2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Universal Visual Decomposer: Long-Horizon Manipulation Made Easy

Zichen Zhang*[1], Yunshuang Li*[2], Osbert Bastani[2], Abhishek Gupta[3],
Dinesh Jayaraman[2], Yecheng Jason Ma†[2], Luca Weihs†[1]
zcczhang.github.io/UVD

***Abstract*— Real-world robotic tasks stretch over extended horizons and encompass multiple stages. Learning long-horizon manipulation tasks, however, is a long-standing challenge, and demands decomposing the overarching task into several manageable subtasks to facilitate policy learning and generalization to unseen tasks. Prior task decomposition methods require task-specific knowledge, are computationally intensive, and cannot readily be applied to new tasks. To address these shortcomings, we propose Universal Visual Decomposer (UVD), an off-the-shelf task decomposition method for visual long-horizon manipulation using pre-trained visual representations designed for robotic control. At a high level, UVD discovers subgoals by detecting phase shifts in the embedding space of the pre-trained representation. Operating purely on visual demonstrations without auxiliary information, UVD can effectively extract visual subgoals embedded in the videos, while incurring zero additional training cost on top of standard visuomotor policy training. Goal-conditioned policies learned with UVD-discovered subgoals exhibit significantly improved compositional generalization at test time to unseen tasks. Furthermore, UVD-discovered subgoals can be used to construct goal-based reward shaping that jump-starts temporally extended exploration for reinforcement learning. We extensively evaluate UVD on both simulation and real-world tasks, and in all cases, UVD substantially outperforms baselines across imitation and reinforcement learning settings on in-domain and out-of-domain task sequences alike, validating the clear advantage of automated visual task decomposition within the simple, compact UVD framework.**

I. Introduction

Real-world household tasks, such as cooking and tidying, often stretch over extended horizons and encompass multiple stages. In order for robots to be deployed in realistic environments, they must possess the capability to learn and perform long-horizon manipulation tasks from visual observations. Learning vision-based complex skills over long timescales, however, is challenging due to the problem of compounding errors, the vastness of the action and observation spaces, and the difficulty in providing meaningful learning signals for each step of the task.

Given these challenges, it is necessary to *decompose* a long-horizon task into several smaller subtasks to make learning manageable. Beyond improving the efficiency of learning, task decomposition facilitates learning reusable skills, promotes data-sharing across different trajectories, and further enables compositional generalization to unseen sequences of the learned subtasks. Despite its usefulness, task decomposition is difficult to perform in practice, and most existing approaches require strong assumptions about tasks, datasets, or robotic platforms [3,10,11,16,17,26,29,34,39–41,46]. These methods cannot be used in common settings where the agent only has access to video demonstrations of desired behavior on their robotic hardware and little else, motivating the need for an off-the-shelf approach that can readily decompose *any* visual demonstration out-of-the-box.

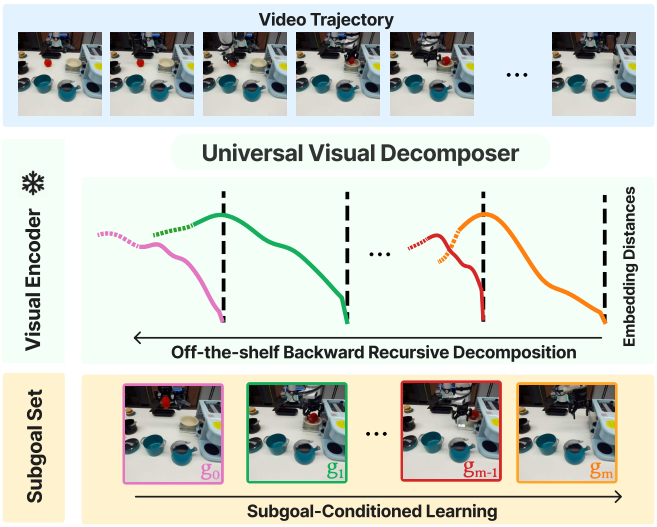

Fig. 1: **Universal Visual Decomposer** uses off-the-shelf pre-trained visual representations to find subgoals from video demonstrations by recursively computing embedding distances from the target goal and setting the first plateau as the new target goal.

In order to decompose any long-horizon task using vision, general knowledge about visual task progression that can discern embedded subtasks in long, unsegmented task videos must be acquired. In this work, we propose Universal Visual Decomposer (UVD), an off-the-shelf unsupervised subgoal decomposition method that re-purposes state-of-the-art pre-trained visual representations [19,23–25,30,33,45] for automated task segmentation. To motivate our approach, we observe that several pre-trained visual representations, such as VIP [24] and R3M [30], are trained to capture temporal task progress on diverse, short videos of humans accomplishing goal-directed behavior [6,13]. These representations have acquired well-behaved embedding distances that can progress near *monotonically* along video frames that depict short-horizon, atomic skills. Our key insight is that when applied to long videos consisting of several subtasks, their training

*Equal Contribution. †Equal Advising
[1]Allen Institute for AI, [2]University of Pennsylvania, [3]University of Washington

on short atomic tasks makes these representations no longer informative about subtask membership. That is, they are not trained to capture whether an earlier frame, which may very well belong to a different subtask, is making progress towards a subtask that appears later in the video, even if the subtasks are related to one another. As a consequence, when the robot task is extended, the embedding distances will *deviate* from monotonicity and exhibit plateaus around frames that correspond to phase shifts in the overall task; this provides an unsupervised signal for detecting when subtasks have taken place in the original long, unsegmented task video. UVD instantiates this insight and proposes an out-of-the-box subgoal discovery procedure that can iteratively extract subgoals using the embedding distance information from the end to the beginning; notably, UVD does not require any domain-specific knowledge or incur additional training cost on top of standard visuomotor policy training. Given its off-the-shelf nature, UVD can be readily applied to a variety of unseen robot domains. See Fig. 1 for a conceptual overview of our approach.

We apply UVD to long-horizon, multi-stage visual manipulation tasks in both simulation and real-world environments. Across these tasks, UVD consistently outputs semantically meaningful subgoals which are used for policy training and evaluation. We consider both in-domain (IND) and out-of-domain (OOD) task evaluations. In IND evaluation, the agent is evaluated on long-horizon tasks for which it has been explicitly trained whereas in OOD evaluation, the agent is evaluated to generalize to new tasks unseen during training. Using UVD-discovered subgoals, we demonstrate substantial policy improvements across these evaluation settings. Firstly, when training agents with reinforcement learning (RL), we show that UVD-subgoals can be used to perform *reward shaping* for each of the intermediate subtasks. Using this approach, we demonstrate that the resulting rewards can successfully guide a vision-based reinforcement learning agent to learn long-horizon tasks in the FrankaKitchen [14] environment. Secondly, when training agents with imitation learning (IL), by virtue of discovering semantically meaning subgoals, our policies can *compositionally generalize* to OOD task sequences unseen during training; this capability greatly reduces the burden of manual data collection for every desired task. Finally, in IND evaluation, we also demonstrate performance improvement on several real-world multi-stage tasks that stretch over several hundred timesteps and exhibit sequential dependency among the subtasks.

In summary, our contributions include:

1) Universal Visual Decomposer (UVD), an off-the-shelf visual decomposition method for long-horizon manipulation using pre-trained visual representations.
2) A reward shaping method for long-horizon visual reinforcement learning using UVD-discovered subgoals.
3) Extensive experiments demonstrating UVD's effectiveness in improving policy performance on IND and OOD evaluations across several simulation and real-robot tasks.

II. Related Work

Learning long-horizon skills has been a long standing challenge in robotic manipulation [14,15,21,26]. Hierarchical reinforcement learning [1,2,4,5,9,14,27,28,31,40,42,47] enables temporally extended exploration by discovering subskills and planning over them. However, these algorithms learn subskills and overall policies from scratch, which is computationally expensive and less suitable for real-world robotics use cases.

When provided with task demonstrations, there are many prior efforts on using subgoal decomposition as a means to break up the long task in order to provide intermediate learning signals and to mitigate compounding action errors. These prior decomposition strategies, however, require task-specific knowledge and cannot be easily applied to new tasks. For example, several approaches use the robot's proprioceptive data within the task demonstrations [3,17,39,41] or explicit knowledge about subtask structure [16,46] to guide decomposition; this limits the types of tasks that can be solved and precludes learning from observed videos. Other works learn latent generative models over subgoals [10,11,18,26,29,34], but demand compute-intensive training on large datasets that cover diverse behavior.

To the best of our knowledge, Universal Visual Decomposer is the first "off-the-shelf" visual task decomposition method that does not require any task-specific knowledge or training. In addition, it demonstrates a novel use case of pre-trained visual representations. While some prior works have considered using pre-trained visual representations to generate rewards [24,38], we are the first to demonstrate that they can also be re-purposed to perform hierarchical decomposition; furthermore, this capability can be combined with the reward specification capability to solve long-horizon tasks using visual reinforcement learning.

III. Problem Setting

Unsupervised Subgoal Discovery (USD). Our goal is to derive a general-purpose subgoal decomposition method that can operate purely from visual inputs on a *per-trajectory* basis. That is, given a full-task demonstration $\tau = (o_0, ..., o_T)$,

$$\texttt{USD}((o_0, ..., o_T)) \rightarrow \tau_{goal} := (g_0, ..., g_m), \tag{1}$$

where $(g_0, ..., g_m)$ are the subset of τ that are selected as subgoals; m may vary across trajectories.

Policy Learning. We provide demonstrations $\mathcal{D} := \{\tau\}_{i=1}^{n}$ for the learning tasks; in the reinforcement learning setting, we assume that there is one task and have $n = 1$ to specify the overall task to be achieved. The evaluation tasks can be both in-domain (IND), the ground-truth sequences of tasks captured in $\mathcal{D}$, or out-of-domain (OOD), consisting of unseen combinations of the subtasks in $\mathcal{D}$.

We assume access to a pre-trained visual representation $\phi : \mathbb{R}^{H \times W \times 3} \rightarrow \mathbb{R}^K$ that maps RGB images to a K-dimensional embedding space. Given ϕ and $\mathcal{D}$, our goal is to learn a goal-conditioned policy $\pi : \mathbb{R}^K \times \mathbb{R}^K \rightarrow \Delta(A)$ that outputs an action based on the embedded observation and goal, $a \sim \pi(\phi(o), \phi(g))$. In the RL setting, the agent is not provided

with reward information, so the agent must also construct rewards using ϕ and $\mathcal{D}$.

Policy Evaluation. For OOD eval., we provide one demonstration τ specifying the subtask sequence to be performed.

IV. METHOD

We first present Universal Visual Decomposer, the core algorithm that powers our off-the-shelf subgoal discovery approach. Then, we discuss various ways we perform policy training as well as goal selection during policy inference.

A. Universal Visual Decomposer

Given an unlabeled video demonstration $\tau = (o_0, ..., o_T)$, how might we discover useful subgoals? The key intuition of Universal Visual Decomposer is that, conditioned on a goal frame o_t, some n frames $(o_{t-n}, ..., o_{t-1})$ preceding it must visually approach the goal frame; once we discover the first frame (o_{t-n}) in this goal-reaching sequence, the frame that precedes it (o_{t-n-1}) is then another subgoal. From o_{t-n-1}, the same procedure can be carried out *recursively* until we reach o_0. There are two central questions to address: (1) how to discover the first subgoal (last in terms of timestamp), and (2) how to determine the stopping point for the current subgoal and declare a new frame as the new subgoal.

The first question is simple to resolve by observing that in a demonstration, the last frame o_T is naturally a goal. Now, conditioned on a subgoal o_t, we attempt to extract the first frame o_{t-n} in the sub-sequence of frames that depicts visual task progression to o_t. To discover this first frame, we exploit the fact that several state-of-the-art pre-trained visual representations for robot control [23,24,30] are trained to capture temporal progress within short videos depicting a single solved task; these representations can effectively produce embedding distances that exhibit *monotone* trend over a short goal-reaching video sequence $\tau = (o_{t-n}, ..., o_t)$:

$$d_\phi(o_s; o_t) \geq d_\phi(o_{s+1}; o_t), \forall s \in \{t-n, \ldots, t-1\}, \quad (2)$$

where d_ϕ is a distance function in the ϕ-representation space; in this work, we set $d_\phi(o; o') := \|\phi(o) - \phi(o')\|_2$ because several state-of-the-art pre-trained representations use the L_2 distance as their embedding metric for learning. Given this, we set the previous subgoal to be the temporally closest observation to o_t for which this monotonicity condition fails:

$$o_{t-n-1} := \arg\max_{o_h} d_\phi(o_h; o_t) < d_\phi(o_{h+1}; o_t), h < t \ . \quad (3)$$

The intuition is that a preceding frame that belongs to the same subtask (i.e., visually apparent that it is progressing towards o_t) should have a higher embedding distance than the succeeding frame if the embedding distance indeed captures temporal progression. As a result, a deviation from the monotonicity indicates that the preceding frame may not exhibit a clear relation to the current subgoal, and instead be a subgoal itself. Now, o_{t-n-1} becomes the new subgoal, and we apply (3) recursively until the full sequence τ is exhausted. For instance, in Figure 1, conditioned on the last frame, g_3 is the first preceding frame that produces an inflection point in the embedding distances and hence selected as a subgoal; then, conditioned on g_3, g_2 is selected, and so on; see Alg. 1 for pseudocode. In practice, (2) may not hold for every step due to noise in the embedding space, and we find that a simple low pass filter procedure to first smoothen the embedding distances make the subgoal criterion (3) effective; see the supplementary website for details.

Algorithm 1: Universal Visual Decomposer

```
Init: frozen visual encoder φ, τ = {o_0, ··· , o_T}
Init: set of subgoals τ_goal = {}, t = T
while t not small enough do
    τ_goal = τ_goal ∪ {o_t}
    Find o_{t-n-1} from Eq. 3
    t = t − n − 1
end
```

Computational Efficiency. We highlight that our entire algorithm does not require any additional neural network training or forward computations on top of the one forward pass required to encode all observations for policy learning.

B. UVD-Guided Policy Learning

Now, we discuss several ways UVD-discovered subgoals can be used to supplement policy learning.

Goal Relabeling. As UVD is performed on a trajectory basis, we can relabel all observations in a trajectory with the closest subgoals that appear later in time. In particular, for an action-labeled trajectory $\tau = (o_0, a_0, ..., o_T, a_T)$ and UVD-discovered subgoals $\tau_{goal} = (g_0, ..., g_m)$, we have that $\texttt{Label}(o_t) = g_k$ where g_k is the first subgoal occurring after time t. This procedure leads to an augmented, goal-relabeled trajectory $\tau_{aug} = \{(o_0, a_0, g_0), ..., (o_T, a_T, g_m)\}$. Now, as all transitions are goal-conditioned, we can learn policies using any goal-conditioned imitation learning algorithm; for simplicity, we use goal-conditioned behavior cloning (GCBC) [8,12].

Reward Shaping. The above goal relabeling strategy applies to the imitation learning (IL) setting. Collecting the demonstrations needed for IL is, however, expensive. Instead, a reinforcement learning paradigm is feasible with much fewer demonstrations and comes with other ancillary benefits such as learned error recovery. This raises the question of how UVD-subgoals might be used with an RL paradigm. In particular, how can UVD help overcome the exploration challenge in long-horizon RL? Given that UVD selects subgoals so that the embedding distances in-between any two consecutive subgoals exhibit monotone trends, we define the *UVD reward* to be the goal-embedding distance *difference* computed using UVD goals:

$$R(o_t, o_{t+1}; \phi, g_i) := d_\phi(o_t; g_i) - d_\phi(o_{t+1}; g_i) \ . \quad (4)$$

where $g_i \in \tau_{goal}$, and g_i will be switched to g_{i+1} automatically during training when $d_\phi(o_{t+1}; g_i)$ is small enough. More details can be found on the supplementary website. This choice of reward encourages making consistent progress towards the goal and has been found in prior

work [7,24,43,44] to be particularly effective when deployed with suitable visual representations.

C. UVD Goal Inference

When deploying our trained subgoal-conditioned policies at inference time, we must determine what subgoals to instruct the policy to follow at each observation step. We study two simple strategies that work well in practice; we describe the high-level approaches here, and include more details on the supplementary website.

Nearest Neighbor. First, when there is only one fixed sequence of subtasks to be learned (*i.e.*, IND), we employ a simple nearest neighbor goal selection strategy. That is, for a new observation, we compute the observation in the training set that has the closest embedding (judged by d_ϕ) and use its associated sub-goal. This can be interpreted as a *non-parameteric* high-level policy that outputs observation-conditioned goal for the low level policy, $\pi(\phi(o), \phi(g))$.

Goal Relaying. When performing OOD or multi-task IND evaluation, the agent must complete a user-instructed task. In these settings, the above nearest neighbor approach may no longer apply as the subgoals seen in training may not be valid for the current, potentially unseen, task. Instead, we propose to relay the currently instructed goals based on embedding distance. Specifically, given a sequence of instructed subgoals $g = (g_0, ..., g_m)$, the policy will condition on the first remaining subgoal until the embedding distance between the current observation and the subgoal is below a certain threshold, at which point the policy will be conditioned on the next subgoal in the sequence.

V. EXPERIMENTS

We study the following research questions:

1) Does UVD enable compositional generalization in multi-stage and multi-task imitation learning?
2) Can UVD subgoals enable reward-shaping for long-horizon reinforcement learning?
3) Can UVD be deployed on real-robot tasks?

A. Simulation Experiments

FrankaKitchen Environment. We use the FrankaKitchen Environment [14] for simulation experiments. In the environment, a Franka robot with a 9-DoF torque-controlled action space can interact with seven objects: a microwave, a kettle, two stove burners, a light switch, a hinge cabinet, and a sliding cabinet. We refined the dataset from [14] to include only successful trajectories, yielding a total of 513 episodes gathered from humans using VR headsets. For each episode, four out of the seven objects are manipulated in an arbitrary sequence, leading to 24 unique completion orders; see Fig. 2.

Visual and Policy Backbones. As UVD is designed to utilize pre-trained visual representations that capture visual task progress, we adopt R3M [30], VIP [24], and LIV [23], three Resnet50-based representations trained with temporal objectives on video data; in particular, VIP and LIV are trained to explicitly encode smooth temporal task progress in their embedding distances. We also consider general

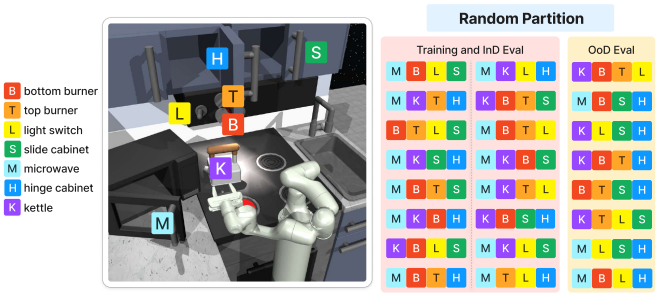

Fig. 2: **Frank Kitchen environment and an example of random training-evaluation partition.** In each demonstration episode, 4 out of 7 objects are manipulated in an arbitrary order. We show an example of 16 completion orders for training and IND evaluation chosen randomly, while the rest of 8 are for OOD generalizations.

vision models trained on static image datasets such as CLIP (ResNet50) [35] and DINO-v2 (ViT-large) [32] to assess the importance of training on temporal data. As our goal is to study the merit of the pretrained representations, as in prior works [24,30], we keep the policy architecture simple and employ a multi-layer perception (MLP) as the policy architecture; More details in supplementary website.

Baselines. We compare with goal-conditioned behavior cloning (GCBC) baselines to demonstrate the value of UVD. Fixing a choice of visual representation, the only difference of GCBC to ours is the how the goals are labeled at training time. For each observation, GCBC labels the final frame in the same trajectory as its goal.

IL Evaluation Protocol. Our training and evaluation design for FrankaKitchen is structured as follows: we train on n combinations of object sequences with IND evaluation, reserving the remaining $24-n$ task sequences for evaluation of unseen OOD scenarios. We use $n = 16$ by default unless otherwise mentioned. For a fair comparison, we utilize the same 3 random seen-unseen partitions, generated by 3 unique pre-defined seeds, for every set of runs.

To evaluate policy performance, we consider both the success rate on the overall task (**success**) as well as the number of subtasks accomplished (**completion**). The success criterion for each subtask is determined by the simulation ground-truth state; this is used solely during evaluations and is not provided to the agent during training. Results are presented in Table I.

Results. Remarkably, by using UVD, ***all*** pre-trained visual representation show significant improvement in OOD sequential generalization, despite their varying IND performances. VIP and LIV, the two representations explicitly trained to learn monotone embedding distances, demonstrate higher *comparative* gains compared to the other representations, despite similar or even lower performances when the representations are not used to decompose subgoals (i.e., GCBC-MLP); this validates our hypothesis that representations capturing visual task progress information are more suited for off-the-shelf subgoal discovery.

Representation	Method	IND success	IND completion	OOD success	OOD completion
VIP (ResNet50) [24]	GCBC	0.736 (0.011)	0.898 (0.006)	0.035 (0.014)	0.236 (0.057)
	GCBC + Ours	0.737 (0.012)	0.903 (0.009)	**0.188 (0.024)**	**0.566 (0.020)**
R3M (ResNet50) [30]	GCBC	0.742 (0.026)	0.856 (0.006)	0.014 (0.007)	0.223 (0.029)
	GCBC + Ours	0.738 (0.024)	**0.879 (0.000)**	**0.084 (0.045)**	**0.427 (0.002)**
LIV (ResNet50) [23]	GCBC	0.608 (0.068)	0.816 (0.046)	0.008 (0.008)	0.116 (0.082)
	GCBC + Ours	**0.649 (0.013)**	**0.868 (0.007)**	**0.066 (0.025)**	**0.496 (0.033)**
CLIP (ResNet50) [20]	GCBC	0.391 (0.017)	0.692 (0.008)	0.005 (0.001)	0.119 (0.017)
	GCBC + Ours	0.394 (0.036)	0.701 (0.012)	**0.073 (0.003)**	**0.403 (0.01)**
DINO-v2 (ViT-large) [32]	GCBC	0.329 (0.025)	0.654 (0.019)	0.012 (0.01)	0.261 (0.213)
	GCBC + Ours	0.322 (0.053)	0.669 (0.037)	**0.055 (0.025)**	**0.446 (0.034)**
VIP (ResNet50) [24]	GCBC-GPT	0.702 (0.029)	0.841 (0.02)	0.039 (0.027)	0.302 (0.028)
	GCBC-GPT + Ours	0.708 (0.056)	**0.897 (0.024)**	**0.213 (0.054)**	**0.600 (0.038)**

TABLE I: **IND and OOD IL Results on FrankaKitchen.** We report the mean and standard deviation of success rate (full-stage completion) and the percentage of the completion (out of 4 stages), evaluated over diverse existing pretrained visual representations trained by GCBC with three seeds. Highlighted scores represent improvements in OOD evaluations and IND results with gains exceeding 0.01.

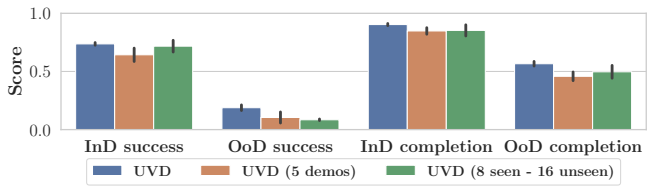

Fig. 3: **Ablations on dataset size and composition.**

Ablations. We present several ablations studying whether UVD remains effective when varying training settings. As VIP stands out as the most promising candidate for UVD-based imitation learning, we perform all ablations using VIP as the backbone representation. First, we ablate the MLP policy architecture with a GPT-like causal transformer policy [36]. As shown in the last row of Table I, this more powerful, history-aware, policy is insufficient to achieve the same level of generalization; UVD again provides sizable generalization improvement.

Beyond policy architecture, we also study the effect of dataset size and diversity. To this end, we consider (1) reducing the training dataset size to 5 demonstrations per training task, and (2) reducing the number of training tasks to 8 but keeping the full number of demonstrations per task. Both IND and OOD performance remains similar, confirming that UVD enables OOD generalization that is robust to the varying sizes and diversity of the training data.

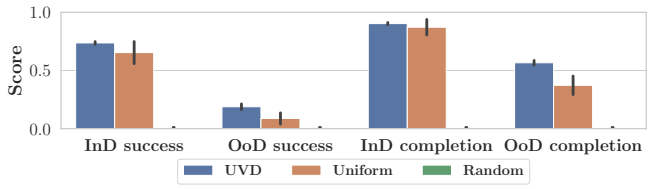

Fig. 4: **Comparison with heuristic goal-labeling methods.**

Finally, we study whether *UVD* is necessary to achieve strong OOD generalization and investigate alternative ways of generating subgoals. We consider **Uniform** and **Random**;

Method	Success	Completion
GCRL-VIP	0.0 / 0.0	0.09 / 0.25
GCRL-VIP + Ours	**0.65 / 1.0**	**0.75 / 1.0**
GCRL-R3M	0.0 / 0.0	0.09 / 0.25
GCRL-R3M + Ours	**0.649 / 1.0**	**0.82 / 1.0**

TABLE II: **RL results on FrankaKitchen**. Full-stage success rate and the percentage of full-stage completion are reported in the format of (average performance with 3 random seeds) / (max performance).

Uniform randomly selects a frame within a fixed size window after the observation; this strategy has been employed in many prior works [14,22]. **Random** randomly selects 3 to 5 frames within the demonstration as subgoal frames. As shown in Fig. 4, the alternatives uniformly hurt performance on all settings and metrics. This is to be expected as these alternatives introduce redundant and less semantically meaningful subgoals; as a result, they may perform comparably IND, but their OOD generalization suffers.

UVD-Guided Reinforcement Learning. We investigate whether UVD can also enhance reinforcement learning by providing goal-based shaped rewards for subtasks (4). Recall that in this setting, only a single video demonstration (without action labels) is given to the agent to specify the learning task. Within the FrankaKitchen environment, we examine a specific task sequence: `open microwave`, `move kettle`, `toggle light switch`, and `slide cabinet`. We select VIP and R3M as candidate representations as they performed best for IL IND evaluations. We consider a goal-conditioned RL baseline, which constructs goal-based rewards by uniformly using the last demonstration frame as goal in (4). We use PPO [37] as the RL algorithm and report the average and the max success rate and percentage of completion over 3 random seeds in Table II; see the supplementary website for more experimental details.

We see baselines fail to make non-trivial task progress with either visual backbone, confirming that goal-based rewards with respect to a distant final goal are not well-shaped to

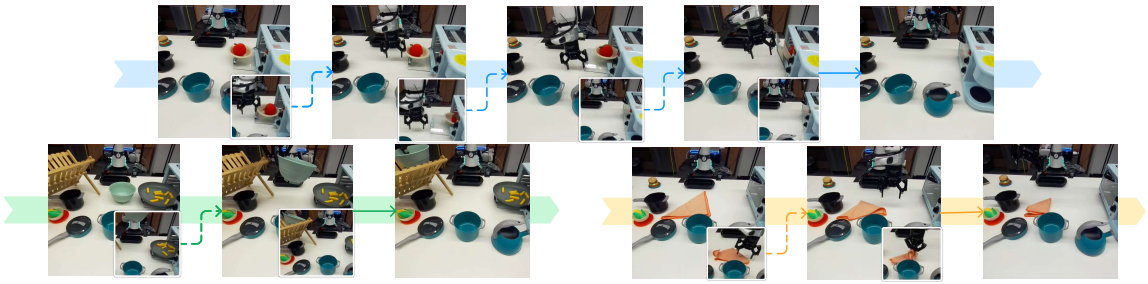

Fig. 5: **Example Sub-Sampled Rollouts on Real-World OOD Tasks.** The initial frame in each sequence is a representative OOD initial observation. The inset image in each frame is the conditioned UVD-discovered goal for that frame.

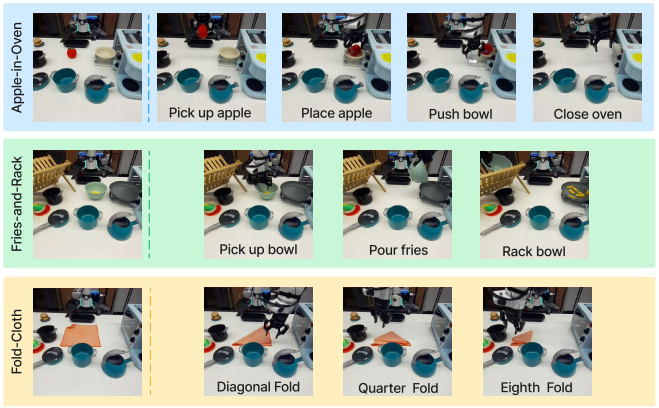

Fig. 6: **Real-World Tasks.** The first picture in each row depicts a representative initial observation, and the following frames are the distinct subtasks.

Task	Method	IND S.	IND C.	OOD S.	OOD C.
Apple-in-Oven	GCBC	0.50	0.438	0.0	0.500
	GCBC + Ours	0.60	**0.750**	**0.25**	**0.625**
Fries-and-Rack	GCBC	0.30	0.567	0.0	0.0
	GCBC + Ours	0.35	**0.750**	**0.25**	**0.500**
Fold-Cloth	GCBC	0.05	0.100	0.0	0.0
	GCBC + Ours	0.15	**0.483**	**0.15**	**0.425**

TABLE III: **IND and OOD Results on Real-World Tasks.** S-success, C-completion.

guide exploration. In contrast, UVD-rewards consistently accelerate RL training and achieve high overall success on the task, validating UVD's utility in not only task generalization but also in task learning.

B. Real-World Experiments

We introduce 3 real-world multi-stage tasks on a real Franka robot. These tasks contain daily household manipulation skills, such as picking, pouring, folding, and manipulating articulated objects. See Fig. 6 for a detailed breakdown of the subtasks in each task. For each task, we have collected 100 demonstrations via teleoperation; for each trajectory, the positions of relevant objects in the scene are randomized within a fixed distribution. The policies are learned via GCBC with MLP architecture as in simulation; see the supplementary website for details.

OOD Evaluation. On our real-world tasks, the subtasks are sequentially dependent and cannot be performed in arbitrary orders. To test compositional generalization, we evaluate whether the policies can *skip* intermediate tasks when their effects in the environment are already achieved. For example, on the `Fries-and-Rack` task, we evaluate on initial states in which the fries are already placed on the plate. In this case, a policy trained with semantically meaningful subgoals should be able to directly proceed from picking up the bowl to racking the bowl. This is because the post-condition of pouring the fries is semantically identical to the pre-condition of racking the bowl – both have the bowl picked up mid air and the fries on the plate. Similarly, on the `Apple-in-Oven` task, we test generalization by having the apple directly placed on the plastic plate, and on the `Fold-Cloth` task, we have the cloth folded diagonally already; see Figure 5 for an illustration of these OOD initial observations. While these OOD tasks are shorter than the training tasks, the exact sequences are still unseen during training and they contain unseen initial state configurations. As before, we test these OOD as well as IND task sequences; for each task sequence, we evaluate on 20 rollouts using the same set of object configurations for every compared method.

Results are presented in Table III. As shown, on all tasks, UVD methods can solve OOD tasks whereas the baseline completely fails, despite their comparable performance on IND tasks. These results corroborate our findings in simulation and make a strong case for the effectiveness of UVD's subgoals and its applicability to challenging real-world tasks.

In Figure 5, we visualize UVD policy rollouts by displaying sub-sampled frames and their conditioned subgoals (the inset frame) on the OOD tasks. In all cases, UVD retrieves meaningful subgoals from the training set and the policy can successfully match the depicted semantic subtask.

VI. Conclusion

We have presented Universal Visual Decomposer, an off-the-shelf task decomposition method for long-horizon visual manipulation tasks using pre-trained visual representations. UVD does not require any task-specific knowledge or training and effectively produces semantically meaning subgoals across both simulated and real-robot environments. UVD-discovered subgoals enable effective reward shaping for solving challenging multi-stage tasks using RL, and policies trained with IL exhibit significantly superior compositional generalization at test time.

REFERENCES

[1] P.-L. Bacon, J. Harb, and D. Precup, "The option-critic architecture," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, no. 1, 2017. 2

[2] A. Bagaria and G. Konidaris, "Option discovery using deep skill chaining," in *International Conference on Learning Representations*, 2019. 2

[3] J. Borja-Diaz, O. Mees, G. Kalweit, L. Hermann, J. Boedecker, and W. Burgard, "Affordance learning from play for sample-efficient policy learning," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6372–6378. 1, 2

[4] E. Chane-Sane, C. Schmid, and I. Laptev, "Goal-conditioned reinforcement learning with imagined subgoals," in *International Conference on Machine Learning*. PMLR, 2021, pp. 1430–1440. 2

[5] J. Co-Reyes, Y. Liu, A. Gupta, B. Eysenbach, P. Abbeel, and S. Levine, "Self-consistent trajectory autoencoder: Hierarchical reinforcement learning with trajectory embeddings," in *International conference on machine learning*. PMLR, 2018, pp. 1009–1018. 2

[6] 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. 1

[7] M. Deitke, W. Han, A. Herrasti, A. Kembhavi, E. Kolve, R. Mottaghi, J. Salvador, D. Schwenk, E. VanderBilt, M. Wallingford, L. Weihs, M. Yatskar, and A. Farhadi, "Robothor: An open simulation-to-real embodied AI platform," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, 2020, pp. 3161–3171. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2020/html/Deitke_RoboTHOR_An_Open_Simulation-to-Real_Embodied_AI_Platform_CVPR_2020_paper.html 4

[8] Y. Ding, C. Florensa, P. Abbeel, and M. Phielipp, "Goal-conditioned imitation learning," *Advances in neural information processing systems*, vol. 32, 2019. 3

[9] B. Eysenbach, R. R. Salakhutdinov, and S. Levine, "Search on the replay buffer: Bridging planning and reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 32, 2019. 2

[10] K. Fang, P. Yin, A. Nair, and S. Levine, "Planning to practice: Efficient online fine-tuning by composing goals in latent space," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 4076–4083. 1, 2

[11] K. Fang, P. Yin, A. Nair, H. R. Walke, G. Yan, and S. Levine, "Generalization with lossy affordances: Leveraging broad offline data for learning visuomotor tasks," in *Conference on Robot Learning*. PMLR, 2023, pp. 106–117. 1, 2

[12] 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. 3

[13] 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. 1

[14] 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. 2, 4, 5

[15] M. Heo, Y. Lee, D. Lee, and J. J. Lim, "Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation," *arXiv preprint arXiv:2305.12821*, 2023. 2

[16] D.-A. Huang, S. Nair, D. Xu, Y. Zhu, A. Garg, L. Fei-Fei, S. Savarese, and J. C. Niebles, "Neural task graphs: Generalizing to unseen tasks from a single video demonstration," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 8565–8574. 1, 2

[17] S. James and A. J. Davison, "Q-attention: Enabling efficient learning for vision-based robotic manipulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1612–1619, 2022. 1, 2

[18] D. Jayaraman, F. Ebert, A. A. Efros, and S. Levine, "Time-agnostic prediction: Predicting predictable video frames," *ICLR*, 2019. 2

[19] A. Khandelwal, L. Weihs, R. Mottaghi, and A. Kembhavi, "Simple but effective: Clip embeddings for embodied ai," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 829–14 838. 1

[20] 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. 5

[21] Y. Lee, E. S. Hu, and J. J. Lim, "Ikea furniture assembly environment for long-horizon complex manipulation tasks," in *2021 ieee international conference on robotics and automation (icra)*. IEEE, 2021, pp. 6343–6349. 2

[22] 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. 5

[23] Y. J. Ma, W. Liang, V. Som, V. Kumar, A. Zhang, O. Bastani, and D. Jayaraman, "Liv: Language-image representations and rewards for robotic control," *arXiv preprint arXiv:2306.00958*, 2023. 1, 3, 4, 5

[24] Y. J. Ma, S. Sodhani, D. Jayaraman, O. Bastani, V. Kumar, and A. Zhang, "Vip: Towards universal visual reward and representation via value-implicit pre-training," *arXiv preprint arXiv:2210.00030*, 2022. 1, 2, 3, 4, 5

[25] A. Majumdar, K. Yadav, S. Arnaud, Y. J. Ma, C. Chen, S. Silwal, A. Jain, V.-P. Berges, P. Abbeel, J. Malik, *et al.*, "Where are we in the search for an artificial visual cortex for embodied intelligence?" *arXiv preprint arXiv:2303.18240*, 2023. 1

[26] A. Mandlekar, D. Xu, R. Martín-Martín, S. Savarese, and L. Fei-Fei, "Learning to generalize across long-horizon tasks from human demonstrations," *arXiv preprint arXiv:2003.06085*, 2020. 1, 2

[27] O. Nachum, S. Gu, H. Lee, and S. Levine, "Near-optimal representation learning for hierarchical reinforcement learning," *arXiv preprint arXiv:1810.01257*, 2018. 2

[28] O. Nachum, S. S. Gu, H. Lee, and S. Levine, "Data-efficient hierarchical reinforcement learning," *Advances in neural information processing systems*, vol. 31, 2018. 2

[29] S. Nair and C. Finn, "Hierarchical foresight: Self-supervised learning of long-horizon tasks via visual subgoal generation," *arXiv preprint arXiv:1909.05829*, 2019. 1, 2

[30] 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. 1, 3, 4, 5

[31] S. Nasiriany, V. Pong, S. Lin, and S. Levine, "Planning with goal-conditioned policies," *Advances in Neural Information Processing Systems*, vol. 32, 2019. 2

[32] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023. 4, 5

[33] S. Parisi, A. Rajeswaran, S. Purushwalkam, and A. Gupta, "The unsurprising effectiveness of pre-trained vision models for control," *arXiv preprint arXiv:2203.03580*, 2022. 1

[34] K. Pertsch, O. Rybkin, F. Ebert, S. Zhou, D. Jayaraman, C. Finn, and S. Levine, "Long-horizon visual planning with goal-conditioned hierarchical predictors," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17 321–17 333, 2020. 1, 2

[35] 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. 4

[36] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019. 5

[37] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017. 5

[38] 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. 2

[39] L. X. Shi, A. Sharma, T. Z. Zhao, and C. Finn, "Waypoint-based imitation learning for robotic manipulation," *arXiv preprint arXiv:2307.14326*, 2023. 1, 2

[40] K. Shiarlis, M. Wulfmeier, S. Salter, S. Whiteson, and I. Posner, "Taco: Learning task decomposition via temporal alignment for control," in *International Conference on Machine Learning*. PMLR, 2018, pp. 4654–4663. 1, 2

[41] M. Shridhar, L. Manuelli, and D. Fox, "Perceiver-actor: A multi-task transformer for robotic manipulation," in *Conference on Robot Learning*. PMLR, 2023, pp. 785–799. 1, 2

[42] R. S. Sutton, D. Precup, and S. Singh, "Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning," *Artificial intelligence*, vol. 112, no. 1-2, pp. 181–211, 1999. 2

[43] L. Weihs, M. Deitke, A. Kembhavi, and R. Mottaghi, "Visual room rearrangement," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, 2021, pp. 5922–5931. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2021/html/Weihs_Visual_Room_Rearrangement_CVPR_2021_paper.html 4

[44] E. Wijmans, A. Kadian, A. Morcos, S. Lee, I. Essa, D. Parikh, M. Savva, and D. Batra, "DD-PPO: learning near-perfect pointgoal navigators from 2.5 billion frames," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. [Online]. Available: https://openreview.net/forum?id=H1gX8C4YPr 4

[45] T. Xiao, I. Radosavovic, T. Darrell, and J. Malik, "Masked visual pre-training for motor control," *arXiv preprint arXiv:2203.06173*, 2022. 1

[46] D. Xu, S. Nair, Y. Zhu, J. Gao, A. Garg, L. Fei-Fei, and S. Savarese, "Neural task programming: Learning to generalize across hierarchical tasks," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 3795–3802. 1, 2

[47] L. Zhang, G. Yang, and B. C. Stadie, "World model as a graph: Learning latent landmarks for planning," in *International Conference on Machine Learning*. PMLR, 2021, pp. 12 611–12 620. 2