Task Robustness via Re-Labelling Vision-Action Robot Data

Artur Kuramshin[1,2], **Özgür Aslan**[1,2], **Cyrus Neary**[1,2,3], **Glen Berseth**[1,2]
[1]Mila — Quebec AI Institute, Canada, [2]Université de Montréal, Canada
[3]The University of British Columbia, Canada
cyrus.neary@ubc.ca, {artur.kuramshin, ozgur.aslan, glen.berseth}@mila.quebec

Abstract:

The recent trend in scaling models for robot learning has resulted in impressive policies that can perform various manipulation tasks and generalize to novel scenarios. However, these policies continue to struggle with following instructions, likely due to the limited linguistic and action sequence diversity in existing robotics datasets. This paper introduces **T**ask **R**obustness via R**E**-Labelling Vision-**A**ction Robot **D**ata (TREAD), a scalable framework that leverages large Vision-Language Models (VLMs) to augment existing robotics datasets without additional data collection, harnessing the transferable knowledge embedded in these models. Our approach leverages a pretrained VLM through three stages: generating semantic sub-tasks from original instruction labels and initial scenes, segmenting demonstration videos conditioned on these sub-tasks, and producing diverse instructions that incorporate object properties, effectively decomposing longer demonstrations into grounded language-action pairs. We further enhance robustness by augmenting the data with linguistically diverse versions of the text goals. Evaluations on LIBERO demonstrate that policies trained on our augmented datasets exhibit improved performance on novel, unseen tasks and goals. Our results show that TREAD enhances both planning generalization through trajectory decomposition and language-conditioned policy generalization through increased linguistic diversity. Project website: https://akuramshin.github.io/tread

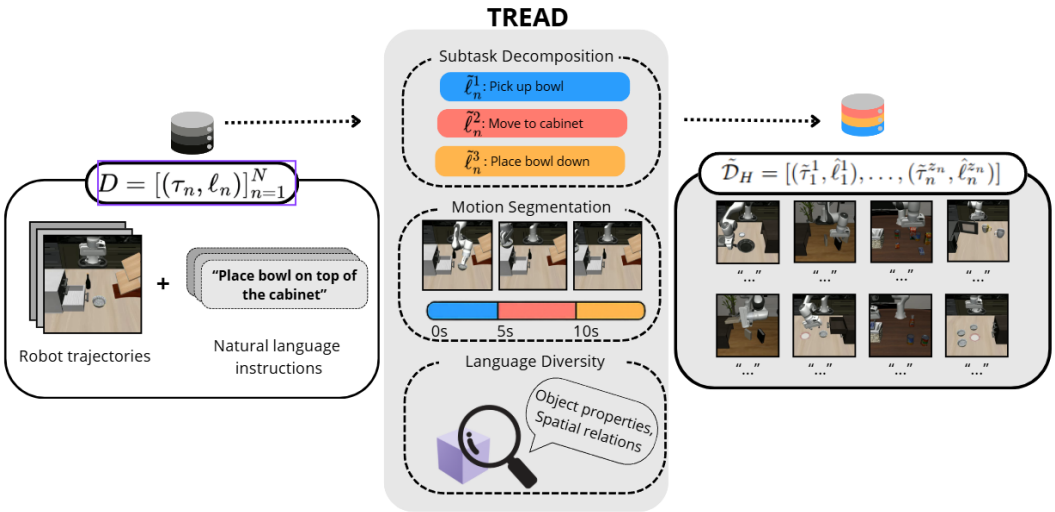

Figure 1: TREAD uses a large-scale VLM to programmatically cut trajectories at sub-goals, label those sub-goals and add variations to the goal-text.

1 Introduction

Recent trends in robot learning research have demonstrated the importance of data scale and diversity for learning more robust and capable robotic manipulation policies [1, 2, 3]. While the robotics community has made progress in expanding dataset sizes through collaborative efforts, these datasets continue to exhibit limited diversity along the text and trajectory modalities [4, 5, 6]. This limitation manifests as a weakness of the current models to reliably follow text-based instructions [7, 8, 9].

The dataset modality imbalance persists due to the challenges of collecting diverse, real-world demonstrations at scale, often requiring multiple robots and months of operation to obtain even modest datasets [10, 5, 11]. However, these vision-based manipulation demonstrations have grown in length and complexity, covering many potential sub-goals per trajectory. While it is possible to manually review the thousands of demonstration videos and solicit human annotators, this will not scale with the demands of modern robotic learning. This raises the question: How can we systematically augment robotics datasets with greater language-action diversity in a scalable way?

Recent advances in large-scale vision-language models (VLMs) [12, 13, 14] offer a promising new approach to address the language-action diversity challenge in robotics datasets. Using VLMs for robot data augmentation is appealing for two reasons. By leveraging internet-scale pretraining, VLMs are capable of zero-shot generation of contextually appropriate language labels that are also grounded in the physical scenes depicted in demonstration videos. Second, these models can effectively reason about temporal sequences in videos [12, 13, 15, 14], enabling segmentation of demonstrations into meaningful sub-tasks. This capability allows for decomposing existing demonstrations into more granular language-action segments, effectively increasing the diversity of our training data without requiring additional data collection.

Motivated by the strong capabilities of large-scale pretrained VLMs, we propose **T**ask **R**obustness via R**E**-Labelling Vision-**A**ction Robot **D**ata (TREAD), a framework to augment robotics datasets. TREAD shown in fig. 1 operates in an iterative three-stage process, where the model's outputs from previous steps are fed back as inputs to the next: First, a pretrained VLM analyzes the original instruction labels and the initial scenes to generate a sequence of semantic sub-task descriptions that collectively accomplish the original goal. Then the demonstration videos are processed by the the VLM to temporally segment the trajectories, identifying which portions of the demonstration correspond to each sub-task description. Lastly, the VLM generates diverse instructions for these sub-taks incorporating object properties and spatial relationships conditioned on the initial scenes of the sub-tasks and the semantic sub-task descriptions. This temporal-semantic alignment creates multiple language-conditioned sub-demonstrations from each original demonstration from the offline dataset.

In this work, we make two key contributions: (1) a novel framework to utilize iterative querying of a VLM to augment and diversify the original dataset, and (2) the augmented dataset obtained by applying our framework, which will be released to facilitate future research. In our experiments with the LIBERO dataset [16], we show that the resulting dataset improves zero-shot performance and text goal following of the vision-language-action policy Octo [17] and π_0-FAST [7].

2 Related Work

Robot learning datasets. Recently, the robotics community has invested substantial effort into collecting larger manipulation datasets. While earlier datasets such as RT-1 [10], MT-Opt [18], and BC-Z [19] boast significant numbers of trajectories, they lack diversity in tasks and scenes, which is more important for generalization [5, 20, 21]. Responding to this need, several works have focused on increasing diversity by scaling data collection, including BridgeV2 [6], DROID [5], and RDT-1B [9]. These newer datasets not only collect data across more scenes but also demonstrate more complicated tasks consisting of multiple meaningful sub-goals. However, this creates a new challenge: although the number of tasks has increased, the language diversity is diluted by the larger amount of image-action frames per instruction due to longer trajectories. Our work presents

a generic framework for expanding the language-image-action distribution of any robotic manipulation dataset without requiring additional data collection, effectively reducing this dilution problem.

Vision-language-action models as scalable robot policies. Vision-Language Models (VLMs) have shown increasing capabilities in complex visual tasks such as state identification, reasoning, and visual question answering [22, 23, 12, 14]. Inspired by the success of VLMs, the robotics community adopted similar multi-modal architectures for imitation learning [10, 17, 24, 9]. More recent methods start from a pre-trained VLM backbone to leverage internet-scale pretraining [2, 3]. Followup works have explored architectural modifications [8, 9, 7] to improve performance in instruction generalization and following. In contrast, our work improves existing policy performance through data augmentation alone. Therefore, our method can be used alongside any policy architecture.

Data augmentation methods. Data augmentation is a powerful technique for enhancing model robustness and generalization by generating additional synthetic training examples from existing data. In robot learning, augmentation methods have developed along three main dimensions: visual augmentation [1, 25, 26, 27], language augmentation [28, 29, 30], and trajectory augmentation [28, 31, 32]. Our work bridges language and trajectory augmentation by using VLMs to both relabel trajectories and identify meaningful sub-task motions within longer demonstrations. DIAL [30], which is most similar to our approach in terms of language augmentation, relies on a predetermined set of labels and requires fine-tuning on related robotics data. In contrast, our method leverages zero-shot VLM capabilities. NILS [33] is another framework that incorporates multiple large models and heuristics to segment and relabel trajectories, whereas our method relies on a single iteratively queried VLM without heuristics, making it simpler to reproduce. Lastly, SPRINT [28], another closely related work, focuses on composing shorter skills into longer sequences. However, our approach does the opposite by decomposing longer-horizon demonstrations into meaningful sub-tasks. This makes our method particularly suitable for enhancing language-action diversity in open-source robotics datasets.

3 Method

Here we describe how we design TREAD to segment and stitch robotics data, in three stages: (1) *segmentation* using a VLM to process trajectories for semantic and motion-based properties to find key sub-goal transitions, (2) *labeling* of the new sub-trajectories with sub-goals and to augment the original dataset with language diversity, and (3) *train* a vision-language-action model on the resulting dataset to improve robustness. The pseudocode for TREAD is shown in Algorithm 1.

3.1 Problem Formulation

We assume access to an offline dataset $\mathcal{D}$ of robot trajectories labelled with natural language task instructions. Formally, we define a dataset $\mathcal{D} = [(\tau_n, \ell_n)]_{n=1}^{N}$ consisting of N labeled trajectories, where each trajectory $\tau_n = [(\mathbf{o}_t^n, \mathbf{a}_t^n)]_{t=1}^{T}$ contains observation-action pairs over T timesteps. Here, $\mathbf{o}_t^n$ and $\mathbf{a}_t^n$ denote the image observation and action at time t respectively, and ℓ_n denotes the natural language instruction describing the task demonstrated in trajectory τ_n. We then train a policy $\pi(\cdot \mid \mathbf{o}_t, \ell)$ via imitation learning to generate action distributions that mimic demonstrated behavior. Specifically, we look at models trained by minimizing the bahavior cloning objective:

$$\hat{\pi}^* = \arg\min_{\pi} \sum_{(\tau,\ell)\in\mathcal{D}} \sum_{(\mathbf{o}_t,\mathbf{a}_t)\in\tau} \mathcal{L}_{BC}(\pi(\cdot \mid \mathbf{o}_t, \ell), \mathbf{a}_t)$$

where $\mathcal{L}_{BC}$ denotes a supervised loss (e.g., negative log-likelihood) between the predicted action distribution and the demonstrated action.

We are interested in using data augmentation to improve the learned policy's generalization performance. Specifically, we address the question: how can we augment $\mathcal{D}$ with additional trajectory-instruction pairs (τ, ℓ) derived from existing demonstrations, without requiring new data collection?

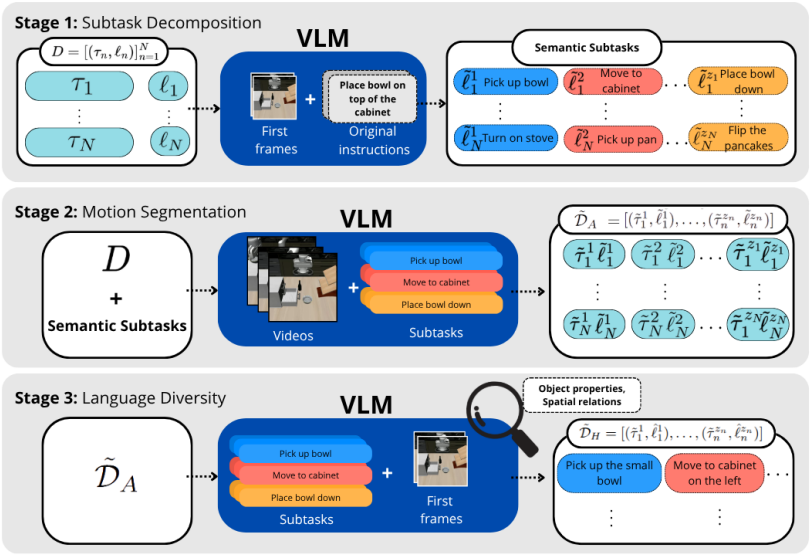

Figure 2: **TREAD pipeline overview.** Given dataset $\mathcal{D}$ of N labeled trajectories $[(\tau_n, \ell_n)]_{n=1}^{N}$, TREAD decomposes the dataset into semantically meaningful sub-trajectories through three stages: (1) **Subtask Decomposition**: Given the original task instruction ℓ and initial frame $\mathbf{o}_1$, we prompt the VLM to generate a sequence of sub-task labels $[\tilde{\ell}^1, \ldots, \tilde{\ell}^{z_n}]$. (2) **Motion Segmentation**: The VLM identifies temporal boundaries for each sub-task, resulting in a set of labeled sub-trajectories $[(\tilde{\tau}^1, \tilde{\ell}^1), \ldots, (\tilde{\tau}^{z_n}, \tilde{\ell}^{z_n})]$. (3) **Language Diversity**: The VLM generates diverse paraphrases for each sub-task instruction by leveraging visual context to incorporate object properties (color, shape, material) and spatial relationships (relative positions, orientations).

3.2 Decomposition of Trajectories

Although our framework does not depend on a specific VLM, in this work we use Gemini Pro 2.5 [14]. Gemini is a multi-modal model that can take as input images, videos, and text, and output text. Given a trajectory τ_n completing a long-horizon task ℓ_n, we first derive semantic sub-tasks and then identify which parts of the trajectory complete those sub-tasks. We found that just asking the VLM to segment the video in one-shot provided poor results; providing the model with subtasks resulted in qualitatively better video segmentation. We accomplish this by chaining two VLM inference calls shown in fig. 2. First, we prompt the VLM to create a plan of sub-task motion labels $[\tilde{\ell}_n^1, \tilde{\ell}_n^2, \ldots, \tilde{\ell}_n^{z_n}]$, given the higher-level task ℓ_n, and the first frame of the trajectory $\mathbf{o}_1^n$ for grounding and context. Here z_n denotes the number of sub-tasks that are required to complete task ℓ_n in the scene as determined by the VLM. For example, the task *"place bowl in the top drawer"* will require a different number of sub-task motions depending on whether the top drawer is currently closed or open and the algorithm should identify this and produce three sub-tasks instead of two.

Subsequently, we query the model with the video of the trajectory and prompt it to retrieve the starting and ending seconds at which the previously identified sub-tasks occur. As a result, we now have a process by which we can decompose a labeled trajectory (τ, ℓ) into an ordered set of labeled sub-trajectories $[(\tilde{\tau}^1, \tilde{\ell}^1), (\tilde{\tau}^2, \tilde{\ell}^2), ..., (\tilde{\tau}^z, \tilde{\ell}^z)]$, where $\tilde{\tau} \subset \tau$ denotes a sub-trajectory. Finally, we can now process $\mathcal{D}$ to construct a new *decomposed* dataset:

$$\tilde{\mathcal{D}}_A = [(\tilde{\tau}_1^1, \tilde{\ell}_1^1), \ldots, (\tilde{\tau}_1^{z_1}, \tilde{\ell}_1^{z_1}), \ldots, (\tilde{\tau}_n^1, \tilde{\ell}_n^1), \ldots, (\tilde{\tau}_n^{z_n}, \tilde{\ell}_n^{z_n})]$$

For details on the full prompts, see Appendix A. Practically, we decompose a dataset of 82 unique scene-instruction tasks into 146 subtasks (see Section 4.1 for details on the dataset).

3.3 Instruction Diversity and Data Augmentation

Recent research has highlighted instruction following as a weakness of current VLA models [8, 7, 9], attributing the difficulty to the models paying more attention to visual inputs than the language goal.

While the mentioned works address the problem algorithmically, we argue that additional textual diversity will help robustness to language. Many robotics datasets have considerably small linguistic diversity which potentially leads to policies being overly sensitive to instruction wording. We use the generated sub-trajectories to augment $\mathcal{D}$. We consider the following textual augmentation method for language diversity:

Grounded textual diversity. We can enhance linguistic diversity by leveraging VLMs to generate visually grounded paraphrases of instructions. For each sub-trajectory $(\tilde{\tau}_i, \tilde{\ell}_i)$, we prompt the VLM with both the original instruction $\tilde{\ell}_i$ and the first visual frame from $\tilde{\tau}_i$, requesting k language alternatives that preserve the task semantics. As shown in fig. 2, including the image in the context allows the model to incorporate spatial relationships and perceivable characteristics for more accurate goal diversity beyond just using synonyms. For instance, given an instruction *"pick up the blue coffee mug,"* our VLM might generate alternatives like *"grasp the small coffee mug"* (object attributes) or *"retrieve the coffee mug next to the laptop,"* (spatial relationships). We apply this language augmentation to our already decomposed dataset $\tilde{\mathcal{D}}_A$ and denote the resulting dataset $\tilde{\mathcal{D}}_H$:

$$\tilde{\mathcal{D}}_H = [(\tilde{\tau}_1^1, \hat{\ell}_1^1), \ldots, (\tilde{\tau}_1^{z_1}, \hat{\ell}_1^{z_1}), \ldots, (\tilde{\tau}_n^1, \hat{\ell}_n^1), \ldots (\tilde{\tau}_n^{z_n}, \hat{\ell}_n^{z_n})]$$

See Appendix A for the templated prompt we use for grounded textual diversity.

Algorithm 1 The TREAD algorithm

Input: Robotic dataset $\mathcal{D}$ and VLM $\mathcal{G}$

function TREAD($\mathcal{D}, \mathcal{G}$)
  $\tilde{\mathcal{D}}_A \leftarrow$ DECOMPOSE($\mathcal{D}, \mathcal{G}$)
  $\tilde{\mathcal{D}}_H \leftarrow$ DIVERSIFY($\tilde{\mathcal{D}}_A, \mathcal{G}$)
  Train policy π_A on mixture of $\mathcal{D}$ and $\mathcal{D}_A$
  Train policy π_H on mixture of $\mathcal{D}$ and $\mathcal{D}_H$
function DECOMPOSE($\mathcal{D}, \mathcal{G}$) ▷ Section 3.2
  $\tilde{\mathcal{D}}_A \leftarrow \{\}$
  for trajectory-instruction pair (τ_n, ℓ_n) in $\mathcal{D}$ **do**
    $\mathbf{o}_1 \leftarrow$ first observation in τ_n
    $\mathbf{v} \leftarrow [\mathbf{o}_1, \ldots, \mathbf{o}_T]$ (video of trajectory τ_n)
    $[\tilde{\ell}_n^1, \ldots, \tilde{\ell}_n^z] \leftarrow \mathcal{G}(\mathbf{o}_1, \ell_n)$
    $[(\tilde{\tau}_n^1, \tilde{\ell}_n^1), \ldots, (\tilde{\tau}_n^z, \tilde{\ell}_n^z)] \leftarrow \mathcal{G}(\mathbf{v}, [\tilde{\ell}_n^1, \ldots, \tilde{\ell}_n^z])$
    $\tilde{\mathcal{D}}_A \leftarrow \tilde{\mathcal{D}}_A \cup [(\tilde{\tau}_n^1, \tilde{\ell}_n^1), \ldots, (\tilde{\tau}_n^z, \tilde{\ell}_n^z)]$
  return $\tilde{\mathcal{D}}_A$
function DIVERSIFY($\tilde{\mathcal{D}}_A, \mathcal{G}$) ▷ Section 3.3
  $\tilde{\mathcal{D}}_H \leftarrow \{\}$
  for sub-trajectory–instruction pair $(\tilde{\tau}_n^i, \tilde{\ell}_n^i)$ in $\tilde{\mathcal{D}}_A$ **do**
    $\mathbf{o}_1 \leftarrow$ first observation in $\tilde{\tau}_n^i$
    $\hat{\ell}_n^i \leftarrow$ choose one from $\mathcal{G}(\mathbf{o}_1, \tilde{\ell}_n^i)$
    $\tilde{\mathcal{D}}_H \leftarrow \tilde{\mathcal{D}}_H \cup (\tilde{\tau}_n^i, \hat{\ell}_n^i)$
  return $\tilde{\mathcal{D}}_H$

3.4 Training a Vision-Language-Action Model

We study the quality of our dataset augmentation by training vision-language-action manipulation policies on the augmented data. Using the datasets generated by TREAD, we finetune two generalist robot policies: Octo (specifically Octo-Small 1.5) [17] and π_0-FAST [7]. Octo is a diffusion-based policy trained on the Open-X dataset [4], while π_0-FAST leverages a pre-trained VLM backbone, PaliGemma [23], trained on the cross-embodied robot data mixture from π_0 [3], with actions discretized using the FAST tokenization scheme.

To determine the dataset weighting ratio such that the finetuned model achieves the best performance across the original full trajectories ($\mathcal{D}$) and relabeled sub-trajectories ($\tilde{\mathcal{D}}_A$/$\tilde{\mathcal{D}}_H$), we conducted ablation studies across multiple mixing ratios. We evaluated ratios ($\mathcal{D}$:$\tilde{\mathcal{D}}_A$/$\tilde{\mathcal{D}}_H$) of *1:2*, *1:1.5* and *1:1.1*.

Figure 3: Example visualization of the sub-task keypoints produced by TREAD for the trajectory "*Put the white mug on the plate and put the chocolate pudding to the left of the plate*". Each frame (from left to right) corresponds to the first frame of the sub-task (labeled above). The sub-task labels in order are: (1) "*Grasp the white mug*" (2) "*Lift the white mug*" (3) "*Move the white mug to the plate*" (4) "*Place the white mug on the plate*" (5) "*Grasp the small rectangular object*" (6) "*Lift the small rectangular object*" (7) "*Move the small rectangular object to the left of the plate*" (8) "*Place the small rectangular object on the table.*"

The final ratio was selected using Re-Mix [34]. For all experiments reported in Section 4.1, we use the best-performing weighting ratio for each respective data mixture. Detailed performance results for each weighting ratio across the evaluation tasks are provided in Appendix B.

For all data mixtures, we employ consistent finetuning parameters across models. We finetune Octo for 50,000 steps, batch size of 256, using a linear warmup followed by cosine decay [35] with a peak learning rate of 3×10^{-4}. For π_0-FAST, we perform full finetuning for 30,000 steps with a batch size of 32, employing the same cosine decay schedule with warmup but with a peak learning rate of 2.5×10^{-5}. Based on test performance, we use checkpoints at step 30,000 for Octo and step 15,000 for π_0-FAST across all dataset ablations (see Section 4).

4 Results

In our evaluation, we design experiments to investigate the effectiveness of training an agent on a dataset augmented by TREAD on unseen tasks. Specifically, our experiments answer the following research questions: (1) Does TREAD's *trajectory* decomposition increase planning generalization? (2) Does *language* augmentation help policy generalization to text-based goals?

We focus on robot manipulation and evaluate our method on the image-based dataset LIBERO [16], a simulation benchmark designed for studying lifelong learning in robotic manipulation. The use of LIBERO to evaluate VLA models pretrained on real-world data is a common practice to show the capabilities of these models [7, 2, 8]. Due to resource constraints, we evaluate our method on a subset of LIBERO as described below.

4.1 Experimental Setup

In our experiments, we evaluate the effects of both trajectory decomposition and instruction diversity on policy performance through ablation studies using the following data mixtures:

1. **Finetuned Octo/π_0-FAST with no augmentations (Original Fine-tuned)**: We finetune Octo/π_0-FAST on the unaltered subset of LIBERO-100 trajectories $\mathcal{D}$. This serves as our comparison point, expected to perform adequately on in-distribution task-instruction pairs but struggle with language-following in novel scenes or instruction text variations.
2. **Finetuned Octo/π_0-FAST with Trajectory Decomposition (TREAD w/o diverse labels)**: We finetune Octo/π_0-FAST on a mixture of the unaltered dataset $\mathcal{D}$ and its decomposition $\tilde{\mathcal{D}}_A$. This ablation isolates the effect of trajectory decomposition, which should help the policy perform tasks in new environments by leveraging subtask compositions seen during training, even when the complete task-environment pairing is novel.
3. **Finetuned Octo/π_0-FAST with Trajectory Decomposition + Label Diversity (TREAD)**: We finetune Octo/π_0-FAST on a mixture of the unaltered dataset $\mathcal{D}$ and its linguistically

enriched decomposition $\tilde{\mathcal{D}}_H$. This full implementation is expected to handle both novel environments and textual variations by bringing diverse text instructions in distribution.

LIBERO. The LIBERO benchmark [16] consists of four task suites that are designed to examine distribution shifts in the object types, the spatial arrangement of objects, the task goals, or the mixture of the previous three. For our experiments, we focus on the LIBERO-100 task suite, which contains 100 tasks involving diverse object interactions and versatile skills with 50 human-teleoperated demonstrations each. In LIBERO-100, tasks are defined by the combination of scene and instruction, meaning that identical instructions performed in different scenes are treated as distinct tasks. Importantly, we utilize the full LIBERO-100 suite rather than the smaller subsets employed by recent works [2, 7]. This larger benchmark presents a more challenging evaluation setting that reduces the risk of overfitting and provides a more rigorous test of our method's ability to improve motion and language generalization (see Section 4.2).

Due to resource constraints, we labeled five demonstrations per task and omitted any tasks performed in `STUDY_SCENE`s for a total of 570 trajectories. An example of a labeled trajectory is shown in Figure 3. To evaluate our research questions, we defined two sets of novel tasks within LIBERO:

4.2 Evaluation Tasks

Motion Generalization (MG): For research question (1), we created new instruction-scene pairs by taking existing language instructions and pairing them with *compatible* but previously unpaired scenes. For example, the instruction *"open the top drawer of the cabinet"* is paired with scene `KITCHEN_SCENE4` containing a cabinet, with this specific combination not appearing in the training data. These tasks will evaluate whether trajectory decomposition helps the agent transfer learned skills to novel environmental contexts. Importantly, these new instruction-scene pairs also evaluate the model's language-following capability, as the model must avoid defaulting to previously trained tasks from those scenes and instead execute the specified instruction. We create 7 such novel scene-instruction environments within the LIBERO framework.

Language Generalization (LG): For research question (2), we created linguistically modified versions of existing LIBERO-100 tasks. These modifications included variations such as reordering clauses around conjunctions (changing *"turn on the stove and put the moka pot on it"* to *"put the moka pot on the stove and turn it on"*) or removing unnecessary object specifiers when context makes them redundant (changing *"close the bottom drawer of the cabinet"* to *"close the drawer"*). These variations assess whether our language augmentation strategies improve policy robustness to variations in instruction wording. We found 14 existing tasks within LIBERO-100 that satisfied our criteria for instruction modification.

We evaluate all dataset ablations across both test cases. For each task, we conduct 30 policy rollouts and report the average success rate. To provide a more nuanced understanding of policy capabilities, we separately analyze performance on single-goal tasks (e.g. *"close the drawer"*) and two-goal tasks (e.g. *"put the bowl in the drawer of the cabinet and close it"*). For two-goal tasks, we report both partial success (completing only one goal) and complete success (achieving both goals).

4.3 Increased Zero-Shot Performance

The Motion Generalization results in Table 1 demonstrate TREAD's effectiveness when policies encounter familiar instructions in novel environments. Both TREAD and TREAD w/o diverse labels outperform Original Fine-tuned across Single Goal tasks, with TREAD achieving 15% and 6% higher success rates on Octo and π_0-FAST respectively. This shows that the automatic segmentation of TREAD is able to provide more trajectory and goal diversity which drives additional generalization across more tasks.

The benefits of trajectory decomposition become more pronounced for complex, multi-step tasks. In two-goal scenarios, TREAD achieves substantially higher first-goal completion rates - 30% improvement for Octo and 9% for π_0-FAST - compared to baseline fine-tuning. The wider performance

Table 1: **LIBERO task performance results.** Success rates (SR) with standard error across the custom evaluation task suites (see Section 4.2) and LIBERO-10 within the LIBERO framework [16]. Results include policies fine-tuned on different data mixtures: original trajectory data (Original Fine-tuned), mixture of the unaltered dataset and its decomposition (TREAD w/o div.), and mixture of the unaltered dataset and its linguistically enriched decomposition (TREAD). Bold and underlined values indicate best and second-best performance.

Test Case	**Metric (%)**	π_0-FAST [7]			Octo [17]		
		Original Fine-tuned	TREAD w/o div.	TREAD	Original Fine-tuned	TREAD w/o div.	TREAD
Language Generalization	Single Goal SR	47±20	**77**±7	<u>67</u>±14	<u>82</u>±2	76±17	**91**±6
	1 of 2 SR	<u>63</u>±11	62±10	**67**±9	<u>76</u>±6	70±6	**77**±4
	2 of 2 SR	<u>36</u>±8	35±7	**39**±10	<u>30</u>±7	28±5	**31**±5
Motion Generalization	Single Goal SR	28±18	<u>31</u>±20	**34**±16	7±4	**27**±10	<u>22</u>±15
	1 of 2 SR	73±3	**82**±11	**82**±9	13±4	**50**±10	<u>43</u>±3
	2 of 2 SR	**7**±3	0±0	0±0	0±0	0±0	0±0
LIBERO-10	1 of 2 SR	**83**±9	72±12	<u>74</u>±8	**76**±5	69±6	<u>72</u>±4
	2 of 2 SR	**57**±10	43±12	**57**±10	<u>40</u>±7	**41**±4	38±4
Average	SR	49±10	50±11	53±10	41±12	45±10	47±11

gap for multi-goal tasks likely stems from the increased opportunities for sub-trajectory overlap between training and test scenarios. Notably, the performance gains are more substantial for Octo, which lacks the pretrained vision-language understanding of π_0-FAST, suggesting that trajectory decomposition provides particularly valuable inductive biases for models without built-in multimodal reasoning capabilities.

4.4 Better Text Goal Following

The Language Generalization results in Table 1 reveal TREAD's ability to handle linguistic variations of familliar task-instructions in known environments. In single-goal tasks, TREAD demonstrates substantial improvements over Original Fine-tuned baselines, achieving 20% higher success rates with Octo and 9% with π_0-FAST. This improvement provides strong evidence that our grounded textual diversity approach effectively prepares policies for the natural linguistic variations.

Interestingly, the performance benefits for Language Generalization are most apparent in shorter, single-goal tasks rather than complex multi-step scenarios. This pattern likely reflects the mismatch between the relatively concise language instructions in our decomposed sub-trajectories and the longer, more complex instructions typical of multi-goal tasks. Additionally, longer-horizon tasks provide more opportunities for compounding errors when policies encounter out-of-distribution language conditioning, making it inherently more challenging to maintain robust language following throughout extended task execution.

Importantly, we verify that our augmentation approach does not compromise performance on in-distribution tasks. When evaluated on the in-distribution long-horizon tasks in LIBERO-10, TREAD maintains comparable performance to Original Fine-tuned for two-goal success rates as seen in the last row in Table 1.

5 Conclusion

In summary, we presented TREAD, a framework that uses off-the-shelf VLMs to perform a type of trajectory segmentation and then stitching via training a large-scale BC model Octo [17]. We find that fine-tuning Octo on our new dataset improves the performance, especially on more difficult longer horizon tasks and novel tasks not seen during training. Future work should explore applying TREAD to larger datasets and larger models. However, our work shows that even with small amounts of properly segmented and augmented data there are obvious performance gains.

6 Limitations and Future Work

Despite these promising results, our work has several limitations that point to interesting directions for future research. First, our work uses a large closed-source VLM, which limits reproducibility. An important next step would be to evaluate TREAD with open-source alternatives [22, 13, 36] and potentially fine-tune these models specifically on robotics data to improve their embodied understanding. Second, while our experiments focused on simulation environments, applying TREAD to real-world datasets like BridgeV2 [6] would provide insight into its effectiveness for physical deployment.

Acknowledgments

If a paper is accepted, the final camera-ready version will (and probably should) include acknowledgments. All acknowledgments go at the end of the paper, including thanks to reviewers who gave useful comments, to colleagues who contributed to the ideas, and to funding agencies and corporate sponsors that provided financial support.

References

[1] 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. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4788–4795. IEEE, 2024.

[2] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[3] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, et al. $\pi 0$: A vision-language-action flow model for general robot control, 2024. *URL https://arxiv. org/abs/2410.24164*.

[4] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

[5] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, P. D. Fagan, J. Hejna, M. Itkina, M. Lepert, Y. J. Ma, P. T. Miller, J. Wu, S. Belkhale, S. Dass, H. Ha, A. Jain, A. Lee, Y. Lee, M. Memmel, S. Park, I. Radosavovic, K. Wang, A. Zhan, K. Black, C. Chi, K. B. Hatch, S. Lin, J. Lu, J. Mercat, A. Rehman, P. R. Sanketi, A. Sharma, C. Simpson, Q. Vuong, H. R. Walke, B. Wulfe, T. Xiao, J. H. Yang, A. Yavary, T. Z. Zhao, C. Agia, R. Baijal, M. G. Castro, D. Chen, Q. Chen, T. Chung, J. Drake, E. P. Foster, J. Gao, V. Guizilini, D. A. Herrera, M. Heo, K. Hsu, J. Hu, M. Z. Irshad, D. Jackson, C. Le, Y. Li, K. Lin, R. Lin, Z. Ma, A. Maddukuri, S. Mirchandani, D. Morton, T. Nguyen, A. O'Neill, R. Scalise, D. Seale, V. Son, S. Tian, E. Tran, A. E. Wang, Y. Wu, A. Xie, J. Yang, P. Yin, Y. Zhang, O. Bastani, G. Berseth, J. Bohg, K. Goldberg, A. Gupta, A. Gupta, D. Jayaraman, J. J. Lim, J. Malik, R. Martín-Martín, S. Ramamoorthy, D. Sadigh, S. Song, J. Wu, M. C. Yip, Y. Zhu, T. Kollar, S. Levine, and C. Finn. Droid: A large-scale in-the-wild robot manipulation dataset. 2024.

[6] H. R. Walke, K. Black, T. Z. Zhao, Q. Vuong, C. Zheng, P. Hansen-Estruch, A. W. He, V. Myers, M. J. Kim, M. Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, pages 1723–1736. PMLR, 2023.

[7] K. Pertsch, K. Stachowicz, B. Ichter, D. Driess, S. Nair, Q. Vuong, O. Mees, C. Finn, and S. Levine. Fast: Efficient action tokenization for vision-language-action models, 2025. URL https://arxiv.org/abs/2501.09747.

[8] M. J. Kim, C. Finn, and P. Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

[9] S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, and J. Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024.

[10] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. In *Proceedings of Robotics: Science and Systems*, 2022.

[11] H.-S. Fang, H. Fang, Z. Tang, J. Liu, C. Wang, J. Wang, H. Zhu, and C. Lu. Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 653–660. IEEE, 2024.

[12] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[13] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, H. Zhong, Y. Zhu, M. Yang, Z. Li, J. Wan, P. Wang, W. Ding, Z. Fu, Y. Xu, J. Ye, X. Zhang, T. Xie, Z. Cheng, H. Zhang, Z. Yang, H. Xu, and J. Lin. Qwen2.5-vl technical report, 2025. URL https://arxiv.org/abs/2502.13923.

[14] G. Comanici, E. Bieber, M. Schaekermann, I. Pasupat, N. Sachdeva, I. Dhillon, M. Blistein, O. Ram, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025. URL https://arxiv.org/abs/2507.06261.

[15] Y. Zhang, J. Wu, W. Li, B. Li, Z. Ma, Z. Liu, and C. Li. Video instruction tuning with synthetic data, 2024. URL https://arxiv.org/abs/2410.02713.

[16] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

[17] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Xu, J. Luo, T. Kreiman, Y. Tan, L. Y. Chen, P. Sanketi, Q. Vuong, T. Xiao, D. Sadigh, C. Finn, and S. Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

[18] D. Kalashnikov, J. Varley, Y. Chebotar, B. Swanson, R. Jonschkowski, C. Finn, S. Levine, and K. Hausman. Mt-opt: Continuous multi-task robotic reinforcement learning at scale. *arXiv preprint arXiv:2104.08212*, 2021.

[19] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022.

[20] F. Lin, Y. Hu, P. Sheng, C. Wen, J. You, and Y. Gao. Data scaling laws in imitation learning for robotic manipulation. *arXiv preprint arXiv:2410.18647*, 2024.

[21] A. Xie, L. Lee, T. Xiao, and C. Finn. Decomposing the generalization gap in imitation learning for visual robotic manipulation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3153–3160. IEEE, 2024.

[22] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. In *NeurIPS*, 2023.

[23] L. Beyer, A. Steiner, A. S. Pinto, A. Kolesnikov, X. Wang, D. Salz, M. Neumann, I. Alabdulmohsin, M. Tschannen, E. Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

[24] Y. Jiang, A. Gupta, Z. Zhang, G. Wang, Y. Dou, Y. Chen, F.-F. Li, A. Anandkumar, Y. Zhu, and L. J. Fan. Vima: Robot manipulation with multimodal prompts. In *International Conference on Machine Learning*, 2023. URL https://api.semanticscholar.org/CorpusID:260844398.

[25] T. Yu, T. Xiao, A. Stone, J. Tompson, A. Brohan, S. Wang, J. Singh, C. Tan, M. Dee, J. Peralta, B. Ichter, K. Hausman, and F. Xia. Scaling robot learning with semantically imagined experience. *ArXiv*, abs/2302.11550, 2023. URL https://api.semanticscholar.org/CorpusID:257079001.

[26] Z. Mandi, H. Bharadhwaj, V. Moens, S. Song, A. Rajeswaran, and V. Kumar. Cacti: A framework for scalable multi-task multi-scene visual imitation learning. *arXiv preprint arXiv:2212.05711*, 2022.

[27] Z. Chen, S. Kiami, A. Gupta, and V. Kumar. Genaug: Retargeting behaviors to unseen situations via generative augmentation. *ArXiv*, abs/2302.06671, 2023. URL https://api.semanticscholar.org/CorpusID:256846801.

[28] J. Zhang, K. Pertsch, J. Zhang, and J. J. Lim. Sprint: Scalable policy pre-training via language instruction relabeling. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9168–9175. IEEE, 2024.

[29] C. Lynch, A. Wahid, J. Tompson, T. Ding, J. Betker, R. Baruch, T. Armstrong, and P. Florence. Interactive language: Talking to robots in real time. *IEEE Robotics and Automation Letters*, pages 1–8, 2023. doi:10.1109/LRA.2023.3295255.

[30] T. Xiao, H. Chan, P. Sermanet, A. Wahid, A. Brohan, K. Hausman, S. Levine, and J. Tompson. Robotic skill acquistion via instruction augmentation with vision-language models. In *Proceedings of Robotics: Science and Systems*, 2023.

[31] J. Chen, H. Fang, H.-S. Fang, and C. Lu. Towards effective utilization of mixed-quality demonstrations in robotic manipulation via segment-level selection and optimization, 2025. URL https://arxiv.org/abs/2409.19917.

[32] D. Raj, O. Patil, W. Gu, C. Baral, and N. Gopalan. Learning temporally composable task segmentations with language. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5195–5202. IEEE, 2024.

[33] N. Blank, M. Reuss, M. Rühle, Ö. E. Yağmurlu, F. Wenzel, O. Mees, and R. Lioutikov. Scaling robot policy learning via zero-shot labeling with foundation models. In *CoRL*, 2024.

[34] J. Hejna, C. Bhateja, Y. Jiang, K. Pertsch, and D. Sadigh. Re-mix: Optimizing data mixtures for large scale imitation learning. *arXiv:2408.14037*, 2024.

[35] I. Loshchilov and F. Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Skq89Scxx.

[36] B. Zhang, K. Li, Z. Cheng, Z. Hu, Y. Yuan, G. Chen, S. Leng, Y. Jiang, H. Zhang, X. Li, P. Jin, W. Zhang, F. Wang, L. Bing, and D. Zhao. Videollama 3: Frontier multimodal foundation models for image and video understanding, 2025. URL https://arxiv.org/abs/2501.13106.

A VLM Prompts

We provide the templated prompts we use for querying Gemini 2.5 for semantic sub-task breakdown and motion identification in Figures 4, 5, and the template prompt for grounded paraphrasing in Figure 6. The three example tasks used for few-shot prompting motion identification are taken from the LIBERO-90 dataset. τ_N

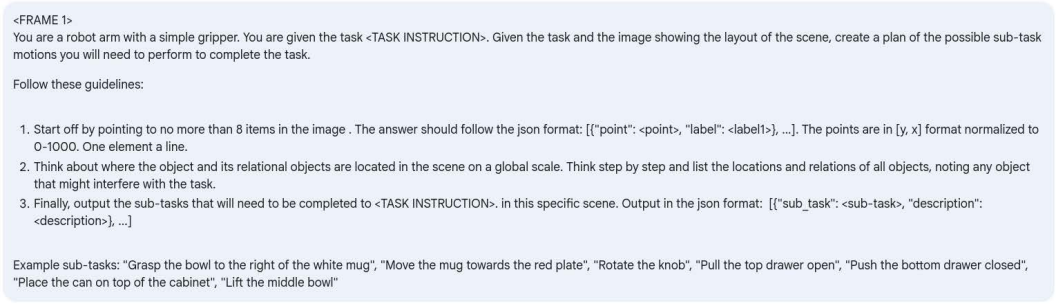

Figure 4: The templated prompt we use for generating semantic sub-tasks. The placeholders `<FRAME 1>` and `<TASK INSTRUCTION>` are replaced with the first image of the trajectory and the trajectory instruction, respectively.

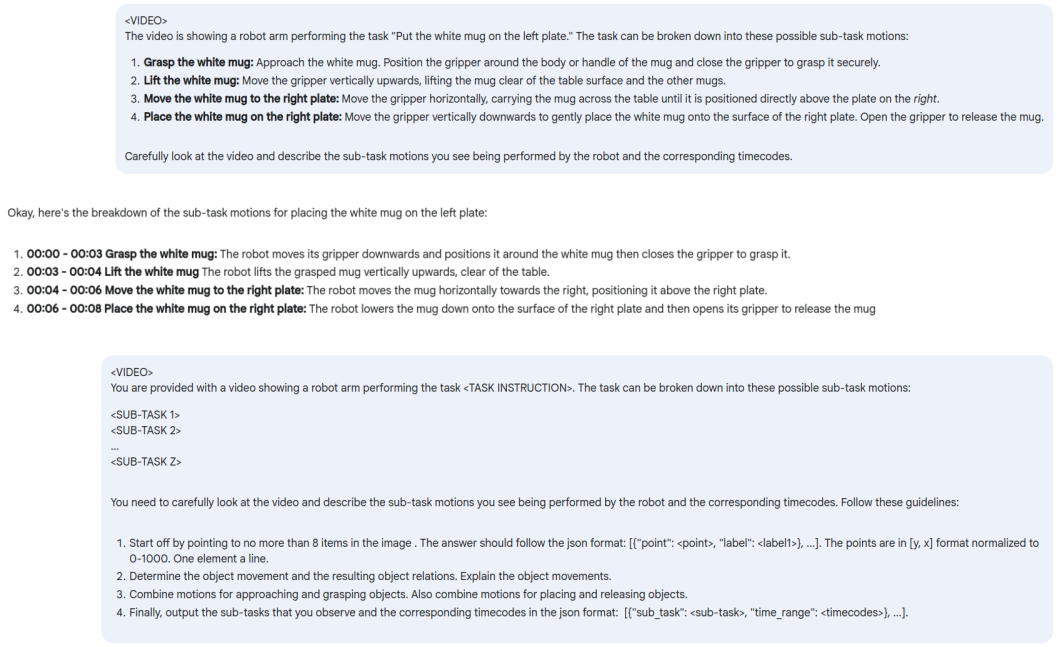

Figure 5: The templated prompt we use for recognizing semantic sub-tasks in a trajectory video. We use few-shot prompting by providing the model with 3 example user-model exchanges before the final query prompt (we include only one example for brevity). The placeholders `<VIDEO>` and `<TASK INSTRUCTION>` are replaced with the trajectory video and the trajectory instruction, respectively.

B Dataset Weighting Ratios

To determine the optimal balance between original demonstrations and augmented sub-trajectories, we conducted ablations across different dataset mixing ratios. Table 2 presents results for Octo, showing success rate performance across three task sets as described in Section 4.2.

<FRAME 1>
You are a robot arm with a simple gripper. You were given the task <TASK INSTRUCTION>. Here is a break down of the sub-task motions you need to perform to complete this task.

<SUB-TASK 1>
<SUB-TASK 2>
...
<SUB-TASK Z>

Help yourself understand the goal by writing 5 alternate descriptions. For the overall task, and each of the sub-task motions write 5 other ways to instruct the same motion.

Follow these guidelines:

1. Start off by pointing to no more than 8 items in the image . The answer should follow the json format: [{"point": <point>, "label": <label1>}, ...]. The points are in [y, x] format normalized to 0-1000. One element a line.
2. Then describe each object both in terms of their properties and their spatial relation to each other and in the global scene (e.g. "the smaller object", "the black object", "the middle object").
3. For the final step rewrite the original task and subtasks using your object descriptions in the json format: [{"task": [<paraphrase-task>,], "subtasks": [[<paraphrase-sub-task>,], [<paraphrase-sub-task>,],]}, ...]

Figure 6: The templated prompt we use for grounded textual diversity (see Section 3.3). The placeholders `<FRAME 1>` and `<TASK INSTRUCTION>` are replaced with the first image of the trajectory and the trajectory instruction, respectively.

Table 2: Success rate performance of Octo [17] finetuned with different dataset weighting ratios for (1) TREAD w/o diverse labels ($\mathcal{D}$:$\tilde{\mathcal{D}}_A$) and (2) TREAD with diverse labels ($\mathcal{D}$:$\tilde{\mathcal{D}}_H$).

Metric	**Metric**	**1:2**		**1:1.5**		**Re-Mix**	
		TREAD w/o div.	TREAD	TREAD w/o div.	TREAD	TREAD w/o div.	TREAD
Language Generalization	Single Goal SR	0.73	0.89	0.51	0.69	0.76	0.73
	1 of 2 SR	0.74	0.79	0.72	0.81	0.72	0.81
	2 of 2 SR	0.30	0.30	0.32	0.33	0.30	0.36
Motion Generalization	Single Goal SR	0.22	0.20	0.27	0.23	0.25	0.17
	1 of 2 SR	0.42	0.40	0.38	0.40	0.49	0.31
	2 of 2 SR	0.00	0.00	0.00	0.00	0.00	0.00
LIBERO-10	1 of 2 SR	0.69	0.77	0.68	0.71	0.74	0.70
	2 of 2 SR	0.37	0.41	0.37	0.37	0.41	0.37

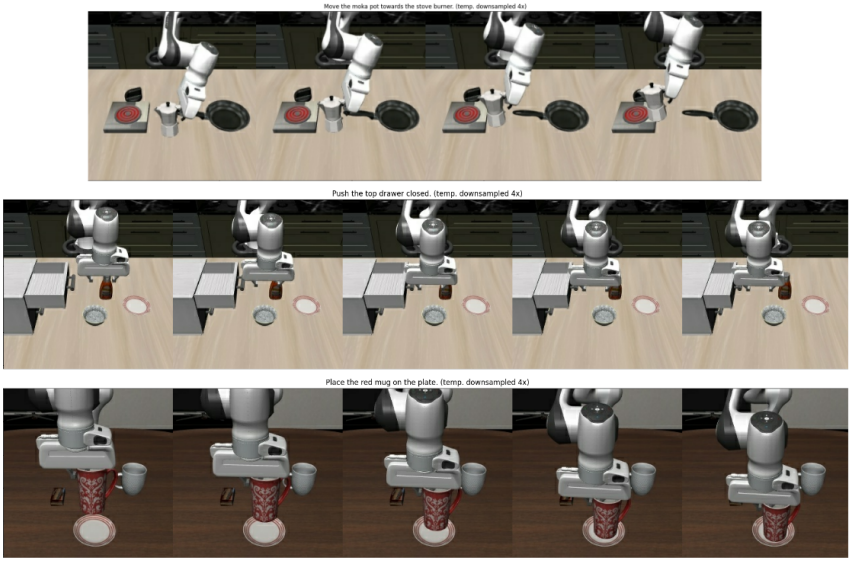

Figure 7: Visualization of some segmented trajectories produced by TREAD. From the top down, the sub-task labels are (1) *"Move the moka pot towards the stove burner"*, (2) *"Push the top drawer closed"*, and (3) *"Place the red mug on the plate."* In visualization (2), we see that some segmentations made by the VLM terminate early.