Focus-Then-Reuse: Fast Adaptation in Visual Perturbation Environments

Jiahui Wang[1,2]*, **Chao Chen**[1,2]*, **Jiacheng Xu**[3], **Zongzhang Zhang**[1,2]†, **Yang Yu**[1,2]
[1]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
[2]School of Artificial Intelligence, Nanjing University, Nanjing, China
[3]Nanyang Technological University, Singapore
{wangjh,chenc}@lamda.nju.edu.cn, jiacheng005@e.ntu.edu.sg,
{zzzhang,yuy}@nju.edu.cn

Abstract

Visual reinforcement learning has shown promise in various real-world applications. However, deploying policies in complex real-world environments with visual perturbations remains a significant challenge. We notice that humans tend to filter information at the object level prior to decision-making, facilitating efficient skill transfer across different contexts. Inspired by this, we introduce Focus-Then-Reuse (FTR), a method utilizing a novel object selection mechanism to **focus** on task-relevant objects, and directly **reuse** the simulation-trained policy on them. The training of the object selection mechanism integrates prior knowledge from a vision-language model and feedback from the environment. Experimental results on challenging tasks based on DeepMind Control Suite and Franka Emika Robotics demonstrate that FTR enables rapid adaptation in visual perturbation environments and achieves state-of-the-art performance. The source code is available at https://github.com/LAMDA-RL/FTR.

1 Introduction

Visual Reinforcement Learning (RL) has achieved breakthroughs in a wide range of real-world applications, including robotic manipulation [1–3] and autonomous navigation [4, 5]. However, bridging the gap between simulation and real-world environments remains a pivotal challenge. In this work, we focus on distracting background, which is a typical form of visual disturbance. Taking robotic grasping as an example, although existing methods can achieve good performance in simulation [6, 7], directly deploying the learned policies in the real world may suffer from performance degradation when there exists complex backgrounds [8, 9].

Recent studies have explored ways to address the challenge of deploying policies in complex environments with visual perturbations. Existing approaches can be categorized into three groups: (1) training policies directly in noisy real-world environments [10–12]; (2) learning generalizable policies that are robust to environment variations [13–22]; and (3) adapting pre-trained policies to target environments with visual perturbations [23–25]. Each of these approaches presents potential limitations. First, policies trained directly in visually disturbed environments often incur high costs and risks, and the policy may generalize poorly to unseen settings. Second, generalization-based methods rely on techniques such as data augmentation to simulate real-world variations during training [26]. However, if the training process fails to capture the diversity of the deployment environment, severe performance degradation may occur. Lastly, existing adaptation methods struggle to

*These authors contributed equally.
†Zongzhang Zhang is the corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

fully preserve policy performance achieved in simulation. We concentrate on an adaptation approach that aims to preserve the source policy performance with minimal degradation.

Inspired by the stark contrast with the limitations of existing adaptation methods, we investigate human skill transfer. Notably, humans possess a natural ability to transfer skills across diverse contexts. We believe this ability can be attributed to two key aspects. First, during perception, humans perform object-level filtering to distinguish between task-relevant and irrelevant objects, effectively reducing the complexity of information processing [27]. Second, in decision-making, humans can leverage both prior knowledge and environmental feedback to guide their actions. Initially, object selection is driven by prior knowledge; if the outcome is unsatisfactory, humans will iteratively adjust their selection policy based on environmental feedback until the task is completed [28].

Humans acquire prior knowledge by summarizing a large number of past experiences. Similarly, foundation models gain strong prior knowledge through pre-training on large-scale and diverse datasets, achieving remarkable success in natural language processing [29, 30] and computer vision [31–33]. Recently, many studies have also explored using foundation models as a source of prior knowledge for downstream tasks [34, 35].

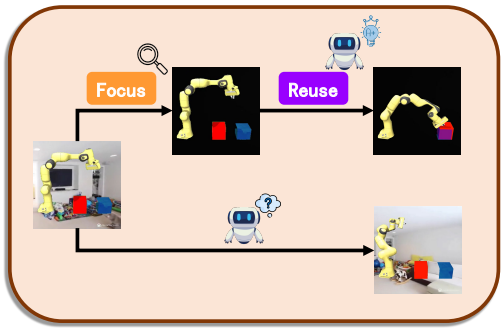


Figure 1: Directly deploying the learned policy in real-world environments may degrade performance. We propose Focus-Then-Reuse, a method that utilizes a novel object selection mechanism to focus on task-relevant objects, and directly reuses the learned policy on them.

Inspired by humans' ability to transfer skills and the success of foundation models, we introduce a new paradigm for visual domain adaptation RL. We propose Focus-Then-Reuse (FTR), a method that directly applies a simulation-trained policy to observations focused by a novel object selection mechanism (Fig. 1). The object selection mechanism consists of a trainable segment selector, a fixed segmentation model, and a fixed tracking model. The training of the segment selector synthesizes prior knowledge from a Vision-Language Model (VLM) and feedback from the environment. Experiments on DeepMind Control Suite and Franka Emika Robotics show that FTR facilitates rapid policy adaptation from clean training environments to visually perturbed target environments and achieves state-of-the-art (SOTA) performance.

We summarize the contribution of this paper as:

- We introduce a novel Focus-Then-Reuse framework for policy adaptation, with the focus stage filtering task-relevant objects and the reuse stage employing a fixed policy for faster and more stable adaptation.
- We propose an adaptation RL method that integrates supervised learning and reinforcement learning to synthesize VLM's prior knowledge and environmental feedback.
- Experiment on challenging tasks demonstrates that our method facilitates quick and effective policy adaptation to visual perturbation environments, achieving SOTA performance.

2 Related works

The pursuit of robust policy deployment across visually perturbed domains has catalyzed research interest in recent years. Some works try to train policies in visually distracted environments directly. DBC [36] first explores this pathway by learning a representation robust to distractions. Building on DBC, Q^2-learning [11] decouples policy learning from behavioral metric learning for stable training. Some model-based methods [12, 37–39] explicitly recognize task-relevant parts. However, policies trained directly in visually perturbed environments often suffer from high training costs and risks, as well as limited generalization to other scenarios.

Generalization methods explore ways to enhance policy performance, which can be classified into three types: data augmentation, inductive bias, and learning invariances [26]. Data augmentation methods [14–18] apply techniques like cropping and color jittering to training images to reduce

the distribution gap between training and testing. Inductive bias methods [19–22, 26] incorporate assumptions about task-relevance (e.g., foreground is of higher importance), among which SimGRL [13] achieves the SOTA performance. Invariance-based methods [40–45] focus on extracting information consistent across diverse training environments. We point out that generalization methods face two issues. First, added perturbations or inductive biases may hurt training performance. Second, when facing unseen disturbances, the performance may drop significantly.

Domain adaptation RL focuses on transferring policies trained in the source domain to the target domain [46–50]. Of particular relevance to this paper is visual domain adaptation RL, which remains relatively under-explored. PAD [23] is the work most closely related to our setting. It introduces a self-supervised objective during training and performs online adaptation at deployment through this objective, enabling policy adaptation to target domains with simple backgrounds. Other studies have explored policy adaptation to target domains with varying camera viewpoints [24], changes in color and object scale [25] or new visual dynamics [51]. However, we suggest that current adaptation methods fail to fully recover the policies' performance, with severe degradation in challenging environments like video background [17].

Foundation models can leverage their prior knowledge to support perception and decision-making in downstream tasks [52, 53]. For perception, some approaches [54, 55] employ foundation models as pre-trained feature extractors, while others [10, 56–59] incorporate promptable segmentation models to enhance scene understanding and representation learning for visual RL agents. For decision-making, a common approach is to employ foundation models as reward generators to provide learning signals for policy optimization [60, 61]. Our method leverages foundation models to assist RL from both aspects. On one hand, we use pre-trained segmentation and tracking models to process complex inputs in a zero-shot manner. On the other hand, we employ a Vision-Language Model (VLM) as a supervision signal in policy training.

3 Preliminaries

Reinforcement learning Traditional RL considers the task in the form of a Markov Decision Process (MDP) $M = (\mathcal{S}, \mathcal{A}, \mathcal{P}, R)$. $\mathcal{S}$ is state space, $\mathcal{A}$ is action space. $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition function that defines the conditional probability distribution $\mathcal{P}(s_{t+1}|s_t, a_t)$ over next states given state $s_t \in \mathcal{S}$ and action $a_t \in \mathcal{A}$ at time t, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a reward function. RL aims to learn a policy $\pi(a|s)$ that maximizes the expected discounted cumulative reward $\mathbb{E}_\pi \left[\sum_t \gamma^t r_t\right]$ where $\gamma \in [0, 1]$ is the discount factor. In visual RL tasks, the agent can only have access to the observation o_t rendered from s_t, and thus the policy becomes $\pi(a|o)$. In this paper, we use Proximal Policy Optimization (PPO) [62]. PPO is an effective algorithm for solving general RL problems. Given policy $\pi_\theta(a|s)$ and value function $V_\psi(s)$, PPO maximizes the following objectives:

$$\begin{aligned} \mathcal{L}(\theta) &= \mathop{\mathbb{E}}_{(s,a,r,s')\sim\pi_{\theta_{\text{old}}}} \left[\min\left(r(\theta)A^{\pi_{\theta_{\text{old}}}}(s,a), \text{clip}(r(\theta), 1-\epsilon, 1+\epsilon)A^{\pi_{\theta_{\text{old}}}}(s,a)\right)\right], \\ \mathcal{L}(\psi) &= \mathop{\mathbb{E}}_{(s,a,r,s')\sim\pi_{\theta_{\text{old}}}} \left[V_\psi(s) - \left(V_{\psi^-}(s) + A^{\pi_{\theta_{\text{old}}}}(s,a)\right)\right]. \end{aligned} \tag{1}$$

Here $r(\theta) = \frac{\pi_\theta(a|s)}{\pi_{\theta_{\text{old}}}(a|s)}$ is the probability ratio measuring how much the current policy deviates from the old policy used for data sampling, while the clip$(\cdot)$ operation prevents overly large policy updates. The hyperparameter ϵ defines the clipping boundary. The value function $V_\psi(s)$ is a parameterized estimate of the expected discounted cumulative reward starting from state s and following policy π_θ thereafter. The advantage $A^{\pi_{\theta_{\text{old}}}}(s, a)$ is usually estimated from $V_\psi(s)$ via Generalized Advantage Estimation (GAE) [63]. ψ^- denotes the target network parameter, which is periodically synchronized with ψ for stable training.

Visual foundation models Visual foundation models are commonly pre-trained on massive datasets and can adapt to various downstream tasks in a zero-shot way. In this work, we focus specifically on foundation models designed for image understanding [33], image segmentation [64], and object tracking [65]. For image understanding task, we use Qwen-VL-Max [33], a multimodal version of the Qwen large model series that outperforms current SOTA generalist models on multiple vision-language tasks. It takes a combination of image and text as input and generate structured textual outputs following a specified format (e.g., JSON). For image segmentation and object tracking, we use SAM 2 [32], which achieves SOTA performance

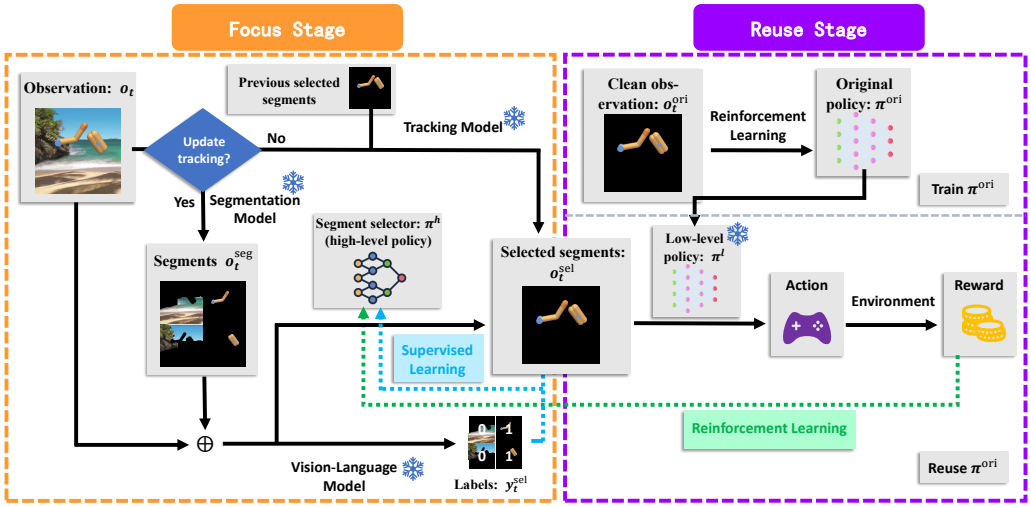


Figure 2: Architecture overview. FTR comprises a focus stage and a reuse stage, depicted in the orange and purple boxes, respectively. The focus stage utilizes a novel object selection mechanism to filter task-relevant segments, and the reuse stage applies a fixed, simulator-trained policy to generate actions based on the selected objects. The object selection mechanism consists of a trainable segment selector, a fixed segmentation model, and a fixed tracking model. The training of the segment selector synthesizes prior knowledge from a VLM and feedback from the environment.

in both tasks. In image segmentation task, the model takes an image $o \in \mathbb{R}^{C\times W\times H}$ (C, W, H are respectively channels, width, and height) as input and outputs a set of k binary masks $\left\{m^i \mid i \in \{1,\cdots,k\}, m^i \in \{0,1\}^{W\times H}\right\}$. Each mask m^i corresponds to a single object instance, with 1 indicating a pixel of the object. Segmented images are then obtained by element-wise multiplication ($\odot$), $\left\{o^i_{\text{obj}} \mid i \in \{1,\cdots,k\}, o^i_{\text{obj}} = o \odot m^i \in \mathbb{R}^{C\times W\times H}\right\}$. In object tracking task, the model takes a video and masks as input and propagates the masks across the video. Given a video with T frames $\left\{o_t \mid t \in \{1,\cdots,T\}, o_t \in \mathbb{R}^{C\times W\times H}\right\}$ and k object masks in the first frame $\left\{m^i_1 \mid i \in \{1,\cdots,k\}, m^i_1 \in \{0,1\}^{W\times H}\right\}$, the goal of tracking is to predict the object masks in subsequent frames, $\left\{m^i_t \mid t \in \{1,\cdots,T\}, i \in \{1,\cdots,k\}\right\}$.

4 Method

We present Focus-Then-Reuse (FTR), a hierarchical framework designed to quickly deploy policies in target domains with visual perturbations. FTR facilitates efficient policy adaptation by maintaining the core functionality of the original policy while dynamically compensating for visual perturbations through a learned "focus" module. In this section, we first provide a brief overview of the FTR framework. Following this, we introduce the forward process of our method, which is divided into two stages: the focus stage and the reuse stage. For the focus stage, we combine two filtering techniques to identify the task-relevant objects. For the reuse stage, we explain the training and reusing of the original policy. Finally, we present the training approach for the segment selector, which incorporates both supervised learning objective and reinforcement learning objective.

4.1 Architecture overview

An overview of FTR is provided in Fig. 2. FTR comprises two distinct stages: a high-level focus stage and a low-level reuse stage. They are in the orange and purple boxes, respectively. In the focus stage, an image observation with visual disturbance is processed. Based on whether the tracked objects require updating, task-relevant segments are obtained either through a segment selector (high-level policy π^h) or a tracking model. In the reuse stage, the selected objects o^{sel}_t are fed into the

original policy, which is acquired via pre-training in a clean environment, to obtain the action. The training objective of π^h combines RL with VLM supervision to enable efficient adaptation.

4.2 Focus stage: segmentation, selection and tracking

In this section, we introduce the focus stage in detail, as shown in the orange box in Fig. 2. At time step t, the image observation of the visually disturbed environment is denoted as $o_t \in \mathbb{R}^{C\times W\times H}$. The focus stage takes o_t as input and outputs the filtered task-relevant objects, denoted as $o_t^{\text{sel}} \in \mathbb{R}^{C\times W\times H}$. Depending on whether the tracked objects need update (blue diamond box in Fig. 2), there are two approaches: The first approach uses a segmentation model and a segment selector (the selection pathway). The second approach employs a tracking model and historical selected images (the tracking pathway). We introduce these two pathways and explain their necessities.

4.2.1 Selection pathway

The selection pathway corresponds to the downward arrow labeled "Yes" in Fig. 2. This pathway filters objects through an image segmentation model and a segment selector π^h. First, an image segmentation model is employed to segment the disturbed observation o_t, yielding a set of k segments: $o_t^{\text{seg}} = \left\{o_t^{\text{seg}_i} \mid i \in \{1,\cdots,k\}, o_t^{\text{seg}_i} \in \mathbb{R}^{C\times W\times H}\right\}$. Subsequently, π^h takes the k segments and the original observation o_t as input, and output the mean values $\boldsymbol{\mu}_t = (\mu_t^1, \mu_t^2, \cdots, \mu_t^k) \in (0,1)^k$. By sampling from a diagonal Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_t^\top, \sigma_h^2\mathbf{I})$ with $\boldsymbol{\mu}_t$ as mean, hyperparameter σ_h as variance ($\sigma_h = 0.1$ by default), and $\mathbf{I}$ as the identity matrix, we get action $a_t^{\text{sel}} = (a_t^{\text{sel}_1}, a_t^{\text{sel}_2}, \cdots, a_t^{\text{sel}_k})$. Using 0.5 as the threshold, the i-th segment is selected if its value $a_t^{\text{sel}_i} > 0.5$; otherwise, it is discarded. The selected segments are then integrated to form $o_t^{\text{sel}} \in \mathbb{R}^{C\times W\times H}$, representing the union of focused objects:

$$o_t^{\text{sel}} = \bigcup_{i\in\mathcal{I}_t} o_t^{\text{seg}_i}, \quad \text{where } \mathcal{I}_t = \left\{ i \mid i \in \{1,\cdots,k\},\ a_t^{\text{sel}_i} > 0.5 \right\}. \tag{2}$$

By obtaining o_t^{sel} through the above sampling method, we can effectively explore different segment selection patterns and prevent premature convergence to local optima.

4.2.2 Tracking pathway

While the selection pathway alone is sufficient to derive o_t^{sel}, this pathway suffers from a major issue: the inconsistency in segment selection. A segment selector that has not been sufficiently trained to develop a stable selection pattern is likely to produce results lacking short-term consistency, resulting in inconsistent actions and low-quality reward signals. Therefore, we propose another approach to derive o_t^{sel}, the tracking pathway. Whenever we obtain the selected objects in the selection pathway, we simultaneously record them in the tracking model. The tracking model then recognizes the selected objects o_t^{sel} using previous selection results.

Till now, we have fully introduced the two pathways for selecting images. We propose a simple yet effective mechanism to combine these two pathways. We introduce a selection interval T_{sel} ($T_{\text{sel}} = 20$ by default). If the current timestep t is divisible by T_{sel}, the selection pathway is called; otherwise, the tracking pathway is used. In a corner case where no segment is selected by π^h, tracking must also be refreshed in the subsequent timestep, regardless of T_{sel}.

4.3 Reuse stage

Before adaptation, the original policy π^{ori} is trained in a clean environment without visual perturbation using the DrQ-v2 algorithm [66]. In the reuse stage, π^{ori} is copied and frozen as the lower-level policy π^l. As a common practice in visual RL, the last 3 frames of focused objects $o_{t-2}^{\text{sel}}, o_{t-1}^{\text{sel}}, o_t^{\text{sel}}$ are stacked and fed into π^l, yielding action a_t for environment interaction to obtain reward r_t. Notably, the focus stage and the reuse stage are highly decoupled, meaning that our method is compatible with a range of RL algorithms. Additionally, our method requires no modifications to π^l.

4.4 Segment selector trained with supervised learning and RL

We introduce the training process of FTR in this section. During training, only the parameters of the segment selector π^h are updated, while those of the segmentation model, tracking model, and low-level policy π^l remain frozen. The training objective of the segment selector consists of two parts: the first is a supervised learning objective based on a VLM (blue dashed line in Fig. 2); the second is an RL objective based on environmental rewards (green dashed line in Fig. 2). Recalling the way humans handle complex environments, as mentioned in Section 1, supervised learning can be seen as leveraging prior knowledge, while RL resembles adjusting based on environmental feedback.

4.4.1 Supervised learning objective

We use the discrepancy between the VLM's selection and the output of π^h as the supervised loss. Given segments o_t^{seg}, an example image from the source domain, and output format, the VLM returns $\mathbf{y}_t = (y_t^1, y_t^2, \cdots, y_t^k)$, where $y_t^i \in \{0, 1\}$, indicating whether each segment should be focused on. $(o_t, o_t^{\text{seg}}, \mathbf{y}_t)$ are added to $\mathcal{D}_{\text{SL}}$, the supervision dataset, for training. We provide more details about VLM in Appendix A.5.

With the selection result of VLM, an intuitive way of training is to use the Binary Cross Entropy (BCE) loss. However, we point out that the rigidity binarize of $\boldsymbol{\mu}_t$ will degrade the subsequent RL optimization. Instead, we use a "softer" margin-regularized loss function:

$$\mathcal{L}_{\text{SL}}(\delta) = \mathbf{E}_{\substack{(o,o^{\text{seg}},\mathbf{y})\sim\mathcal{D}_{\text{SL}} \\ a^{\text{sel}}\sim\pi^h(\cdot|o,o^{\text{seg}})}} \Big[\mathbf{y}\cdot\max(0, 0.5+\delta-a^{\text{sel}}) + (\mathbf{1}-\mathbf{y})\cdot\max(0, a^{\text{sel}}-(0.5-\delta))\Big], \quad (3)$$

where δ is a predefined margin hyperparameter ($\delta = 0.1$ by default). For positive samples ($y^i = 1$), the loss penalizes predictions a^{sel_i} that fall below $0.5 + \delta$ with linearly increasing loss. For negative samples ($y^i = 0$), it penalizes predictions exceeding $0.5 - \delta$.

4.4.2 RL objective

Although the VLM exhibits impressive reasoning capabilities, its predictions can be unreliable in certain scenarios, and its latency of several seconds is unacceptable for real-time control tasks. These limitations motivate the integration of RL to refine the policy. Specifically, π^h is optimized using the PPO algorithm within a hierarchical framework. The state, action, and reward of π^h are:

- **State** includes observation o_t and segments o_t^{seg} when π^h is called on the selection pathway.
- **Action** is the selection vector a_t^{sel} sampled from $\mathcal{N}(\boldsymbol{\mu}_t^\top, \sigma_h^2\mathbf{I})$.
- **Reward** is the cumulative reward $\sum_{\tau=t}^{t+T_{\text{sel}}-1} r_\tau$ between two consecutive invocations of π^h.

Based on the MDP formulation, we train the high-level policy π^h with the PPO algorithm, and the loss function $\mathcal{L}_{\text{RL}}$ is defined in Eq. 1. The RL objective relies on a simple yet valid assumption: the higher the accuracy of the segment selector, the greater the cumulative reward obtained by the low-level policy, and vice versa.

4.4.3 Combination of the objectives

Both objectives involve trade-offs: supervised learning enables rapid convergence but is limited by the VLM's accuracy and latency, whereas RL can achieve better performance but may result in slower training. The total loss function is defined as follows:

$$\mathcal{L} = \eta_{\text{SL}}\mathcal{L}_{\text{SL}} + \eta_{\text{RL}}\mathcal{L}_{\text{RL}}, \quad (4)$$

where the weights of $\mathcal{L}_{\text{SL}}$ and $\mathcal{L}_{\text{RL}}$ are η_{SL} and η_{RL}, respectively. To complement the two objectives, we propose a dynamic adjustment mechanism of η_{SL} and η_{RL}. This strategy allows the training to benefit from supervised learning for fast convergence in the early phase, while gradually shifting towards reinforcement learning for fine-tuning in the later phase. Specifically, when $t < T_1$, $\eta_{\text{SL}} = 1, \eta_{\text{RL}} = 0$. During $T_1 \le t < T_2$, η_{SL} decays linearly from 1 to 0 while η_{RL} rises from 0 to 1, mitigating gradient misalignment between objectives. Note that the VLM is called only when $t < T_1$ with its outputs added to $\mathcal{D}_{\text{SL}}$. When $t > T_2$, $\eta_{\text{SL}} = 0, \eta_{\text{RL}} = 1$. By default, we set $T_1 = 5000$ and $T_2 = 10000$, thereby restricting VLM calls to $T_1/T_{\text{sel}} = 250$ times.

5 Experiments

In this section, we present the experimental results of FTR on 11 tasks, including 8 tasks of DeepMind Control Suite (DMC) [67] and 3 tasks of Franka Emika Robotics [68, 69]. DMC is a widely used benchmark of visual RL, while Franka Emika Robotics can better reflect real-world training scenarios. Finally, we conduct ablation studies to validate each module's effectiveness.

5.1 Experimental setup

We use 8 tasks from DMC: pendulum-swingup (ps), cartpole-swingup (cs), finger-spin (fs), hopper-stand (hs), hopper-hop (hh), cheetah-run (cr), walker-walk (ww), walker-run (wr), and 3 tasks for robotic manipulation: franka-reach (fr), franka-push (fp), and franka-door (fd). These tasks are diverse and comprehensive, ranging from single objects to multiple objects and from easy locomotion tasks to dexterous manipulation tasks.

Our research explores fast adaptation from source to target domains. The source domain corresponds to environments with no background. Policies are trained using the DrQ-v2 algorithm. We perform three independent runs and choose the policy with the best performance as the original policy π^{ori}. To simulate real-world visual disturbances, we use five diverse videos from the DMC-Generalization Benchmark [17], as shown in Fig. 3. These videos cover a range of indoor and outdoor scenes and serve as the backgrounds of five target domains. We perform domain adaptation independently on each target domain and report the average performance.

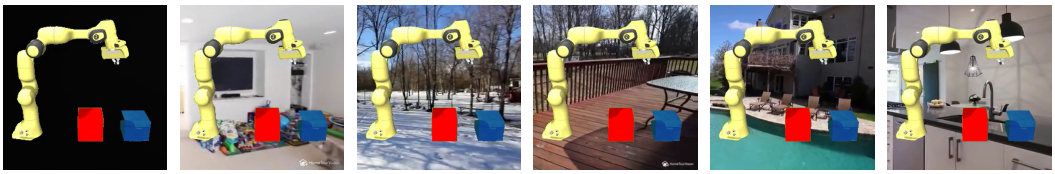


Figure 3: Source domain (leftmost) and five target domains of task franka-push.

We compare FTR with the following baseline methods:

- **DrQ-v2 (clean)** [66]. DrQ-v2 is a classic visual RL algorithm. It is also used as the default method by FTR for training the original policy in the source domain. DrQ-v2 (clean) can be regarded as the potential upper bound for FTR's domain adaptation performance.
- **SimGRL** [13]. SimGRL achieves SOTA performance in visual generalization RL, demonstrating effectiveness in test environments with video backgrounds.
- **PAD** [23]. PAD is a classical visual domain adaptation method in reinforcement learning, and has shown effectiveness in adapting to target domains with simple backgrounds.
- **Q^2-learning** [11]. Q^2-learning is a representation learning-based method for performing RL directly on complex visual inputs.

We also compare against **FTR w/o SL** and **FTR w/o RL**. FTR w/o SL refers to the variant of FTR trained without supervised learning, relying solely on the RL objective. Conversely, FTR w/o RL denotes the variant trained without RL.

FTR and the baseline methods can be categorized into four groups: classical visual RL method DrQ-v2, generalization method that requires no interaction with target domains, adaptation methods that involve limited interaction, and robust training method that is trained directly on the target domains. Although we present the results of all methods in the same table, we clarify their differences:

- **Classical visual RL method (DrQ-v2 (clean))**: We perform three independent runs in the source domain and report the best performance.
- **Generalization method (SimGRL)**: Policies are trained in the source domain for 500k steps across three runs. The trained policies are then evaluated in the five target domains.

- **Adaptation methods (FTR, PAD)**: We first conduct three training runs in the source domain and select the best-performing policy as the initial policy. This policy is then adapted to each of the five target domains using three different seeds for 200k steps.
- **Robust training method (Q^2-learning)**: Policies are trained directly in each target domain for 500k steps.

We report the mean and standard deviation of performance across the five target domains for all methods except DrQ-v2 (clean).

5.2 Experiment results

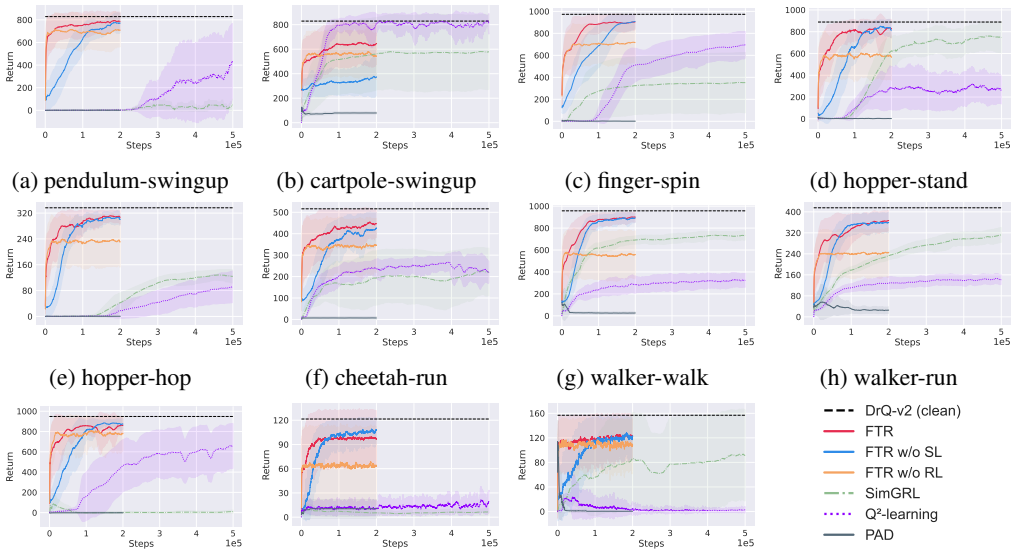
Table 1 and Fig. 4 show the performance for FTR and baseline methods. Table 1 records the final performance. The solid lines and shaded area in Fig. 4 correspond to the mean and variance. FTR achieves the best performance on 10 out of 11 tasks.

Compared to baselines, FTR demonstrates clear advantages in both sample efficiency and performance. In terms of sample efficiency, FTR leverages prior knowledge from the VLM to achieve strong initial performance and improves rapidly using environmental feedback, achieving convergence within 50k steps in most cases. Regarding final performance, FTR outperforms baselines on all tasks except cartpole-swingup. On average across all tasks, FTR retains over 85% of the potential upper bound represented by DrQ-v2 (clean). On the most complex robotic manipulation tasks, FTR shows considerable advantages over other methods. SimGRL, the SOTA visual generalization RL method, performs decently on most tasks, but compared to FTR, it only holds an advantage on the cartpole-swingup task, a task with sparse rewards and challenging object segmentation. This suggests that even with strong augmentation during training, encountering unseen disturbances at deployment can still lead to substantial performance degradation. Similarly, Q^2-learning can obtain a certain level of performance on most tasks. However, its performance remains inferior to FTR except on cartpole-swingup. This indicates that traditional representation learning methods still struggle to handle complex visual perturbations. Despite being a classic visual domain adaptation RL method, PAD performs poorly on all 11 tasks. This indicates that existing adaptation methods struggle with complex, near-real-world target domains with video background.

As an ablation study to evaluate the respective contributions of supervised learning and reinforcement learning in FTR, we compare FTR with its two variants: FTR w/o SL and FTR w/o RL. Comparing FTR and FTR w/o SL, we observe similar final performance; however, FTR converges significantly faster. This suggests that the prior knowledge from VLM offers a well-informed initialization. On the cartpole-swingup task, the performance of FTR w/o SL exhibits a notable performance drop. We speculate that this is due to the task's reward being sensitive to state variations, which causes excessive noise in the RL objective. Comparing FTR and FTR w/o RL, it can be observed that incorporating RL consistently improves performance. This indicates that the supervisory signal alone is insufficient, as errors from the VLM can lead to sub-optimal policies. Therefore, RL is essential for achieving optimal domain-adaptive performance.

Table 1: Performance comparison of FTR and baselines (mean $\pm$ std). Note that we train and evaluate DrQ-v2 in the clean source domain, and test all other methods in visually perturbed target environments. For DrQ-v2 (clean), we report the best performance over three runs as the potential upper bound for other methods.

Task	DrQ-v2 (clean)	PAD	Q^2-learning	SimGRL	FTR w/o RL	FTR w/o SL	FTR (ours)
ps	829.0	0.5 ± 0.4	436.6 ± 331.2	46.1 ± 49.8	708.1 ± 145.5	770.9 ± 85.4	$\mathbf{786.7 \pm 82.3}$
cs	829.3	79.3 ± 8.9	$\mathbf{807.3 \pm 84.1}$	578.6 ± 306.1	541.2 ± 187.2	373.7 ± 152.1	646.0 ± 168.5
fs	973.3	0.8 ± 0.7	695.8 ± 121.8	351.6 ± 273.8	718.0 ± 183.8	$\mathbf{909.1 \pm 62.3}$	903.9 ± 71.2
hs	888.9	2.2 ± 2.2	264.2 ± 125.8	746.9 ± 139.3	563.0 ± 209.3	813.7 ± 68.0	$\mathbf{825.5 \pm 83.6}$
hh	336.4	0.5 ± 0.7	91.7 ± 48.9	124.2 ± 20.3	231.1 ± 77.2	300.9 ± 17.1	$\mathbf{307.3 \pm 20.8}$
cr	516.0	6.8 ± 6.8	221.6 ± 49.8	221.4 ± 110.6	344.9 ± 124.7	427.3 ± 70.5	$\mathbf{446.1 \pm 65.0}$
ww	957.1	26.3 ± 10.5	325.3 ± 62.2	734.0 ± 41.6	558.6 ± 212.2	891.5 ± 41.3	$\mathbf{899.7 \pm 45.4}$
wr	415.4	25.8 ± 12.4	141.5 ± 17.9	312.1 ± 27.8	244.3 ± 94.3	359.2 ± 36.7	$\mathbf{366.5 \pm 37.8}$
fr	948.3	-1.1 ± 6.4	657.7 ± 220.0	10.3 ± 14.9	781.8 ± 161.9	$\mathbf{873.3 \pm 52.0}$	860.5 ± 70.6
fp	121.5	10.4 ± 11.6	19.0 ± 12.9	6.5 ± 4.2	64.2 ± 47.4	$\mathbf{107.4 \pm 15.1}$	96.8 ± 33.8
fd	156.8	0.3 ± 0.4	2.7 ± 4.9	91.5 ± 74.2	108.6 ± 37.2	122.7 ± 28.5	$\mathbf{123.0 \pm 34.9}$



(a) pendulum-swingup (b) cartpole-swingup (c) finger-spin (d) hopper-stand

(e) hopper-hop (f) cheetah-run (g) walker-walk (h) walker-run

(i) franka-reach (j) franka-push (k) franka-door (l) Legends

Figure 4: Training curves on DeepMind Control Suite (a-h) and Franka Emika Robotics (i-k). Note that for the adaptation methods (FTR, PAD), we set the adaptation duration to 200k steps, and for other methods, we train for 500k steps. FTR has converged by 100k steps across all tasks.

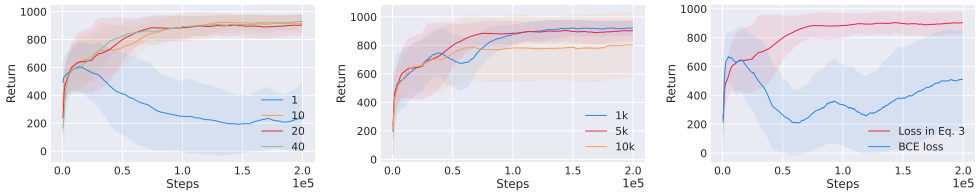
5.3 Further ablation studies

We conduct ablation studies on different selection intervals T_{sel}, SL-to-RL transition timesteps T_1, and supervised learning objectives $\mathcal{L}_{\text{SL}}$ on the finger-spin task.

The results under different values of T_{sel} are in Fig. 5(a). When $T_{\text{sel}} = 1$ (using only the selection pathway while disabling the tracking pathway), the RL process fails to converge, leading to poor final performance. This highlights the critical role of the tracking pathway in FTR. As mentioned in Section 4.2.2, the tracking pathway helps maintain consistency in segment selection, thereby enhancing the stability in RL training. For $T_{\text{sel}} \in \{10, 20, 40\}$, performance exhibits minor variation.

In Fig. 5(b), the performance shows negligible differences between $T_1 = 1000$ and $T_1 = 5000$, demonstrating the effectiveness of our method even with limited supervision from the VLM. When $T_1 = 10000$, a slight performance drop is observed, which may be attributed to overfitting to $\mathcal{D}_{\text{SL}}$.

As shown in Fig. 5(c), using BCE loss as the supervised learning objective leads to a significant degradation in performance, especially during the transition from supervised learning to reinforcement learning. This underscores the importance of the proposed margin-regularized loss in Eq. 3 for ensuring a stable transition from supervised learning to reinforcement learning.



(a) Ablation on T_{sel} (b) Ablation on T_1 (c) Ablation on $\mathcal{L}_{\text{SL}}$

Figure 5: Ablation studies on different selection intervals T_{sel}, SL-to-RL transition timesteps T_1, and supervised learning objectives $\mathcal{L}_{\text{SL}}$ on the finger-spin task.

6 Conclusion

In this work, we propose a novel Focus-Then-Reuse (FTR) framework to achieve rapid policy deployment in real-world environments with background disturbances. The core of FTR lies in training a segment selector using both environmental rewards and VLM's supervision to identify task-relevant objects, while directly applying source domain policy on the filtered visual inputs. The focus stage and the reuse stage are highly decoupled, meaning that FTR is compatible with a range of generalization RL algorithms and has the potential to handle complex distribution shift. Furthermore, we propose a novel object selection mechanism that combines segmentation model and tracking model to improve object selection consistency and enhance the stability in RL training. Experimental results on the DeepMind Control Suite and Franka Emika Robotics indicate that our method effectively synthesizes prior knowledge from the VLM and environmental feedback, demonstrating advantages over baseline methods in performance, efficiency, and interpretability.

For future work, we suggest three aspects worth improving and exploring. First, while FTR demonstrates effective deployment in target domains with visual perturbation, it struggles in additional disturbances such as camera pose variations. A potential solution involves incorporating visual generalization RL approaches to enhance the robustness of the source domain policy. Second, the fixed selection interval T_{sel} relies on manual configuration. Developing an adaptive scheduler for T_{sel} could potentially enhance performance. Finally, the performance of FTR depends on the segmentation and tracking model, suggesting the need for foundation models specifically tailored for downstream RL tasks.

Acknowledgments

We thank the reviewers for their insightful and valuable comments. This work is supported by the National Science Foundation of China (62276126, 62250069).

References

[1] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40, 2016.

[2] Rongrong Liu, Florent Nageotte, Philippe Zanne, Michel de Mathelin, and Birgitta Dresp-Langley. Deep reinforcement learning for the control of robotic manipulation: A focussed mini-review. *Robotics*, 10(1):22, 2021.

[3] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1): 3–20, 2020.

[4] Piotr Mirowski, Razvan Pascanu, Fabio Viola, Hubert Soyer, Andy Ballard, Andrea Banino, Misha Denil, Ross Goroshin, Laurent Sifre, Koray Kavukcuoglu, Dharshan Kumaran, and Raia Hadsell. Learning to navigate in complex environments. In *International Conference on Learning Representations*, 2017.

[5] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *IEEE International Conference on Robotics and Automation*, pages 3357–3364, 2017.

[6] Yinzhen Xu, Weikang Wan, Jialiang Zhang, Haoran Liu, Zikang Shan, Hao Shen, Ruicheng Wang, Haoran Geng, Yijia Weng, Jiayi Chen, et al. Unidexgrasp: Universal robotic dexterous grasping via learning diverse proposal generation and goal-conditioned policy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4737–4746, 2023.

[7] Weikang Wan, Haoran Geng, Yun Liu, Zikang Shan, Yaodong Yang, Li Yi, and He Wang. Unidexgrasp++: Improving dexterous grasping policy learning via geometry-aware curriculum and iterative generalist-specialist learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3891–3902, 2023.

[8] Naomi Chukwurah, Abiodun Sunday Adebayo, and Olanrewaju Oluwaseun Ajayi. Sim-to-real transfer in robotics: Addressing the gap between simulation and real-world performance. *International Journal of Robotics and Simulation*, 6(1):89–102, 2024.

[9] Longchao Da, Justin Turnau, Thirulogasankar Pranav Kutralingam, Alvaro Velasquez, Paulo Shakarian, and Hua Wei. A survey of sim-to-real methods in RL: Progress, prospects and challenges with foundation models. *arXiv preprint arXiv:2502.13187*, 2025.

[10] Chao Chen, Jiacheng Xu, Weijian Liao, Hao Ding, Zongzhang Zhang, Yang Yu, and Rui Zhao. Focus-then-decide: Segmentation-assisted reinforcement learning. In *AAAI Conference on Artificial Intelligence*, pages 11240–11248, 2024.

[11] Weijian Liao, Zongzhang Zhang, and Yang Yu. Policy-independent behavioral metric-based representation for deep reinforcement learning. In *AAAI Conference on Artificial Intelligence*, pages 8746–8754, 2023.

[12] Tongzhou Wang, Simon S. Du, Antonio Torralba, Phillip Isola, Amy Zhang, and Yuandong Tian. Denoised MDPs: Learning world models better than the world itself. In *International Conference on Machine Learning*, pages 22591–22612, 2022.

[13] Wonil Song, Hyesong Choi, Kwanghoon Sohn, and Dongbo Min. A simple framework for generalization in visual RL under dynamic scene perturbations. In *Advances in Neural Information Processing Systems*, pages 121790–121826, 2024.

[14] Roberta Raileanu, Maxwell Goldstein, Denis Yarats, Ilya Kostrikov, and Rob Fergus. Automatic data augmentation for generalization in reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 5402–5415, 2021.

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

[16] Nicklas Hansen, Hao Su, and Xiaolong Wang. Stabilizing deep q-learning with convnets and vision transformers under data augmentation. In *Advances in Neural Information Processing Systems*, pages 3680–3693, 2021.

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

[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 *International Conference on Machine Learning*, pages 3088–3099, 2021.

[19] 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 Cision and Pattern Recognition*, pages 6677–6687, 2021.

[20] David Bertoin, Adil Zouitine, Mehdi Zouitine, and Emmanuel Rachelson. Look where you look! Saliency-guided q-networks for generalization in visual reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 30693–30706, 2022.

[21] Bram Grooten, Tristan Tomilin, Gautham Vasan, Matthew E Taylor, A Rupam Mahmood, Meng Fang, Mykola Pechenizkiy, and Decebal Constantin Mocanu. MaDi: Learning to mask distractions for generalization in visual deep reinforcement learning. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 733–742, 2024.

[22] Di Zhang, Bowen Lv, Hai Zhang, Feifan Yang, Junqiao Zhao, Hang Yu, Chang Huang, Hongtu Zhou, Chen Ye, et al. Focus on what matters: Separated models for visual-based RL generalization. In *Advances in Neural Information Processing Systems*, pages 116960–116986, 2024.

[23] Nicklas Hansen, Rishabh Jangir, Yu Sun, Guillem Alenyà, Pieter Abbeel, Alexei A Efros, Lerrel Pinto, and Xiaolong Wang. Self-supervised policy adaptation during deployment. In *International Conference on Learning Representations*, 2021.

[24] Sizhe Yang, Yanjie Ze, and Huazhe Xu. MoVie: Visual model-based policy adaptation for view generalization. In *Advances in Neural Information Processing Systems*, pages 21507–21523, 2023.

[25] Biwei Huang, Fan Feng, Chaochao Lu, Sara Magliacane, and Kun Zhang. AdaRL: What, where, and how to adapt in transfer reinforcement learning. In *International Conference on Learning Representations*, 2022.

[26] Robert Kirk, Amy Zhang, Edward Grefenstette, and Tim Rocktäschel. A survey of zero-shot generalisation in deep reinforcement learning. *Journal of Artificial Intelligence Research*, 76: 201–264, 2023.

[27] Courtney Stevens and Daphne Bavelier. The role of selective attention on academic foundations: A cognitive neuroscience perspective. *Developmental cognitive neuroscience*, pages S30–S48, 2012.

[28] Jeremy Roschelle. Learning in interactive environments: Prior knowledge and new experience. *Museums as Institutions for Personal Learning*, 1999.

[29] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[30] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, et al. DeepSeek-R1: incentivizes reasoning in llms through reinforcement learning. *Nature*, 645:633–638, 2025.

[31] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[32] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. In *International Conference on Learning Representations*, 2025.

[33] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-VL: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023.

[34] Muhammad Awais, Muzammal Naseer, Salman Khan, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Foundation models defining a new era in vision: a survey and outlook. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 2245–2264, 2025.

[35] Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangjing Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, et al. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *International Journal of Machine Learning and Cybernetics*, pages 1–65, 2024.

[36] Amy Zhang, Rowan Thomas McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations*, 2021.

[37] Xiang Fu, Ge Yang, Pulkit Agrawal, and Tommi Jaakkola. Learning task informed abstractions. In *International Conference on Machine Learning*, pages 3480–3491, 2021.

[38] Minting Pan, Xiangming Zhu, Yitao Zheng, Yunbo Wang, and Xiaokang Yang. Model-based reinforcement learning with isolated imaginations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):2788–2803, 2023.

[39] Miles Richard Hutson, Isaac Kauvar, and Nick Haber. Policy-shaped prediction: Avoiding distractions in model-based reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 13124–13148, 2024.

[40] Rishabh Agarwal, Marlos C Machado, Pablo Samuel Castro, and Marc G Bellemare. Contrastive behavioral similarity embeddings for generalization in reinforcement learning. In *International Conference on Learning Representations*, 2021.

[41] Donghu Kim, Hojoon Lee, Kyungmin Lee, Dongyoon Hwang, and Jaegul Choo. Investigating pre-training objectives for generalization in vision-based reinforcement learning. In *International Conference on Machine Learning*, pages 24294–24326, 2024.

[42] Denis Yarats, Amy Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and Rob Fergus. Improving sample efficiency in model-free reinforcement learning from images. In *AAAI Conference on Artificial Intelligence*, pages 10674–10681, 2021.

[43] Jinwei Xing, Takashi Nagata, Kexin Chen, Xinyun Zou, Emre Neftci, and Jeffrey L Krichmar. Domain adaptation in reinforcement learning via latent unified state representation. In *AAAI Conference on Artificial Intelligence*, pages 10452–10459, 2021.

[44] Dibya Ghosh, Jad Rahme, Aviral Kumar, Amy Zhang, Ryan P Adams, and Sergey Levine. Why generalization in RL is difficult: Epistemic pomdps and implicit partial observability. In *Advances in Neural Information Processing Systems*, pages 25502–25515, 2021.

[45] Bonnie Li, Vincent François-Lavet, Thang Doan, and Joelle Pineau. Domain adversarial reinforcement learning. *arXiv preprint arXiv:2102.07097*, 2021.

[46] Lanqing Li, Rui Yang, and Dijun Luo. FOCAL: Efficient fully-offline meta-reinforcement learning via distance metric learning and behavior regularization. In *International Conference on Learning Representations*, 2021.

[47] Kate Rakelly, Aurick Zhou, Chelsea Finn, Sergey Levine, and Deirdre Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International Conference on Machine Learning*, pages 5331–5340, 2019.

[48] Yossi Gandelsman, Yu Sun, Xinlei Chen, and Alexei Efros. Test-time training with masked autoencoders. In *Advances in Neural Information Processing Systems*, pages 29374–29385, 2022.

[49] Fan-Ming Luo, Shengyi Jiang, Yang Yu, Zongzhang Zhang, and Yi-Feng Zhang. Adapt to environment sudden changes by learning a context sensitive policy. In *AAAI Conference on Artificial Intelligence*, pages 7637–7646, 2022.

[50] Xinyu Zhang, Wenjie Qiu, Yi-Chen Li, Lei Yuan, Chengxing Jia, Zongzhang Zhang, and Yang Yu. Debiased offline representation learning for fast online adaptation in non-stationary dynamics. In *International Conference on Machine Learning*, pages 59741–59758, 2024.

[51] Lin Yen-Chen, Maria Bauza, and Phillip Isola. Experience-embedded visual foresight. In *Conference on Robot Learning*, pages 1015–1024, 2019.

[52] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[53] Sherry Yang, Ofir Nachum, Yilun Du, Jason Wei, Pieter Abbeel, and Dale Schuurmans. Foundation models for decision making: Problems, methods, and opportunities. *arXiv preprint arXiv:2303.04129*, 2023.

[54] Che Wang, Xufang Luo, Keith Ross, and Dongsheng Li. Vrl3: A data-driven framework for visual deep reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 32974–32988, 2022.

[55] Zhecheng Yuan, Zhengrong Xue, Bo Yuan, Xueqian Wang, Yi Wu, Yang Gao, and Huazhe Xu. Pre-trained image encoder for generalizable visual reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 13022–13037, 2022.

[56] Junjie Zhang, Chenjia Bai, Haoran He, Zhigang Wang, Bin Zhao, Xiu Li, and Xuelong Li. SAM-E: Leveraging visual foundation model with sequence imitation for embodied manipulation. In *International Conference on Machine Learning*, pages 58579–58598, 2024.

[57] Ziyu Wang, Yanjie Ze, Yifei Sun, Zhecheng Yuan, and Huazhe Xu. Generalizable visual reinforcement learning with segment anything model. *arXiv preprint arXiv:2312.17116*, 2023.

[58] Kyungmin Kim, JB Lanier, and Roy Fox. Make the pertinent salient: Task-relevant reconstruction for visual control with distractions. In *Reinforcement Learning Conference*, 2025.

[59] Junyao Shi, Jianing Qian, Yecheng Jason Ma, and Dinesh Jayaraman. Composing pre-trained object-centric representations for robotics from" what" and" where" foundation models. In *IEEE International Conference on Robotics and Automation*, pages 15424–15432, 2024.

[60] Yufei Wang, Zhanyi Sun, Jesse Zhang, Zhou Xian, Erdem Biyik, David Held, and Zackory Erickson. RL-VLM-F: Reinforcement learning from vision language foundation model feedback. In *International Conference on Machine Learning*, pages 51484–51501, 2024.

[61] Zilin Huang, Zihao Sheng, Yansong Qu, Junwei You, and Sikai Chen. VLM-RL: A unified vision language models and reinforcement learning framework for safe autonomous driving. *arXiv preprint arXiv:2412.15544*, 2024.

[62] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[63] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations*, 2016.

[64] Abdul Mueed Hafiz and Ghulam Mohiuddin Bhat. A survey on instance segmentation: State of the art. *International Journal of Multimedia Information Retrieval*, 9(3):171–189, 2020.

[65] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *ACM Computing Surveys*, 38(4):13, 2006.

[66] Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. In *International Conference on Learning Representations*, 2022.

[67] 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.

[68] Zhecheng Yuan, Sizhe Yang, Pu Hua, Can Chang, Kaizhe Hu, and Huazhe Xu. RL-ViGen: A reinforcement learning benchmark for visual generalization. In *Advances in Neural Information Processing Systems*, pages 6720–6747, 2023.

[69] Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Soroush Nasiriany, and Yifeng Zhu. Robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020.

Appendix

A Implementation Details

In this section, we describe the implementation details of our work.

A.1 FTR implementation

The source code is available at `https://github.com/LAMDA-RL/FTR`. The code is modified from DMC-Generalization Benchmark [17] and FTD [10]. The PPO algorithm used in FTR is implemented based on `https://iclr-blog-track.github.io/2022/03/25/ppo-implementation-details/`. The DrQ-v2 algorithm is implemented based on `https://github.com/facebookresearch/drqv2`. For details on hyperparameters and network architecture, please refer to the "Hyperparameters" section and the "Network Architecture" section. Algorithm for FTR is shown in Algorithm 1. Most experiments are conducted on a server outfitted with 2 AMD EPYC 7542 32-Core Processor CPUs, 504GB of RAM, and 8 GPUs, each with a performance of over 35 TFLOPS, running Ubuntu 22.04. Training in the source domain using the DrQ-v2 algorithm for 500k steps takes about 1 day. Adapting in the target domain for 200k steps takes about 8 hours.

Algorithm 1 Focus-Then-Reuse

Initialize: segmentation model, tracking model, VLM, segment selector π^h, original policy trained in clean environment π^{ori}, replay buffer $\mathcal{D}_{\text{RL}}$, VLM supervision dataset $\mathcal{D}_{\text{SL}}$
for $t = 1$ to N **do**
  if $t \,\%\, T_{\text{sel}} == 0$ **then**
    use the segmentation model to generate o_t^{seg} from o_t
    generate high-level action $a_t^{\text{sel}} = \pi^h(o_t, o_t^{\text{seg}})$
    select o_t^{sel} according to a_t^{sel}
    initialize the tracking model with o_t^{sel} and o_t
    if $t < T_1$ **then**
      gain VLM's supervision information $\mathbf{y}_t$
      add $\langle o_t, o_t^{\text{seg}}, \mathbf{y}_t \rangle$ to $\mathcal{D}_{\text{SL}}$
    end if
    add $\langle \left\{ o_{t-T_{\text{sel}}}, o_{t-T_{\text{sel}}}^{\text{seg}} \right\}, \left\{ o_t, o_t^{\text{seg}} \right\}, a_{t-T_{\text{sel}}}^{\text{sel}}, \sum_{\tau=t-T_{\text{sel}}}^{t-1} r_\tau \rangle$ to $\mathcal{D}_{\text{RL}}$
  else
    use the tracking model to recognize o_t^{sel} from o_t
  end if
  input $(o_{t-2}^{\text{sel}}, o_{t-1}^{\text{sel}}, o_t^{\text{sel}})$ into π^{ori} to obtain a_t
  take action a_t in the environment to receive the next observation o_{t+1} and reward r_t
  sample a batch $\mathcal{B}_{\text{SL}}$ from $\mathcal{D}_{\text{SL}}$ and calculate $\mathcal{L}_{\text{SL}}$ according to Eq. 3
  sample a batch $\mathcal{B}_{\text{RL}}$ from $\mathcal{D}_{\text{RL}}$ and calculate $\mathcal{L}_{\text{RL}}$ according to Eq. 1
  calculate $\mathcal{L}$ according to Eq. 4 and update π^h
  update η_{SL} and η_{RL}
end for

Segment Anything Model 2 (SAM 2) serves as the default segmentation model and tracking model in FTR. The official implementation of SAM 2 only supports offline video object tracking, not live streaming video. Therefore, we utilize the implementation from `https://github.com/Gy920/segment-anything-2-real-time`, which makes SAM 2 possible for real-time video applications. To improve training speed and memory efficiency, we align with FTD's preprocessing [10] to heuristically filter the segmentation model's outputs into $k = 9$ instances.

The default VLM used in FTR is Qwen-VL-Max [33]. For more details, please refer to the "VLM Details" section.

A.2 Baseline implementation

SimGRL and PAD are implemented following the open-source code, Table 2 includes the links of them. Q^2-learning is implemented according to the original paper, and is included in our code.

Table 2: Links to the open-source code of baseline methods.

Method	Open-Source URL
SimGRL	`https://github.com/W-Song11/SimGRL-Code`
PAD	`https://github.com/nicklashansen/dmcontrol-generalization-benchmark`

A.3 Hyperparameters

Table 3 shows the hyperparameters for reproducing the experiments.

Table 3: Hyperparameters.

Hyperparameters of environments	
frame size	168×168 (franka-push, franka-door), 84×84 (otherwise)
frame stack	3
episode length	200 (franka-push, franka-door), 1000 (otherwise)
action repeat	2 (finger-spin, pendulum-swingup), 4 (otherwise)
Hyperparameters of DrQ-v2	
train steps	5×10^5
replay buffer size	1×10^5
exploration steps	1×10^4
n-step returns	3
batch size	256
optimizer	Adam
actor & critic learning rate	1×10^{-4}
discount factor	0.99
critic Q-function soft-update rate τ	0.01
exploration stddev. clip	0.3
exploration stddev. schedule	linear(1.0,0.1,100000)
Hyperparameters of focus stage	
SAM 2 checkpoint	sam2_hiera_tiny
adapt steps	2×10^5
number of segments k	9
selection interval T_{sel}	20
SL-to-RL transition timestep T_1	5000
transition end timestep T_2	10000
policy stddev. σ_h	0.1
optimizer	Adam
batch size	128
learning rate	3×10^{-4}
clip ratio of PPO	0.2
discount factor	0.5
GAE lambda	0.95
$\mathcal{L}_{\text{SL}}$ objective margin δ	0.1

A.4 Network architecture

Below are the network architectures of the main components of FTR. Here, `MLP(n)` denotes a fully-connected layer with output size of n; `LayerNorm()` denotes applying layer normalization; `Conv2D(c, k, s, p)` denotes a 2D convolution layer of output channel c, kernel size k, stride s, and padding p; `Maxpool2D(k, s)` denotes a 2D max-pooling layer of kernel size k and stride s; `Flatten()` denotes a flatten layer; `ReLU()` denotes a rectified linear unit; `Tanh()` denotes a hyperbolic tangent function; $\{\cdots\} \times k$ denotes repeating the layers within brace for k times.

A.4.1 Network architecture of the original policy

The original policy π^{ori} is trained in the clean environment without visual perturbation using the DrQ-v2 algorithm. The actor and critic share the same image encoder.

Encoder:

`Conv2D(32, 3, 2, 0) => ReLU() => {Conv2D(32, 3, 1, 0) => ReLU()}`$\times 3$ `=> Flatten()`

Actor:

`MLP(50) => LayerNorm() => Tanh() => {MLP(1024) => ReLU()}`$\times 2$ `=> MLP(action_dim)`

Critic:

`MLP(50) => LayerNorm() => Tanh() => {MLP(1024) => ReLU()}`$\times 2$ `=> MLP(1)`

A.4.2 Network architecture of the segment selector

The segment selector (high-level policy π^h) is trained with the PPO algorithm. The actor π^h and value function V share the same embedding module ϕ.

Embedding module ϕ:

`Conv2D(32,3,2,1) => {ReLU() => Conv2D(32,3,1,1) => Maxpool2D(2,2)}`$\times 4$ `=> Flatten() => MLP(128)`

Actor π^h:

π^h adopts an attention-like mechanism to capture the relationship between o_t^{seg} and o_t. First, the inputs are transformed into latent representation $\phi(o_t) \in \mathbb{R}^{1\times D}$ and $\phi(o_t^{\text{seg}}) \in \mathbb{R}^{k\times D}$ through an embedding module ϕ, where D is the dimension of the latent space. Next, $\phi(o_t)$ undergoes linear projection to generate query vector $\mathbf{q}_t \in \mathbb{R}^{1\times D}$, while $\phi(o_t^{\text{seg}})$ are mapped to key vectors $\mathbf{k}_t \in \mathbb{R}^{k\times D}$. Scaled dot-product scores are computed between $\mathbf{q}_t$ and $\mathbf{k}_t$:

$$\text{score}_t = \frac{\mathbf{q}_t \mathbf{k}_t^\top}{\sqrt{D}} \in \mathbb{R}^{1\times k}. \tag{5}$$

Linear projection W_i^k and $W_i^q, (i = 1, 2, 3, 4)$: `MLP(128)`

The scores can be interpreted as the relevance of k segments to the task, where a higher value indicates stronger task relevance of the corresponding segments. Then, a sigmoid function is used to generate the probability of sampling each segment:

$$\boldsymbol{\mu}_t = (\mu_t^1, \mu_t^2, \cdots, \mu_t^k) = \text{Sigmoid}(\text{score}_t) \in (0, 1)^k. \tag{6}$$

Value V:

V adopts an attention mechanism. Given latent representation $\phi(o_t)$ and $\phi(o_t^{\text{seg}})$ through the embedding module ϕ, $\phi(o_t)$ undergoes linear projection to generate query vector, while $\phi(o_t^{\text{seg}})$ are mapped to key vectors and value vectors.

Linear projection W_i^k and $W_i^q, (i = 1, 2, 3, 4)$, W^v: `MLP(128)`

The outputs of the attention mechanism are fed into a value head to get the value of current state.

Value head: `MLP(1)`

A.5 VLM details

We use Qwen-VL-Max [33] by default. Given segments o_t^{seg} and prompt, the VLM returns $\mathbf{y}_t = (y_t^1, y_t^2, \cdots, y_t^k), y_t^i \in \{0, 1\}$, indicating whether each segment should be focused on. The prompt template is shown in Fig. 6, including an example image from source domain, segments o_t^{seg} and output format. The example images for each task used in the prompt are shown in Fig. 7. The total cost of API calls is below \$100 for our experiment. The average response time for API calls is

approximately 7 seconds. The number of VLM calls for a single run in our experiment is $T_1/T_{\text{sel}} = 5000/20 = 250$.

Task: Determine if the object in each of the candidate images 1-9 is part of the *articulated* object(s) in the target image. Note that the articulated object may be in *different poses* or joint configurations across images. Output the results in JSON format.

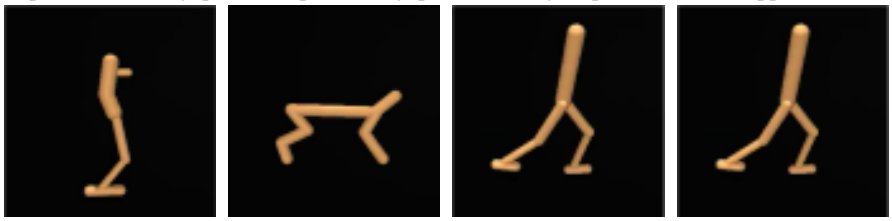
Target Image: [Image Placeholder]
Candidate Image 1: [Image Placeholder]
Candidate Image 2: [Image Placeholder]
Candidate Image 3: [Image Placeholder]
Candidate Image 4: [Image Placeholder]
Candidate Image 5: [Image Placeholder]
Candidate Image 6: [Image Placeholder]
Candidate Image 7: [Image Placeholder]
Candidate Image 8: [Image Placeholder]
Candidate Image 9: [Image Placeholder]

Please return the results in JSON format as an array of objects. **The order of objects in the array must correspond to the order of candidate images provided (from 1 to 9).** Each object should contain the following fields:
- "image_id": Candidate image number (1-9)
- - "is_same_object": Boolean value (true if same object, false if different)

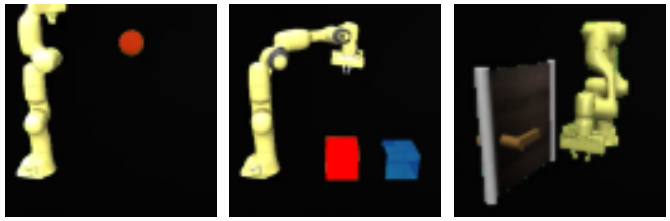
JSON Output Template:
```json[
{ "image_id": "1", "is_same_object": boolean },
{ "image_id": "2", "is_same_object": boolean },
{ "image_id": "3", "is_same_object": boolean },
{ "image_id": "4", "is_same_object": boolean },
{ "image_id": "5", "is_same_object": boolean },
{ "image_id": "6", "is_same_object": boolean },
{ "image_id": "7", "is_same_object": boolean },
{ "image_id": "8", "is_same_object": boolean },
{ "image_id": "9", "is_same_object": boolean }]```

Figure 6: VLM prompt template.

(a) pendulum-swingup (b) cartpole-swingup (c) finger-spin (d) hopper-stand



(e) hopper-hop (f) cheetah-run (g) walker-walk (h) walker-run



(i) franka-reach (j) franka-push (k) franka-door

Figure 7: Source domain of the DeepMind Control Suite (a-h) and Franka Emika Robotics (i-k) tasks.
```

B Additional Experimental Results

B.1 Further ablation studies

We conduct ablation studies on different selection intervals T_{sel}, SL-to-RL transition timesteps T_1, and supervised learning objectives $\mathcal{L}_{\text{SL}}$ on the finger-spin task. Results are shown in Table 4, Table 5, Table 6, and Fig. 5.

Table 4: Ablation on T_{sel} on the finger-spin task.

T_{sel}	Performance (mean $\pm$ std)
1	249.6 ± 244.0
10	915.7 ± 55.9
20	903.9 ± 71.2
40	928.9 ± 47.8

Table 5: Ablation on T_1 on the finger-spin task.

T_1	Performance (mean $\pm$ std)
1000	922.5 ± 44.4
5000	903.9 ± 71.2
10000	802.8 ± 229.5

Table 6: Ablation on $\mathcal{L}_{\text{SL}}$ on the finger-spin task.

$\mathcal{L}_{\text{SL}}$	Performance (mean $\pm$ std)
The proposed loss in Eq. 3	903.9 ± 71.2
Binary Cross Entropy (BCE) loss	510.4 ± 337.8

B.2 Performance of the baselines in the clean environments

Table 7 shows the results of the baselines in the clean environment. The experiments are conducted with three random seeds. PAD and SimGRL perform well on most tasks, while Q^2-learning performs decently except in franka-push and franka-door.

Table 7: Performance of the baselines in the clean environments (mean $\pm$ std).

Task	PAD	Q^2-learning	SimGRL
pendulum-swingup	665.0 ± 437.7	912.0 ± 32.0	910.0 ± 6.0
cartpole-swingup	859.6 ± 14.5	874.2 ± 0.7	862.8 ± 7.2
finger-spin	908.5 ± 18.5	653.2 ± 262.9	979.5 ± 6.4
hopper-stand	814.7 ± 25.6	845.0 ± 18.3	866.2 ± 5.8
hopper-hop	114.6 ± 5.3	170.7 ± 14.0	150.9 ± 5.2
cheetah-run	343.2 ± 36.8	385.5 ± 13.3	318.8 ± 4.6
walker-walk	915.4 ± 20.4	616.9 ± 43.9	879.1 ± 43.9
walker-run	318.3 ± 7.3	251.2 ± 37.3	349.4 ± 11.3
franka-reach	942.7 ± 3.3	884.2 ± 44.6	932.7 ± 22.4
franka-push	102.8 ± 2.1	54.8 ± 5.6	95.3 ± 11.2
franka-door	152.4 ± 4.3	20.4 ± 0.3	163.3 ± 0.8

B.3 VLM accuracy

We calculate the accuracy of Qwen-VL-Max across different tasks in Table 8. The VLM's judgment is considered accurate when it selects task-related segments and refrains from selecting task-irrelevant ones.

Table 8: Accuracy of Qwen-VL-Max.

Task	Accuracy (%)
pendulum-swingup	97.84
cartpole-swingup	88.67
finger-spin	86.40
hopper-stand	94.23
hopper-hop	94.33
cheetah-run	96.32
walker-walk	89.77
walker-run	89.11
franka-reach	89.04
franka-push	85.52
franka-door	91.41

B.4 Relationship between segment selection accuracy and reward

We conduct supplementary experiments on the franka-reach task to validate the relationship between segment selection accuracy and cumulative reward in an episode over five random seeds in Table 9. The results in the table demonstrate a clear correlation: When the segment selector accurately focuses on task-relevant objects, π_l achieves the highest rewards. Conversely, incorrect selection leads to substantially lower rewards.

Table 9: Relationship between segment selection accuracy and performance (mean $\pm$ std).

segment selection	Performance (mean $\pm$ std)
Only task-relevant objects	892.4 ± 22.9
Task-relevant objects + 1 irrelevant object	570.9 ± 223.1
Task-relevant objects + 2 irrelevant objects	495.3 ± 259.3
All objects	234.4 ± 282.9
All task-irrelevant objects	9.6 ± 17.2

B.5 Experiments on more distractions

B.5.1 Task-similar objects

To verify FTR's ability to handle task-similar objects in the target domain that could easily be misidentified, we introduce two scenarios: another cube in the background and another uncontrolled robotic arm in the background. We conduct experiments on the franka-reach task over three random seeds. The results, shown in Table 10, indicate that even with such objects in the background, FTR can still filter out task-relevant objects using environmental reward, thereby maintaining high performance.

Table 10: Performance of FTR on the franka-reach task in face of distracting objects (mean $\pm$ std).

Another cube	Another uncontrolled robotic arm
913.0 ± 18.5	871.8 ± 38.6

B.5.2 Illumination variation

We evaluate FTR's illumination generalization on the franka-reach task over three random seeds. To enhance the adapted policy's performance, as discussed in the conclusion of our paper, we introduce illumination perturbation during low-level policy π_l training to enhance robustness. The results are in Table 11.

Row 1 shows performance without illumination-robust training, while Row 2 demonstrates FTR's adaptation under both illumination and background perturbations, maintaining the performance of π_l despite illumination sensitivity. We introduce varying directions of illumination perturbations

during the training of π_l in the source domain to enhance its robustness. As presented in Row 3, π_l's performance improved significantly. Row 4 shows FTR's adaptation performance under both illumination and background perturbations, demonstrates that **FTR consistently maintained the performance of** π_l. Furthermore, benefiting from the enhanced robustness of π_l, the performance showed a notable improvement compared to Row 2.

Table 11: Performance of FTR in face of illumination variation (mean $\pm$ std).

With background perturbation?	π_l robust to illumination variation?	With FTR adaptation?	Performance (mean $\pm$ std)
no	no	no	659.3 ± 52.5
yes	no	yes	599.1 ± 62.1
no	yes	no	952.3 ± 4.1
yes	yes	yes	929.5 ± 11.6

B.6 Fine-tuning π_l after adaptation

When π_l lacks robustness or is deployed in an unforeseen perturbed environment, we show fine-tuning π_l after adaptation (while fixing the selector) can further improve performance.

To demonstrate this, we conduct experiments on the franka-reach task over three random seeds and introduce two types of target domain perturbations: a 15° rotation of the camera around the z-axis and a 15° horizontal inclination of the robot arm's base. After training the selector for 200k steps, we fix its parameters and unfreeze π_l's parameters to fine-tune π_l for 50k steps using environmental reward. The results are in Table 12. The adapted FTR performance initially degrades. Nevertheless, after the 50k-step fine-tuning, the performance recovers to a commendable level.

Table 12: Performance after adaptation and fine-tuning π_l (mean $\pm$ std).

	After FTR adaptation	After fine-tuing π_l
Camera rotation	733.3 ± 15.7	904.6 ± 8.7
Base inclination	716.2 ± 51.0	906.0 ± 13.7

We posit that the reason for the effectiveness of the "adapt + fine-tune" process and the performance maintenance with non-robust π_l is the same: The selector's RL training does not impose stringent requirements on the optimality of π_l. The selector can be guided towards correct learning as long as π_l satisfies a key condition: the reward for a correctly chosen action by the selector is greater than that for an incorrectly chosen one. As shown in the experiment of Appendix B.4, this condition can be easily satisfied. Once the selector learns the correct selection patterns, fine-tuning π_l based on the filtered images becomes highly efficient.

C Visualization

We visualize o_t, o_t^{sel}, o_t^{seg}, and the selection result of o_t^{seg} across all tasks (Figs. 8 to 18).

We show images from $t = 0$ to $t = 36$, sampled every 4 time steps. Recall that we set selection interval $T_{\text{sel}} = 20$. When $t = 0$ and $t = 20$, the selection pathway is called; otherwise, the tracking pathway is used. In the selection pathway, we displays the segments $o_t^{\text{seg}}, t \in \{0, 20\}$, and their selection probabilities. Here, a segment is selected if its probability > 0.5. The selected segments are then integrated to form $o_t^{\text{sel}}, t \in \{0, 20\}$, representing the union of focused objects. Whenever we obtain the selected objects $o_t^{\text{sel}}, t \in \{0, 20\}$ in the selection pathway, we simultaneously record them in the tracking model. In the tracking pathway, the tracking model recognizes the selected objects o_t^{sel} using observation o_t.

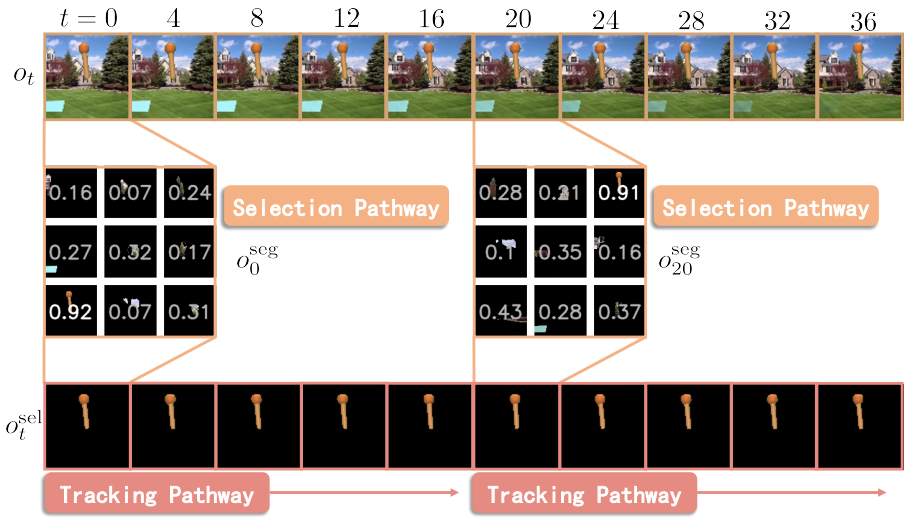


Figure 8: Visualization of pendulum-swingup.

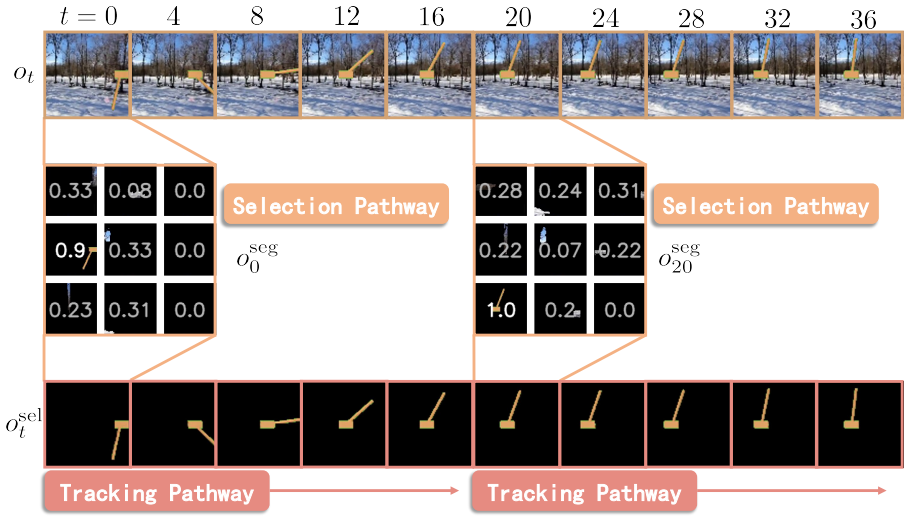


Figure 9: Visualization of cartpole-swingup.

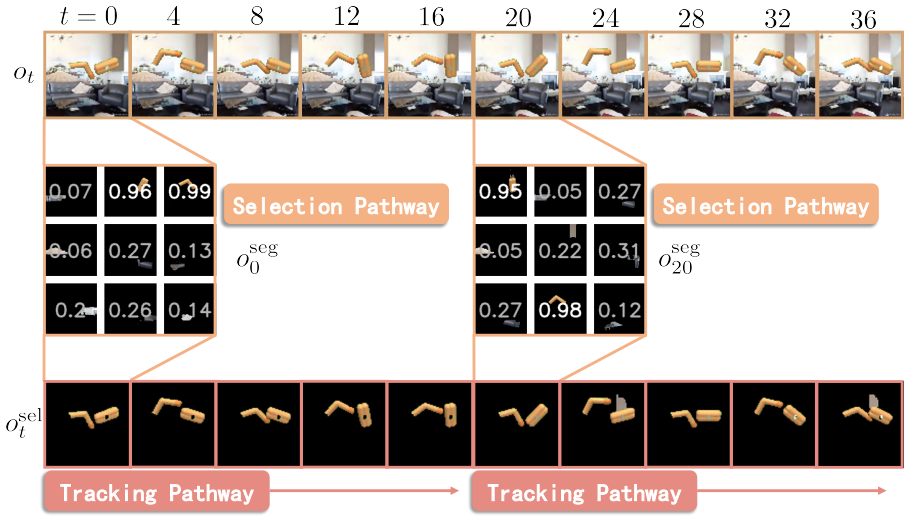


Figure 10: Visualization of finger-spin.

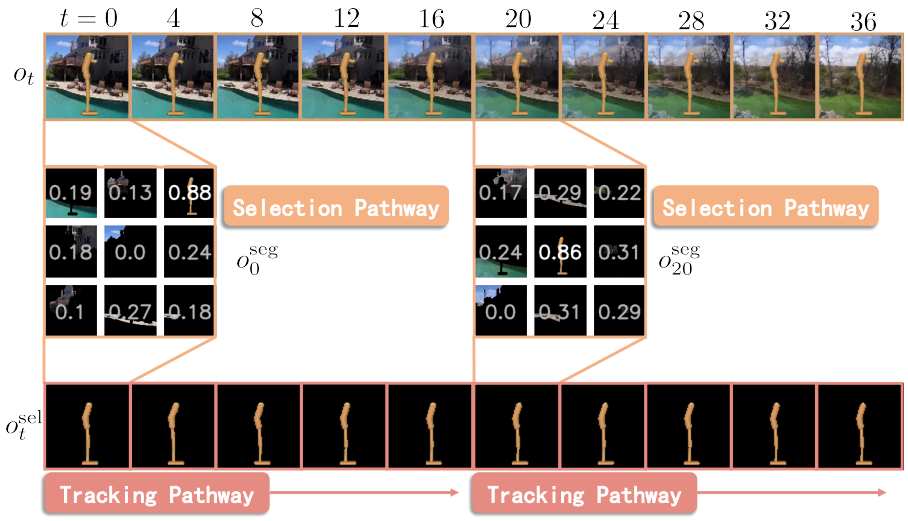


Figure 11: Visualization of hopper-stand.

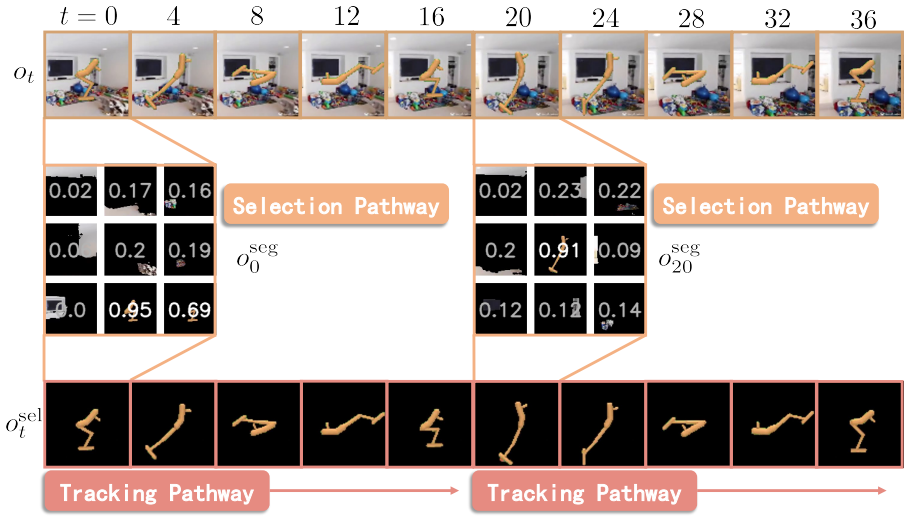


Figure 12: Visualization of hopper-hop.

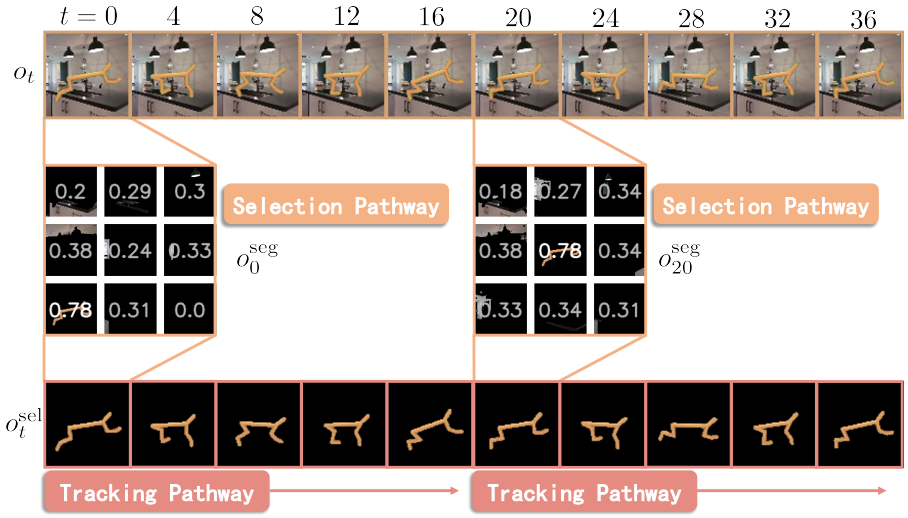


Figure 13: Visualization of cheetah-run.

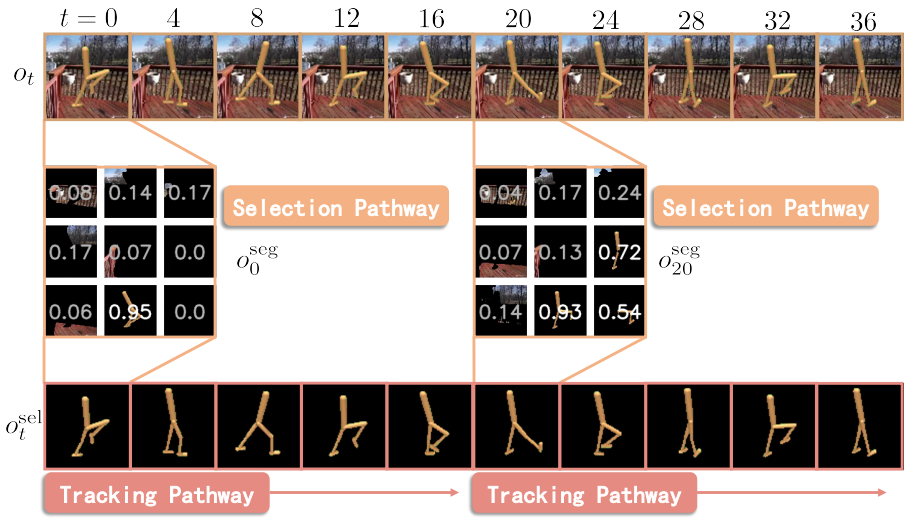


Figure 14: Visualization of walker-walk.

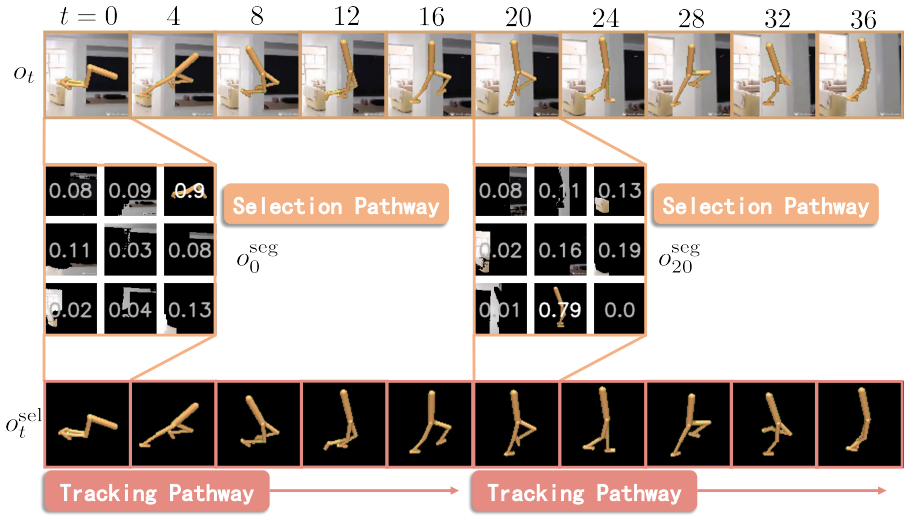


Figure 15: Visualization of walker-run.

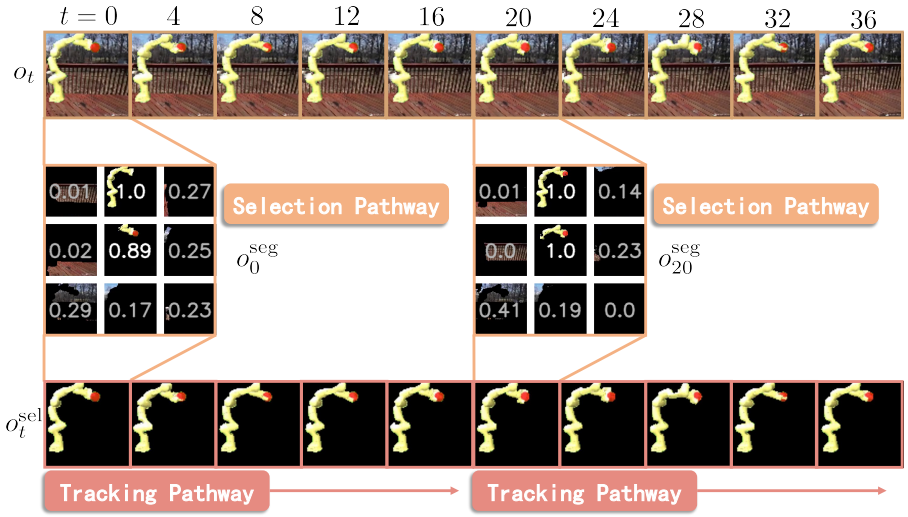


Figure 16: Visualization of franka-reach.

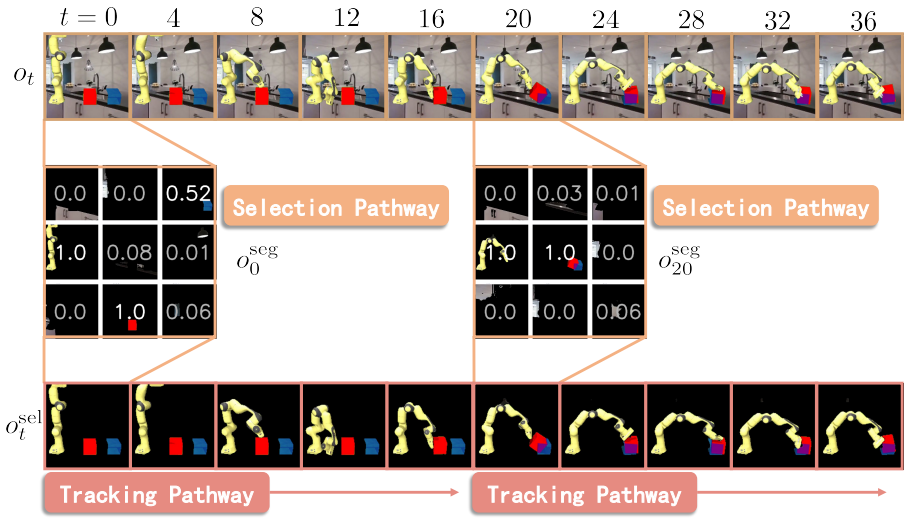


Figure 17: Visualization of franka-push.

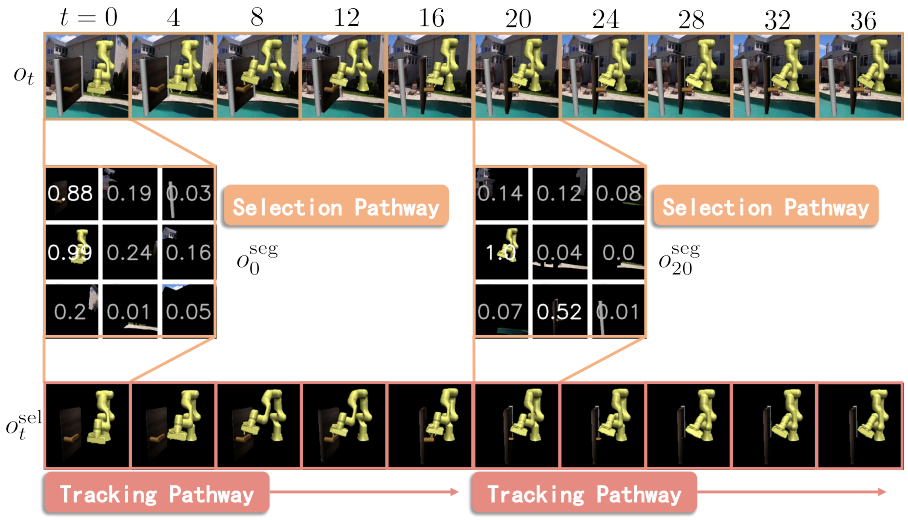


Figure 18: Visualization of franka-door.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction include the claims made in the paper, which accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the possible limitations in Section 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We present the implementation details in Section 4 and Appendix A.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The source code is available at `https://github.com/LAMDA-RL/FTR`.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so No is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We present the experimental setting in Section 5 and Appendix A

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The results and accompanied by error bars.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide sufficient information on the computer resources in Appendix A.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: I have read and followed the ethics guidelines.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Creators or original owners of assets used in the paper, are properly credited in Appendix A.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: The paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: The paper use the Vision-Language Model for the component of the method. Details are provided in Section 4 and Appendix A.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.