From Temporal-evolving to Spatial-fixing: A Keypoints-based Learning Paradigm for Visual Robotic Manipulation

Kevin Riou, Kaiwen Dong, Kévin Subrin, Yanjing Sun, Patrick Le Callet

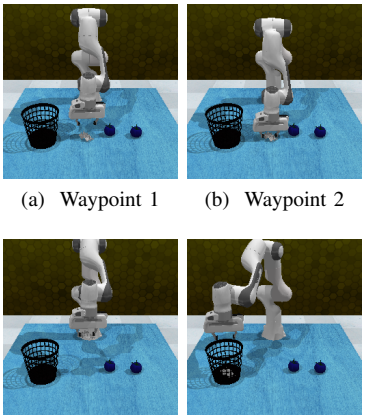
From Temporal-evolving to Spatial-fixing: A Keypoints-based Learning Paradigm for Visual Robotic Manipulation

Kevin Riou[1§], Kaiwen Dong[2§], Kevin Subrin[1], Yanjing Sun[2], and Patrick Le Callet[1]

***Abstract*— The current learning pipelines for robotics manipulation infer movement primitives sequentially along the temporal-evolving axis, which can result in an accumulation of prediction errors and subsequently cause the visual observations to fall out of the training distribution. This paper proposes a novel hierarchical behavior cloning approach which tries to dissociate standard behaviour cloning (BC) pipeline to two stages. The intuition of this approach is to eliminate accumulation errors using a fixed spatial representation. At first stage, a high-level planner will be employed to translate the initial observation of the scene into task-specific spatial waypoints. Then, a low-level robotic path planner takes over the task of guiding the robot by executing a set of pre-defined elementary movements or actions known as primitives, with the goal of reaching the previously predicted waypoints. Our hierarchical keypoints-based paradigm aims to simplify existing temporal-evolving approach to a more simple way: directly spatialize the whole sequential primitives as a set of 8D waypoints only from the very first observation. Plentiful experiments demonstrate that our paradigm can achieve comparable results with Reinforcement Learning (RL) and outperforms existing offline BC approaches, with only a single-shot inference from the initial observation. Code and models are available at : `https://github.com/KevinRiou22/spatial-fixing-il`**

I. INTRODUCTION

Robotic Manipulation Learning, i.e., the ability for a robot to learn various manipulation skills from large-scale human-labeled datasets or interaction experiences, is a crucial approach for an autonomous system. Two paradigms are well-studied for developing this approach: reinforcement learning (RL) and behavior cloning (BC). Although impressive progress has been achieved in endowing robot to learn skills in visual rich scenarios, reinforcement learning is notorious for difficult training and time-consuming. This is because the requirement of huge amount of environment interactions and the challenges of reward function definition, which is impractical and even infeasible in some cases. Behavior cloning can effectively alleviate this issue in an offline training manner, but it still encounters the problem of accumulated error arising from sequential prediction. With this temporal-evolving setting, actions need to be inferred after each observation (up to 376 actions taken for a simple cube lifting action in RLBench[6]), which is inefficient, especially for edge-devices. Additionally, the worsening accumulated error can lead to out-of-distribution observations, which exacerbates the accumulated error in a vicious cycle.



(a) Waypoint 1 (b) Waypoint 2

(c) Waypoint 3 (d) Waypoint 4

Fig. 1: From (a) to (d): crucial waypoints for task of "put rubbish in bin".

To alleviate aforementioned issues, we propose a hierarchical keypoints-based learning paradigm which simplifies the movement primitives as a set of waypoints by encoding the fixed spatial configuration of the entire scene. Our paradigm is made of two stages: (1) A high-level trajectory planner that predicts the minimal sequence of key waypoints for executing the task from multi-view RGB images, given only the first observation of the scene. (2) An off-the-shelf robotic path planner, that can reach the waypoints proposed by the trajectory planner. Our paradigm benefits from a fixed spatial encoding from the initial observation, which eliminates the possibility of error accumulation and enables single-shot inference. Last but not least, it should be noted that conventional BC paradigms used for vision-based robotic manipulation learning demand that the demonstrations be gathered from the robot's viewpoint in order to avoid observation distributional shifts between training and inference. To obtain such demonstrations, typical methods involve teleoperating the robot [15] or manually moving the robot [8], which can be both impractical and time-consuming. In contrast, our approach allows for the manipulating agent to shift between the collection of demonstrations and the actual inference time, as long as the agent is not visible in the initial scene observation. Our contributions can be summarized as following the points:

- A high-level trajectory planner is proposed that encodes

[1]Nantes Université, Ecole Centrale Nantes, CNRS, LS2N, UMR 6004, Nantes, France. {kevin.riou, kevin.subrin, patrick.lecallet}@univ-nantes.fr

[2]School of Information and Control Engineering, China University of Mining and Technology, Xuzhou, 221116, China. {dongkaiwen, yjsun}@cumt.edu.cn

[§]**Equal contribution**

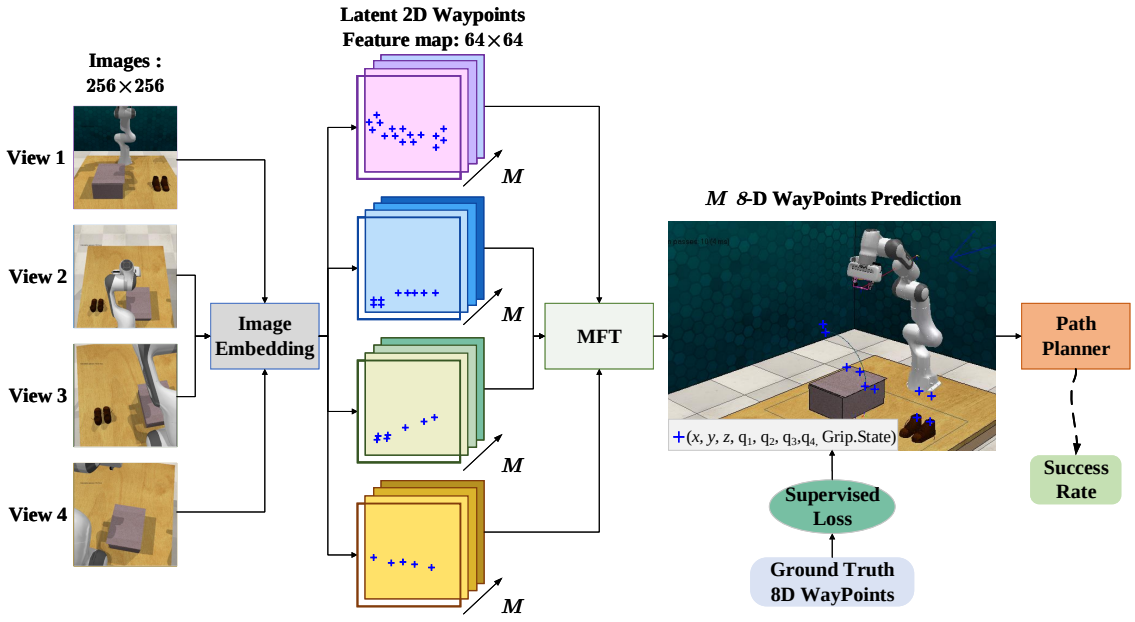


Fig. 2: Overview of our keypoints-based manipulation learning paradigm: The input is 4 independent images from different views with resolution of 256×256. An image embedding module is built to extract 2d waypoints representation in a latent space. Consequently, the latent representation will be sent into our proposed Multi-view Fusion Transformer (MFT) module to obtain the 8D waypoints prediction in 3D space, which can be used as the reference for following path planner. The whole framework will be trained in an end-to-end way. And the the success rate will be evacuated under RLbench simulation.

spatial configuration and high-level plan strategy. Consequently, low-level planning is delegated to established reliable robotics solutions.

- Under the hypothesis of fixed spatial configuration, we predict the motion primitives for the entire execution sequence directly based on the initial observation of the scene. This eliminates the accumulating prediction errors along the task execution, since all actions are predicted at once rather than in a temporal-evolving way.
- The research includes an adapted manipulation dataset that is specifically designed for predicting waypoint sequences, along with tools for assessing the trained model's efficacy. These resources are built upon the RLBench simulation.

II. RELATED WORKS

In this section, we will revise primary knowledge from three aspects of Visual Robotic Manipulation: Reinforcement Learning (RL), behaviour cloning (BC) and keypoint-based manipulation learning.

A. Reinforcement Learning (RL) for manipulation tasks.

Online Reinforcement Learning highly relies on iterative interactions between agent and environment, which is impractical in real-life scenarios due to expensive or dangerous experience collection. The success of Deep Neural Network (DNNs) has promoted the advent of offline Reinforcement Learning for large-scale data representation. Kumar *et al.* [12] proposed a practical algorithm to reduce error accumulation when training from offline data. Ebert *et al.* [3] presented a model-based reinforcement learning method centered around prediction of raw sensory observation by taking use of prediction in the context of robotic manipulation. However, both online and offline approaches focus on learning a direct mapping from environment states to robot actions, which means (1) they are not purpose-aware and (2) they have to learn primitive skills that roboticists can easily solve using off-the-shelf solutions. Primitive based Reinforcement learning is an augmentation for standard Reinforcement Learning with a predefined library of behavior primitives learnt from data, which is more robust and reusable for alleviating the generalization challenge. Strudel *et al.* [18] designed a sample efficient pipeline to learn robust RL policies confined with primitive skills. Nasiriany *et al.*[16] introduced a Deep Reinforcement Learning (DRL) framework which utilized predefined hierarchical primitive skills, to narrow the exploration space of DRL. Dalal *et al.* [1] manually coded primitives with arguments that are learned by RL policy, leading to a better adaptability for the considered tasks.

B. Behavior cloning for manipulation tasks learning.

Benefit from the promising development of Deep Neural Network (DNN) over the past decade, the effectiveness of data-driven learning methods have been proved over many visual sensory applications[20]. Behavior cloning is a straightforward imitation learning method that utilizes the representation capability of DNNs to map the expert's actions to the agent's observations using abundant data. It does not require the agent to interact with the environment during

training, making it a simple approach. However, behavior cloning suffers from several limitations, such as not being purpose-aware, having to learn primitive skills that can be easily solved using off-the-shelf solutions, and well-known error accumulation during task execution. To overcome these issues, researchers have developed an alternative approach called Hierarchical Behavior cloning (HBC), which involves a high-level model that learns intermediate goals and a low-level model that predicts sequences of actions to reach sub-goals, as demonstrated in studies such as [24], [2], [23]. HBC uses an end-to-end framework that significantly reduces the error accumulation and enhances purpose-awareness. Nevertheless, HBC still needs to learn some primitive skills that roboticists can easily solve using off-the-shelf solutions.

C. Keypoint-based manipulation learning

As a task-relevant context on object, keypoints of object surface come with a significant importance for Visual Robotic Manipulation (VRM). With the progressive development of object keypoint discovery approaches [20], [14], [11], [21], many manipulation learning methods have been proposed to leverage keypoints-based context for geometry and semantic representations. But most of these methods leverage keypoints as a bridge for facilitating the transfer of manipulation skills between demonstrators and imitators. Gao *et al.*[4] decomposition robotic manipulation task into keypoints constrain representation and control policies selection, which allows the successful reproductions across various tasks. Yang *et al.* [25] utilized keypoints detector to represent the similarity between a human demonstration and robot execution, then maximized it by Bayesian optimization. Wada *et al.* [22] represent 6D pose of objects in voxels to improve multi-object reasoning in cluttered scenes. Kulkarni *et al.* [11] proposed a unsupervised keypoints discovery network, then apply the learned object keypoints as state input that related to policy explorations over Reinforcement Learning (RL) settings.

III. Background

A. Behavior Cloning for Visual Robotic Manipulation

Behavior Cloning (BC) refers to a supervised learning approaches used to learn sensorimotor policies from offline data. BC only requires pairs of sensory observations, e.g., images, associated to expert actions. Given access to an expert agent, we can build a BC dataset, $D = \{(o_i, a_i)\}_{i=1}^{N}$, where o_i are sensory observations, and $a_i = \pi^*(o_i)$ are the respective actions taken by the expert π^*. When using BC for robotic manipulation, actions are usually described as the Cartesian-position of next gripper Tool Center Point (TCP) and Quaternion-orientation, in the scene, also associated with the gripper state, e.g. opening amount, $a_i = (x_{tcp}, y_{tcp}, z_{tcp}, q_{1_{tcp}}, q_{2_{tcp}}, q_{3_{tcp}}, q_{4_{tcp}}, Grip.State)$. Typical solutions to acquire such expert actions consist of robot teleoperating by human, robot behavior hard-coding with omniscient knowledge in simulation, expert RL policy training using hard-coded reward function as well as huge amount of simulation trials.

When considering visual robotic manipulation, the observations are limited to raw images of the environment $o_i = \{I_i^v\}_{v=1}^{V}$, where I_i^v represents the image from view v in a multi-view configuration. The goal of a BC algorithm is to learn a policy π, parameterised by θ, that produce similar actions to the expert π^* when provided with the same observation o_i. Optimal parameters θ^* are found by minimizing the BC loss 1 :

$$\theta^* = \arg\min_{\theta} \sum_{i} l\left(\pi\left(o_i; \theta\right), a_i\right)$$

At inference phase, the trained policy is used to sequentially predict actions from observations to execute the target task.

B. Limitations

BC is used to solve a sequential decision problem, where future observations depend on previous actions. At training phase, it violates the i.i.d. assumption made in statistical learning. Moreover, at inference phase, errors of the action predictions accumulate along the task execution, which leads to a distributional shift between training and inference observations, which subsequently results in out-of-distribution prediction.

To alleviate this problem, hierarchical algorithms has been proposed to decompose trajectories of manipulation task into sub-trajectories that mostly consisted in existing off-the-shelf robotics primitives, e.g. "reaching" and "grasping". However, both traditional BC and RL still have to learn by themselves to accomplish the target tasks.

IV. Approach

Instead of considering (observation, action) tuples to train our model, we format the dataset as a set of trajectories, $D = \{T_i\}_{i=1}^{N}$, in which each trajectory can be represented as $T = \{(o_t, a_t)\}_{t=1}^{T}$.

We can then further decompose a task as sequence, $T' = \{(o_m, wp_m)\}_{m=1}^{M}, M << T$ made of a few waypoints wp_m that can be sequentially reached using off-the-shelf robotics primitives. Fig. 1 illustrates the observation associated to 4 waypoints which define a "cube lifting" task.

Given these statements, we designed a model that can directly predict such set of waypoints from the first observation of the scene. In this scenario, the training dataset D can be formatted as a set of tasks $T_i = \left(o_0^i, \{wp_0^i, ..., wp_M^i\}\right)$, where the few waypoints needed to accomplish the task must be directly predicted from o_0. This approach benefits from several advantages. First, the sequential decision making problem solved by traditional BC or RL approaches is changed to a one-step prediction problem, where the whole trajectory is predicted from initial observation of the scene, which eliminates both accumulation error during inference and the violation of the i.i.d. assumption during the training phase. Secondly, the trained model focuses on understanding the high level structure, i.e. the purpose of the task, while sub-trajectories prediction is remained for existing and more robust robotics primitives.

This section details the framework proposed to solve the one-step trajectory prediction problem mentioned above.

A. Overall framework description

Fig.2 describes the whole proposed framework. The framework is made of two main components: (1)The deep learning model, predicting waypoints from a multi-view images of the initial observation, denoted as Multi-View Fusion Transformer (MFT) in Fig.2; (2)A path planner, that can drive the robot to the predicted waypoints.

The waypoints prediction framework can be divided in two parts: (1)The image processing part aims to inherently learn to locate the waypoints in the 2D images; (2)The 2D-to-8D projection part that learns to map the muli-view 2D information extracted by the the image processing part to actual 8D robot poses $(x_{tcp}, y_{tcp}, z_{tcp}, q_{1_{tcp}}, q_{2_{tcp}}, q_{3_{tcp}}, q_{4_{tcp}}, Grip.State)$ that can be understood by the robotics path planner. The whole framework is trained end-to-end from ground truth 8D waypoints recovered from expert demonstrations. The method used to choose and extract the waypoints from the demonstrations is detailed in section V-A.

B. Framework details

The architectural choices for the framework parts were guided by the actual functions they had to satisfy. The aim of the image embedding module is to extract pertinent characteristics from various individual views. These features are later fused by the 2D-to-8D projection model to recover the final 8D waypoints. Furthermore, we can state that the image embedding module aims to extract information about the 8D waypoints projected in various 2D views. The 2D-to-8D projection module can be interpreted as a sophisticated learned triangulation method that projects 2D spatial information into the 3D scene space.

1) Image Embedding: We took inspiration from a cutting-edge deep learning architecture for 2D human pose estimation, called HRNet [20], to develop our image embedding module. HRNet has the capability of generating feature maps with high spatial resolution as well as rich semantic information. This combination is critical for learning manipulation tasks, particularly those involving small objects, as it provides a comprehensive understanding of the objects in the scene. In this work, we employ the HRNet-W32 [20] architecture as the feature extraction backbone, which will consequently generate M-channels feature maps for each view in 2D space.

2) MFT: Our Multi-view Fusion Transformer (MFT) is a simple Transformer-based network which aims to encode the latent waypoints features in 2D space, to 8D waypoints in 3D space, as shown in 3. Firstly, the latent feature map of each view is flattened to M 1-D tensor, which are then fed into a multi-layer perceptron (MLP) to obtain implicit features for 8D waypoints. The shape of these features is $M \times 16$, where M is then number of waypoints, and 16 is the latent dimension of the features. After the pre-embedding step, the generated features from each view are processed using our proposed MFT module, which includes learnable spatial embedding. Subsequently, four implicit vectors are produced after passing through a residual MLP layer, followed by four transformer blocks in parallel, with a cross-connection configuration[13]. The residual connection is also utilized in this block to moderate gradient vanishing. Finally, the resulting outputs are concatenated and passed through MLP layers to generate the final output.

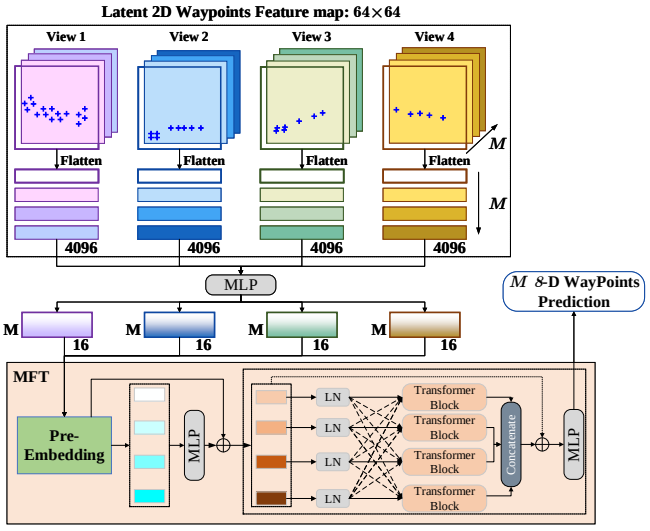


Fig. 3: Framework of our MFT module. A high-level implicit representation with shape of $M \times 16$ will be obtained from the latent 2D waypoints feature map through a flatten operation and MLP layers, prior to its transmission to MFT. Subsequently, these implicit representations will be mapped into vectors with same shape of input through a pre-embedding and a residual MLP layer. Taking generated multi-view vectors as input, our 4 branches cross-transformer will enable the interaction among multi-views features. Besides, a residual connection is also used to alleviate the gradient vanishing.

3) Training details: The whole framework is trained end-to-end to minimize the following supervised loss.

$$L = \sum_{i=1}^{N} \frac{1}{M} \sum_{m=1}^{M} \left(w_m^i - w_m^{i^*} \right)^2$$

, where $\left\{ w_m^i \right\}_{m=1}^{M} = \pi \left(o_0^i; \theta \right)$. π represents the framework that predicts M waypoints from the initial observation of the scene o_0; w_m^i and $w_m^{i^*}$ represent respectively the m-th predicted and ground-truth waypoints for the i-th trajectory in the training dataset.

4) Path Planning: As we conducted our experiments in RLBench [6] simulation, we used the integrated 'ABS_EE_POSE_PLAN_WORLD_FRAME' path planer, which can drive the robot TCP to a specified absolute position and with the specified orientation in the world frame. This path planner relies on the Open Motion Planning Library [19] combined with a feedback-control loop. The gripper control is also integrated to the path planer and is discretized to either 'open_gripper' or 'close_gripper'. The planning always operates in two steps, (1) driving the robot TCP to the desired position, and (2) updating the gripper

	Trash.	Reach	Grill	PickLift	Unplug	MoneyOut
M	4	1	5	4	3	4
T_{max}	564	83	399	460	343	306
T_{min}	117	22	108	103	97	162

TABLE I: Number of waypoints (M) vs min and max total number of steps (respectively T_{max} and T_{min}) when using RLbench demonstration generator. The tasks Trash., Reach, Grill, PickLift, Unplug and MoneyOut correspond to RLBench tasks considered in our experiments, respectively 'put_rubbish_in_bin', 'reach_target', 'meat_off_grill', 'pick_and_lift', 'unplug_charger' and 'take_money_out_safe'.

state.

V. RESULTS

A. Simulation

We conducted our experiments using RLBench [6] simulation because (1) it provides a multi-camera working environment, (2) it gives access to a wide variety of tasks, and most importantly, (3) within the simulation, an automatic task demonstration generator is integrated. This generator utilizes sets of pre-defined waypoints for each available task to create custom waypoint-based datasets that can be used for further analysis and training. As a result, we propose a tool that can generate datasets specifically tailored to the novel paradigm proposed in this work using RLBench. The generated datasets comprise demonstrations of tasks, consisting of initial observations of the scene, and their corresponding "ground truth" sequences of 8D waypoints necessary for successfully completing the respective tasks. As illustrated in Table I, the automatic demonstration generator provides almost two orders of magnitude more steps than the number of waypoints that are actually needed to accomplish the task.

The state of the simulation is also saved for each generated demonstration, so that the tasks can be later replayed with the exact same scene configuration. By leveraging this feature, an evaluation of the trained framework can be executed in three steps. (1) A part of the generated dataset is kept apart for evaluation during the framework training. (2) Once the training is done, the evaluation data are inferred by the trained model, which will predict waypoint sequences corresponding to the evaluation examples. (3) Simulation is iteratively loaded with scene configurations that correspond to the evaluation data. The predicted waypoints generated from these configurations are then utilized by the path planner in an attempt to accomplish the assigned tasks. A success rate can be computed over the evaluation data by reporting the number of successful examples over the total number of examples.

B. Is our approach competitive with existing solutions for visual-robotic-manipulation learning ?

To compare our approach to existing robotic manipulation learning solutions, we referred to James *et al.* [7], who ran two sets of experiments in their work. In the first set of experiments, they compared common imitation learning and reinforcement learning methods , including ARM [5], BC, SAC+AE [26], DAC[9], SQIL [17] and DRQ [10], and their baseline, C2F-ARM, [7] , on a set of tasks that can be solved from the front camera only. In a second set of experiments, they evaluated ARM [5] and C2F-ARM [7] on tasks that require more than one camera to be achievable.

ARM, C2F-ARM, SAC+AE, DAC, SQIL and DrQ are RL-based approaches, which means they require thousands of interactions with the environment to learn a task. James *et al.* [7] provided all baselines, including RL ones, with 100 demonstrations for each task, except for their C2F-ARM baseline, which was provided with 10 demonstrations. However, they used a data augmentation method to extend their data. Since we didn't use data augmentation strategies in our case, we provide our model with 300 demonstrations. However, in section VI, we discuss how our paradigm allows considerably more efficient data collection than previous imitation learning paradigms.

As a proof of concept, we selected, (i) two tasks in the first set of experiments, to validate the performances of our framework compared to a wide range of robot manipulation learning methods, and (ii) two tasks in the second set of experiments to validate that our approach can compete with existing multi-view approaches.

We point out that in both sets, the authors had access to both RGB and depth information from the considered cameras, while our method learns to recover depth information from RGB images only, using our multi-view fusion module. For this reason, our framework always considers the four cameras placed around the scene in the simulation, namely 'front', 'overhead', 'over-shoulder-right' and 'over-shoulder-left' cameras, but only accesses RGB information.

Note that in section V-D, we discuss two additional tasks that we tried but actually partially failed to train.

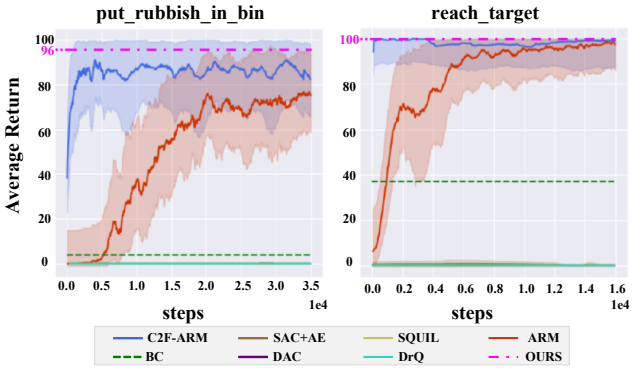


Fig. 4: Our training results compared with [7] learning curves on monocular setting.

Fig. 4 shows the results for the first set of experiments. It highlights that our method (1) outperforms traditional BC by a large margin and (2) slightly outperforms mean results of the best state-of-the-art RL based approach, C2F-ARM, on tasks where the manipulated objects can be seen from all view points on the initial observation of the scene. Other RL

approaches were already outperformed by C2F-ARM, and by extension we outperform them as well.

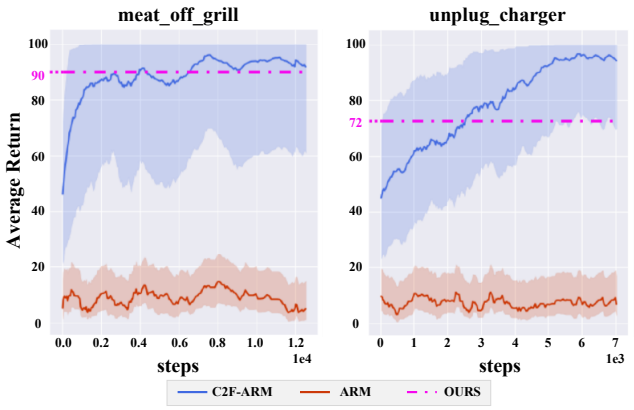


Fig. 5: Our training results compared with [7] learning curves on multi-view setting.

Fig. 5 shows the results for the second set of experiments. It highlights that our method can still compete with C2F-ARM when they also access multiple views.

C. *Does the one-shot waypoints prediction help reduce error accumulation during task execution?*

Section V-B reports evaluation results of various baselines, but the differences among them can be attributed to several factors. Firstly, C2F-ARM, our model, and other existing baselines employ different networks for observation representation. Secondly, while our model predicts waypoints, other approaches predict the gripper pose of the next step. Finally, our model predicts the entire trajectory configuration at once, whereas other approaches rely on sequential decision-making, which can result in prediction errors and observation distribution shifts.

To isolate the impact of error accumulation and associated observation distribution shift on the same tasks studied in section V-B, we trained a behavior cloning model to predict the next waypoint sequentially from low-dim observations. This model has access to the 6D poses of each object in the scene, enabling it to infer ground-truth object positions directly without relying on a vision model to process images of the scene. While the model predicts waypoints like our approach, it can still be affected by prediction errors and observation distribution shifts due to its sequential predictions.

The results of the waypoint-based behavior cloning trained from Ground Truth 6D objects poses and our framework are compared in Table II. This table displays both the training loss and success rate outcomes. To gain a better understanding of how the training results impact the success rate on model deployment, we separated the training loss into three losses: The "Train Pose" loss, which concentrates on the first three dimensions of the 8D waypoints, corresponding to the position of the robot TCP. The "Train Quat" loss focuses on the orientation of the TCP, while the "Train Grip" loss is concerned with the gripper state.

TABLE II: Evaluation of the distribution shift between training and inference.

	Trash.	Grill.	Unplug	PickLift	MoneyOut
Behavior Cloning From 3D Ground Truth					
Train Pose	$3e10^{-5}$	$3e10^{-4}$	$3e10^{-5}$	$3.5e10^{-4}$	$1e10^{-5}$
Train Quat.	$3e10^{-4}$	$4.6e10^{-3}$	$5e10^{-5}$	$5.2e10^{-2}$	$4e10^{-5}$
Train Grip.	$6.3e10^{-3}$	$1.4e10^{-3}$	$2.8e10^{-3}$	$5e10^{-6}$	$1.5e10^{-2}$
Success rate	0	100	70	0.8	0.8
Ours					
Train Pose	$1.6e10^{-2}$	$9.5e10^{-3}$	$8.6e10^{-3}$	$5.7e10^{-2}$	$1.3e10^{-2}$
Train Quat.	$2.3e10^{-2}$	$2.2e10^{-2}$	$2.4e10^{-2}$	0.53	$4.3e10^{-2}$
Train Grip.	$1.7e10^{-3}$	$2.7e10^{-3}$	$1.3e10^{-3}$	$2.2e10^{-3}$	$3.9e10^{-3}$
Success Rate	96	90	72	–	46

Compared to our approach, the behavior cloning model trained from object poses has significantly lower "Train Pose" and "Train Quat" losses. Without any observation distribution shift between training and inference, or prediction error accumulation, the behavior cloning model should be much more accurate than our framework, and, by extension, should achieve a better success rate. However, our framework can produce similar or superior success rates. In shorter, despite the behavior cloning approach shows better prediction performances during training, the final success rate is affected by error accumulation during task execution, which is not the case with our framework. The exception is the 'pick_and_lift' task, which our model failed to grasp. For the 'put_rubbish_in_bin' task, which completely fails, we noticed that the behavior cloning model was trapped in the neighborhood of the first waypoint, while never actually reaching this waypoint. This was an out-of-distribution case for the trained model

D. *Failure cases*

This section highlights the current limitations of our framework. Fig.6 points out mitigated results on take_money_out task, while Fig.7 highlights that typical failure case for our framework is caused by critical occlusions on the initial observation of the scene.

Fig. 6: Moderate results compared with [7] learning curves.

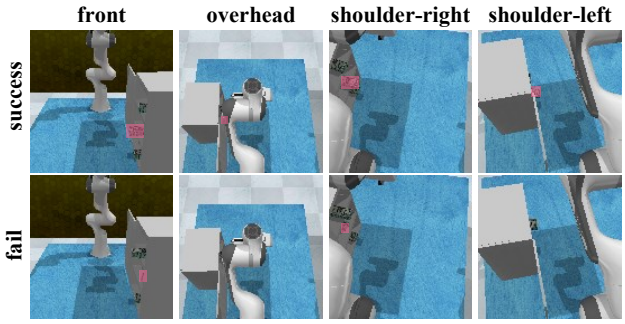


Fig. 7: Visualization of initial observation of the scene in failure vs in success cases on take_money_out task.

VI. CONCLUSION

Although impressive progress has been made, RL and standard BC still facing some intrinsic flaws, e.g. huge amount of interaction for RL and accumulation error for BC. In this work, we proposed an offline hierarchical behavior cloning paradigm, with motivation of eliminating accumulation errors using a fixed spatial representation. Two stages are combined in our framework. Firstly, a keypoints-based high-level planner is employed to represent the fixed spatial configuration only from the very initial observation, with the form of task-relevant waypoints. Secondly, a low-level robotic path planner is used to guide the robot by reaching predicted waypoints. Extensive experiment illustrated that our method can achieve similar performance with RL approaches, and outperforms existing BC approaches in some task, with only inference the initial observation. However, since all the trajectory waypoints are predicted before the task execution, the proposed architecture is mostly suitable for static scenes.

REFERENCES

[1] Murtaza Dalal, Deepak Pathak, and Ruslan Salakhutdinov. Accelerating robotic reinforcement learning via parameterized action primitives. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[2] Coline Devin, Abhishek Gupta, Trevor Darrell, Pieter Abbeel, and Sergey Levine. Learning modular neural network policies for multi-task and multi-robot transfer. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 2169–2176. IEEE, 2017.

[3] Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv preprint arXiv:1812.00568*, 2018.

[4] Jianfeng Gao, Zhi Tao, Noémie Jaquier, and Tamim Asfour. K-vil: Keypoints-based visual imitation learning. *arXiv preprint arXiv:2209.03277*, 2022.

[5] Stephen James and Andrew J Davison. Q-attention: Enabling efficient learning for vision-based robotic manipulation. *IEEE Robotics and Automation Letters*, 7(2):1612–1619, 2022.

[6] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

[7] Stephen James, Kentaro Wada, Tristan Laidlow, and Andrew J Davison. Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13739–13748, 2022.

[8] Edward Johns. Coarse-to-fine imitation learning: Robot manipulation from a single demonstration. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[9] Ilya Kostrikov, Kumar Krishna Agrawal, Debidatta Dwibedi, Sergey Levine, and Jonathan Tompson. Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. In *International Conference on Learning Representations*, 2019.

[10] Ilya Kostrikov, Denis Yarats, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*, 2020.

[11] Tejas D Kulkarni, Ankush Gupta, Catalin Ionescu, Sebastian Borgeaud, Malcolm Reynolds, Andrew Zisserman, and Volodymyr Mnih. Unsupervised learning of object keypoints for perception and control. *Advances in neural information processing systems*, 32, 2019.

[12] Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, 32, 2019.

[13] Wenhao Li, Hong Liu, Hao Tang, Pichao Wang, and Luc Van Gool. Mhformer: Multi-hypothesis transformer for 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13147–13156, 2022.

[14] Dimitrios Mallis, Enrique Sanchez, Matt Bell, and Georgios Tzimiropoulos. From keypoints to object landmarks via self-training correspondence: A novel approach to unsupervised landmark discovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[15] Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *arXiv preprint arXiv:2108.03298*, 2021.

[16] Soroush Nasiriany, Huihan Liu, and Yuke Zhu. Augmenting reinforcement learning with behavior primitives for diverse manipulation tasks. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 7477–7484. IEEE, 2022.

[17] Siddharth Reddy, Anca D Dragan, and Sergey Levine. Sqil: Imitation learning via reinforcement learning with sparse rewards. *arXiv preprint arXiv:1905.11108*, 2019.

[18] Robin Strudel, Alexander Pashevich, Igor Kalevatykh, Ivan Laptev, Josef Sivic, and Cordelia Schmid. Learning to combine primitive skills: A step towards versatile robotic manipulation §. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4637–4643. IEEE, 2020.

[19] Ioan A. Sucan, Mark Moll, and Lydia E. Kavraki. The open motion planning library. *IEEE Robotics Automation Magazine*, 19(4):72–82, 2012.

[20] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5693–5703, 2019.

[21] Supasorn Suwajanakorn, Noah Snavely, Jonathan J Tompson, and Mohammad Norouzi. Discovery of latent 3d keypoints via end-to-end geometric reasoning. *Advances in neural information processing systems*, 31, 2018.

[22] Kentaro Wada, Edgar Sucar, Stephen James, Daniel Lenton, and Andrew J Davison. Morefusion: Multi-object reasoning for 6d pose estimation from volumetric fusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14540–14549, 2020.

[23] Zhuo Xu, Haonan Chang, Chen Tang, Changliu Liu, and Masayoshi Tomizuka. Toward modularization of neural network autonomous driving policy using parallel attribute networks. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 1400–1407. IEEE, 2019.

[24] Chuanyu Yang, Kai Yuan, Qiuguo Zhu, Wanming Yu, and Zhibin Li. Multi-expert learning of adaptive legged locomotion. *Science Robotics*, 5(49):eabb2174, 2020.

[25] Jingyun Yang, Junwu Zhang, Connor Settle, Akshara Rai, Rika Antonova, and Jeannette Bohg. Learning periodic tasks from human demonstrations. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8658–8665. IEEE, 2022.

[26] Denis Yarats, Amy Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and Rob Fergus. Improving sample efficiency in model-free reinforcement learning from images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10674–10681, 2021.