Robotic Offline RL from Internet Videos via Value-Function Pre-Training

Chethan Bhateja[*,1], **Derek Guo**[*,1], **Dibya Ghosh**[*,1], **Anikait Singh**[1,2], **Manan Tomar**[1], **Quan Vuong**[2], **Yevgen Chebotar**[2], **Sergey Levine**[1,2], **and Aviral Kumar**[1,2]

[*]Equal contributions, [1]UC Berkeley, [2]Google DeepMind

Abstract

The full paper is available here. Pre-training on Internet data has proven to be a key ingredient for broad generalization in many modern ML systems. For robotics applications, data remains limited and video, the largest prior source of data available, offers observation-only experience without the action or reward annotations that cannot easily be incorporated in robotic learning methods. In this paper, we develop a system for leveraging large-scale human video datasets in robotic offline RL, based entirely on learning value functions via temporal-difference learning. We show that value learning on video datasets learns representations that are more conducive to downstream robotic offline RL than other approaches for learning from video data. Our system, called V-PTR, combines the benefits of pre-training on video data with robotic offline RL approaches that train on diverse robot data, resulting policies that perform better, act robustly, and generalize broadly. On several manipulation tasks on a real WidowX robot, our framework produces policies that greatly improve over prior methods. Videos can be found on our project website.

1 Introduction

The full paper is available here. Developing methods capable of acquiring robotic skills that generalize widely to new scenarios is an important problem in robotic learning. In other areas of machine learning, broad generalization has been fueled primarily by pre-training on large datasets with a diversity of behavior. It seems compelling that the same formula may be applied to robotic learning, but in practice, even our largest robotic datasets contain data for relatively limited tasks and scenarios. In principle, robotic reinforcement learning (RL) should be able to learn from more general sources of data like human video, which are far more abundant and capture a broader set of skills, situations, and interactions. However, these datasets are difficult to incorporate into RL methods that exist today, since internet-scale video data does not come with action or reward annotations present in typical robot data.

Motivated by the above desiderata for video pre-training, we aim to develop an approach that pre-trains on Internet-scale human video to produce representations for downstream offline RL. Our main contribution is a system, which we call **V**ideo **P**re-**T**raining for **R**obots (V-PTR), that fits value functions to model long-term outcomes achieved when solving tasks on action-free video data.

Concretely, V-PTR pre-trains on human videos by learning an intent-conditioned value function [5]. This approach eschews self-supervised representation learning objectives utilized in prior works [14; 11; 15] in favor of a TD value learning objective which reflects downstream finetuning. Next, we fine-tune on a multi-task robotic dataset, annotated with actions, tasks, and rewards, using value-based offline RL [10] followed by a target dataset. Each phase gradually incorporates the knowledge of "what future outcomes can be achieved" (video pre-training), "what robot actions lead to these outcomes" (robot pre-training), and "how can the desired task be solved" (fine-tuning).

NeurIPS 2023 Workshop on Robot Learning: Pretraining, Fine-Tuning, and Generalization with Large Scale Models, New Orleans, United States of America

Our experiments on several manipulation tasks on a real WidowX robot show that by pre-training on human video data (Ego4D [6]) and multi-task robot data (Bridge data [4]), V-PTR endows downstream RL methods with improved generalization to different distractors, positions, and other variations in the workspace compared to prior methods that learn from videos, significantly outperforming methods like VIP [11] and R3M [14]. To our knowledge, our work presents the first large-scale demonstration showing that TD-learning alone induces effective video pre-training for robotic RL.

2 Related Work

A number of prior approaches learn representations from video by applying image-level representation objectives on individual frames like reconstruction [12; 17; 21; 8] or contrastive learning on images [20]. While these objectives are widely used in computer vision, resulting representations do not capture any information about environment dynamics. Other approaches model long-term dynamics from video by predicting the next frame [18], learning value functions [11; 5], running time-contrastive learning [19; 13], or learning language-video alignment [8].

The most closely related work is value-implicit pre-training (VIP) [11], which pre-trains a value function using time-contrastive prediction for downstream reward shaping. Both learn value functions during pre-training, albeit with different algorithms (contrastive BRM learning vs. TD learning), and different policies (dataset policy vs. intent-conditioned policy) [2; 3]. Furthermore, the system desiderata differ for VIP and V-PTR: VIP focuses on learning visual reward functions for weighted behavioral cloning, while we seek value initializations for downstream offline RL.

3 Problem Statement and Background

We aim to leverage Internet-scale video data and multi-task robotic data to boost the robustness and generalization of robotic offline RL. We have access to an Internet-scale video dataset $\mathcal{D}_{\text{video}}$ (e.g., the Ego4D dataset [6]), a target dataset, $\mathcal{D}_{\text{target}}$ of a limited number of demonstrations for a given target task on the robot, and optionally a dataset of multi-task robot behaviors, $\mathcal{D}_{\text{robot}}$ (e.g. Bridge data, RTX data). We formalize the learning problem as maximization of expected reward in an MDP; please see the appendix for more details.

Our system utilizes a generalized formulation of goal-conditioned RL and temporal-difference learning. In a nutshell, the goal-conditioned RL problem trains the agent to achieve arbitrary goal frames $\mathbf{g}$, where rewards are specified by the sparse signal of $\mathbb{I}\left(\mathbf{s}=\mathbf{g}\right)$ when the frame is identical to the goal frame. Although the reward signal is sparse, goals and rewards can be defined by hindsight relabelling [1]. To learn a policy for the downstream task, we use value-based offline RL methods, which optimize π against a learned Q-function $Q^\pi(\mathbf{s},\mathbf{a})$. The Q-value function measures the expected long-term reward attained when executing action $\mathbf{a}$ at state $\mathbf{s}$, then following policy π thereafter, and satisfies the Bellman equation $Q^\pi(\mathbf{s},\mathbf{a}) = r(\mathbf{s},\mathbf{a}) + \gamma \mathbb{E}_{\mathbf{s}',\mathbf{a}'}[Q^\pi(\mathbf{s}',\mathbf{a}')]$.

4 Video Pre-Training for Robotic Offline RL

We develop V-PTR, our system that pre-trains general value functions on Internet-scale video data to extract visual representations useful for downstream RL.

System overview. Our system, V-PTR, pre-trains in three phases. In the first phase, we train an intention-conditioned value function [5] on video data using a value-learning objective to model the outcomes associated with solving the parametric family of goal-achieving tasks. Next, we refine this representation with multi-task robot data with actions and rewards, by training a state-action Q-function on this representation using offline RL. In our final phase, to adapt the system to a new target task, our system fine-tunes the Q-function and the policy on the target dataset.

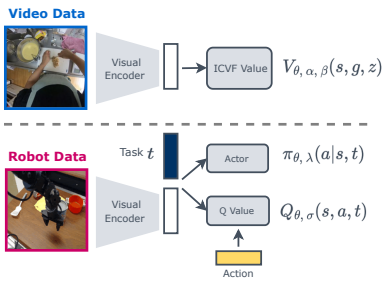

Figure 1: Network architecture.

4.1 Phase 1: Video Pre-Training via TD-Learning

Since the goal of video pre-training is to improve the performance of downstream value-based offline RL, we turn to learning value functions on the video data as a natural pre-training procedure. We choose to pre-train by learning an intent-conditioned value function (ICVF), a recently-proposed general value function that can be efficiently trained on passive data without action labels [5] via

TD-learning. An ICVF, annotated $V(\mathbf{s}_{\text{video}}, \mathbf{g}_{\text{video}}, \mathbf{z})$ computes the value obtained towards reaching a goal $\mathbf{g}_{\text{video}}$, assuming the policy *intended* to reach a different intended goal $\mathbf{z}$, formally defined as

$$V(\mathbf{s}_{\text{video}}, \mathbf{g}_{\text{video}}, \mathbf{z}) = \mathbb{E}_{a_t \sim \pi^*_{\mathbf{z}}(\cdot|\mathbf{s}_t)}\Big[\sum_t \gamma^t \mathbb{I}\left(\mathbf{s}_{\text{video}} = \mathbf{g}_{\text{video}}\right)\Big].$$

We follow [5] and parameterize our estimated value function as

$$V(\mathbf{s}_{\text{video}}, \mathbf{g}_{\text{video}}, \mathbf{z}) := \phi(\mathbf{s}_{\text{video}})^\top T(\mathbf{z}) \psi(\mathbf{g}_{\text{video}}),$$

where ϕ_θ and ψ_α denote models that transform the observation and the goal observation respectively into low-dimensional representations, and T_β, a learned mapping aligning the two representations. At convergence, the ICVF provides a measure of temporal spatiality, and the learned representation $\phi_\theta(\mathbf{s})$ offers useful features for downstream value functions.

4.2 Phase 2: Multi-Task Robot Pre-Training via Offline RL

In the next phase, we refine the learned representation on a multi-task robot dataset, $\mathcal{D}_{\text{robot}}$, to bridge the domain gap between robot image observations and human video, and to provide information about the target robot embodiment. The tasks and workspaces in this robot dataset are explicitly disjoint from the target tasks used in the downstream evaluation.

V-PTR initializes a task-conditioned Q-function and policy with the representation encoder $\phi_{\theta^*_{\text{video}}}$ obtained at the end of phase 1 and trains them using multi-task conservative Q-learning (CQL) [9].

4.3 Phase 3: Fine-Tuning to a Target Task

Finally, we fine-tune the value function and policy to the target task on the target dataset $\mathcal{D}_{\text{target}}$. We follow [10] and treat the target data simply as a new task; fine-tuning involves assigning a new task identifier to the target data, and continuing multi-task offline CQL on the robot pre-training and target tasks jointly. To emphasize the target task during fine-tuning, we perform stratified sampling where 90% proportion of the training batch comes from $\mathcal{D}_{\text{robot}}$ and 10% from $\mathcal{D}_{\text{target}}$.

5 Experimental Results

We validate the effectiveness of V-PTR in boosting the generalization of robotic offline RL by evaluating it in new scenes, compare to other approaches for incorporating video data, and performing additional diagnostic experiments to understand how value pre-training can provide useful representations for downstream robotic RL. Videos of our evaluations can be found on our **project website**.

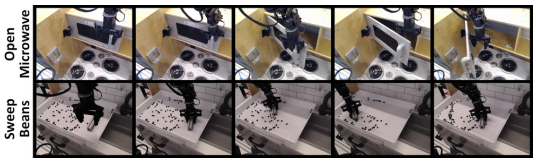

Figure 2: Examples of setup for complex tasks. We utilize the robot setup from the Bridge dataset [4]. **Top:** Two-phase open microwave; **Bottom:** Sweep beans into pile with tool.

Comparisons to prior methods. We compare V-PTR to approaches that do not utilize video data (PTR [10], BC [4]), as well as other methods for video pre-training (R3M [14], MVP [22; 15], and VIP [11]). Following the protocols in these prior works, we fine-tune all these representations with imitation learning on multi-task and target robot data (phases 2 and 3). Additionally, we train VIP with CQL for an apples-to-apples comparison to V-PTR.

5.1 Real-World Results

Table 1: Task success rates of V-PTR and prior methods on several manipulation tasks over 12 trials. Note that V-PTR outperforms all prior methods, including those approaches that freeze the learned representation, use imitation learning for downstream control, or use only robot data.

		Video pre-training					No videos	No robot data
	Task	**V-PTR (Ours)**	**R3M+BC**	**MVP+BC**	**VIP+CQL**	**VIP$_{\text{frozen}}$+CQL**	**PTR**	**V-PTR w/o phase 2**
No Distractors	Croissant from bowl	7 / 12	0 / 12	4 / 12	2 / 12	0 / 12	3 / 12	5 / 12
	Sweet potato on plate	6 / 12	0 / 12	1 / 12	0 / 12	0 / 12	1 / 12	1 / 12
	Knife in pot	6 / 12	0 / 12	0 / 12	0 / 12	0 / 12	0 / 12	0 / 12
	Cucumber in pot	5 / 12	0 / 12	1 / 12	0 / 12	0 / 12	1 / 12	1 / 12
	Total	24 / 48	0 / 48	6 / 48	2 / 48	0 / 48	5 / 48	7 / 48
With Distractors	Croissant from bowl	8 / 12	0 / 12	3 / 12	2 / 12	0 / 12	0 / 12	3 / 12
	Sweet potato on plate	4 / 12	0 / 12	2 / 12	0 / 12	0 / 12	1 / 12	2 / 12
	Knife in pot	4 / 12	0 / 12	0 / 12	1 / 12	0 / 12	0 / 12	0 / 12
	Cucumber in pot	4 / 12	0 / 12	0 / 12	1 / 12	0 / 12	0 / 12	1 / 12
	Total	20 / 48	0 / 48	5 / 48	4 / 48	0 / 48	1 / 48	6 / 48

Our main results are summarized in Table 1. First, we evaluate the performance of the policy trained by V-PTR as we vary the robot's initial pose and the position of objects in scene. Then, we test performance after adding novel distractors to the scene. We also compare V-PTR in more detail to VIP (which also uses similar video data) on more complex tasks ("open microwave" and "sweep beans") in Table 2. Specifically, we compare to different variants of VIP as discussed above (VIP+CQL; $\text{VIP}_{\text{reward}}$) and find that V-PTR outperforms these variants.

Table 2: Performance of V-PTR, VIP, and PTR on more complex tasks. V-PTR outperforms PTR as well as VIP with both CQL or BC with reward shaping from [11].

			No CQL	
Task	**V-PTR**	**VIP [11]+CQL**	**PTR [10]**	**$\text{VIP}_{\text{reward}}$ [11]**
Open Microwave	5 / 12	2 / 12	0 / 12	0 / 12
Sweep Beans	6 / 12	5 / 12	2 / 12	2 / 12

5.2 Visualizations and Diagnostic Experiments

Video pre-training via V-PTR improves target value estimation. We visualize the learned value function on frames from different rollouts in Figure 3 (left), where the true value function should monotonically increase throughout a successful rollout. In the presence of distractor objects, V-PTR obtains smoother and more monotonic value functions compared to PTR or VIP.

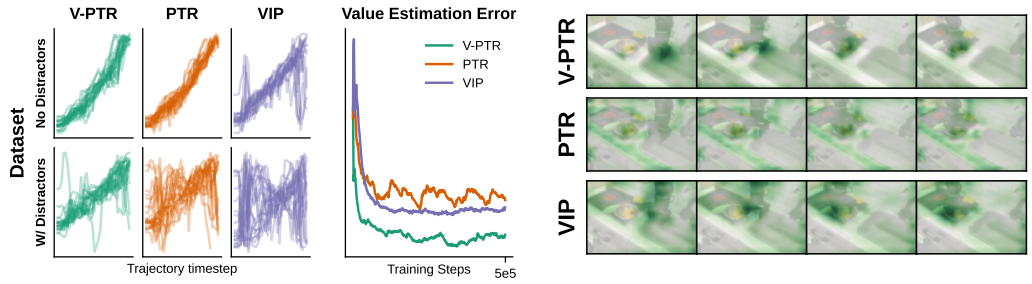

Figure 3: Left: Visualizing the learned values $V(\mathbf{s}_t)$. Note that V-PTR values tend to be much smoother than those from PTR and VIP, especially on held-out rollouts with novel distractors. **Right: Grad-CAM visuals**. V-PTR focuses on more pertinent objects (the gripper, and the target object) than the compared methods

To more precisely measure the ability of V-PTR to fit downstream value functions, we train a SARSA value function (for which we may compute a closed-form optimal solution) on top of frozen pre-trained representations from PTR (no video data), VIP [11], and V-PTR, and report the error between the predicted value and the ground-truth value on a held-out dataset in Figure 3 (middle). V-PTR attains the smallest fitting error compared to both PTR and VIP.

What kind of visual cues do representations trained by V-PTR capture? We probe which parts of the image are attended to by the learned policy for V-PTR and other baselines, by utilizing Grad-CAM [16] to mark patches of the frame that influence the output of the learned policy in green. We observe that V-PTR policies discard the scene background and focus on cues like the object and gripper positions while PTR and VIP place higher focuses on the scene background.

6 Discussion and Conclusion

We designed V-PTR, which uses value function pre-training on the large-scale Ego4D video dataset [6] and the robot Bridge dataset [4] for downstream robotic learning. While V-PTR outperforms prior methods for learning from video, we found that all the current methods remain sensitive to deviations in workspace height, camera angle, and robot configurations. There also exist many opportunities to scale, like multi-robot datasets, larger human video datasets with language, or larger models. We are quite excited about the promise of using RL-like value pre-training on video data for improving the general ability of robot learning algorithms.

References

[1] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017. 2, 7

[2] Leemon Baird. Residual Algorithms : Reinforcement Learning with Function Approximation. In *International Conference on Machine Learning (ICML)*, 1995. 2

[3] Christoph Dann, Gerhard Neumann, Jan Peters, et al. Policy evaluation with temporal differences: A survey and comparison. *Journal of Machine Learning Research*, 15:809–883, 2014. 2

[4] Frederik Ebert, Yanlai Yang, Karl Schmeckpeper, Bernadette Bucher, Georgios Georgakis, Kostas Daniilidis, Chelsea Finn, and Sergey Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. *arXiv preprint arXiv:2109.13396*, 2021. 2, 3, 4, 8

[5] Dibya Ghosh, Chethan Bhateja, and Sergey Levine. Reinforcement learning from passive data via latent intentions. *arXiv preprint arXiv:2304.04782*, 2023. 1, 2, 3, 7, 8

[6] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022. 2, 4, 7, 8

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 9

[8] Siddharth Karamcheti, Suraj Nair, Annie S Chen, Thomas Kollar, Chelsea Finn, Dorsa Sadigh, and Percy Liang. Language-driven representation learning for robotics. *arXiv preprint arXiv:2302.12766*, 2023. 2

[9] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *arXiv preprint arXiv:2006.04779*, 2020. 3, 7, 9

[10] Aviral Kumar, Anikait Singh, Frederik Ebert, Yanlai Yang, Chelsea Finn, and Sergey Levine. Pre-training for robots: Offline rl enables learning new tasks from a handful of trials. *RSS 2023; arXiv:2210.05178*, 2023. 1, 3, 4, 7, 8, 9, 10

[11] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022. 1, 2, 3, 4, 8, 9, 10

[12] A. Nair, S. Bahl, A. Khazatsky, V. Pong, G. Berseth, and S. Levine. Contextual imagined goals for self-supervised robotic learning. In *Conference on Robot Learning (CoRL)*, 2019. 2

[13] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhi Gupta. R3m: A universal visual representation for robot manipulation. *ArXiv*, abs/2203.12601, 2022. 2, 8, 9

[14] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022. 1, 2, 3

[15] Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training. *arXiv preprint arXiv:2210.03109*, 2022. 1, 3, 9

[16] RR Selvaraju, M Cogswell, A Das, R Vedantam, D Parikh, and D Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. arxiv 2016. *arXiv preprint arXiv:1610.02391*. 4

[17] Younggyo Seo, Danijar Hafner, Hao Liu, Fangchen Liu, Stephen James, Kimin Lee, and P. Abbeel. Masked world models for visual control. *ArXiv*, abs/2206.14244, 2022. 2

[18] Younggyo Seo, Kimin Lee, Stephen James, and P. Abbeel. Reinforcement learning with action-free pre-training from videos. *ArXiv*, abs/2203.13880, 2022. 2

[19] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, and Sergey Levine. Time-contrastive networks: Self-supervised learning from video. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1134–1141, 2017. 2

[20] A. Srinivas, Michael Laskin, and P. Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*, 2020. 2

[21] Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for motor control. *ArXiv*, abs/2203.06173, 2022. 2, 9

[22] Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022. 3

Appendices

A Problem Statement and Background

We aim to leverage Internet-scale video data and multi-task robotic data to boost the robustness and generalization of robotic offline RL. We formulate the robot skill learning problem as the problem of maximizing infinite-horizon discounted reward in a Markov decision process (MDP).

Formal problem statement. We assume access to two pre-training datasets: an Internet-scale video dataset $\mathcal{D}_{\text{video}}$ (e.g., the Ego4D dataset [6]) and a target dataset, $\mathcal{D}_{\text{target}}$ of a limited number of demonstrations for a given target task on the robot. Additionally we are also provided a dataset of multi-task robot behaviors, $\mathcal{D}_{\text{robot}}$, which may not contain any data relevant to the target task. The video dataset $\mathcal{D}_{\text{video}}$ consists of sequences of frames (i.e., observations in the MDP), with no action or rewards. Denoting a frame as $\mathbf{s}_{i,j}$, we define $\mathcal{D}_{\text{video}} := \left\{ \left(\mathbf{s}_{i,0}, \mathbf{s}_{i,1}, \cdots\right) \right\}_{i=1}^{n_{\text{video}}}$. The target dataset, $\mathcal{D}_{\text{target}}$, comprises of a few demonstrations of the target task on the robot $\mathcal{D}_{\text{target}} := \left\{ \left(\mathbf{s}_{i,0}, \mathbf{a}_{i,0}, r_{i,0}, \mathbf{s}_{i,1}, \cdots\right) \right\}_{i=1}^{n_{\text{target}}}$, where the reward, $r_{i,j}$ is annotated to be +1 only on the final three timesteps of the demonstration (following [10]). The multi-task robot dataset $\mathcal{D}_{\text{robot}}$ is organized identically to the target robot dataset, but with an additional task annotation on each trajectory t_i, which is specified either as a one-hot identifier or by natural language. Our goal is train policy π which maximizes the γ-discounted cumulative reward, $\mathbb{E}_{\mathbf{s}_0 \sim \rho_0, \mathbf{a}_{0:\infty}, \mathbf{s}_{1:\infty} \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t r(\mathbf{s}_t, \mathbf{a}_t)\right]$, starting from a more diverse set of initial states indicated by the distribution ρ_0 than what was observed in the target dataset (e.g., more variation in distractor objects).

Background. Our system utilizes a generalized formulation of goal-conditioned RL and temporal-difference learning for pre-training value functions. In a nutshell, the goal-conditioned RL problem trains the agent to achieve arbitrary goal frames $\mathbf{g}$, where rewards are specified by the sparse signal of $\mathbb{I}\left(\mathbf{s} = \mathbf{g}\right)$ when the frame is identical to the goal frame. Although the reward signal is sparse, goals and rewards can be defined by hindsight relabelling [1]. To learn a policy for the downstream task, we use value-based offline RL methods, which optimize π against a learned Q-function $Q^\pi(\mathbf{s}, \mathbf{a})$. The Q-value function measures the expected long-term reward attained when executing action $\mathbf{a}$ at state $\mathbf{s}$, then following policy π thereafter, and satisfies the Bellman equation $Q^\pi(\mathbf{s}, \mathbf{a}) = r(\mathbf{s}, \mathbf{a}) + \gamma \mathbb{E}_{\mathbf{s}', \mathbf{a}'}[Q^\pi(\mathbf{s}', \mathbf{a}')]$.

B Full System Description: V-PTR

In this section, we will provide further details regarding the 3 phases of V-PTRİn particular, we will describe the video pre-training with ICVF [5] as well as Multi-robot pretraining with PTR [10], describing both the networks present during training as well as the objective function.

B.1 Video Pre-Training with ICVF

Network. We define the visual backbone $f_\theta(\mathbf{s})$ to be a ResNet50-v2 model, which outputs a 2048-dimensional vector. The ICVF heads ϕ, ψ, T are 2-layer MLPs with hidden dimensions of 256 and a final dimension of 256. The ICVF is then defined as $V(\mathbf{s}, \mathbf{g}, \mathbf{z}) = \phi(f(\mathbf{s}))^\top T(f(\mathbf{z}))\psi(f(\mathbf{g}))$ for video frames $\mathbf{s}, \mathbf{g}, \mathbf{z}$.

B.2 Multi-robot pretraining with PTR

Networks. We mirror the experimental setup of [10], with no modifications except for a different visual backbone. PTR uses CQL [9] as the base offline RL method, meaning it trains two Q functions, a separate policy, and delayed target copies of the Q-functions. Each network is represented by a 3-layer MLP head with width of 256, after the visual representation. To make comparisons between PTR and V-PTR exact, we also use separate encoders for the actor and critic, although both are initialized at the beginning of Phase 2 using the same visual representation as learned during Phase 1. We refer to [10] for a more complete description.

C Environment Setup and Dataset Details

In this section, we will describe the setup for our experimental results with respect to both the real-world experiments and the sim diagnostic experiments. There will be a description of the task setup as well as the corresponding datasets associated with the tasks.

C.1 Description of State and Action Space

Our state and action description follows that of PTR [10]. The state is a 128×128 RGB image captured from an over-the-shoulder camera, a one-hot vector identifying which task is to be performed, the position and rotation of the end-effector, and how much the gripper is closed. The action is the translational and angular velocity of the robot end-effector, as well as how much the gripper is closed. Rotations and angular velocities are expressed using Euler angles.

C.2 Description of Real Robotic Setup

We mirror our real-world experimental setup from Bridge [4] and PTR [10]. In particular, we designed and evaluated the performance of our method under several distinct conditions in 2 different toy kitchen domains. The robot that is used is a 6-DoF WidowX 250 robot with a fixed side camera. The scene in which the robot was placed in was a toy kitchen with elements such as stove tops, sinks, microwaves, and food objects found in the scene. A picture of our setup is found in Figure 4.

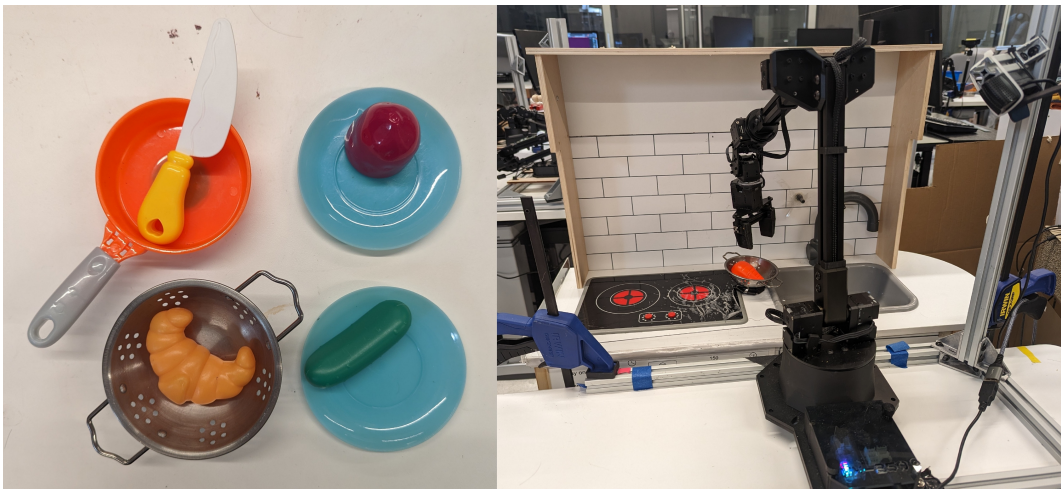

Figure 4: Real-robot experiments. We utilize the setup from the Bridge dataset [4]. The bridge dataset is collected on a 6-DoF WidowX 250 robot, with a fixed side camera placed in diverse toy-kitchens. Observations for the tasks consist of one 128×128 RGB image from a side camera, as well as robot proprioception. **Left:** task objects and containers for the tasks that we study. **Right:** Evaluation setup pick place environment.

We evaluate several pick place tasks, in which the agent has to place an object into a container amongst distractors found in the scene. Distractor objects in this scene can include other objects and containers that may have even been shown to be corresponding to different tasks.

C.3 Description of Video Pre-Training Dataset

The Video Pre-training Dataset that we utilize in Stage 1 of our method, PTR, is Ego4D [6]. We used the same pre-processed Ego4D dataset as in R3M [13] and VIP [11]. In particular, long videos that are raw in nature are clipped to be shorter consisting of 10-150 frames. From here, the clipped video is decomposed into individual frames that are pre-processed with a random crop at the video level. The frame is then resized and center-cropped to be of size $224 \times 224 \times 3$. These processed video frames are then fed into the replay buffer to be individual transitions for the ICVF objective [5].

C.4 Description of Real-World Multi-Task Robot Datasets

For the pre-training and target datasets in Stage 2 of V-PTR for the real-world multi-task training, we utilize subsets of the Bridge Dataset [4]. Mirroring the setup of Scenario 3 in PTR [10], the pre-training data comprises of all pick-and-place data found in the bridge dataset except for any demonstration data collected in the toy kitchen that was evaluated on. For the target dataset, we collect 44 new demonstrations for each of the 4 tasks: Removing Croissant from Colander, Placing Knife in Pot, Placing Sweet Potato on Plate, and Placing Cucumber in Bowl. A total of 218 successful demonstrations were collected with a human demonstrator using an Oculus Quest Headset for Phase 3 of V-PTR.

Table 3: The hyperparameters used by V-PTR. After pre-training on multi-task robot data, in the second pre-training phase V-PTR fine-tunes the representation using multi-task CQL [9] on diverse robot data. Finally the third fine-tuning phase aims to customize this policy for the target task. The above table presents hyperparameters for V-PTR that we utilize in our experiments.

Hyperparameters	
α	0.1, 1,5,10
policy architecture	ResNet-50, ViT-B
critic architecture	ResNet-50, ViT-B
policy learning rate	1e-4
critic learning rate	3e-4
reward scale	11
reward bias	-1
batch size	64

D Evaluation Details

In this section, we will provide how real and simulated environments were evaluated fairly across our method and baselines. This protocol is similar in nature to the one presented in PTR [10].

D.1 Evaluation Protocol for Real-World Experiments

To evaluate a policy in the real world, we loop through 4 starting gripper transformations that move the gripper to the left front, right front, left rear, and right rear of the kitchen environment. For each of these starting transformations, we evaluate a total of 3 times: once with the object to pick up directly underneath the gripper and twice with it shifted slightly so that the policy cannot simply move straight down from the starting location. For each of these evaluations, we let the policy run for 60 time steps.

For our experiments testing generalization to novel distractor objects, we place two novel distractor objects into the scene. We shift the locations of these objects between each evaluation and switch out the objects themselves when we change start transforms so that each combination of distractors is evaluated 3 times.

When taking objects out of containers, we do not count knocking over the container so that the object falls out as a success - the gripper must lift up the object in the air and set it outside the container.

E Experimental Details for V-PTR and Baseline Methods

E.1 CQL Finetuning [10]

For our second pre-training phase on multi-task robot data and fine-tuning on target data, we utilized CQL [9] as the downstream offline RL algorithm. Following the design decisions in the official implementation of PTR [10], we utilized a variant of Bellman backup that computes the target value by performing a maximization over target values computed for $n = 4$ actions sampled from the policy at the next state (max_q_backup). In each domain, we swept over the alpha values of $\alpha = 0.1, 1, 5, 10$. We built our code upon the CQL implementation from https://github.com/Asap7772/PTR [10].

E.2 MVP [21; 15]

For masked autoencoding (MAE)-based methods such as MVP [15], we loaded in the pre-trained PyTorch Checkpoints for the ViT-Base model. For this method, we kept the pre-trained representation frozen and finetuned a Multi-Layer Perceptron (MLP) that takes in the class token of the ViT as input to the model and outputs an action dimensional vector. The output of the MAE is normalized with layer normalization. Other hyperparameters follow the CQL fine-tuning section.

E.3 VIP [11] and R3M [13]

Prior methods have shown that TD-based updates with networks with batch normalization have instabilities during training. Given this, methods such as VIP and R3M that utilize a standard ResNet-50 [7] backbone with batch normalization do not finetune stably for methods such as CQL. For this reason, any comparisons using standard ResNet 50 either use frozen batch statistics, or are trained with a behavioral cloning objective.

All hyperparameters described above are summarized in Table 3.

F Reducing the Number of Fine-Tuning Demonstrations

We examine how reducing the number of demonstrations from 44 to 10 degrades performance in Table 4. V-PTR continues to significantly outperform baselines.

Table 4: Task success rates of V-PTR and prior methods with only 10 demonstrations. As should be expected, the performances of all approaches, including ours, degrade with less data, but V-PTR performs significantly better than other pre-training methods.

		Video pre-training	**No videos**
Task	**V-PTR (Ours)**	**VIP [11]+CQL**	**PTR [10]**
Croissant out of Colander	4 / 12	1 / 12	0 / 12
Sweet Potato on Plate	5 / 12	0 / 12	0 / 12
Knife in Pan	2 / 12	0 / 12	0 / 12
Cucumber in Pot	4 / 12	0 / 12	0 / 12
Open Microwave	8 / 12	2 / 12	4 / 12
Sweep Beans	2 / 12	0 / 12	5 / 12