2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Learning Temporally Composable Task Segmentations with Language

Divyanshu Raj, Omkar Patil, Weiwei Gu, Chitta Baral and Nakul Gopalan

***Abstract*— In this work, we present an approach to identify sub-tasks within a demonstrated robot trajectory with the supervision provided by language instructions. Learning longer horizon tasks is challenging with techniques such as reinforcement learning and behavior cloning. Previous approaches have split these long tasks into shorter tasks that are easier to learn by using statistical change point detection methods. However, classical changepoint detection methods function only with low dimensional robot trajectory data and not with high dimensional inputs such as vision. Our goal in this work is to split longer horizon tasks, represented by trajectories into shorter horizon tasks that can be learned using conventional behavior cloning approaches using guidance from language. In our approach we use techniques from the video moment retrieval problem on robot trajectory data to demonstrate a high-dimensional generalizable change-point detection approach. Our proposed moment retrieval-based approach shows a more than 30% improvement in mean average precision (mAP) for identifying trajectory sub-tasks with language guidance compared to that without language. We perform ablations to understand the effects of domain randomization, sample complexity, views, and sim-to-real transfer of our method. In our data ablation we find that just with a 100 labelled trajectories we can achieve a 61.41 mAP, demonstrating the sample efficiency of using such an approach. Further, behavior cloning models trained on our segmented trajectories outperform a single model trained on the whole trajectory by up to 20%.**

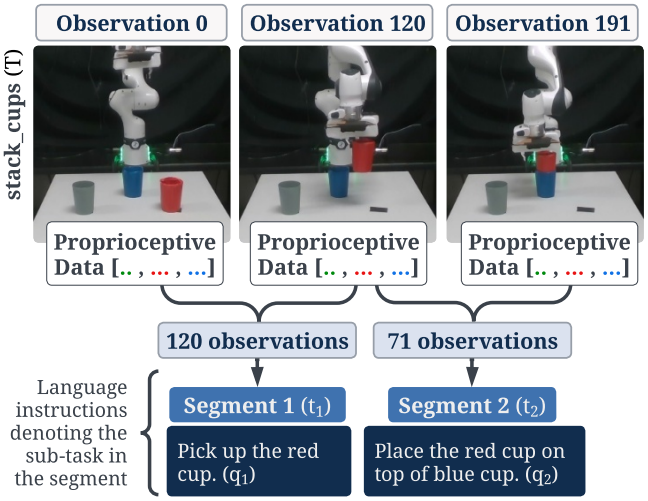

Fig. 1: An illustrative example showcasing the segmentation of a trajectory (T) into sub-tasks (t_i) with language instructions (q_i) for the task of stacking cups on the real robot. The task of stacking cups is composed of two distinct segments of pick and place actions.

I. INTRODUCTION

Learning long-horizon tasks is challenging for existing robot learning frameworks. Policy learning algorithms either require a large number of human demonstrations, or an intractable number of trials in the real environment [1], [2], [3], [4], [5]. Moreover, the learned policy for a long-horizon task has poor generalization to other tasks that share parts of the learned task. For example, consider the tasks of putting a bowl in the dishwasher versus taking a bowl out of the dishwasher. These two tasks share the some of the same abstractions: open the dishwasher, pick up a bowl, place the bowl down, and close the dishwasher. However, existing learning algorithms do not have the ability to generalize to the other longer task even after learning one of the two tasks. Prior work in task segmentation to learn longer tasks is on low dimensional robot data [6], [7], [8]. In this work, we propose a novel robot trajectory segmentation framework that converts high-dimensional long-horizon trajectories into shorter and re-usable segments with linguistic guidance. Furthermore, we demonstrate that composing behavior cloning policies trained with segments of the tasks learned by our approach achieve higher or comparable success rates when compared to traditional behavior cloning methods with greater magnitude of improvement in low data scenarios.

Previous approaches used abstractions to leverage learning from demonstration(LfD) in robots[8], [7], [9], [10]. These works formulate the robot trajectory segmentation as a change point detection problem, where the transitions between different segments are demarcated as statistically significant change points within the robot trajectory. However, these methods focus on low-dimensional data; therefore cannot work with high-dimension input data such as images. On the other hand, video moment retrieval is a problem statement naturally akin to retrieving segments for high-dimensional robot trajectory, albeit without extension in robotics domain [11], [12].

We propose to learn high-dimensional robot trajectory segmentation with inspiration from the video moment retrieval problem using guidance from language. We can then learn the corresponding motor skills from the segmented high-dimensional but shorter horizon data. Our approach alleviates the complications of learning from long-horizon trajectories such as compounding errors, and allows the agent to compose the learned motor skills to solve long-horizon tasks. Moreover, the agent can generalize to novel tasks that are combinations of the learned motor skills. In summary,

School of Computing and Augmented Intelligence, Arizona State University, Tempe, Arizona, United States. Divyanshu Raj conducted this work before joining Amazon. Contact: divraz@amazon.com, {opatil3, weiweigu, cbaral, ng}@asu.edu

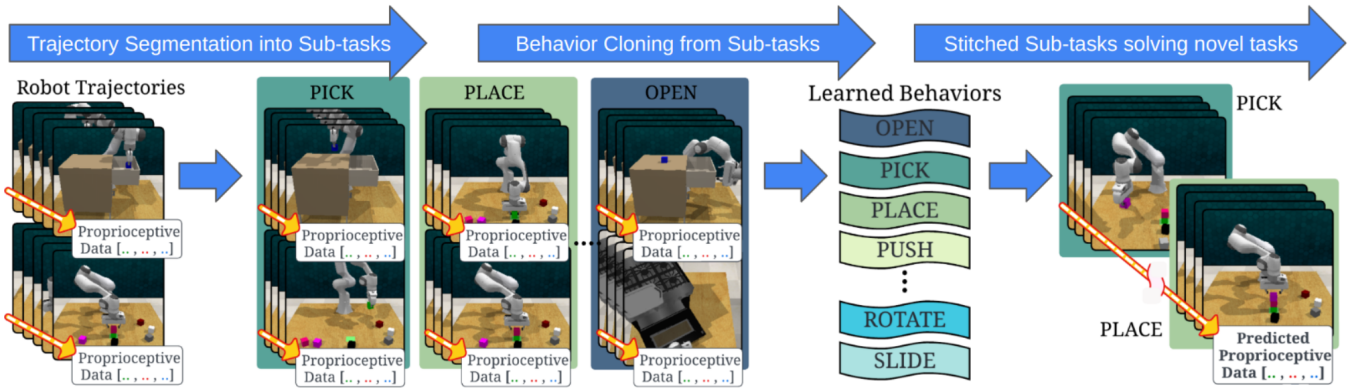

Fig. 2: Our entire pipeline performs changepoint detection and robot learning. Firstly, we segment robot trajectories into sub-tasks with ourtrajectory segmentation model. These segmented trajectories are used to train policies using standard behavior cloning approaches, where each policy corresponds to one specific motor skill. In evaluation, we compose the learned policies to perform long-horizon tasks.

the contribution of our work is the following:

- We demonstrate the use of transformer based moment retrieval frameworks to segment robot trajectories with image and proprioceptive data conditioned on language instructions with a novel dataset created with the RLBench [13] simulator. We significantly improved the average mAP values by 34%, with language guidance and proprioceptive observations for segmenting robot trajectories. We report an average mAP value of 35.15, when applying the segmentation model trained on simulation to evaluate real-world robot trajectories.
- We conduct various ablations, with one significant result demonstrating that labeling just a 100 trajectories with change points and language achieves a mAP of 61.41 with an IoU threshold of 0.5. Such small amounts of labelling can ease the learning of long horizon tasks on the robot.
- While traditional methods for robot change point detection are limited in their ability to select the level of subtask granularity, our approach demonstrates the ability to generalize across different sub-task granularity levels. Our method allows task segmentation at the granularity of "pick a block" and at the level of "make a stack."
- Finally, we show that learning from segmented trajectories can improve the reusability and data efficiency for behavior cloning models. Our method has improved the performance of the state-of-the-art behavior cloning model on long-horizon tasks by up to 20%.

II. RELATED WORK

Previous research in robot task learning has explored the use of changepoint detection to learn long horizon behaviors with fewer samples [9], [14], [10]. Generally, unsupervised Bayesian changepoint detection approaches [15], [16] are used to provide task segmentation with some known parameters such as observation noise in a robot's proprioceptive state. These techniques have even been used with inverse reinforcement learning to segment unstructured demonstrations to learn long horizon behaviors [14]. However, these changepoint detection techniques are low dimensional with few avenues to advance towards using images, 3D data, and proprioceptive state to segment tasks.

Recent research has explored the use of language in solving robot instruction-following problems. Semantic parsing has been used to solve goal based instruction-following problems [17], [18]. MacGlashan et al. [19] learned to map language to a reward function through Inverse Reinforcement Learning, where objects and rooms were pre-specified in their domains rather than learned from scratch. Other end-to-end learning methods require millions of episodes to learn simple behaviors using reinforcement learning [20], [21]. Du et al. [22] introduce a method that utilizes large-scale language model pretraining to shape exploration in reinforcement learning. A related work Gopalan et al. [8] focuses on using change point detection to enable robots to learn sub-tasks and interpret langauge instructions but they do not use language to learn sub-tasks themselves.

In our pursuit to allow high-dimensional sub-task identification, we identified similarities between the problem of moment retrieval used for querying videos with language and identifying change points within trajectories in robotics. Previous work in video moment retrieval has demonstrated the use of language to retrieve a segment of a video that matches the language query [23], [24], [12], [25]. More recent approaches use multi-modal language-based end-to-end transformer models to identify spoken sentences [26] or moments within videos [27], [28], [29], [30] specified using a language query. The paper by Lei et al. [12] introduces a transformer encoder-decoder model called Moment-DETR, which approaches moment retrieval by framing it as a direct set prediction task which makes it suitable for a changepoint detection problem. Our work leverages language-conditioned trajectory data to train models for robot manipulation tasks, utilizing image frames and proprioceptive robot data, generated with RLBench [13] simulations.

Once trajectories are segmented into sub-tasks, behavior

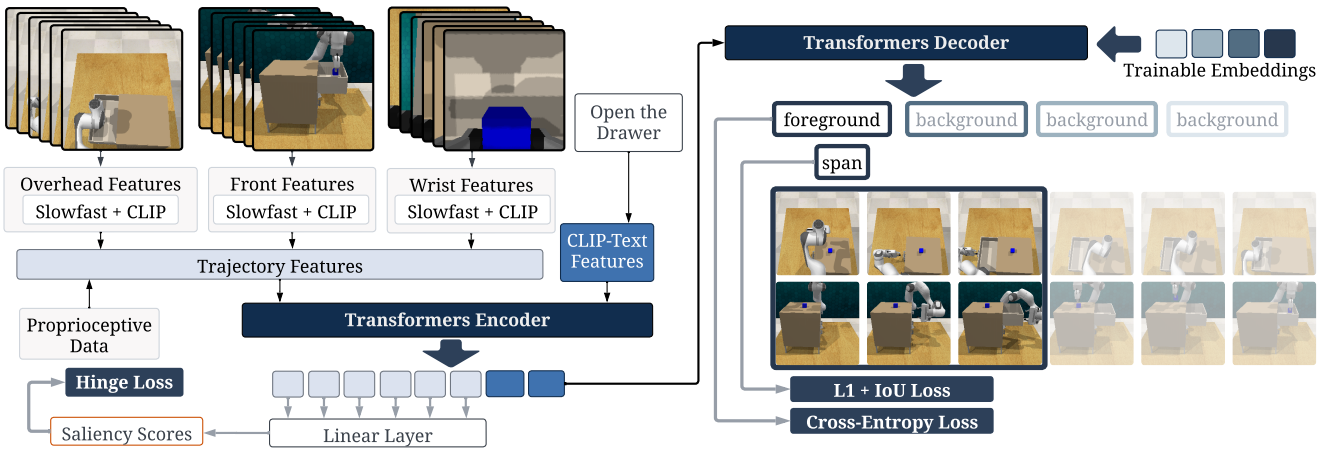

Fig. 3: The diagram illustrates an overview of the Change-Point detection model. The trajectory features include slowfast and CLIP features for the three camera's points of view along with robot proprioceptive data and CLIP features for language instruction. The architecture has a transformer encoder-decoder with three prediction heads for predicting foreground vs back-ground classifications, and change-point coordinates.

cloning [31], [32] offers a direct approach for learning behaviors and assessing the re-usability of learned behaviors for new trajectory demonstrations as shown in Fig 2. Behavior cloning is widely used in robotics to solve various tasks such as robot manipulation [33], [1], [2] and navigation [34], [35]. Robotics Transformers [1], [2], [3] show that provided a sufficient amount of data, behavior cloning models can achieve outstanding performance on manipulation tasks. Previous work also shows that it is possible to learn fine-grained manipulation from behavior cloning [33], or learn a unified model for various policies [36]. These methods rely on the emergent behaviors from foundation models that are pre-trained with hundreds of thousands of robot trajectories. On the contrary, our work focuses on learning from behavior cloning with data efficiency. By segmenting a long-horizon trajectory into a sequence of shorter horizon trajectories, our language condition changepoint detection approach augments behavior cloning to provide significant sample efficiency.

III. DATASET CREATION

In this section, we describe the dataset creation process. Our dataset includes over 10,000 robot trajectories annotated with language-based sub-task segmentation from both the real robot and simulation. We first curate 20 long-horizon tasks with rich semantics on the RLBench [13] simulation environment and collect segmented robot trajectories from the built-in planner. Then, we collect trajectories for a subset of the tasks on the real robot with expert demonstrations and manually label the segmentation for these trajectories.

A. RLBench Environment

RLBench [13] is a learning environment that is widely used to benchmark robot learning with over 100 tasks at varying levels of difficulties. The simulator allows us to capture proprioceptive observations and visual observations, including RGB, depth, and segmentation masks from cameras of multiple different angles. Each individual task in RLBench includes various initial configurations of objects to guarantee the robustness of the benchmark, with demonstrations provided by the in-built planner and the waypoints from the environment.

B. Task Segmentation

To create semantically meaningful task segmentation for each task T, we decompose T into a sequence of sub-tasks $t_1, \ldots, t_k$ and define the corresponding language instructions $q_1, \ldots, q_k$. Additionally, to increase the diversity and richness of the language instructions, we expand each language instruction with more variations using generative language models [37]. A trajectory of a given task T can be divided into k segments, where each changepoint c_i is defined as the success condition of the corresponding sub-task t_i being fulfilled. We then use the corresponding language instruction q_i as the description for each segment. The process of segmenting a trajectory is illustrated in Fig. 1.

C. Simulation Data Generation

We pick 20 long-horizon tasks from the RLBench environment with a diverse range of sub-tasks such as *pick*, *place*, *push*, *pull*, etc. For each of these tasks, we randomly sample 500 initial scenes and collect the corresponding robot trajectory using the in-built planner from the environment. Visual observations from three camera points of view and proprioceptive observation are recorded for each trajectory. To increase the robustness of the model trained with the dataset, we use domain randomization by changing object textures on visual observation with half of the trajectories. We label these robot trajectories with a list of changepoints, which chunk the trajectory into segments, and language instructions that correspond to each segment. The changepoints are the time steps where success conditions of the

Observations				Changepoint Detection						Highlight Detection
Camera Angles			Action	Short (0 - 6 seconds)		Medium (7 - 10 Seconds)		Long (11 - 40 Seconds)		mAP @
F	O	W		R1 @ 0.7	mAP avg	R1 @ 0.7	mAP @ avg	R1 @ 0.7	mAP @ avg	Very Good
✓				$48.89_{\pm 1.72}$	$48.70_{\pm 0.53}$	$54.88_{\pm 1.25}$	$57.83_{\pm 0.62}$	$57.22_{\pm 1.29}$	$60.64_{\pm 1.14}$	$84.50_{\pm 0.51}$
✓			✓	$52.06_{\pm 1.27}$	$50.68_{\pm 0.65}$	$54.77_{\pm 1.43}$	$58.33_{\pm 0.48}$	$58.20_{\pm 1.43}$	$61.81_{\pm 0.71}$	$86.63_{\pm 0.31}$
	✓			$47.19_{\pm 2.00}$	$47.65_{\pm 0.98}$	$54.87_{\pm 1.84}$	$58.05_{\pm 1.19}$	$58.07_{\pm 1.52}$	$60.68_{\pm 0.64}$	$83.32_{\pm 0.67}$
	✓		✓	$50.31_{\pm 1.22}$	$49.82_{\pm 0.63}$	$54.92_{\pm 2.01}$	$58.60_{\pm 0.70}$	$58.36_{\pm 1.63}$	$61.38_{\pm 1.13}$	$85.74_{\pm 0.48}$
		✓		$\mathbf{62.78}_{\pm 1.63}$	$\mathbf{57.67}_{\pm 0.81}$	$\mathbf{65.86}_{\pm 1.57}$	$\mathbf{66.80}_{\pm 1.27}$	$\mathbf{63.08}_{\pm 1.80}$	$\mathbf{66.99}_{\pm 1.32}$	$\mathbf{97.10}_{\pm 0.27}$
		✓	✓	$\mathbf{63.24}_{\pm 2.05}$	$\mathbf{57.62}_{\pm 0.74}$	$\mathbf{66.30}_{\pm 1.21}$	$\mathbf{67.28}_{\pm 1.08}$	$\mathbf{63.20}_{\pm 1.69}$	$\mathbf{67.27}_{\pm 0.94}$	$\mathbf{96.77}_{\pm 0.63}$
✓	✓			$50.15_{\pm 1.58}$	$49.47_{\pm 0.85}$	$55.03_{\pm 1.82}$	$58.78_{\pm 0.24}$	$57.33_{\pm 1.74}$	$60.87_{\pm 1.31}$	$85.15_{\pm 0.49}$
✓	✓		✓	$52.23_{\pm 1.68}$	$50.74_{\pm 0.75}$	$55.70_{\pm 2.19}$	$58.84_{\pm 0.81}$	$58.62_{\pm 1.71}$	$62.58_{\pm 0.93}$	$86.95_{\pm 0.42}$
	✓	✓		$\mathbf{62.16}_{\pm 1.98}$	$\mathbf{57.49}_{\pm 0.76}$	$\mathbf{65.08}_{\pm 1.23}$	$\mathbf{66.81}_{\pm 0.88}$	$\mathbf{62.91}_{\pm 2.23}$	$\mathbf{66.76}_{\pm 0.54}$	$95.54_{\pm 0.72}$
	✓	✓	✓	$\mathbf{62.37}_{\pm 1.70}$	$\mathbf{57.49}_{\pm 0.83}$	$\mathbf{64.80}_{\pm 1.49}$	$\mathbf{66.69}_{\pm 0.48}$	$\mathbf{63.07}_{\pm 2.07}$	$\mathbf{67.10}_{\pm 0.96}$	$95.99_{\pm 0.59}$
✓		✓		$\mathbf{63.24}_{\pm 1.27}$	$\mathbf{57.36}_{\pm 0.60}$	$\mathbf{65.62}_{\pm 1.38}$	$\mathbf{66.81}_{\pm 0.68}$	$\mathbf{63.08}_{\pm 2.10}$	$\mathbf{67.04}_{\pm 0.68}$	$\mathbf{96.56}_{\pm 0.27}$
✓		✓	✓	$\mathbf{62.58}_{\pm 0.83}$	$\mathbf{57.71}_{\pm 0.72}$	$\mathbf{65.88}_{\pm 0.76}$	$\mathbf{66.67}_{\pm 0.79}$	$\mathbf{62.89}_{\pm 1.79}$	$\mathbf{66.22}_{\pm 0.72}$	$\mathbf{96.38}_{\pm 0.48}$
✓	✓	✓		$\mathbf{62.06}_{\pm 1.36}$	$\mathbf{57.21}_{\pm 0.17}$	$\mathbf{64.75}_{\pm 0.96}$	$\mathbf{66.26}_{\pm 1.05}$	$\mathbf{62.78}_{\pm 1.73}$	$\mathbf{67.21}_{\pm 0.81}$	$95.59_{\pm 0.33}$
✓	✓	✓	✓	$\mathbf{61.56}_{\pm 2.77}$	$\mathbf{56.86}_{\pm 1.48}$	$\mathbf{65.00}_{\pm 2.00}$	$\mathbf{66.02}_{\pm 0.87}$	$\mathbf{63.03}_{\pm 2.32}$	$\mathbf{67.44}_{\pm 1.08}$	$\mathbf{95.30}_{\pm 1.65}$

TABLE I: Results from Ablation Study 1: for short, medium, and long horizon sub-tasks (F: Front, O: Overhead, W: Wrist)

sub-tasks are satisfied, which are automatically recorded by the environment when the robot trajectory is created. The language instructions are generated from the task and sub-task specifications.

D. Post-processing

We extract compact features from the trajectory observations by first converting visual observations into 10 FPS videos. These videos are then segmented into 2-second clips, each labeled with the associated language instruction. The 2-second interval strikes a balance between capturing temporal information and computational efficiency. To capture the importance on clip level, we assign highlight scores ranging from 0 to 4 to each clip. We use the HERO feature extractor [11] to convert these clips into features and concatenate the features from all the cameras. At the end of every 2-second clip, we capture the robot's proprioceptive data as part of the observation. These observations, along with the visual features extracted from the clip constitute the trajectory features.

E. Robotics Setup for Real-world Trajectories

To collect robot trajectories in the real world, we use a Franka Emika Research 3 arm (FR3) and three real-sense D435 depth cameras. The three cameras provide views from the front, overhead, and the robot's wrist respectively to match the RLBench simulator's setup. We collect these trajectories paired with videos from a two-step process. Firstly, we collect the robot trajectories through kinematic teaching from a human expert. We then replay the demonstrated trajectories to obtain the visual and proprioceptive data.

IV. METHODS

We combine a segmentation model and a sequence of behavior cloning models. The segmentation model decomposes a long-horizon task into a sequence of sub-tasks, each accompanied by language guidance. The behavior cloning model is trained to perform one of these sub-tasks. By sequencing the predictions of the behavior cloning models, our method enables the execution of a long-horizon task with a higher success rate. We will describe each component in this section.

A. Trajectory Segmentation Model

We first describe the trajectory segmentation model. Conditioned on natural language instructions, the trajectory segmentation model predicts the relevant segments from a trajectory.

1) Architecture: We adopt Moment-DETR [12], a transformer-based video localization model as the backbone for our changepoint detection model as shown in Fig. 3. The input to the model includes a natural language instruction q and observations $\mathcal{O}^\tau$ over a trajectory τ, where at each timestep, an observation O_t^τ represents a clip of 2 seconds. Firstly, we jointly encode the natural language instruction q and the observation $\mathcal{O}^\tau$ with a transformer encoder. The representation obtained from the encoder, denoted as E_{encode}, is fed into two subsequent modules, a feedforward neural network, and a transformer decoder. We use the feedforward neural network to predict the saliency score for each timestep of the encoded sequence E_{encode}. Along with the encoded sequence E_{encode}, a sequence of trainable positional embeddings is fed into the transformer decoder as the initial representation of the segment. The decoder then computes the representation for each segment. We determine the relevance of each segment to the natural language instruction by computing a span and classifying it using a feedforward neural network on the representation derived from the transformer decoder.

2) Training: We optimize the segmentation model with a composite loss function that is used for moment localization[12]. To evaluate change-point detection, we use Intersection over Union (IoU) metric that tells the accuracy of segmentation by assessing the overlap between predicted and ground truth segments. We use mean average precision (mAP) that measures the accuracy of model's prediction considering both precision and recall across different confidence thresholds range $[0.5 : 0.05 : 0.95]$. We also use a standard

metric Recall@1 (R1), similar to prior moment detection works [12]. The loss function consists of a loss for the segment prediction $\mathcal{L}_{\text{segment}}$ and a loss for the saliency score $\mathcal{L}_{\text{salience}}$. The segment prediction loss $\mathcal{L}_{\text{segment}}$ quantifies the dissimilarity between a prediction and a ground truth segment, which further breaks down into a classification loss for the relevance with the natural language instruction q and a loss for span prediction:

$$\mathcal{L}_{\text{segment}} = \lambda_{\text{classification}}\mathcal{L}_{\text{classification}} + \mathcal{L}_{\text{span}},$$

where $\lambda_{\text{classification}}$ is a hyperparameter. We use cross-entropy loss as the classification loss, and a combination of $L1$ and a 1D temporal IoU as the loss for span prediction. The span prediction loss can be written as:

$$\mathcal{L}_{\text{span}}(s_i, \hat{s}_i) = \lambda_{L1}||s_i - \hat{s}_i|| + \mathcal{L}_{\text{IoU}}(s_i, \hat{s}_i),$$

where λ_{L1} is a hyperparameter that controls the contribution of the $L1$ loss. The saliency loss is a classification loss for the relevance of each token of the observation. The overall loss $\mathcal{L}$ can be written as a linear combination of the segment prediction loss and the saliency loss, denoted as:

$$\mathcal{L} = \lambda_{\text{saliency}}\mathcal{L}_{\text{saliency}} + \mathcal{L}_{\text{segment}},$$

where $\lambda_{\text{saliency}}$ is a hyperparameter for the saliency loss.

B. Behavior cloning

We use a behavior cloning model similar to the state-of-the-art ACT(Action Chunking Transformer) [33] but without the CVAE encoder. The model takes visual and proprioceptive observations as input and predicts a chunk of actions, where the size of the chunk is a hyper-parameter. Based on empirical results from existing work, we use a chunk size of 100 for configurations, and directly optimize the model by minimizing the $L1$ loss between the predicted action chunks and the ground truth action chunks. The same architecture is used both as the baseline that we compare against, and as our model to learn the sub-tasks of a long-horizon task.

V. RESULTS

Our results have two major components. Firstly, we present a thorough experimentation with the trajectory segmentation model. These experiments demonstrate that the trajectory segmentation model is capable of segmenting robot trajectories into sub-tasks conditioned on language instructions. Secondly, we empirically demonstrate that training behavior cloning models on data segmented using our approach is more sample efficient than training a single model for long-horizon tasks. To evaluate change-point detection, we use Intersection over Union (IoU) metric that tells the accuracy of segmentation by assessing the overlap between predicted and ground truth segments. We use mean average precision (mAP) that measures the accuracy of model's prediction considering both precision and recall across different confidence thresholds range [0.5 : 0.05 : 0.95]. We also use a standard metric Recall@1 (R1), similar to prior moment detection works [12]. Due to space limitations on the manuscript, details of the change-point detection model, dataset, code, and detailed results of all the ablation studies are present on the companion website [1].

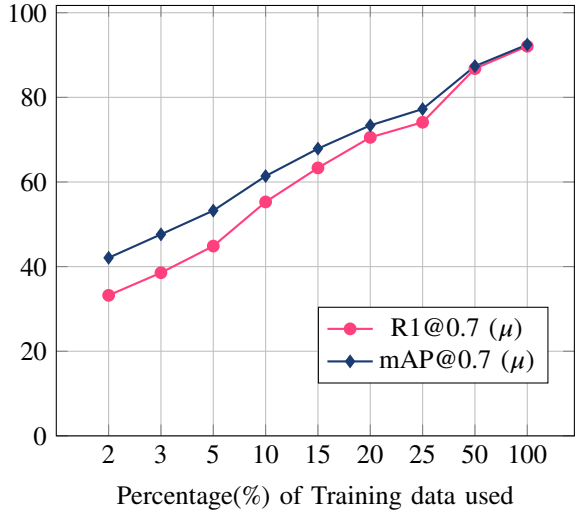

Fig. 4: Evaluation metrics by decreasing the training size. Even with a 100 (2% of data) trajectories we get around 40% mAP@0.7 performance which points to the usability of our approach in sparse data domains.

A. Experiment on Trajectory Segmentation

The experiment results on trajectory segmentation have two major components, results from simulation data and the evaluation done on real robot data. For the results of the simulation data, we consider evaluations across two levels of generalization: 1) Generalization to new robot trajectories and new object positions in seen tasks, and 2) Generalization to unseen tasks. We split our simulation trajectories accordingly and report the results of five-fold validation for both levels of generalization. The simulation trajectories were divided into two parts, one with domain randomization and one without. We conduct a study with exhaustive combinations of observations on both data sets, with and without domain randomization across both levels of generalization. We evaluate the models trained with all combinations against the real robot data.

*1) **Generalization to unseen robot trajectories**:* Here we test unseen combinations of object and robot positions to evaluate models trained on the same tasks, assessing the ability of our model to generalize in novel scenarios beyond the training data.

a) Experiments on Data without Domain Randomization: We present the experiment results on simulation data without domain randomization for unseen trajectories in Table I. Our experiments demonstrate that the configuration of the inputs consisting of the wrist camera and proprioceptive data of the robot performs best, with an average mAP of 57.62, 67.28, and 67.27 for short, medium, and long-horizon sub-tasks respectively. All the configurations involving the wrist camera have an average mAP above 66

[1] https://sites.google.com/asu.edu/change-point

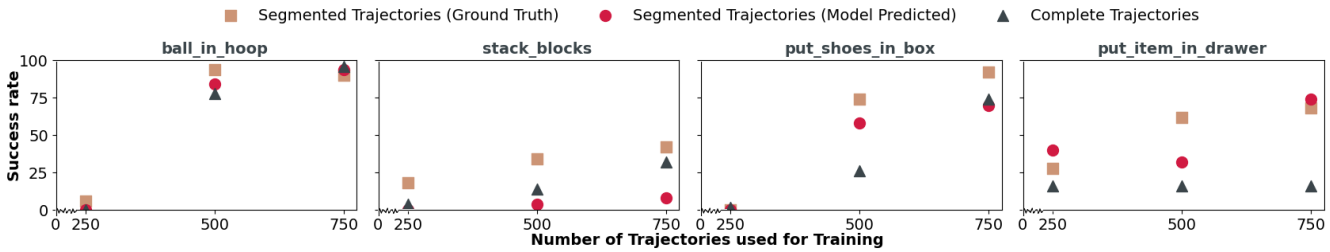

Fig. 5: This figure presents the success rate on various tasks for behavior cloning models trained with different strategies. Triangles(△) present the results for BC models that are trained with the complete trajectories. Squares(□) present the results for BC models that are trained with ground truth segmented trajectories. Circles(○) are the results for BC models trained with trajectories segmented by our segmentation model. We use different figures to denote results from different tasks: *ball_in_hoop*, *stack_blocks*, *put_shoes_in_box*, and *put_item_in_drawer*. There are two points to note. Firstly, the BC models trained with complete trajectories (△), perform worse than ground truth segmented (□) BC models or models segmented with our method (○), that is, they are never the best approach across all tasks or sample sizes. Secondly, our predicted model does well in three of the four tasks compared with the BC models trained on full trajectories. We still under-perform when compared to ground truth segmentation based BC models with significant room for improvements. Our model performs the worst on the tasks of *stack_blocks* consistently because it requires a precise placement which might fail with a segmentation model that is imprecise at the task of segmentation. We also want to point out that our segmentation model got no training data for the *ball_in_hoop* task. Across other tasks as well our model generalizes to novel placements of objects and has never seen the exact same problem being solved.

for medium and long horizon sub-tasks. We hypothesize that this configuration performs better because the wrist camera directly associates the gripper movements and object grasps with the trajectory change points.

We define the baseline experiment as training and evaluating the trajectory segmentation model using the same methodology as the language-conditioned approach, but without providing language instructions as input. No existing algorithm for change-point detection works on high-dimensional trajectories, so our experiments are limited to ablation studies.

For the configuration with wrist camera and proprioceptive data, the baseline models yield an average mAP of 32.89, compared to the average mAP of 67.27 in Table I for long horizon sub-tasks. These findings demonstrate a significant enhancement in sub-task identification, highlighting the significance of language guidance. Detailed results are available at our website's section "Baseline[1]".

We varied the granularity of the sub-task segmentation to demonstrate the model's ability to generalize across different levels of sub-tasks. Instead of segmenting the trajectory into fine-grained sub-tasks like 'pick a red block' and 'place on the yellow block,' we evaluated trajectories where these actions are combined into a single, higher-level sub-task, such as 'pick the red block and place it on the yellow block.' For the configuration with wrist camera and proprioceptive data, training a model with data that has different levels of segmentation yields an average mAP of 86.71 for long horizon sub-tasks. The comparison with the average mAP of 67.27 in Table I for the same configuration demonstrates that our model can generalize across varying granularities of tasks, a capability that standard change point detection models cannot achieve. Detailed results are available at our website's section "Ablation Study 3[1]".

b) Experiments on Data with Domain Randomization: We train a model where the object texture is randomized along with object positions. The configuration where the input consists of wrist camera and proprioceptive data of the robot performs best with average mAP of 58.04, 67.08, and 68.00 for short, medium, and long-horizon sub-tasks respectively, and performs slightly better than non-domain randomization results from Table I. Detailed results are available at "Ablation Study 2[1]".

c) Evaluation on real robot data: The evaluation of real-world data compares segmentation models developed with and without domain randomization, utilizing data collected from the Franka Emika Research 3 (FR3) robot. The evaluation yields an average mAP of 35.15 as compared to the average mAP of 64.95 in Table I for long horizon sub-tasks. The detailed results available at "Evaluation 1[1]" demonstrate that the models trained with domain randomization perform better on real-world data. Again the models with a wrist camera as the input perform better than other models.

*2) **Generalization to unseen tasks**:* The evaluation focuses on assessing the trajectory segmentation models under conditions where specific tasks are withheld during training. This study aims to evaluate the models' generalization capabilities when encountering unseen tasks during the evaluation phase.

a) Experiments on Data without Domain Randomization: The configuration where the input consists of the wrist camera and proprioceptive data of the robot performs best, with an average mAP of 28.55, 32.17, and 20.69 for short, medium, and long-horizon sub-tasks respectively. All the configurations involving the wrist camera have an average

mAP above 25 for medium horizon sub-tasks which is higher than the configurations without wrist camera. The detailed results available at "Ablation Study 4[1]" demonstrate that the configuration where the input consists of the wrist camera performs better for all ranges of sub-tasks. The empirical results support our hypothesis that the wrist camera point of view is important for change-point detection due to the clear visibility of objects and robot grasps, facilitating generalization to unseen tasks.

b) Experiment on Data with Domain Randomization: The empirical results demonstrate that the configuration where the input consists of the wrist camera performs best, with an average mAP of 21.87 and 27.00 for short and medium horizon sub-tasks respectively, whereas the configuration with both the wrist camera and proprioceptive data performs best for long horizon sub-tasks, with an average mAP of 20.97. Detailed results available at "Ablation Study 5[1]", support our hypothesis of the importance of the wrist camera.

c) Evaluation on real robot data: The evaluation of real-world data compares models trained with and without domain randomization where some of the tasks are withheld. The evaluation yields an average mAP of 26.94 as compared to the average mAP of 64.95 in Table I for long horizon sub-tasks. Detailed evaluation available at "Evaluation 2[1]" demonstrates that the configuration of front and wrist cameras perform better for short-horizon sub-tasks with an average mAP of 5.01 whereas the configuration of wrist and overhead cameras perform better for medium-horizon sub-tasks with an average mAP of 17.29. The average mAP values for all horizon sub-tasks are comparatively lower for the sim-to-real transfer than those in Table I, yet they still support our hypothesis regarding the importance of the wrist camera's point of view in change point detection.

*3) **Sample Efficiency**:* In a real-world robotics scenario, having a large number of sample trajectories can be impossible, so this study aims to evaluate how many trajectories would be required on an actual robot. In the ablation study shown in Figure 4 we demonstrate the effect on R1@0.7 and mAP@0.7 metrics by decreasing the number of trajectories in training data. With just 100 trajectories, we demonstrate a mAP@0.7 of 42.85. With 750 trajectories, we get the mAP@0.7 of 68.44 which is comparable to the average mAP of 64.95 in Table I. The results demonstrate the sample complexity strength of our approach.

B. Behavior Cloning Results

We train and evaluate our behavior cloning model on 4 of the 20 tasks – *put_item_in_drawer*, *put_shoes_in_box*, *ball_in_hoop* and *stack_blocks*. We chose the tasks to have a wide range of lengths, various numbers of segments, and different precision requirements, providing a thorough comparison between our method and the baseline behavior cloning model. We chose only four tasks because of limited availability of computational resources. The experimental results for behavior cloning (BC) models trained on ground truth segmentation (□), segmentation produced by our model (○) and baseline behavior cloning with no segmentation (△) are presented in Fig. 5.

The results demonstrate that stitching short and robust policies learned from ground truth segmented data are more sample efficient than learning a single long-horizon policy. This result was expected as the ground truth model can always be composed together and by reducing the horizon of learning the behavior cloning model performs better. The BC models trained on task segments provided by our model outperform a single long-horizon policy on most tasks, while consistently performing poorly on tasks that demand precision. The behavior cloning results in Fig. 5 show that a standard behavior cloning model augmented with our learned task segmentation model performs better than just standard behavior cloning by up to 20% on task completion for most tasks under different amounts of data provided. Moreover, to demonstrate the robustness of our trajectory segmentation model, we leave the task *ball_in_hoop* out of it's training data. Our segmentation model generalizes this task out of the box. The BC models are then trained on the splits generated from the change-points predicted by our segmentation model on this unseen task and consistently perform better demonstrating the strengths of such an approach.

However, the standard behavior cloning policy performs better than our method for the *stack_blocks* task that requires high precision, which is harder to achieve with imperfect predicted task segmentation. On the other hand, a clear improvement in performance can be observed for the task *put_item_in_drawer* on using our trajectory segmentation method which requires avoiding collisions during task solving which the BC model can focus on in shorter segments. Our method also shows clear improvements in the low data regime, while standard behavior cloning tends to catch up with an increasing number of training demonstrations due to the variability in the boundary of the segments predicted by our model. For instance, we obtain better performance on the task *put_shoes_in_box* at lower data regimes showcasing the the sample-efficiency of our method.

As the ground truth segmentation assisted behavior cloning model still performs the best across all tasks (squares in Fig. 5 are the highest) there is a performance gap that our approach still needs to close. The performance gap might be closed with reinforcement learning based fine-tuning allowing our imperfect models to improve the goal reachability of skills by trial and error and learn the fine manipulations required for tasks such as *stack_blocks*. We still believe that the generalization with the *ball_in_the_hoop* task demonstrates the strengths of an approach like ours in learning long horizon manipulation skills. More exploration in this area is required.

VI. CONCLUSION

We present a language conditioned task segmentation approach for high-dimensional robot trajecories. Our model demonstrates significant improvements in trajectory segmentation accuracy on high dimensional data (34% improvement) while demonstrating sample efficiency(mAP of 61.41

with only 100). Moreover, we demonstrate the ability to change sub-task granularity and a pathway towards sim-to-real transfer of the changepoint detection models. Our model demonstrates generalization by segmenting unseen tasks to improve learning with standard behavior cloning models. Further, behavior cloning models trained on our segmented trajectories outperform a single model trained on the whole trajectory by up to 20%. While our model does not outperform all tasks when compared to a standard behavior cloning pipeline, the high performance of the ground truth model suggests that more research is needed in this area. Overall, our approach aims to improve the sample efficiency of language conditioned robotic learning approaches, while presenting a through analysis of the strengths and weaknesses our approach in multiple domains.

REFERENCES

[1] A. Brohan et al., "Rt-1: Robotics transformer for real-world control at scale," 2023.

[2] ——, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," 2023.

[3] A. Padalkar et al., "Open x-embodiment: Robotic learning datasets and rt-x models," 2023.

[4] S. Levine et al., "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International journal of robotics research*, vol. 37, no. 4-5, pp. 421–436, 2018.

[5] D. Kalashnikov et al., "Scalable deep reinforcement learning for vision-based robotic manipulation," in *Conference on robot learning*. PMLR, 2018, pp. 651–673.

[6] G. Konidaris et al., "Constructing skill trees for reinforcement learning agents from demonstration trajectories," in *Advances in Neural Information Processing Systems*, J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds., vol. 23. Curran Associates, Inc., 2010. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2010/file/27ed0fb950b856b06e1273989422e7d3-Paper.pdf

[7] S. Niekum et al., "Learning grounded finite-state representations from unstructured demonstrations," *The International Journal of Robotics Research*, vol. 34, no. 2, pp. 131–157, 2015. [Online]. Available: https://doi.org/10.1177/0278364914554471

[8] N. Gopalan et al., "Simultaneously learning transferable symbols and language groundings from perceptual data for instruction following," *RSS*, 2020. [Online]. Available: https://www.roboticsproceedings.org/rss16/p102.pdf

[9] G. Konidaris et al., "Robot learning from demonstration by constructing skill trees," *The International Journal of Robotics Research*, vol. 31, no. 3, pp. 360–375, 2012. [Online]. Available: https://doi.org/10.1177/0278364911428653

[10] S. Niekum et al., "Learning and generalization of complex tasks from unstructured demonstrations," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 5239–5246.

[11] L. Li et al., "Hero: Hierarchical encoder for video+language omni-representation pre-training," *EMNLP*, 2020.

[12] J. Lei et al., "Qvhighlights: Detecting moments and highlights in videos via natural language queries," *NeurIPS*, 2021.

[13] S. James et al., "Rlbench: The robot learning benchmark & learning environment," *IEEE Robotics and Automation Letters*, 2020.

[14] P. Ranchod et al., "Nonparametric bayesian reward segmentation for skill discovery using inverse reinforcement learning," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 471–477.

[15] P. Fearnhead et al., "Efficient bayesian analysis of multiple changepoint models with dependence across segments," 2009.

[16] E. Fox et al., "A sticky hdp-hmm with application to speaker diarization," *The Annals of Applied Statistics*, pp. 1020–1056, 2011.

[17] M. MacMahon, B. Stankiewicz, and B. Kuipers, "Walk the talk: Connecting language, knowledge, and action in route instructions." 01 2006.

[18] Y. Artzi et al., "Weakly supervised learning of semantic parsers for mapping instructions to actions," *TACL*, vol. 1, 2013. [Online]. Available: https://aclanthology.org/Q13-1005

[19] J. MacGlashan et al., "Grounding english commands to reward functions," *RSS*, 2015. [Online]. Available: https://www.roboticsproceedings.org/rss11/p18.pdf

[20] K. Hermann et al., "Grounded language learning in a simulated 3d world," *arXiv preprint arXiv:1706.06551*, 2017.

[21] D. Chaplot et al., "Gated-attention architectures for task-oriented language grounding," *AAAI*, 2018.

[22] Y. Du et al., "Guiding pretraining in reinforcement learning with large language models," *ICML*, 2023.

[23] J. Gao et al., "Tall: Temporal activity localization via language query," *ICCV*, 2017.

[24] L. Hendricks et al., "Localizing moments in video with natural language," 2017.

[25] J. Lei et al., "Tvr: A large-scale dataset for video-subtitle moment retrieval," *ECCV*, 2020.

[26] Y. Yuan et al., "To find where you talk: Temporal sentence localization in video with attention based location regression," *AAAI*, 2019.

[27] M. Cao et al., "On pursuit of designing multi-modal transformer for video grounding," *EMNLP*, 2021.

[28] S. Zhang et al., "Exploiting temporal relationships in video moment localization with natural language," *ACM*, 2019.

[29] C. Rodriguez-Opazo et al., "Proposal-free temporal moment localization of a natural-language query in video using guided attention," *WACV*, 2020.

[30] J. Nam et al., "Zero-shot natural language video localization," *ICCV*, 2021.

[31] H. Ravichandar et al., "Recent advances in robot learning from demonstration," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, no. 1, pp. 297–330, 2020. [Online]. Available: https://doi.org/10.1146/annurev-control-100819-063206

[32] B. Argall et al., "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889008001772

[33] T. Zhao et al., "Learning fine-grained bimanual manipulation with low-cost hardware," 2023.

[34] M. Shridhar et al., "Alfred: A benchmark for interpreting grounded instructions for everyday tasks," *CVPR*, 2020. [Online]. Available: https://arxiv.org/abs/1912.01734

[35] K. Nguyen et al., "Vision-based navigation with language-based assistance via imitation learning with indirect intervention," 2019.

[36] H. Bharadhwaj, J. Vakil, M. Sharma, A. Gupta, S. Tulsiani, and V. Kumar, "Roboagent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking," 2023.

[37] T. Brown et al., "Language models are few-shot learners," 2020.