arXiv:2408.12674v1 [cs.RO] 22 Aug 2024

One-shot Video Imitation via Parameterized Symbolic Abstraction Graphs

Jianren Wang, Kangni Liu, Dingkun Guo, Zhou Xian, Christopher G. Atkeson
Robotics Institute
Carnegie Mellon University United States

Abstract: Learning to manipulate dynamic and deformable objects from a single demonstration video holds great promise in terms of scalability. Previous approaches have predominantly focused on either replaying object relationships or actor trajectories. The former often struggles to generalize across diverse tasks, while the latter suffers from data inefficiency. Moreover, both methodologies encounter challenges in capturing invisible physical attributes, such as forces. In this paper, we propose to interpret video demonstrations through Parameterized Symbolic Abstraction Graphs (PSAG), where nodes represent objects and edges denote relationships between objects. We further ground geometric constraints through simulation to estimate non-geometric, visually imperceptible attributes. The augmented PSAG is then applied in real robot experiments. Our approach has been validated across a range of tasks, such as Cutting Avocado, Cutting Vegetable, Pouring Liquid, Rolling Dough, and Slicing Pizza. We demonstrate successful generalization to novel objects with distinct visual and physical properties. For visualizations of the learned policies please check: https://www.jianrenw.com/PSAG/

Keywords: Learning from Video, Imitation Learning

1 Introduction

Humans can learn to manipulate dynamic and deformable objects by simply watching one *single* demonstration video. It is desired for robots to acquire similar capabilities and learn from the massive amount of videos covering numerous skills available on the internet, which could potentially bring a "ChatGPT moment" to the field of robotics and make a significant step forward in robot learning and autonomy. However, despite considerable progress in artificial intelligence and robotics in recent years, the development of robot systems that can match human-level capabilities of learning from video demonstrations remains elusive. Why is this difficult? We identify three primary challenges in learning from few-shot video demonstrations. First, the extraction of the demonstrator's intentions and actions from videos is complicated by the presence of extraneous information in videos, making it difficult to isolate relevant cues. Second, the task of translating observed intentions into adaptable skills—skills that robots can apply in a wide range of objects and environments, each with its own set of visual and physical characteristics—presents a significant challenge. This difficulty stems from the large variability in visual and physical attributes across different objects and environments. Third, developing efficient learning algorithms that enable robots to acquire skills from a limited number of demonstrations, is also intrinsically a difficult task.

Two primary approaches to learning from demonstration have been explored in the past. One seminal approach developed nearly half a century ago focused on interpreting the demonstrator's intentions as object relationships and replicating those relationships [1]. Subsequent research in this domain focuses on utilizing perceived abstract relationships as objectives for planning techniques, including Task-level Planning [2] and Task and Motion Planning [3]. Notably, these approaches often overlook detailed metrics of objects and their relationships, and in many instances, the precise

8th Conference on Robot Learning (CoRL 2024), Munich, Germany.

motions of actors are de-emphasized and sometimes not even measured. While these methods efficiently eliminate most irrelevant information and are adept at learning from a single demonstration, they often rely on strong human priors, incorporating hardcoded elements, which poses a challenge in addressing the first challenge [4].

Another avenue in learning from demonstration involves direct robot teaching, wherein desired positions and trajectories are demonstrated to a robot through teleoperation [5], motion capture technology [6], or computer vision [7]. The robot then learns to predict actions that align with the demonstration data [8, 9, 10]. These approaches assume that, through training on a diverse dataset, the system will implicitly comprehend the demonstrators' intentions. However, this assumption is not always accurate and can be highly inefficient, leading to a failure in addressing the third challenge.

More importantly, both approaches fail to address the second challenge, as most physical characteristics other than movement are challenging to observe from videos. One purpose of this paper is to make the point that behavior is more than just replaying object relationships or trajectories interpolated from a set of observed trajectories. This is especially true in tasks where the exact values of forces matter, such as in dynamic tasks [11], and tasks involving deformation, separation, and combination of materials.

In contrast to the aforementioned approaches, our proposal involves interpreting video demonstrations using Parameterized Symbolic Abstraction Graphs (PSAG). These graphs consist of nodes and edges as abstractions, which are parameterized by their geometric and non-geometric attributes. In this framework, each node can represent a rigid or deformable object, incorporating attributes that capture the six degrees of freedom (6DOF) of the object pose. The edge information defines relationships between objects, encompassing aspects like contact, and is parameterized according to the contact region and forces acting between them.

To build the PSAG, we begin by utilizing off-the-shelf detectors, depth estimation, and optical flow estimation to evaluate object poses and relationships. Subsequently, we learn to simulate the demonstration with this geometric information (*e.g.* positions, contact points) to calculate forces through this process. This enables us to parameterize the non-geometric imperceptible attributes of the edges (e.g., forces), laying the groundwork for transferring the skill to the real world.

Our proposed method helps the agent disregard irrelevant information and effectively learn from a **single** video. We validate the efficacy of our approach through experiments conducted on five challenging tasks: *Cutting Avocado*, *Cutting Vegetable*, *Pouring Liquid*, *Rolling Dough* and *Slicing Pizza*. Notably, the test environments differ substantially from the learning environments, encompassing variations in geometry, appearance, and physics.

To summarize, our contribution includes:

- Proposing to interpret video demonstrations as parameterized symbolic abstraction graphs (PSAG)
- Proposing to learn simulations from demonstrations with minimal human input, enabling the addition of non-geometric temporally parameterized visually imperceptible attributes to the edges.
- Demonstrating the efficacy of our approach in performing dynamics and deformable manipulation tasks with generalizability.

2 Related Work

Symbolic Visual Reasoning Symbolic methods have shown good data efficiency and generalization capabilities across various computer vision tasks, ranging from visual question answering [12, 13, 14] to image captioning [15, 16, 17]. Recent advancements have extended these methods into 3D object relationship reasoning [18, 19, 20], object physics modeling [21, 22, 23, 24, 25, 26, 27], and video action understanding [28, 29, 30].

Diverging from methods that involve learning to perceive physics or exploiting physical constraints to enhance the comprehension of a given video [21, 31], our research centers on the interpretation of video demonstrations to guide robots in performing tasks involving varying objects or environments. And in contrast to approaches that emphasize reasoning about intervention [32], which predominantly revolves around predicting outcomes, our work focuses on predicting how to achieve the same desired outcome when presented with a new set of objects.

Learning from Human Video A large field of Learning from Demonstration (LfD) [33, 34, 35] focus on learning from expert demonstrations [10, 9, 5]. These approaches often have limitations as they rely on a large number of expert demonstrations and assume a shared observation and action space between the imitator and demonstrator. These constraints significantly restrict the potential for effective learning from videos. Instead of learning from robot demonstrations, an alternative approach involves learning from human demonstrations, which are easy and cost-effective to collect. However, the challenge lies in the absence of ground truth actions. One direct method is to imitate human motion [36, 37, 7] or object motion [11, 4]. Often, this line of work focuses on replicating low-level actions of the demonstration rather than developing more generalizable abstractions. In the pursuit of learning higher-level abstractions related to manipulation, some studies attempt to predict visual affordances, indicating where to interact in an image and providing local information on how to interact [38, 39]. While these approaches can serve as effective initializations for a robotic policy, they are not standalone solutions for task completion. Typically, they are employed in conjunction with online learning, necessitating several hours of deployment-time training and robot data [40]. Importantly, these affordances fall short for complex tasks, such as cutting, where crucial information like force cannot be observed from the video, limiting the applicability to tasks beyond simple pick-and-place tasks. In contrast to previous approaches, our proposed method grounds the object relationships in simulation. This allows the robot to uncover unseen information from videos and thus facilitates learning from a diverse array of tasks.

Deformable Object Manipulation Deformable object manipulation poses a longstanding challenge in robotics. Prior research has primarily concentrated on various tasks such as rope manipulation [41], pouring liquid [42], and cloth manipulation [43]. Additionally, manipulating elastoplastic objects, such as deforming them through grasping [44], rolling [45], or cutting [46], has been explored in other studies. Instead of individually learning each skill through model-free reinforcement learning or model-based planning, our paper focuses on learning a diverse set of skills from single video demonstrations.

3 Methods

In this section, we first outline the process of constructing parameterized symbolic abstraction graphs (PSAG) from videos using off-the-shelf tools (Sec. 3.1). Following that, we describe the process of learning simulations from PSAG and adding non-geometric visually imperceptible attributes to the edges. (Sec. 3.2). We then explain the method for transferring the PSAG to the real world (Sec. 3.3) (Fig.1).

3.1 Building PSAG

To construct parameterized symbolic abstraction graphs from videos, our approach consists of three key steps. First, we employ computer vision techniques to extract depth information, perform instance segmentation, and calculate optical flow, which enable us to reconstruct a semantic point cloud for each frame. Next, we fine-tune the depth estimator by incorporating 3D geometric constraints to achieve temporal consistency in video depth estimation. Finally, we retain only objects of interest and calculate their attributes and relationships with other objects, facilitating the construction of the PSAG. We now introduce each module in detail.

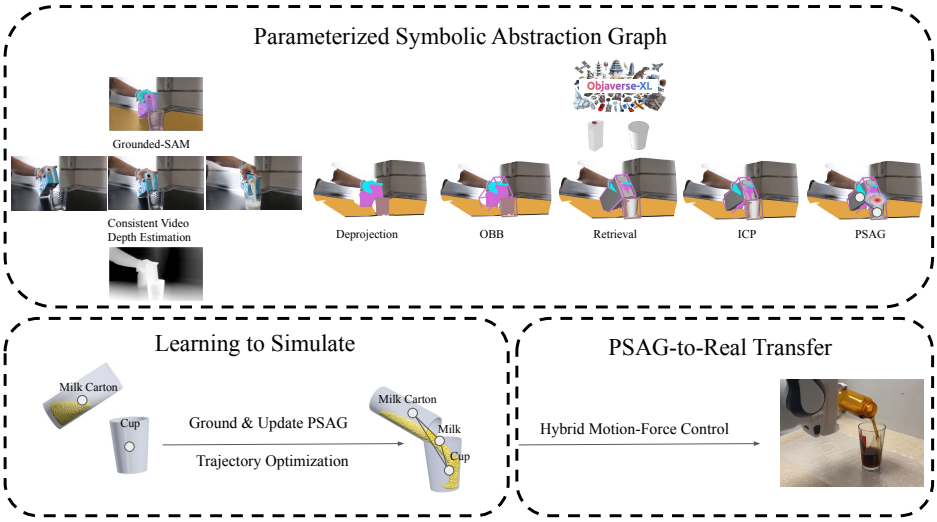

Figure 1: Overview of our pipeline for learning from videos. (a) Building Parameterized Symbolic Abstraction Graphs (PSAG): PSAGs are generated by instance segmentation, consistent video depth estimation, and object relationship calculation. (b) Learning to Simulate: Constructing digital twin to ground geometric constraints via trajectory optimization (c) PSAG-to-Real Transfer using hybrid motion-force control.

Semantic Point Cloud Reconstruction We commence by estimating the monocular depth using ZoeDepth [47], which provides metric-scale depth information for each RGB frame. It is crucial to note that the depth information across adjacent frames lacks coherence, an issue addressed in the subsequent paragraphs. Subsequently, we employ Grounding DINO [48] for object detection and Segment Anything (SAM)[49] for instance segmentation. We leverage GMFlow[50] for optical flow estimation, which facilitates object tracking across frames. This process enables us to construct a semantic point cloud that encapsulates both spatial and semantic details.

Consistent Video Depth Estimation We employ Consistent Video Depth Estimation (CVDE) [51] to produce temporally coherent and geometrically consistent depth maps throughout the entire video. CVDE fine-tunes the pre-trained single-image depth estimation model [47] to minimize geometric inconsistency errors across multiple frames specific to the given video. Following the fine-tuning stage, the final depth estimation results for the video are computed using the fine-tuned model.

PSAG Generation To generate the PSAG, our first step is to filter out irrelevant objects. We specifically identify objects that interact with the hand as objects of interest. This concept of interaction can be hierarchically propagated. For example, objects directly interacting with the hand are classified as the first level of interaction, and objects interacting with the first-level objects are classified as the second level of interaction, and so on. In our current implementation we preserve objects with up to three levels of interaction.

Next, we generate a graph representation to capture attribute changes and relationship dynamics among the objects of interest. Given that 4D reconstruction remains an open problem with no comprehensive solution, we propose a retrieval-based approach for estimating 6DOF object poses. Initially, we compute an oriented bounding box (OBB) around each object using the method from [52]. Next, employing Pointnet++ [53], we retrieve the nearest neighbor from a subset of Objaverse-XL [54]. We then resize and orient the retrieved object to fit the OBB of each corresponding object. Due to potential occlusion, the estimated OBBs may not tightly bound the objects. To address this, we further refine the object poses using iterative closest points (ICP) [55], which are used are node attributes. Additionally, we incorporate edge information to indicate whether two objects are in

contact with each other. This involves calculating the Chamfer distance between their respective point clouds. If the minimum distance falls below a predefined threshold, we determine that the two objects are in contact. Additionally, we capture the closest points of each pair of objects in contact. To enhance the representation, we apply Gaussian filters to smooth the contact region, and these smoothed regions are then utilized for optimization. Through these processes, we construct a PSAG for a given video. In this graph, each node corresponds to an object of interest, and each edge denotes the relationship between them. It's worth noting that the current edge attributes only include geometric information (*e.g.* contact regions). We will now elaborate on how to incorporate non-geometric visually imperceptible attributes into the edges.

3.2 Learning to Simulate

With the provided PSAG, we propose to learn simulations for estimating non-geometric attributes. Considering that most general tasks from videos involve the interaction of both rigid and deformable bodies, we advocate the utilization of the Moving Least Squares Material Point Method (MLS-MPM)[56], as per the approach outlined in[57, 58].

To ground the object relationships over the entire video, we initially decompose each task into multiple subtasks based on changes in these constraints, such as the establishment and breaking of contacts. For each subtask, we utilize the object relationships of the subsequent subtask as optimization goals. We frame this process within a Markov Decision Process (MDP) defined by a set of states $s \in \mathcal{S}$, actions $a \in \mathcal{A}$, and a deterministic, differentiable transition dynamics $s_{t+1} = p(s_t, a_t)$, where t denotes discrete time, and states are composed of different objects $s_t = \{s_t^i\}_{i=1,...,n}$. For any pair of objects (s_t^i, s_t^j), the geometric constraints can either exist (in contact) or not (not in contact), and a cost function is denoted as $C(s_t^i, s_t^j)$. For any reference state (6DOF poses of objects) $\hat{s}_t$, a distance function is denoted as $D(s_t, \hat{s}_t)$. Following Wen *et al.* [4], object poses are expressed in the receptacle's coordinate frame (*e.g.*, milk carton's pose relative to the cup in the pouring task), which allows generalization to new scene configurations regardless of absolute poses. We also adopted the Non-uniform Normalized Object Coordinate Space (NUNOCS) [59] for category-level trajectory projection, which enables generalization to arbitrary, unknown instances. The objective is to determine a trajectory that minimizes the total loss L. Following [60], we use gradient-based trajectory optimization to solve for an open-loop action sequence:

$$\begin{aligned}\underset{a_0,...,a_{T-1}}{\operatorname{argmin}} L(a_0, ..., a_{T-1}) = \underset{a_0,...,a_{T-1}}{\operatorname{argmin}} \lambda_1 \times \sum_{t=1}^{T} \sum_{i,j} C(s_t^i, s_t^j) \\ + \lambda_2 \times \sum_{t=1}^{T} D(s_t, \hat{s}_t) + \lambda_3 \times \sum_{t=1}^{T} E(a_t) \end{aligned} \tag{1}$$

where $s_{t+1} = p(s_t, a_t)$.

$C(s_T^i, s_T^j)$ represents the KL divergence [61] between contacting distributions if there are geometric constraints between (s_t^i, s_t^j); otherwise, it is 0. We have additionally formulated a cost function for scenarios involving the separation of one object into two parts, such as cutting. In this context, the cost is defined as the minimum distance between each pair of separated parts. The action a_t encompasses both the translation velocity and angular velocity of each object, and $E(a_t)$ denotes the energy associated with executing action a_t. Additionally, $\lambda_1, \lambda_2, \lambda_3$ are weighting parameters.

We address Equation 1 by iteratively updating the action sequence with ∇L_{a_t}, where t ranges from 0 to T, employing an Adam optimizer [62] initialized with reference trajectories. The modified trajectories are then utilized to update the geometric attribute of the PSAG. Furthermore, at each timestep, we compute the force f_t and torque τ_t observations, which are incorporated into the edge attributes. This augmented PSAG enables the implementation of a hybrid motion-force controller for real-world applications.

3.3 PSAG-to-Real Transfer

Instead of transferring an end-to-end policy that directly operates in real environments based on visual inputs [63, 64], our approach involves transferring abstract poses and forces. Notably, the PSAG in Sec 3.2 is represented in the receptacle's coordinate frame and Non-uniform Normalized Object Coordinate Space (NUNOCS) to enable category-level generalization. To adapt them to the real world, we must update the PSAG with the current environment, transitioning from the receptacle's coordinate frame and NUNOCS to the real-world frame at the actual scale.

The initial step involves transitioning from the receptacle's frame to the world frame, requiring knowledge of object poses and shapes in the real world. We constructed a multi-camera system as depicted in Fig. 2(f). Utilizing eight RealSense D435 depth cameras calibrated with Multical [65] using an AprilTag [66] board, we deproject the depth and color images into a point cloud. By consolidating information from all eight cameras, we obtain a comprehensive spatial understanding. Following segmentation, we crop the point cloud of each object and fit an object to the position as mentioned in Sec 3.1, providing us with the position and orientation of each object.

The second step is to adopt NUNOCS to capture the geometric variation across all instances, which is not direct. Consider milk pouring as an example. If the commonly selected center-of-mass (CoM) is used as the canonical coordinate frame origin, when aligning a novel object instance to the demonstrated one, it may collide with or float away from the receptacle. To address this, we initialize the trajectory from the processed PSAG, where the origin of the category-level canonical coordinate system is the CoM. Then we redo the process described in Sec 3.2 again with the test-time objects, but this time, everything is represented in the world frame with real size. After this, we obtain the optimal trajectory in the real-world frame at the actual object scale.

Finally, we transfer optimal trajectories from the simulation to a real robot using a hybrid motion-force controller. The transfer mechanism is described by Equation (2):

$$p_t^r = k_1 \times p_t^s + k_2 \times (f_t^r - f_t^s) \tag{2}$$

Here, p_t^r, f_t^r denote the positions and forces of the real robot, while p_t^s, f_t^s represent the positions and forces obtained from simulation. The parameters k_1 and k_2 are compliance parameters. The robot is position-controlled in Cartesian space to achieve both the target position and the desired force. We then employ inverse kinematics to convert Cartesian coordinates to joint space, where the robot is controlled by a Proportional-Derivative (PD) controller.

4 Experiments

In this section, we present experiments conducted in both simulated and real-world environments, followed by results and ablation studies.

4.1 Experimental Setup

Tasks Our experiments were designed to evaluate the performance of our approach on five tasks involving deformable objects: *Cutting Avocado*, *Cutting Vegetable*, *Pouring Liquid*, *Rolling Dough* and *Slicing Pizza*. Deformable objects are ubiquitous in kitchen settings, a critical environment for future robotic applications. We selected these tasks to evaluate the model's generalizability and adaptability, which are essential for its success in real-world scenarios.

Evaluation We devised task-specific metrics for qualitative evaluation over 20 trials for each task. For cutting avocados, success is cleanly cutting around the core without cutting into it. For vegetables, success is making a smooth planar cut without generating excessive force on the cutting board. For pouring liquids, success is pouring Coca-Cola/water/yogurt into the cup without spilling. For rolling dough, success is maintaining contact and flattening play sand/play-dough/dough without separating pieces. For slicing pizza, success is dividing it into two parts without generating excessive force or tearing pizza (Figure 2).

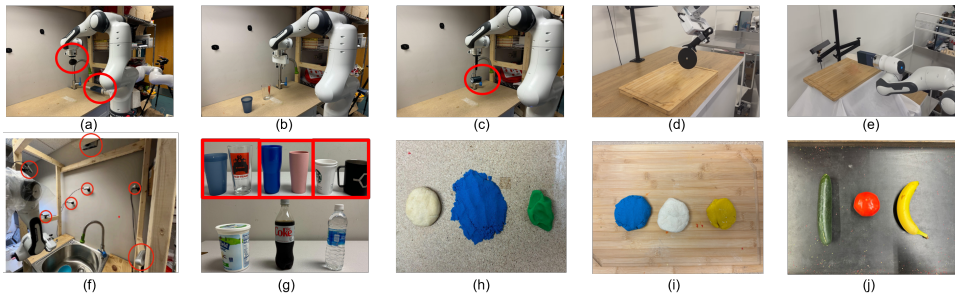

Figure 2: Experiment Settings: (a) Robot arm with an avocado holder and another arm with a knife and force sensor.(b) Robot arm and cups for the pouring task. (c) Rolling pin mounted on the robot arm for rolling dough.(d) Slicer affixed to the robot arm for slicing pizza.(e) Knife mounted on the robot arm for cutting vegetables. (f) Multi-camera system. (g) Cups, yogurt, Coke, and water for the pouring task. (h, i) Dough, play sand, and play dough for the rolling dough and slicing pizza experiments. (j) Cucumber, tomato, and banana for the cutting vegetables task.

Method	Cutting Avocado	Cutting Vegetable	Pouring Liquid	Rolling Dough	Slicing Pizza
YODO	10%	55%	15%	10%	50%
TF	5%	50%	5%	10%	40%
IRL	0%	0%	10%	0%	0%
IW	10%	65%	80%	25%	60%
Ours	75%	75%	80%	70%	70%

Table 1: Quantitative Results: Each row represents the success rate of the five tasks. Rows 1-4 demonstrate the results of You Only Demonstrate Once, Trajectory Following, Inverse Reinforcement Learning, and Interaction Warping, which serve as baseline methods for comparison. Row 5 illustrates that our method consistently outperforms baseline methods by a large margin.

Baselines and Ablations As there is no direct baseline for comparison, we evaluate our method against four variants of existing approaches: You Only Demonstrate Once (YODO)[4], Trajectory Following (TF), Inverse Reinforcement Learning (IRL), and Interaction Warping (IW)[67]. YODO and TF focus on replaying the trajectory observed in the video, while IRL optimizes object relationships without using trajectories. All baselines solely leverage features directly observable in the video. Please refer to the supplementary material for more details.

As shown in Table 1, our method outperforms all baseline methods by a large margin. The three components—geometric constraints, reference trajectories, and force information—each contribute to learning a robust policy. Firstly, comparing YODO and TF with our method, the absence of geometric constraints in these baselines results in no reward mechanism promoting behaviors like rolling or pouring, leading to a lower success rate. Secondly, comparing IRL with our method shows that omitting reference trajectories and only encouraging geometric constraints leads to either aggressive trajectories (e.g., pouring) or getting trapped in a local minimum, preventing the acquisition of crucial rotational behavior necessary for task completion (e.g., cutting avocado), resulting in a low success rate. Thirdly, comparing IW with our method, IW's results are heavily task-specific. For tasks where force is crucial (e.g., cutting avocado and rolling dough), our method performs significantly better than the baseline methods. This demonstrates that merely transferring observable information from the video to the robot program is insufficient.

We also present the human demonstrations (from YouTube) and the sampled robot trajectories in Fig. 3. Please refer to the supplementary video for more qualitative results.

5 Limitations and Conclusions

Despite enabling one-shot video imitation across a diverse range of tasks, our method faces three major limitations. Firstly, detection, segmentation, and tracking errors present significant challenges.

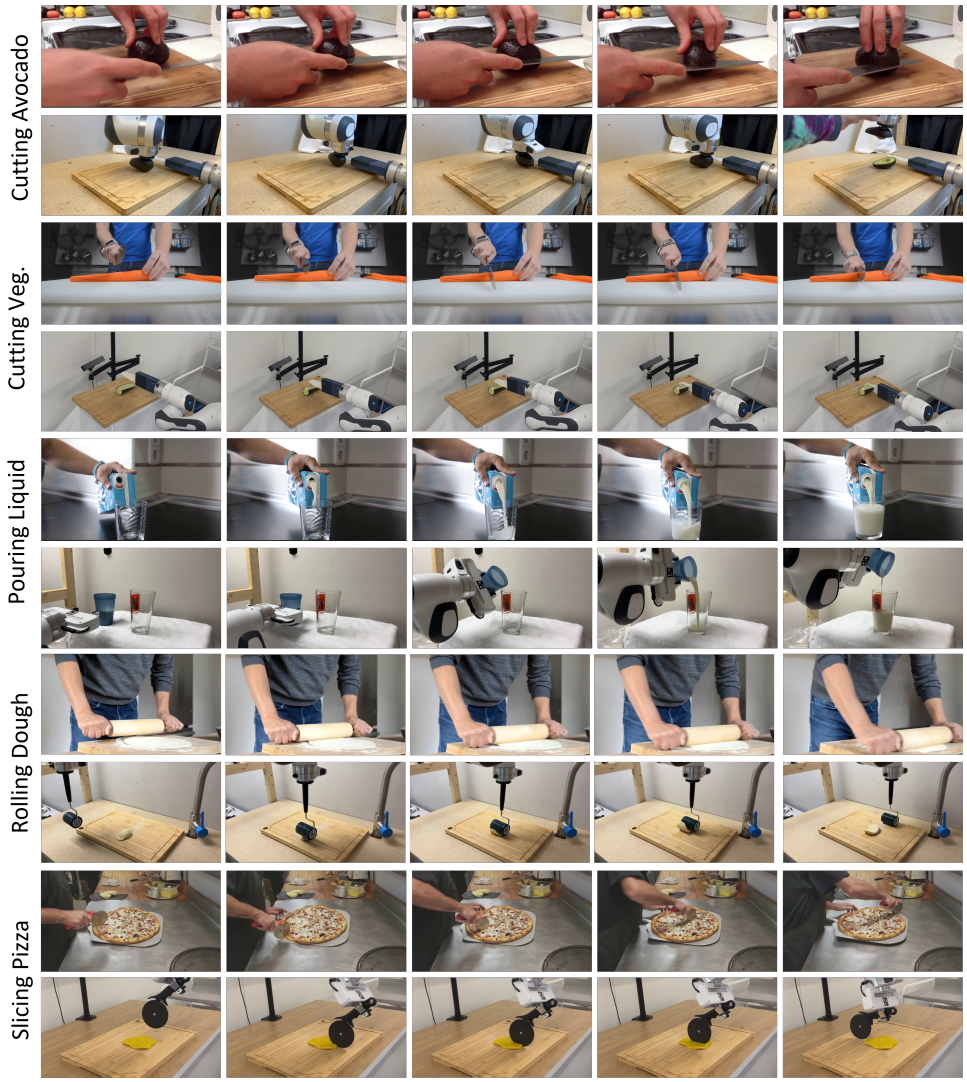

Figure 3: For each task, we present the video demonstration (top) and the robot trajectories (bottom). Our proposed method allows the robot to perform challenging tasks such as cutting an avocado, cutting vegetables, pouring liquid, rolling dough, and slicing pizza from a single demonstration.

While existing works often assume precise results, achieving this in real-world scenarios is complex and requires meticulous hyperparameter tuning. Secondly, hyperparameter tuning during learning remains a challenge. Similar to many trajectory optimization or reinforcement learning methods, the weights of each cost term must be carefully adjusted based on the observed policy performance. Lastly, tuning simulation properties, such as the density of the simulation grid or material viscosity, also poses hurdles. Although a single set of parameters can often be applied to multiple tasks, achieving autonomous generalization remains difficult.

In conclusion, our work highlights that behavior extends beyond merely replicating object relationships or actor trajectories. We propose interpreting video demonstrations as Parameterized Symbolic Abstraction Graphs (PSAG), where nodes represent objects and edges signify relationships. By grounding geometric relationships in simulation and incorporating non-geometric, visually imperceptible attributes such as forces, our method effectively learns to manipulate diverse dynamics and deformable objects from a single video demonstration. This approach points towards reducing the reliance on teleoperated robot demonstration data and emphasizes learning from single human demonstrations.

References

[1] P. H. Winston. Learning structural descriptions from examples. 1970.

[2] T. Lozano-Pérez, J. L. Jones, E. Mazer, and P. A. O'Donnell. Task-level planning of pick-and-place robot motions. *Computer*, 22(3):21–29, 1989.

[3] C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez. Integrated task and motion planning. *Annual review of control, robotics, and autonomous systems*, 4:265–293, 2021.

[4] B. Wen, W. Lian, K. Bekris, and S. Schaal. You only demonstrate once: Category-level manipulation from single visual demonstration. In *Robotics: Science and Systems (RSS)*, 2022.

[5] Z. Fu, T. Z. Zhao, and C. Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. In *arXiv*, 2024.

[6] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black. AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, pages 5442–5451, Oct. 2019.

[7] S. Bahl, A. Gupta, and D. Pathak. Human-to-robot imitation in the wild. *RSS*, 2022.

[8] S. Ross, G. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635, 2011.

[9] J. Wang, S. Dasari, M. K. Srirama, S. Tulsiani, and A. Gupta. Manipulate by seeing: Creating manipulation controllers from pre-trained representations. *ICCV*, 2023.

[10] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[11] C. G. Atkeson and S. Schaal. Robot learning from demonstration. In *ICML*, volume 97, pages 12–20, 1997.

[12] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Learning to compose neural networks for question answering. In *Proceedings of NAACL-HLT*, pages 1545–1554, 2016.

[13] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Neural module networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 39–48, 2016.

[14] D. Hudson and C. D. Manning. Learning by abstraction: The neural state machine. *Advances in Neural Information Processing Systems*, 32, 2019.

[15] P. Anderson, B. Fernando, M. Johnson, and S. Gould. Spice: Semantic propositional image caption evaluation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*, pages 382–398. Springer, 2016.

[16] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, and L. Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3668–3678, 2015.

[17] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73, 2017.

[18] J. Hsu, J. Mao, and J. Wu. Ns3d: Neuro-symbolic grounding of 3d objects and relations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2614–2623, 2023.

[19] I. Armeni, Z.-Y. He, J. Gwak, A. R. Zamir, M. Fischer, J. Malik, and S. Savarese. 3d scene graph: A structure for unified semantics, 3d space, and camera. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5664–5673, 2019.

[20] U.-H. Kim, J.-M. Park, T.-J. Song, and J.-H. Kim. 3-d scene graph: A sparse and semantic representation of physical environments for intelligent agents. *IEEE transactions on cybernetics*, 50(12):4921–4933, 2019.

[21] A. Gupta, A. A. Efros, and M. Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*, pages 482–496. Springer, 2010.

[22] Z. Chen, J. Mao, J. Wu, K.-Y. K. Wong, J. B. Tenenbaum, and C. Gan. Grounding physical concepts of objects and events through dynamic visual reasoning. *arXiv preprint arXiv:2103.16564*, 2021.

[23] T. Ates, M. S. Atesoglu, C. Yigit, I. Kesen, M. Kobas, E. Erdem, A. Erdem, T. Goksun, and D. Yuret. Craft: A benchmark for causal reasoning about forces and interactions. *arXiv preprint arXiv:2012.04293*, 2020.

[24] A. Melnik, R. Schiewer, M. Lange, A. Muresanu, M. Saeidi, A. Garg, and H. Ritter. Benchmarks for physical reasoning ai. *arXiv preprint arXiv:2312.10728*, 2023.

[25] Z. Chen, K. Yi, Y. Li, M. Ding, A. Torralba, J. B. Tenenbaum, and C. Gan. Comphy: Compositional physical reasoning of objects and events from videos. *arXiv preprint arXiv:2205.01089*, 2022.

[26] M. Ding, Z. Chen, T. Du, P. Luo, J. Tenenbaum, and C. Gan. Dynamic visual reasoning by learning differentiable physics models from video and language. *Advances in Neural Information Processing Systems*, 34:887–899, 2021.

[27] J. Wu, E. Lu, P. Kohli, B. Freeman, and J. Tenenbaum. Learning to see physics via visual de-animation. *Advances in Neural Information Processing Systems*, 30, 2017.

[28] E. Mavroudi, B. B. Haro, and R. Vidal. Representation learning on visual-symbolic graphs for video understanding. In *European Conference on Computer Vision*, pages 71–90. Springer, 2020.

[29] N. Rai, H. Chen, J. Ji, R. Desai, K. Kozuka, S. Ishizaka, E. Adeli, and J. C. Niebles. Home action genome: Cooperative compositional action understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11184–11193, 2021.

[30] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei. Visual relationship detection with language priors. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 852–869. Springer, 2016.

[31] J. Z. Zhang, S. Yang, G. Yang, A. L. Bishop, S. Gurumurthy, D. Ramanan, and Z. Manchester. Slomo: A general system for legged robot motion imitation from casual videos. *IEEE Robotics and Automation Letters*, 2023.

[32] Y. Li, K. Li, P. Wang, D. Wei, H. Pfister, and C.-Y. Chan. Intervention-based recurrent casual model for non-stationary video causal discovery. 2021.

[33] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

[34] A. Billard, S. Calinon, R. Dillmann, and S. Schaal. Survey: Robot programming by demonstration. *Handbook of robotics*, 59(BOOK_CHAP), 2008.

[35] S. Schaal. Is imitation learning the route to humanoid robots? *Trends in cognitive sciences*, 3 (6):233–242, 1999.

[36] H. Bharadhwaj, A. Gupta, S. Tulsiani, and V. Kumar. Zero-shot robot manipulation from passive human videos. In *ICRA2023 Workshop on Pretraining for Robotics (PT4R)*, 2023.

[37] C. Wang, L. Fan, J. Sun, R. Zhang, L. Fei-Fei, D. Xu, Y. Zhu, and A. Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. *CoRL 2023*, 2023.

[38] M. Goyal, S. Modi, R. Goyal, and S. Gupta. Human hands as probes for interactive object understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3293–3303, 2022.

[39] S. Liu, S. Tripathi, S. Majumdar, and X. Wang. Joint hand motion and interaction hotspots prediction from egocentric videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3282–3292, 2022.

[40] S. Bahl, R. Mendonca, L. Chen, U. Jain, and D. Pathak. Affordances from human videos as a versatile representation for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13778–13790, 2023.

[41] A. Nair, D. Chen, P. Agrawal, P. Isola, P. Abbeel, J. Malik, and S. Levine. Combining self-supervised learning and imitation for vision-based rope manipulation. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 2146–2153. IEEE, 2017.

[42] Y. Li, S. Li, V. Sitzmann, P. Agrawal, and A. Torralba. 3d neural scene representations for visuomotor control. In *Conference on Robot Learning*, pages 112–123. PMLR, 2022.

[43] H. Ha and S. Song. Flingbot: The unreasonable effectiveness of dynamic manipulation for cloth unfolding. In *Conference on Robot Learning*, pages 24–33. PMLR, 2022.

[44] H. Shi, H. Xu, Z. Huang, Y. Li, and J. Wu. Robocraft: Learning to see, simulate, and shape elasto-plastic objects in 3d with graph networks. *The International Journal of Robotics Research*, 43(4):533–549, 2024.

[45] X. Lin, C. Qi, Y. Zhang, Z. Huang, K. Fragkiadaki, Y. Li, C. Gan, and D. Held. Planning with spatial-temporal abstraction from point clouds for deformable object manipulation. In *Conference on Robot Learning*, pages 1640–1651. PMLR, 2023.

[46] E. Heiden, M. Macklin, Y. Narang, D. Fox, A. Garg, and F. Ramos. Disect: A differentiable simulation engine for autonomous robotic cutting. *Robotics: Science and Systems XVII*, 2021.

[47] S. F. Bhat, R. Birkl, D. Wofk, P. Wonka, and M. Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023.

[48] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

[49] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollar, and R. Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, October 2023.

[50] H. Xu, J. Zhang, J. Cai, H. Rezatofighi, and D. Tao. Gmflow: Learning optical flow via global matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8121–8130, 2022.

[51] X. Luo, J.-B. Huang, R. Szeliski, K. Matzen, and J. Kopf. Consistent video depth estimation. *ACM Transactions on Graphics (ToG)*, 39(4):71–1, 2020.

[52] J. O'Rourke. Finding minimal enclosing boxes. *International journal of computer & information sciences*, 14:183–199, 1985.

[53] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[54] M. Deitke, R. Liu, M. Wallingford, H. Ngo, O. Michel, A. Kusupati, A. Fan, C. Laforte, V. Voleti, S. Y. Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *arXiv preprint arXiv:2307.05663*, 2023.

[55] M. BeslPJ. A method for registration of 3d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239, 1992.

[56] Y. Hu, Y. Fang, Z. Ge, Z. Qu, Y. Zhu, A. Pradhana, and C. Jiang. A moving least squares material point method with displacement discontinuity and two-way rigid body coupling. *ACM Transactions on Graphics*, 37(4):150, 2018.

[57] Z. Xian, B. Zhu, Z. Xu, H.-Y. Tung, A. Torralba, K. Fragkiadaki, and C. Gan. Fluidlab: A differentiable environment for benchmarking complex fluid manipulation. In *International Conference on Learning Representations*, 2023.

[58] Z. Xu, Z. Xian, X. Lin, C. Chi, Z. Huang, C. Gan, and S. Song. Roboninja: Learning an adaptive cutting policy for multi-material objects. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[59] B. Wen, W. Lian, K. Bekris, and S. Schaal. Catgrasp: Learning category-level task-relevant grasping in clutter from simulation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6401–6408. IEEE, 2022.

[60] X. Lin, Z. Huang, Y. Li, J. B. Tenenbaum, D. Held, and C. Gan. Diffskill: Skill abstraction from differentiable physics for deformable object manipulations with tools. 2022.

[61] S. Kullback and R. A. Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[62] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[63] Z. Zhuang, Z. Fu, J. Wang, C. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao. Robot parkour learning. *CoRL*, 2023.

[64] T. Chen, M. Tippur, S. Wu, V. Kumar, E. Adelson, and P. Agrawal. Visual dexterity: In-hand reorientation of novel and complex object shapes. *Science Robotics*, 8(84):eadc9244, 2023. doi:10.1126/scirobotics.adc9244. URL https://www.science.org/doi/abs/10.1126/scirobotics.adc9244.

[65] O. Batchelor. Multi-camera calibration using one or more calibration patterns. *https://github.com/oliver-batchelor/multical.2.4.1*, 2023.

[66] E. Olson. Apriltag: A robust and flexible visual fiducial system. 2011.

[67] O. Biza, S. Thompson, K. R. Pagidi, A. Kumar, E. van der Pol, R. Walters, T. Kipf, J.-W. van de Meent, L. L. Wong, and R. Platt. One-shot imitation learning via interaction warping. In *7th Annual Conference on Robot Learning*, 2023.

[68] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. Aparicio, and K. Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. *Robotics: Science and Systems XIII*, 2017.

[69] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[70] Y. Lin and A. Wang. Polymetis: a real-time pytorch controller manager. In *https://github.com/facebookresearch/polymetis*, 2021.

A Method Details

In this section, we present more details of our method.

A.1 Grasping

The final detail to discuss is grasping. For cutting avocado, the knife is attached to the X-arm robot, while the avocado is held by the end-effector of a Franka robot. For cutting vegetables, the knife is attached directly to the end-effector of a Franka robot. In the task of slicing pizza, the slicer is mounted on the end-effector of a Franka robot. Similarly, for rolling dough, the roller is directly attached to the robot using a customized tool. For the pouring task, we used a heuristic grasping planner, similar to those used in many one-shot imitation learning studies [4]. We sampled 100 collinear antipodal grasps around the middle of the cup and selected the one with point patches having the highest agreeable normals to the object surface, which proved effective for our purposes. This grasp planning approach could also be replaced with a more advanced grasping planner [68].

B Experiment Details

In this section, we present more details of our experiment.

B.1 Hardware Setup and Control Stack

Our real-world experiments utilize a Franka Panda robot arm with all state data logged at 50 Hz. The robot's action space is a 6 DoF homogeneous transformation, transitioning from the previous end-effector pose to the new one. The new joint positions are calculated using inverse kinematics via Mujoco [69]. These joint positions are subsequently sent to Facebook Polymetis [70] to control the Franka robot.

B.2 Baselines

As there is no direct baseline for comparison, we evaluate our method against four variants of existing approaches: You Only Demonstrate Once (YODO)[4], Trajectory Following (TF), Inverse Reinforcement Learning (IRL), and Interaction Warping (IW)[67]. To make the comparisons fair, we use the same video processing pipeline for all baseline methods.

- You Only Demonstrate Once (YODO). We compared our approach with an adapted version of the You Only Demonstrate Once [4], which combines a collision-aware path planning algorithm with a keypose-based last-inch manipulation algorithm. However, instead of using their pretrained NUNOCS Net, which only works for batteries and gears, we employed the same video interpretation system as proposed in our method.
- Trajectory Following (TF). We implemented a trajectory-following baseline by eliminating the influence of geometric constraints by setting λ_1 in Eq. 1 to zero.
- Inverse Reinforcement Learning (IRL). We implemented an inverse reinforcement learning algorithm with sparse rewards by nullifying the influence of reference trajectories, setting λ_2 in Eq. 1 to zero.
- Interaction Warping (IW). We devised a baseline approach solely leveraging features directly observable in the video. This was achieved by disregarding force information, setting k_2 in Eq. 2 to zero. Consequently, our method loses the ability to transfer any unobservable information from the video to the robot program, akin to the concept of interaction warping proposed by Biza et al. [67]