior. Second, the increased variability necessitates more interactions with the environment to obtain stable value estimates, thereby reducing sample efficiency. Since the augmented views introduce representational noise, the model must observe more data to average out inconsistencies and recover reliable learning signals—making training more resource-intensive and time-consuming.

Although data augmentation adds diversity to training samples, providing a broader range of scenarios for the model to learn from, it necessitates more samples for stable value estimations and effective policy performance. The increased diversity can improve the model's generalization abilities but complicates the learning process by introducing more variability that the model must account for. As shown in Figure 3b, agents exhibit significant variance in value estimates across different data augmentation settings, leading to training instability and thus reducing sample efficiency. The results illustrates that value estimates for the same state vary widely when subjected to different data augmentations. This wide variance indicates that the model's perception of the same underlying state changes significantly with each augmentation, making it difficult for the model to converge on a stable policy, thereby reducing training sample efficiency. Implementing our method to stabilize value estimations improves consistency across different augmentation settings, thereby enhancing training efficiency.



(a) Cosine similarity between paired feature for various augmented data



(b) Value estimation associated with various augmented data

Figure 3. Qualitative analysis of feature similarity across different pairs of augmentations, along with the corresponding differences in normalized value estimations.

4. Method

To enhance sample efficiency in visual RL under data augmentation, we propose FCSD, a framework designed to leverage data augmentation more effectively by improving feature extraction and stabilizing value estimation. Our approach comprises two primary components. First, we generate two augmented batches from the same data and apply FC modulation to maintain consistent features across different augmented views. This addresses the challenges of representation learning that arise from significant variances between augmentations, thereby improving training efficiency. Second, we implement a SD as a regularization strategy to decrease value estimation uncertainty, which stabilizes optimization process and enhances sample efficiency. Our framework is compatible with various standard RL algorithms without

requiring modifications to their network structures, auxiliary tasks, or additional learnable parameters.

4.1. Overall framework

We present our FCSD, involving minimal adaptation of a base RL algorithm by integrating feature consistency and self-value distillation components. Figure 4 illustrates the architectural design. We decompose the visual RL pipeline into a feature extraction function ϕ and a value function Q. Specifically, in our neural network, all layers before the final layer constitute the feature extraction function, while the value function is represented by the final linear combination layer. For an image, we first generate two augmented views using two random data augmentation modules. These augmented views are subsequently processed by a shared feature extractor and value network. To ensure a fair comparison, we adopt the same augmentation configurations as those used in the baseline methods throughout our experiments.

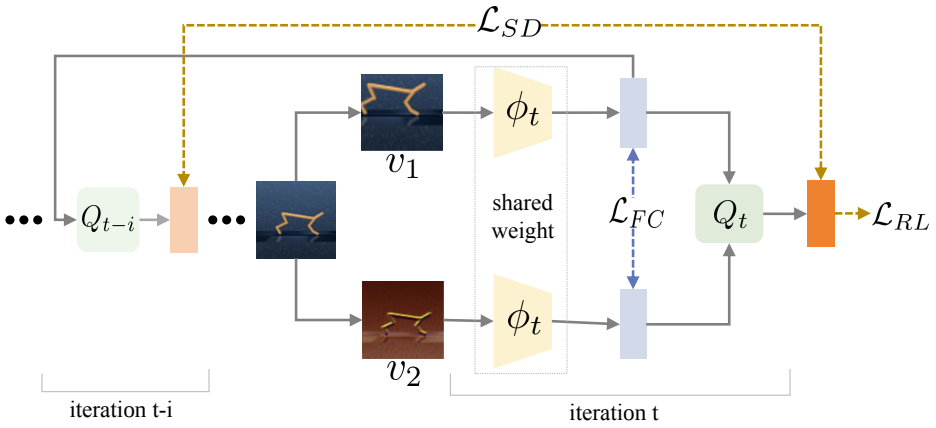


Figure 4. The overall framework of FCSD is designed to leverage data augmentation more effectively by enhancing feature extraction and stabilizing value estimation. FC promotes invariance across augmented views by minimizing the discrepancy between their feature embeddings. SD regularizes value estimation by aligning current value predictions with targets obtained from the historical value network.

The training process proceeds as follows: in each training iteration, we sample a batch of images of size N and generate two augmented views for each, resulting in $2N$ images. We calculate the TD error loss $\mathscr{L}_{\text{RL}}$ and the self-value distillation loss $\mathscr{L}_{\text{SD}}$ for each image, and the feature consistency loss $\mathscr{L}_{\text{FC}}$ for each pair of images with different augmented views. By integrating these loss functions, we formulate the following optimization objective:

$$\mathscr{L} = \mathscr{L}_{\text{RL}} + \alpha\mathscr{L}_{\text{FC}} + \beta\mathscr{L}_{\text{SD}}, \tag{3}$$

where α and β are hyperparameters that control the contributions of the feature consistency and self-value distillation terms to the overall loss function. To reduce the cost of hyperparameter tuning in our experimental setup, we set α and β to identical values. Additional details are provided in the appendix. In the following section, we provide detailed explanations of the FC and SD components.

4.2. Feature consistency

We propose enforcing the consistency of feature similarity between different augmented views of the same image within a RL framework. FC aims to ensure that the model learns invariant features despite variations introduced by data augmentation, thus improving sample efficiency. We use the L2 loss to measure and enforce feature consistency between different augmented views. The assumption is that similar images, even when augmented, should display similar feature representations. For an input state s, let $v_1 = \text{aug}_1(s)$ and $v_2 = \text{aug}_2(s)$ represent two different augmented views of s. Here, $\text{aug}_\lambda(\cdot)$ is the λ-th augmentation function. The feature consistency loss is defined as:

$$\mathscr{L}_{FC} = \|\phi(v_1) - \phi(v_2)\|^2. \tag{4}$$

where $\phi(\cdot)$ denotes the shared feature encoder that maps augmented observations to latent representations.

FC encourages the model to focus on the essential attributes of input data while suppressing irrelevant variations introduced by data augmentation. This leads to more stable and efficient representation learning, particularly advantageous in environments where data collection is costly. In contrast to contrastive learning methods that require both attracting positive pairs and repelling negative pairs, FC solely minimizes intra-sample discrepancies between augmented views. It eliminates the need for negative samples or projection heads, making it a lightweight and easily integrable component for reinforcement learning frameworks.

4.3. Self-value distillation

To mitigate the uncertainty in value estimation caused by data augmentation, we propose a lightweight auxiliary distillation strategy—SD. This approach leverages an exponential moving average (EMA) of the current model as a temporal teacher to provide more stable value supervision during training, thereby improving sample efficiency. The key intuition is that although past value predictions may not be perfect, they serve as temporally smoothed anchors that regularize the current learning signal [53]. This helps suppress gradient noise and alleviates the inconsistencies introduced by diverse augmented views.

Specifically, we maintain a teacher network θ' by applying EMA updates to the current student parameters θ, as follows:

$$\theta' \leftarrow (1-\tau)\theta' + \tau\theta, \tag{5}$$

where $\tau \in [0,1]$ is the smoothing coefficient. The teacher is not updated by backpropagation, which keeps it stable and computationally efficient. We distill only the value function, excluding the encoder, to simplify training. The SD loss is then defined as:

$$\mathscr{L}_{SD}(\theta) = \|Q_\theta(s,a) - Q_{\theta'}(s,a)\|^2. \tag{6}$$

From a gradient dynamics perspective, SD improves training stability by implicitly incorporating information from past model behavior. Since the EMA-based teacher smooths temporal fluctuations in value estimates, the resulting gradients are more consistent over time. This suppresses noisy updates, reduces variance, and leads to a more stable optimization trajectory. As shown in Figure 5 and Figure 6, SD significantly reduces gradient oscillations, which can facilitate faster convergence by enabling more stable and efficient learning.

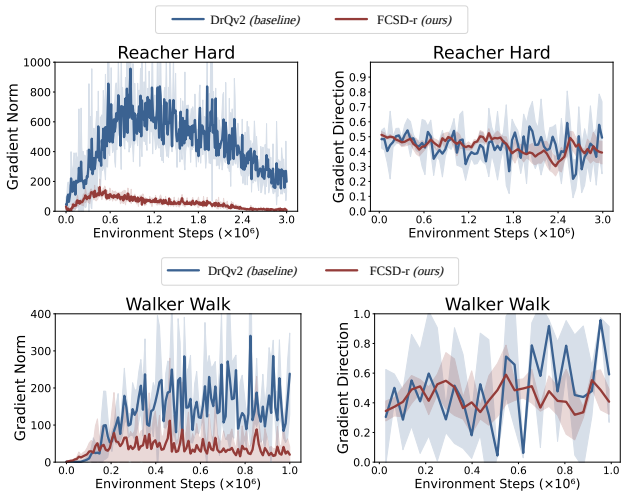


Figure 5. FCSD-r contributes to gradient stability. Left: It leads to smaller gradient norms and reduced variations. Right: It reduces the degree of changes in the gradient direction.

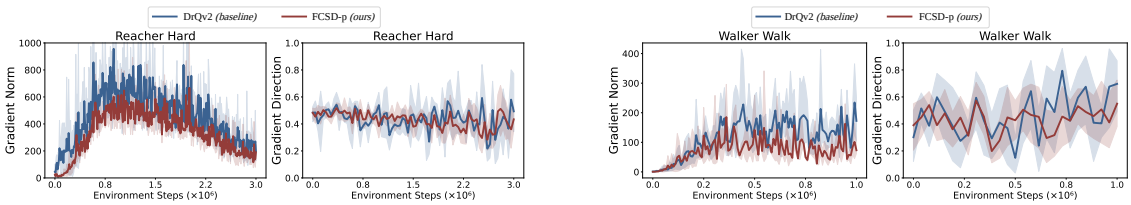


Figure 6. FCSD-p contributes to gradient stability. Left: It leads to smaller gradient norms and reduced variations. Right: It reduces the degree of changes in the gradient direction.

Theoretical support. We demonstrate that SD enhances training sample efficiency by prioritizing samples needing further learning, similar to classic resampling techniques in off-policy reinforcement learning but simpler and more broadly applicable. To isolate the effect of the self-value distillation term $\mathscr{L}_{\text{SD}}$, we define a simplified loss by removing the feature consistency component $\mathscr{L}_{\text{FC}}$ from Equation (3). We refer to the simplified loss composed of $\mathscr{L}_{\text{RL}} + \beta\mathscr{L}_{\text{SD}}$ as SDRL, to emphasize the interaction between reinforcement learning and self-distillation in the absence of the feature consistency term. For clarity, we denote the gradient of this surrogate loss with respect to the Q-value at iteration t as ∂^t_{SDRL}, defined as:

$$\partial^t_{SDRL} \triangleq \frac{\partial \mathscr{L}^t_{\text{SDRL}}}{\partial Q} = (Q_t - \bar{Q}) + \beta(Q_t - Q_{t-i}). \tag{7}$$

Here, Q_t is the current Q-value, $\bar{Q}$ is the target Q-value, t represents the current iteration, and $t-i$ represents a previous i step before the current one.

To understand the gradient rescaling factor induced by SD, consider the ratio of L1 norms: $\sum_k \left|\partial^{t,k}_{SDRL}\right| / \sum_k \left|\partial^{t,k}\right|$, where k indexes over all samples, and $\partial^{t,k} = Q_t - \bar{Q}$ represents the standard gradient for the k-th sample at step t without SD. Substituting the expression for $\partial^{t,k}_{SDRL}$, we get:

$\frac{\sum_k \left|(Q_t-\bar{Q})+\beta(Q_t-Q_{t-i})\right|}{\sum_k \left|Q_t-\bar{Q}\right|}$. Simplifying this, we have:

$$\frac{\sum_k \left|\partial^{t,k}_{SDRL}\right|}{\sum_k \left|\partial^{t,k}\right|} = 1+\beta-\beta\frac{Q_{t-i}-\bar{Q}}{Q_t-\bar{Q}} = 1+\beta-\beta\frac{\Delta_{t-i}}{\Delta_t}. \tag{8}$$

Here, Δ_t represents the distance between the model's value estimate and the target value at step t, and Δ_{t-i} represents the distance at step $t-i$.

A large $\frac{\Delta_{t-i}}{\Delta_t}$ indicates that the previous estimate is further from the target compared to the current estimate, suggesting that the sample is improving. Conversely, a small $\frac{\Delta_{t-i}}{\Delta_t}$ suggests that the sample's current estimate is still far from the target value and needs further learning. This means that SD assigns more weight to samples needing further learning during training, thereby improving sample efficiency while also ensuring stability as the model converges. It is worth noting that the above explanation is effective when the model has not yet converged. As the model converges, $\frac{\Delta_{t-i}}{\Delta_t}$ and $\frac{\Delta_t}{\Delta_t}$ approach each other. Therefore, the distinction between samples becomes less significant, which also demonstrates that our approach does not negatively affect convergence.

Two configurations of SD. In our experiments, we evaluate two configurations of the SD mechanism based on different teacher update strategies: (1) SD-p employs an EMA teacher with a decay coefficient $\tau > 0$, providing temporally smoothed supervision that evolves alongside the student model. (2) SD-r sets $\tau = 0$, resulting in a static teacher network fixed at initialization, which offers a more stable but unchanging supervision signal. While such a fixed teacher may seem simplistic, recent studies in supervised learning suggest that even randomly initialized, non-updated teacher networks can serve as useful anchors during training—providing implicit regularization and guiding the student toward stable convergence [54].

Importantly, SD-p and SD-r are not fundamentally different methods but two configurations of the same distillation mechanism, controlled by the choice of the EMA update rate τ. We distinguish them in our experiments to illustrate a critical trade-off between adaptivity and stability. SD-r incurs negligible computational overhead and is well-suited for environments with stable dynamics or limited computational budgets. In contrast, SD-p provides adaptive supervision better aligned with evolving policy targets, which proves more effective in visually complex or highly dynamic environments. Despite their differences, both variants share a unified objective: to provide temporally consistent and robust value targets that mitigate the noise and instability introduced by data augmentation, thereby improving training efficiency and convergence stability.

5. Experiments

We conduct comprehensive experiments to evaluate the effectiveness of the proposed FCSD. Our evaluation focuses on the following four key aspects:

(1) Sample efficiency in visual RL (Section 5.2). To assess whether FCSD improves sample efficiency, we integrate it into both DrQv2 [4] and SAC [51], and evaluate performance on the DMControl benchmark [32].

(2) Compatibility across different RL algorithms (Section 5.3). We validate the cross-algorithm compatibility of FCSD by integrating it into both off-policy (e.g., SPR [41]) and on-policy (e.g.,

PPO [55]) algorithms. Experiments are conducted on Atari for discrete control and on ProcGen for generalization evaluation.

(3) Analysis and ablation study (Section 5.4). We perform ablations to analyze the contributions of key components in FCSD and visualize training dynamics by monitoring various gradient statistics over time.

5.1. Environments and training details

5.1.1. DMControl

We conduct experimental evaluations on a set of MuJoCo tasks provided by the DMControl suite, a widely adopted benchmark for continuous control in reinforcement learning. Our study focuses on the pixel-based learning setting, where observations are constructed by stacking three consecutive RGB frames with a resolution of 84×84 along the channel dimension. This design allows the agent to infer motion-related information for effective action decision-making. In our experimental design, we first integrate FCSD into the DrQv2 algorithm, which has been shown to be a strong and computationally efficient baseline for pixel-based control. This integration allows us to examine whether FCSD can further enhance sample efficiency and training stability when combined with a high-performing algorithm. In addition, we apply FCSD to the standard Soft Actor-Critic (SAC) algorithm to assess whether our method can also yield performance improvements in more basic, augmentation-free reinforcement learning settings. All training hyperparameters and experimental settings follow standard practices established in prior reinforcement learning work, ensuring fair comparisons. Detailed configurations and additional implementation details are provided in the appendix.

5.1.2. Atari

We evaluate the FCSD method on the pixel-based Atari benchmark, which includes 26 discrete control tasks. In our experiments, FCSD is integrated into the SPR [41] algorithm. All experimental settings strictly follow the original SPR implementation. We compare FCSD against a set of representative baselines, each incorporating data augmentation at different levels. CURL [25] and DrQ [39] apply explicit image-level augmentations to enhance the robustness of visual representations. PSRL [56] introduces temporal perturbations through frame-to-frame spatial consistency learning, which can be regarded as a form of implicit temporal augmentation. DynaSTI [57] combines explicit image perturbations in the input space with structured representation learning via dynamics modeling, representing a hybrid augmentation strategy.

5.1.3. ProcGen

ProcGen is a standard benchmark suite consisting of 16 procedurally generated game environments. Each environment features a 15-dimensional discrete action space and provides $64 \times 64 \times 3$ RGB observations. Due to its high task diversity, stochastic nature, and inherent generalization challenges, ProcGen is widely regarded as an ideal platform for evaluating generalization in RL. In this setting, we integrate FCSD into the on-policy PPO algorithm and systematically evaluate its performance and sample efficiency across all

16 ProcGen games. During training, agents are exposed to 200 procedurally generated levels per game and are tested on the full distribution of levels. We compare FCSD against several representative reinforcement learning methods that incorporate data augmentation. These include RAD [35], MixReg [38], and UCB-DrAC [14], which introduce perturbations at the image or feature level. Additionally, we include ExpGen [58], which can be considered a broader form of data augmentation—it promotes behavioral diversity via policy perturbation, thereby enriching the distribution of data in behavior space.

Table 1. Aggregated scores of different sample-efficiency methods on the DMControl (aggregated over 20 tasks) at 100K, 500K and 1000K environment steps. Our methods are integrated into the DrQv2. All results are computed over 10 random seeds. Detailed per-task scores and corresponding training curves are provided in Figure 7.

		100K			**500K**			**1000K**		
	Method	**IQM**	**Mean**	**Median**	**IQM**	**Mean**	**Median**	**IQM**	**Mean**	**Median**
Model-based	Dreamer	87.85	147.14	82.64	498.90	548.14	474.04	709.12	694.48	725.94
	DreamerV2	223.61	300.89	225.77	615.82	625.56	609.60	778.74	745.75	803.51
Model-free	SAC	96.48	188.30	96.37	79.73	211.75	59.10	90.32	225.58	65.09
	CURL	257.25	349.44	228.71	431.55	465.72	394.06	500.63	504.41	468.71
	DrQ	253.68	348.56	213.37	475.79	495.82	451.24	540.12	534.18	539.47
	DrQv2 *baseline*	192.10	267.25	165.42	590.91	559.23	662.63	741.12	675.69	751.09
	FCSD-p ours	247.76	289.66	149.93	722.69	592.57	664.93	883.43	708.38	827.33
	FCSD-r ours	296.03	314.58	234.59	803.05	642.13	811.69	921.06	738.60	846.07

The reported results in Table 1 of these baselines are from https://github.com/facebookresearch/drqv2/tree/main/curves, https://github.com/danijar/dreamer/tree/master/scores, https://github.com/danijar/dreamerv2/tree/main/scores, https://github.com/danijar/dreamerv3/tree/main/scores/data.

5.2. Sample efficiency in visual RL

Comparative evaluation. To evaluate the effectiveness of our method in improving sample efficiency, we first apply FCSD to the DrQv2 algorithm and assess its performance on 20 tasks from the pixel-based DMControl benchmark. Table 1 presents the aggregated performance at 100K, 500K, and 1000K environment steps. Detailed results for each task are provided in Figure 7.

The results show that integrating FCSD into DrQv2 significantly improves both overall performance and sample efficiency compared to DrQv2 and other methods. While FCSD-p performs slightly below FCSD-r in this setting, it consistently outperforms DrQv2 across all tasks. FCSD-r can be viewed as a simplified version of FCSD-p: when the EMA decay factor $\tau = 0$, FCSD-p reduces to FCSD-r, meaning the teacher network remains fixed throughout training. The DMControl suite features simple visuals, and dense rewards, leading to relatively stable training. Under such conditions, using a static teacher (as in FCSD-r) helps avoid target fluctuations introduced by EMA updates, thereby improving stability. However, as we show in subsequent experiments (e.g., Atari), FCSD-p excels in more visually complex or dynamically unstable environments, where its adaptive supervision leads to stronger generalization and robustness. Thus, the choice between FCSD-p and FCSD-r should be guided by the environment's visual variability and training dynamics. It is also worth noting that FCSD achieves higher sample efficiency

than model-based methods such as Dreamer [59] and DreamerV2 [60] in several tasks, despite not relying on any learned environment model.

We further apply FCSD to the standard Soft Actor-Critic (SAC) algorithm. Since SAC does not include any built-in data augmentation or auxiliary representation learning, it provides a clean and controlled testbed to isolate the effects of FCSD. As shown in Figure 8, FCSD significantly enhances both the sample efficiency and final performance of vanilla SAC across multiple tasks. These results indicate that FCSD not only improves advanced methods like DrQv2, but also provides substantial benefits for more basic reinforcement learning algorithms.

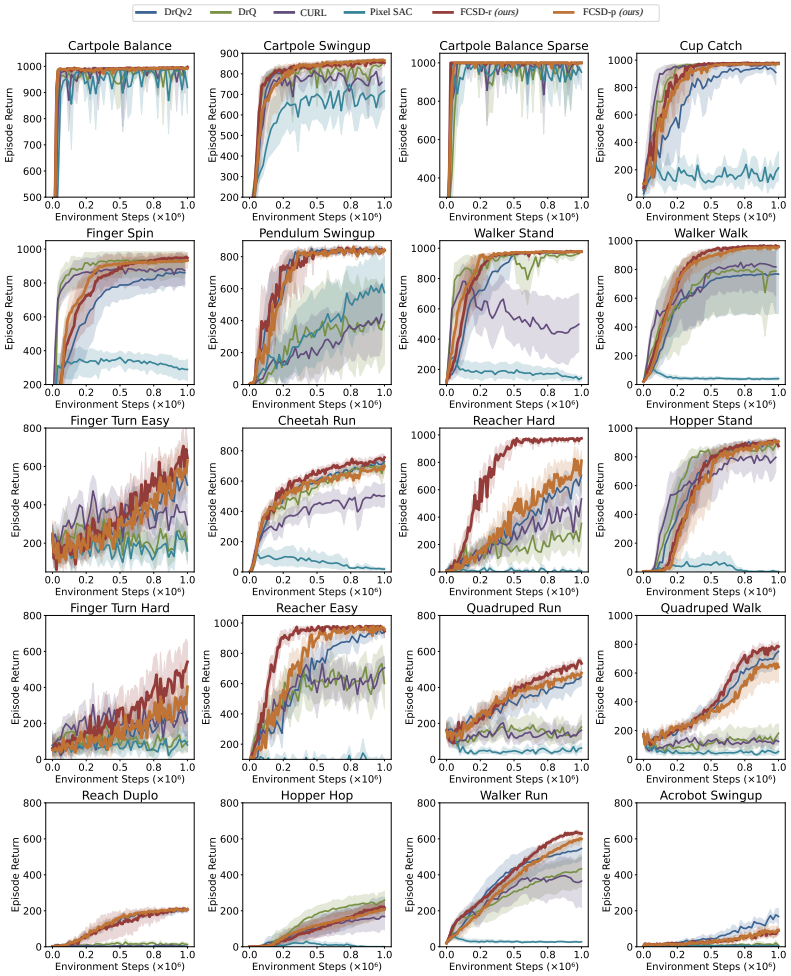


Figure 7. Performance of FCSD integrated into DrQv2 on DMControl benchmark. The mean and standard deviation are computed over 10 runs with different random seeds. Aggregated scores across all 20 tasks are reported in Table 1.

5.3. Compatibility across RL algorithms

Integration with PPO. Tables 2 and 3 report the performance improvements obtained by integrating FCSD into PPO on the ProcGen benchmark. Beyond significantly outperforming the PPO baseline, FCSD also demonstrates strong competitiveness against other data augmentation-based methods. Specifically, it achieves the highest average score in 8 out of 16 games during both training and test, while maintaining competitive performance on the remaining tasks. These results suggest that FCSD strikes a better balance between training effectiveness and generalization ability.

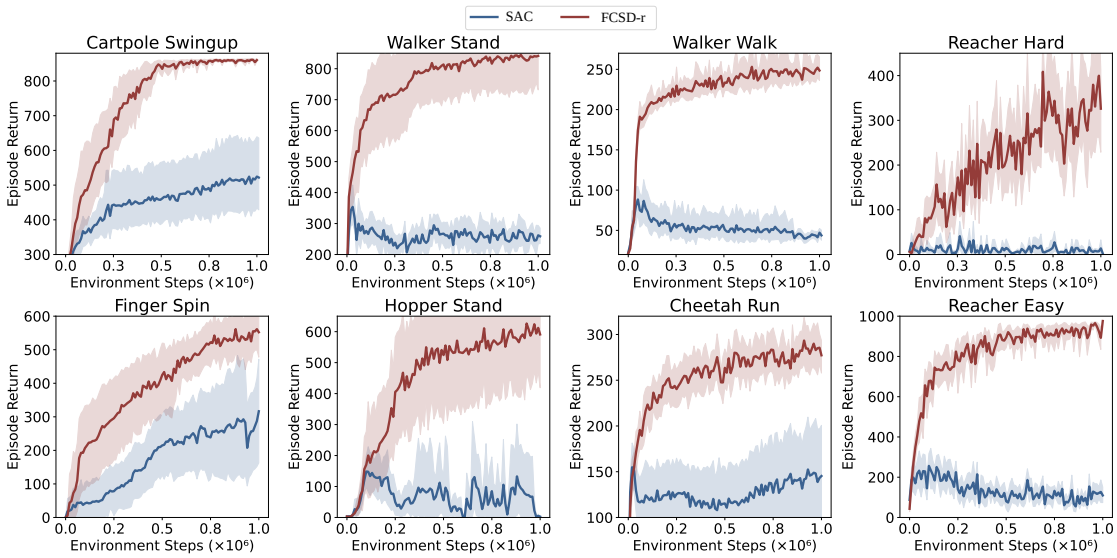


Figure 8. The performance of FCSD integrated into SAC on 8 tasks of DMControl. Under the same number of environment steps, FCSD achieves higher performance with faster convergence. The mean and standard deviation are computed over 10 runs with different seeds.

Table 2. Training Performance for the ProcGen benchmark after training on 25M environment steps. Our method is integrated into PPO, and all agents are trained on 200 levels and evaluated on the full distribution of levels for each game. Results are averaged over 10 random seeds, and both mean and standard deviation are reported.

	Bigfish	**Starpilot**	**Fruitbot**	**Bossfight**	**Ninja**	**Plunder**	**Caveflyer**	**Coinrun**
PPO baseline	9.2 ± 2.7	29.0 ± 1.1	28.8 ± 0.6	8.0 ± 0.4	7.3 ± 0.3	6.1 ± 0.8	6.7 ± 0.6	9.4 ± 0.3
Mixreg	15.0 ± 1.3	28.7 ± 1.1	29.9 ± 0.5	7.9 ± 0.8	8.2 ± 0.4	6.2 ± 0.3	6.2 ± 0.7	9.5 ± 0.2
RAD	13.1 ± 2.8	36.5 ± 3.9	26.1 ± 3.0	8.1 ± 1.1	8.9 ± 0.9	8.4 ± 1.5	6.0 ± 0.8	9.6 ± 0.4
UCB-DrAC	13.2 ± 2.2	35.3 ± 2.2	29.5 ± 1.2	8.2 ± 0.8	8.5 ± 0.3	11.1 ± 1.6	5.7 ± 0.6	9.5 ± 0.3
ExpGen	7.0 ± 0.4	34.3 ± 1.6	28.9 ± 0.6	7.9 ± 0.6	8.5 ± 0.3	5.8 ± 1.4	6.8 ± 0.4	9.8 ± 0.1
FCSD-r ours	14.1 ± 2.2	45.1 ± 1.6	29.5 ± 1.3	10.0 ± 0.9	8.5 ± 1.1	7.7 ± 0.6	8.0 ± 0.9	10.0 ± 0.0
	Jumper	**Chaser**	**Climber**	**Dodgeball**	**Heist**	**Leaper**	**Maze**	**Miner**
PPO baseline	8.3 ± 0.2	4.1 ± 0.3	6.9 ± 1.0	5.3 ± 2.3	7.1 ± 0.5	5.5 ± 0.4	9.1 ± 0.2	11.7 ± 0.5
Mixreg	8.5 ± 0.4	3.4 ± 0.9	7.5 ± 0.8	9.1 ± 0.5	4.4 ± 0.3	3.2 ± 1.2	8.7 ± 0.7	8.9 ± 0.9
RAD	8.6 ± 0.4	6.4 ± 1.0	9.3 ± 1.1	5.0 ± 0.7	6.2 ± 0.9	4.9 ± 0.9	8.4 ± 0.7	12.6 ± 1.0
UCB-DrAC	8.1 ± 0.7	7.6 ± 1.0	9.0 ± 0.4	8.3 ± 0.9	6.9 ± 0.4	5.3 ± 0.5	8.7 ± 0.6	12.5 ± 0.2
ExpGen	7.9 ± 0.2	4.7 ± 1.8	7.7 ± 0.4	5.8 ± 0.5	9.4 ± 0.1	4.3 ± 2.0	9.6 ± 0.1	9.0 ± 0.8
FCSD-r ours	9.1 ± 1.1	6.8 ± 1.0	10.4 ± 1.0	9.6 ± 1.2	6.4 ± 1.0	7.3 ± 1.2	9.1 ± 0.5	12.4 ± 0.5

In contrast, methods such as RAD [35] and MixReg [38] enhance generalization through aggressive augmentations, but often suffer from feature drift and instability in value estimation, an issue particularly pronounced in on-policy algorithms like PPO, where each sample is used only once. Similarly, UCB-DrAC [14] and ExpGen [58] improve generalization by automatically selecting effective data augmentations or behavioral diversification, but they do not explicitly address the instability caused by augmentation. FCSD mitigates this issue through self-value distillation and latent feature consistency, leading to more stable value learning under augmented views.

These findings indicate that FCSD is also well suited for on-policy reinforcement learning algorithms, providing stable, architecture-agnostic improvements across a wide range of challenging visual tasks.

Integration with SPR. As shown in Table 4, our methods FCSD-p and FCSD-r outperform the baseline SPR (Baseline results are reported from [41]), as well as other data augmentation-based visual reinforcement learning methods, including CURL [25], DrQ [39], PSRL [56], and DynaSTI [57], in terms of aggregated performance metrics, namely IQM, median, and mean human-normalized scores. This improvement primarily stems from FCSD's ability to effectively alleviate representation drift and value estimation instability caused by data augmentation, leading to higher returns even in the early stages of training. These results demonstrate that FCSD is not only effective but also readily compatible with off-policy methods such as SPR.

Table 3. Test Performance for the ProcGen benchmark after training on 25M environment steps. Our method is integrated into PPO, and all agents are trained on 200 levels and evaluated on the full distribution of levels for each game. Results are averaged over 10 random seeds, and both mean and standard deviation are reported.

	Bigfish	**Starpilot**	**Fruitbot**	**Bossfight**	**Ninja**	**Plunder**	**Caveflyer**	**Coinrun**
PPO baseline	3.7 ± 1.3	24.9 ± 1.0	26.2 ± 1.2	7.4 ± 0.4	5.9 ± 0.2	5.2 ± 0.6	5.1 ± 0.4	8.6 ± 0.2
Mixreg	7.1 ± 1.6	32.4 ± 1.5	27.3 ± 0.8	8.2 ± 0.7	6.8 ± 0.5	5.9 ± 0.5	6.1 ± 0.6	8.6 ± 0.3
RAD	9.9 ± 1.7	33.4 ± 5.1	27.3 ± 1.8	7.9 ± 0.6	6.9 ± 0.8	8.5 ± 1.2	5.1 ± 0.6	9.0 ± 0.8
UCB-DrAC	9.7 ± 1.0	30.2 ± 2.8	28.1 ± 0.9	8.3 ± 0.8	6.9 ± 0.6	8.9 ± 1.0	5.3 ± 0.9	8.5 ± 0.6
ExpGen	6.0 ± 0.5	31.0 ± 0.9	26.2 ± 0.6	7.7 ± 0.2	6.6 ± 0.2	5.5 ± 1.3	5.7 ± 0.3	8.8 ± 0.1
FCSD-r ours	7.4 ± 1.5	38.3 ± 1.1	28.2 ± 1.2	8.4 ± 1.2	6.4 ± 0.9	6.6 ± 0.7	6.2 ± 1.1	8.8 ± 1.0
	Jumper	**Chaser**	**Climber**	**Dodgeball**	**Heist**	**Leaper**	**Maze**	**Miner**
PPO baseline	5.9 ± 0.2	3.5 ± 0.9	5.6 ± 0.5	1.6 ± 0.1	2.5 ± 0.6	4.9 ± 2.2	5.5 ± 0.3	8.4 ± 0.7
Mixreg	6.0 ± 0.3	5.8 ± 1.1	6.9 ± 0.7	1.7 ± 0.4	2.6 ± 0.4	5.3 ± 1.1	5.2 ± 0.5	9.4 ± 0.4
RAD	6.5 ± 0.6	5.9 ± 1.0	6.9 ± 0.8	2.8 ± 0.7	4.1 ± 1.0	4.3 ± 1.0	6.1 ± 1.0	9.4 ± 1.2
UCB-DrAC	6.4 ± 0.6	5.7 ± 0.6	6.5 ± 0.8	4.7 ± 0.7	4.0 ± 0.7	5.0 ± 0.3	6.3 ± 0.6	9.7 ± 0.7
ExpGen	6.7 ± 0.3	3.6 ± 1.6	5.9 ± 0.5	2.9 ± 0.3	7.4 ± 0.2	4.0 ± 1.7	8.3 ± 0.2	8.0 ± 0.7
FCSD-r ours	6.0 ± 1.0	6.4 ± 1.3	8.4 ± 1.3	3.0 ± 1.2	2.0 ± 1.2	6.7 ± 1.6	5.9 ± 1.0	9.9 ± 1.0

Table 4. Aggregated performance on 26 Atari games after 100K environment steps. FCSD is integrated into SPR for all experiments. All results are computed over 10 random seeds.

Game	**CURL**	**SPR baseline**	**DrQ**	**PSRL**	**DynaSTI**	**FCSD-p ours**	**FCSD-r ours**
IQM	0.203	0.485	0.265	0.451	0.305	0.547	0.568
Median	0.175	0.415	0.268	0.345	0.293	0.522	0.488
Mean	0.381	0.704	0.357	0.610	0.631	0.738	0.719

5.4. Ablation study and analysis

5.4.1. Analysis experiments

To better understand how FCSD contributes to training stability and improved sample efficiency in reinforcement learning, we examine the dynamics of gradient behavior throughout training. Training instability often manifests as sharp fluctuations in gradient magnitudes or frequent changes in gradient directions, which can lead to inefficient learning and slow convergence [61,62]. Monitoring these metrics therefore provides valuable insight into the stability of the optimization process.

As illustrated in Figures 5 and 6, FCSD reduces overall gradient norms, suppresses variance in gradient magnitudes, and minimizes directional changes across updates. Notably, FCSD-r demonstrates a stronger stabilizing effect compared to FCSD-p in DMControl tasks. This observation aligns with the performance trends reported in Table 1, where both variants improve baseline sample efficiency, but FCSD-r achieves more significant gains. Overall, our findings suggest that FCSD enhances training efficiency in visual RL by stabilizing gradient behavior. This promotes smoother policy updates, more predictable learning dynamics, and ultimately, faster and more reliable convergence.

5.4.2. Ablation on different components

We systematically remove each of the two main modules in our method, FC and SD, to observe their individual impacts on sample efficiency thereby assess their contributions. Given the demonstrated advantages of SD-r in the DMControl, as shown in Table 1, all mentions of SD in this and subsequent sections refer to SD-r. The experimental setup here is consistent with that described previously in Section 5.1.1. As detailed in Table 5, it is clear that each module plays a critical role in enhancing the overall functionality of our method. Without the FC module, the model significantly struggles to integrate and process the variable inputs resulting from data augmentation, which notably impairs the training data efficiency in RL. Without the SD module, the model cannot address the large fluctuations in value estimation, leading to training instability and adversely affecting sample efficiency. Therefore, maintaining both modules is crucial for optimizing the learning process and achieving better results.

Table 5. Ablation on different components on the DMControl after 500K and 1000K environment steps. Results are averaged over 10 random seeds. The baseline for comparison is DrQv2.

	Method	**Cup Catch**	**Walker Walk**	**Reacher Easy**	**Reacher Hard**
500k	Baseline	844 ± 174	696 ± 370	748 ± 229	355 ± 173
	SD	968 ± 94	903 ± 83	924 ± 62	838 ± 56
	FC	909 ± 103	872 ± 184	834 ± 195	728 ± 165
	FCSD	972 ± 22	930 ± 57	965 ± 35	962 ± 24
1000k	Baseline	859 ± 128	787 ± 150	885 ± 125	682 ± 132
	SD	976 ± 28	947 ± 63	953 ± 72	962 ± 22
	FC	957 ± 69	923 ± 92	904 ± 90	873 ± 109
	FCSD	978 ± 11	960 ± 34	960 ± 32	975 ± 15

5.4.3. Comparison with other regularization methods

Research suggests that distillation and regularization have similar effects [63–65]. To validate the advantages of our SD in the regularization of visual RL, we compare SD with other common regularization techniques, including L_2 regularization and Dropout. Through these comparisons, we aim to assess the impact of SD on improving model performance and sample efficiency. As Table 6 shows, SD consistently surpasses the performance and sample efficiency of traditional regularization techniques. Specifically, it achieves higher cumulative rewards with fewer environment steps. Our findings highlight the greater benefits of using SD in visual RL, showcasing its ability to increase model stability through better regularization [66], thereby improving training efficiency.

Table 6. Comparison with other regularization methods on the DMControl after 500K and 1000K environment steps. Results of ours are averaged over 10 random seeds. The baseline for comparison is DrQv2.

	Method	**Cup Catch**	**Walker Walk**	**Reacher Easy**	**Reacher Hard**
500k	Baseline	844 ± 174	696 ± 370	748 ± 229	355 ± 173
	L_2	852 ± 103	712 ± 189	791 ± 167	414 ± 121
	Dropout	836 ± 167	687 ± 249	752 ± 197	353 ± 132
	SD	968 ± 94	903 ± 83	924 ± 62	838 ± 56
1000k	Baseline	859 ± 128	787 ± 150	885 ± 125	682 ± 132
	L_2	884 ± 74	893 ± 93	915 ± 59	822 ± 87
	Dropout	861 ± 103	762 ± 134	896 ± 93	691 ± 110
	SD	976 ± 28	947 ± 63	953 ± 72	962 ± 22

5.4.4. Ablation on data augmentation

To assess the impact of data augmentation on our proposed method, we evaluate the performance of our FCSD and its baseline, DrQv2, under various data augmentation conditions in the DMControl. As shown in Table 7, we analyze how our method performs without and with data augmentations, specifically focusing on cropping and translation techniques. Firstly, when data augmentations are applied, our method demonstrates higher sample efficiency than its baseline, achieving better performance with relatively fewer training steps and continuously improving performance over long-term training, resulting in higher scores. Secondly, the results show that our method is capable of enhancing sample efficiency and performance even without the use of data augmentation. This improvement can be attributed to the SD, which stabilizes value estimates and reduces their uncertainty. Additionally, when comparing the effects of using data augmentation versus not using it, our FCSD shows a greater improvement compared to the baseline. This indicates that FCSD benefits more from data augmentation, demonstrating the effectiveness of our method.

Table 7. Ablation on data augmentation on the DMControl after 500K and 1000K environment steps. Results of ours are averaged over 10 random seeds.

	DrQv2 baseline	**Cup Catch**	**Walker Walk**	**Reacher Easy**	**Reacher Hard**
500k	NoAug	772 ± 118	621 ± 191	685 ± 171	289 ± 97
	Crop	844 ± 174	696 ± 370	748 ± 229	355 ± 173
	Translate	831 ± 142	702 ± 434	682 ± 217	317 ± 128
1000k	NoAug	807 ± 89	753 ± 105	837 ± 102	593 ± 74
	Crop	859 ± 128	787 ± 150	885 ± 125	682 ± 132
	Translate	846 ± 119	793 ± 165	801 ± 119	619 ± 174
	FCSD ours	**Cup Catch**	**Walker Walk**	**Reacher Easy**	**Reacher Hard**
500k	NoAug	889 ± 32	843 ± 63	797 ± 45	732 ± 23
	Crop	972 ± 22	930 ± 57	965 ± 35	962 ± 24
	Translate	923 ± 16	901 ± 69	967 ± 14	952 ± 28
1000k	NoAug	901 ± 21	914 ± 19	903 ± 63	824 ± 48
	Crop	978 ± 11	960 ± 34	960 ± 32	975 ± 15
	Translate	957 ± 27	938 ± 42	972 ± 7	972 ± 13

6. Conclusion

Data augmentation plays a key role in visual RL by providing diverse training samples, thereby improving generalization and overall performance. However, it also introduces challenges that increase the complexity of representation learning and the uncertainty of value estimation, potentially creating a bottleneck to further improvements in sample efficiency. To address these issues, we propose FCSD, encouraging consistent representations across augmented views and stabilizes value learning through self-value distillation. By mitigating augmentation-induced instability, FCSD effectively improves sample efficiency. Extensive evaluations on diverse RL benchmarks demonstrate that FCSD consistently improves both policy performance and sample efficiency, without relying on complex auxiliary tasks or requiring changes to existing RL architectures. Moreover, FCSD outperforms several data augmentation-based visual RL methods across different environments, demonstrating its robustness and practical utility. Future work will explore more fine-grained strategies for consistency enforcement and distillation, as well as extensions to real-world environments.

Data availability statement

No supplementary or additional data were generated in this study.

Declaration of generative AI and AI assisted technologies

During the preparation of this manuscript, the authors used generative AI tools only to improve language and readability. The authors take full responsibility for the content of the manuscript.

Acknowledgments

This work was supported in part by the National Key R&D Program of China under Grant No. 2024YFB4707600, NSFC under grant No. 62125305, No. U23A20339, No. 62573339, No. 62503380, Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China under grant No. JYB2025XDXM210, and Natural Science Foundation of Shaanxi Province under Grant No. 2025SYSSYSZD-083.

Authors' contribution

Conceptualization, Ziru Wang; methodology, Ziru Wang; software, Ziru Wang and Wanli Jiang; validation, Ziru Wang, Wanli Jiang and Qian Kou; formal analysis, Ziru Wang and Ru Peng; investigation, Ziru Wang; resources, Xuguang Lan; data curation, Ziru Wang; writing—original draft preparation, Ziru Wang; writing—review and editing, Wanli Jiang and Xuguang Lan; visualization, Ziru Wang and Qian Kou; supervision, Xuguang Lan; project administration, Xuguang Lan; funding acquisition, Xuguang Lan. All authors have read and agreed to the published version of the manuscript.

Conflicts of interest

Xuguang Lan holds the position of Associate Editor for *Robot Learning* and has not peer reviewed or made any editorial decisions for this paper.

Appendix

A1. DrQv2 series

For DrQv2 and our improved methods based on it, we use default hyperparameters and training settings in [4]. In total, we consider 20 different tasks for our methods, which have been grouped into two categories by [4]: easy tasks (Cartpole Balance, Cartpole Balance Sparse, Cartpole Swingup, Cup Catch, Finger Spin, Hopper Stand, Pendulum Swingup, Walker Stand, Walker Walk) and medium tasks (Acrobot Swingup, Cheetah Run, Finger Turn Easy, Finger Turn Hard, Hopper Hop, Quadruped Run, Quadruped Walk, Reach Duplo, Reacher Easy, Reacher Hard, Walker Run). The complete list of hyperparameters, including our new hyperparameters, is provided in Table A1. However, there are a few specific deviations for certain tasks:

Walker Stand/Walk/Run: Following DrQv2, we use mini-batch size of 512 and n-step return of 1.

Quadruper Run: Following DrQv2, we use replay buffer size of 10^5.

Finger Turn Easy: We set λ in PM to 0.1.

Cartpole Balance, Cartpole Balance Sparse, Cartpole Swingup, Finger Spin, Hopper Stand, Pendulum Swingup, Acrobot Swingup, Hopper Hop: We set β in FCSD-r to 0.001.

Table A1. Hyperparameters for DrQv2 series experiments.

Parameter	Setting
β in FCSD-p	0.01
τ in FCSD-p	0.01
β in FCSD-r	0.1
Replay buffer Capacity	10^6
Action repeat	2
Seed frames	4000
Exploration steps	2000
n-step returns	3
Mini-batch size	256
Discount γ	0.99
Optimizer	Adam
Learning rate	10^{-4}
Agent update frequency	2
Critic Q-function soft-update rate τ	0.01
Features dim	50
Hidden dim	1024
Exploration stddev. clip	0.3
Exploration stddev. schedule	easy: linear(1.0, 0.1, 100000) medium: linear(1.0, 0.1, 500000)

We use the DrQv2 codebase from https://github.com/facebookresearch/drqv2. https://github.com/rraileanu/idaac.

A2. PPO series

We utilize the open-source implementation of PPO from [67] as the foundation for our methods. The results obtained for PPO on ProcGen in our paper are from [67]. The hyperparameters used to obtain the results of our method can be found in Table A2.

Table A2. Hyperparameters for PPO series experiments.

Parameter	Setting
β in FCSD-r	0.01
Discount γ	0.999
Timesteps per rollout	256
Epochs per rollout	3
Minibatches per epoch	8
Entropy bonus	0.01
Clip range	0.2
Reward normalization	yes
Learning rate	5e-4
Workers	1
Environments per worker	64
Total timesteps	25M
Optimizer	Adam
LSTM	no
Frame stack	no

A3. SPR series

For FCSD-r and FCSD-p, we adhere to the same hyperparameters and training specifics as SPR [41] We furnish a complete set of hyperparameters in Table A3, inclusive of the new hyperparameters introduced by our methods. The performance across different tasks is reported in Table 4.

Table A3. Hyperparameters for SPR series on Atari100K.

Parameter	Setting
β in FCSD-p	0.01
τ in FCSD-p	0.01
β in FCSD-r	2.0
Gray-scaling	True
Observation down-sampling	84×84
Frames stacked	4
Action repetitions	4
Reward clipping	$[-1, 1]$
Terminal on loss of life	True
Max frames per episode	108K
Update	Distributional Q
Dueling	True
Support of Q-distribution	51
Discount factor	0.99
Minibatch size	32
Optimizer	Adam
Optimizer learning rate	0.0001
Optimizer: β_1	0.9
Optimizer: β_2	0.999
Optimizer: ε	0.00015
Max gradient norm	10
Priority exponent	0.5
Priority correction	$0.4 \rightarrow 1$
Exploration	Noisy nets
Noisy nets parameter	0.5
Training steps	100K
Evaluation trajectories	100
Min replay size for sampling	2000
Replay peried every	1 step
Update per step	2
Multi-step return length	10
Q network: channels	32, 64, 64
Q network: filter size	$8 \times 8, 4 \times 4, 3 \times 3$
Q network: stride	$4, 2, 1$
Q network: hidden units	256
Non-linearity	ReLU
Target network: update period	1
SPR loss coefficient	2
Prediction Depth	5

We use the SPR codebase from https://github.com/mila-iqia/spr.

References

[1] Zhang S, Wang J, Wang E, Sun Y. An autonomous stair climbing method for the crawler robot based on a hybrid reinforcement learning controller. *Neurocomputing* 2024, 593:127802.

[2] Zhang Z, Wei Z, Huang Z, Niu R, Wang P. One for all: one-stage referring expression comprehension with dynamic reasoning. *Neurocomputing* 2023, 518:523–532.

[3] Yi Q, Zhang R, Peng S, Guo J, Hu X, *et al.* Learning controllable elements oriented representations for reinforcement learning. *Neurocomputing* 2023, 549:126455.

[4] Yarats D, Fergus R, Lazaric A, Pinto L. Mastering visual continuous control: improved data-augmented reinforcement learning. In *International Conference on Learning Representations*, Virtual Conference, April 25–29, 2022.

[5] Liu S, Cao J, Wang Y, Chen W, Liu Y. Self-play reinforcement learning with comprehensive critic in computer games. *Neurocomputing* 2021, 449:207–213.

[6] Xu H, Yan Z, Xuan J, Zhang G, Lu J. Improving proximal policy optimization with alpha divergence. *Neurocomputing* 2023, 534:94–105.

[7] Lu C, Schroecker Y, Gu A, Parisotto E, Foerster J, *et al.* Structured state space models for in-context reinforcement learning. In *Advances in Neural Information Processing Systems*, New Orleans, USA, December 10–16, 2023.

[8] Yarats D, Zhang A, Kostrikov I, Amos B, Pineau J, *et al.* Improving sample efficiency in model-free reinforcement learning from images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Virtual Conference, February 2–9, 2021, pp. 10674–10681.

[9] Yang S, Xiao W, Zhang M, Guo S, Zhao J, *et al.* Image data augmentation for deep learning: A survey. *arXiv* 2022, arXiv:2204.08610.

[10] Shorten C, Khoshgoftaar TM. A survey on image data augmentation for deep learning. *Int. J. Big Data Intel.* 2019, 6(1):1–48.

[11] Shorten C, Khoshgoftaar TM, Furht B. Text data augmentation for deep learning. *Int. J. Big Data Intel.* 2021, 8(1):101.

[12] Maharana K, Mondal S, Nemade B. A review: data pre-processing and data augmentation techniques. *Global Transitions Proc.* 2022, 3(1):91–99.

[13] Ma G, Zhang L, Wang H, Li L, Wang Z, *et al.* Learning better with less: Effective augmentation for sample-efficient visual reinforcement learning. In *Advances in Neural Information Processing Systems*, New Orleans, USA, December 10–16, 2023, pp. 59832–59859.

[14] Raileanu R, Goldstein M, Yarats D, Kostrikov I, Fergus R. Automatic data augmentation for generalization in reinforcement learning. In *Advances in Neural Information Processing Systems*, Montreal, Canada, December 6–14, 2021, pp. 5402–5415.

[15] Fan J, Li W. Dribo: robust deep reinforcement learning via multi-view information bottleneck. In *International Conference on Machine Learning*, Baltimore, USA, July 17–23, 2022, pp. 6074–6102.

[16] Sun C, Qian H, Miao C. Cclf: A contrastive-curiosity-driven learning framework for sample-efficient reinforcement learning. *arXiv* 2022, arXiv:2205.00943.

[17] Hu J, Jiang Y, Weng P. Revisiting data augmentation in deep reinforcement learning. In *The Twelfth International Conference on Learning Representations*, Vienna, Austria, May 7–11, 2024.

[18] Kurtuluş E, Li Z, Dauphin Y, Cubuk ED. Tied-augment: controlling representation similarity improves data augmentation. In *International Conference on Machine Learning*, Honolulu, USA, July 23–29, 2023, pp. 17994–18007.

[19] Yang S, Dong Y, Ward R, Dhillon IS, Sanghavi S, *et al.* Sample efficiency of data augmentation consistency regularization. In *International Conference on Artificial Intelligence and Statistics*, Valencia, Spain, April 25–27, 2023, pp. 3825–3853.

[20] Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, *et al.* An image is worth 16×16 words: transformers for image recognition at scale. In *International Conference on Learning Representations*, Virtual Event, Austria, May 3–7, 2021.

[21] Zhong Z, Zheng L, Kang G, Li S, Yang Y. Random erasing data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, New York, USA, February 7–12, 2020, pp. 13001–13008.

[22] Hong Y, Zhang J, Sun Z, Yan K. Safa: sample-adaptive feature augmentation for long-tailed image classification. In *European Conference on Computer Vision*, Tel Aviv, Israel, October 23–27, 2022, pp. 587–603.

[23] Ding Z, Huang Y, Yuan H, Dong H. Introduction to reinforcement learning. In *Deep Reinforcement Learning: Fundamentals, Research and Applications*, Berlin: Springer, 2020, pp. 47–123.

[24] Yu T, Zhang Z, Lan C, Lu Y, Chen Z. Mask-based latent reconstruction for reinforcement learning. In *Advances in Neural Information Processing Systems*, New Orleans, USA, November 28–December 9, 2022, pp. 25117–25131.

[25] Laskin M, Srinivas A, Abbeel P. CURL: contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*, Vienna, Austria, July 13–18, 2020, pp. 5639–5650.

[26] Wang C, Luo X, Ross K, Li D. VRL3: A data-driven framework for visual deep reinforcement learning. In *Advances in Neural Information Processing Systems*, New Orleans, USA, November 28–December 9, 2022, pp. 32974–32988.

[27] Igl M, Farquhar G, Luketina J, Boehmer W, Whiteson S. Transient non-stationarity and generalisation in deep reinforcement learning. In *International Conference on Learning Representations*, Virtual, May 3–7, 2021.

[28] Tsantekidis A, Passalis N, Tefas A. Diversity-driven knowledge distillation for financial trading using deep reinforcement learning. *Neural Networks.* 2021, 140:193–202.

[29] Zhang T, Wang X, Liang B, Yuan B. Catastrophic interference in reinforcement learning: a solution based on context division and knowledge distillation. *IEEE Trans. Neural Networks.* 2022, 34(12):9925–9939.

[30] Yu T, Kumar S, Gupta A, Levine S, Hausman K, *et al.* Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, Virtual, December 6–12, 2020, pp. 5824–5836.

[31] Liu B, Liu X, Jin X, Stone LQ Peter and. Conflict-averse gradient descent for multi-task learning. In *Advances in Neural Information Processing Systems*, Virtual, December 6–14, 2021, pp. 18878–18890.

[32] Tassa Y, Doron Y, Muldal A, Erez T, Li Y, *et al.* Deepmind control suite. *arXiv* 2018, arXiv:1801.00690.

[33] Cobbe K, Hesse C, Hilton J, Schulman J. Leveraging procedural generation to benchmark reinforcement learning. In *International Conference on Machine Learning*, Virtual, July 13–18, 2020, pp. 2048–2056.

[34] Bellemare MG, Naddaf Y, Veness J, Bowling M. The arcade learning environment: an evaluation platform for general agents. *J. Artif. Intell. Res.* 2013, 47:253–279.

[35] Laskin M, Lee K, Stooke A, Pinto L, Abbeel P, *et al.* Reinforcement learning with augmented data. In *Advances in Neural Information Processing Systems*, Virtual Conference, December 6–12, 2020, pp. 19884–19895.

[36] Wu K, Wu M, Chen Z, Xu Y, Li X. Generalizing reinforcement learning through fusing self-supervised learning into intrinsic motivation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Virtual Conference, February 22–March 1, 2022, pp. 8683–8690.

[37] Nafi NM, Hsu W. Reinforcement learning with augmentation invariant representation: a non-contrastive approach. In *NeurIPS 2023 Workshop on Generalization in Planning*, New Orleans, USA, December 16, 2023.

[38] Wang K, Kang B, Shao J, Feng J. Improving generalization in reinforcement learning with mixture regularization. In *Advances in Neural Information Processing Systems*, Virtual Conference, December 6–12, 2020, pp. 7968–7978.

[39] Yarats D, Kostrikov I, Fergus R. Image augmentation is all you need: regularizing deep reinforcement learning from pixels. In *International conference on learning representations*, Virtual Conference, May 3–7, 2021.

[40] Zhang H, Cisse M, Dauphin YN, Lopez-Paz D. mixup: Beyond empirical risk minimization. *arXiv* 2017, arXiv:1710.09412.

[41] Schwarzer M, Anand A, Goel R, Hjelm RD, Courville A, *et al.* Data-efficient reinforcement learning with self-predictive representations. *arXiv* 2020, arXiv:2007.05929.

[42] Yu T, Lan C, Zeng W, Feng M, Zhang Z, *et al.* Playvirtual: augmenting cycle-consistent virtual trajectories for reinforcement learning. In *Advances in Neural Information Processing Systems*, Virtual Conference, December 6–14, 2021, pp. 5276–5289.

[43] Zhu J, Xia Y, Wu L, Deng J, Zhou W, *et al.* Masked contrastive representation learning for reinforcement learning. *IEEE Trans. Pattern Anal. Mach. Intell.* 2022, 45(3):3421–3433.

[44] Mazoure B, Tachet des Combes R, Doan TL, Bachman P, Hjelm RD. Deep reinforcement and infomax learning. In *Advances in Neural Information Processing Systems*, Virtual Conference, December 6–12, 2020, pp. 3686–3698.

[45] Parisi S, Rajeswaran A, Purushwalkam S, Gupta A. The unsurprising effectiveness of pre-trained vision models for control. In *International Conference on Machine Learning*, Baltimore, USA, July 17–23, 2022, pp. 17359–17371.

[46] Xiao T, Radosavovic I, Darrell T, Malik J. Masked visual pre-training for motor control. *arXiv* 2022, arXiv:2203.06173.

[47] Seo Y, Hafner D, Liu H, Liu F, James S, *et al.* Masked world models for visual control. In *Conference on Robot Learning*, Atlanta, USA, November 6–9, 2023, pp. 1332–1344.

[48] Deng F, Jang I, Ahn S. Dreamerpro: Reconstruction-free model-based reinforcement learning with prototypical representations. In *International Conference on Machine Learning*, Baltimore, USA, July 17–23, 2022, pp. 4956–4975.

[49] Wu P, Escontrela A, Hafner D, Abbeel P, Goldberg K. Daydreamer: World models for physical robot learning. In *Conference on Robot Learning*, Atlanta, USA, November 6–9, 2023, pp. 2226–2240.

[50] Puterman ML. In *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, 1st ed. New York: John Wiley & Sons, 2014.

[51] Haarnoja T, Zhou A, Abbeel P, Levine S. Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, Stockholm, Sweden, July 10–15, 2018, pp. 1861–1870.

[52] Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, *et al.* Continuous control with deep reinforcement learning. In *4th International Conference on Learning Representations (ICLR 2016)*, San Juan, Puerto Rico, May 2–4, 2016.

[53] Safaryan M, Peste A, Alistar D. Knowledge distillation performs partial variance reduction. In *Advances in Neural Information Processing Systems*, Vancouver, Canada, December 9–15, 2024.

[54] Sarnthein F, Bachmann G, Anagnostidis S, Hofmann T. Random teachers are good teachers. In *International Conference on Machine Learning*, Honolulu, USA, July 23–29, 2023, pp. 30022–30041.

[55] Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. *arXiv* 2017, arXiv:1707.06347.

[56] Choi H, Lee H, Song W, Jeon S, Sohn K, *et al.* Local-guided global: paired similarity representation for visual reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Vancouver, Canada, June 18–22, 2023, pp. 15072–15082.

[57] Kim J, Lee YJ, Kwak M, Park YJ, Kim SB. DynaSTI: dynamics modeling with sequential temporal information for reinforcement learning in Atari. *Knowledge-Based Syst.* 2024, 299:112103.

[58] Zisselman E, Lavie I, Soudry D, Tamar A. Explore to generalize in zero-shot rl. In *Advances in Neural Information Processing Systems*, New Orleans, USA, December 10–16, 2023, pp. 63174–63196.

[59] Hafner D, Lillicrap T, Ba J, Norouzi M. Dream to control: learning behaviors by latent imagination. In *International Conference on Learning Representations*, Addis Ababa, Ethiopia, April 26–30, 2020.

[60] Hafner D, Lillicrap T, Norouzi M, Ba J. Mastering atari with discrete world models. *arXiv* 2020, arXiv:2010.02193.

[61] Farnia F, Ozdaglar A. Train simultaneously, generalize better: stability of gradient-based minimax learners. In *International Conference on Machine Learning*, Virtual Conference, July 18–24, 2021, pp. 3174–3185.

[62] Hardt M, Recht B, Singer Y. Train faster, generalize better: stability of stochastic gradient descent. In *International conference on machine learning*, New York, USA, June 19–24, 2016, pp. 1225–1234.

[63] Mobahi H, Farajtabar M, Bartlett P. Self-distillation amplifies regularization in hilbert space. In *Advances in Neural Information Processing Systems*, Virtual Conference, December 6–12, 2020, pp. 3351–3361.

[64] Wang J, Zhang P, He Q, Li Y, Hu Y. Revisiting label smoothing regularization with knowledge distillation. *Applied Sciences.* 2021, 11(10):4699.

[65] Yuan L, Tay FE, Li G, Wang T, Feng J. Revisiting knowledge distillation via label smoothing regularization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, Virtual Conference, June 14–19, 2020, pp. 3903–3911.

[66] Wu L, Su WJ. The implicit regularization of dynamical stability in stochastic gradient descent. In *International Conference on Machine Learning*, Honolulu, USA, July 23–29, 2023, pp. 37656–37684.

[67] Raileanu R, Fergus R. Decoupling value and policy for generalization in reinforcement learning. In *International Conference on Machine Learning*, Virtual Conference, July 18–24, 2021, pp. 8787–8798.