

Learning Video-Conditioned Policies for Unseen Manipulation Tasks

Elliot Chane-Sane¹, Cordelia Schmid¹, Ivan Laptev¹

Abstract—The ability to specify robot commands by a non-expert user is critical for building generalist agents capable of solving a large variety of tasks. One convenient way to specify the intended robot goal is by a video of a person demonstrating the target task. While prior work typically aims to imitate human demonstrations performed in robot environments, here we focus on a more realistic and challenging setup with demonstrations recorded in natural and diverse human environments. We propose *Video-conditioned Policy learning (ViP)*, a data-driven approach that maps human demonstrations of previously unseen tasks to robot manipulation skills. To this end, we learn our policy to generate appropriate actions given current scene observations and a video of the target task. To encourage generalization to new tasks, we avoid particular tasks during training and learn our policy from unlabelled robot trajectories and corresponding robot videos. Both robot and human videos in our framework are represented by video embeddings pre-trained for human action recognition. At test time we first translate human videos to robot videos in the common video embedding space, and then use resulting embeddings to condition our policies. Notably, our approach enables robot control by human demonstrations in a *zero-shot manner*, i.e., without using robot trajectories paired with human instructions during training. We validate our approach on a set of challenging multi-task robot manipulation environments and outperform state of the art. Our method also demonstrates excellent performance in a new challenging zero-shot setup where no paired data is used during training.

I. INTRODUCTION

Significant progress has been made in recent years towards learning a generalist robot agent capable of accomplishing a wide array of skills across many environments [1], [2], [3], [4]. Central to this challenge is the ability to effectively specify tasks and rewards to the robot system in a user-friendly manner. In reinforcement learning, a task is commonly defined through a reward function [5]. However, designing good reward functions for each task is often challenging and restricting policy learning to a fixed set of tasks hinders generalization to new tasks. Goal-conditioned imitation and reinforcement learning [6], [7], [8], [9], [10], [11] can learn agents capable of performing a wide diversity of tasks with less supervision. But providing the right goal to define the task requires expert operators to come up with a suitable robot observation of the desired configuration. Other works have shown that we can learn to command generalist robots through language instructions [2], [12], [13], [14] and human videos [2], [15], which are easy to provide for non-expert operators and can generalize to unseen inputs, including behaviors beyond goal-reaching skills. Furthermore,

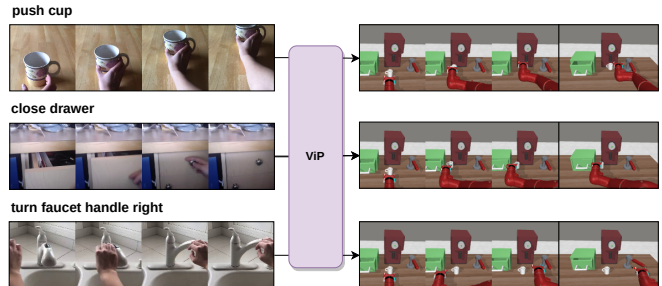


Fig. 1. Given a human video instruction in a non-robotic scene, our video-conditioned policy ViP controls the robot to perform a similar task zero-shot, i.e. the agent never observes robot data paired with human instructions during training and figures out which manipulation skill it is expected to perform in its environment at test time. We illustrate three examples of different tasks demonstrated by people and the corresponding roll-outs generated by our method for the TableTop robotics environment.

being able to specify robot skills through language or video commands unlocks solving more complex long-horizon tasks by chaining human instructions [1], [16]. Nonetheless, these methods rely on annotated demonstrations for a large set of robot skills, which is often tedious to provide, especially since task annotation must be repeated for each new robot environment.

In this work, we propose *Video-conditioned Policy learning (ViP)*, a method that learns to perform manipulation skills given a human video of the desired task in vision-based multi-task robotic environments (Figure 1). We demonstrate that, due to the similarity between robot manipulation and videos of humans performing manipulation skills, we can leverage existing large datasets of annotated human videos, such as the Something-Something-v2 dataset [17] (SSv2), to learn to map human videos to robot behaviors *in a zero-shot manner*, without training on paired data between human instructions and robot demonstrations. For instance, the robot may interact in an environment that includes a drawer, yet has received no supervision on what it is expected to do when commanded with a video instruction of a human closing a possibly different drawer. We do so by learning a video encoder using Supervised Contrastive Learning [18], where embeddings of videos of the same task are closer together in cosine distance, and show that video models trained on such large datasets of annotated human videos, which can be easily collected from the internet [17], [19], [20], [21], generalize to the robot video domain.

In addition, we show that video encoders trained for human video action recognition readily provide relevant task embeddings for multi-task policy learning. Following recent trends in data-driven robotics, our approach learns from large collections of offline robot experience that can be collected

¹Inria, École normale supérieure, CNRS, PSL Research University, 75005 Paris, France. Corresponding author: elliot.chane-sane@inria.fr
Project website: <https://www.di.ens.fr/willow/research/vip/>

in different ways, e.g. by expert demonstrations given by motion planners, teleoperated play data, random data generation processes. Given an offline dataset of robot demonstrations, we learn a video conditioned policy to regress the action conditioned on both the robot state and a video embedding of the full trajectory the state-action pair belongs to. We also keep each embedding in a library of robot embeddings. At inference, we perform nearest neighbors regression on this library using the cosine distance to the embedding of the human instruction to generate an appropriate robot embedding that is both (a) relevant to the instruction and (b) is executable by the policy. We can then execute the human video instruction by decoding the selected embedding into a robot trajectory using the learned policy.

Overall, our approach demonstrates that large collections of human videos can enable less supervision in data-driven robotics. Our contributions can be summarized as follows:

- we designed a method to train a policy conditioned on robot video embeddings given by a video encoder pretrained on a large dataset of annotated human videos;
- our method can map human video instructions to robot manipulation skills without supervision from paired data to bridge the gap between the human and robot domain;
- our approach outperforms prior works on a set of multi-task robotic environments.

II. RELATED WORK

Different methods have been explored in recent years towards robot learning with human videos. Many prior works hence consider the problem of learning to follow human demonstrations using different techniques such as pose or keypoints estimation [22], [23], [24], image translation and inpainting [25], [26], [27], [28], learning object centric representations [29], [30], simulators [31], [32] and meta learning [33]. Contrary to these prior works that often consider closely aligned human videos and robot environments e.g. humans demonstrating the task in the same lab environment as the robot, we assume that human video instructions are collected "in-the-wild" and therefore there exists a large domain gap between human videos and robot workspace.

Recent works attempt to perform model pretraining from large-scale human videos in order to get good image representations for robotic control [34], [35]. Other works have shown that we can infer states and actions from diverse videos and use it for reinforcement learning [36], [37], [38], [39]. In our work, we show that encoders trained on large datasets of videos for human action recognition provide good task embeddings for robotic manipulation.

Prior works have also considered leveraging large datasets of human videos to learn reward functions for robotics manipulations [40], [15]. The most relevant work to ours is Domain-agnostic Video Discriminator (DVD) [15] which also tackles the problem of commanding robots using in-the-wild human videos. DVD learns a video similarity by training a discriminator network to classify whether two videos are performing the same task on both annotated robot demonstrations and a subset of SSv2 videos. The similarity

score is then used as a reward for planning with an action-conditioned video generation method [41] trained on randomly collected robot experience. In contrast, our approach does not require any annotated robot demonstration and can accommodate both randomly collected robot experience and expert demonstrations. Moreover, [15] plans several sub-trajectories as high-dimensional synthetic videos per episode whereas our approach plans a full trajectory in the embedding space of robot videos.

Contrastive learning on large-scale datasets has led to significant progresses in a range of computer vision tasks [42], [43]. Contrastive learning has also been used to learn language-conditioned policies by training on large scale datasets of images [44], [45] and videos [46]. In this work, we leverage Supervised Contrastive Learning [18] for human action recognition on the SSv2 dataset to learn a mapping between human video instructions and robotic manipulation.

Our work falls under outcome-conditioned action regression [7], [47], [48]. Such works cast reinforcement learning as learning policies conditioned on trajectory information such as future returns [49], [50], [51], many previous timesteps [52], [53] or a desired goal-configuration [54], [6], [7], [47] to learn multi-task policies. In contrast, our policy is conditioned on a video embedding of the full robot trajectory. [2] also showed that we can learn a video-conditioned controllers by conditioning the policy on both a video of the full robot trajectory and paired sets of human videos appropriately collected for each task. Other works [12], [13] show that pairing a small subset of robot data with language instructions enables generalizable language-conditioned task execution. Our method does not require any robot data paired with human instructions and, hence, can learn generic policies from unlabelled robot datasets.

III. METHOD

In Section III-A we first present an overview of our approach that enables robots to mimic new tasks demonstrated by people in natural human environments. We then detail how we learn a generic video-conditioned policy from randomly-generated demonstrations in Section III-B. We further describe how we condition our policy on human videos in Section III-C. Finally, we describe how we learn a similarity function for matching human videos to robot roll-outs without using paired human-robot data in Section III-D.

A. Method overview

We aim to perform a robot manipulation task conditioned on a human video in a vision-based multi-task environment. At test time our system receives an input video of a previously unseen task performed by a person, such as pushing a mug or closing a drawer, and controls a robot arm with the intent of performing a similar task in the robot environment.

More formally, we consider a set of Markov Decision Processes (MDP) $\mathcal{MDP}_i = (\mathcal{S}, \mathcal{A}, \mathcal{R}_i, p)_i$ sharing the same observation space \mathcal{S} , action space \mathcal{A} and dynamics p but with different reward functions \mathcal{R}_i corresponding to different tasks we would like to solve. We do not assume that the

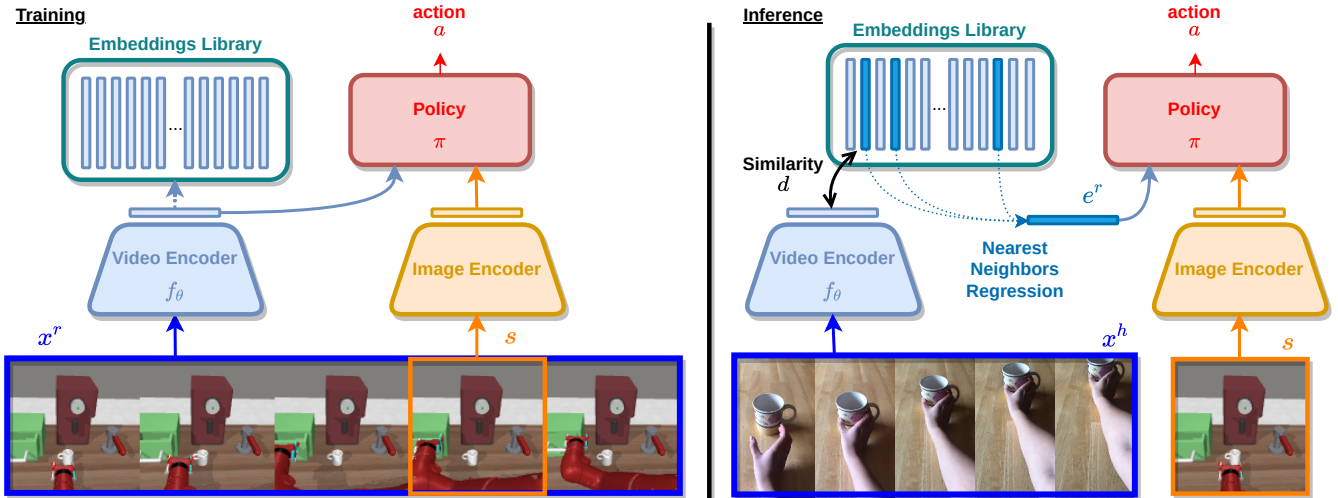


Fig. 2. (Left) During training, we learn a manipulation policy conditioned on robot video embeddings of the full robot trajectories from the robot dataset. At the same time, the robot video embedding of each trajectory in the robot dataset is added to an embeddings library. (Right) At inference, we encode the human video instruction into a human video embedding. We then average the robot embeddings from the library that have highest cosine similarity to the human embedding into a selected robot embedding. Finally, we execute the policy conditioned on this selected embedding.

reward functions \mathcal{R}_i are observed. Instead, \mathcal{R}_i must be inferred through a human video of the task $x^h \in \mathcal{X}$. Our goal is to learn a video-conditioned controller $\pi(\cdot|s, x^h)$ that predicts actions $a \in \mathcal{A}$ given the current states $s \in \mathcal{S}$ and human videos $x^h \in \mathcal{X}$ to maximize the reward functions associated with the input human video.

There may exist a large domain gap between videos of humans and robots performing similar tasks. To bridge this gap, we leverage the large-scale video dataset Something-Something-v2 (SSv2) [17] with labeled human actions $D^h = \{x_i^h, y_i^h\}_i$ where labels y_i^h for videos x_i^h correspond to different manipulation actions such as Opening Something, Moving Something Away from the Camera, etc. Following DVD [15], we train a similarity function $d(\cdot, \cdot)$ that assigns high values to pairs of videos representing the same task and low values to video pairs of different tasks. Unlike DVD, however, we learn such similarity without any annotated robot videos. Given a human video, we use the learned similarity as a reward $R(\cdot) = d(\cdot, x^h)$ for our controller.

To learn the policy, we assume access to a dataset of unlabelled robot demonstrations $D^r = \{x_i^r, (s_i^t)_t, (a_i^t)_t\}_i$, where $(s_i^t)_t$ is a sequence of robot observations, $(a_i^t)_t$ is the corresponding sequence of executed robot actions and x_i^r is a video of the demonstration. For instance, if the observation space \mathcal{S} only contains images (without e.g. proprioceptive information), the videos can simply correspond to the whole sequence of states in the trajectory $x_i^r = (s_i^t)_t$. This offline dataset can be collected in many different ways: by expert demonstrators, rollouts of other policies, through teleoperation or by random data generation. Importantly, we do not assume access to any further information about these demonstrations.

Figure 2 presents an overview of our approach: we leverage a video encoder f_θ trained for human action recognition on SSv2 and a similarity metric between videos. During training, we learn a behavior cloning policy conditioned on

robot embeddings given by the video encoder while storing all embeddings of the robot training dataset in a library. At inference, we first use this library and the similarity metric to predict a robot embedding relevant to the human video instruction, then rollout the policy in the environment.

B. Video-conditioned policy learning

We now describe how to encode a large array of behaviors into a single policy. A key to our approach is the use of the embedding space of a video encoder pretrained on human videos to condition our policy. This video embedding can be seen as a task embedding for our generic multi-task policy. We will explain more in detail how we can obtain meaningful video embeddings for control in Section III-D.

During training, we learn a policy π_ϕ to regress an action given the current state and a video embedding of a full robot trajectory. Video embeddings act as context defining the global task. The policy is trained with behavior-cloning by minimizing the loss:

$$\mathcal{L}_\pi(\phi) = -\mathbb{E}_{s, a, x^r \sim D^r} \log \pi_\phi(a|s, f_\theta(x^r)). \quad (1)$$

At test time, this policy can be commanded to reproduce a robot video input x^r by first encoding the video into an embedding e^r and then executing the actions $a \sim \pi(\cdot|s, e^r)$ predicted by the policy at each step s visited during the roll-out. As we will see from experiments, however, using human videos to directly condition the policy results in poor performance due to the large domain gap between robot and human videos. We therefore propose to first translate human videos to robot video embeddings as described in the next section.

C. Inference with human instructions

At inference, our first step is to translate the human video instruction x^h into the robot video embedding e^h that both (1) corresponds to the target task and (2) is in distribution for the robot policy. While many methods can instantiate



Fig. 3. Illustrations of environments used in our experiments. From left to right: TableTop env1-env4 followed by Kitchen left and right views.

this, we choose to simply use nearest neighbors regression in the robot embedding space. We first encode the videos contained in the robot dataset D^r into a library of robot embeddings $D^e = (e_i^r = f_\theta(x_i^r))_i$. We also encode the video instruction into a human embedding e^h . We then perform k nearest neighbors regression using the distance function d , by first computing all the distances between the human embedding and each embeddings in the library $d_i = d(e^h, e_i^r)$, then averaging the top k embeddings of the library $e^r = \frac{1}{k} \sum_{i \in 1}^k e_i^r$. Finally, we perform policy rollout conditioned on this embedding. As result, although the policy was trained with behavior cloning, this approach allows us to maximize the similarity of the robot trajectory to the video prompt at inference.

D. Learning a task similarity from human videos

Many choices of distance metric d between videos can be used to match a human video to an appropriate robot embedding. In this work, we consider adapting Supervised Contrastive Learning [18] to our video action recognition task on the Something-Something-v2 dataset. We learn our video encoder $f_\theta(\cdot) = f_\theta^p(f^b(\cdot))$ composed of a backbone $f^b(\cdot)$ which maps videos to representation vectors and a projection network $f_\theta^p(\cdot)$ which maps representation vectors to embedding vectors e , such that embeddings from the same class are pulled closer together in cosine distance than embeddings from different classes. As a result, the cosine distance characterizes the similarity between two videos.

In contrastive representation learning, the backbone and projection net are typically trained end-to-end from scratch and, after training, the projection net is discarded and a classifier head is learned on top of the representations. Instead, we start from available pretrained backbones for SSv2 classification and simply train the projection net using the Supervised Contrastive Learning loss. Given a batch of N video/label pairs sampled from the SSv2 dataset $\{x_k^h, y_k^h\}_{k \in [1, \dots, N]} \sim D^h$, we build a multiview batch consisting of $2N$ pairs, $\{\tilde{x}_l^h, \tilde{y}_l^h\}_{l \in [1, \dots, 2N]}$, where \tilde{x}_{2k-1}^h and \tilde{x}_{2k}^h are two random augmentations of video x_k^h and $\tilde{y}_{2k-1}^h = \tilde{y}_{2k}^h = y_k^h$. We train f_θ to minimize the Supervised Contrastive Loss:

$$\mathcal{L}_{SupCon}(\theta) = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(\langle f_\theta(\tilde{x}_i^h), f_\theta(\tilde{x}_p^h) \rangle / \tau)}{\sum_{a \in A(i)} \exp(\langle f_\theta(\tilde{x}_i^h), f_\theta(\tilde{x}_a^h) \rangle / \tau)} \quad (2)$$

where $\langle \cdot, \cdot \rangle$ is the cosine similarity, $I = [1, \dots, 2N]$, $A(i) = I \setminus \{i\}$, $P(i) = \{p \in A(i) : \tilde{y}_p^h = \tilde{y}_i^h\}$ and τ is a hyper-parameter. After training, we use the distance $d(x^h, x^r) = \langle f_\theta(x^h), f_\theta(x^r) \rangle$ between videos x^h and x^r as measure of

similarity that focuses on the semantic aspects of the video. As we show in the experimental section, despite being trained only on human videos, this similarity metric generalizes to robot manipulation tasks. In our experiments, we use the same video backbone as [40] and [15].

Alternative options for d include the DVD similarity network [15], which learns to classify whether two videos correspond to the same task and uses the classification score between human and robot videos as a distance metric d on top of the video encoder f^b . While our approach implicitly corresponds to the same classification objective, supervised contrastive learning readily embeds our encoder with a similarity metric.

IV. EXPERIMENTS

This section presents experiments validating our proposed approach and its training procedure. We evaluate our method on several tasks in two challenging environments and compare results to the state-of-the-art method [15]. In particular, we present ablations of our method and show its advantage in zero-shot settings, i.e. for tasks that have not been observed during training.

A. Experimental Setup

For our experiments, we consider the TableTop environments introduced in [15]. The agent controls a robot arm in four variations of a simulated environment containing a drawer, a cup in front of a coffee machine and a faucet handle. The four environments differ by object positions, camera locations and colors. Robot experience for policy training is collected by controlling the robot end effector to go through three keypoints randomly sampled in the environment. We follow the experimental setup of [15] and consider three tasks: close the drawer, push the cup and turn the faucet to the right. The image encoder representing the current state of the environment is learned end-to-end together with the policy. These environments challenge the ability of our approach to generate meaningful manipulation skills from random robot trajectories.

Furthermore, we consider the Kitchen environment initially introduced in [10]. The agent controls a robot arm with a clamp in a kitchen environment containing diverse objects: a microwave, three doors, a kettle and light and knob switches. We consider 3 opening tasks: open the microwave, open the left door and open the sliding door. Robot demonstrations accomplishing these tasks are available from [34]. However, these demonstrations are not annotated and the agent learns jointly from all these demonstrations without knowing what each demonstration accomplishes. For the

TABLE I
RESULTS FOR THE TABLETOP ENVIRONMENT USING ROBOT DEMONSTRATIONS. DVD RESULTS ARE OBTAINED FROM [15] EXCEPT THE ONES MARKED WITH "*" FOR WHICH WE RUN THE CODE PROVIDED BY THE AUTHORS.

Setting	Method	env 1	env 2	env 3	env 4	Avg
seen robot demos	DVD [15]	65.2 ^(4.1)	62.3 ^(2.8)	53.8 ^(3.9)	39.6 ^(0.7)	55.2 ^(2.9)
	ViP (DVD similarity)	97.1 ^(2.8)	72.0 ^(15.9)	82.1 ^(16.4)	42.0 ^(11.2)	73.3 ^(11.6)
unseen robot demos	DVD [15]	55.1 ^(2.0)	51.1 * ^(2.5)	38.4* ^(1.6)	35.0* ^(1.9)	44.9* ^(2.0)
	ViP (DVD similarity)	68.2 ^(11.2)	55.2 ^(13.0)	60.3 ^(8.0)	43.6 ^(8.0)	56.8 ^(10.0)

Kitchen environment we train the policy on top of R3M image representations following [34] using all demonstrations. This environment is challenging as it requires to distinguish which task is the one intended by the user among very similar opening tasks.

Figure 3 illustrates the environments. Following [15], for each of these tasks, we consider three different human video instructions collected in diverse environments to prompt our method at inference. Note that, in both environments, our approach can use the same video encoder and similarity metric.

B. Comparison to DVD [15]

We first validate our video-conditioned control policy and compare it to the planning method of DVD [15] on the TableTop environments in the following settings:

- *Seen robot demos*: the video similarity is trained on a subset of task-related SSv2 classes and on labelled videos of robot demonstrations for the target tasks collected in env 1 and the rearranged version of env 1.
- *Unseen robot demos*: the video similarity is trained on a subset of task-related SSv2 classes and on labelled videos of robot demonstrations corresponding to non-target tasks in env 1 and the rearranged version of env 1.

We compare our approach to [15] that uses the DVD similarity on top of a learned action-conditioned video prediction model for control [41]. For the *Seen robot demos* setting we take results reported in [15]. For the *Unseen robot demos*, we take the best success rate reported in [15] for environment 1 and run the code of the authors to obtain results for environments 2, 3 and 4 (marked with *). For a fair comparison of our control policy, we here run ViP using the same video similarity function as in DVD which was trained on paired data between human videos and robot demonstrations.

Table I presents results for the TableTop environments where, for each environment, we report the average success rates across the three tasks and three human video instructions. ViP control policy outperforms DVD in both settings and most environments. We believe these improvements are due to comparing the human video prompt to full robot trajectories instead of shorter sub-trajectories and not relying on synthetic video generation for control as in [15]. To demonstrate stability of our approach, we report results over four training seeds while [15] reports results over evaluation runs. Moreover, our method is significantly faster than DVD as it operates in the space of pre-computed embeddings, while DVD requires to execute its video prediction model for

each sampled trajectory. As result, in our experiments ViP requires less than 1 second to complete an episode in the TableTop environment while it takes more than 16 seconds to complete the same task with DVD.

We observe that the gains obtained by ViP compared to DVD are lower in the *unseen robot demos* setting. This can be attributed to the fact that our approach relies more on human videos of the target tasks to achieve good results.

C. ViP without paired data

While DVD uses paired data with robot and human videos demonstrating execution of similar tasks, obtaining such data is cumbersome at scale, see discussion in Section I. In this section we demonstrate that ViP is able to cope with a more challenging setting where the training of video similarity is done using pairs of human videos only and without access to any robot data.

Table II presents our results. In this setting, our approach (*ViP (Ours)*) significantly outperforms DVD, succeeding to perform intended tasks more than 70% of the time on average across environments, while DVD performs only slightly better than chance. Moreover, we can see that our results in this setting are similar to results in Table I where paired robot demonstrations are used for similarity learning. Indeed, having a global understanding of full robot trajectories makes our action recognition approach less reliant on robot demos, which are more important when planning for short horizon as in DVD. Finally, ViP with the DVD similarity metric also performs well, showing that our approach can accommodate alternative choices of video similarity. Qualitative results of ViP are presented in Figure 1 and in the supplementary video, where we show that failure cases include failing to manipulate the desired object and failure to perform the manipulation task.

We further compare our approach to a baseline where we bypass our translation module and directly use human video embeddings to condition the ViP policy (*Human video as input*). Since ViP was only trained on robot video embeddings, embeddings of human videos are out of its training distribution. As result, the policy fails to perform the intended tasks. This highlights the importance of our translation procedure which matches human videos to a library of robot demonstrations.

In Figure 4 we evaluate the sensitivity of our method to the number of nearest neighbor samples k used for translation of human videos in Section III-C. High values of k ensure that the generated embedding e^r is an average of many appropriate robot embeddings, whereas low values of k prioritize

TABLE II

RESULTS FOR THE TABLETOP ENVIRONMENT WITHOUT PAIRED DATA. DVD RESULTS ARE OBTAINED BY RUNNING THE CODE OF THE AUTHORS.

Method	env 1	env 2	env 3	env 4	Avg
Random	25.6	25.1	22.2	18.2	22.8
Human video as input	22.7 (7.9)	32.9 (12.9)	11.2 (7.8)	14.3 (9.6)	20.3 (9.6)
DVD[15]	43.0* (2.6)	44.3* (2.0)	32.8* (1.7)	27.0* (2.6)	36.8* (2.2)
ViP (Cosine distance in repr. space)	36.9 (4.5)	66.4 (17.5)	39.3 (15.0)	36.4 (15.4)	44.8 (13.1)
ViP (DVD similarity)	50.8 (10.2)	67.3 (17.4)	72.6 (16.3)	53.4 (9.6)	61.0 (13.4)
ViP (Ours)	79.9 (11.4)	77.9 (16.2)	70.6 (17.8)	56.9 (8.7)	71.3 (13.5)

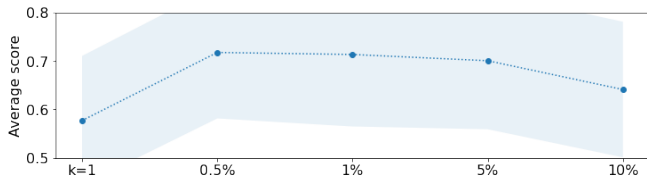


Fig. 4. Average success rate achieved by the policy on the TableTop environments for different values of k . We compare $k = 1$ against values of k corresponding to different percentages of the library size.

maximising the similarity score between the predicted robot embedding and the human instruction embedding. While our approach is relatively robust to this hyperparameter, too high values of k result in lower performance. On the other side, for $k = 1$, meaning that we choose a robot demonstration with the highest similarity to the human video, the performance drops. We hypothesise that performing a nearest-neighbor regression has a regularization effect, as naively maximizing the similarity might result in choosing an adversarial robot embedding that is misleadingly for the target task. We set k to 1% of the library size for the TableTop environments and to 0.5% for the Kitchen environment.

In Table II we also ablate an alternative choice for video similarity where video embeddings obtained from the SSv2 action classification network are directly compared using a cosine distance measure (*Cosine distance in repr. space*). Using such a naive video similarity together with ViP results in superior performance compared to DVD, however, it is outperformed by both the DVD similarity with ViP (*ViP (DVD similarity)*) and our full approach, highlighting the importance of training an explicit distance between videos for the success of our method.

D. Kitchen environment

We evaluate our approach on the kitchen environments in the zero-shot setting. We compare our approach against a version of our method where we select a robot demonstration that solves the target task (*ViP (Oracle)*) as well as a version that randomly selects robot demonstrations from the library (*ViP (Random)*). We also compare against single-task R3M [34], where we trained specialized policies for each task using all the demonstrations available.

We report the success rates for each task averaged over the 3 corresponding human video instructions across 4 training seeds in Table III. Comparison between R3M and *Oracle* shows that our approach successfully learns from multi-task data for precise manipulation skills, achieving competitive results to R3M trained for single tasks. When prompted with a human video instruction in the left camera view

(L), our approach successfully opens the correct door when presented with human videos of opening common doors and microwaves. However, the method fails to map human videos of opening sliding doors to appropriate robot skills. On the right camera view (R), our approach struggles more, often opening the wrong object when prompted with videos of sliding doors and microwaves. These results show the limit of training on SSv2, where different human videos of opening something are encouraged to be grouped together by our similarity metric disregarding manipulated objects.

TABLE III
RESULTS ON THE KITCHEN ENVIRONMENT

	Method	left door	sliding door	microwave
L	ViP (Random)	22.5 (2.7)	29.8 (0.4)	13.5 (4.7)
	ViP (Oracle)	83.2 (16.5)	99.2 (0.8)	43.0 (10.2)
	R3M[34]	63.8 (10.3)	100 (0.0)	53.8 (11.0)
	ViP (Ours)	69.4 (11.0)	41.7 (27.6)	25.1 (8.5)
R	ViP (Random)	22.0 (2.9)	30.0 (0.0)	8.5 (3.0)
	ViP (Oracle)	92.0 (7.4)	99.8 (0.4)	27.2 (9.1)
	R3M[34]	57.5 (2.9)	100 (0.0)	30.0 (0.1)
	ViP (Ours)	66.7 (3.3)	41.7 (14.4)	6.6 (2.9)

V. CONCLUSION

We propose ViP, a method that learns to map human video instructions to robot skills in a zero-shot manner. We show that by conditioning on video embeddings, we can learn from multi-task robot data without supervision and prompt our policy with an unseen human video instruction at test time. As a step further towards less supervision in data-driven robotics, we demonstrate that, by training on large datasets of diverse labelled human videos, we don't need to pair human instructions to robot data during training.

Our experiments demonstrate that ViP can accommodate many different similarity metrics between human instructions and robot manipulations. Future work could explore other forms of similarities, such as mapping language instructions to robotic skills.

Acknowledgements: This work was funded in part by the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR19-P3IA-0001 (PRAIRIE 3IA Institute), the ANR project VideoPredict, reference ANR-21-FAI1-0002-01, and the Louis Vuitton ENS Chair on Artificial Intelligence. This work was granted access to the HPC resources of IDRIS under the allocation 2021-AD011012947 made by GENCI. We would like to thank Mathis Clautrier and Minttu Alakuijala for helpful feedbacks.

REFERENCES

- [1] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog *et al.*, “Do as i can, not as i say: Grounding language in robotic affordances,” *arXiv preprint arXiv:2204.01691*, 2022.
- [2] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn, “Bc-z: Zero-shot task generalization with robotic imitation learning,” in *Conference on Robot Learning*. PMLR, 2022, pp. 991–1002.
- [3] Y. Lu, K. Hausman, Y. Chebotar, M. Yan, E. Jang, A. Herzog, T. Xiao, A. Irpan, M. Khansari, D. Kalashnikov *et al.*, “Aw-opt: Learning robotic skills with imitation and reinforcement at scale,” in *Conference on Robot Learning*. PMLR, 2022, pp. 1078–1088.
- [4] Y. Chebotar, K. Hausman, Y. Lu, T. Xiao, D. Kalashnikov, J. Varley, A. Irpan, B. Eysenbach, R. Julian, C. Finn *et al.*, “Actionable models: Unsupervised offline reinforcement learning of robotic skills,” *arXiv preprint arXiv:2104.07749*, 2021.
- [5] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
- [6] D. Ghosh, A. Gupta, A. Reddy, J. Fu, C. Devin, B. Eysenbach, and S. Levine, “Learning to reach goals via iterated supervised learning,” *arXiv preprint arXiv:1912.06088*, 2019.
- [7] C. Lynch, M. Khansari, T. Xiao, V. Kumar, J. Tompson, S. Levine, and P. Sermanet, “Learning latent plans from play,” in *Conference on robot learning*. PMLR, 2020, pp. 1113–1132.
- [8] B. Eysenbach, R. Salakhutdinov, and S. Levine, “C-learning: Learning to achieve goals via recursive classification,” *arXiv preprint arXiv:2011.08909*, 2020.
- [9] A. V. Nair, V. Pong, M. Dalal, S. Bahl, S. Lin, and S. Levine, “Visual reinforcement learning with imagined goals,” *Advances in neural information processing systems*, vol. 31, 2018.
- [10] A. Gupta, V. Kumar, C. Lynch, S. Levine, and K. Hausman, “Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning,” *arXiv preprint arXiv:1910.11956*, 2019.
- [11] V. Pong, S. Gu, M. Dalal, and S. Levine, “Temporal difference models: Model-free deep rl for model-based control,” *arXiv preprint arXiv:1802.09081*, 2018.
- [12] C. Lynch and P. Sermanet, “Language conditioned imitation learning over unstructured data,” *arXiv preprint arXiv:2005.07648*, 2020.
- [13] S. Nair, E. Mitchell, K. Chen, S. Savarese, C. Finn *et al.*, “Learning language-conditioned robot behavior from offline data and crowd-sourced annotation,” in *Conference on Robot Learning*. PMLR, 2022, pp. 1303–1315.
- [14] O. Mees, L. Hermann, and W. Burgard, “What matters in language conditioned robotic imitation learning,” *arXiv preprint arXiv:2204.06252*, 2022.
- [15] A. S. Chen, S. Nair, and C. Finn, “Learning generalizable robotic reward functions from” in-the-wild” human videos,” *arXiv preprint arXiv:2103.16817*, 2021.
- [16] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard, “Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks,” *IEEE Robotics and Automation Letters*, 2022.
- [17] R. Goyal, S. Ebrahimi Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag *et al.*, “The” something something” video database for learning and evaluating visual common sense,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5842–5850.
- [18] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, “Supervised contrastive learning,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 18 661–18 673, 2020.
- [19] A. Miech, D. Zhukov, J.-B. Alayrac, M. Tapaswi, I. Laptev, and J. Sivic, “Howto100m: Learning a text-video embedding by watching hundred million narrated video clips,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2630–2640.
- [20] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. 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*, 2022, pp. 18 995–19 012.
- [21] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price *et al.*, “Scaling egocentric vision: The epic-kitchens dataset,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 720–736.
- [22] X. B. Peng, A. Kanazawa, J. Malik, P. Abbeel, and S. Levine, “Sfv: Reinforcement learning of physical skills from videos,” *ACM Transactions On Graphics (TOG)*, vol. 37, no. 6, pp. 1–14, 2018.
- [23] H. Xiong, Q. Li, Y.-C. Chen, H. Bharadhwaj, S. Sinha, and A. Garg, “Learning by watching: Physical imitation of manipulation skills from human videos,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 7827–7834.
- [24] N. Das, S. Bechtle, T. Davchev, D. Jayaraman, A. Rai, and F. Meier, “Model-based inverse reinforcement learning from visual demonstrations,” *arXiv preprint arXiv:2010.09034*, 2020.
- [25] Y. Liu, A. Gupta, P. Abbeel, and S. Levine, “Imitation from observation: Learning to imitate behaviors from raw video via context translation,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1118–1125.
- [26] P. Sharma, D. Pathak, and A. Gupta, “Third-person visual imitation learning via decoupled hierarchical controller,” *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [27] L. Smith, N. Dhawan, M. Zhang, P. Abbeel, and S. Levine, “Avid: Learning multi-stage tasks via pixel-level translation of human videos,” *arXiv preprint arXiv:1912.04443*, 2019.
- [28] S. Bahl, A. Gupta, and D. Pathak, “Human-to-robot imitation in the wild,” *arXiv preprint arXiv:2207.09450*, 2022.
- [29] S. Pirk, M. Khansari, Y. Bai, C. Lynch, and P. Sermanet, “On-line object representations with contrastive learning,” *arXiv preprint arXiv:1906.04312*, 2019.
- [30] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, S. Levine, and G. Brain, “Time-contrastive networks: Self-supervised learning from video,” in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 1134–1141.
- [31] V. Petrik, M. Tapaswi, I. Laptev, and J. Sivic, “Learning object manipulation skills via approximate state estimation from real videos,” *arXiv preprint arXiv:2011.06813*, 2020.
- [32] A. Bonardi, S. James, and A. J. Davison, “Learning one-shot imitation from humans without humans,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3533–3539, 2020.
- [33] T. Yu, C. Finn, A. Xie, S. Dasari, T. Zhang, P. Abbeel, and S. Levine, “One-shot imitation from observing humans via domain-adaptive meta-learning,” *arXiv preprint arXiv:1802.01557*, 2018.
- [34] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta, “R3m: A universal visual representation for robot manipulation,” *arXiv preprint arXiv:2203.12601*, 2022.
- [35] T. Xiao, I. Radosavovic, T. Darrell, and J. Malik, “Masked visual pre-training for motor control,” *arXiv preprint arXiv:2203.06173*, 2022.
- [36] A. D. Edwards and C. L. Isbell, “Perceptual values from observation,” *arXiv preprint arXiv:1905.07861*, 2019.
- [37] K. Schmeckpeper, O. Rybkin, K. Daniilidis, S. Levine, and C. Finn, “Reinforcement learning with videos: Combining offline observations with interaction,” *arXiv preprint arXiv:2011.06507*, 2020.
- [38] K. Schmeckpeper, A. Xie, O. Rybkin, S. Tian, K. Daniilidis, S. Levine, and C. Finn, “Learning predictive models from observation and interaction,” in *European Conference on Computer Vision*. Springer, 2020, pp. 708–725.
- [39] Y. Seo, K. Lee, S. L. James, and P. Abbeel, “Reinforcement learning with action-free pre-training from videos,” in *International Conference on Machine Learning*. PMLR, 2022, pp. 19 561–19 579.
- [40] L. Shao, T. Migimatsu, Q. Zhang, K. Yang, and J. Bohg, “Concept2robot: Learning manipulation concepts from instructions and human demonstrations,” *The International Journal of Robotics Research*, vol. 40, no. 12-14, pp. 1419–1434, 2021.
- [41] M. Babaeizadeh, C. Finn, D. Erhan, R. H. Campbell, and S. Levine, “Stochastic variational video prediction,” *arXiv preprint arXiv:1710.11252*, 2017.
- [42] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.
- [43] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9729–9738.
- [44] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, “Learning transferable visual models from natural language supervision,” in *International Conference on Machine Learning*. PMLR, 2021, pp. 8748–8763.

- [45] M. Shridhar, L. Manuelli, and D. Fox, “Cliport: What and where pathways for robotic manipulation,” in *Conference on Robot Learning*. PMLR, 2022, pp. 894–906.
- [46] L. Fan, G. Wang, Y. Jiang, A. Mandlekar, Y. Yang, H. Zhu, A. Tang, D.-A. Huang, Y. Zhu, and A. Anandkumar, “Minedojo: Building open-ended embodied agents with internet-scale knowledge,” *arXiv preprint arXiv:2206.08853*, 2022.
- [47] S. Emmons, B. Eysenbach, I. Kostrikov, and S. Levine, “Rvs: What is essential for offline rl via supervised learning?” *arXiv preprint arXiv:2112.10751*, 2021.
- [48] H. Furuta, Y. Matsuo, and S. S. Gu, “Generalized decision transformer for offline hindsight information matching,” *arXiv preprint arXiv:2111.10364*, 2021.
- [49] A. Kumar, X. B. Peng, and S. Levine, “Reward-conditioned policies,” *arXiv preprint arXiv:1912.13465*, 2019.
- [50] J. Schmidhuber, “Reinforcement learning upside down: Don’t predict rewards—just map them to actions,” *arXiv preprint arXiv:1912.02875*, 2019.
- [51] R. K. Srivastava, P. Shyam, F. Mutz, W. Jaśkowski, and J. Schmidhuber, “Training agents using upside-down reinforcement learning,” *arXiv preprint arXiv:1912.02877*, 2019.
- [52] 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.
- [53] M. Janner, Q. Li, and S. Levine, “Offline reinforcement learning as one big sequence modeling problem,” *Advances in neural information processing systems*, vol. 34, pp. 1273–1286, 2021.
- [54] Y. Ding, C. Florensa, P. Abbeel, and M. Phielipp, “Goal-conditioned imitation learning,” *Advances in neural information processing systems*, vol. 32, 2019.