Pre-Trained Image Encoder for Generalizable Visual Reinforcement Learning

Zhecheng Yuan[1], **Zhengrong Xue**[2], **Bo Yuan**[3], **Xueqian Wang**[1],
Yi Wu[1,4], **Yang Gao**[1,4], **Huazhe Xu**[1,4]

[1] Tsinghua University [2] Shanghai Jiao Tong University
[3] Qianyuan Institute of Sciences [4] Shanghai Qi Zhi Institute
yuanzc20@mails.tsinghua.edu.cn, xuhuazhe12@gmail.com

Abstract

Learning generalizable policies that can adapt to unseen environments remains challenging in visual Reinforcement Learning (RL). Existing approaches try to acquire a robust representation via diversifying the appearances of in-domain observations for better generalization. Limited by the specific observations of the environment, these methods ignore the possibility of exploring diverse real-world image datasets. In this paper, we investigate how a visual RL agent would benefit from the off-the-shelf visual representations. Surprisingly, we find that the early layers in an ImageNet pre-trained ResNet model could provide rather generalizable representations for visual RL. Hence, we propose **P**re-trained **I**mage **E**ncoder for **G**eneralizable visual reinforcement learning (PIE-G), a simple yet effective framework that can generalize to the unseen visual scenarios in a zero-shot manner. Extensive experiments are conducted on DMControl Generalization Benchmark, DMControl Manipulation Tasks, Drawer World, and CARLA to verify the effectiveness of PIE-G. Empirical evidence suggests PIE-G improves sample efficiency and significantly outperforms previous state-of-the-art methods in terms of generalization performance. In particular, PIE-G boasts a $\mathbf{55}\%$ generalization performance gain on average in the challenging video background setting. Project Page: https://sites.google.com/view/pie-g/home.

1 Introduction

Visual Reinforcement Learning (RL) has achieved significant success in learning complex behaviors directly from image observations [48, 35, 37]. Despite the progress, RL agents are often plagued by the overfitting problem [67], especially in high-dimensional observation space. Previous studies show that it is difficult for the visual agents to generalize to unseen scenarios [8, 40], which severely limits their deployment in real-world applications.

In general, visual RL methods rely on their encoders to learn a visual representation to perceive the world. Recent studies have found that data augmentation [65] leads to more generalizable representations so that the agents can adapt to the unseen environments with different visual appearances [58, 74]. However, most of those approaches only augment the observations of the training environments [37, 39, 68], which is unable to provide enough diversity for generalization over large domain gaps. Furthermore, naively applying data augmentation may damage the robustness of learned representations and decrease training sample efficiency [28, 83].

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

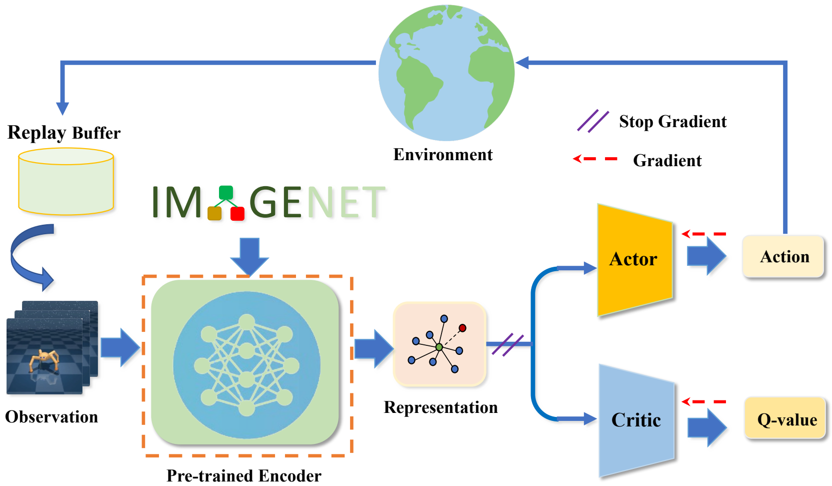


Figure 1: **Overview of PIE-G**. This figure shows the general framework of PIE-G where visual encoders embed high-dimensional images into low-dimensional representations for downstream decision-making tasks. Instead of training the encoder from scratch, PIE-G selects an ImageNet pre-trained ResNet model as the encoder and freezes its parameters during the entire training process.

To overcome these drawbacks, what we require is a universal representation that can generalize to a variety of unseen scenarios. Recent works in representation learning demonstrate promising results in enabling pre-trained models to provide strong priors for downstream tasks [31, 11]. The pre-trained models contain representations obtained from a wide range of existing real-world image datasets. These representations are proved to be robust to noises and capable of distinguishing salient features despite the diversity and the inconsistency [12]. Based on the observations, we would like to ask the following question: is it possible to train a visual RL agent that is augmented with pre-trained visual representations so that it can better generalize to novel tasks?

Towards answering the question, the main contribution of this paper is a surprising discovery that the off-the-shelf features of frozen models trained with ImageNet can be used as universal representations for visual RL. Based on such findings, we present **P**retrained **I**mage **E**ncoder for **G**eneralizable visual reinforcement learning (PIE-G), a visual RL framework that allows agents to obtain enhanced training efficiency and generalization ability via integrating the extracted representations from a pre-trained ResNet [30] encoder into RL training. Straightforward as the framework appears, PIE-G enjoys thoughtful details and nuanced design choices to acquire representations that are suitable for control and generalizable to novel scenarios. Specifically, we show that the choice of early layer features and the ever-updating Batch Normalization (BatchNorm) [33] are crucial for the performance gain.

To validate the effectiveness of our framework, we conduct a series of experiments on 4 benchmarks: DMControl Generalization Benchmark (DMC-GB) [26], DMControl Manipulation Tasks [72], Drawer World [75] that is modified from Meta World [82], and CARLA, a realistic autonomous driving simulator. Empirical studies have demonstrated that PIE-G achieves better or competitive results in training sample efficiency, and significantly outperforms previous state-of-the-art methods in generalization capability without bells and whistles.

Our main contributions are summarized as follows: (i) We find that pre-trained encoders from off-the-shelf image datasets with the early layer features and ever-updating BatchNorm provide generalizable representations in visual RL. (ii) We propose PIE-G, a simple yet effective framework with a pre-trained encoder that can boost the sample efficiency and generalization ability in visual RL. (iii) PIE-G outperforms state-of-the-art methods in 4 visual generalization benchmarks by a large margin, with a $\mathbf{55\%}$ boost on average on the hardest setting in DMC-GB.

2 Related Work

Representation learning in RL. A large corpus of literature has sought to leverage representation learning in the setting of RL [61, 22, 68, 60, 81, 69]. More recently, there are a branch of methods that perform unsupervised pre-training to encourage exploration for better sample efficiency [53, 44, 17, 43, 38, 29]. A typical approach is to use a contrastive learning method to jointly incentivize exploration and acquire useful representations [15]. Liu et al. [44] introduce a new type of pre-training techniques via entropy maximization in embedding space for better exploration. In particular, inspired by the new representation algorithm in computer vision, Yarats et al. [79] propose a SwAV-like architecture [4] for pre-training with an exploration scheme that maximizes the entropy of the state visitation distribution. However, all these methods require the data collected in the target environments, resulting in additional sample cost. We instead propose a cross-domain pre-training way to improve visual RL agents' performance without any in-domain interaction during the pre-training time.

Pre-trained visual encoders for RL. Applying the pre-trained vision model from other domains to the control tasks has gradually attracted researchers' attention [80, 66, 36, 62, 16, 51, 76]. For example, Shah et al. [63] and Parisi et al. [52] suggest that with the help of experts' demonstrations, the pre-trained ResNet [30] representations can achieve competitive performance with state-based inputs. Moreover, human video datasets are introduced to pre-train a visual representation for downstream policy learning [49]. The pre-trained model has also been proposed for goal-specification via behavior cloning [9]. However, few works explore the effectiveness of pre-trained models for generalization. In contrast to prior approaches, PIE-G enables agents to generalize well to the unseen visual scenarios with a large distributional shift in a zero-shot manner while achieving high sample efficiency in a standard RL training paradigm.

Generalization in visual RL. Researchers have investigated to improve visual RL agents' generalization ability from various aspects [1, 27, 40, 73, 2, 50, 77]. Data augmentation [40, 74, 28, 83, 18, 47] and domain randomization [71, 56, 54, 59, 5] are effective ways for generalization in visually different environments. Notably, Hansen et al. [26] employ a BYOL-like[24] architecture to decouple the augmentation from policy learning for better generalization. In order to control the high variance when implementing data augmentation, Hansen et al. [28] add a regularization term of the Q function between un-agumented and augmented data as an implicit variance reduction technique. Meanwhile, Yuan et al. [83] propose a task-aware data augmentation method with the Lipschitz constant [19] for maintaining training stability. Fan et al. [18] apply data augmentation in the imitation learning paradigm. Prior works rely on data augmentation to gain a robust representation for each task. In this work, we tackle this challenge from a different perspective, utilizing a single pre-trained encoder with the universal visual representations for all the tasks.

3 Preliminaries

Reinforcement learning. Due to partial state observability from images in visual RL [34], we consider learning in a Partially Observable Markov Decision Process (POMDP) [3] formulated by the tuple $\langle\mathcal{S}, \mathcal{O}, \mathcal{A}, r, \mathcal{P}, \gamma\rangle$ where $\mathcal{S}$ is the state space, $\mathcal{O}$ is the observation space, $\mathcal{A}$ is the action space, $r : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ is a reward function, $\mathcal{P}\left(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t\right)$ is the state transition function, and $\gamma \in [0, 1)$ is the discount factor. The goal is to find a policy π^* to maximize the expected cumulative return $\pi^* = \arg\max_\pi \mathbb{E}_{\mathbf{a}_t \sim \pi(\cdot|\mathbf{s}_t), \mathbf{s}_t \sim \mathcal{P}} \left[\sum_{t=1}^{T} \gamma^t r\left(\mathbf{s}_t, \mathbf{a}_t\right)\right]$, starting from an initial state $\mathbf{s}_0 \in \mathcal{S}$ and obtained by following the policy $\pi_\theta\left(\cdot \mid \mathbf{s}_t\right)$ which is parameterized by learnable parameters θ.

Generalization. In terms of generalization, we consider a set of POMDPs: $\mathbb{M} = \{\mathcal{M}_1, \mathcal{M}_2, ..., \mathcal{M}_n\}$ that shares the same dynamics and structures. The only difference among them is the observation space $\mathcal{O}$. This setting is more formally described as "Block MDPs" [14]. During the training process, we only have the access to a fixed POMDP denoted $\mathcal{M}_i$. Our purpose is to train an agent on a specific scenario $\mathcal{M}_i$ to learn a policy π_G^* which can maximize the expected cumulative return over the whole set of POMDPs in a zero-shot generalization manner.

4 Method

In this section, we introduce PIE-G, a simple yet effective framework for visual RL which benefits from the pre-trained encoders on other domains to facilitate sample efficiency and generalization ability.

4.1 Pre-trained Encoder

PIE-G explicitly leverages the pre-trained models as the representation extractor without any modification. The pre-trained encoder projects high-dimensional image observations into compact, low-dimensional embeddings that are later used by RL policies. Note that PIE-G is as simple as importing a pre-trained ResNet model from the *torchvision* [46] library. This avoids the design of any auxiliary tasks to acquire useful representations.

For all the training tasks on different benchmarks, the encoder's parameters are frozen to obtain universal visual representations. Since the pre-trained model contains the priors from a wide range of real-world images, we hypothesize that the inherited power from a pre-trained model may help to capture and distinguish the main components of different tasks' observations regardless of the changes of visual appearances or deformed shapes, and will further improve the sample efficiency and generalization abilities of RL agents.

To validate our hypothesis, we first encode each observation independently to obtain embeddings. Then, the embeddings from the second layer of the pre-trained model are fused as input features to the policy networks [64, 51]. Moreover, we enable BatchNorm [33] to keep updating the running mean and running standard deviation during the policy training. The key findings are: 1) early layers of a neural network would provide better representations for visual RL generalization, which resonates with prior works in imitation learning [52]; 2) the always updating statistics in BatchNorm helps better adapt to the shift in observation space and thus improve the generalization ability. More detailed discussion can be found in Section 5.4.

4.2 Reinforcement Learning Backbone

We implement DrQ-v2 [78] as the base visual reinforcement learning algorithm. DrQ-v2 is the state-of-the-art method for visual continuous control tasks, which adopts DDPG [41] coupling with clipped Double Q-learning [20] to alleviate the overestimation bias of target Q-value. The agents are trained with two Q_{θ_k} value functions and their corresponding target network $Q_{\bar{\theta}_k}$. The critic loss function is as follows, and the mini-batch of transitions $\tau = (\mathbf{s}_t, \mathbf{a}_t, r_{t:t+n-1}, \mathbf{s}_{t+n})$ is sampled from the replay buffer $\mathcal{D}$:

$$\mathcal{L}(\theta_k) = \mathbb{E}_{\tau\sim\mathcal{D}}\left[\left(Q_{\theta_k}\left(\mathbf{s}_t, \mathbf{a}_t\right) - y\right)^2\right] \quad \forall k \in \{1, 2\}, \tag{1}$$

with n-step TD target y:

$$y = \sum_{i=0}^{n-1} \gamma^i r_{t+i} + \gamma^n \min_{k=1,2} Q_{\bar{\theta}_k}\left(\mathbf{s}_{t+n}, \mathbf{a}_{t+n}\right),$$

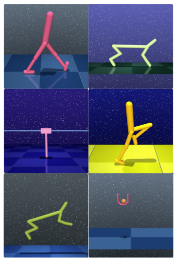
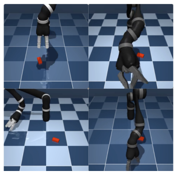
The actor π_ϕ is trained with the following objective:

$$\mathcal{L}_\phi(\mathcal{D}) = -\mathbb{E}_{\mathbf{s}_t\sim\mathcal{D}}\left[\min_{k=1,2} Q_{\theta_k}\left(\mathbf{s}_t, \mathbf{a}_t\right)\right], \tag{2}$$

where $\mathbf{s}_t$ is augmented by random shift, $\mathbf{a}_t = \pi_\phi(\mathbf{s}_t) + \epsilon$, ϵ is sampled from $\operatorname{clip}\left(\mathcal{N}\left(0, \sigma^2\right), -c, c\right)$ with a decaying exploration noise σ.

Thanks to the efficiency of DrQ-v2, PIE-G enjoys faster wall-clock training time and fewer computational footprints. We emphasize that PIE-G does not need any other proprioceptive states and sensory information as the inputs besides the representations extracted from original image observations. In previous works [37, 78, 58, 28], different schemes of data augmentation are proposed to improve sample efficiency and generalization performance. In practice, weak augmentation methods (e.g., random shift) of DrQ [37] and DrQ-v2 [78] are found to be beneficial for sample efficiency. In the setting of generalization, we follow the way of using strong augmentation methods (e.g., mixup) of SVEA [28] and DrAC [58] to further boost the performance. It is worth mentioning that since the gradient is stopped before it reaches the encoder, all the data augmentation techniques discussed here do not affect the pre-trained visual representation. Meanwhile, unlike Rutav et al. [63] and Simone et al. [52], we purely train the agent in a standard RL paradigm without any expert's demonstration.

Table 1: **Generalization on color-jittered observations.** Experiments are conducted on multiple tasks in the DMC-GB (*Top*) and Manipulation Tasks (*Bottom*) environments with varying color backgrounds. For a certain task, the color of the setting in evaluation will be altered. The agent is required to adapt to the changes in a zero-shot manner. Compared with its counterparts, PIE-G gains comparable and better performance in **9** out of **10** settings.

Setting	DMControl Tasks	SAC	DrQ	DrQ-v2	SVEA	TLDA	**PIE-G**
	Cartpole, Swingup	248±24	586±52	277±80	**837±23**	760±60	749±46
	Walker, Stand	365±79	770±71	413±61	942±26	947±26	**960±15**
	Walker, Walk	144±19	520±91	168±90	760±145	823±58	**884±20**
	Ball_in_cup, Catch	151±36	365±210	469±99	**961±7**	932±32	**964±7**
	Cheetah, Run	133±26	100±27	109±45	273±23	**371±51**	**369±53**
	Manipulation, Training	2.5±1.8	130±20	**204±11**	49±48	124±32	**199±13**
	Modified, Arm	0.3±0.5	68±20	29±9	21±25	55±21	**122±30**
	Modified, Platform	0.5±0.3	0.8±1.3	1.5±1.7	24±25	89±40	**96±23**
	Modified, Both	0.4±0.8	1.0±2.0	0.8±1.5	13±14	36±25	**44±16**

5 Experiments

In this section, we investigate the following questions: (1) Can PIE-G improve the agent's generalization ability? Specifically, how well does PIE-G deal with jittered color, moving video background, and deformed shapes of robots? (2) Can PIE-G improve training sample efficiency? (3) How do the choice of layers in the encoder and the use of BatchNorm [33] affect the performance? (4) Can further finetuned RL agents outperform those with frozen visual encoders?

5.1 Setup

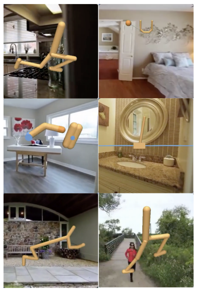
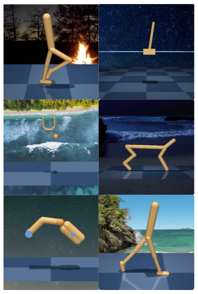
We evaluate our method on a wide range of tasks, including DMControl Generalization Benchmark (DMC-GB) [26], DeepMind Manipulation tasks [72], and Drawer World [75]. PIE-G is trained for 500k interaction steps with 2 action repeat and evaluated with 100 episodes for every task on the testing benchmarks. All the generalization evaluations are in a zero-shot manner. By default, the encoder uses the ResNet18 architecture [30]. To be more specific, the feature maps of the second layer are flattened and passed through an additional fully connected layer to serve as the representations of the observations. More training details and environment descriptions are in Appendix B.

5.2 Evaluation on Generalization Ability

We compare the generalization ability of PIE-G with state-of-the-art methods and strong baselines: **SAC** [25]: a widely used off-policy RL algorithm; **DrQ** [37]: a SAC-based visual RL algorithm with augmented observations; **DrQ-v2** [78]: the prior state-of-the-art model-free visual RL algorithm in terms of sample efficiency; **SVEA** [28]: the prior state-of-the-art method in terms of generalization via reducing the Q-variance through an auxiliary loss; **TLDA** [83]: another state-of-the-art method in generalization by using task-aware data augmentation.

Generalization on color-jittered observations. The agent's generalization ability is evaluated on DMC-GB with randomly jittered color. For the manipulation tasks, the colors of different objects (e.g., floors, arms) are modified. As shown in Table 1, PIE-G obtains better or competitive performance in **9** out of **10** instances. These results suggest that the visual representation from the pre-trained model is more robust to the color-changing than the one trained by standard RL algorithms.

Table 2: **Generalization on unseen moving backgrounds.** Episode return in two types of unseen dynamic video background environments, i.e., *video easy* (*Bottom*) and *video hard* (*Top*). PIE-G achieves competitive or better performance in **9** out of **12** tasks. In *video hard* setting, we significantly outperforms other algorithms with **+55%** improvement on average.

Setting	DMControl Tasks	DrQ	DrQ-v2	SVEA	TLDA	**PIE-G**
	Cartpole, Swingup	138±9	130±3	393±45	286±47	**401**±**21** (+2.0%)
	Walker, Stand	289±49	151±13	834±46	602±51	**852**±**56** (+2.2%)
	Walker. Walk	104±22	34±11	377±93	271±55	**600**±**28** (+59.2%)
	Ball_in_cup, Catch	92±23	97±27	403±174	257±57	**786**±**47** (+95.0%)
	Cheetah, Run	32±13	23±5	105±37	90±27	**154**±**17** (+46.6%)
	Finger, Spin	71±45	21±4	335±58	241±29	**762**±**59** (+127%)
	Cartpole, Swingup	485±105	267±41	**782**±**27**	671±57	587±61
	Walker, Stand	873±83	560±48	961±8	**973**±**6**	957±12
	Walker. Walk	682±89	175±117	819±71	**873**±**34**	**871**±**22**
	Ball_in_cup, Catch	318±157	454±60	871±106	892±68	**922**±**20**
	Cheetah, Run	102±30	64±22	249±20	**366**±**57**	287±20
	Finger, Spin	533±119	456±15	808±33	744±18	**837**±**107**

Generalization on unseen and/or moving backgrounds. We then evaluate PIE-G on the more challenging settings: *video easy* and *video hard* in DMC-GB. The *video hard* setting consists of more complicated and fast-switching video backgrounds that are drastically different from the training environments. Notably, even the reference plane of the ground is removed in this setting.

The comparison results are shown in Table 2. PIE-G achieves better or comparable performance with the prior state-of-the-art methods in **9** out of **12** instances. In particular, PIE-G gains significant improvement in the *video hard* setting over all the previous methods with **+55%** improvement on average. For example, in the *Finger Spin*, *Cup Catch*, and *Walker Walk* tasks, PIE-G outperforms the best of the other methods by substantial margins **127.0%**, **95.0%**, and **59.2%** respectively.

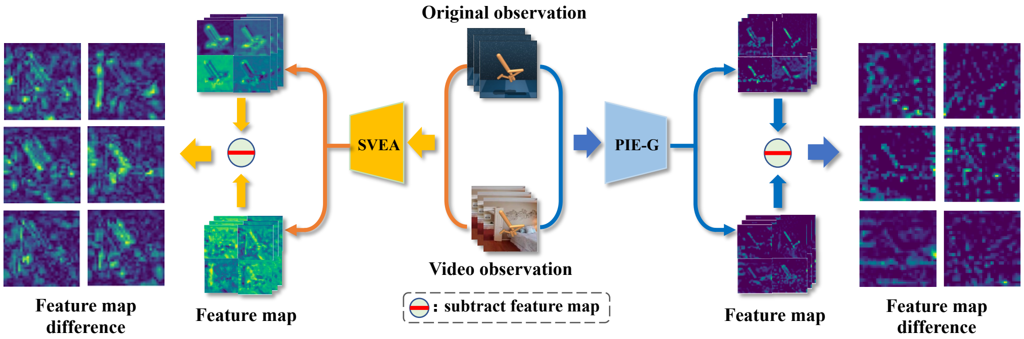
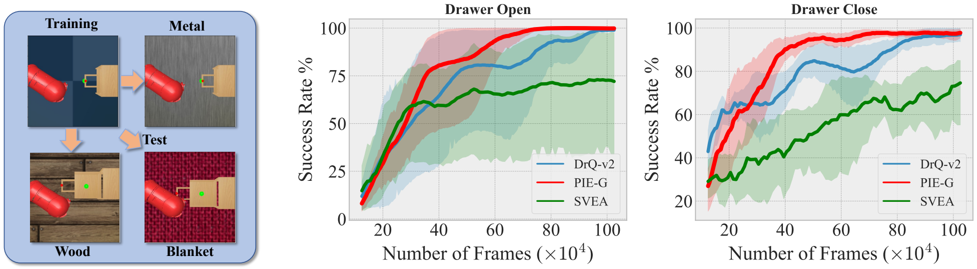


Figure 2: **Visualized feature map differences of two inputs from the same state with different backgrounds.** The difference of the feature maps with PIE-G as the encoder is closer to zero than that with SVEA, indicating PIE-G enjoys better generalization ability.



(a) **Meta World**

(b) **Training Curve**

Figure 4: **Training on Meta World.** *Left*: The visualization of Meta World with different textures. *Right*: The training curves. PIE-G *(Red line)* demonstrates better sample efficiency than DrQv2 *(Blue line)* and SVEA *(Green line)*.

Attempting to explain the success, we visualize the difference of the normalized feature maps extracted from the encoder whose inputs are two Walkers of the same pose but with different backgrounds, as is shown in Figure 2. Ideally, a well-generalizable encoder would map the observations of the two Walkers to exactly the same embedding, and therefore the difference should be zero. In practice, as shown in Figure 2, the encoder of PIE-G produces a difference much closer to zero than that of SVEA. Numerically, we calculate the average pixel intensity in the difference of normalized feature maps, and the intensity is decreased by 50.9% with PIE-G than that with SVEA.

Then, we evaluate PIE-G on the CARLA [13] autonomous driving system which contains realistic observations and complex driving scenarios. The default setting is adopted from Zhang et al. [85]. PIE-G and other algorithms are benchmarked on 4 diverse weather environments. As shown in Figure 3, all the prior state-of-the-art methods cannot adapt to the new unseen weather with different lighting, humidity, road conditions etc. The behind reason is that compared to DMC-GB whose images merely consist of a single control agent and the background, the observations of CARLA contain more distracting objects and factors; therefore, only depending on data augmentation to provide diverse data cannot tackle this complicated visual driving task. The experimental results in Figure 3 exhibits that thanks to the ImageNet pre-trained encoder, PIE-G can generalize well on the complicated scenes without large performance drop.

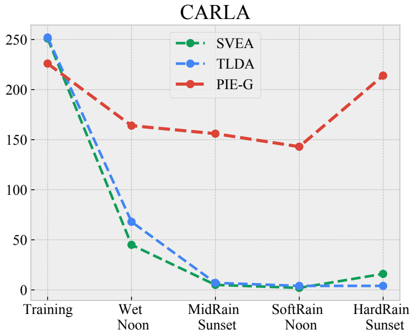


Figure 3: **Performance on CARLA.** PIE-G can well adapt to unseen scenarios.

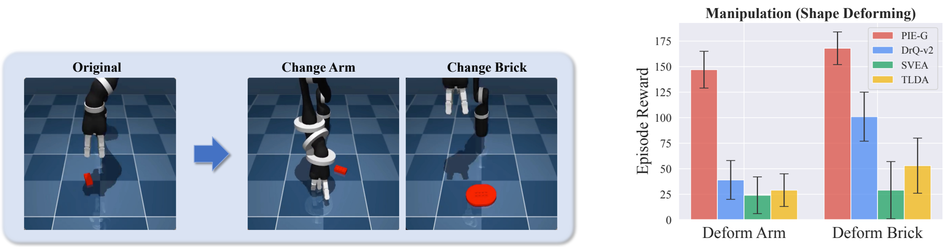
Furthermore, we conduct experiments on the *Drawer World* benchmark to test the agent's generalization ability in manipulation tasks with different background textures. The visualized observations are shown in Figure 4a. *Success Rate* is adopted as the evaluation metric for its goal-conditioned nature. Table 3 illustrates that PIE-G can achieve better or comparable generalization performance in all the settings with **+24%** boost on average while other approaches may suffer from the CNN's sensitivity in the face of various textures [21].

Task	Setting	SAC	DrQ-v2	SVEA	**PIE-G**
Drawer-Close	Training	**100%**	**98%**	70%	**99%**
	Wood	0%	32%	49%	**59%**
	Metal	0%	46%	69%	**95%**
	Blanket	0%	8%	**72%**	**71%**

Task	Setting	SAC	DrQ-v2	SVEA	**PIE-G**
Drawer-Open	Training	**98%**	**100%**	75%	**97%**
	Wood	18%	2%	47%	**79%**
	Metal	35%	53%	71%	**97%**
	Blanket	28%	5%	37%	**85%**

Table 3: **Generalization on Drawer World.** Evaluation on distracting textures. PIE-G is robust to the texture changing.

Generalization on deformed shapes. To verify agent's robustness in terms of the deformed shapes, we modify the shapes of the jaco arm and the target objects in the manipulation tasks, as shown in Figure 5a. Figure 5b demonstrates that PIE-G also improves the agents' generalization ability with various shapes while other methods could barely generalize to these changes. We attribute this to the lack of shape changing in previous data augmentation techniques. Conversely, our pre-trained encoder is learned from a multitude of real-world images with various poses and shapes, thus enhancing its generalization ability on deformed shapes.



(a) **Deform the shape of the object** (b) **Generalization performance**

Figure 5: **Deforming the shape.** *Left*: Aiming at evaluating the agent's robustness of the shape, we deform the robot's arm and the target brick. *Right*: The results demonstrate that PIE-G is well-generalizable in the face of deformed shapes.

5.3 Evaluation on Sample Efficiency

We evaluate the sample efficiency of PIE-G on 8 relatively challenging tasks on the DeepMind Control Suite and the Manipulation task (*Reach Duplo*). Figure 6 demonstrates that PIE-G achieves better or comparable sample efficiency and asymptotic performance than DrQ-v2 in **7** out of **8** tasks. To eliminate the effect of varied network size, we also include *random enc*, a baseline that has the same network architecture with frozen random initialized parameters. In addition, Figure 4b shows that PIE-G is also superior over the baselines on the Drawer world benchmark. The results demonstrate that the pre-trained encoder inherits the powerful feature extraction ability trained from ImageNet and acquires better training sample efficiency in control tasks.

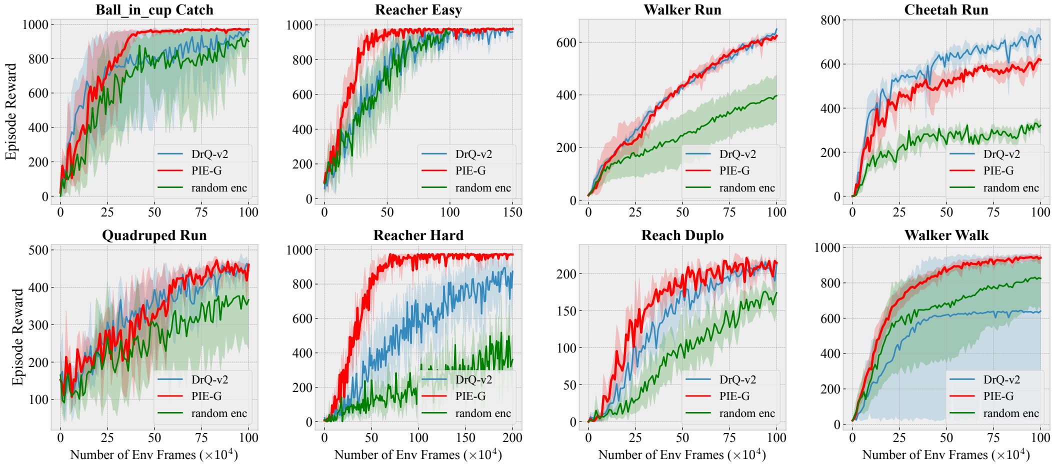


Figure 6: **Training sample efficiency**. Average episode rewards on 8 challenging DMControl tasks with means and standard deviations calculated over 5 seeds. We compare PIE-G (*Red line*) with DrQ-v2 (*Blue line*) and random enc (*Green line*) with respect to sample efficiency. Our method achieves better or comparable performance in **7** out of **8** instances.

5.4 Ablation study

To verify the necessity of the design choices in PIE-G, we conduct a series of ablation studies to take a closer look at the proposed method. More results are shown in Appendix C.

Task	Layer 1	Layer 2	Layer 3	Layer 4
Walker Walk	840±32	**884±20**	845±27	306±31
Cheetah Run	**366±56**	**369±53**	294±60	111±19
Walker Stand	953±8	**964±7**	**957±7**	625±116

Table 4: **Different layers.** We employ the feature map of different layers of a ResNet model as the visual representation. Among them, the Layer 2 exhibits the best generalization performance.

Choice of layers. In convolutional neural networks, the later layers capture high-level semantic features, while the early layers are responsible for extracting low-level information [45, 84, 42]. Figure 7 and Table 4 investigate how much control tasks can benefit from the features extracted from different layers. As shown in Figure 7, the early layers preserve rich details of edges and corners, while the later layers only provide very abstract information. Intuitively, for control tasks, a trade-off is required between low-level details and high-level semantics. Table 4 and Figure 8 in Appendix show that the Layer 2 gains better generalization and sample efficiency performance than the other layers.

Batch normalization. Batch Normalization (BatchNorm) [33] is a popular technique in computer vision. However, it is not widely adopted in RL algorithms. In contrast to conventional wisdom, BatchNorm is found to be useful and important in PIE-G. Specifically, we find that calculating the mean and variance of the observations during evaluation rather than using the statistics from training data would boost the performance. Figure 9 demonstrates that, in the most challenging settings, PIE-G with the use of BatchNorm can further improve the generalization performance. This is largely because the distribution of observations is determined by the agent, violating the assumption of independent and identical distribution (i.i.d.). This use of BatchNorm also reassures the recommendation from Ioffe et al. [33] that recomputation of the statistical means and variances allows the BatchNorm layer to generalize to new data distributions.

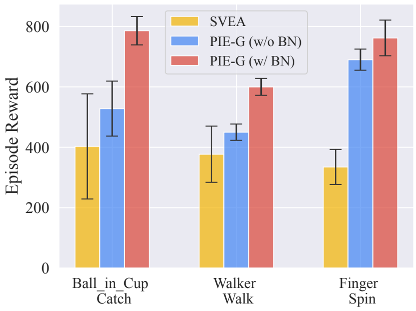


Figure 9: **Leveraging BatchNorm.** The ever-updating BatchNorm is beneficial for better performance.

Tasks	ImageNet	CLIP	Ego4D	SVEA
Walker Walk	600 ±28	615 ±30	441 ±15	377 ±93
Cheetah Run	154 ±17	115 ±62	101 ±13	105 ±37
Walker Stand	852 ±56	849 ±23	647 ±59	441 ±15
Finger Spin	762 ±59	676 ±116	515 ±104	335 ±58

Table 5: **Adopting other datasets for pre-training.** All agents pre-trained with different datasets gain considerable generalization performance.

Adopting other datasets for pre-training. Besides ImageNet [10], we also implement pre-trained visual encoders with other novel and popular datasets: CLIP [57] and Ego4D [23]. CLIP (Contrastive Language–Image Pre-training) trained a large number of (image, text) pairs collected from Internet to jointly acquire visual and text representations. The Ego4D is a egocentric human video dataset which contains massive daily-life activity videos in hundreds of scenarios. Table 5 shows that the agents pre-trained with CLIP achieves comparable performance with those pre-trained with ImageNet. Since the Ego4D collects the videos with the *first-person* view, the view difference between the tasks and the dataset leads to a decrease in performance; nevertheless, the Ego4D pre-trained agents still obtain comparable results with the prior state-of-the-art methods.

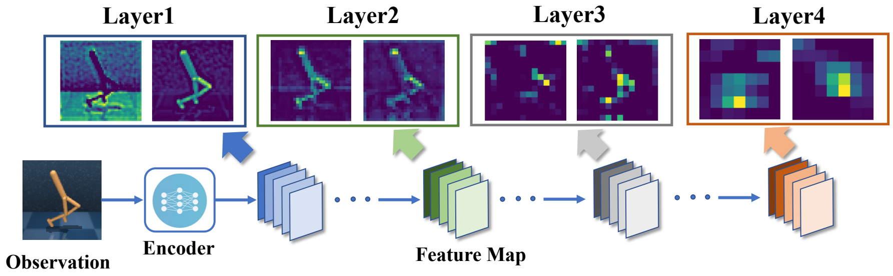


Figure 7: **Visualization of the feature maps of different layers .** The feature map of Layer 2 largely preserves the outline of the Walker that is advantageous to the control tasks, and at the same time discards redundant details.

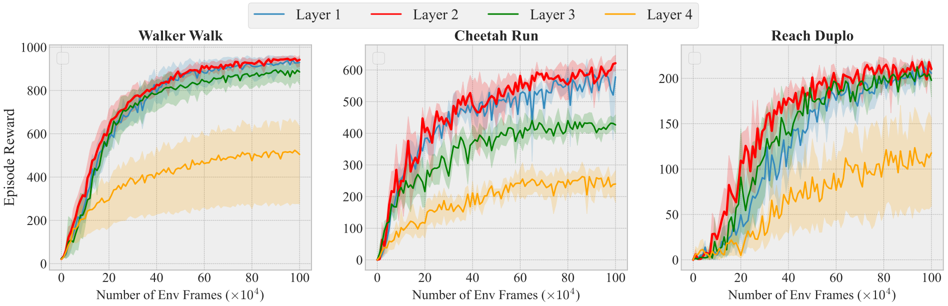


Figure 8: **Choice of layers in terms of sample efficiency.** This figure indicates that the early layers have better sample efficiency than the later layers.

Finetuning the pre-trained model. We also conduct research to finetune the encoder's parameters instead of keeping it frozen. Previous works [32, 55] have found that finetuning pre-trained models is challenging. Consistent with these studies, Table 6 suggests that compared with the frozen representations from pre-trained models, the finetuned representations suffer from the out-of-distribution problem [6] and lead to a performance drop in terms of the generalization ability.

Task	PIE-G (Finetune)	PIE-G (Frozen)
Walker Walk	455±67	**600±28**
Cheetah Run	122±15	**150±19**
Walker Stand	771±25	**852±56**

Table 6: **Finetuning the pre-trained models.** We compare the generalization performance between the frozen visual representations and the finetuned ones.

Adopting other pre-trained models. Additionally, we investigate the efficacy of other visual representations. MoCo-v2 [7] is a pre-trained model optimized via contrastive learning to learn representations. We find that PIE-G with the pre-trained representations of MoCo-v2 can also obtain a comparable performance in terms of both the sample efficiency and generalization ability. More results are shown in Appendix C.

Task	PIE-G	PIE-G (w / MoCo-v2)
Walker Walk	600±28	585±30
Cheetah Run	154±17	150±19
Walker Stand	852±56	856±51

Table 7: **Adopting other pre-trained models.** PIE-G with MoCo-v2 can obtain comparable generalization performance.

6 Conclusion

In this work, we propose PIE-G, a simple yet effective framework that leverages off-the-shelf features of ImageNet pre-trained ResNet models for better generalization in visual RL. Extensive experiments on a variety of tasks in four RL environments confirm the merits of universal visual representations, which endow the agents with improved sample efficiency and better generalization performance. In addition, we show that the choice of layers and the use of BatchNorm are crucial for the performance gain. Our exploration may inspire more researchers to dig into the great potential of utilizing pre-trained representations in visual RL.

Limitations. We study generalization in the simulated environments. However, there might be new challenges in the real-world applications. In the future, we would like to establish and test on benchmarks of real-world scenarios.

Acknowledgements: We would like to thank Boyuan Chen, Kaizhe Hu and Guozheng Ma for giving helpful suggestions. We are sincerely grateful to the anonymous reviewers for taking the time to review this work and giving insightful advice. Yang Gao is supported by the Ministry of Science and Technology of the People´s Republic of China, the 2030 Innovation Megaprojects "Program on New Generation Artificial Intelligence" (Grant No. 2021AAA0150000). Yang Gao is also supported by a grant from the Guoqiang Institute, Tsinghua University.

References

[1] Rishabh Agarwal, Marlos C Machado, Pablo Samuel Castro, and Marc G Bellemare. Contrastive behavioral similarity embeddings for generalization in reinforcement learning. *arXiv preprint arXiv:2101.05265*, 2021.

[2] Charles Beattie, Joel Z Leibo, Denis Teplyashin, Tom Ward, Marcus Wainwright, Heinrich Küttler, Andrew Lefrancq, Simon Green, Víctor Valdés, Amir Sadik, et al. Deepmind lab. *arXiv preprint arXiv:1612.03801*, 2016.

[3] Richard Bellman. A markovian decision process. *Journal of mathematics and mechanics*, pages 679–684, 1957.

[4] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020.

[5] Yevgen Chebotar, Ankur Handa, Viktor Makoviychuk, Miles Macklin, Jan Issac, Nathan Ratliff, and Dieter Fox. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8973–8979. IEEE, 2019.

[6] Bryan Chen, Alexander Sax, Gene Lewis, Iro Armeni, Silvio Savarese, Amir Zamir, Jitendra Malik, and Lerrel Pinto. Robust policies via mid-level visual representations: An experimental study in manipulation and navigation. *arXiv preprint arXiv:2011.06698*, 2020.

[7] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[8] Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. Quantifying generalization in reinforcement learning. In *International Conference on Machine Learning*, pages 1282–1289. PMLR, 2019.

[9] Yuchen Cui, Scott Niekum, Abhinav Gupta, Vikash Kumar, and Aravind Rajeswaran. Can foundation models perform zero-shot task specification for robot manipulation? *arXiv preprint arXiv:2204.11134*, 2022.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[12] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655. PMLR, 2014.

[13] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.

[14] Simon Du, Akshay Krishnamurthy, Nan Jiang, Alekh Agarwal, Miroslav Dudik, and John Langford. Provably efficient rl with rich observations via latent state decoding. In *International Conference on Machine Learning*, pages 1665–1674. PMLR, 2019.

[15] Yilun Du, Chuang Gan, and Phillip Isola. Curious representation learning for embodied intelligence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10408–10417, 2021.

[16] Frederik Ebert, Yanlai Yang, Karl Schmeckpeper, Bernadette Bucher, Georgios Georgakis, Kostas Daniilidis, Chelsea Finn, and Sergey Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. *arXiv preprint arXiv:2109.13396*, 2021.

[17] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

[18] Linxi Fan, Guanzhi Wang, De-An Huang, Zhiding Yu, Li Fei-Fei, Yuke Zhu, and Animashree Anandkumar. Secant: Self-expert cloning for zero-shot generalization of visual policies. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 3088–3099. PMLR, 18–24 Jul 2021.

[19] Chris Finlay, Jeff Calder, Bilal Abbasi, and Adam Oberman. Lipschitz regularized deep neural networks generalize and are adversarially robust. *arXiv preprint arXiv:1808.09540*, 2018.

[20] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.

[21] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

[22] Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G Bellemare. Deepmdp: Learning continuous latent space models for representation learning. In *International Conference on Machine Learning*, pages 2170–2179. PMLR, 2019.

[23] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022.

[24] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020.

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

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

[27] Nicklas Hansen, Rishabh Jangir, Yu Sun, Guillem Alenyà, Pieter Abbeel, Alexei A Efros, Lerrel Pinto, and Xiaolong Wang. Self-supervised policy adaptation during deployment. *arXiv preprint arXiv:2007.04309*, 2020.

[28] Nicklas Hansen, Hao Su, and Xiaolong Wang. Stabilizing deep q-learning with convnets and vision transformers under data augmentation. *Advances in Neural Information Processing Systems*, 34, 2021.

[29] Steven Hansen, Will Dabney, Andre Barreto, Tom Van de Wiele, David Warde-Farley, and Volodymyr Mnih. Fast task inference with variational intrinsic successor features. *arXiv preprint arXiv:1906.05030*, 2019.

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[31] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[32] Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *International Conference on Machine Learning*, pages 4182–4192. PMLR, 2020.

[33] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[34] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

[35] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on Robot Learning*, pages 651–673. PMLR, 2018.

[36] Apoorv Khandelwal, Luca Weihs, Roozbeh Mottaghi, and Aniruddha Kembhavi. Simple but effective: Clip embeddings for embodied ai. *arXiv preprint arXiv:2111.09888*, 2021.

[37] Ilya Kostrikov, Denis Yarats, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*, 2020.

[38] Michael Laskin, Hao Liu, Xue Bin Peng, Denis Yarats, Aravind Rajeswaran, and Pieter Abbeel. Cic: Contrastive intrinsic control for unsupervised skill discovery. *arXiv preprint arXiv:2202.00161*, 2022.

[39] Misha Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. *Advances in Neural Information Processing Systems*, 33:19884–19895, 2020.

[40] Kimin Lee, Kibok Lee, Jinwoo Shin, and Honglak Lee. Network randomization: A simple technique for generalization in deep reinforcement learning. *arXiv preprint arXiv:1910.05396*, 2019.

[41] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[42] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[43] Hao Liu and Pieter Abbeel. Aps: Active pretraining with successor features. In *International Conference on Machine Learning*, pages 6736–6747. PMLR, 2021.

[44] Hao Liu and Pieter Abbeel. Behavior from the void: Unsupervised active pre-training. *Advances in Neural Information Processing Systems*, 34, 2021.

[45] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang. Hierarchical convolutional features for visual tracking. In *Proceedings of the IEEE international conference on computer vision*, pages 3074–3082, 2015.

[46] Sébastien Marcel and Yann Rodriguez. Torchvision the machine-vision package of torch. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1485–1488, 2010.

[47] Berenson Dmitry Mitrano Peter. Data augmentation for manipulation. *arXiv preprint arXiv:2205.02886*, 2022.

[48] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[49] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

[50] Charles Packer, Katelyn Gao, Jernej Kos, Philipp Krähenbühl, Vladlen Koltun, and Dawn Song. Assessing generalization in deep reinforcement learning. *arXiv preprint arXiv:1810.12282*, 2018.

[51] Jyothish Pari, Nur Muhammad, Sridhar Pandian Arunachalam, Lerrel Pinto, et al. The surprising effectiveness of representation learning for visual imitation. *arXiv preprint arXiv:2112.01511*, 2021.

[52] Simone Parisi, Aravind Rajeswaran, Senthil Purushwalkam, and Abhinav Gupta. The unsurprising effectiveness of pre-trained vision models for control. *arXiv preprint arXiv:2203.03580*, 2022.

[53] Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement. In *International conference on machine learning*, pages 5062–5071. PMLR, 2019.

[54] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 3803–3810. IEEE, 2018.

[55] Matthew E Peters, Sebastian Ruder, and Noah A Smith. To tune or not to tune? adapting pretrained representations to diverse tasks. *arXiv preprint arXiv:1903.05987*, 2019.

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

[57] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[58] Roberta Raileanu, Maxwell Goldstein, Denis Yarats, Ilya Kostrikov, and Rob Fergus. Automatic data augmentation for generalization in reinforcement learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[59] Fabio Ramos, Rafael Carvalhaes Possas, and Dieter Fox. Bayessim: adaptive domain randomization via probabilistic inference for robotics simulators. *arXiv preprint arXiv:1906.01728*, 2019.

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

[61] Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8583–8592. PMLR, 13–18 Jul 2020.

[62] Younggyo Seo, Kimin Lee, Stephen James, and Pieter Abbeel. Reinforcement learning with action-free pre-training from videos. *arXiv preprint arXiv:2203.13880*, 2022.

[63] Rutav Shah and Vikash Kumar. Rrl: Resnet as representation for reinforcement learning. *arXiv preprint arXiv:2107.03380*, 2021.

[64] Wenling Shang, Xiaofei Wang, Aravind Srinivas, Aravind Rajeswaran, Yang Gao, Pieter Abbeel, and Misha Laskin. Reinforcement learning with latent flow. *Advances in Neural Information Processing Systems*, 34, 2021.

[65] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[66] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on Robot Learning*, pages 894–906. PMLR, 2022.

[67] Xingyou Song, Yiding Jiang, Stephen Tu, Yilun Du, and Behnam Neyshabur. Observational overfitting in reinforcement learning. *arXiv preprint arXiv:1912.02975*, 2019.

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

[69] Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling representation learning from reinforcement learning. In *International Conference on Machine Learning*, pages 9870–9879. PMLR, 2021.

[70] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

[71] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017.

[72] Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy Lillicrap, Nicolas Heess, and Yuval Tassa. dm_control: Software and tasks for continuous control. *Software Impacts*, 6:100022, 2020.

[73] Jane X Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Remi Munos, Charles Blundell, Dharshan Kumaran, and Matt Botvinick. Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*, 2016.

[74] Kaixin Wang, Bingyi Kang, Jie Shao, and Jiashi Feng. Improving generalization in reinforcement learning with mixture regularization. *Advances in Neural Information Processing Systems*, 33:7968–7978, 2020.

[75] Xudong Wang, Long Lian, and Stella X Yu. Unsupervised visual attention and invariance for reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6677–6687, 2021.

[76] Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022.

[77] Huazhe Xu, Boyuan Chen, Yang Gao, and Trevor Darrell. Zero-shot policy learning with spatial temporal reward decomposition on contingency-aware observation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10786–10792. IEEE, 2021.

[78] Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. *arXiv preprint arXiv:2107.09645*, 2021.

[79] Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Reinforcement learning with prototypical representations. In *International Conference on Machine Learning*, pages 11920–11931. PMLR, 2021.

[80] Lin Yen-Chen, Andy Zeng, Shuran Song, Phillip Isola, and Tsung-Yi Lin. Learning to see before learning to act: Visual pre-training for manipulation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7286–7293. IEEE, 2020.

[81] Tao Yu, Cuiling Lan, Wenjun Zeng, Mingxiao Feng, Zhizheng Zhang, and Zhibo Chen. Playvirtual: Augmenting cycle-consistent virtual trajectories for reinforcement learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[82] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning*, pages 1094–1100. PMLR, 2020.

[83] Zhecheng Yuan, Guozheng Ma, Yao Mu, Bo Xia, Bo Yuan, Xueqian Wang, Ping Luo, and Huazhe Xu. Don't touch what matters: Task-aware lipschitz data augmentation for visual reinforcement learning. *arXiv preprint arXiv:2202.09982*, 2022.

[84] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[85] Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. *arXiv preprint arXiv:2006.10742*, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] Our work focus on designing an algorithm for boosting the generalization ability, rather than real-world application.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] In the supplementary material
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] In the supplementary material
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] In the experiments
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] In the supplementary material
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Environment Details

DeepMind control suite. DMControl Suite [70] is a widely used benchmark, which contains a variety of continuous control tasks. For generalization evaluation, we test methods on the DMControl Generalization Benchmark (DMC-GB) [26] that is developed based on DMControl Suite. DMC-GB provides different levels of difficulty in terms of generalization performance for visual RL. Visualized observations are in the *Setting* column of Table 1 (*Top*) and Table 2.

DeepMind control manipulation tasks. DeepMind Control [72] contains dexterous manipulation tasks with a multi-joint Jaco arm and snap-together bricks. In this paper, we modify the colors and shapes of the arms and the bricks in the task of *Reach Duplo* to test the agents' generalization ability. Visualized observations are shown in the *Setting* column of Table 1 (*Bottom*).

Drawer world benchmarks. Meta-world [82] contains a series of vision-based robotic manipulation tasks. Wang et al. [75] propose a variant of Meta-world, Drawer World, with a variety of realistic textures to evaluate the generalization ability of the agent. These tasks require a Sawyer robot arm to open or close a drawer, respectively. The visualizations of the environment are shown in Figure 4a.

CARLA autonomous driving. CARLA [13] is a realistic simulator for autonomous driving. Many recent works utilize this challenging benchmark in visual RL setting. The trained agents are evaluated on different weather and road conditions.

B Implementation Details

In this section, we provide PIE-G's detailed settings. As shown in Table 8, we set up our hyperparmeters and environmental details in three benchmarks. Our method is trained for 500k interaction steps (1000k environment steps with 2 action repeat). All experiments are run with a single GeForce GTX 3090 GPU and AMD EPYC 7H12 64-Core Processor CPU. All code assets used for this project came with MIT licenses. Code: `https://anonymous.4open.science/r/PIE-G-EF75/`

Table 8: Hyperparameter of PIE-G in 4 benchmarks.

Hyperparameter	DMControl-GB	Drawer World	Manipulation Tasks	CARLA
Input size	84×84	84×84	84×84	84×84
Discount factor γ	0.99	0.99	0.99	0.99
Action repeat	8 (cartpole) 2 (otherwise)	4	2	2
Frame stack	3	3	3	3
Learning rate	1e-4	5e-5	1e-4	1e-4
Random shifting padding	4	4	4	4
Training step	500k	250k	500k	500k
Evaluation episodes	100	100	100	50
Optimizer	Adam	Adam	Adam	Adam

Vision models. In this paper, we use the ready-made models from the following link: ResNet: `https://github.com/pytorch/vision`; Moco: `https://github.com/facebookresearch/moco` (v2 version trained with 800 epochs); CLIP: `https://github.com/OpenAI/CLIP`; R3M: `https://github.com/facebookresearch/r3m`.

In terms of implementing data augmentation, We choose *random overlay* (integrate a distract image $\mathcal{I}$ with the observation o linearly, $o' = \alpha o + (1-\alpha)\mathcal{I}$) as our augmentation method. Similar to the previous works [58, 28] we add a regularization term $\mathcal{R}_\theta$ to the critic objective $\mathcal{F}_\theta$ without introducing extra hyperparameters and other techniques. Our critic loss $\mathcal{J}_\theta$ is as follows, where $\mathcal{D}$ is the replay buffer, $\mathbf{s_t}^{\text{aug}}$ is the augmented observation, $\widehat{Q}\left(\mathbf{s_t}, \mathbf{a_t}\right) = r\left(\mathbf{s_t}, \mathbf{a_t}\right) + \gamma \mathbb{E}_{\mathbf{s_{t+1}} \sim \mathcal{P}}\left[V\left(\mathbf{s_{t+1}}\right)\right]$:

$$\mathcal{J}_Q(\theta) = \mathcal{F}_Q(\theta) + \mathcal{R}_Q(\theta), \tag{3}$$

with

$$\mathcal{F}_Q(\theta) = \mathbb{E}_{(\mathbf{s_t}, \mathbf{a_t}) \sim \mathcal{D}}\left[\frac{1}{2}\left(Q_\theta\left(\mathbf{s_t}, \mathbf{a_t}\right) - \widehat{Q}\left(\mathbf{s_t}, \mathbf{a_t}\right)\right)^2\right]$$

$$\mathcal{R}_Q(\theta) = \mathbb{E}_{(\mathbf{s_t}, \mathbf{a_t}) \sim \mathcal{D}}\left[\frac{1}{2}\left(Q_\theta\left(\mathbf{s_t}^{\text{aug}}, \mathbf{a_t}\right) - \widehat{Q}\left(\mathbf{s_t}, \mathbf{a_t}\right)\right)^2\right]$$

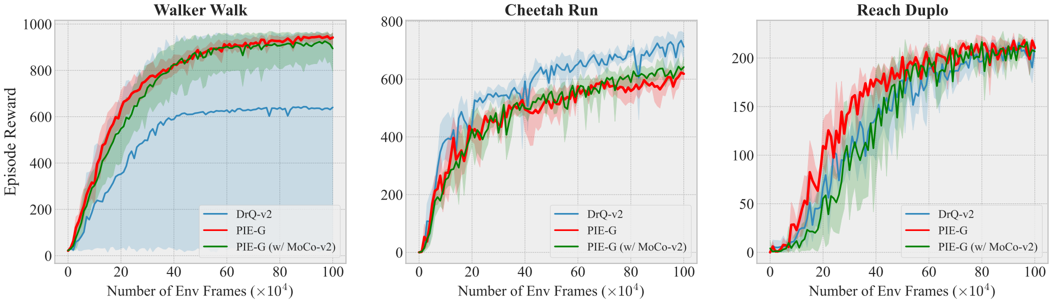


Figure 10: **Other pre-trained models.** PIE-G with MoCo-v2 also achieves competitive sample efficiency.

Drawer World. For the Drawer World task, we use a small learning rate in order to maintain the training stability. The episode lengths in Drawer World tasks are 200 steps with 4 action repeat. Random Conv is applied as the data augmentation method. Meanwhile, the original reward setting in the Drawer World is prone to the Q-value divergence. Therefore, we scale the reward by 0.01.

Manipulation tasks. The episode lengths in Manipulation tasks are 1000 steps with 2 action repeat. Since for generalization the data augmentation will degrade the training efficiency, we do not apply *random overlay* on this benchmark. We change the physical parameters of *geom size* to deform shape. All methods are evaluated with 100 episodes on different settings.

CARLA. We adopt the setting from Zhang et al. [85] (e.g., the reward function and training weather conditions). The maximum episode length in CARLA tasks is 1000 steps with 2 action repeat.

C Additional Results

In this section, we provide additional experimental results about PIE-G in various aspects.

C.1 Comparison with RRL

RRL [63] is another ResNet pre-trained algorithm that can achieve comparable sample efficiency with the state-based algorithms and be robust to the visual distractors. Here we compare the generalization ability of PIE-G with RRL. To compare fairly , we re-implement RRL with DrQ-v2 [78] as the base algorithm which is the state-of-the-art methods in DMControl Suite. Table 9 shows that RRL cannot adapt to the environments with distributional shifts while PIE-G exhibits considerable generalization ability when facing new visual scenarios. We suggest that the choice of layers and ever-updating BatchNorm are the crucial factors for bridging domain gaps and boosting agents' generalization performance.

Task	RRL	PIE-G
Walker Walk	46±15	**600**±**28**
Cheetah Run	29±10	**154**±**17**
Walker Stand	154±12	**856**±**51**

Table 9: **Compare with RRL.** RRL barely generalize to the new environments in the DMC-GB.

C.2 Other Pre-trained Models

As shown in Figure 10, PIE-G with the MoCo-v2 pre-trained model also gains a competitive sample efficiency with the help of the off-the-shelf visual representations.

C.3 Choice of Architectures

Tasks	ResNet18	ResNet34	ResNet50
Walker Walk	600 ±28	620 ±38	563 ±57
Cheetah Run	154 ±17	143 ±20	149 ±21
Walker Stand	852 ±56	867 ±24	871 ±22

Table 10: **Choice of architectures.** PIE-G with different architectures gains comparable generalization performance.

We further explore the impact of different network architectures. Since Layer 2 shows better performance, here we choose this layer to extract features. As shown in Table 10, three kinds of network architectures show comparable generalization performance . Since ResNet18 is less

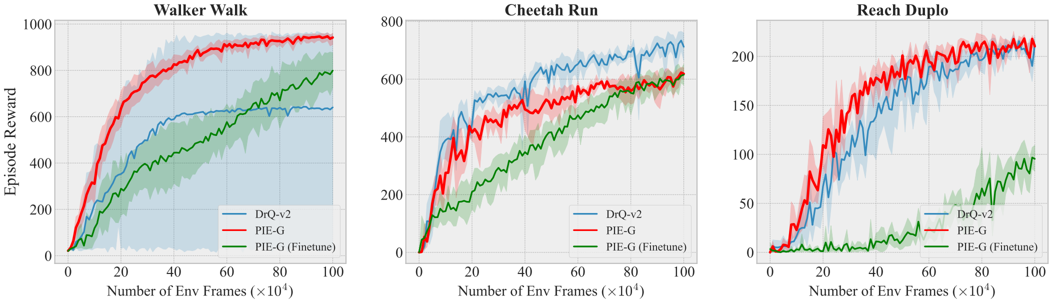


Figure 11: **Finetune the model.** This figure indicates that finetuning the encoder *green line* will sharply reduce the sample efficiency during training.

computationally demanding and with faster wall-clock time than the other two architectures, we choose it as the network backbone.

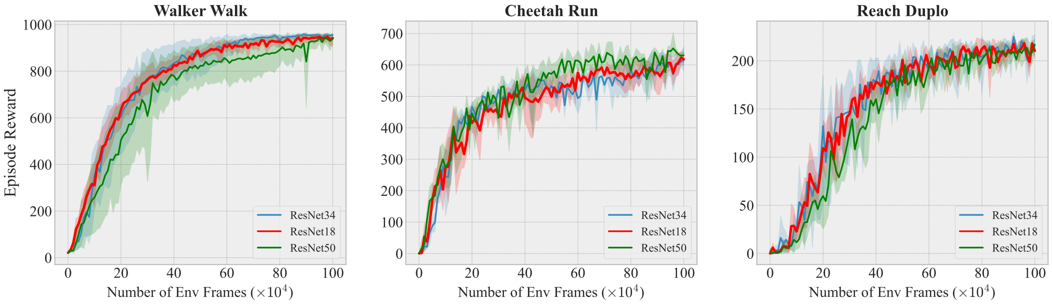


Figure 12: **Choice of architectures.** This figure indicates that PIE-G with various architectures achieves comparable sample efficiency.

C.4 Finetune Models

As shown in Figure 11, finetuning encoders will significantly reduce sample efficiency. We suggest that during the finetuning process, the encoders have to adapt to the new data distribution and unable to inherit the useful representations learned from the ImageNet, thus severely hindering the improvement in sample efficiency. Additionally, Table 6 and Figure 8 indicate that finetuning the model will make the agents overfit to the training environment.