Stabilizing Deep Q-Learning with ConvNets and Vision Transformers under Data Augmentation

Nicklas Hansen
UC San Diego

Hao Su
UC San Diego

Xiaolong Wang
UC San Diego

***Abstract*—While agents trained by Reinforcement Learning (RL) can solve increasingly challenging tasks directly from visual observations, generalizing learned skills to novel environments remains very challenging. Extensive use of data augmentation is a promising technique for improving generalization in RL, but it is often found to decrease sample efficiency and can even lead to divergence. In this paper, we investigate causes of instability when using data augmentation in common off-policy RL algorithms. We identify two problems, both rooted in high-variance Q-targets. Based on our findings, we propose a simple yet effective technique for stabilizing this class of algorithms under augmentation. We perform extensive empirical evaluation using both ConvNets and Vision Transformers (ViT) on a family of benchmarks based on DeepMind Control Suite, as well as in robotic manipulation tasks. Our method greatly improves stability and sample efficiency of ConvNets under augmentation, and achieves generalization results competitive with state-of-the-art methods for image-based RL. We further show that our method scales to RL with ViT-based architectures, and that data augmentation may be especially important in this setting.**

I. Introduction

Reinforcement Learning (RL) from visual observations has achieved tremendous success [19, 1, 32, 16, 35]. However, it is still very challenging for current methods to generalize skills to novel environments, and policies trained by RL can easily overfit to the training environment [34, 5], especially for high-dimensional observation spaces, e.g. images [2, 24].

Increasing the variability in training data via domain randomization [31, 20] and data augmentation [23, 14, 13, 21] has demonstrated encouraging results for learning policies invariant to changes in the environment. Specifically, recent works on data augmentation [14, 13, 11] both show improvements in sample efficiency from simple cropping and translation augmentations, but also conclude that stronger data augmentation in fact *decreases* sample efficiency and even cause divergence. While these augmentations have the potential to improve generalization, the increasingly varied data makes the optimization more challenging and risks instability. Unlike supervised learning, balancing the trade-off between stability and generalization in RL requires substantial trial and error.

In this paper, we illuminate theoretically grounded causes of instability when using data augmentation in off-policy RL [19, 17, 6, 8]. Specifically, we find two main causes of instability in previous work's application of data augmentation: (i) indiscriminate application of data augmentation resulting in high-variance Q-targets; and (ii) that Q-value estimation strictly from augmented data results in over-regularization.

To address these problems, we propose **SVEA**: **S**tabilized Q-**V**alue **E**stimation under **A**ugmentation, a simple yet effective framework for data augmentation in off-policy RL that greatly improves stability of Q-value estimation. Our method consists of the following three components: Firstly, by only applying augmentation in Q-value estimation of the *current* state, *without* augmenting Q-targets used for bootstrapping, SVEA circumvents erroneous bootstrapping caused by data augmentation; Secondly, we formulate a modified Q-objective that optimizes Q-value estimation jointly over both augmented and unaugmented copies of the observations; Lastly, for SVEA implemented with an actor-critic algorithm, we optimize the actor strictly on unaugmented data, and instead learn a generalizable policy indirectly through parameter-sharing. Our framework can be implemented efficiently without additional forward passes nor additional learnable parameters.

We verify our hypotheses empirically through extensive experiments on the DeepMind Control Suite [30] and extensions of it, including the DMControl Generalization Benchmark [11] and the Distracting Control Suite [26], as well as a set of robotic manipulation tasks. Our method successfully stabilizes Q-value estimation under a set of strong data augmentations, and achieves sample efficiency, asymptotic performance, and generalization that is competitive or better than previous state-of-the-art methods in all tasks considered, at a significantly lower computational cost. Finally, we show that our method scales to RL with ViT-based architectures, and that data augmentation may be especially important in this setting.

II. Preliminaries

Common model-free off-policy RL algorithms aim to estimate an optimal state-action value function $Q^* : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ as $Q_\theta(\mathbf{s}, \mathbf{a}) \approx Q^*(\mathbf{s}, \mathbf{a}) = \max_{\pi_\theta} \mathbb{E}\left[R_t | \mathbf{s}_t = \mathbf{s}, \mathbf{a}_t = \mathbf{a}\right]$ using function approximation. In practice, this is achieved by means of the single-step Bellman residual $\left(r(\mathbf{s}_t, \mathbf{a}_t) + \gamma \max_{\mathbf{a}'_t} Q^{\text{tgt}}_\psi(\mathbf{s}_{t+1}, \mathbf{a}'_t)\right) - Q_\theta(\mathbf{s}_t, \mathbf{a}_t)$ [28], where ψ parameterizes a *target* state-action value function Q^{tgt}. We can choose to minimize this residual (also known as the *temporal difference* error) directly wrt θ using a mean squared error loss, which gives us the objective

$$q^{\text{tgt}} = r(\mathbf{s}_t, \mathbf{a}_t) + \gamma \max_{\mathbf{a}'_t} Q^{\text{tgt}}_\psi(\mathbf{s}_{t+1}, \mathbf{a}'_t) \tag{1}$$

$$\mathcal{L}_Q(\theta, \psi) = \mathbb{E}_{\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1} \sim \mathcal{B}} \left[\frac{1}{2} \left[q^{\text{tgt}} - Q_\theta(\mathbf{s}_t, \mathbf{a}_t) \right]^2 \right], \tag{2}$$

where $\mathcal{B}$ is a replay buffer with transitions collected by a behavioral policy [18]. While $Q^{\text{tgt}} = Q$ and periodically setting $\psi \longleftarrow \theta$ exactly recovers the objective of DQN [19], several improvements have been proposed to improve stability of Eq. 2, such as updating target parameters using a slow-moving average of the online Q-network [17]:

$$\psi_{n+1} \longleftarrow (1-\zeta)\psi_n + \zeta\theta_n \tag{3}$$

for an iteration step n and a momentum coefficient $\zeta \in (0, 1]$. As computing $\max_{\mathbf{a}'_t} Q^{\text{tgt}}_\psi(\mathbf{s}_{t+1}, \mathbf{a}'_t)$ in Eq. 2 is intractable for large and continuous action spaces, a number of prominent *actor-critic* algorithms that additionally learn a policy $\pi_\theta(\mathbf{s}_t) \approx \arg\max_{\mathbf{a}_t} Q_\theta(\mathbf{s}_t, \mathbf{a}_t)$ have therefore been proposed [17, 6, 8]. We use Soft Actor-Critic (SAC) [8] in experiments.

III. Pitfalls of Augmentation in Deep Q-Learning

Our goal is to learn a Q-function Q_θ that generalizes to novel MDPs, and we leverage data augmentation as an optimality-invariant state transformation τ between a state $\mathbf{s}$ and its transformed (augmented) counterpart $\mathbf{s}^{\text{aug}} = \tau(\mathbf{s}, \nu)$ with parameters $\nu \sim \mathcal{V}$.

Definition 1 (Optimality-Invariant State Transformation [13])**.** *Given an MDP $\mathcal{M}$, a state transformation $\tau\colon \mathcal{S} \times \mathcal{V} \mapsto \mathcal{S}$ is an optimality-invariant state transformation if $Q(\mathbf{s}, \mathbf{a}) = Q(\tau(\mathbf{s}, \nu), \mathbf{a}) \quad \forall\ \mathbf{s} \in \mathcal{S},\ \mathbf{a} \in \mathcal{A},\ \nu \in \mathcal{V}$, where $\nu \in \mathcal{V}$ parameterizes τ.*

If we choose the set of parameters $\mathcal{V}$ of a state transformation τ to be sufficiently large, we can therefore expect to improve generalization to state spaces not seen during training. However, while naïve application of data augmentation as in [14, 13, 27, 22] may potentially improve generalization, it can be harmful to Q-value estimation. We hypothesize that this is primarily because it dramatically increases the size of the observed state space, and consequently also increases variance $\text{Var}\left[Q(\tau(\mathbf{s}, \nu))\right] \geq \text{Var}\left[Q(\mathbf{s})\right],\ \nu \sim \mathcal{V}$ when $\mathcal{V}$ is large. Concretely, we identify the following two issues:

Pitfall 1: Non-deterministic Q-target. For deep Q-learning algorithms, previous work [14, 13, 27, 22] applies augmentation to both state $\mathbf{s}^{\text{aug}}_t \triangleq \tau(\mathbf{s}_t, \nu)$ and successor state $\mathbf{s}^{\text{aug}}_{t+1} \triangleq \tau(\mathbf{s}_{t+1}, \nu')$ where $\nu, \nu' \sim \mathcal{V}$. Compared with DQN [19] that uses a deterministic (periodically updated) Q-target, this practice introduces a non-deterministic Q-target $r(\mathbf{s}_t, \mathbf{a}_t) + \gamma \max_{\mathbf{a}'_t} Q^{\text{tgt}}_\psi(\mathbf{s}^{\text{aug}}_{t+1}, \mathbf{a}'_t)$ depending on the augmentation parameters ν'. As observed in the original DQN paper, high-variance target values are detrimental to Q-learning algorithms, and may cause divergence due to the "deadly triad" of function approximation, bootstrapping, and off-policy learning [29]. Because data augmentation is inherently non-deterministic, it greatly increases variance in Q-target estimation and exacerbates the issue of volatility. This is particularly troubling in actor-critic algorithms such as DDPG [17] and SAC [8], where the Q-target is estimated from $(\mathbf{s}_{t+1}, \mathbf{a}')$, $\mathbf{a}' \sim \pi(\cdot|\mathbf{s}_{t+1})$, which introduces an additional source of error from π that is non-negligible especially when $\mathbf{s}_{t+1}$ is augmented.

Pitfall 2: Over-regularization. Data augmentation was originally introduced in the supervised learning regime as a regularizer to prevent overfitting of high-capacity models. However, for RL, even learning a policy in the training environment is hard. While data augmentation may improve generalization, it greatly increases the difficulty of policy learning, i.e., optimizing θ for Q_θ and potentially a behavior network π_θ. Particularly, when the temporal difference loss from Eq. 2 cannot be well minimized, the large amount of augmented states dominate the gradient, which significantly impacts Q-value estimation of both augmented and unaugmented states. We refer to this issue as over-regularization.

IV. Method

We propose **SVEA**: **S**tabilized Q-**V**alue **E**stimation under **A**ugmentation, a general framework for generalization by data augmentation in RL. We describe our method in the following.

A. Architectural Overview

An overview is provided in Figure 1. We subdivide the neural network and corresponding learnable parameters of a state-action value function into sub-networks f_θ (denoted the state *encoder*) and Q_θ (denoted the Q-*function*) s.t $q_t \triangleq Q_\theta(f_\theta(\mathbf{s}_t), \mathbf{a}_t)$ is the predicted Q-value corresponding to a given state-action pair $(\mathbf{s}_t, \mathbf{a}_t)$. We define the target state-action value function s.t. $q^{\text{tgt}}_t \triangleq r(\mathbf{s}_t, \mathbf{a}_t) + \gamma \max_{\mathbf{a}'_t} Q^{\text{tgt}}_\psi(f^{\text{tgt}}_\psi(\mathbf{s}_{t+1}), \mathbf{a}')$ is the target Q-value for $(\mathbf{s}_t, \mathbf{a}_t)$, and we define ψ as an exponential moving average of θ as in Eq. 3. Depending on the base algorithm, we may choose to additionally learn a parameterized policy π_θ that shares encoder parameters with Q_θ and selects actions $\mathbf{a}_t \sim \pi_\theta(\cdot|f_\theta(\mathbf{s}_t))$.

To circumvent erroneous bootstrapping from augmented data (as discussed in Section III), we strictly apply data augmentation in Q-value estimation of the *current* state $\mathbf{s}_t$, *without* applying data augmentation to the successor state $\mathbf{s}_{t+1}$ used in Eq. 2 for bootstrapping with Q^{tgt}_ψ (and π_θ if applicable), which addresses Pitfall 1. If π_θ is learned (i.e. SVEA is implemented with an actor-critic algorithm), we also optimize it strictly from unaugmented data. To mitigate over-regularization of f_θ and Q_θ (Pitfall 2), we further employ a novel Q-objective which is described in the following.

B. Learning Objective

Our method redefines the temporal difference objective from Eq. 2 to better leverage data augmentation. First, recall that $q^{\text{tgt}}_t = r(\mathbf{s}_t, \mathbf{a}_t) + \gamma \max_{\mathbf{a}'_t} Q^{\text{tgt}}_\psi(f^{\text{tgt}}_\psi(\mathbf{s}_{t+1}), \mathbf{a}')$. Instead of learning to predict q^{tgt}_t only from state $\mathbf{s}_t$, we propose to minimize a linear combination of $\mathcal{L}_Q$ over two individual data streams, $\mathbf{s}_t$ and $\mathbf{s}^{\text{aug}}_t = \tau(\mathbf{s}_t, \nu),\ \nu \sim \mathcal{V}$, which we define as

$$\mathcal{L}^{\textbf{SVEA}}_Q(\theta, \psi) \triangleq \alpha\mathcal{L}_Q\left(\mathbf{s}_t, q^{\text{tgt}}_t\right) + \beta\mathcal{L}_Q\left(\mathbf{s}^{\text{aug}}_t, q^{\text{tgt}}_t\right) \tag{4}$$

$$= \mathbb{E}_{\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1} \sim \mathcal{B}}\Big[\alpha \left\| Q_\theta(f_\theta(\mathbf{s}_t), \mathbf{a}_t) - q^{\text{tgt}}_t \right\|^2_2 \tag{5}$$

$$+ \beta \left\| Q_\theta(f_\theta(\mathbf{s}^{\text{aug}}_t), \mathbf{a}_t) - q^{\text{tgt}}_t \right\|^2_2\Big], \tag{6}$$

where α, β are constant coefficients that balance the ratio of the unaugmented and augmented data streams, respectively, and q^{tgt}_t is computed strictly from unaugmented data.

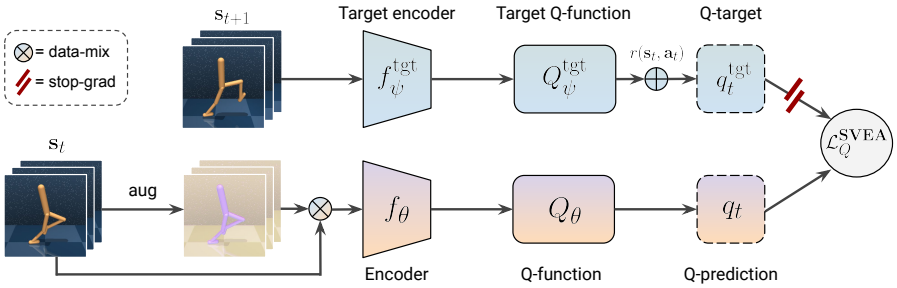

Fig. 1. **Overview.** An observation $\mathbf{s}_t$ is transformed by data augmentation $\tau(\cdot, \nu),\ \nu \sim \mathcal{V}$ to produce a view $\mathbf{s}_t^{\text{aug}}$. The Q-function Q_θ is then jointly optimized on both augmented and unaugmented data wrt the objective in Eq. 8, with the Q-target of the Bellman equation computed from an unaugmented observation $\mathbf{s}_{t+1}$. We illustrate our data-mixing strategy by the $\otimes$ operator.

$\mathcal{L}_Q^{\text{SVEA}}(\theta, \psi)$ serves as a *data-mixing* strategy that oversamples unaugmented data as an implicit variance reduction technique. As we will verify empirically in Section V, data-mixing is a simple and effective technique for variance reduction that works well in tandem with our proposed modifications to bootstrapping. For $\alpha = \beta$, the objective in Eq. 5 can be evaluated in a single, batched forward-pass by rewriting it as:

$$\mathbf{g}_t = \left[\mathbf{s}_t, \tau(\mathbf{s}_t, \nu)\right]_{\text{N}}, \quad h_t = \left[q_t^{\text{tgt}}, q_t^{\text{tgt}}\right]_{\text{N}}, \tag{7}$$

$$\mathcal{L}_Q^{\text{SVEA}}(\theta, \psi) = \mathbb{E}_{\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1} \sim \mathcal{B},\ \nu \sim \mathcal{V}} \tag{8}$$

$$\left[(\alpha + \beta) \left\| Q_\theta(f_\theta(\mathbf{g}_t), \mathbf{a}_t) - h_t \right\|_2^2\right], \tag{9}$$

where $[\cdot]_{\text{N}}$ is a concatenation operator along the batch dimension N for $\mathbf{s}_t, \mathbf{s}_t^{\text{aug}} \in \mathbb{R}^{N \times C \times H \times W}$ and $q_t^{\text{tgt}} \in \mathbb{R}^{N \times 1}$, which is illustrated as $\otimes$ in Figure 1. Empirically, we find $\alpha = 0.5, \beta = 0.5$ to be both effective and practical to implement. If the base algorithm learns a policy π_θ, its objective $\mathcal{L}_\pi(\theta)$ is optimized solely on unaugmented states $\mathbf{s}_t$ without changes to the objective, and a `stop-grad` operation is applied after f_θ to prevent non-stationary gradients of $\mathcal{L}_\pi(\theta)$ from interfering with Q-value estimation, i.e. only the objective from Eq. 5 or optionally Eq. 8 updates f_θ using SGD. As described in Section IV-A, parameters ψ are updated using an exponential moving average of θ and a `stop-grad` operation is therefore similarly applied after Q_ψ^{tgt}.

V. Experiments

We evaluate both sample efficiency, asymptotic performance, and generalization of our method and a set of strong baselines in tasks from DeepMind Control Suite (DMControl) [30] as well as a set of robotic manipulation tasks. DMControl offers challenging and diverse continuous control tasks and is widely used as a benchmark for image-based RL [9, 10, 33, 25, 14, 13]. To evaluate generalization of our method and baselines, we test methods under challenging distribution shifts (as illustrated in Figure 2) from the DMControl Generalization Benchmark (DMControl-GB) [11], the Distracting Control Suite (DistractingCS) [26], as well as distribution shifts unique to the robotic manipulation environment. Code is available at https://github.com/nicklashansen/dmcontrol-generalization-benchmark.

Setup. We implement our method and baselines using SAC [8] as base algorithm, and we apply random shift augmentation

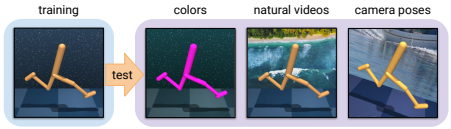

Fig. 2. **Experimental setup.** Agents are trained in a fixed environment and are expected to generalize to unseen environments with random colors, backgrounds, and camera poses.

to all methods by default, which makes our base algorithm equivalent to DrQ [13]. Architecture and hyperparameters are adopted from [11]. In DMControl experiments, all methods are evaluated on the full set of tasks from DMControl-GB.

Baselines and data augmentations. We benchmark our method against the following strong baselines: (1) **CURL** [25], a contrastive learning method for RL; (2) **RAD** that applies a random crop; (3) **DrQ** that applies a random shift; (4) **PAD** [12] that adapts to test environments using self-supervision; and (5) **SODA** [11] that applies data augmentation in auxiliary learning. We experiment with a diverse set of data augmentations proposed in previous work on RL and computer vision, namely random *shift* [13], random convolution (denoted *conv*) [15], random *overlay* [11], random *cutout* [3], Gaussian *blur*, random *affine-jitter*, and random *rotation* [14, 7].

A. Stability and Generalization on DMControl

We evaluate sample efficiency, asymptotic performance, and generalization of SVEA, DrQ, and a set of ablations across all 5 tasks from DMControl-GB. Figure 3 shows the stability of SVEA and DrQ under 6 common data augmentations. While the sample efficiency of DrQ degrades substantially for most augmentations, SVEA is relatively unaffected by the choice of data augmentation and improves sample efficiency *in all 18 instances*. To benchmark the generalization ability of SVEA, we compare its test performance to 5 recent state-of-the-art methods for image-based RL on the challenging `color_hard` and `video_easy` benchmarks from DMControl-GB, and report results in Table I. All methods use the same architecture and hyperparameters whenever applicable, and we here use *conv* and *overlay* augmentations for fair comparison to SODA. SVEA outperforms all methods considered in **9** out of **10** instances, and at a significantly lower computational cost than CURL, PAD, and SODA that learn auxiliary tasks.

We exclude generalization results on DistractingCS due to space constraints, but emphasize that SVEA improves gener-

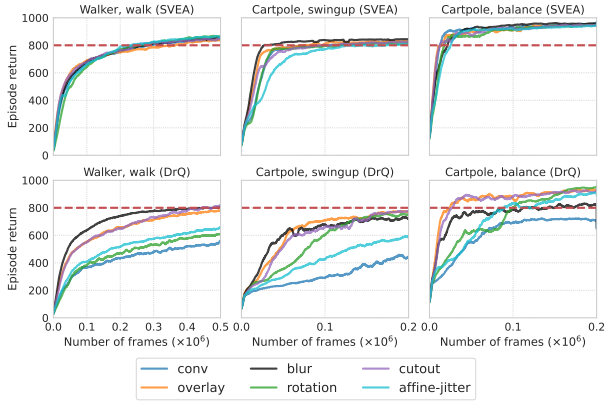

Fig. 3. **Data augmentations.** Training performance of SVEA (top) and DrQ (bottom) under 6 common data augmentations. Mean of 5 seeds. Red line is for visual guidance only.

TABLE I. **Comparison to state-of-the-art.** Test performance (episode return) of methods trained in a fixed environment and evaluated on: *(top)* randomized colors; and *(bottom)* natural video backgrounds as visual distraction. Results for CURL, RAD, PAD, and SODA are obtained from [11] and we report mean and ±1 std. of 5 seeds.

DMControl-GB (random colors)	CURL	RAD	DrQ	PAD	SODA (conv)	SODA (overlay)	**SVEA** (conv)	**SVEA** (overlay)
walker, walk	445 ±99	400 ±61	520 ±91	468 ±47	697 ±66	692 ±68	**760 ±145**	749 ±61
walker, stand	662 ±54	644 ±88	770 ±71	797 ±46	930 ±12	893 ±12	**942 ±26**	933 ±24
cartpole, swingup	454 ±110	590 ±53	586 ±52	630 ±63	831 ±21	805 ±28	**837 ±23**	832 ±23
ball_in_cup, catch	231 ±92	541 ±29	365 ±210	563 ±50	892 ±37	949 ±19	**961 ±7**	959 ±5
finger, spin	691 ±12	667 ±154	776 ±134	803 ±72	901 ±51	793 ±128	**977 ±5**	972 ±6
DMControl-GB (natural videos)	**CURL**	**RAD**	**DrQ**	**PAD**	**SODA (conv)**	**SODA (overlay)**	**SVEA (conv)**	**SVEA (overlay)**
walker, walk	556 ±133	606 ±63	682 ±89	717 ±79	635 ±48	768 ±38	612 ±144	**819 ±71**
walker, stand	852 ±75	745 ±146	873 ±83	935 ±20	903 ±56	955 ±13	795 ±70	**961 ±8**
cartpole, swingup	404 ±67	373 ±72	485 ±105	521 ±76	474 ±143	758 ±62	606 ±85	**782 ±27**
ball_in_cup, catch	316 ±119	481 ±26	318 ±157	436 ±55	539 ±111	**875 ±56**	659 ±110	871 ±106
finger, spin	502 ±19	400 ±64	533 ±119	691 ±80	363 ±185	695 ±97	764 ±86	**808 ±33**

alization by **42**% over DrQ at low randomization intensities, and degrades significantly slower than DrQ for high intensities (aggregated across 5 seeds for each of the 5 tasks from DMControl-GB).

B. RL with Vision Transformers

Vision Transformers (ViT) [4] have recently achieved impressive results on downstream tasks in computer vision. We replace all convolutional layers from the previous experiments with a 4-layer ViT encoder that operates on raw pixels in 8×8 space-time patches, and evaluate our method using data augmentation in conjunction with ViT encoders. Results are shown in Figure 4. We are, to the best of our knowledge, the first to successfully solve image-based RL tasks without CNNs. We observe that DrQ overfits significantly to the training environment compared to its CNN counterpart. SVEA achieves comparable sample efficiency and improves generalization by **706**% and **233**% on *Walker, walk* and *Cartpole, swingup*, respectively, over DrQ, while DrQ + conv remains unstable. SVEA might therefore be a promising technique for future RL studies with CNN-free architectures, where data augmentation appears to be especially important. We exclude additional ViT experiments due to space constraints but emphasize that other tasks from DMControl and robotic manipulation yield similar findings.

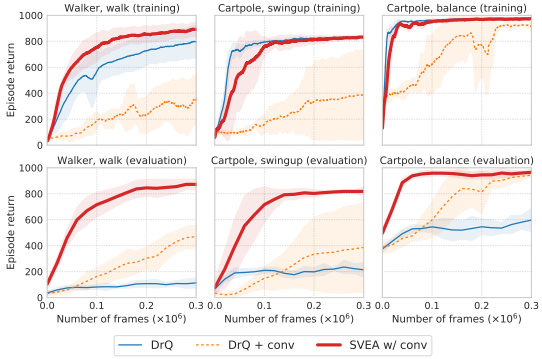

Fig. 4. **RL with Vision Transformers.** Train and test performance (`color_hard`) of SVEA and DrQ using ViT encoders. Mean of 5 seeds, shaded area is ±1 std. SVEA is stable under augmentation and dramatically improves generalization.

TABLE II. **Robotic manipulation.** Task success rate in 25 different test environments with randomized camera pose, colors, lighting, and background. Mean of 5 seeds.

Robotic manipulation	reach (train)	reach (test)	mv.trg. (train)	mv.trg. (test)	push (train)	push (test)
DrQ	**1.00**	0.60	**1.00**	0.69	**0.76**	0.26
DrQ + conv	0.59	0.77	0.60	0.89	0.13	0.12
SVEA w/ conv	**1.00**	**0.89**	**1.00**	**0.96**	0.72	**0.48**

C. Robotic Manipulation

We additionally consider a set of goal-conditioned robotic manipulation tasks using a simulated Kinova Gen3 arm: (i) *reach*, a task in which the robot needs to position its gripper above a goal indicated by a red mark; (ii) *reach moving target*, a task similar to (i) but where the robot needs to follow a red mark moving continuously in a zig-zag pattern at a random velocity; and (iii) *push*, a task in which the robot needs to push a cube to a red mark. We evaluate success rate during training and measure generalization to 25 different variations of the environment; results are shown in Table II. SVEA trained with *conv* augmentation exhibits similar stability and sample efficiency as DrQ trained without, while DrQ + conv has poor sample efficiency and fails to solve the challenging *push* task. SVEA outperforms both baselines in terms of generalization.

Conclusion. SVEA is found to greatly improve both stability and sample efficiency under augmentation, while achieving competitive generalization results. Our experiments indicate that our method scales to ViT-based architectures, and it may therefore be a promising technique for large-scale RL experiments where data augmentation is expected to play an increasingly important role.

References

[1] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemyslaw Debiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Christopher Hesse, R. Józefowicz, Scott Gray, Catherine Olsson, Jakub W. Pachocki, Michael Petrov, Henrique Pondé de Oliveira Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, J. Schneider, S. Sidor, Ilya Sutskever, Jie Tang, F. Wolski, and Susan Zhang. Dota 2 with large scale deep reinforcement learning. *ArXiv*, abs/1912.06680, 2019. 1

[2] K. Cobbe, Oleg Klimov, Christopher Hesse, Taehoon Kim, and John Schulman. Quantifying generalization in reinforcement learning. In *ICML*, 2019. 1

[3] Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. Quantifying generalization in reinforcement learning, 2018. 3

[4] A. Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, M. Dehghani, Matthias Minderer, G. Heigold, S. Gelly, Jakob Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2020. 4

[5] Jesse Farebrother, Marlos C. Machado, and Michael H. Bowling. Generalization and regularization in dqn. *ArXiv*, abs/1810.00123, 2018. 1

[6] Scott Fujimoto, H. V. Hoof, and D. Meger. Addressing function approximation error in actor-critic methods. *ArXiv*, abs/1802.09477, 2018. 1, 2

[7] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations, 2018. 3

[8] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018. 1, 2, 3

[9] Danijar Hafner, T. Lillicrap, Ian S. Fischer, R. Villegas, David R Ha, H. Lee, and James Davidson. Learning latent dynamics for planning from pixels. *ArXiv*, abs/1811.04551, 2019. 3

[10] Danijar Hafner, T. Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *ArXiv*, abs/1912.01603, 2020. 3

[11] Nicklas Hansen and Xiaolong Wang. Generalization in reinforcement learning by soft data augmentation. In *International Conference on Robotics and Automation*, 2021. 1, 3, 4

[12] Nicklas Hansen, Yu Sun, Pieter Abbeel, Alexei A. Efros, Lerrel Pinto, and Xiaolong Wang. Self-supervised policy adaptation during deployment. 2020. 3

[13] Ilya Kostrikov, Denis Yarats, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. 2020. 1, 2, 3

[14] Michael Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. *arXiv preprint arXiv:2004.14990*, 2020. 1, 2, 3

[15] Kimin Lee, Kibok Lee, Jinwoo Shin, and Honglak Lee. A simple randomization technique for generalization in deep reinforcement learning. *ArXiv*, abs/1910.05396, 2019. 3

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

[17] T. Lillicrap, J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *CoRR*, abs/1509.02971, 2016. 1, 2

[18] L. J. Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, 8:293–321, 2004. 2

[19] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013. 1, 2

[20] Lerrel Pinto, Marcin Andrychowicz, Peter Welinder, Wojciech Zaremba, and Pieter Abbeel. Asymmetric actor critic for image-based robot learning. *arXiv preprint arXiv:1710.06542*, 2017. 1

[21] Roberta Raileanu, M. Goldstein, Denis Yarats, Ilya Kostrikov, and R. Fergus. Automatic data augmentation for generalization in deep reinforcement learning. *ArXiv*, abs/2006.12862, 2020. 1

[22] Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. *arXiv preprint arXiv:2007.05929*, 2020. 2

[23] Connor Shorten and T. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6:1–48, 2019. 1

[24] Xingyou Song, Yiding Jiang, Stephen Tu, Yilun Du, and Behnam Neyshabur. Observational overfitting in reinforcement learning. *ArXiv*, abs/1912.02975, 2020. 1

[25] Aravind Srinivas, Michael Laskin, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. *arXiv preprint arXiv:2004.04136*, 2020. 3

[26] Austin Stone, Oscar Ramirez, K. Konolige, and Rico Jonschkowski. The distracting control suite - a challenging benchmark for reinforcement learning from pixels. *ArXiv*, abs/2101.02722, 2021. 1, 3

[27] Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling representation learning from reinforcement learning. arXiv:2004.14990. 2

[28] R. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9–44, 2005. 1

[29] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018. 2

[30] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. DeepMind control suite. Technical report, DeepMind, January 2018. 1, 3

[31] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep 2017. 1

[32] Oriol Vinyals, I. Babuschkin, Wojciech Czarnecki, Michaël Mathieu, Andrew Dudzik, J. Chung, D. Choi, Richard Powell, Timo Ewalds, P. Georgiev, Junhyuk Oh, Dan Horgan, M. Kroiss, Ivo Danihelka, Aja Huang, L. Sifre, Trevor Cai, J. Agapiou, Max Jaderberg, A. Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, D. Budden, Yury Sulsky, James Molloy, T. Paine, Caglar Gulcehre, Ziyu Wang, T. Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, T. Lillicrap, K. Kavukcuoglu, D. Hassabis, C. Apps, and D. Silver. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, pages 1–5, 2019. 1

[33] Denis Yarats, Amy Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and Rob Fergus. Improving sample efficiency in model-free reinforcement learning from images, 2019. 3

[34] C. Zhang, Oriol Vinyals, R. Munos, and S. Bengio. A study on overfitting in deep reinforcement learning. *ArXiv*, abs/1804.06893, 2018. 1

[35] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement

learning. In *ICRA*, pages 3357–3364. IEEE, 2017. 1