SECANT: Self-Expert Cloning for Zero-Shot Generalization of Visual Policies

Linxi Fan [1 2 *] Guanzhi Wang [1] De-An Huang [2] Zhiding Yu [2]
Li Fei-Fei [1] Yuke Zhu [3 2] Anima Anandkumar [4 2]

Abstract

Generalization has been a long-standing challenge for reinforcement learning (RL). Visual RL, in particular, can be easily distracted by irrelevant factors in high-dimensional observation space. In this work, we consider robust policy learning which targets zero-shot generalization to unseen visual environments with large distributional shift. We propose SECANT, a novel self-expert cloning technique that leverages image augmentation in two stages to *decouple* robust representation learning from policy optimization. Specifically, an expert policy is first trained by RL from scratch with *weak* augmentations. A student network then learns to mimic the expert policy by supervised learning with *strong* augmentations, making its representation more robust against visual variations compared to the expert. Extensive experiments demonstrate that SECANT significantly advances the state of the art in zero-shot generalization across 4 challenging domains. Our average reward improvements over prior SOTAs are: DeepMind Control (+26.5%), robotic manipulation (+337.8%), vision-based autonomous driving (+47.7%), and indoor object navigation (+15.8%). Code release and video are available at this link.

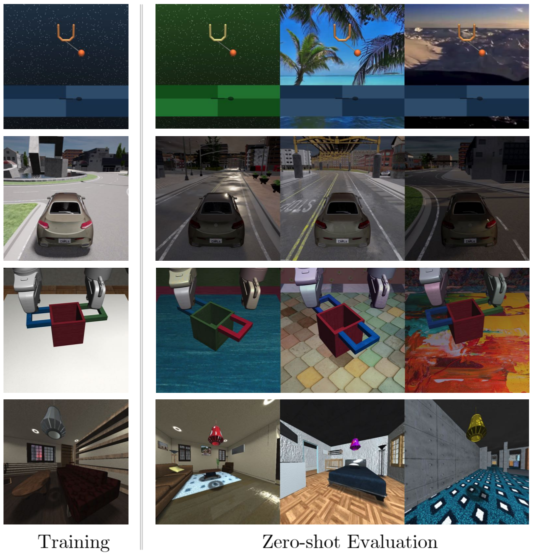


Figure 1. Our proposed benchmark for visual policy generalization in 4 diverse domains. Top to bottom: DMControl Suite (15 settings), CARLA autonomous driving (5 weathers), Robosuite (12 settings), and iGibson indoor navigation (20 rooms).

1. Introduction

Deep reinforcement learning (RL) from image observations has seen much success in various domains (Mnih et al., 2013; Levine et al., 2016; Andrychowicz et al., 2020). However, generalization remains a major obstacle towards reliable deployment. Recent studies have shown that RL agents struggle to generalize to new environments, even with similar tasks (Farebrother et al., 2018; Gamrian & Goldberg, 2019; Cobbe et al., 2019; Song et al., 2020). This suggests that the learned RL policies fail to develop robust representations against irrelevant environmental variations.

Factors of variation in RL problems can be grouped into three main categories: generalization over different visual appearances (Cobbe et al., 2018; Gamrian & Goldberg, 2019; Lee et al., 2020b), dynamics (Packer et al., 2018), and environment structures (Wang et al., 2016; Beattie et al., 2016; Cobbe et al., 2019). In this work, we mainly focus on zero-shot generalization to unseen environments of different visual appearances, but the same semantics.

One well-explored solution for better generalization is data augmentation (LeCun et al., 1998). For image observations in RL, augmentation can be either manually engineered into the simulator, also known as domain randomization (Tobin et al., 2017), or automatic (Laskin et al., 2020). Prior works (Berthelot et al., 2019a; Sohn et al., 2020) distinguish between *weak* augmentations like random cropping, and *strong* augmentations that heavily distort the image, such as Mixup (Zhang et al., 2017) and Cutmix (Yun et al., 2019).

[*] Work done during an internship at NVIDIA. [1] Stanford University, CA, USA. [2] NVIDIA, CA, USA. [3] The University of Texas at Austin, TX, USA. [4] California Institute of Technology, CA, USA. Correspondence to: Linxi Fan <jimfan@cs.stanford.edu>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

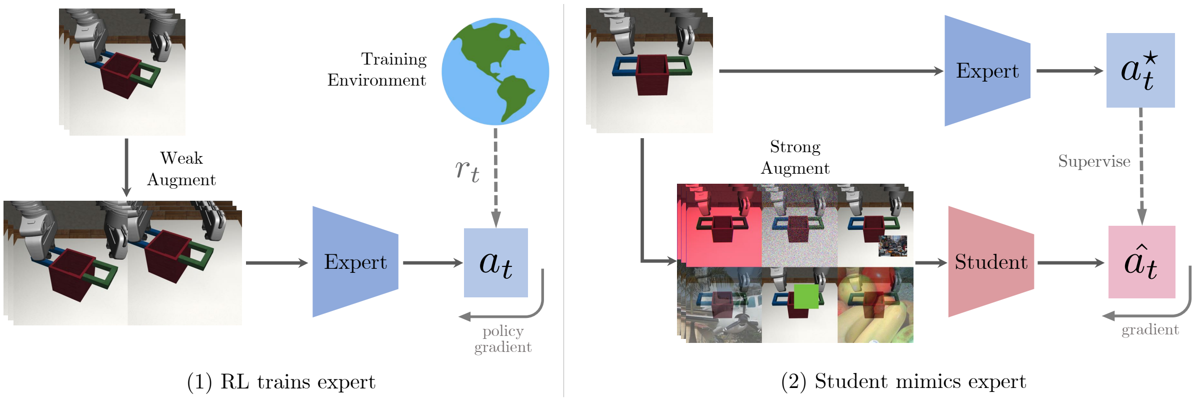


Figure 2. Algorithm overview. SECANT training is split into two stages. **Left, stage 1**: expert policy is trained by RL with weak augmentation (random cropping). **Right, stage 2**: student receives ground-truth action supervision from the expert at every time step, conditioned on the same observation but with strong augmentations, such as cutout-color, Gaussian noise, Mixup, and Cutmix. The student learns robust visual representations invariant to environment distractions, while maintaining high policy performance.

Strong augmentations are known to induce robust and generalizable representations for image classification (Hendrycks et al., 2019). However, naively transplanting them to RL hinders training and results in suboptimal performance (Laskin et al., 2020). Therefore, weak augmentations like random cropping are the most effective for RL at training time (Kostrikov et al., 2020). This poses a dilemma: more aggressive augmentations are necessary to cultivate better generalization for the visual domain (Hendrycks et al., 2019), but RL does not benefit to the same extent as supervised learning since the training is fragile to excessive data variations.

We argue that the dilemma exists because it conflates two problems: policy learning and robust representation learning. To decouple them, we draw inspiration from policy distillation (Rusu et al., 2015) where a student policy distills knowledge from one or more experts. The technique is used for different purposes, such as efficient policy deployment, multi-task RL, and policy transfer (Teh et al., 2017; Arora et al., 2018; Czarnecki et al., 2019). In this work, we introduce a new instantiation of policy distillation that addresses the dilemma effectively.

Summary of our contributions:

- We propose SECANT (**S**elf **E**xpert **C**loning for **A**daptation to **N**ovel **T**est-environments), a novel algorithm that solves policy learning and robust representation learning *sequentially*, which achieves strong zero-shot generalization performance to unseen visual environments.

- We design and standardize a diverse and challenging suite of benchmarks in 4 domains: Deepmind Control Suite (DMControl), autonomous driving, robotic manipulation, and indoor object navigation. Except for DMControl, the other 3 environments feature test-time visual appearance drifts that are representative of real-world applications.

- We demonstrate that SECANT is able to dominate prior state-of-the-art methods on the majority of tasks, often by substantial margins, across all 4 domains.

Our **key insight** is to solve policy optimization first, and then to robustify its representation by imitation learning with strong augmentations. First, an expert neural network is trained by RL with random cropping on the original environment. It learns a high-performance policy but cannot handle distribution shifts. Second, a student network learns to mimic the behavior of the expert, but with a crucial difference: the expert computes the ground-truth actions from *unmodified* observations, while the student learns to predict the same actions from heavily *corrupted* observations. The student optimizes a supervised learning objective, which has better training stability than RL, and the strong augmentations greatly remedy overfitting at the same time. Thus, SECANT is able to acquire robust representations without sacrificing policy performance.

Our method is strictly zero-shot, because no reward signal is allowed at test time, and neither the expert nor the student sees the test environments during training. SECANT is trained once and does not perform any test-time adaptation. In contrast, PAD (Hansen et al., 2020), a prior SOTA method, adds a self-supervised auxiliary loss on intermediate representations during training, and continues to fine-tune the policy weights using this loss signal at testing. SECANT is more efficient compared to PAD, because the latter requires expensive gradient computation at every inference step and is impractical to deploy on mobile robots.

We benchmark on Deepmind Control Suite (DMControl) with randomized color and video backgrounds (Hansen et al., 2020), and show that SECANT is able to outperform prior SOTA in **14 out of 15** unseen environments with an average score increase of **26.5%**. While DMControl is a popular benchmark, its test-time variations are artificial and not representative of real applications. Therefore, we further construct 3 new benchmarks with more realistic distribution shifts (Fig. 1), based on existing simulators: **(1) Robosuite** (Zhu et al., 2020): single-hand and bimanual robotic tasks. We add new appearances of table, background, and objects of interest that are varied at test time; **(2) CARLA** (Dosovitskiy et al., 2017): autonomous driving across 5 unseen weather conditions that feature highly realistic rendering of raining, sunlight changes, and shadow effects. **(3) iGibson** (Shen et al., 2020): indoor object navigation in 20 distinct rooms with a large variety of interior design and layouts that we standardize. We hope that these new challenging environments will facilitate more progress towards generalizable visual policy learning.

2. Related Work

Generalization in Deep RL. There is a plethora of literature that highlights the overfitting problem in deep RL (Rajeswaran et al., 2017; Packer et al., 2018; Zhang et al., 2018; Justesen et al., 2018; Machado et al., 2018; Cobbe et al., 2018; Wang et al., 2019; Cobbe et al., 2019; Yarats et al., 2019; Raileanu & Rocktäschel, 2020). One class of approach is to re-design the training objectives to induce invariant representations directly. Zhang et al. (2020b) and Srinivas et al. (2020) aim to learn robust features via deep metric learning (Ferns & Precup, 2014). Rao et al. (2020) combines RL with CycleGAN. Jiang et al. (2020) employs automatic curriculum for generalization. PAD (Hansen et al., 2020) adds a self-supervised auxiliary component that can be adapted at test time. In contrast to these prior works, SECANT is a plug-and-play method that neither modifies the existing RL algorithm, nor requires computationally expensive test-time fine-tuning. Similar to us, ATC (Stooke et al., 2020) separates representation learning from RL. It pretrains an encoder, fine-tunes with reward, and evaluates in the *same* environment. In contrast, SECANT solves policy learning first before robustification, and focuses heavily on zero-shot generalization instead.

Other works (Farebrother et al., 2018; Cobbe et al., 2018) apply regularization techniques originally developed for supervised learning, such as L2 regularization, BatchNorm (Ioffe & Szegedy, 2015), and dropout (Srivastava et al., 2014). Igl et al. (2019) regularizes RL agents via selective noise injection and information bottleneck. These methods improve policy generalization in Atari games and CoinRun (Cobbe et al., 2018). SECANT is orthogonal to these techniques and can be combined for further improvements. We also contribute a new benchmark with more realistic tasks and variations than video games.

Data augmentation and robustness. Semantic-preserving image transformations have been widely adopted to improve the performance and robustness of computer vision systems (Hendrycks & Dietterich, 2018; Hendrycks et al., 2019; Berthelot et al., 2019b; Sohn et al., 2020). Domain randomization (DR) (Tobin et al., 2017; Peng et al., 2018) produces randomized textures of environmental components. It is a special type of data augmentation that requires extensive manual engineering and tuning of the simulator (Pinto et al., 2017; Yang et al., 2019). RL training, however, benefits the most from weak forms of augmentations that do not add extra difficulty to the policy optimization process (Laskin et al., 2020; Kostrikov et al., 2020; Raileanu et al., 2020; Hansen et al., 2020). By design, SECANT unlocks a multitude of strong augmentation operators that are otherwise suboptimal for training in prior works. We successfully employ techniques from supervised image classification like Cutmix (Yun et al., 2019) and Mixup (Zhang et al., 2017); the latter has also been explored in (Wang et al., 2020).

Policy distillation. SECANT belongs to the policy distillation family, a special form of knowledge distillation (Hinton et al., 2015) for RL. Prior works use policy distillation for different purposes (Czarnecki et al., 2019). Chen et al. (2020) and Lee et al. (2020a) train an expert with privileged simulator information (e.g. groundtruth physical states) to supervise a student policy that can only access limited sensors at deployment. Zhou et al. (2020) transfers navigation policies across domains through an intermediate proxy model. Igl et al. (2020) reduces the non-stationary effects of RL environment by repeated knowledge transfer. Other works involve multi-task student networks (Rusu et al., 2015; Teh et al., 2017; Arora et al., 2018) that distill from multiple experts simultaneously. SECANT differs from these works because our expert and student share the same task and observation information, but shoulder different responsibilities: expert handles policy optimization while student addresses visual generalization.

Our method is related to FixMatch (Sohn et al., 2020), which imposes a pseudo-label distillation loss on two different augmentations of the same image. In contrast, our expert only needs to overfit to the training environment, while the student distills from a frozen expert to learn robust representation. Concurrent work Hansen & Wang (2020) also validates the benefit of decoupling and strong augmentation. In comparison, SECANT is conceptually simpler and does not require modifying the RL training pipeline. Another closely related field is imitation learning (Schaal et al., 1997; Argall et al., 2009; Ross et al., 2011; Ho & Ermon, 2016). Our student imitates without external demonstration data, hence the name "self-expert cloning".

3. Preliminaries

Soft Actor-Critic. In this work, we mainly consider continuous control from raw pixels. The agent receives an image observation $o \in \mathbb{R}^{C\times H\times W}$ and outputs a continuous action $a \in \mathbb{R}^d$. SAC (Haarnoja et al., 2018a;b) is a state-of-the-art off-policy RL algorithm. It learns a policy $\pi(a|o)$ and a critic $Q(o, a)$ that maximize a weighted combination of reward and policy entropy, $\mathbb{E}_{(o_t,a_t)\sim\pi}\left[\sum_t r_t + \alpha\mathcal{H}(\pi(\cdot|o_t))\right]$. SAC stores experiences into a replay buffer $\mathcal{D}$. The critic parameters are updated by minimizing the Bellman error using transitions sampled from $\mathcal{D}$:

$$\mathcal{L}_Q = \mathbb{E}_{(o_t,a_t)\sim\mathcal{D}}\left[\left(Q(o_t, a_t) - (r_t + \gamma V(o_{t+1}))\right)^2\right] \tag{1}$$

By sampling an action under the current policy, we can estimate the soft state value as following:

$$V(o_{t+1}) = \mathbb{E}_{a'\sim\pi}\left[\bar{Q}(o_{t+1}, a') - \alpha\log\pi(a'|o_{t+1})\right] \tag{2}$$

where $\bar{Q}$ denotes an exponential moving average of the critic network. The policy is updated by minimizing the divergence from the exponential of the soft-Q function:

$$\mathcal{L}_\pi = -\mathbb{E}_{a_t\sim\pi}\left[Q(o_t, a_t) - \alpha\log\pi(a_t|o_t)\right] \tag{3}$$

where α is a learnable temperature parameter that controls the stochasticity of the optimal policy.

Dataset Aggregation (DAgger). Ross et al. (2011) is an iterative imitation learning algorithm with strong performance guarantees. First, it rolls out an expert policy π_e to seed an experience dataset $\mathcal{D}^0$. The student policy π_s^0 is trained by supervised learning to best mimic the expert on those trajectories. Then at iteration i, it rolls out π_s^i to collect more trajectories that will be added to $\mathcal{D}^i$. π_s^{i+1} will then be trained on the new *aggregated* dataset $\mathcal{D}^{i+1}$, and the process repeats until convergence. Even though more advanced imitation algorithms have been developed (Ho & Ermon, 2016), DAgger is conceptually simple and well-suited for SECANT because our student network can query the expert for dense supervision at every time step.

4. SECANT

The goal of our proposed self-expert cloning technique is to learn a robust policy that can generalize zero-shot to unseen visual variations. SECANT training can be decomposed into two stages. Algorithm 1 shows the full pseudocode.

4.1. Expert policy

In the first stage, we train a high-performance expert policy in the original environment with weak augmentations. In visual continuous control tasks, the policy is parametrized by a feed-forward deep convolutional network $\pi_e(\mathcal{O}; \theta_e) : \mathbb{R}^{C\times H\times W} \to \mathbb{R}^d$ that maps an image observation to a d-dimensional continuous action vector. In practice, we employ a frame stacking technique that concatenates T consecutive image observations along the channel dimension to incorporate temporal information (Mnih et al., 2013). The augmentation operator is a semantic-preserving image transformation $f : \mathbb{R}^{C\times H\times W} \to \mathbb{R}^{C'\times H'\times W'}$. Prior works have found that random cropping performs the best in a range of environments, therefore we adopt it as the default weak augmentation for the expert (Laskin et al., 2020).

Algorithm 1 SECANT: Self-Expert Cloning

```
$\pi_e$, $\pi_s$: randomly initialized expert and student policies
$\mathcal{F}_{weak}$, $\mathcal{F}_{strong}$: sets of image augmentations
$\mathcal{B}$: experience replay buffer
for $t$ in $1, \dots, T_{RL}$ do
    Sample experience batch $\tau_t = (o_t, a_t, o_{t+1}, r) \sim \mathcal{B}$
    Sample weak augmentation $f \sim \mathcal{F}_{weak}$
    Augment $o_t = f(o_t); o_{t+1} = f(o_{t+1})$
    Update $\pi_e$ to minimize $\mathcal{L}_{RL}(\tau_t)$
end for
Roll out $\pi_e$ to collect an initial dataset $\mathcal{D}$ of trajectories
for $t$ in $1, \dots, T_{imitate}$ do
    Sample observation batch $o \sim \mathcal{D}$
    Sample strong augmentation $f \sim \mathcal{F}_{strong}$
    Update $\pi_s$ to minimize $\|\pi_s(f(o)) - \pi_e(o)\|_F$
    Roll out $\pi_s$ for one environment step and add to the
        dataset $\mathcal{D} \leftarrow \mathcal{D} \cup \{o_s\}$
end for
```

The expert can be optimized by any standard RL algorithm. We select Soft Actor-Critic (SAC) due to its wide adoption in continuous control tasks (Haarnoja et al., 2018a;b). The expert is optimized by gradient descent to minimize the SAC objectives (Equations 1 and 3). Since we place little restrictions on the expert, our method can even be used to robustify pre-trained policy network checkpoints, such as the RL Model Zoo (Raffin, 2018).

4.2. Student policy distillation

In the second stage, we train a student network to predict the optimal actions taken by the expert, conditioned on the same observation but with heavy image corruption. This stage does *not* need further access to the reward signal. Formally, the student is also a deep convolutional network $\pi_s(\mathcal{O}; \theta_s) : \mathbb{R}^{C\times H\times W} \to \mathbb{R}^d$ that may have different architecture from the expert. The student policy distills from the expert following the DAgger imitation procedure (Sec 3). First, we roll out the *expert* policy to collect an initial dataset $\mathcal{D}$ of trajectories. Next, at each iteration, we select a strong augmentation operator $f \sim \mathcal{F}_{strong}$ and apply it to a batch of observations o sampled from $\mathcal{D}$. We alternate between (1) updating the student's parameters by gradient descent on a supervised regression loss: $\mathcal{L}(o; \theta_s) = \|\pi_s(f(o)) - \pi_e(o)\|_F$ and (2) adding more experiences to $\mathcal{D}$ under the latest *student* policy.

In the experiments, we consider 1 type of weak augmenta-

Table 1. DMControl: SECANT outperforms prior SOTA methods substantially in **14 out of 15** settings with **+26.5%** boost on average.

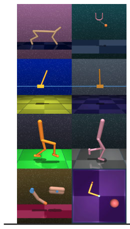
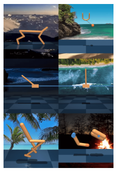
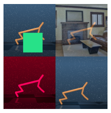
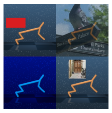
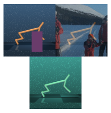
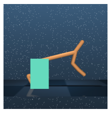
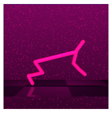
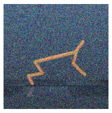
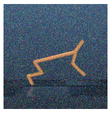
Setting	Task	SECANT (Ours)		SAC	SAC+crop	DR	NetRand	SAC+IDM	PAD
	Cheetah run	**582 ± 64**	(+88.3%)	133 ± 26	100 ± 27	145 ± 29	309 ± 66	121 ± 38	159 ± 28
	Ball in cup catch	**958 ± 7**	(+ 8.1%)	151 ± 36	359 ± 76	470 ± 252	886 ± 57	471 ± 75	563 ± 50
	Cartpole swingup	**866 ± 15**	(+27.2%)	248 ± 24	537 ± 98	647 ± 48	681 ± 122	585 ± 73	630 ± 63
	Cartpole balance	**992 ± 6**	(+ 0.8%)	930 ± 36	769 ± 63	867 ± 37	984 ± 13	835 ± 40	848 ± 29
	Walker walk	**856 ± 31**	(+27.6%)	144 ± 19	191 ± 33	594 ± 104	671 ± 69	406 ± 29	468 ± 47
	Walker stand	**939 ± 7**	(+ 4.3%)	365 ± 79	748 ± 60	715 ± 96	900 ± 75	743 ± 37	797 ± 46
	Finger spin	**910 ± 115**	(+ 3.1%)	504 ± 114	847 ± 116	465 ± 314	883 ± 156	757 ± 62	803 ± 72
	Reacher easy	**639 ± 63**	(+29.1%)	185 ± 70	231 ± 79	105 ± 37	495 ± 101	201 ± 32	214 ± 44
	Cheetah run	**428 ± 70**	(+56.8%)	80 ± 19	102 ± 30	150 ± 34	273 ± 26	164 ± 42	206 ± 34
	Ball in cup catch	**903 ± 49**	(+57.3%)	172 ± 46	477 ± 40	271 ± 189	574 ± 82	362 ± 69	436 ± 55
	Cartpole swingup	**752 ± 38**	(+44.3%)	204 ± 20	442 ± 74	485 ± 67	445 ± 50	487 ± 90	521 ± 76
	Cartpole balance	**863 ± 32**	(+12.7%)	569 ± 79	641 ± 37	766 ± 92	708 ± 28	691 ± 76	687 ± 58
	Walker walk	**842 ± 47**	(+17.4%)	104 ± 14	244 ± 83	655 ± 55	503 ± 55	694 ± 85	717 ± 79
	Walker stand	932 ± 15		274 ± 39	601 ± 36	869 ± 60	769 ± 78	902 ± 51	**935 ± 20**
	Finger spin	**861 ± 102**	(+21.6%)	276 ± 81	425 ± 69	338 ± 207	708 ± 170	605 ± 61	691 ± 80

Table 2. Ablation on student augmentations: given the same experts trained with random cropping, we ablate 6 strong augmentations and their mixtures for the student. Combo[1-3] randomly select an augmentation from their pool to apply to each observation.

Setting	Tasks	Combo1	Combo2	Combo3	Cutout-color	Conv	Mixup	Cutmix	Gaussian	Impulse
Random Color	Cartpole swingup	**866 ± 15**	865 ± 17	863 ± 15	776 ± 33	860 ± 15	825 ± 16	751 ± 43	720 ± 86	751 ± 45
	Cheetah run	**582 ± 64**	522 ± 166	570 ± 50	343 ± 153	318 ± 123	222 ± 38	303 ± 82	373 ± 110	382 ± 121
	Walker walk	856 ± 31	854 ± 27	832 ± 45	701 ± 75	**866 ± 22**	756 ± 46	658 ± 54	770 ± 56	727 ± 47
Random Video	Cartpole swingup	752 ± 38	765 ± 55	**778 ± 37**	607 ± 31	556 ± 61	677 ± 43	647 ± 56	580 ± 61	639 ± 22
	Cheetah run	**428 ± 70**	409 ± 31	406 ± 33	183 ± 46	229 ± 30	309 ± 65	209 ± 43	196 ± 38	224 ± 25
	Walker walk	**842 ± 47**	836 ± 36	792 ± 59	631 ± 52	531 ± 54	759 ± 64	675 ± 22	488 ± 31	471 ± 17
Example Augmentations										

tion (random cropping) and 6 types of strong augmentation techniques developed in RL and robust image classification literature (Hendrycks & Dietterich, 2018; Hendrycks et al., 2019; Laskin et al., 2020; Lee et al., 2020b). We refer to weak augmentations as the ones that can substantially improve RL optimization at training time, while strong augmentations are less effective as they make training more difficult. We focus only on random cropping for weak augmentation in this work, and defer other potential operators to future works. Below are brief descriptions of the augmentations we study:

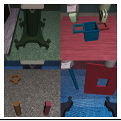
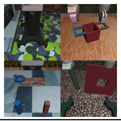
Cutout-color (Cc): inserts a small rectangular patch of random color into the observation at a random position. **Random convolution (Cv)**: passes the input observation through a random convolutional layer. **Gaussian (G)**: adds Gaussian noise. **Impulse (I)**: adds the color analogue of salt-and-pepper noise. **Mixup (M)** (Zhang et al., 2017): linearly blends the observation with a distracting image I: $f(o) = \alpha o + (1 - \alpha)I$. We randomly sample I from 50K COCO images (Lin et al., 2014), and sample $\alpha \sim \text{Uniform}(0.2, 0.6)$. **Cutmix (Cm)** (Yun et al., 2019): similar to Cutout-color except that the patch is randomly sampled from COCO images. These augmentations can be categorized into low-frequency noise (**Cc** and **Cv**), high-frequency unstructured noise (**G** and **I**), and high-frequency structured noise (**M** and **Cm**). Mixup and Cutmix with image distractions are novel operators that have not been studied for visual policy generalization.

We also investigate combinations of the above, and find empirically that random sampling from low-frequency and high-frequency *structured* noise types yields the best overall results. We note that adding random cropping to the mix benefits performance slightly, likely because it improves the spatial invariance of the student's representation. We design three combination recipes (Table 2): Combo1 (Cc+Cv+M+Crop), Combo2 (Cc+Cv+M+Cm+Crop), and Combo3 (Cc+Cv+M). We have not done an exhaustive search, so it is possible that better combinations exist.

5. Experiments

We propose a new benchmark of 4 diverse domains (Fig. 1) to systematically assess the generalization ability of visual

Table 3. Robosuite results. The 3 sets of test environments are progressively harder (*easy*, *hard*, and *extreme*) with more distracting textures of the table, floor, and objects. SECANT gains an average of +337.8% reward over prior SOTA.

Setting	Tasks	SECANT (Ours)	SAC	SAC+crop	DR	NetRand	SAC+IDM	PAD
	Door opening	**782** ± **93** (+ 78.5%)	17 ± 12	10 ± 8	177 ± 163	438 ± 157	3 ± 2	2 ± 1
	Nut assembly	**419** ± **63** (+ 73.1%)	3 ± 2	6 ± 5	12 ± 7	242 ± 28	13 ± 12	11 ± 10
	Two-arm lifting	**610** ± **28** (+883.9%)	29 ± 11	23 ± 10	41 ± 9	62 ± 43	20 ± 8	22 ± 7
	Peg-in-hole	**837** ± **42** (+114.6%)	186 ± 62	134 ± 72	139 ± 37	390 ± 68	150 ± 41	142 ± 37
	Door opening	**522** ± **131** (+292.5%)	11 ± 10	11 ± 7	37 ± 31	133 ± 82	2 ± 1	2 ± 1
	Nut assembly	**437** ± **102** (+141.4%)	6 ± 7	9 ± 8	33 ± 18	181 ± 53	34 ± 28	24 ± 26
	Two-arm lifting	**624** ± **40** (+923.0%)	28 ± 11	27 ± 9	61 ± 15	41 ± 25	17 ± 6	19 ± 8
	Peg-in-hole	**774** ± **76** (+140.4%)	204 ± 81	143 ± 62	194 ± 41	322 ± 72	165 ± 75	164 ± 69
	Door opening	**309** ± **147** (+120.7%)	11 ± 10	6 ± 4	52 ± 46	140 ± 107	2 ± 1	2 ± 1
	Nut assembly	**138** ± **56** (+ 53.3%)	2 ± 1	10 ± 7	12 ± 7	90 ± 61	4 ± 3	4 ± 3
	Two-arm lifting	**377** ± **37** (+1156.7%)	25 ± 7	12 ± 6	30 ± 13	12 ± 11	24 ± 10	21 ± 10
	Peg-in-hole	**520** ± **47** (+ 75.7%)	164 ± 63	130 ± 81	154 ± 34	296 ± 90	155 ± 73	154 ± 72

agents. They offer a wide spectrum of visual distribution shifts for testing. In each domain, we investigate how well an algorithm trained in one environment performs on various unseen environments in a zero-shot setting, which disallows reward signal and extra trials at test time.

For each task, we benchmark SECANT extensively against prior state-of-the-art algorithms: **SAC**: plain SAC with no augmentation. **SAC+crop**: SAC with time-consistent random cropping (Kostrikov et al., 2020). **DR**: domain randomization. To simulate realistic deployment, our randomized training distributions are narrower than the test distributions. **NetRand**: Network Randomization (Lee et al., 2020b), which augments the observation image by random convolution. **SAC+IDM**: SAC trained with an auxiliary inverse dynamics loss (Pathak et al., 2017). **PAD**: prior SOTA method on top of SAC+IDM that fine-tunes the auxiliary head at test time (Hansen et al., 2020). Following prior works (Hansen et al., 2020) on DMControl, we repeat training across 10 random seeds to report the mean and standard deviation of the rewards. We use 5 random seeds for all other simulators and ablation studies.

Algorithm details. SECANT builds upon SAC, and adopts similar hyperparameters and network architecture as Kostrikov et al. (2020). Observations are stacks of 3 consecutive RGB frames. For all tasks, we use a 4-layer feed-forward ConvNet with no residual connection as encoder for both the SECANT expert and student, although they do not have to be identical. PAD, however, requires a deeper encoder network (11 layers) to perform well in DMControl (Hansen et al., 2020). For all other simulators, we conduct a small grid search and find that 6-layer encoders work best for both SAC+IDM and PAD. After the encoder, 3 additional fully connected layers map the visual embedding to action. We include a detailed account of all hyperparameters and architecture in the supplementary.

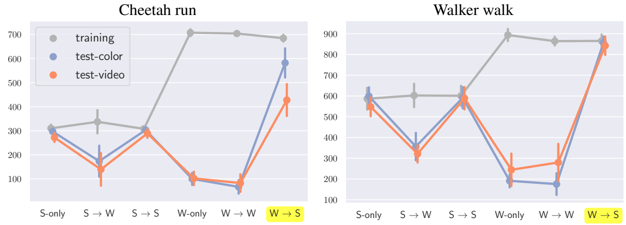


Figure 3. Ablation on different strategies to apply augmentation. "S-only" denotes single-stage policy trained with strong augmentation, and S → W means strongly-augmented expert imitated by weakly-augmented student. The recipe for SECANT is W → S.

5.1. Deepmind Control Suite

We follow the settings in Hansen et al. (2020) and experiment with 8 tasks from DMControl. We measure generalization to (1) randomized colors of the background and robot itself, and (2) natural videos as dynamic background (Fig. 1). SECANT *significantly outperforms* prior SOTA in all but one task, often by substantial margins up to **88.3%** (Table 1). All methods are trained for 500K steps with dense task-specific rewards. SAC+crop is the same as SECANT's expert, from which the student distills for up to 30K steps without reward. SAC+IDM and PAD numbers are from Hansen et al. (2020).

Choice of student augmentations (Table 2). We hypothesize that SECANT needs multiple kinds of augmentations to resist a wide variety of distribution shifts at test time. Keeping the experts fixed, we study the effect of 6 different augmentations and their combinations on the student. In the challenging random video environments, Mixup and Cutmix tend to outperform other single operators. In most tasks, sampling from a mixture of augmentations generalizes better than solos, thus confirming our hypothesis. We adopt Combo1 for all SECANT results in Table 1.

Table 4. Ablation on imitation strategies. DAgger outperforms Expert-only and Student-only data collection in the second stage.

Setting	Task	DAgger	Expert	Student
Random Color	Cheetah run	**582 ± 64**	519 ± 73	347 ± 326
	Walker walk	**856 ± 31**	818 ± 41	854 ± 33
Random Video	Cheetah run	**428 ± 70**	291 ± 41	264 ± 241
	Walker walk	**842 ± 47**	778 ± 67	822 ± 55

Table 5. Ablation on SECANT-Parallel variant. It is advantageous to train expert and student sequentially rather than in parallel.

Setting	Task	SECANT	SECANT-Parallel
Random Color	Cheetah run	**582 ± 64**	302 ± 248
	Ball in cup catch	**958 ± 7**	790 ± 332
	Cartpole swingup	**866 ± 15**	834 ± 8
	Walker walk	**856 ± 31**	768 ± 22
Random Video	Cheetah run	**428 ± 70**	276 ± 216
	Ball in cup catch	**903 ± 49**	676 ± 280
	Cartpole swingup	752 ± 38	**764 ± 17**
	Walker walk	**842 ± 47**	699 ± 21

Single stage vs two-stage augmentation (Fig. 3). The premise of SECANT is that we cannot effectively apply strong augmentation (Combo1) in one stage to learn robust policies. We put this assumption to test. In Fig. 3, "S-only" and "W-only" are single-stage policies trained with strong or weak augmentations. $X \rightarrow Y$ denotes two-stage training, e.g. S → W means a strongly-augmented expert is trained first, and then a weakly-augmented student imitates it. We highlight **4 key findings**: **(1)** single-stage RL trained with strong augmentation (S-only) underperforms in both training and test environments consistently, due to poor optimization. **(2)** The student is typically upper-bounded by the expert's performance, thus both S → W and S → S produce sub-optimal policies. **(3)** single-stage policy trained with random cropping (W-only) overfits on the training environment and generalizes poorly. Adding a weakly-augmented student (W → W) does not remedy the overfitting. **(4)** The only effective strategy is a weakly-augmented expert followed by a strongly-augmented student (W → S), which is exactly SECANT. It recovers the strong performance on the training environment, and bridges the generalization gap in unseen test environments. We include more extensive ablation results with different 2-stage augmentation strategies in the supplementary.

Ablation on imitation strategies (Table 4). SECANT uses DAgger (Sec. 3) in the second stage, which rolls out the expert policy to collect initial trajectories, and then follows the student's policy. The alternatives are using expert or student policy alone to collect all trajectory data. The former approach lacks data diversity, while the latter slows down learning due to random actions in the beginning. Table 4 validates the choice of DAgger for policy distillation.

Ablation on the parallel-distillation variant (Table 5). Can we train the expert and the student at the same time, rather than sequentially? We consider a variant of our method, called SECANT-Parallel, that trains the expert alongside the student while keeping all other settings fixed. Similar to SECANT, it also enjoys the nice property of disentangling robust representation learning from policy optimization. However, the student in SECANT distills from a fully-trained and frozen expert, while the student in SECANT-Parallel has to distill from a non-stationary expert, which leads to suboptimal performances. Table 5 demonstrates that it is more beneficial to adopt our proposed two-stage procedure, as SECANT outperforms SECANT-Parallel in a majority of tasks. We include more SECANT-Parallel results on Robosuite in the supplementary.

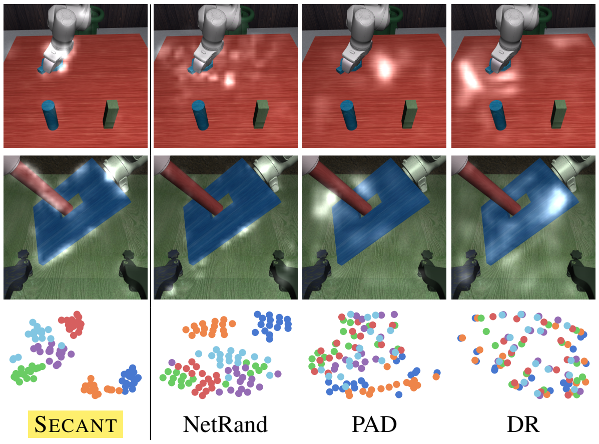


Figure 4. Row 1 and 2: saliency map of the learned policies in unseen tests. SECANT attends to the components crucial to the task, while other agents often focus on irrelevant places. Row 3: t-SNE visualization of state embeddings. Our method correctly groups semantically similar states with different visual appearances.

5.2. Robosuite: Robotic Manipulation

Robosuite (Zhu et al., 2020) is a modular simulator for robotic research. We benchmark SECANT and prior methods on 4 challenging single-arm and bimanual manipulation tasks. We use the Franka Panda robot model with operational space control, and train with task-specific dense reward. All agents receive a 168×168 egocentric RGB view as input (example in Table 3, high-res version in supplementary). The positions of moveable objects are randomized in each episode. **Door opening**: a robot arm must turn the handle and open the door in front of it. **Nut assembly**: two colored pegs (square and round) are mounted on the tabletop. The robot must fit the round nut onto the round peg. **Two-arm lifting**: two arms on opposite ends must each grab a handle of a large pot and lift it up above certain height. **Peg-in-hole**: one arm holds a board with a square hole in the center, and the other holds a long peg. Two arms must coordinate to insert the peg into the hole.

Table 6. Robustness analysis in Robosuite: we measure the cycle consistency of observation embeddings across trajectories of the same task but different appearances. The higher the accuracy, the more robust the representation is against visual variations.

Cycle	Tasks	SECANT (Ours)	SAC	SAC+crop	DR	NetRand	SAC+IDM	PAD
2-way	Nut assembly	**77.3 ± 7.6** (+29.3%)	24.0 ± 6.0	16.0 ± 3.7	25.3 ± 11.9	48.0 ± 15.9	29.3 ± 13.0	26.7 ± 9.4
	Two arm lifting	**72.0 ± 9.9** (+32.0%)	20.0 ± 0.0	18.7 ± 3.0	24.0 ± 10.1	40.0 ± 14.9	18.7 ± 3.0	18.7 ± 5.6
3-way	Nut assembly	**33.3 ± 8.2** (+17.3%)	16.0 ± 8.9	16.0 ± 3.7	8.0 ± 11.0	9.3 ± 10.1	10.7 ± 7.6	10.7 ± 7.6
	Two arm lifting	**32.0 ± 8.7** (+20.0%)	6.7 ± 9.4	2.7 ± 3.7	10.7 ± 10.1	6.7 ± 6.7	12.0 ± 7.3	12.0 ± 7.3

Table 7. CARLA autonomous driving. The different weathers feature highly realistic raining, shadow, and sunlight changes. We report distance (m) travelled in a town without collision. SECANT drives **+47.7%** farther on average than other agents at test time.

Setting	Weather	SECANT (Ours)	SAC	SAC+crop	DR	NetRand	SAC+IDM	PAD
Training	Clear noon	596 ± 77	282 ± 71	**684 ± 114**	486 ± 141	648 ± 61	582 ± 96	632 ± 126
Test Weathers	Wet sunset	**397 ± 99** (+ 39.8%)	57 ± 14	26 ± 18	9 ± 11	284 ± 84	25 ± 11	36 ± 12
	Wet cloudy noon	**629 ± 204** (+ 5.7%)	180 ± 45	283 ± 85	595 ± 260	557 ± 107	433 ± 105	515 ± 52
	Soft rain sunset	**435 ± 66** (+ 73.3%)	55 ± 28	38 ± 25	25 ± 41	251 ± 104	36 ± 32	41 ± 37
	Mid rain sunset	**470 ± 202** (+101.7%)	50 ± 8	37 ± 16	24 ± 24	233 ± 117	42 ± 23	32 ± 21
	Hard rain noon	**541 ± 96** (+ 18.1%)	237 ± 85	235 ± 129	341 ± 96	458 ± 72	156 ± 194	308 ± 141

All agents are trained with clean background and objects, and evaluated on 3 progressively harder sets of environments (Table 3). We design 10 variations for each task and difficulty level, and report the mean reward over 100 evaluation episodes (10 per variation). Reward below 100 indicates a failure to solve the task. SECANT gains an average of **+287.5%** more reward in *easy* set, **+374.3%** in *hard* set, and **+351.6%** in *extreme* set over the best prior method. The *hard* and *extreme* settings are particularly challenging because the objects of interest are difficult to discern from the noisy background. For nut assembly and two-arm lifting, SECANT is the only agent able to obtain non-trivial partial rewards in *hard* and *extreme* modes consistently.

Embedding robustness analysis. To verify that our method develops high-quality representation, we measure the cycle consistency metric proposed in Aytar et al. (2018). First, given two trajectories U and V, observation $u_i \in U$ locates its nearest neighbor in V: $v_j = \arg\min_{v \in V} \|\phi(u_i) - \phi(v)\|^2$, where $\phi(\cdot)$ denotes the 50-D embedding from the visual encoder of the learned policies. Then in reverse, v_j finds its nearest neighbor from U: $u_k = \arg\min_{u \in U} \|\phi(u) - \phi(v_j)\|^2$. u_i is cycle consistent if and only if $|i - k| \leq 1$, i.e. it returns to the original position. High cycle consistency indicates that the two trajectories are accurately aligned in the embedding space, despite their visual appearance shifts. We also evaluate 3-way cycle consistency that involves a third trajectory W, and measure whether u_i can return to itself along *both* $U \to V \to W \to U$ and $U \to W \to V \to U$. In Table 6, we sample 15 observations from each trajectory, and report the mean cycle consistency over 5 trials. In Fig. 4, we also visualize the state embeddings of door-opening task with t-SNE (Maaten & Hinton, 2008). Both quantitative and qualitative analyses show that SECANT significantly improves the robustness of visual representation over the baselines.

Saliency visualization. To better understand how the agents execute their policies, we compute saliency maps as described in Greydanus et al. (2018). We add Gaussian perturbation to the observation image at every 5×5 pixel patch, and visualize the saliency patterns in Fig. 4. SECANT is able to focus on the most task-relevant objects, even with novel textures it has not encountered during training.

5.3. CARLA: Autonomous Driving

To further validate SECANT's generalization ability on natural variations, we construct a realistic driving scenario with visual observations in the CARLA simulator (Dosovitskiy et al., 2017). The goal is to drive as far as possible along a figure-8 highway (CARLA Town 4) in 1000 time steps without colliding into 60 moving pedestrians or vehicles. Our reward function is similar to Zhang et al. (2020a), which rewards progression, penalizes collisions, and discourages abrupt steering. The RGB observation is a 300-degree panorama of 84×420 pixels, formed by concatenating 5 cameras on the vehicle's roof with 60-degree view each. The output action is a 2D vector of thrust (brake is negative thrust) and steering.

The agents are trained at "clear noon", and evaluated on a variety of dynamic weather and lighting conditions at noon and sunset (Fig. 1). For instance, the *wet* weathers feature roads with highly reflective spots. Averaged over 10 episodes per weather and 5 training runs, SECANT is able to drive **+47.7%** farther than prior SOTAs in tests.

Inference speed. The latency between observing and acting is critical for safe autonomous driving. Unlike SECANT, *Table 8.* iGibson object navigation. The goal is to find and navigate to a ceiling lamp in unseen rooms with novel decoration, furniture, and layouts (sample floor plan below). In testing, SECANT has **+15.8%** higher success rate (absolute percentage) than competing methods.

Setting	SECANT (Ours)	SAC	SAC+crop	DR	NetRand	SAC+IDM	PAD
Training	64.0 ± 3.7	$\mathbf{68.7 \pm 2.5}$	51.0 ± 12.0	49.6 ± 12.7	56.4 ± 3.8	54.2 ± 8.8	59.0 ± 13.4
Test: Easy	$\mathbf{56.8 \pm 17.2}$ (+17.6%)	13.8 ± 7.5	12.9 ± 7.1	17.6 ± 13.2	39.2 ± 11.7	25.9 ± 12.4	30.9 ± 12.4
Test: Hard	$\mathbf{47.7 \pm 11.3}$ (+13.9%)	9.3 ± 7.6	7.9 ± 5.3	15.2 ± 15.3	33.8 ± 11.8	12.7 ± 8.3	26.1 ± 23.0

PAD requires extra inference-time gradient computation. We benchmark both methods on actual hardware. Averaged over 1000 inference steps, SECANT is **65**× faster than PAD on Intel Xeon Gold 5220 (2.2 GHz) CPU, and **42**× faster on Nvidia RTX 2080Ti GPU.

5.4. iGibson: Indoor Object Navigation

iGibson (Xia et al., 2020; Shen et al., 2020) is an interactive simulator with highly realistic 3D rooms and furniture (Fig. 1). The goal is to navigate to a lamp as closely as possible. The reward function incentivizes the agent to maximize the proportion of pixels that the lamp occupies in view, and success is achieved when this proportion exceeds 5% over 10 consecutive steps. Our benchmark features 1 training room and 20 test rooms, which include distinct furniture, layout, and interior design from training. The lamp is gray in training, but has much richer textures in testing. We construct 2 difficulty levels with 10 rooms each, depending on the extent of visual shift. The agent is randomly spawned in a room with *only* RGB observation (168 × 168), and outputs a 2D vector of linear and angular velocities.

We evaluate on each test room for 20 episodes and report success rates in Table 8. SAC without augmentation is better than SAC+crop because the lamp can be cropped out accidentally, which interferes with the reward function. Therefore we use plain SAC as the expert for SECANT. We consider this an edge case, since random cropping is otherwise broadly applicable. SECANT achieves **+15.8%** higher success rate than prior methods in unseen rooms.

6. Conclusion

Zero-shot generalization in visual RL has been a long-standing challenge. We introduce SECANT, a novel technique that addresses policy optimization and robust representation learning *sequentially*. We demonstrate that SECANT significantly outperforms prior SOTA in 4 challenging domains with realistic test-time variations. We also systematically study different augmentation recipes, strategies, and distillation approaches. Compared to prior methods, we find that SECANT develops more robust visual representations and better task-specific saliency maps.

Acknowledgements

We are extremely grateful to Chris Choy, Jean Kossaifi, Shikun Liu, Zhiyuan "Jerry" Lin, Josiah Wong, Huaizu Jiang, Guanya Shi, Jacob Austin, Ismail Elezi, Ajay Mandlekar, Fei Xia, Agrim Gupta, Shyamal Buch, and many other colleagues for their helpful feedback and insightful discussions.

References

Andrychowicz, O. M., Baker, B., Chociej, M., Jozefowicz, R., McGrew, B., Pachocki, J., Petron, A., Plappert, M., Powell, G., Ray, A., et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

Argall, B. D., Chernova, S., Veloso, M., and Browning, B. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

Arora, H., Kumar, R., Krone, J., and Li, C. Multi-task learning for continuous control. *arXiv preprint arXiv:1802.01034*, 2018.

Aytar, Y., Pfaff, T., Budden, D., Paine, T., Wang, Z., and de Freitas, N. Playing hard exploration games by watching youtube. In *Advances in Neural Information Processing Systems*, pp. 2930–2941, 2018.

Beattie, C., Leibo, J. Z., Teplyashin, D., Ward, T., Wainwright, M., Küttler, H., Lefrancq, A., Green, S., Valdés, V., Sadik, A., Schrittwieser, J., Anderson, K., York, S., Cant, M., Cain, A., Bolton, A., Gaffney, S., King, H., Hassabis, D., Legg, S., and Petersen, S. Deepmind lab. *ArXiv*, abs/1612.03801, 2016.

Berthelot, D., Carlini, N., Cubuk, E. D., Kurakin, A., Sohn, K., Zhang, H., and Raffel, C. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*, 2019a.

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., and Raffel, C. Mixmatch: A holistic approach to semi-supervised learning. *arXiv preprint arXiv:1905.02249*, 2019b.

Chen, D., Zhou, B., Koltun, V., and Krähenbühl, P. Learning by cheating. In *Conference on Robot Learning*, pp. 66–75. PMLR, 2020.

Cobbe, K., Klimov, O., Hesse, C., Kim, T., and Schulman, J. Quantifying generalization in reinforcement learning. *arXiv preprint arXiv:1812.02341*, 2018.

Cobbe, K., Hesse, C., Hilton, J., and Schulman, J. Leveraging procedural generation to benchmark reinforcement learning. *arXiv preprint arXiv:1912.01588*, 2019.

Czarnecki, W. M., Pascanu, R., Osindero, S., Jayakumar, S. M., Swirszcz, G., and Jaderberg, M. Distilling policy distillation. *arXiv preprint arXiv:1902.02186*, 2019.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. Carla: An open urban driving simulator. *arXiv preprint arXiv:1711.03938*, 2017.

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

Ferns, N. and Precup, D. Bisimulation metrics are optimal value functions. In *UAI*, pp. 210–219. Citeseer, 2014.

Gamrian, S. and Goldberg, Y. Transfer learning for related reinforcement learning tasks via image-to-image translation. *ArXiv*, abs/1806.07377, 2019.

Greydanus, S., Koul, A., Dodge, J., and Fern, A. Visualizing and understanding atari agents. In *International Conference on Machine Learning*, pp. 1792–1801. PMLR, 2018.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *ICML*, 2018a.

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., and Levine, S. Soft actor-critic algorithms and applications, 2018b.

Hansen, N. and Wang, X. Generalization in reinforcement learning by soft data augmentation. *arXiv preprint arXiv:2011.13389*, 2020.

Hansen, N., Sun, Y., Abbeel, P., Efros, A. A., Pinto, L., and Wang, X. Self-supervised policy adaptation during deployment. *arXiv preprint arXiv:2007.04309*, 2020.

Hendrycks, D. and Dietterich, T. G. Benchmarking neural network robustness to common corruptions and surface variations. *arXiv: Learning*, 2018.

Hendrycks, D., Mu, N., Cubuk, E. D., Zoph, B., Gilmer, J., and Lakshminarayanan, B. Augmix: A simple data processing method to improve robustness and uncertainty. In *International Conference on Learning Representations*, 2019.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Ho, J. and Ermon, S. Generative adversarial imitation learning. *arXiv preprint arXiv:1606.03476*, 2016.

Igl, M., Ciosek, K., Li, Y., Tschiatschek, S., Zhang, C., Devlin, S., and Hofmann, K. Generalization in reinforcement learning with selective noise injection and information bottleneck. *Advances in Neural Information Processing Systems*, 32:13978–13990, 2019.

Igl, M., Farquhar, G., Luketina, J., Böhmer, W., and Whiteson, S. The impact of non-stationarity on generalisation in deep reinforcement learning. *ArXiv*, abs/2006.05826, 2020.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *ArXiv*, abs/1502.03167, 2015.

Jiang, M., Grefenstette, E., and Rocktäschel, T. Prioritized level replay. *arXiv preprint arXiv:2010.03934*, 2020.

Justesen, N., Torrado, R. R., Bontrager, P., Khalifa, A., Togelius, J., and Risi, S. Illuminating generalization in deep reinforcement learning through procedural level generation. *arXiv: Learning*, 2018.

Kostrikov, I., Yarats, D., and Fergus, R. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*, 2020.

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

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lee, J., Hwangbo, J., Wellhausen, L., Koltun, V., and Hutter, M. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47), 2020a.

Lee, K., Lee, K., Shin, J., and Lee, H. Network randomization: A simple technique for generalization in deep reinforcement learning. In *International Conference on Learning Representations. https://openreview. net/forum*, 2020b.

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.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Maaten, L. v. d. and Hinton, G. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov): 2579–2605, 2008.

Machado, M. C., Bellemare, M. G., Talvitie, E., Veness, J., Hausknecht, M. J., and Bowling, M. H. Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents. In *IJCAI*, 2018.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. A. Playing atari with deep reinforcement learning. *ArXiv*, abs/1312.5602, 2013.

Packer, C., Gao, K., Kos, J., Krähenbühl, P., Koltun, V., and Song, D. X. Assessing generalization in deep reinforcement learning. *ArXiv*, abs/1810.12282, 2018.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. Curiosity-driven exploration by self-supervised prediction. In *ICML*, 2017.

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.

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.

Raffin, A. Rl baselines zoo. https://github.com/araffin/rl-baselines-zoo, 2018.

Raileanu, R. and Rocktäschel, T. Ride: Rewarding impact-driven exploration for procedurally-generated environments. *ArXiv*, abs/2002.12292, 2020.

Raileanu, R., Goldstein, M., Yarats, D., Kostrikov, I., and Fergus, R. Automatic data augmentation for generalization in deep reinforcement learning. *arXiv preprint arXiv:2006.12862*, 2020.

Rajeswaran, A., Lowrey, K., Todorov, E., and Kakade, S. M. Towards generalization and simplicity in continuous control. *ArXiv*, abs/1703.02660, 2017.

Rao, K., Harris, C., Irpan, A., Levine, S., Ibarz, J., and Khansari, M. Rl-cyclegan: Reinforcement learning aware simulation-to-real. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11157–11166, 2020.

Ross, S., Gordon, G., and Bagnell, D. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635, 2011.

Rusu, A. A., Colmenarejo, S. G., Gulcehre, C., Desjardins, G., Kirkpatrick, J., Pascanu, R., Mnih, V., Kavukcuoglu, K., and Hadsell, R. Policy distillation. *arXiv preprint arXiv:1511.06295*, 2015.

Schaal, S. et al. Learning from demonstration. *Advances in neural information processing systems*, pp. 1040–1046, 1997.

Shen, B., Xia, F., Li, C., Martın-Martın, R., Fan, L., Wang, G., Buch, S., D'Arpino, C., Srivastava, S., Tchapmi, L. P., Vainio, K., Fei-Fei, L., and Savarese, S. igibson, a simulation environment for interactive tasks in large realistic scenes. *arXiv preprint*, 2020.

Sohn, K., Berthelot, D., Li, C.-L., Zhang, Z., Carlini, N., Cubuk, E. D., Kurakin, A., Zhang, H., and Raffel, C. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020.

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

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

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15:1929–1958, 2014.

Stooke, A., Lee, K., Abbeel, P., and Laskin, M. Decoupling representation learning from reinforcement learning. *arXiv preprint arXiv:2009.08319*, 2020.

Teh, Y., Bapst, V., Czarnecki, W. M., Quan, J., Kirkpatrick, J., Hadsell, R., Heess, N., and Pascanu, R. Distral: Robust multitask reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 4496–4506, 2017.

Tobin, J., Fong, R. H., 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)*, pp. 23–30, Sep 2017.

Wang, H., Zheng, S., Xiong, C., and Socher, R. On the generalization gap in reparameterizable reinforcement learning. In *ICML*, 2019.

Wang, J. X., Kurth-Nelson, Z., Tirumala, D., Soyer, H., Leibo, J. Z., Munos, R., Blundell, C., Kumaran, D., and Botvinick, M. Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*, 2016.

Wang, K., Kang, B., Shao, J., and Feng, J. Improving generalization in reinforcement learning with mixture regularization. *arXiv preprint arXiv:2010.10814*, 2020.

Xia, F., Shen, W. B., Li, C., Kasimbeg, P., Tchapmi, M. E., Toshev, A., Martín-Martín, R., and Savarese, S. Interactive gibson benchmark: A benchmark for interactive navigation in cluttered environments. *IEEE Robotics and Automation Letters*, 5(2):713–720, 2020.

Yang, J., Petersen, B., Zha, H., and Faissol, D. Single episode policy transfer in reinforcement learning, 2019.

Yarats, D., Zhang, A., Kostrikov, I., Amos, B., Pineau, J., and Fergus, R. Improving sample efficiency in model-free reinforcement learning from images. *ArXiv*, abs/1910.01741, 2019.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 6023–6032, 2019.

Zhang, A., Ballas, N., and Pineau, J. A dissection of overfitting and generalization in continuous reinforcement learning. *ArXiv*, abs/1806.07937, 2018.

Zhang, A., McAllister, R., Calandra, R., Gal, Y., and Levine, S. Learning invariant representations for reinforcement learning without reconstruction. *arXiv preprint arXiv:2006.10742*, 2020a.

Zhang, A., McAllister, R., Calandra, R., Gal, Y., and Levine, S. Learning invariant representations for reinforcement learning without reconstruction. *arXiv preprint arXiv:2006.10742*, 2020b.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Zhou, B., Kalra, N., and Krähenbühl, P. Domain adaptation through task distillation. In *European Conference on Computer Vision*, pp. 664–680. Springer, 2020.

Zhu, Y., Wong, J., Mandlekar, A., and Martín-Martín, R. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020.