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

Nicklas Hansen[1] Hao Su[1] Xiaolong Wang[1]

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 of image-based RL 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.[1]

1. Introduction

Reinforcement Learning (RL) from visual observations has achieved tremendous success in various applications such as video-games (Mnih et al., 2013; Berner et al., 2019; Vinyals et al., 2019), robotic manipulation (Levine et al., 2016), and autonomous navigation (Mirowski et al., 2017; Zhu et al., 2017). However, it is still very challenging for current methods to generalize the learned skills to novel environments, and policies trained by RL can easily overfit to the training environment (Zhang et al., 2018; Farebrother et al., 2018), especially for high-dimensional observation spaces, e.g. images (Cobbe et al., 2019; Song et al., 2020).

Increasing variability in training data via domain randomization (Tobin et al., 2017; Pinto et al., 2017) and data augmentation (Shorten & Khoshgoftaar, 2019; Laskin et al., 2020; Kostrikov et al., 2020; Raileanu et al., 2020) has demonstrated encouraging results for learning policies invariant to environment changes. Specifically, recent works on data augmentation (Laskin et al., 2020; Kostrikov et al., 2020; Hansen & Wang, 2021) 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, 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 applying data augmentation to common off-policy RL algorithms (Mnih et al., 2013; Lillicrap et al., 2016; Fujimoto et al., 2018; Haarnoja et al., 2018). Based on our findings, we provide an intuitive method for stabilizing this class of algorithms under use of strong data augmentation. 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 indi-

[1] All authors affiliated with UC San Diego. Project page: https://nicklashansen.github.io/SVEA.

rectly through parameter-sharing. Our framework can be implemented efficiently without additional forward passes nor introducing additional learnable parameters.

We perform extensive empirical evaluation on the DeepMind Control Suite (Tassa et al., 2018) and extensions of it, including the DMControl Generalization Benchmark (Hansen & Wang, 2021) and the Distracting Control Suite (Stone et al., 2021), as well as a set of robotic manipulation tasks. Our method successfully stabilizes Q-value estimation with ConvNets 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. Finally, we show that our method scales to RL with Vision Transformers (ViT) (Dosovitskiy et al., 2020). We find that ViT-based architectures are especially prone to overfitting, and data augmentation may therefore be a key component for large-scale RL.

2. Related Work

Representation Learning. Learning visual invariances using data augmentation and self-supervised objectives has proven highly successful in computer vision (Pathak et al., 2016; Noroozi & Favaro, 2016; Zhang et al., 2016; Wu et al., 2018; van den Oord et al., 2018; Tian et al., 2019; Xu et al., 2020; He et al., 2020; Chen et al., 2020). For example, Chen et al. (Chen et al., 2020) perform an extensive study on data augmentation (e.g. random cropping and image distortions) for contrastive learning, and show that representations pre-trained with such transformations transfer effectively to downstream tasks. While our work also uses data augmentation for learning visual invariances, we leverage the Q-objective of deep Q-learning algorithms instead of auxiliary representation learning tasks.

Visual Learning for RL. Numerous methods have been proposed with the goal of improving sample efficiency (Jaderberg et al., 2016; Shelhamer et al., 2017; van den Oord et al., 2018; Yarats et al., 2019; Lin et al., 2019; Srinivas et al., 2020; Stooke et al.; Schwarzer et al., 2020; Yarats et al., 2021) of image-based RL. Recently, using self-supervision to improve generalization in RL has also gained interest (Zhang et al., 2020; Pathak et al., 2019; Sekar et al., 2020; Agarwal et al., 2021; Hansen et al., 2020; Hansen & Wang, 2021; Wang et al., 2021). Notably, Zhang et al. (Zhang et al., 2020) and Agarwal et al. (Agarwal et al., 2021) propose to learn behavioral similarity embeddings via auxiliary tasks (bisimulation metrics and contrastive learning, respectively), and Hansen et al. (Hansen & Wang, 2021) learn visual invariances through an auxiliary prediction task. While these results are encouraging, it has also been shown in (Jaderberg et al., 2016; Lin et al., 2019; Hansen et al., 2020; Yu et al., 2020; Lyle et al., 2021) that the best choice of auxiliary tasks depends on the particular RL task, and that joint optimization with sub-optimally chosen tasks can lead to gradient interference. We achieve competitive sample-efficiency and generalization results *without* the need for carefully chosen auxiliary tasks, and our method is therefore applicable to a larger variety of RL tasks.

Data Augmentation and Randomization for RL. Our work is directly inspired by previous work on generalization in RL by domain randomization (Tobin et al., 2017; Pinto et al., 2017; Peng et al., 2018; Ramos et al., 2019; Chebotar et al., 2019) and data augmentation (Lee et al., 2019; Cobbe et al., 2018; Wang et al., 2020; Laskin et al., 2020; Kostrikov et al., 2020; Raileanu et al., 2020; Stooke et al.; Hansen & Wang, 2021). For example, Tobin et al. (Tobin et al., 2017) show that a neural network trained for object localization in a simulation with randomized visual augmentations improves real world generalization. Similarly, Lee et al.(Lee et al., 2019) show that application of a random convolutional layer to observations during training improve generalization in 3D navigation tasks. More recently, extensive studies on data augmentation (Laskin et al., 2020; Kostrikov et al., 2020) have been conducted with RL, and conclude that, while small random crops and translations can improve sample efficiency, most data augmentations *decrease* sample efficiency and cause divergence. We illuminate main causes of instability, and propose a framework for data augmentation in deep Q-learning algorithms that drastically improves stability and generalization.

Improving Deep Q-Learning. While deep Q-learning algorithms such as Deep Q-Networks (DQN) (Mnih et al., 2013) have achieved impressive results in image-based RL, the temporal difference objective is known to have inherent instabilities when used in conjunction with function approximation and off-policy data (Sutton & Barto, 2018). Therefore, a variety of algorithmic improvements have been proposed to improve convergence (Hasselt et al., 2016b; Wang et al., 2016; Hausknecht & Stone, 2015; Hasselt et al., 2016a; Schaul et al., 2016; Lillicrap et al., 2016; Fujimoto et al., 2018; Fortunato et al., 2018; Hessel et al., 2018). For example, Hasselt et al. (Hasselt et al., 2016b) reduce overestimation of Q-values by decomposing the target Q-value estimation into action selection and action evaluation using separate networks. Lillicrap et al. (Lillicrap et al., 2016) reduce target variance by defining the target Q-network as a slow-moving average of the online Q-network. Our method also improves Q-value estimation, but we specifically address the instability of deep Q-learning algorithms on augmented data.

3. Preliminaries

Problem formulation. We formulate the interaction between environment and policy as a Markov Decision Process (MDP) (Bellman, 1957) $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma \rangle$, where

$\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P}\colon \mathcal{S} \times \mathcal{A} \mapsto \mathcal{S}$ is the state transition function that defines a conditional probability distribution $\mathcal{P}\left(\cdot|\mathbf{s}_t, \mathbf{a}_t\right)$ over all possible next states given a state $\mathbf{s}_t \in \mathcal{S}$ and action $\mathbf{a}_t \in \mathcal{A}$ taken at time t, $r\colon \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ is a reward function, and $\gamma \in [0, 1)$ is the discount factor. Because image observations only offer partial state observability (Kaelbling et al., 1998), we define a state $\mathbf{s}_t$ as a sequence of $k+1$ consecutive frames $(\mathbf{o}_t, \mathbf{o}_{t-1}, \dots, \mathbf{o}_{t-k})$, $\mathbf{o} \in \mathcal{O}$, where $\mathcal{O}$ is the high-dimensional image space, as is common practice (Mnih et al., 2013). The goal is then to learn a policy $\pi\colon \mathcal{S} \mapsto \mathcal{A}$ that maximizes discounted return $R_t = \mathbb{E}_{\Gamma\sim\pi}[\sum_{t=1}^{T} \gamma^t r(\mathbf{s}_t, \mathbf{a}_t)]$ along a trajectory $\Gamma = (\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_T)$ obtained by following policy π from an initial state $\mathbf{s}_0 \in \mathcal{S}$ to a state $\mathbf{s}_T$ with state transitions sampled from $\mathcal{P}$, and π is parameterized by a collection of learnable parameters θ. For clarity, we hereon generically denote parameterization with subscript, e.g. π_θ. We further aim to learn parameters θ s.t. π_θ generalizes well (i.e. obtains high discounted return) to novel MDPs, which is generally unfeasible without further assumptions about the structure of the space of MDPs. In this work, we therefore consider generalization to MDPs $\overline{\mathcal{M}} = \langle \overline{\mathcal{S}}, \mathcal{A}, \mathcal{P}, r, \gamma \rangle$, where states $\overline{\mathbf{s}}_t \in \overline{\mathcal{S}}$ are constructed from observations $\overline{\mathbf{o}}_t \in \overline{\mathcal{O}}$, $\mathcal{O} \subseteq \overline{\mathcal{O}}$ of a novel observation space $\overline{\mathcal{O}}$, and $\overline{\mathcal{M}} \sim \mathbb{M}$ for a space of MDPs $\mathbb{M}$.

Deep Q-Learning. Common model-free off-policy RL algorithms aim to estimate an optimal state-action value function $Q^*\colon \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ 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 $\left(r(\mathbf{s}_t, \mathbf{a}_t) + \gamma \max_{\mathbf{a}'_t} Q^{\text{tgt}}_\psi(\mathbf{s}_{t+1}, \mathbf{a}'_t)\right) - Q_\theta(\mathbf{s}_t, \mathbf{a}_t)$ (Sutton, 2005), 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

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

$$q^{\text{tgt}} = r(\mathbf{s}_t, \mathbf{a}_t) + \gamma \max_{\mathbf{a}'_t} Q^{\text{tgt}}_\psi(\mathbf{s}_{t+1}, \mathbf{a}'_t) \,, \quad (2)$$

where $\mathcal{B}$ is a replay buffer with transitions collected by a behavioral policy (Lin, 2004). From here, we can derive a greedy policy directly by selecting actions $\mathbf{a}_t = \arg\max_{\mathbf{a}_t} Q_\theta(\mathbf{s}_t, \mathbf{a}_t)$. While $Q^{\text{tgt}} = Q$ and periodically setting $\psi \longleftarrow \theta$ exactly recovers the objective of DQN (Mnih et al., 2013), several improvements have been proposed to improve stability of Eq. 1, such as Double Q-learning (Hasselt et al., 2016b), Dueling Q-networks (Wang et al., 2016), updating target parameters using a slow-moving average of the online Q-network (Lillicrap et al., 2016):

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

for an iteration step n and a momentum coefficient $\zeta \in (0, 1]$, and others (Hausknecht & Stone, 2015; Hasselt et al., 2016a; Schaul et al., 2016; Fortunato et al., 2018; Hessel et al., 2018). As computing $\max_{\mathbf{a}'_t} Q^{\text{tgt}}_\psi(\mathbf{s}_{t+1}, \mathbf{a}'_t)$ in Eq. 1 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 (Lillicrap et al., 2016; Fujimoto et al., 2018; Haarnoja et al., 2018).

Soft Actor-Critic (SAC) (Haarnoja et al., 2018) is an off-policy actor-critic algorithm that learns a state-action value function Q_θ and a stochastic policy π_θ (and optionally a temperature parameter), where Q_θ is optimized using a variant of the objective in Eq. 1 and π_θ is optimized using a γ-discounted maximum-entropy objective (Ziebart et al., 2008). To improve stability, SAC is also commonly implemented using Double Q-learning and the slow-moving target parameters from Eq. 3. We will in the remainder of this work describe our method in the context of a generic off-policy RL algorithm that learns a parameterized state-action value function Q_θ, while we in our experiments discussed in Section 6 evaluate of our method using SAC.

4. Pitfalls of Data Augmentation in Deep Q-Learning

In this section, we aim to illuminate the main causes of instability from naïve application of data augmentation in Q-value estimation. Our goal is to learn a Q-function Q_θ for an MDP $\mathcal{M}$ that generalizes to novel MDPs $\overline{\mathcal{M}} \sim \mathbb{M}$, and we leverage data augmentation as an optimality-invariant state transformation τ to induce a bisimulation relation (Larsen & Skou, 1989; Givan et al., 2003) 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 (Kostrikov et al., 2020))**.** *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 $Q(\mathbf{s}, \mathbf{a}) = Q(\tau(\mathbf{s}, \nu), \mathbf{a}) \;\; \forall\; \mathbf{s} \in \mathcal{S},\; \mathbf{a} \in \mathcal{A},\; \nu \in \mathcal{V}$, where $\nu \in \mathcal{V}$ parameterizes τ.*

Following our definitions of $\mathcal{M}, \overline{\mathcal{M}}$ from Section 3, we can further extend the concept of optimality-invariant transformations to MDPs, noting that a change of state space itself can be described as a transformation $\overline{\tau}\colon \mathcal{S} \times \overline{\mathcal{V}} \mapsto \overline{\mathcal{S}}$ with unknown parameters $\overline{\nu} \in \overline{\mathcal{V}}$. 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 (Laskin et al., 2020; Kostrikov et al., 2020; Stooke et al.; Schwarzer et al., 2020) may potentially improve generalization, it can be harmful to Q-value estimation. We hypothesize that this

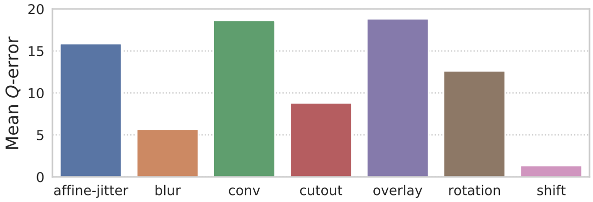

Figure 1. **Mean difference in Q-value estimation on augmented vs. non-augmented data.** We measure the mean absolute error in Q-value estimation from converged DrQ (Kostrikov et al., 2020) agents (trained with *shift* augmentation) on the same observations before and after augmentation. Averaged across 5 seeds of DrQ for each of the 5 tasks from DMControl-GB.

is primarily because it dramatically increases the size of the observed state space, and consequently also increases variance $\mathrm{Var}\left[Q(\tau(\mathbf{s},\nu))\right] \geq \mathrm{Var}\left[Q(\mathbf{s})\right],\ \nu \sim \mathcal{V}$ 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 (Laskin et al., 2020; Kostrikov et al., 2020; Stooke et al.; Schwarzer et al., 2020) applies augmentation to both state $\mathbf{s}_t^{\text{aug}} \triangleq \tau(\mathbf{s}_t, \nu)$ and successor state $\mathbf{s}_{t+1}^{\text{aug}} \triangleq \tau(\mathbf{s}_{t+1}, \nu')$ where $\nu, \nu' \sim \mathcal{V}$. Compared with DQN (Mnih et al., 2013) that uses a deterministic (more precisely, periodically updated) Q-target, this practice introduces a non-deterministic Q-target $r(\mathbf{s}_t, \mathbf{a}_t) + \gamma \max_{\mathbf{a}_t'} Q_\psi^{\text{tgt}}(\mathbf{s}_{t+1}^{\text{aug}}, \mathbf{a}_t')$ 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 (Sutton & Barto, 2018). This motivates the work to introduce a slowly changing target network, and several other works have refined the Q-target update rule (Lillicrap et al., 2016; Fujimoto et al., 2018) to further reduce volatility. However, 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 (Lillicrap et al., 2016) and SAC (Haarnoja et al., 2018), 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. 1 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 by data augmentation. Figure 1 shows the mean difference in Q-predictions made with augmented vs. unaugmented data in fully converged DrQ (Kostrikov et al., 2020) agents trained with *shift* augmentation. Augmentations such as affine-jitter, random convolution, and random overlay incur large differences in estimated Q-values. While such difference can be reduced by regularizing the optimization with each individual augmentation, we emphasize that even the minimal shift augmentation used throughout training incurs non-negligible difference. Since ψ is commonly chosen to be a moving average of θ as in Eq. 3, such differences caused by over-regularization affect Q_θ and Q_ψ^{tgt} equally, and optimization may therefore still diverge depending on the choice of data augmentation. As such, there is an inherent trade-off between accurate Q-value estimation and generalization when using data augmentation. In the following section, we address these pitfalls.

5. Method

We propose **SVEA**: **S**tabilized Q-**V**alue **E**stimation under **A**ugmentation, a general framework for generalization by data augmentation in RL. SVEA applies data augmentation in a novel learning framework leveraging two data streams – with and without augmented data, respectively. Our method is compatible with any standard off-policy RL algorithm without changes to the underlying neural network that parameterizes the policy, and it requires no additional forward passes, auxiliary tasks, nor learnable parameters. While SVEA does not make any assumptions about the structure of states $\mathbf{s}_t \in \mathcal{S}$, we here describe our method in the context of image-based RL.

5.1. Architectural Overview

An overview of the SVEA architecture is provided in Figure 2. Our method leverages properties of common neural network architectures used in off-policy RL without introducing additional learnable parameters. We subdivide the neural network layers and corresponding learnable parameters of a state-action value function into subnetworks 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 similarly define the target state-action value function s.t. $q_t^{\text{tgt}} \triangleq r(\mathbf{s}_t, \mathbf{a}_t) + \gamma \max_{\mathbf{a}_t'} Q_\psi^{\text{tgt}}(f_\psi^{\text{tgt}}(\mathbf{s}_{t+1}), \mathbf{a}')$ is the target Q-value for $(\mathbf{s}_t, \mathbf{a}_t)$, and we define parameters ψ as an exponential moving average of θ as in Eq. 3. Depending on the choice of underlying 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

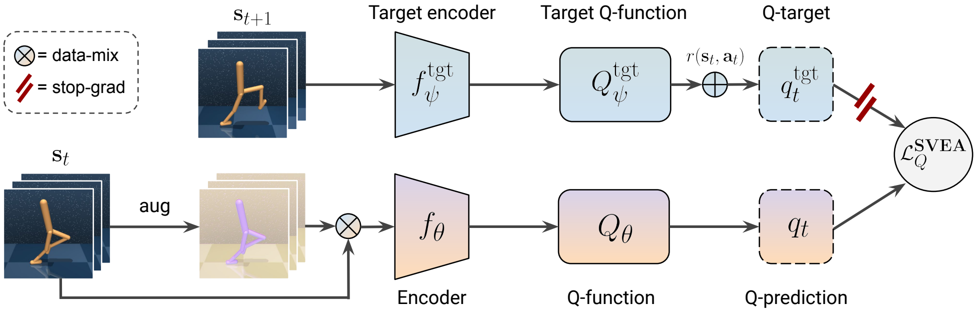

Figure 2. **Overview.** An observation $\mathbf{s}_t$ is transformed by data augmentation $\tau(\cdot, \nu),\ \nu \sim \mathcal{V}$ to produce a view $\mathbf{s}_t^{\text{aug}}$. The Q-function Q_θ is then jointly optimized on both augmented and unaugmented data wrt the objective in Eq. 8, with the Q-target of the Bellman equation computed from an unaugmented observation $\mathbf{s}_{t+1}$. We illustrate our data-mixing strategy by the ⊗ operator.

data (as discussed in Section 4), 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. 1 for bootstrapping with Q_ψ^{tgt} (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 in optimization of f_θ and Q_θ (Pitfall 2), we further employ a novel Q-objective that leverages both augmented and unaugmented data, which we introduce in the following section.

5.2. Learning Objective

Our method redefines the temporal difference objective from Eq. 1 to better leverage data augmentation. First, recall that $q_t^{\text{tgt}} = r(\mathbf{s}_t, \mathbf{a}_t) + \gamma \max_{\mathbf{a}'_t} Q_\psi^{\text{tgt}}(f_\psi^{\text{tgt}}(\mathbf{s}_{t+1}), \mathbf{a}')$. Instead of learning to predict q_t^{tgt} 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 $\mathbf{s}_t^{\text{aug}} = \tau(\mathbf{s}_t, \nu),\ \nu \sim \mathcal{V}$, which we define as the objective

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

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

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

where α, β are constant coefficients that balance the ratio of the unaugmented and augmented data streams, respectively, and q_t^{tgt} is computed strictly from unaugmented data. $\mathcal{L}_Q^{\textbf{SVEA}}(\theta, \psi)$ serves as a *data-mixing* strategy that oversamples unaugmented data as an implicit variance reduction technique. As we will verify empirically in Section 6, 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:

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

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

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

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 ⊗ in Figure 2. Empirically, we find $\alpha = 0.5, \beta = 0.5$ to be both effective and practical to implement. If the base algorithm of choice 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 stochastic gradient descent. As described in Section 5.1, parameters ψ are updated using an exponential moving average of θ and a `stop-grad` operation is therefore similarly applied after Q_ψ^{tgt}. We summarize our method for $\alpha = \beta$ applied to a generic off-policy algorithm in Algorithm 1.

6. 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) (Tassa et al., 2018) 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 (Hafner et al., 2019; 2020; Yarats et al., 2019; Srinivas et al., 2020; Laskin et al., 2020; Kostrikov et al., 2020). To evaluate generalization of our method and baselines, we test methods under challenging distribution shifts (as illustrated in Figure 3) from the DMControl Generalization Benchmark (DMControl-GB) (Hansen & Wang, 2021), the Distracting Control Suite (DistractingCS) (Stone et al., 2021), as well as distribution shifts unique to the robotic manipulation environment. Code is avail-

Algorithm 1 Generic **SVEA** off-policy algorithm (▶ naïve augmentation, ▶ our modifications)

```
$\theta, \theta_\pi, \psi$: randomly initialized network parameters, $\psi \longleftarrow \theta$        ▷ Initialize $\psi$ to be equal to $\theta$
$\eta, \zeta$: learning rate and momentum coefficient
$\alpha, \beta$: loss coefficients, default: $(\alpha = 0.5, \beta = 0.5)$
for timestep $t = 1...T$ do
act:
    $\mathbf{a}_t \sim \pi_\theta\left(\cdot | f_\theta(\mathbf{s}_t)\right)$        ▷ Sample action from policy
    $\mathbf{s}'_t \sim \mathcal{P}(\cdot | \mathbf{s}_t, \mathbf{a}_t)$        ▷ Sample transition from environment
    $\mathcal{B} \leftarrow \mathcal{B} \cup (\mathbf{s}_t, \mathbf{a}_t, r(\mathbf{s}_t, \mathbf{a}_t), \mathbf{s}'_t)$        ▷ Add transition to replay buffer
update:
    $\{\mathbf{s}_i, \mathbf{a}_i, r(\mathbf{s}_i, \mathbf{a}_i), \mathbf{s}'_i \mid i = 1...N\} \sim \mathcal{B}$        ▷ Sample batch of transitions
    $\mathbf{s}_i = \tau(\mathbf{s}_i, \nu_i),\ \mathbf{s}'_i = \tau(\mathbf{s}'_i, \nu'_i),\ \nu_i, \nu'_i \sim \mathcal{V}$        ▶ Naïve application of data augmentation
    for transition $i = 1..N$ do
        $\theta_\pi \longleftarrow \theta_\pi - \eta \nabla_{\theta_\pi} \mathcal{L}_\pi\left(\mathbf{s}_i; \theta_\pi\right)$ (if applicable)        ▷ Optimize $\pi_\theta$ with SGD
        $q^{\text{tgt}}_i = r(\mathbf{s}_i, \mathbf{a}_i) + \gamma \max_{\mathbf{a}'_i} Q^{\text{tgt}}_\psi(f^{\text{tgt}}_\psi(\mathbf{s}'_i), \mathbf{a}'_i)$        ▷ Compute $Q$-target
        $\mathbf{s}^{\text{aug}}_i = \tau(\mathbf{s}_i, \nu_i),\ \nu_i \sim \mathcal{V}$        ▶ Apply stochastic data augmentation
        $\mathbf{g}_i = \left[\mathbf{s}_i, \mathbf{s}^{\text{aug}}_i\right]_{\text{N}},\ h_i = \left[q^{\text{tgt}}_i, q^{\text{tgt}}_i\right]_{\text{N}}$        ▶ Pack data streams
        $\theta \longleftarrow \theta - \eta \nabla_\theta \mathcal{L}^{\textbf{SVEA}}_Q\left(\mathbf{g}_i, h_i; \theta, \psi\right)$        ▶ Optimize $f_\theta$ and $Q_\theta$ with SGD
        $\psi \longleftarrow (1 - \zeta)\psi + \zeta\theta$        ▷ Update $\psi$ using EMA of $\theta$
```

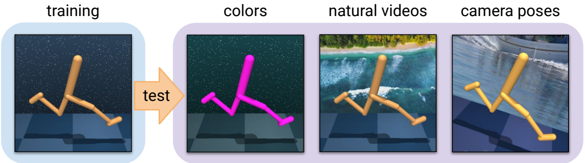

Figure 3. **Experimental setup.** Agents are trained in a fixed environment (the *training* environment) and are expected to generalize to novel environments with e.g. random colors, backgrounds, and camera poses.

able at `https://github.com/nicklashansen/dmcontrol-generalization-benchmark`.

Setup. We implement our method and baselines using SAC (Haarnoja et al., 2018) as base algorithm, and we apply random shift augmentation to all methods by default, which makes our base algorithm equivalent to DrQ (Kostrikov et al., 2020). Network architecture and hyperparameters are adopted from (Hansen & Wang, 2021), and observations are stacks of 3 RGB frames of size $84 \times 84 \times 3$. In our DMControl-GB and DistractingCS experiments, all methods are trained for 500k frames[2] and evaluated on the full set of tasks proposed in DMControl-GB. For simplicity, we adopt the same experimental setup for robotic manipulation.

Baselines and data augmentations. We benchmark our method against the following strong baselines: (1) **CURL** (Srinivas et al., 2020), a contrastive learning method for RL; (2) **RAD** that applies a random crop; (3) **DrQ** that applies a random shift; (4) **PAD** (Hansen et al., 2020) that adapts to test environments using self-supervision; (5) **SODA** (Hansen & Wang, 2021) that applies data augmentation in auxiliary learning; as well as a number of ablations. We experiment with a diverse set of data augmentations proposed in previous work on RL and computer vision, namely random *shift* (Kostrikov et al., 2020), random convolution (denoted *conv*) (Lee et al., 2019), random *overlay* (Hansen & Wang, 2021), random *cutout* (Cobbe et al., 2018), Gaussian *blur*, random *affine-jitter*, and random *rotation* (Laskin et al., 2020; Gidaris et al., 2018).

6.1. 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, and report training and test curves in Figure 4. SVEA *outperforms all baselines* on the test environment – often by a large margin – and maintains a sample efficiency comparable to DrQ trained without strong augmentation, while DrQ degrades substantially when using the additional *conv* augmentation. We examine the reason for SVEA's success with the following two ablations: a *data-mixing only* variant that applies our data-mixing strategy in both Q_θ, Q^{tgt}_ψ, and π_θ, as well as a variant of SVEA that only applies augmentation in Q_θ but does not apply data-mixing. We find that both components of SVEA are necessary to achieve both stability *and* the generalization benefits of strong data augmentation. We further evaluate the stability of SVEA and DrQ under 6 common data augmentations; results are shown in Figure 5. While the sample efficiency of DrQ degrades substantially for most augmentations, SVEA is relatively unaffected by the choice of data augmentation and improves sample effi-

[2] Note that some works using DMControl evaluate after a number of *simulation steps*, which is comparably less frames for a frame skip > 1. We follow previous work on generalization (Hansen et al., 2020; Hansen & Wang, 2021; Wang et al., 2021).

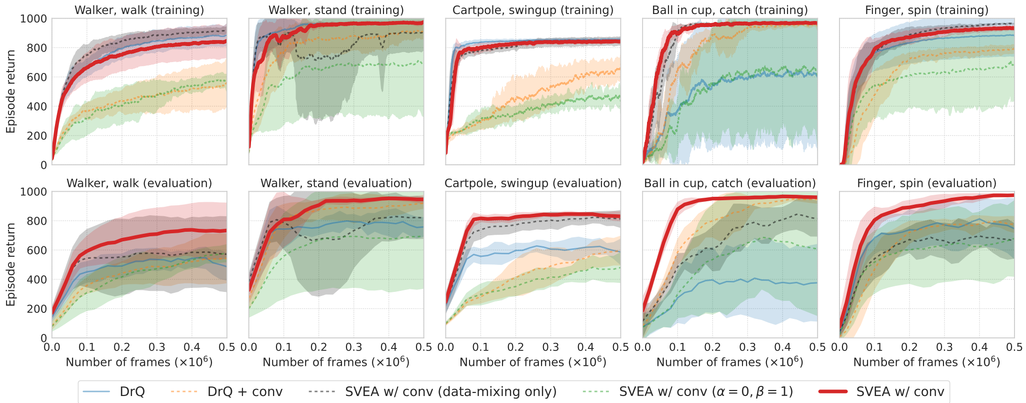

Figure 4. **Training and test performance.** We compare SVEA to DrQ with and without random convolution augmentation, as well as a set of ablations. *Data-mixing only* indiscriminately applies our data-mixing strategy to all data streams, and $(\alpha = 0, \beta = 1)$ only augments Q-predictions but without data-mixing. We find both components to contribute to SVEA's success. *Top:* episode return on the training environment during training. *Bottom:* generalization measured by episode return on the `color_hard` benchmark of DMControl-GB. Mean of 5 seeds, shaded area is ±1 std. deviation.

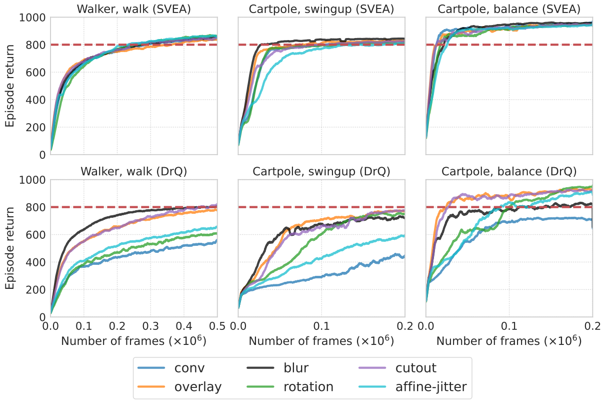

Figure 5. **Data augmentations.** Training performance of SVEA (top) and DrQ (bottom) under 6 common data augmentations. Mean of 5 seeds. Red line for visual guidance.

ciency *in all 18 instances*. See supplementary material for a per-augmentation comparison. Because we empirically find the *conv* augmentation to be particularly difficult to optimize, we provide additional stability experiments in Section 6.2 and 6.3 using this augmentation.

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 1. 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 further evaluate generalization on DistractingCS, an extremely challenging benchmark for generalization where camera pose, background, and colors are continually changing throughout an episode. Figure 6 (top) shows a comparison of SVEA with *conv* and *overlay* augmentations to DrQ over a range of gradually increasing intensity of distractions, averaged across all 5 tasks from DMControl-GB. SVEA improves generalization by **42**% at low intensity, and degrades significantly slower than DrQ for high intensities.

Table 1. **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 (Hansen & Wang, 2021) and we report mean and std. deviation of 5 seeds. SVEA achieves competitive results in all tasks.

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

6.2. RL with Vision Transformers

Vision Transformers (ViT) (Dosovitskiy et al., 2020) 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

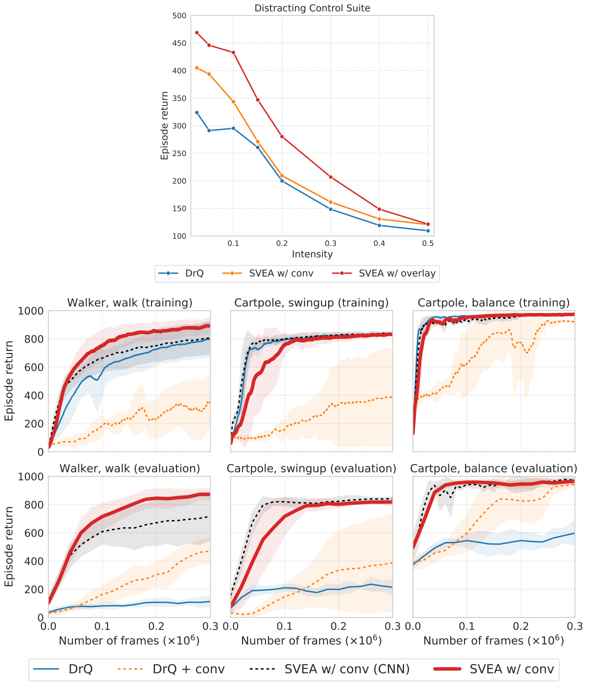

Figure 6. (Top) **DistractingCS.** Mean return as a function of randomization intensity, aggregated across 5 seeds and all 5 tasks from DMControl-GB. SVEA improves generalization at all intensities. *(Bottom)* **RL with Vision Transformers.** Training and test performance (`color_hard`) of SVEA and DrQ using ViT encoders. Mean of 5 seeds, shaded area is ±1 std. deviation. SVEA is stable under augmentation and dramatically improves generalization.

with ViT encoders. Results are shown in Figure 6 (bottom). 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 $\mathbf{706}\%$ and $\mathbf{233}\%$ on *Walker, walk* and *Cartpole, swingup*, respectively, over DrQ, while DrQ + conv remains unstable. Interestingly, we observe that our ViT-based implementation of SVEA achieves a mean episode return of $\mathbf{877}$ on the `color_hard` benchmark of the challenging *Walker, walk* task (vs. $\mathbf{760}$ using a CNN). SVEA might therefore be a promising technique for future RL studies with CNN-free architectures, where data augmentation appears to be especially important.

6.3. 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. The initial configuration

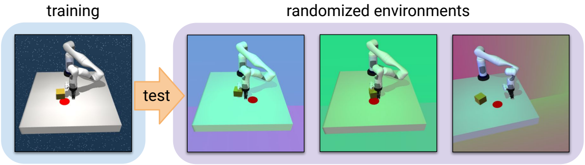

Figure 7. **Robotic manipulation.** Agents are trained in a fixed environment and evaluated on challenging environments with randomized colors, lighting, background, and camera pose.

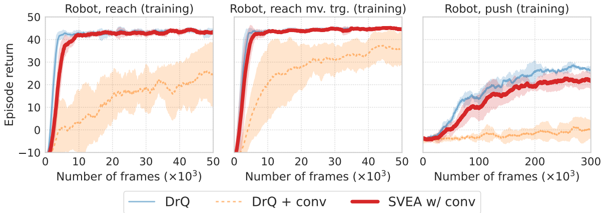

Figure 8. **Stability.** Training performance (episode return) of SVEA and DrQ in 3 robotic manipulation tasks. Mean and std. deviation of 5 seeds. Success rates are shown in Table 2.

Table 2. **Generalization in 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**

of gripper, object, and goal is randomized, the agent uses 2D positional control, and policies are trained using dense rewards. Observations are stacks of RGB frames with no access to state information. Training and test environments are shown in Figure 7. Results are shown in Figure 8 and Table 2. SVEA trained with *conv* augmentation exhibits similar stability and sample efficiency as DrQ trained without, while DrQ + conv is found to have poor sample efficiency and fails to solve the challenging *push* task. SVEA outperforms both baselines in terms of generalization. Interestingly, we find that naïve application of data augmentation has a higher success rate in test environments than the DrQ baseline, despite being less successful in the training environment, which we conjecture is because it is optimized only from augmented data. Conversely, SVEA achieves high success rates during both training and testing.

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

Agarwal, R., Machado, M. C., Castro, P. S., and Bellemare, M. G. Contrastive behavioral similarity embeddings for generalization in reinforcement learning. *ArXiv*, abs/2101.05265, 2021. 2

Bellman, R. A markovian decision process. *Journal of Mathematics and Mechanics*, 6(5):679–684, 1957. 2

Berner, C., Brockman, G., Chan, B., Cheung, V., Debiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., Józefowicz, R., Gray, S., Olsson, C., Pachocki, J. W., Petrov, M., de Oliveira Pinto, H. P., Raiman, J., Salimans, T., Schlatter, J., Schneider, J., Sidor, S., Sutskever, I., Tang, J., Wolski, F., and Zhang, S. Dota 2 with large scale deep reinforcement learning. *ArXiv*, abs/1912.06680, 2019. 1

Chebotar, Y., Handa, A., Makoviychuk, V., Macklin, M., Issac, J., Ratliff, N. D., and Fox, D. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8973–8979, 2019. 2

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations, 2020. 2

Cobbe, K., Klimov, O., Hesse, C., Kim, T., and Schulman, J. Quantifying generalization in reinforcement learning, 2018. 2, 6

Cobbe, K., Klimov, O., Hesse, C., Kim, T., and Schulman, J. Quantifying generalization in reinforcement learning. In *ICML*, 2019. 1

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2020. 2, 7

Farebrother, J., Machado, M. C., and Bowling, M. H. Generalization and regularization in dqn. *ArXiv*, abs/1810.00123, 2018. 1

Fortunato, M., Azar, M. G., Piot, B., Menick, J., Osband, I., Graves, A., Mnih, V., Munos, R., Hassabis, D., Pietquin, O., Blundell, C., and Legg, S. Noisy networks for exploration. *ArXiv*, abs/1706.10295, 2018. 2, 3

Fujimoto, S., Hoof, H. V., and Meger, D. Addressing function approximation error in actor-critic methods. *ArXiv*, abs/1802.09477, 2018. 1, 2, 3, 4

Gidaris, S., Singh, P., and Komodakis, N. Unsupervised representation learning by predicting image rotations, 2018. 6

Givan, R., Dean, T., and Greig, M. Equivalence notions and model minimization in markov decision processes. *Artif. Intell.*, 147:163–223, 2003. 3

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018. 1, 3, 4, 6

Hafner, D., Lillicrap, T., Fischer, I. S., Villegas, R., Ha, D. R., Lee, H., and Davidson, J. Learning latent dynamics for planning from pixels. *ArXiv*, abs/1811.04551, 2019. 5

Hafner, D., Lillicrap, T., Ba, J., and Norouzi, M. Dream to control: Learning behaviors by latent imagination. *ArXiv*, abs/1912.01603, 2020. 5

Hansen, N. and Wang, X. Generalization in reinforcement learning by soft data augmentation. In *International Conference on Robotics and Automation*, 2021. 1, 2, 5, 6, 7

Hansen, N., Sun, Y., Abbeel, P., Efros, A. A., Pinto, L., and Wang, X. Self-supervised policy adaptation during deployment. 2020. 2, 6

Hasselt, H. V., Guez, A., Hessel, M., Mnih, V., and Silver, D. Learning values across many orders of magnitude. In *NIPS*, 2016a. 2, 3

Hasselt, H. V., Guez, A., and Silver, D. Deep reinforcement learning with double q-learning. In *AAAI*, 2016b. 2, 3

Hausknecht, M. and Stone, P. Deep recurrent q-learning for partially observable mdps. In *AAAI Fall Symposia*, 2015. 2, 3

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. B. Momentum contrast for unsupervised visual representation learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9726–9735, 2020. 2

Hessel, M., Modayil, J., Hasselt, H. V., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M. G., and Silver, D. Rainbow: Combining improvements in deep reinforcement learning. In *AAAI*, 2018. 2, 3

Jaderberg, M., Mnih, V., Czarnecki, W. M., Schaul, T., Leibo, J. Z., Silver, D., and Kavukcuoglu, K. Reinforcement learning with unsupervised auxiliary tasks, 2016. 2

Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 1998. 3

Kostrikov, I., Yarats, D., and Fergus, R. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. 2020. 1, 2, 3, 4, 5, 6

Larsen, K. G. and Skou, A. Bisimulation through probabilistic testing (preliminary report). In *Proceedings of the 16th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, 1989. 3

Laskin, M., Lee, K., Stooke, A., Pinto, L., Abbeel, P., and Srinivas, A. Reinforcement learning with augmented data. *arXiv preprint arXiv:2004.14990*, 2020. 1, 2, 3, 4, 5, 6

Lee, K., Lee, K., Shin, J., and Lee, H. A simple randomization technique for generalization in deep reinforcement learning. *ArXiv*, abs/1910.05396, 2019. 2, 6

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

Lillicrap, T., Hunt, J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *CoRR*, abs/1509.02971, 2016. 1, 2, 3, 4

Lin, L. J. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, 8:293–321, 2004. 3

Lin, X., Baweja, H. S., Kantor, G., and Held, D. Adaptive auxiliary task weighting for reinforcement learning. *Advances in neural information processing systems*, 32, 2019. 2

Lyle, C., Rowland, M., Ostrovski, G., and Dabney, W. On the effect of auxiliary tasks on representation dynamics. In *AISTATS*, 2021. 2

Mirowski, P., Pascanu, R., Viola, F., Soyer, H., Ballard, A., Banino, A., Denil, M., Goroshin, R., Sifre, L., Kavukcuoglu, K., Kumaran, D., and Hadsell, R. Learning to navigate in complex environments. *ArXiv*, abs/1611.03673, 2017. 1

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013. 1, 2, 3, 4

Noroozi, M. and Favaro, P. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pp. 69–84. Springer, 2016. 2

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., and Efros, A. A. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2536–2544, 2016. 2

Pathak, D., Gandhi, D., and Gupta, A. Self-supervised exploration via disagreement. *ArXiv*, abs/1906.04161, 2019. 2

Peng, X. B., Andrychowicz, M., Zaremba, W., and Abbeel, P. Sim-to-real transfer of robotic control with dynamics randomization. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018. 2

Pinto, L., Andrychowicz, M., Welinder, P., Zaremba, W., and Abbeel, P. Asymmetric actor critic for image-based robot learning. *arXiv preprint arXiv:1710.06542*, 2017. 1, 2

Raileanu, R., Goldstein, M., Yarats, D., Kostrikov, I., and Fergus, R. Automatic data augmentation for generalization in deep reinforcement learning. *ArXiv*, abs/2006.12862, 2020. 1, 2

Ramos, F., Possas, R., and Fox, D. Bayessim: Adaptive domain randomization via probabilistic inference for robotics simulators. *Robotics: Science and Systems XV*, Jun 2019. 2

Schaul, T., Quan, J., Antonoglou, I., and Silver, D. Prioritized experience replay. *CoRR*, abs/1511.05952, 2016. 2, 3

Schwarzer, M., Anand, A., Goel, R., Hjelm, R. D., Courville, A., and Bachman, P. Data-efficient reinforcement learning with self-predictive representations. *arXiv preprint arXiv:2007.05929*, 2020. 2, 3, 4

Sekar, R., Rybkin, O., Daniilidis, K., Abbeel, P., Hafner, D., and Pathak, D. Planning to explore via self-supervised world models, 2020. 2

Shelhamer, E., Mahmoudieh, P., Argus, M., and Darrell, T. Loss is its own reward: Self-supervision for reinforcement learning. *ArXiv*, abs/1612.07307, 2017. 2

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

Song, X., Jiang, Y., Tu, S., Du, Y., and Neyshabur, B. Observational overfitting in reinforcement learning. *ArXiv*, abs/1912.02975, 2020. 1

Srinivas, A., Laskin, M., and Abbeel, P. Curl: Contrastive unsupervised representations for reinforcement learning. *arXiv preprint arXiv:2004.04136*, 2020. 2, 5, 6

Stone, A., Ramirez, O., Konolige, K., and Jonschkowski, R. The distracting control suite - a challenging benchmark for reinforcement learning from pixels. *ArXiv*, abs/2101.02722, 2021. 2, 5

Stooke, A., Lee, K., Abbeel, P., and Laskin, M. Decoupling representation learning from reinforcement learning. arXiv:2004.14990. 2, 3, 4

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

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018. 2, 4

Tassa, Y., Doron, Y., Muldal, A., Erez, T., Li, Y., de Las Casas, D., Budden, D., Abdolmaleki, A., Merel, J., Lefrancq, A., Lillicrap, T., and Riedmiller, M. DeepMind control suite. Technical report, DeepMind, January 2018. 2, 5

Tian, Y., Krishnan, D., and Isola, P. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019. 2

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., and Abbeel, P. 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, 2

van den Oord, A., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding, 2018. 2

Vinyals, O., Babuschkin, I., Czarnecki, W., Mathieu, M., Dudzik, A., Chung, J., Choi, D., Powell, R., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J., Jaderberg, M., Vezhnevets, A., Leblond, R., Pohlen, T., Dalibard, V., Budden, D., Sulsky, Y., Molloy, J., Paine, T., Gulcehre, C., Wang, Z., Pfaff, T., Wu, Y., Ring, R., Yogatama, D., Wünsch, D., McKinney, K., Smith, O., Schaul, T., Lillicrap, T., Kavukcuoglu, K., Hassabis, D., Apps, C., and Silver, D. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, pp. 1–5, 2019. 1

Wang, K., Kang, B., Shao, J., and Feng, J. Improving generalization in reinforcement learning with mixture regularization. *ArXiv*, abs/2010.10814, 2020. 2

Wang, X., Lian, L., and Yu, S. X. Unsupervised visual attention and invariance for reinforcement learning. *ArXiv*, abs/2104.02921, 2021. 2, 6

Wang, Z., Schaul, T., Hessel, M., Hasselt, H. V., Lanctot, M., and Freitas, N. D. Dueling network architectures for deep reinforcement learning. *ArXiv*, abs/1511.06581, 2016. 2, 3

Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3733–3742, 2018. 2

Xu, Z., Liu, D., Yang, J., and Niethammer, M. Robust and generalizable visual representation learning via random convolutions. *ArXiv*, abs/2007.13003, 2020. 2

Yarats, D., Zhang, A., Kostrikov, I., Amos, B., Pineau, J., and Fergus, R. Improving sample efficiency in model-free reinforcement learning from images, 2019. 2, 5

Yarats, D., Fergus, R., Lazaric, A., and Pinto, L. Reinforcement learning with prototypical representations. *arXiv preprint arXiv:2102.11271*, 2021. 2

Yu, T., Kumar, S., Gupta, A., Levine, S., Hausman, K., and Finn, C. Gradient surgery for multi-task learning. *ArXiv*, abs/2001.06782, 2020. 2

Zhang, A., McAllister, R., Calandra, R., Gal, Y., and Levine, S. Learning invariant representations for reinforcement learning without reconstruction. *ArXiv*, abs/2006.10742, 2020. 2

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

Zhang, R., Isola, P., and Efros, A. A. Colorful image colorization. In *European conference on computer vision*, pp. 649–666. Springer, 2016. 2

Zhu, Y., Mottaghi, R., Kolve, E., Lim, J. J., Gupta, A., Fei-Fei, L., and Farhadi, A. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *ICRA*, pp. 3357–3364. IEEE, 2017. 1

Ziebart, B. D., Maas, A., Bagnell, J. A., and Dey, A. K. Maximum entropy inverse reinforcement learning. In *Proceedings of the 23rd National Conference on Artificial Intelligence*, volume 3, 2008. 3