# Robust Visual Reinforcement Learning by Prompt Tuning

Tung Tran[1]*, Khoat Than[2], and Danilo Vargas[1]

[1] Kyushu University, Japan
tran.tung.son.949@s.kyushu-u.ac.jp
vargas@inf.kyushu-u.ac.jp
[2] Hanoi University of Science and Technology, Vietnam
khoattq@soict.hust.edu.vn

**Abstract.** Training an agent based solely on observational data in a single environment, which then performs well in a zero-shot manner in unseen contexts, presents a significant challenge in the field of Reinforcement Learning. Given that environmental signals are limited to pixel-based inputs, the development of a generalized visual encoder is crucial for enhancing the agent's robustness. While pre-trained image encoders provide a straightforward and effective means of obtaining universal representations, the inability to perform end-to-end retraining on off-the-shelf models limits them from acquiring essential in-domain knowledge. This paper explores the promising potential of Visual Prompt Tuning to construct a more resilient image encoder for the agent. Extensive empirical evaluations are conducted on multiple benchmarks derived from the DeepMind Control Suite. The findings indicate notable improvements in both episode rewards and sample efficiency.

**Keywords:** Visual Prompt Tuning · Visual Reinforcement Learning · Data augmentation

## 1 Introduction

A Reinforcement Learning (RL) agent being trained with visual observations has shown remarkable ability in various domains such as video games [29,30], robotic manipulation [20, 23], and visual navigation [48]. However, making it adaptable to unseen environments is still a challenge, especially for high-dimensional observation spaces such as images, where overfitting [9] can be a significant issue.

Given that the agent perceives its environment through images, a visual encoder is essential for obtaining meaningful representations. Consistent with advancements in supervised learning, techniques such as Domain Randomization [40] and Data Augmentation [43] have been extensively utilized to enhance the variability of training data. Nevertheless, these augmentation methods [12] can increase the variance of Q-values, resulting in sample inefficiency and instability during training. To address the challenge of constructing a generalized encoder,

* Corresponding author.

pre-trained image encoders [46] have been explored to leverage out-of-domain data effectively. However, the straightforward application of pre-trained models to downstream tasks can be hindered by the domain gap issue in transfer learning [49], potentially leading to suboptimal performance.

A recently emerged method for fine-tuning pre-trained models, known as Visual Prompt Tuning (VPT) [18], has shown significant promise. Inspired by the prompt technique used in fine-tuning large language models [26, 27], VPT incorporates a small number of trainable parameters into the input space while keeping the backbone model frozen. This approach preserves the effectiveness and robustness of the pre-trained models while simultaneously allowing the use of data augmentation techniques. In this paper, we investigate the potential of VPT for developing more robust visual RL agents in continuous control tasks, aiming to achieve high performance in a zero-shot manner in unseen contexts.

To validate the effectiveness of this framework, we train the agent on various tasks from the DeepMind Control Suite [39] and evaluate it using the DMControl Generalization Benchmark (DMC-GB) [13]. Our empirical studies demonstrate improvements or competitive results compared to state-of-the-art methods, both in terms of episode rewards and sample efficiency. We named our method as *PromptAgent*.

In conclusion, our contributions are threefold:

- We validate the potential of VPT for fine-tuning visual foundation models to obtain a more robust representation encoder within the continuous control tasks.
- We propose a general framework that combines data augmentation techniques with the VPT method.
- We conduct extensive experiments and ablation studies to examine various aspects of the proposed framework.

## 2 Related works

**Representation learning in RL.** Self-supervised learning objectives [3, 5, 14, 15] are widely employed to learn invariant representations in computer vision. Pre-trained models have demonstrated significant success in downstream tasks such as object detection and semantic segmentation. In the realm of Reinforcement Learning (RL), there is a substantial body of research [11, 22, 35, 36, 42] focused on leveraging representation learning. A prevalent strategy involves integrating contrastive learning objectives with data augmentation to acquire more generalized representations. Another method to learn invariant visual representations is through auxiliary tasks [1, 47]. However, it is crucial to select these tasks carefully, as improper choices can result in gradient interference, as discussed in [17, 25, 28].

**Data augmentation for RL.** Domain randomization [4, 32–34] and data augmentation [12, 13, 41, 43] are effective strategies for achieving generalization across

visually diverse environments. However, incorporating data augmentation directly into the target network can increase the variance of value estimates and potentially lead to divergence [38]. This issue has been addressed and mitigated in [12] by introducing a regularization term that aligns the augmented and unaugmented versions of the same observation, and excluding its use in the target network. Additionally, researchers in [45] explore the preservation of high Lipschitz value pixels through the construction of masks to maintain Lipschitz continuity.

**Pre-trained visual encoder for RL.** Large visual pre-trained models are expected to be effective in zero-shot or few-shot scenarios for downstream tasks, alleviating the burden of curating datasets for training deep learning models. However, the straightforward application of these models often suffers from suboptimal performance due to the significant gap between the pre-training and current domains. With the aid of expert demonstrations, researchers in [31, 37] demonstrate the effectiveness of ResNet as a representation learner for RL agents. Addressing the problem of generalization in RL, PIE-G [46] shows through experiments that the features encoded in the early layers of ResNet can effectively handle large distribution shifts in a zero-shot manner while maintaining high sample efficiency. A limitation of relying solely on a pre-trained backbone is its inability to mitigate distribution shifts between the pre-training data and observations encountered. Addressing this challenge necessitates the integration of in-domain knowledge. Introduced in [8], ConPE is a framework for constructing a visual prompt by initially building a prompt pool through contrastive learning from expert demonstrations and subsequently using attention weights for fusion. Although this approach addresses the problem of navigation under egocentric settings, it demonstrates the potential of tuning visual pre-trained models for robust RL.

## 3 Preliminaries

**Reinforcement learning.** Since visual reinforcement learning (RL) offers only a partial view of the state space through image observations [19], we formulate the problem as a Partially Observable Markov Decision Process (POMDP) [2]. This is represented as a tuple $\langle \mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{P}, r, \gamma \rangle$, where:

- $\mathcal{S}$ is the state space,
- $\mathcal{O}$ is the observation space,
- $\mathcal{A}$ is the action space,
- $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function,
- $\mathcal{P}(.|s_t, a_t)$ is the state transition function, defining a conditional probability distribution over all possible next states given a state $s_t \in \mathcal{S}$ and action $a_t \in \mathcal{A}$ taken at time $t$,
- $\gamma \in [0, 1]$ is the reward discount factor.

A state $s_t$ is a sequence of $k+1$ consecutive frames $(o_t, o_{t-1}, \ldots, o_{t-k})$ with $o \in \mathcal{O}$, as proposed in [29]. The goal is then to learn a policy $\pi : \mathcal{S} \to \mathcal{A}$ that maximizes the discounted return $R_t = \mathbb{E}_{\Gamma\sim\pi}\left[\sum_{t=1}^{T} \gamma^t r(s_t, a_t)\right]$ along a trajectory $\Gamma$ obtained by following policy $\pi$ from an initial state $s_0$ to a state $s_T$ with state transitions sampled from $\mathcal{P}$. The policy $\pi$ is parameterized by learnable parameters $\theta$ and denoted as $\pi_\theta$. As formulated in [12], we further learn the policy to generalize well under an unseen environment $\bar{\mathcal{S}}$, where states $\bar{s_t} \in \bar{\mathcal{S}}$ are constructed from observations $\bar{o_t} \in \bar{\mathcal{O}}$, a perturbed observation space.

**DrQ-v2.** Introduced in [41], DrQ-v2 is a model-free off-policy algorithm built upon Deep Deterministic Policy Gradient (DDPG) [24] to optimize a stochastic policy. It also employs the clipped double-Q trick [10] and slow-moving target parameters [24] for stabilizing the training process. The critic network contains two $Q_{\phi_k}$ value functions and their corresponding target networks $Q_{\psi_k}$. The critic network is optimized with the following loss function, where a mini-batch $\tau = (s_t, a_t, r_{t:t+n-1}, s_{t+n})$ is sampled from the replay buffer $\mathcal{B}$:

$$\mathcal{L}(\phi_i, \mathcal{B}) = \mathbb{E}_{\tau\sim\mathcal{B}}\Big[(Q_{\phi_i}(h_t, a_t) - y)^2\Big] \quad \forall k \in \{1, 2\} \tag{1}$$

with n-step TD target $y$ is defined as:

$$y = \sum_{i=0}^{n-1} \gamma^i r_{t+i} + \gamma^n \min_{k=1,2} Q_{\psi_k}(h_{t+n}, a_{t+n}) \tag{2}$$

where $h_t = f_\xi(s_t), h_{t+n} = f_\xi(s_{t+n}), a_t = \pi_\theta(h_t) + \epsilon, a_{t+n} = \pi_\theta(h_{t+n}) + \epsilon$, and $\epsilon \sim \text{clip}(\mathcal{N}(0, \sigma^2), -c, c)$ with a decaying exploration noise $\sigma$, and $f_\xi$ is the image encoder parameterized by $\xi$ as proposed in [44].

The actor network (or parameterized policy $\pi_\theta$) is trained with the following loss:

$$\mathcal{L}(\theta, \mathcal{B}) = -\mathbb{E}_{s_t\sim\mathcal{B}}\Big[\min_{k=1,2} Q_{\phi_k}(h_t, a_t)\Big] \tag{3}$$

The target network's parameters at step $t$ will be updated as follows:

$$\psi_{t+1} \longleftarrow (1-\zeta)\psi_t + \zeta\theta_t \tag{4}$$

where $\zeta \in (0, 1]$.

We model the Actor network and the Q-value functions of the Critic and Target networks using a Multi-Layer Perceptron. Additionally, we utilize the replay buffer implementation from DrQ-v2, which significantly reduces memory requirements and can accommodate over 1 million states.

## 4 PromptAgent - The proposed method

In this section, we introduce the application of Visual Prompting Tuning (VPT) to the ResNet model, which we use as an example for further analysis. However, VPT can be easily applied to any convolutional neural network (ConvNet)

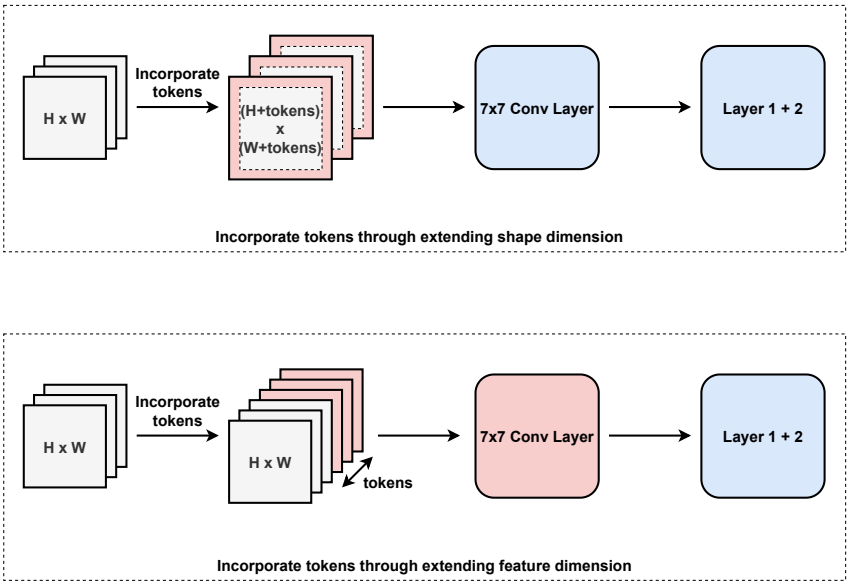



**Fig. 1:** Two variants of incorporating tokens [18] to pre-trained ResNet models. The top extends the shape dimension (referred to as *pad*) by padding both the height and width of the input images with learnable *tokens*, while the bottom extends the feature dimension (referred to as *below*) by adding *tokens* along the feature dimension. The colors red and blue indicate learnable and frozen elements of the backbone, respectively. In the original architecture, the size of the input features for the *ConvLayer* is 3 so we have to modify this layer in the *below* method. The reason why we don't use the features extracted from the last layer but stop at the second layer is explained in [46], where the feature maps in this layer largely preserve the details of the main subject.

model. Additionally, we demonstrate the overall framework for utilizing data augmentation techniques to fine-tune pre-trained models.

### 4.1 Visual Prompt Tuning for ResNet

Instead of the traditional framework for fine-tuning pre-trained models, which involves end-to-end retraining of the entire model on a downstream dataset, VPT injects a significantly smaller number of learnable parameters into the input spaces while keeping most of the backbone parameters frozen. These tokens are optimized via gradient descent. This method has been proven successful for numerous downstream tasks, as shown in [18], outperforming the fully fine-tuning method. These added tokens are expected to help us learn in-domain useful information through data augmentation. By combining this with the existing

capability of the backbone encoder to capture useful features, we hypothesize that the features extracted from this encoder will improve both the sample efficiency and the generalization ability of the RL agent. Fig. 1 demonstrates two variants of incorporating tokens, namely *pad* and *below*. The *ConvLayer* refers to the sequence of the first convolution, batch normalization, ReLU, and pooling layer, while *Layer* $1 + 2$ refers to the first and second layers extracted from the original ResNet family models [16]. The percentage of trainable parameters over all backbone parameters of ResNet-18 is 3.5% for the *pad* method with 20 *tokens*.

### 4.2 PromptAgent design

An overview of our method, namely *PromptAgent*, is presented in Fig. 2. Our framework draws inspiration from the architecture proposed in [41], wherein an image encoder transforms observations from pixel space to latent space. The encoded representation is subsequently fed into the critic network, target network, and actor network. The actor network is omitted in Fig. 2 due to its similarity to previous designs.

The image encoder contains a *Prompt* layer, designed to incorporate learnable tokens into input spaces, a *Frozen* layer adopted from a pre-trained model, and a *Head* layer implemented as a Multi-layer Perceptron (MLP) to reduce feature dimensionality. In this architecture, only the parameters in the *Prompt* and *Head* layers are trainable.

A critical aspect of our design is the application of data augmentation exclusively to the current state observation $s_t$ and not to the subsequent state observation $s_{t+1}$, which is used for computing the target Q-value. This approach mitigates the issue of high variance in Q-value estimation through bootstrapping, as discussed in [12].

## 5 Empirical evaluation

We first describe our experiment setting in Sec. 5.1, followed by the results on adaptation ability and sample efficiency in Sec. 5.2 and Sec. 5.3, respectively, and conclude with a systematic study of the effect of different design choices in Sec. 5.4.

### 5.1 Experiment Setup

**Pre-trained Backbone.** We select ResNet-18 [16] as the backbone for our image encoder. For the majority of the experiments, we employ *pad* with 20 tokens as the VPT method.

**Baselines.** Our evaluation concentrates on two primary metrics: *Generalization Ability* and *Training Sample Efficiency*. For the generalization benchmark, we compare our results with various algorithms reported in [46]. For the sample efficiency, we compare our method with the DrQ-v2 and PIE-G algorithms.

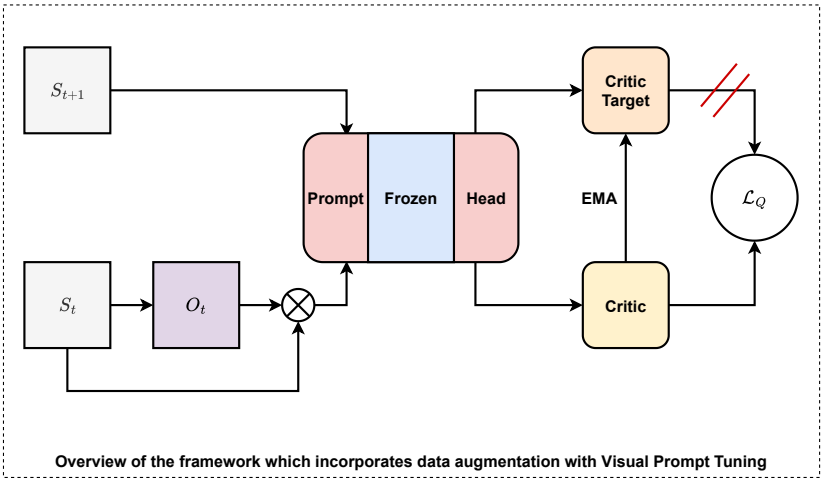



**Fig. 2:** PromptAgent for robust reinforcement learning from pixels. We incorporate data augmentation techniques to process the observation $s_t$ and obtain a perturbed version, $o_t$. Both $s_t$ and $o_t$ are then fed into the critic network using the concatenation operator. For the target network, we utilize only the next observation $s_{t+1}$ without applying any data augmentation. This step is crucial for reducing the variance of the Q-value function. The loss value $\mathcal{L}_Q$ is computed following the method outlined by Eq. (1). The gradient is propagated solely to the critic network, and the parameters of the target network are updated using the Exponential Moving Average as described in Eq. (4).

**Training Environment.** We train the agent across multiple tasks within the DeepMind Control Suite environment [39], which includes a collection of continuous control tasks characterized by a standardized structure and interpretable rewards. This suite is extensively utilized in the visual reinforcement learning research community. All training processes are executed on a single A100 GPU. The Adam optimizer [21] is employed for the visual encoder, the critic, and the actor networks.

**Settings.** In the generalization benchmark, the agent is trained for 500,000 steps with 2 action repeats. Evaluation is conducted in the DMC-GB [13] in a zero-shot manner over 10 episodes, with results averaged across multiple seeds. In the sample efficiency benchmark, the agent is trained for 100,000 steps with 2 action repeats, and the episode reward is recorded and reported.

**Table 1: Generalization in a color-jittered environment.** In this environment, the color of the subject or background is modified during the evaluation phase, requiring the agent to adapt to these changes in a zero-shot manner. The reported result includes mean and standard deviation over 4 seeds. *PromptAgent* shows the best performance in all tasks considered. The best mean score will be bold, while the second best score will be italicized.

| Task | DrQ | DrQ-v2 | SVEA | TLDA | PIE-G | **PromptAgent** |
|---|---|---|---|---|---|---|
| Cartpole, Swingup | 586 ± 52 | 277 ± 80 | *837* ± 23 | 760 ± 60 | 749 ± 46 | **842** ± 44 |
| Walker, Stand | 770 ± 71 | 413 ± 61 | 942 ± 26 | 947 ± 26 | *960* ± 15 | **961** ± 32 |
| Walker, Walk | 520 ± 91 | 168 ± 90 | 760 ± 145 | 832 ± 58 | *884* ± 20 | **906** ± 48 |
| Cup, Catch | 365 ± 210 | 469 ± 99 | 961 ± 7 | 932 ± 32 | *964* ± 7 | **970** ± 15 |
| Cheetah, Run | 100 ± 27 | 109 ± 45 | 273 ± 23 | *371* ± 51 | 369 ± 53 | **453** ± 25 |

### 5.2 Evaluation on Generalization

To assess the generalization capabilities, we refer to findings reported in [46] for comparison. The evaluated methods include **DrQ** [43], an off-policy actor-critic algorithm incorporating data augmentation; **DrQ-v2** [41], an enhanced version of **DrQ** featuring optimizations for easier data augmentation and accelerated training; **SVEA** [12], addressing issues of high variance and over-regularization in Q-value estimation; **TLDA** [45], which mitigates Q-value variance by selectively augmenting pixels; and **PIE-G** [46], exploring the use of feature extraction from pre-trained models in visual RL contexts.

**Generalization in the color-jittered environment.** In this environment, the color of the subject or background will be modified. Tab. 1 has shown that *PromptAgent* has achieved the best performance across all tasks, showing robustness in the color-changing environment. The test benchmark is referred to as *color hard* in DMC-GB.

**Generalization in the unseen background environment.** In this environment, the background during the evaluation phase will be replaced by a rapidly changing one. The *video easy* mode will substitute the background with a natural scene, whereas the *video hard* mode will alter both the background and the floor to depict a daily scene.

The results are presented in the Tab. 2. *PromptAgent* demonstrates superior performance in 7 out of 12 tasks and maintains competitive performance in the remaining tasks. Notably, for the *Walker Walk* task, the performance gain over PIE-G of *PromptAgent* is significant at **42%**.

**Table 2: Generalization in the unseen background environment.** Environment with a dynamic background. *PromptAgent* shows the best performance in 7 out of 12 tasks and achieves competitive performance in the remaining one. The average improvement on the *video hard* setting is **+12.5%**. The best mean score will be bold, while the second best score will be italicized.

| Settings | Task | DrQ | DrQ-v2 | SVEA | TLDA | PIE-G | **PromptAgent** |
|---|---|---|---|---|---|---|---|
| Video Hard | Cartpole, Swingup | 138 ± 9 | 130 ± 3 | 393 ± 45 | 286 ± 47 | *401* ± *21* | **423** ± **89** |
| | Walker, Stand | 289 ± 49 | 151 ± 13 | 834 ± 46 | 602 ± 51 | *852* ± *56* | **928** ± **38** |
| | Walker, Walk | 104 ± 22 | 34 ± 11 | 377 ± 93 | 271 ± 55 | *600* ± *28* | **862** ± **48** |
| | Cup, Catch | 92 ± 23 | 97 ± 27 | 403 ± 174 | 257 ± 57 | *786* ± *47* | **882** ± **79** |
| | Cheetah, Run | 32 ± 13 | 23 ± 5 | 105 ± 37 | 90 ± 27 | **154** ± **17** | *152* ± *17* |
| | Finger, Spin | 71 ± 45 | 21 ± 4 | 335 ± 58 | 241 ± 29 | *762* ± *59* | **795** ± **27** |
| Video Easy | Cartpole, Swingup | 485 ± 105 | 267 ± 41 | **782** ± **27** | 671 ± 57 | 587 ± 61 | *676* ± *56* |
| | Walker, Stand | 873 ± 83 | 560 ± 48 | *961* ± *8* | **973** ± **6** | 957 ± 12 | 944 ± 44 |
| | Walker, Walk | 682 ± 89 | 175 ± 117 | 819 ± 71 | *873* ± *34* | 871 ± 22 | **908** ± **42** |
| | Cup, Catch | 318 ± 157 | 454 ± 60 | 871 ± 106 | 892 ± 68 | *922* ± *20* | **957** ± **25** |
| | Cheetah, Run | 102 ± 30 | 64 ± 22 | 249 ± 20 | **366** ± **57** | 287 ± 20 | *299* ± *25* |
| | Finger, Spin | 533 ± 119 | 456 ± 15 | *808* ± *33* | 744 ± 18 | **837** ± **107** | 806 ± 18 |

### 5.3 Evaluation on Sample Efficiency

We report the sample efficiency of *PromptAgent* in comparison with DrQ-v2 [41] and PIE-G [46] on 5 tasks in the DeepMind Control Suite [39] in Fig. 3. For this benchmark, we retrain DrQ-v2 and PIE-G using their publicly available code.

### 5.4 Ablation study

We conducted a series of ablation studies to closely examine various design choices for the framework of *PromptAgent*.

**Method of incorporating tokens.** As discussed in Sec. 4.1, we investigated two strategies for incorporating varying numbers of tokens to determine which

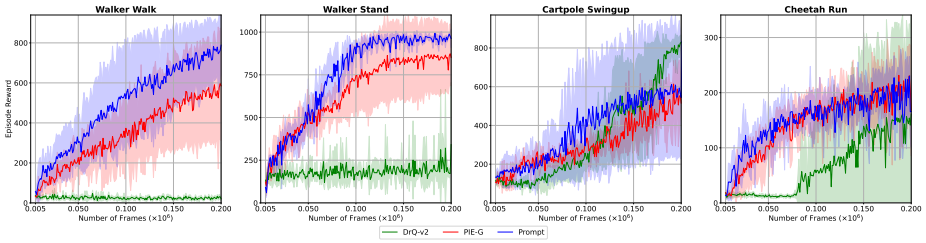



**Fig. 3:** Average episode rewards for continuous control tasks, with means and standard deviations calculated over 4 seeds. Our approach is compared against DrQ-v2 and PIE-G. Our method consistently outperforms PIE-G across all tasks, highlighting the advantages of integrating domain-specific knowledge into a pretrained model.

training approach could achieve optimal performance in Fig. 4. Given the constrained size of the observation ($3 \times 84 \times 84$), increasing the number of tokens does not necessarily enhance efficiency. This is because token-derived information may dominate the observation's inherent details, potentially erasing essential low-level features. Conversely, we hypothesize that a smaller number of tokens such as 5 may enhance sample efficiency without fully leveraging the agent's generalization capability. However, further investigation is warranted to explore this hypothesis thoroughly.

**Choice of feature layer.** In deep learning models, deeper layers are known to capture high-level semantic features, while shallower layers retain low-level details. Consistent with the findings reported in [46], our observations indicate that features extracted from Layer 2 show the best trade-off in the continuous control tasks and yield the best generalization performance. Fig. 5a illustrates the effects of using different layers on sample efficiency.

**Choice of pre-training method.** We also examined the impact of different pre-training methods, specifically comparing two self-supervised training methods, MoCo-v2 [6] and MoCo-v3 [7], with fully supervised training from the ImageNet dataset. Fig. 5b illustrates that the choice of pre-training method does not significantly affect sample efficiency.

**Choice of network size.** Fig. 6a demonstrates the effect of varying network sizes. We conducted experiments with five versions of the ResNet model, ranging from 18 to 152 layers. In general, increasing the depth of the model does not yield significant performance gains but requires substantially more training time. Specifically, the 152-layer network needs four times the wall-clock time compared to the 18-layer version when training the agent for 200,000 frames. To address the limitations of smaller networks, we can increase the mini-batch size, which will be investigated in the next subsection.

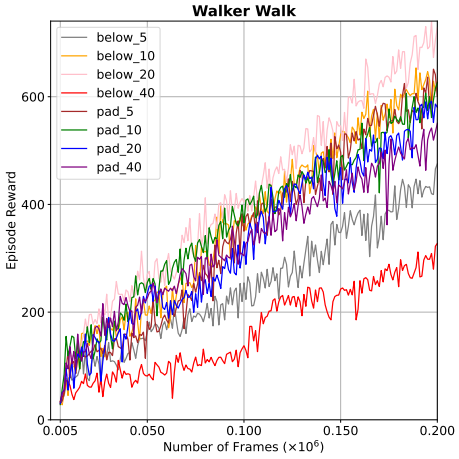


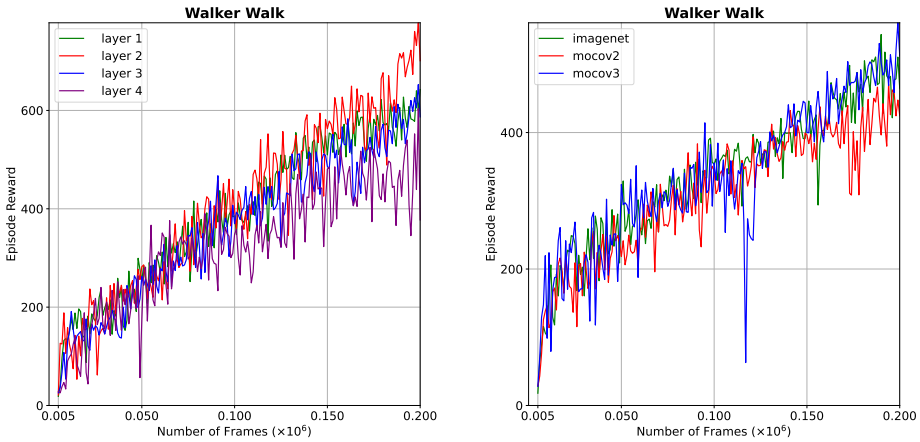

**Fig. 4:** Episode reward from agents trained with different VPT methods. We hypothesize that given the observation size normalized to $84 \times 84$, padding with 20 tokens will effectively double the input size, offering a balanced trade-off between sample efficiency and generalization capability.

**Choice of mini-batch size.** Fig. 6b illustrates the impact of various mini-batch size choices. Interestingly, the performance with a batch size of 256 is inferior to that with a batch size of 128. Generally, a larger batch size improves the training efficiency of the agent. However, the computational time and memory requirements increase linearly with the batch size. Therefore, selecting an appropriate batch size for specific tasks within the evaluation environment is important.

## 6 Conclusion

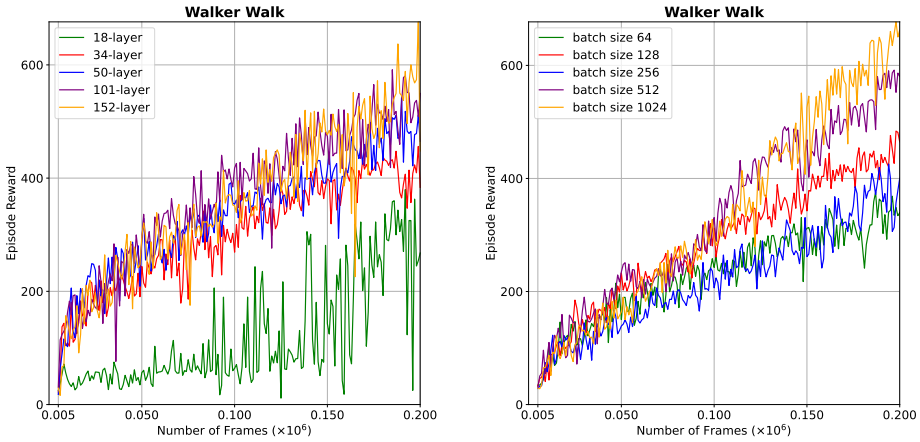
In this paper, we propose *PromptAgent*, a method that utilizes Visual Prompt Tuning and data augmentation techniques to fine-tune a pre-trained visual foundation model. In the context of controlling agents using images alone, a robust visual representation is essential to ensure the generalization and sample efficiency of the agent. We conduct extensive experiments to verify the benefits of the proposed methods and perform thorough ablation studies to highlight the most important design aspects. This work can serve as a stepping stone for further research on the use of pre-trained models in visual reinforcement learning.

**Acknowledgement.** This work was supported by JSPS Research on Academic Transformation Areas (A) - Grant Number JP22H05194.




**(a)** Choice of feature layer **(b)** Choice of pretraining method

**Fig. 5:** Variation of layers and pre-trained weights in terms of the sample efficiency




**(a)** Choice of network size **(b)** Choice of batch size

**Fig. 6:** Variation of network size and mini-batch size in terms of the sample efficiency

## References


1. Agarwal, R., Machado, M.C., Castro, P.S., Bellemare, M.G.: Contrastive behavioral similarity embeddings for generalization in reinforcement learning. arXiv preprint arXiv:2101.05265 (2021)
2. Bellman, R.: A markovian decision process. Journal of mathematics and mechanics pp. 679–684 (1957)
3. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. Advances in neural information processing systems **33**, 9912–9924 (2020)
4. Chebotar, Y., Handa, A., Makoviychuk, V., Macklin, M., Issac, J., Ratliff, N., Fox, D.: Closing the sim-to-real loop: Adapting simulation randomization with real world experience. In: 2019 International Conference on Robotics and Automation (ICRA). pp. 8973–8979. IEEE (2019)
5. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020)
6. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297 (2020)
7. Chen, X., Xie, S., He, K.: An empirical study of training self-supervised vision transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9640–9649 (2021)
8. Choi, W., Kim, W.K., Kim, S., Woo, H.: Efficient policy adaptation with contrastive prompt ensemble for embodied agents. In: Thirty-seventh Conference on Neural Information Processing Systems (2023), `https://openreview.net/forum?id=Ny3GcHLyzj`
9. Cobbe, K., Klimov, O., Hesse, C., Kim, T., Schulman, J.: Quantifying generalization in reinforcement learning. In: International conference on machine learning. pp. 1282–1289. PMLR (2019)
10. Fujimoto, S., Hoof, H., Meger, D.: Addressing function approximation error in actor-critic methods. In: International conference on machine learning. pp. 1587–1596. PMLR (2018)
11. Gelada, C., Kumar, S., Buckman, J., Nachum, O., Bellemare, M.G.: Deepmdp: Learning continuous latent space models for representation learning. In: International conference on machine learning. pp. 2170–2179. PMLR (2019)
12. Hansen, N., Su, H., Wang, X.: Stabilizing deep q-learning with convnets and vision transformers under data augmentation. Advances in neural information processing systems **34**, 3680–3693 (2021)
13. Hansen, N., Wang, X.: Generalization in reinforcement learning by soft data augmentation. In: 2021 IEEE International Conference on Robotics and Automation (ICRA). pp. 13611–13617. IEEE (2021)
14. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16000–16009 (2022)
15. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9729–9738 (2020)
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)

17. Jaderberg, M., Mnih, V., Czarnecki, W.M., Schaul, T., Leibo, J.Z., Silver, D., Kavukcuoglu, K.: Reinforcement learning with unsupervised auxiliary tasks. arXiv preprint arXiv:1611.05397 (2016)
18. Jia, M., Tang, L., Chen, B.C., Cardie, C., Belongie, S., Hariharan, B., Lim, S.N.: Visual prompt tuning. In: European Conference on Computer Vision. pp. 709–727. Springer (2022)
19. Kaelbling, L.P., Littman, M.L., Cassandra, A.R.: Planning and acting in partially observable stochastic domains. Artificial intelligence **101**(1-2), 99–134 (1998)
20. Kalashnikov, D., Irpan, A., Pastor, P., Ibarz, J., Herzog, A., Jang, E., Quillen, D., Holly, E., Kalakrishnan, M., Vanhoucke, V., et al.: Scalable deep reinforcement learning for vision-based robotic manipulation. In: Conference on robot learning. pp. 651–673. PMLR (2018)
21. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
22. Laskin, M., Srinivas, A., Abbeel, P.: Curl: Contrastive unsupervised representations for reinforcement learning. In: International conference on machine learning. pp. 5639–5650. PMLR (2020)
23. Levine, S., Finn, C., Darrell, T., Abbeel, P.: End-to-end training of deep visuomotor policies. Journal of Machine Learning Research **17**(39), 1–40 (2016)
24. Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D.: Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971 (2015)
25. Lin, X., Baweja, H., Kantor, G., Held, D.: Adaptive auxiliary task weighting for reinforcement learning. Advances in neural information processing systems **32** (2019)
26. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G.: Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. ACM Computing Surveys **55**(9), 1–35 (2023)
27. Liu, X., Ji, K., Fu, Y., Tam, W.L., Du, Z., Yang, Z., Tang, J.: P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. arXiv preprint arXiv:2110.07602 (2021)
28. Lyle, C., Rowland, M., Ostrovski, G., Dabney, W.: On the effect of auxiliary tasks on representation dynamics. In: International Conference on Artificial Intelligence and Statistics. pp. 1–9. PMLR (2021)
29. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M.: Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602 (2013)
30. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. nature **518**(7540), 529–533 (2015)
31. Parisi, S., Rajeswaran, A., Purushwalkam, S., Gupta, A.: The unsurprising effectiveness of pre-trained vision models for control. In: international conference on machine learning. pp. 17359–17371. PMLR (2022)
32. Peng, X.B., Andrychowicz, M., Zaremba, W., Abbeel, P.: Sim-to-real transfer of robotic control with dynamics randomization. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE (May 2018). https://doi.org/10.1109/icra.2018.8460528, http://dx.doi.org/10.1109/ICRA.2018.8460528
33. Pinto, L., Andrychowicz, M., Welinder, P., Zaremba, W., Abbeel, P.: Asymmetric actor critic for image-based robot learning (2017)
34. Ramos, F., Possas, R.C., Fox, D.: Bayessim: adaptive domain randomization via probabilistic inference for robotics simulators. arXiv preprint arXiv:1906.01728 (2019)

35. Schwarzer, M., Anand, A., Goel, R., Hjelm, R.D., Courville, A., Bachman, P.: Data-efficient reinforcement learning with self-predictive representations. arXiv preprint arXiv:2007.05929 (2020)
36. Sekar, R., Rybkin, O., Daniilidis, K., Abbeel, P., Hafner, D., Pathak, D.: Planning to explore via self-supervised world models. In: International conference on machine learning. pp. 8583–8592. PMLR (2020)
37. Shah, R., Kumar, V.: Rrl: Resnet as representation for reinforcement learning. arXiv preprint arXiv:2107.03380 (2021)
38. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. MIT press (2018)
39. Tassa, Y., Doron, Y., Muldal, A., Erez, T., Li, Y., Casas, D.d.L., Budden, D., Abdolmaleki, A., Merel, J., Lefrancq, A., et al.: Deepmind control suite. arXiv preprint arXiv:1801.00690 (2018)
40. Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., Abbeel, P.: 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). pp. 23–30. IEEE (2017)
41. Yarats, D., Fergus, R., Lazaric, A., Pinto, L.: Mastering visual continuous control: Improved data-augmented reinforcement learning. arXiv preprint arXiv:2107.09645 (2021)
42. Yarats, D., Fergus, R., Lazaric, A., Pinto, L.: Reinforcement learning with prototypical representations. In: International Conference on Machine Learning. pp. 11920–11931. PMLR (2021)
43. Yarats, D., Kostrikov, I., Fergus, R.: Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In: International conference on learning representations (2020)
44. Yarats, D., Zhang, A., Kostrikov, I., Amos, B., Pineau, J., Fergus, R.: Improving sample efficiency in model-free reinforcement learning from images. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 10674–10681 (2021)
45. Yuan, Z., Ma, G., Mu, Y., Xia, B., Yuan, B., Wang, X., Luo, P., Xu, H.: Don't touch what matters: Task-aware lipschitz data augmentation for visual reinforcement learning. arXiv preprint arXiv:2202.09982 (2022)
46. Yuan, Z., Xue, Z., Yuan, B., Wang, X., Wu, Y., Gao, Y., Xu, H.: Pre-trained image encoder for generalizable visual reinforcement learning. Advances in Neural Information Processing Systems **35**, 13022–13037 (2022)
47. Zhang, A., McAllister, R., Calandra, R., Gal, Y., Levine, S.: Learning invariant representations for reinforcement learning without reconstruction. arXiv preprint arXiv:2006.10742 (2020)
48. Zhu, Y., Mottaghi, R., Kolve, E., Lim, J.J., Gupta, A., Fei-Fei, L., Farhadi, A.: Target-driven visual navigation in indoor scenes using deep reinforcement learning. In: 2017 IEEE international conference on robotics and automation (ICRA). pp. 3357–3364. IEEE (2017)
49. Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., He, Q.: A comprehensive survey on transfer learning. Proceedings of the IEEE **109**(1), 43–76 (2020)