

Domain Adaptation of Visual Policies with a Single Demonstration

Weiyao Wang¹ and Gregory D. Hager¹

Abstract—Deploying machine learning algorithms for robot tasks in real-world applications presents a core challenge: overcoming the domain gap between the training and the deployment environment. This is particularly difficult for visuomotor policies that utilize high-dimensional images as input, particularly when those images are generated via simulation. A common method to tackle this issue is through domain randomization, which aims to broaden the span of the training distribution to cover the test-time distribution. However, this approach is only effective when the domain randomization encompasses the actual shifts in the test-time distribution. We take a different approach, where we make use of a single demonstration (a prompt) to learn policy that adapts to the testing target environment. Our proposed framework, PromptAdapt, leverages the Transformer architecture’s capacity to model sequential data to learn demonstration-conditioned visual policies, allowing for in-context adaptation to a target domain that is distinct from training. Our experiments in both simulation and real-world settings show that PromptAdapt is a strong domain-adapting policy that outperforms baseline methods by a large margin under a range of domain shifts, including variations in lighting, color, texture, and camera pose. Videos and more information can be viewed at project webpage: <https://sites.google.com/view/promptadapt>.

I. INTRODUCTION

Reinforcement learning (RL) has demonstrated impressive achievements in complex robotic tasks that use image observations [1], [2], [3]. At the same time, advances in simulation [4], [5] have enabled RL algorithms to overcome their high sample complexity by training in a simulated environment before deployment on a physical robot. However, the discrepancies between the simulated data and the real-world, the well-known sim-to-real gap [6], often leads to poor real-world performance [7]. This is an extreme case of the more general problem of ensuring RL algorithms are able to accommodate domain shifts on their inputs. The present work mainly focuses on robustness under changes in visual appearances, such as changes in lighting and texture, a significant category of environmental shifts within the sim-to-real gap.

A common approach to addressing distribution shift is to attempt to learn a visual policy that is *robust* to appearance changes through domain randomization [8], [9], [10], [11]. Applying a large amount of randomization generally increases (but does not guarantee) the likelihood that visual policy will work in a novel test environment, but can also lead to increased training difficulty and decreased average

policy performance. A variety of other approaches have been proposed to mitigate the difficulty of policy learning [12], [13], [14], [15], [16], [17], [18]. However accommodating large domain shifts between the training environment and the deployment environment, which may be unknown during training, remains a challenge.

Instead of attempting to learn a visual policy that is robust to all unknown environmental changes, we instead pose the following question: *Can a single demonstration serve as a prompt for visual domain adaptation in visuomotor policies?* Our approach to answering this question builds on a related set of ideas that use state-action trajectories, in the form of a demonstration, as a form of context or "prompt" to enable rapid adaptation to new tasks [19], [20], [21]. In the case of domain shift, we formulate the problem as a demonstration-conditioned policy distillation challenge. The demonstration trajectory provides crucial information about both the optimal policy and the transition dynamics within the target domain. In many real-world scenarios, a human operator can offer this brief demonstration to illustrate how to complete the task under the conditions of the target domain.

In brief, in our prompt-based domain adaptation framework, termed PromptAdapt, we first train a teacher policy with a ground truth state vector as input. We then train a Transformer-based student visual policy to imitate the teacher policy’s behavior, conditioned on a demonstration with image observations obtained under various domain randomization perturbations (the prompt). During testing in a new environment, the algorithm is given a corresponding demonstration acquired in the new environment. The agent leverages the observations and policy information in the prompt to adapt to the domain shift and act accordingly. This framework is attractive because the visual policy is domain adapted with only one short trajectory segment and no explicit finetuning. It combines policy distillation and the benefits of the Transformer architecture to efficiently adapt to the target domain.

To validate the performance of our method, we conducted tests in both simulated and real-world environments. Our results indicate that PromptAdapt is highly effective in adapting to visual domain shifts, such as changes in lighting, color, texture, and camera pose. Through extensive experiments, we demonstrate its superior performance over a range of baseline approaches. Our method consistently achieves higher cumulative rewards or success rates for both in-distribution and out-of-distribution shifts from the training set, highlighting its practicality for adapting to unknown environmental changes that may occur in real-world deployments.

This work was supported by the U.S. National Science Foundation grants IIS-1900952 to Johns Hopkins University.

¹W. Wang and Gregory D. Hager are with the Department of Computer Science, Johns Hopkins University, Baltimore, MD 21218, USA wwang121@jhu.edu and hager@cs.jhu.edu

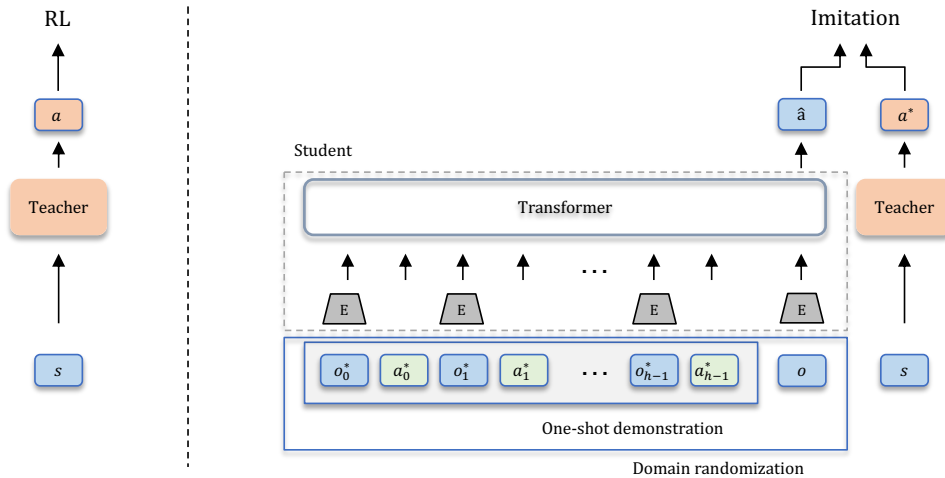


Fig. 1. **Left:** We first train a teacher policy using privileged ground truth state information. **Right:** We then distill the learned policy to a visual input-only student policy through imitation learning. For the student policy, our PromptAdapt architecture leverages a Transformer network to condition on a single demonstration, efficiently adapting to the target domain during testing. Same domain randomization function is applied to images in demonstration and per step observation to enable demonstration conditioned adaptation to visual appearance changes.

II. RELATED WORK

Visual Reinforcement Learning: Visual reinforcement learning has gained significant attention due to its potential applicability in a wide range of real-world tasks. Methods that combine convolutional neural networks (ConvNets) with reinforcement learning algorithms have achieved remarkable success in learning policies directly from raw pixel inputs. More recent work has explored the use of unsupervised or self-supervised auxiliary tasks [22], [23], [24], [25] and data augmentations [3], [26], [27], [28], [29] to learn robust and performing visual policies.

Imitation Learning and Policy Distillation: Imitation learning [13], [14] is an effective approach to transfer expert knowledge to a student policy. Policy distillation [15], [16], [17], [18] is a related technique where a student policy is trained to imitate a teacher policy, often with the goal of compressing or refining the teacher policy. In the context of image based RL, [12] proposes to disentangle between policy learning and visual generalization by policy distillation. In some approaches [17], [30], [31], teacher can access to additional privileged information (e.g., grounded truth state) which student policy does not have. Building upon this framework, our method extends it to adapt visual policies using a single demonstrations.

Domain Transfer for Control Policy: Domain transfer has been extensively studied for control policies, often in the context of sim-to-real transfer. In such scenarios, the goal is to deploy a policy trained in a simulator onto a real physical robot, where there may be discrepancies in physics, configurations, and appearances. One widely used approach to bridge this gap is domain randomization. This strategy aims to reduce the discrepancies between the source (simulated) and target (real) domains [8], [9]. Techniques such as [10], [11] have utilized domain randomization to enhance the generalization capabilities of learned policies. They introduce

variability in the simulation during training to mitigate shifts in factors such as system dynamics [30], [32] and visual appearances [33]. In addition to mere augmentation of the training environment, domain adaptation methods [34], [35] are explored to refine usually state-based policies using test-time trajectory sequences. In contexts closely related to ours, focusing on adapting visual policies, [36] employs inverse dynamics modeling with a self-supervised learning objective to fine-tune the image encoder during test time, thereby enhancing performance. Some recent studies have delved into the use of generative adversarial training to aid sim-to-real transfer [37], [38], [39].

Meta-Learning: Meta-learning aims to quickly adapt to new machine learning tasks with limited data at test time [40], [41], [42], [43]. In the context of reinforcement learning, meta-learning has been used to enable fast adaptation of policies to perform new tasks [43], [44], [45], [46]. Our work is related to these approaches, as we leverage a single demonstrations to adapt visual policies; however, our method does not make weight updates through finetuning. Instead, we focus on the Transformer architecture as a means to effectively condition on the demonstration data for adaptation in an in-context manner.

One-Shot Imitation Learning with Transformer: The Transformer architecture [47], [48] has been highly successful in the field of natural language processing and has been adapted for various other domains, including reinforcement learning and robotic control. Originating in the realm of NLP, the prompt-based approach [49], [50] has gained traction as an effective method for in-context task adaptation. In this approach, a task-specific prompt is added as a prefix to the input sequence, guiding the model toward the desired output. This design has been adopted in the field of robot learning recently, particularly for adaptation to new tasks encountered during testing, and is framed within the context

of one-shot imitation learning. For example, [19] proposes a one-shot imitation learning method that employs the Transformer architecture to learn new tasks based on a single demonstration from an expert. Subsequent work [20] further improves generalization capabilities by incorporating temporal contrastive learning. In the context of offline RL, [21] extends the recently proposed Decision Transformer [51] to enable few-shot generalization for unseen tasks.

The majority of current research in this area studies using a single demonstration to perform unseen testing tasks. However, these approaches often overlook the challenge of adapting to different visual domains, an obstacle that remains significant for important applications such as sim-to-real adaptation. Our aim is to extend the current framework of Transformer-based, demonstration-conditioned policy adaptation to include visual domain adaptation.

III. METHOD

We frame our problem as a single demonstration conditioned domain adaptation problem for visual policies. Formally, consider a Markov decision process (MDP) [52] \mathcal{M} parameterized by the tuple $\langle S, A, O, R, P, \gamma \rangle$, where S and A are the state and action spaces. $P : S \times A \times S \rightarrow [0, 1]$ is the transition function, $R : S \times A \rightarrow \mathbb{R}$ is the scalar reward, and γ is the discount factor. We assume that the desired visual RL agent cannot directly access state space S but only receives higher-dimensional input $o \in O$ (e.g., pixel images) as observations. The source environment \mathcal{M}_{src} and target environment \mathcal{M}_{tgt} differ only in the mapping from state space S to observation space O due to visual appearance shift, while all other elements remain the same.

We assume access to an h -step demonstration trajectory $\tau^* = (o_0^*, a_0^*, \dots, o_{h-1}^*, a_{h-1}^*)$ in the target environment. In this work, we aim to design an algorithm that can efficiently leverage such a single demonstration to perform well on \mathcal{M}_{tgt} , which is not known a priori.

Policy distillation: To concentrate on the issue of domain adaptation through a single demonstration, we divide policy learning and policy adaptation under the framework of policy distillation. In the initial stage, we train a high-performing teacher policy $\pi_t(s) : S \rightarrow A$ using a standard RL algorithm directly from state space. The objective is to maximize cumulative episodic reward in the source environment:

$$J(\pi_t) = \mathbb{E}_{\tau \sim \pi_t}[R(\tau)] = \mathbb{E}_{\tau \sim \pi_t} \left[\sum_{t=0}^T \gamma^t r_t \right], \quad (1)$$

where τ is the rollout trajectory induced by executing teaching policy π_t . For this stage, we choose Soft Actor-Critic (SAC) [53] as it is a state-of-the-art RL algorithm that is widely used in continuous control tasks.

In the second stage, we train a student policy $\pi_s(o; \tau^*)$ to imitate the optimal actions taken by the teacher, given the observation and a single demonstration τ^* , both of which are under the same domain randomization function f_{DR} from a domain randomization function set F_{DR} . This design brings demonstrations and per step observation to the same

Algorithm 1 Domain adapted visual policy via single demonstration

- 1: **Input:** π_t, π_s : trained teacher policy and randomly initialized student policy, \mathcal{D}_{dagger} : empty dagger dataset, F : domain randomization function set, N : number of steps for each trial and K : number of epochs.
 - 2: Roll out π_t to collect initial samples in \mathcal{D}_{dagger}
 - 3: **for** $k = 1, 2, \dots, K$ **do**
 - 4: Sample domain randomization function $f_{DR} \sim F_{DR}$
 - 5: Roll out π_t to get demonstration τ' under f_{DR}
 - 6: **for** $n = 1, 2, \dots, N$ **do**
 - 7: Roll out π_s to collect o', s' samples under f_{DR}
 - 8: Add new data $\mathcal{D}_{dagger} \leftarrow \mathcal{D}_{dagger} \cup \{o', s', \tau'\}$
 - 9: \triangleright Collect data
 - 10: Sample tuple $(o, s, \tau^*) \sim \mathcal{D}_{dagger}$
 - 11: Update π_s to minimize $\|\pi_s(o; \tau^*) - \pi_t(s)\|^2$
 - 12: \triangleright Update student policy
-

randomized visual domain and encourages the student policy to learn to adapt its policy based on the demonstration τ^* .

The student policy is distilled from the teacher using the DAgger (Dataset Aggregation) [13] imitation procedure. To begin with, we initialize a DAgger dataset \mathcal{D}_{dagger} by rolling out the teacher policy. In each iteration, we select a domain randomization function $f_{DR} \in F_{DR}$ and apply it to the policy rollout process to collect teacher demonstration τ' and student rollout observations o', s' from the simulator. The process alternates between (1) updating the student's parameters through gradient descent on a supervised regression loss, and (2) adding more samples to \mathcal{D}_{dagger} using the latest student policy.

Transformer-based prompt-conditioned visual policy:

We build our model based on the Transformer architecture. We first process the visual input through a deep convolutional neural network (ConvNet) to extract features. Given an input observation o , the ConvNet encoder E generates an output representation $E(o) \in \mathbb{R}^{d_e}$, where d_e is the encoder output dimension. We then define two embedding functions, $f_o : \mathbb{R}^{d_e} \rightarrow \mathbb{R}^{d_h}$ and $f_a : \mathbb{R}^{d_a} \rightarrow \mathbb{R}^{d_h}$ to embed the observation features and actions, respectively, to embedding with dimension of d_h . The resulting embeddings $f_o(E(o))$ and $f_a(a)$ are then used as input to the Transformer's encoder layer. To provide sequential information in the trajectory, we incorporate the learnable linear positional encodings $P \in \mathbb{R}^{(h+1) \times d_h}$ as in [51]. We add the positional encodings to the input elements before they are processed by the Transformer encoder, allowing the self-attention mechanism in the Transformer to selectively focus on different parts of the demonstration trajectory and the current observation.

$$\text{Input} = \text{Concat} \left([f_o(E(o_0^*)), f_a(a_0^*), \dots, (f_o(E(o_{h-1}^*)), f_a(a_{h-1}^*)), f_o(E(o))] \right) + P \quad (2)$$

We then predict the next action to take \hat{a} based on the last element of the output from the Transformer. To train

the student policy, we minimize the L_2 distance between the predicted action \hat{a} and the teacher policy’s output a^* , which is obtained by applying the teacher policy to the state s . The loss function L can be formulated as:

$$L(\pi_s) = \|\hat{a} - a^*\|^2 = \|\pi_s(o; \tau^*) - \pi_\tau(s)\|^2, \quad (3)$$

where $\|\cdot\|^2$ denotes the squared L_2 distance. By minimizing the imitation loss, the student policy learns to mimic the behavior of the teacher policy. This learning objective results in adaptive policy capable of effectively extracting knowledge from given demonstration and adjusting policies accordingly.

IV. EXPERIMENTS

A. Settings

Environments: We selected four tasks from the Distracting Control Suite (Distracting CS) [54] and further implemented two tasks for the UR5 robot in both simulation and real world settings to benchmark our algorithm’s performance. Simulation observations can be found in Figure 2.

Based on the DeepMind Control Suite [55], Distracting CS is a collection of continuous control tasks designed to challenge an agent’s ability to learn in the presence of domain shift. This is introduced by varying levels of changes in colors, backgrounds, and camera poses, making it an ideal benchmark for testing the generalization of reinforcement learning algorithms. We selected four tasks: `cartpole swingup`, `ball_in_cup catch`, `finger spin`, and `walker walk`. The evaluation metric is the episode accumulated reward. We treat the distractions (background, color and camera pose) introduced in this benchmark as domain randomization elements. We train with environments using random distractions of 0.2 intensity as defined in [55]. We then test with random distractions of either 0.2 intensity to evaluate in-distribution (out-of-sample) adaption ability or 0.4 intensity to evaluate out-of-distribution adaption ability.

To further evaluate the applicability of our approach to robotic arm manipulation tasks, we conducted tests with two UR5 robot tasks: reach and push. We have simulated environment implemented with the Mujoco physics engine [4] and a physical UR5 robot for real-world experiments under the sim-to-real setting. The evaluation metric is the task success rate. The success of the reach task is defined as moving the end effector within 5 cm of a colored circular chip, while the success of the push task is defined as pushing a block over a finish line. The policy is trained in simulation with domain randomization including lighting conditions, object colors, table textures, and camera poses. During testing in simulation, we evaluate on a random generated environments with either in-distribution or out-of-distribution domain randomization hyperparameters to test domain adaptation abilities. For testing on physical robot, we also replicate the in-distribution and out-of-distribution settings with corresponding colors and camera poses to evaluate the applicability to real-world scenarios.

Baseline Methods: In order to benchmark the efficacy of our method in learning domain-adapting visual policies

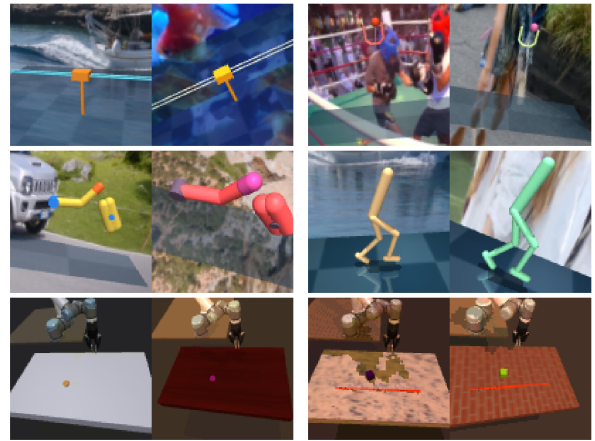


Fig. 2. Sample observations in simulated environments. In-distribution image sample is shown left in each pair and out-of-distribution image sample is shown right. First row: `cartpole swingup`, `ball_in_cup catch`; Second row: `finger spin` and `walker walk`; Third row: UR5 reach and UR5 push.

given demonstration, we compare it against three baseline methods: RL with domain randomization (DR) [10], policy distillation with domain randomization and policy distillation with model-agnostic meta-learning (MAML) [43]. DR serves as a baseline that improves robustness solely by expanding the input observation distribution, without utilizing the demonstration at test time. This technique is combined with direct image based RL with SAC [53] and policy distillation respectively. Also used under policy distillation framework, MAML utilizes the demonstration by adapting the policy’s weights through gradient descent on observation-action pairs from the demonstration. We use the same policy distillation framework as our method: imitation from trained teacher policy using the DAgger [13] algorithm.

Algorithm Details: In our algorithm, PromptAdapt, the teacher policy is learned using SAC [53], with the state vector serving as input. The actor and critic networks are implemented with a multilayer perceptron (MLP) consisting of two hidden layers, each with a size of 1024 units. The teacher policy is trained with 500K environment steps for Distracting CS environments and 100K for UR5 environments. The student policy is constructed using a 3-layer Transformer, accompanied by an 11-layer ConvNet image tokenizer and a linear action tokenizer. To encapsulate temporal information, we stack three consecutive frames for each observation. The student policy is trained with 100K environment steps. During testing in simulation, we use the same teacher policy with groundtruth state access to produce demonstration trajectories; for testing in real-world, we have human tele-operated trajectories as demonstration.

We implement baseline methods using an 11-layer ConvNet, followed by an MLP with two hidden layers, each of size 1024 units. We ensure that the total number of trainable parameters for our method and both baselines are comparable around 2.7M. Both baseline methods are trained by imitating the same teacher policy by 100K steps as our

TABLE I

EVALUATION IN DISTRACTING CS ENVIRONMENTS BY EPISODIC CUMULATIVE REWARDS (MEAN \pm STANDARD DEVIATION)

Task	In-Distribution Reward				Out-of-Distribution Reward			
	RL	Policy Distillation			RL	Policy Distillation		
	DR	DR	MAML	PA (ours)	DR	DR	MAML	PA (ours)
ball_in_cup catch	109 \pm 28	718 \pm 70	773 \pm 80	883 \pm 25	129 \pm 41	372 \pm 61	444 \pm 87	538 \pm 76
cartpole swingup	123 \pm 30	258 \pm 32	234 \pm 75	432 \pm 195	117 \pm 48	222 \pm 23	203 \pm 43	280 \pm 86
finger spin	13 \pm 6	167 \pm 65	192 \pm 79	408 \pm 210	12 \pm 4	25 \pm 14	39 \pm 21	103 \pm 43
walker walk	372 \pm 304	304 \pm 101	329 \pm 91	601 \pm 106	99 \pm 65	145 \pm 32	154 \pm 29	220 \pm 91

method. All experiments are repeated across 4 random seeds and evaluated with 20 episodes for simulated experiments and 10 episodes for real-world experiments.

B. Results

1. *Does the incorporation of a single demonstrations enhance agent performance in both in-distribution and out-of-distribution settings?*

We conduct experiments with Distracting CS benchmarks and simulated UR5 robot tasks to evaluate our method’s performance. As can be seen from the results presented in Table I and Table II, our method consistently outperforms baselines of RL with DR and policy distillation with DR and MAML across all environments, exhibiting superior performance in both in-distribution and out-of-distribution settings. By leveraging the teacher policy embedded in demonstrations, our method is capable of effectively capturing and exploiting the nuances of the visual elements in new environments. The enhanced performance in out-of-distribution settings demonstrates that the agent has strong capability for generalizing to new scenarios.

2. *How does the proposed approach improve generalization across different environmental elements?*

We further examine the effectiveness of our method, PromptAdapt (PA), compared to the baselines in terms of generalization over various out-of-distribution factors in the UR5 simulated environments. As illustrated in Figure 3, our method consistently outperforms baseline methods across multiple out-of-distribution factors. A significant improvement is observed in the camera pose factor, which is presumably the most challenging aspect in the domain difference. The superior performance of PromptAdapt underscores its ability to capture and adapt to complex changes across domains. We leave RL with DR out of this comparison because it does not perform well in in-distribution setting already due to the difficulty in training RL directly under heavy domain changes.

PromptAdapt also demonstrates improved performance in other out-of-distribution factors, such as lighting and texture. These improvements further support that PromptAdapt can adapt to novel situations with different visual conditions. The consistent performance gains across all out-of-distribution factors suggest that PromptAdapt is a promising approach for creating adaptable agents capable of handling complex tasks in diverse environments.

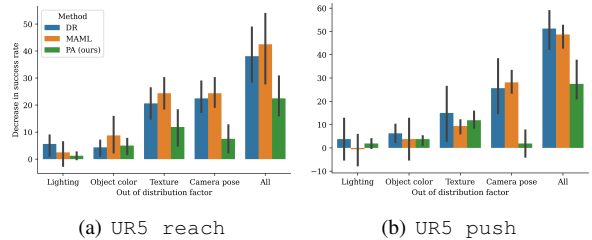


Fig. 3. Ablation study in UR5 simulation environments. Our method leads to lower decrease in success for all out of distribution factors including lighting, object color, table texture and camera pose. Error bar indicates one standard deviation.

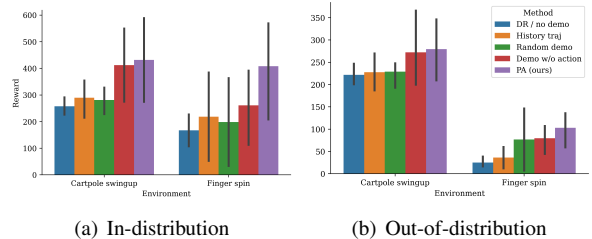


Fig. 4. Ablation study in Distracting CS over different variants in demonstration.

3. *How do different components in demonstration trajectories contribute to the observed performance of the proposed approach?*

To understand how various components in the trajectory prompt influence the performance of PromptAdapt, we conducted an ablation study in two Distracting CS environments, focusing on the following variants. **History Trajectory**: replace the demonstration with the previous h steps of student rollout; **Random Demonstration**: use a random rollout instead of expert demonstration for prompt; **Demonstration without Action**: use the expert demonstration but excludes action. All variants are trained and tested both in the given configuration.

The findings from our study, as shown in Figure 4, suggest that each of the variants results in some degree of performance drop in both in-distribution and out-of-distribution settings. This implies that a single demonstration contains valuable information about the target environment that isn’t fully captured by the history trajectory, a random demonstration, or a demonstration without action. The History Trajectory variant, which utilizes past

TABLE II
EVALUATION IN SIMULATED UR5 ENVIRONMENTS BY TASK SUCCESS RATE

Task	In-Distribution Reward				Out-of-Distribution Reward			
	RL	Policy Distillation			RL	Policy Distillation		
	DR	DR	MAML	PA (ours)	DR	DR	MAML	PA (ours)
UR5 reach	48±16	90±12	96±4	99±1	11±1	52±10	53±12	77±9
UR5 push	39±44	88±8	87±6	96±1	9±14	36±8	38±5	68±9

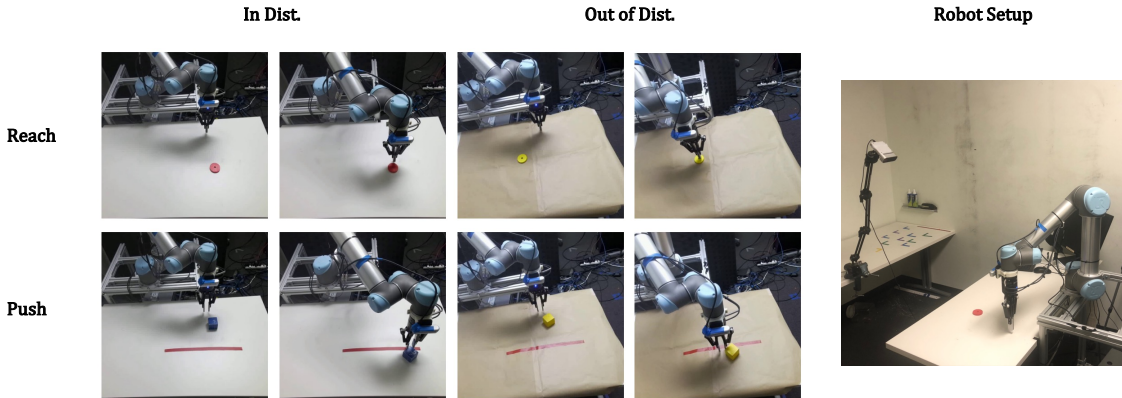


Fig. 5. **Left:** Real-world observations for UR5 reach and UR5 push. **Right:** Experiment setup.

student rollouts, shows a substantial decrease in performance, suggesting that these rollouts may not fully encapsulate the specifics of the target task. The Random Demonstration variant also shows a similar performance dip, hinting that the quality of the demonstration, rather than merely the presence of test domain trajectories, is critical in effective adaptation. Finally, when action information is excluded in the Demonstration without Action variant, we observe a slight drop in performance, indicating that action sequences in the single demonstration also help adaptation by providing information about transition dynamics and optimal policy. Overall, these results indicate the value of expert demonstrations, inclusive of action sequences, in the successful adaptation of our method.

4. How does our approach apply to real-world?

We evaluate the real-world applicability of our method under the sim-to-real setting for the two UR5 robot manipulations tasks compared to policy distillation based baseline methods. Demonstration in real-world is recorded by a human tele-operator. As demonstrated in Table III, our method consistently outperforms the baseline approaches in the UR5 reach and UR5 push. This highlights the effectiveness of our method in handling both in-distribution and out-of-distribution settings, leading to improved performance under the domain shift from simulation to real-world. Figure 5 presents visual examples of the real-world experiment.

V. LIMITATION

Through our sim-to-real experiments, we have demonstrated that demonstrations provided by human operators during testing can align well with those learned via RL during training. Meanwhile, the proposed method relies on

TABLE III
EVALUATION ON REAL-WORLD ROBOT BY SUCCESS RATES.

	In distribution success rate		
	DR	MAML	PA (ours)
UR5 reach	48±10	50±8	80±8
UR5 push	30±8	33±10	53±13

	Out of distribution success rate		
	DR	MAML	PA (ours)
UR5 reach	25±6	28±5	50±8
UR5 push	13±5	13±10	33±5

the availability of a demonstration trajectory on the same robot to bridge the domain gap between training and testing environments. This could limit the applicability when only cross-embodiment demonstrations are available (e.g., from a different type of robot). Further analysis and solution to this problem would be in the scope of future work.

VI. CONCLUSION

In this paper, we introduced a novel approach to learn domain-adapting visual policies. Our method, utilizing the strengths of policy distillation and the Transformer architecture, efficiently adapts visuomotor policies to target environments using a single demonstration. Through comprehensive experiments, we demonstrate the effectiveness of our method across various simulated and real-world robotics tasks. Ablation studies further illustrate the vital roles of different components in the trajectory prompt, underscoring the significant role of demonstrations in our approach. Our proposed method effectively improve the adaptation of policies to novel environments, paving the way for more efficient and effective domain-adapting visuomotor policies.

REFERENCES

- [1] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke *et al.*, “Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation,” *arXiv preprint arXiv:1806.10293*, 2018.
- [2] J. Kulhánek, E. Derner, T. De Bruin, and R. Babuška, “Vision-based navigation using deep reinforcement learning,” in *2019 european conference on mobile robots (ECMR)*. IEEE, 2019, pp. 1–8.
- [3] N. Hansen, H. Su, and X. Wang, “Stabilizing deep q-learning with convnets and vision transformers under data augmentation,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 3680–3693, 2021.
- [4] E. Todorov, T. Erez, and Y. Tassa, “Mujoco: A physics engine for model-based control,” in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 5026–5033.
- [5] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State, “Isaac gym: High performance gpu-based physics simulation for robot learning,” 2021.
- [6] W. Zhao, J. P. Queralta, and T. Westerlund, “Sim-to-real transfer in deep reinforcement learning for robotics: a survey,” in *2020 IEEE symposium series on computational intelligence (SSCI)*. IEEE, 2020, pp. 737–744.
- [7] K. Cobbe, C. Hesse, J. Hilton, and J. Schulman, “Leveraging procedural generation to benchmark reinforcement learning,” in *International conference on machine learning*. PMLR, 2020, pp. 2048–2056.
- [8] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, “Domain-adversarial training of neural networks,” *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.
- [9] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, “Adversarial discriminative domain adaptation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7167–7176.
- [10] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, “Domain randomization for transferring deep neural networks from simulation to the real world,” in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017, pp. 23–30.
- [11] X. Chen, J. Hu, C. Jin, L. Li, and L. Wang, “Understanding domain randomization for sim-to-real transfer,” *arXiv preprint arXiv:2110.03239*, 2021.
- [12] L. Fan, G. Wang, D.-A. Huang, Z. Yu, L. Fei-Fei, Y. Zhu, and A. Anandkumar, “Secant: Self-expert cloning for zero-shot generalization of visual policies,” *arXiv preprint arXiv:2106.09678*, 2021.
- [13] S. Ross, G. Gordon, and D. Bagnell, “A reduction of imitation learning and structured prediction to no-regret online learning,” in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.
- [14] J. Ho and S. Ermon, “Generative adversarial imitation learning,” *Advances in neural information processing systems*, vol. 29, 2016.
- [15] A. A. Rusu, S. G. Colmenarejo, C. Gulcehre, G. Desjardins, J. Kirkpatrick, R. Pascanu, V. Mnih, K. Kavukcuoglu, and R. Hadsell, “Policy distillation,” *arXiv preprint arXiv:1511.06295*, 2015.
- [16] E. Parisotto, J. L. Ba, and R. Salakhutdinov, “Actor-mimic: Deep multitask and transfer reinforcement learning,” *arXiv preprint arXiv:1511.06342*, 2015.
- [17] T. Chen, J. Xu, and P. Agrawal, “A system for general in-hand object re-orientation,” in *Conference on Robot Learning*. PMLR, 2022, pp. 297–307.
- [18] H. Lai, W. Zhang, X. He, C. Yu, Z. Tian, Y. Yu, and J. Wang, “Sim-to-real transfer for quadrupedal locomotion via terrain transformer,” *arXiv preprint arXiv:2212.07740*, 2022.
- [19] S. Dasari and A. Gupta, “Transformers for one-shot visual imitation,” in *Conference on Robot Learning*. PMLR, 2021, pp. 2071–2084.
- [20] Z. Mandi, F. Liu, K. Lee, and P. Abbeel, “Towards more generalizable one-shot visual imitation learning,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 2434–2444.
- [21] M. Xu, Y. Shen, S. Zhang, Y. Lu, D. Zhao, J. Tenenbaum, and C. Gan, “Prompting decision transformer for few-shot policy generalization,” in *International Conference on Machine Learning*. PMLR, 2022, pp. 24 631–24 645.
- [22] C. Finn, X. Y. Tan, Y. Duan, T. Darrell, S. Levine, and P. Abbeel, “Deep spatial autoencoders for visuomotor learning,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 512–519.
- [23] X. Lin, H. Bawaja, G. Kantor, and D. Held, “Adaptive auxiliary task weighting for reinforcement learning,” *Advances in neural information processing systems*, vol. 32, 2019.
- [24] C. Lyle, M. Rowland, G. Ostrovski, and W. Dabney, “On the effect of auxiliary tasks on representation dynamics,” in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 1–9.
- [25] T. He, Y. Zhang, K. Ren, M. Liu, C. Wang, W. Zhang, Y. Yang, and D. Li, “Reinforcement learning with automated auxiliary loss search,” *arXiv preprint arXiv:2210.06041*, 2022.
- [26] I. Kostrikov, D. Yarats, and R. Fergus, “Image augmentation is all you need: Regularizing deep reinforcement learning from pixels,” *arXiv preprint arXiv:2004.13649*, 2020.
- [27] N. Hansen and X. Wang, “Generalization in reinforcement learning by soft data augmentation,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 611–13 617.
- [28] G. Ma, Z. Wang, Z. Yuan, X. Wang, B. Yuan, and D. Tao, “A comprehensive survey of data augmentation in visual reinforcement learning,” *arXiv preprint arXiv:2210.04561*, 2022.
- [29] D. Bertoin, A. Zouitine, M. Zouitine, and E. Rachelson, “Look where you look! saliency-guided q-networks for visual rl tasks,” in *Thirty-sixth Conference on Neural Information Processing Systems (NeurIPS 2022)*, 2022.
- [30] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, “Learning quadrupedal locomotion over challenging terrain,” *Science Robotics*, vol. 5, no. 47, p. eabc5986, 2020.
- [31] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, “Learning robust perceptive locomotion for quadrupedal robots in the wild,” *Science Robotics*, vol. 7, no. 62, p. eabk2822, 2022.
- [32] V. Lim, H. Huang, L. Y. Chen, J. Wang, J. Ichnowski, D. Seita, M. Laskey, and K. Goldberg, “Planar robot casting with real2sim2real self-supervised learning,” *arXiv preprint arXiv:2111.04814*, 2021.
- [33] —, “Real2sim2real: Self-supervised learning of physical single-step dynamic actions for planar robot casting,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 8282–8289.
- [34] W. Yu, J. Tan, C. K. Liu, and G. Turk, “Preparing for the unknown: Learning a universal policy with online system identification,” *arXiv preprint arXiv:1702.02453*, 2017.
- [35] A. Kumar, Z. Fu, D. Pathak, and J. Malik, “Rma: Rapid motor adaptation for legged robots,” *arXiv preprint arXiv:2107.04034*, 2021.
- [36] N. Hansen, R. Jangir, Y. Sun, G. Alenyà, P. Abbeel, A. A. Efros, L. Pinto, and X. Wang, “Self-supervised policy adaptation during deployment,” *arXiv preprint arXiv:2007.04309*, 2020.
- [37] K. Rao, C. Harris, A. Irpan, S. Levine, J. Ibarz, and M. Khansari, “RI-cycleGAN: Reinforcement learning aware simulation-to-real,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 157–11 166.
- [38] D. Ho, K. Rao, Z. Xu, E. Jang, M. Khansari, and Y. Bai, “Retinagan: An object-aware approach to sim-to-real transfer,” 2020. [Online]. Available: <https://arxiv.org/abs/2011.03148>
- [39] T. Yoneda, G. Yang, M. R. Walter, and B. Stadie, “Invariance through inference,” *arXiv preprint arXiv:2112.08526*, 2021.
- [40] K. Rakelly, A. Zhou, C. Finn, S. Levine, and D. Quillen, “Efficient off-policy meta-reinforcement learning via probabilistic context variables,” in *International conference on machine learning*. PMLR, 2019, pp. 5331–5340.
- [41] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, “Meta-learning in neural networks: A survey,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 9, pp. 5149–5169, 2021.
- [42] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, “Generalizing from a few examples: A survey on few-shot learning,” *ACM computing surveys (csur)*, vol. 53, no. 3, pp. 1–34, 2020.
- [43] C. Finn, P. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” in *International conference on machine learning*. PMLR, 2017, pp. 1126–1135.
- [44] Y. Duan, M. Andrychowicz, B. Stadie, O. Jonathan Ho, J. Schneider, I. Sutskever, P. Abbeel, and W. Zaremba, “One-shot imitation learning,” *Advances in neural information processing systems*, vol. 30, 2017.
- [45] L. Kirsch, S. van Steenkiste, and J. Schmidhuber, “Improving generalization in meta reinforcement learning using learned objectives,” *arXiv preprint arXiv:1910.04098*, 2019.

- [46] X. Song, Y. Yang, K. Choromanski, K. Caluwaerts, W. Gao, C. Finn, and J. Tan, "Rapidly adaptable legged robots via evolutionary meta-learning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 3769–3776.
- [47] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
- [48] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *AI Open*, 2022.
- [49] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *arXiv preprint arXiv:2107.13586*, 2021.
- [50] J. Wei, X. Wang, D. Schuurmans, M. Bosma, E. Chi, Q. Le, and D. Zhou, "Chain of thought prompting elicits reasoning in large language models," *arXiv preprint arXiv:2201.11903*, 2022.
- [51] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, "Decision transformer: Reinforcement learning via sequence modeling," *Advances in neural information processing systems*, vol. 34, pp. 15 084–15 097, 2021.
- [52] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
- [53] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel *et al.*, "Soft actor-critic algorithms and applications," *arXiv preprint arXiv:1812.05905*, 2018.
- [54] A. Stone, O. Ramirez, K. Konolige, and R. Jonschkowski, "The distracting control suite—a challenging benchmark for reinforcement learning from pixels," *arXiv preprint arXiv:2101.02722*, 2021.
- [55] Y. Tassa, Y. Doron, A. Muldal, T. Erez, Y. Li, D. de Las Casas, D. Budden, A. Abdolmaleki, J. Merel, A. Lefrancq, T. Lillicrap, and M. Riedmiller, "Deepmind control suite," 2018.