Universal Visual Decomposer: Long-Horizon Manipulation Made Easy

Anonymous Author(s)
Affiliation
Address
email

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. We provide videos and experiments results in uvd2023.github.io and Appendix.

1 Method

1.1 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 [1–3] 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 $\tau = (o_{t-n}, ..., o_t)$:

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

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:

$$o_{t-n-1} := \arg\max_{o_h} d_\phi(o_h; o_t) < d_\phi(o_{h+1}; o_t), h < t\,. \tag{2}$$

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 (2) recursively until the full sequence τ is exhausted. see Fig. 1 for pseudocode. In practice, (1) 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 (2) effective; see the supplementary website for details.

Algorithm 1: Universal Visual Decomposer

Init: frozen visual encoder ϕ, $\tau = \{o_0, \cdots, o_T\}$
Init: set of subgoals $\tau_{goal} = \{\}$, $t = T$
while *t not small enough* **do**
    $\tau_{goal} = \tau_{goal} \cup \{o_t\}$
    Find o_{t-n-1} from Eq. 2
    $t = t - n - 1$
end

Figure 1: **Universal Visual Decomposer** Pseudocode

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.

1.2 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) [4, 5].

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:

$$R(o_t, o_{t+1}; \phi, g_i) := d_\phi(o_t; g_i) - d_\phi(o_{t+1}; g_i)\,. \tag{3}$$

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 [6–8, 2] to be particularly effective when deployed with suitable visual representations.

References

[1] 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. 2

[2] 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. 2

[3] 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. 2

[4] Y. Ding, C. Florensa, P. Abbeel, and M. Phielipp, "Goal-conditioned imitation learning," *Advances in neural information processing systems*, vol. 32, 2019. 2

[5] 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. 2

[6] 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 2

[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

[8] 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 2