**Robotics: Science and Systems 2023**
**Daegu, Republic of Korea, July 10-July 14, 2023**

# Robotic Skill Acquisition via Instruction Augmentation with Vision-Language Models

Ted Xiao[1,*] Harris Chan[1,2,*] Pierre Sermanet[1] Ayzaan Wahid[1] Anthony Brohan[1]
Karol Hausman[1] Sergey Levine[1] Jonathan Tompson[1]
[1]Robotics at Google [2]University of Toronto
Project website: https://instructionaugmentation.github.io

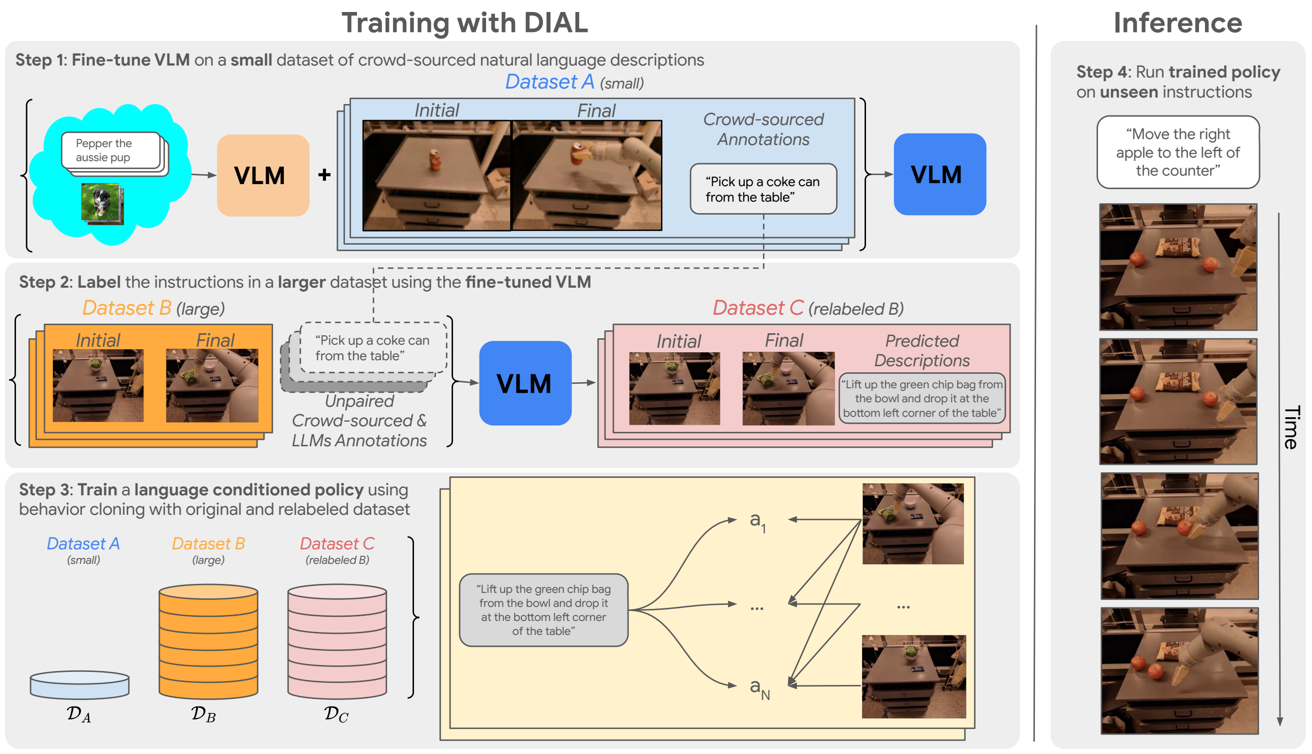

Training with DIAL
Step 1: Fine-tune VLM on a small dataset of crowd-sourced natural language descriptions
Dataset A (small)
Pepper the aussie pup
VLM
Initial
Final
Crowd-sourced Annotations
"Pick up a coke can from the table"
VLM
Step 2: Label the instructions in a larger dataset using the fine-tuned VLM
Dataset B (large)
Initial
Final
"Pick up a coke can from the table"
Unpaired Crowd-sourced & LLMs Annotations
VLM
Dataset C (relabeled B)
Initial
Final
Predicted Descriptions
"Lift up the green chip bag from the bowl and drop it at the bottom left corner of the table"
Step 3: Train a language conditioned policy using behavior cloning with original and relabeled dataset
Dataset A (small)
Dataset B (large)
Dataset C (relabeled B)
$\mathcal{D}_A$
$\mathcal{D}_B$
$\mathcal{D}_C$
"Lift up the green chip bag from the bowl and drop it at the bottom left corner of the table"
$a_1$
...
$a_N$
Inference
Step 4: Run trained policy on unseen instructions
"Move the right apple to the left of the counter"
Time

Fig. 1: DIAL consists of three steps: (1) Contrastive fine-tuning of a vision-language model (VLM) such as CLIP [39] on a small dataset of robot manipulation trajectories with crowd-sourced natural language annotation, (2) labeling a larger dataset of trajectories using the fine-tuned VLM (in dashed outline), and (3) training a language-conditioned policy using behavior cloning on the original and relabeled dataset. We evaluate the trained policy on unseen instructions. See Section III for more details.

***Abstract*—Robotic manipulation policies that follow natural language instructions are typically trained from corpora of robot-language data that were either collected with specific tasks in mind or expensively relabeled by humans with varied language descriptions in hindsight. Recently, large-scale pretrained vision-language models (VLMs) have been applied to robotics for learning representations and scene descriptors. Can these pretrained models serve as automatic labelers for robot data, effectively importing Internet-scale knowledge into existing datasets with limited ground truth annotations? For example, if the original annotations contained templated task descriptions such as "pick apple", a pretrained VLM-based labeler could significantly expand the number of semantic concepts available in the data and introduce spatial concepts such as "the apple on the right side of the table" or alternative phrasings such as "the red colored fruit". To accomplish this, we introduce Data-driven Instruction Augmentation for Language-conditioned control (DIAL): we utilize semi-supervised language labels to propagate CLIP's semantic knowledge onto large datasets of unlabeled demonstration data, from which we then train language-conditioned policies. This method enables cheaper acquisition of useful language descriptions compared to expensive human labels, allowing for more efficient label coverage of large-scale datasets. We apply DIAL to a challenging real-world robotic manipulation domain where only 3.5% of the 80,000 demonstrations contain crowd-sourced language annotations. Through a large-scale study of over 1,300 real world evaluations, we find that DIAL enables imitation learning policies to acquire new capabilities and generalize to 60 novel instructions unseen in the original dataset.**

[*] Equal contribution

## I. Introduction

Advances in deep learning architectures have made it possible to train end-to-end robotic control policies for following a wide range of textual instructions, often by integrating state-of-the-art language embeddings and pretrained encoders with imitation learning on top of large, manually collected datasets of robotic demonstrations annotated with text commands [24]. However the performance of such methods is critically dependent on the quantity and breadth of instruction-labeled demonstration data that is available [32], and producing expert demonstrations of robot motion often requires expertise and time [33]. Can we squeeze more generalization capacity out of a given set of demonstrations, with minimal additional human effort? The key observation we make is that a given demonstration might illustrate more than one behavior – e.g., a motion that picks up the leftmost can in Figure 1 is an example of how to pick up a coke can, the left can in a row of three, a first step toward clearing the table, and more. Can we leverage this observation to relabel a given demonstration dataset to enable a robot to master a broader range of semantic behaviors, and can we do this in a largely automated and scalable way?

One possibility is to leverage large-scale pretrained language models (LLMs) [9, 15] and vision-language models (VLMs) [2, 39], which can be pretrained on Internet-scale data and then be applied to downstream domains. In robotics, they have been used as representations for perception [37, 42], as task representation for language [24, 30], or as planners [1, 22]. In contrast, we seek to apply pretrained VLMs to the datasets themselves: can we use VLMs for *instruction augmentation*, where we relabel existing offline trajectory datasets with additional language instructions?

In this work, we introduce **D**ata-driven **I**nstruction **A**ugmentation for **L**anguage-conditioned Control (DIAL), a method that performs instruction augmentation with pretrained VLMs to weakly relabel offline control datasets. We implement an instantiation of our method with CLIP [39] on a challenging real-world robotic manipulation setting with 80,000 teleoperated demonstrations, which include 2,800 demonstrations that are labeled by crowd-sourced language annotators. By performing a large quantitative evaluation of over 1,300 real world robot evaluations, we compare our method with baselines and instruction augmentation methods that are not visually grounded. We find that DIAL enables policies to acquire understanding of new concepts not contained in the original task labels and improving performance on 60 novel evaluation instructions by over 41%. Sample emergent capabilities of our method are shown in Figure 5.

## II. Related Work

*a) Language instruction following in robotics:* Language-instruction following agents have been extensively explored with engineered symbolic representations [17, 45], with reinforcement learning (RL) [5, 20, 29], and with imitation learning [3, 6, 24, 30]. Recent advances in deep learning with large amounts of data have led to advances in methods for learning instruction-conditioned policies [1, 28, 34, 43, 44]. Latent Motor Policies (LMP) [31] learns hierarchical goal-conditioned policies. Subsequent Language from Play (LfP) [30] uses language goals provided by large dataset of hindsight human labels on robotic play data. Similarly, Interactive Language [32] uses crowd-sourced hindsight labels on diverse demonstration data for table-top object rearrangements. In contrast, our method does not rely on crowd-sourced language labels at scale, but instead leverages a modest number of language labels by using a learned model to provide weak hindsight labeling for the rest of the data.

*b) Pretrained VLMs and LLMs for language-conditioned control:* Prior works have leveraged pretrained VLMs and LLMs for language-conditioned control, as part of reward modeling [18, 36], as part of the agent architecture [37, 42], or as planners for long-horizon tasks [1, 22, 23]. MineCLIP [18] fine-tunes CLIP [39] encoders using a contrastive loss on a large offline dataset of Minecraft videos and optimizes a language-conditioned control policy on top of the finetuned CLIP representations through online RL. LOReL [36] learns a reward function from offline robot datasets with crowd sourced annotations using a neural network trained from scratch combined with a pretraind DistilBERT sentence embedding [41] using a binary cross entropy loss. CLIPort [42] uses a frozen CLIP vision and text encoders in combination with Transporter networks [47] for imitation learning. R3M [37] uses representations pretrained constrastively on Ego4D [21] human video datasets for robotic policy learning via imitation learning. For long-horizon language instructions, LLMs have been used as planners both in simulated [22] and real-world robotics settings [1]. Our approach fine-tunes CLIP on our *real* robot offline dataset and is used for instruction augmentation for a behavior cloning agent, instead of directly using the CLIP model as a reward model and optimizing an RL agent.

*c) Hindsight relabeling for goal-conditioned reinforcement learning:* The relabeling approach for goal-conditioned reinforcement learning [38] is used in tabular [26] and continuous [4] settings, where the desired goals are relabeled with achieved goals to generate positive examples in sparse reward environments. This method has been applied to environments with goals represented as images [12], task IDs [27], and language instructions [11, 13, 25]. Previous works with language goals used environment simulators [11, 25] or learned models [13, 40] to provide hindsight labels. Our work introduces the novel contribution of visual grounding by leveraging VLMs to generate unstructured natural language relabeling instructions, enabling scaling to complex real robot environments.

## III. Data-driven Instruction Augmentation for Language-conditioned Control

In this section, we describe our method, DIAL, which consists of three stages: (1) fine-tuning a VLM's vision and language representations on a small offline dataset of trajectories with crowd-sourced episode-level natural language descriptions, (2) generating alternative instructions for a larger offline dataset of trajectories with the VLM, and (3) learning a language-conditioned policy via behavioral cloning on this instruction-augmented dataset.

### A. Fine-tuning Vision-Language Model Representations

We first collect a dataset of robot trajectories, from either human teleoperated demonstrations on a wide variety of tasks

[1], or from unstructured robotic "play" data [31]. We partition this dataset with uniform sampling into two subsets: a small subset to be annotated by human annotators, and a much larger subset to be labeled by the VLM finetuned on the former. The smaller portion is selected because the process of human labeling is time-consuming and requires significant effort and cost.

Let the small dataset of $N$ trajectories be $[\tau_1, \ldots, \tau_N]$, $\tau_n = ([(s_0^n, a_0^n), (s_1^n, a_1^n), \ldots, (s_T^n)])$, where $s_t^n$ and $a_t^n$ denote the observed state and action, respectively, at time $t$ for the $n$-th episode. We then collect a corresponding natural language annotation $l^n$ for the $n$-th episode describing what the robot agent did in the episode via crowd-sourcing. When producing these descriptions, the crowd-sourced evaluators observe the first frame, $s_0$, and last frame, $s_T$, from the agent's first-person view. We refer to these instructions as *crowd-sourced instructions*. Together, we denote the first dataset $\mathcal{D}_A = [(\tau_1, l^1), \ldots, (\tau_N, l^N)]$ as the paired trajectories and crowd-sourced labels. Our method then fine-tunes a vision and language model representation on $\mathcal{D}_A$.

Motivated by promising results of CLIP in robotics in prior works [35, 42], our instantiation of DIAL uses CLIP [39] for both instruction augmentation and task representation; nonetheless, other VLMs or captioning models could also be used to propose instruction augmentations. Given a batch of $B$ initial state $s_0$, final state $s_T$, and crowd-sourced instruction $l$ tuple, the model is trained to predict which of the $B^2$ (initial-final state, crowd-sourced instruction) pairs co-occurred. We use CLIP's Transformer-based text encoder $T_{enc}$ to embed the crowd-sourced instruction to a latent space $z_l^n = T_{enc}(l^n)/\|T_{enc}(l^n)\| \in \mathbb{R}^d$ and CLIP's Vision Transformer-based (ViT) [16] image encoder $I_{enc}$ to embed the initial and final state, and further concatenate these two embeddings and pass through a fully connected neural network $f_\theta$, producing the vision embedding $z_s^n = f_\theta([I_{enc}(s_0^n); I_{enc}(s_T^n)])/\|f_\theta([I_{enc}(s_0^n); I_{enc}(s_T^n)])\| \in \mathbb{R}^d$. $B^2$ similarity logits are formed by applying dot product across all state-instruction pairs, and a symmetric cross entropy loss term is calculated by applying softmax normalization with temperature $\alpha$ across the states and texts:

$$\mathcal{L}_\theta = -\sum_{n=1}^{B}\left[\log\left(\frac{e^{z_l^n \cdot z_s^n/\alpha}}{\sum_{k=1}^{B} e^{z_l^k \cdot z_s^n/\alpha}}\right) + \log\left(\frac{e^{z_l^n \cdot z_s^n/\alpha}}{\sum_{k=1}^{B} e^{z_l^n \cdot z_s^k/\alpha}}\right)\right]$$

## B. Instruction Augmentation

In the larger partition of the original dataset, which we denote as dataset $\mathcal{D}_B$, contains $M \gg N$ trajectories $[\hat{\tau}_1, \ldots, \hat{\tau}_M]$, where $\hat{\tau}_m = ([(\hat{s}_0^m, \hat{a}_0^m), (\hat{s}_1^m, \hat{a}_1^m), \ldots, (\hat{s}_T^m)])$. In contrast to $\mathcal{D}_A$, we assume that trajectories in $\mathcal{D}_B$ do not have any associated natural language labels.

We use the fine-tuned VLM model to propose natural language instructions $\tilde{l}^m$ for each trajectory $\hat{\tau}_m$ to augment $\mathcal{D}_B$. While $\tilde{l}^m$ could be drawn from any reasonable corpus, our specific instantiation of DIAL sources these candidate instructions from $\mathcal{D}_A$ as well as additional instructions drawing from GPT-3 [9] proposals of possible tasks, which we denote

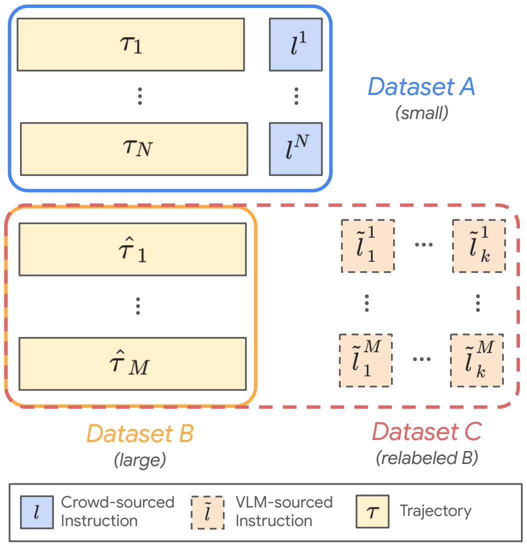


Fig. 3: The construction of datasets: Dataset A ($\mathcal{D}_A$) (blue) consists of the $N$ trajectories $\{\tau_n\}_{n=1}^N$ labeled with crowd-sourced instructions $\{l^n\}_{n=1}^N$ describing what the robot agent performed in the episode. Dataset B ($\mathcal{D}_B$) (yellow) consists of a much larger set of trajectories, $\{\hat{\tau}_m\}_{m=1}^M$ *without* crowd-sourced instructions. Dataset C ($\mathcal{D}_C$) (red, dashed) contains Dataset B trajectories relabeled with VLM-sourced hindsight instruction(s) $\{\tilde{l}_1^m, \ldots, \tilde{l}_k^m\}_{m=1}^M$.

as $\mathcal{D}_{GPT-3}$ (the details of this procedure will be covered in Section IV-B). We use the CLIP text encoder to independently embed these candidate natural language instructions, i.e. $\tilde{l}^m \in L = \{l^1, \ldots, l^N\} \sim \mathcal{D}_A \cup \mathcal{D}_{GPT-3}$:

$$\{z_l^1, \ldots, z_l^N\} = \{T_{enc}(l^1), \ldots, T_{enc}(l^N)\}$$

Similarly, we use the fine-tuned CLIP image encoder and MLP fusion to embed the initial and final observations from the second dataset:

$$\{\hat{z}_s^1, \ldots, \hat{z}_s^M\} = \{f_\theta([I_{enc}(\hat{s}_0^i); I_{enc}(\hat{s}_T^i)])\}_{i=1}^M$$

With these embeddings pre-computed, we can retrieve the most likely candidates using $k$-Nearest Neighbors [19] with cosine similarity between the vision-language embedding pairs $d(z_l^n, \hat{z}_s^m) = \frac{z_l^n \cdot \hat{z}_s^m}{\|z_l^n \cdot \hat{z}_s^m\|}$ as the metric. We then use the cosine similarity to select a subset of candidate instructions to construct a new *relabeled* dataset $\mathcal{D}_C = [(\hat{\tau}_1, \tilde{l}_1^1), \ldots, (\hat{\tau}_1, \tilde{l}_k^1), \ldots, (\hat{\tau}_M, \tilde{l}_1^M), \ldots, (\hat{\tau}_M, \tilde{l}_k^M)]$. Figure 3 visualizes the three datasets generated. There are several potential strategies for candidate instruction selection:

*a) Top-$k$ selection:* For each trajectory, we rank the candidate instructions in descending order based on their cosine similarity distances and output the top-$k$ instructions. The hyperparameter $k$ trades off precision and recall of the relabeled dataset. A smaller $k$ will return mostly relevant candidate instructions, while a larger $k$ value can recall a broader spectrum of potential hindsight descriptions for the episode at the expense of introducing erroneous instructions.

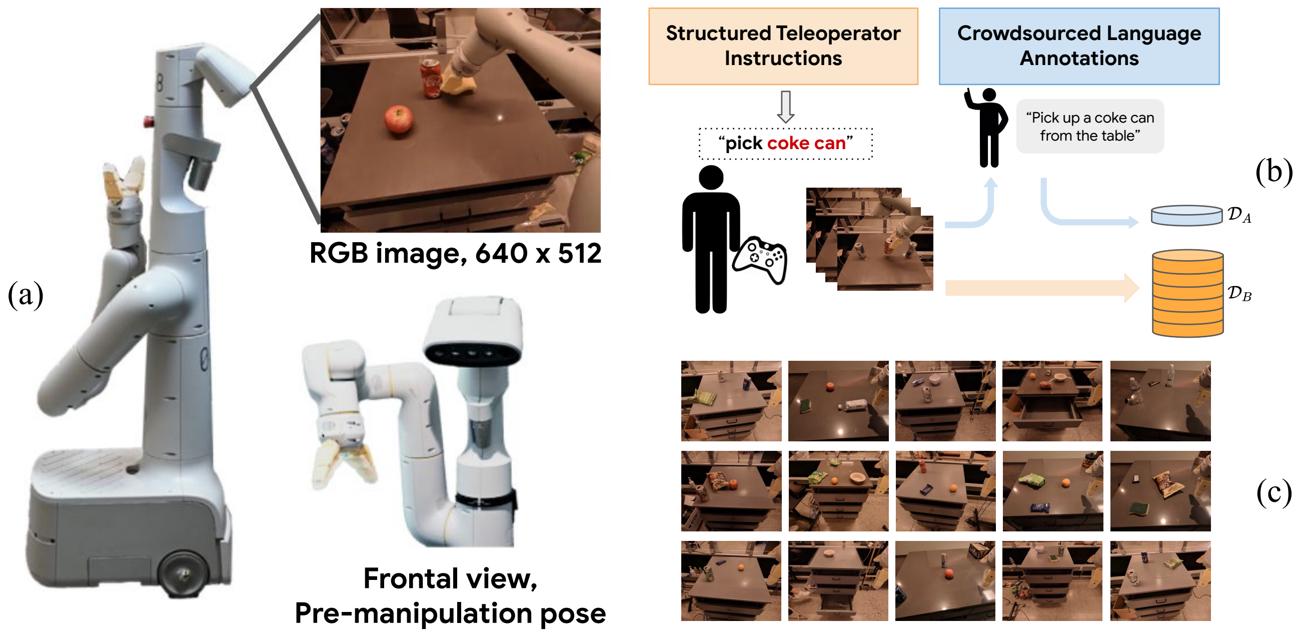


Fig. 4: (a) A mobile manipulator robot receives RGB images from an onboard camera and uses a 7 DoF arm with parallel-jaw grippers. (b) Teleoperators receive instructions drawn from a set of 551 structured commands to perform a total of 80,000 demonstrations. 2,800 of these episodes are sent for crowdsourced language annotations. (c) A sample of scenes in the demonstration dataset which range across various countertops, drawers, and object arrangements in an office kitchen setting.

*b) Min-p selection:* Instead of outputting a fixed number of candidate instructions per trajectory, we dynamically adjust this number based on a minimum probability $p$ parameter, representing the minimum confidence for each instruction. We first convert the cosine similarity between the vision-language embedding pair to a probability that the $m$-th episode has language label $l^n$ by taking the softmax over all the candidate instructions with temperature parameter $\alpha$ from CLIP:

$$P(\tilde{l}^m = l^n | (\hat{s}_0^m, \hat{s}_T^m)) = \frac{\exp(d(z_l^n, \hat{z}_s^m)/\alpha)}{\sum_{n'} \exp(d(z_l^{n'}, \hat{z}_s^m)/\alpha)} \quad (1)$$

We truncate the candidate instructions to the set $L^{(p)} \subset L$ such that each instruction has a minimum hurdle probability $p > 0$:

$$P(\tilde{l}^m = l) \geq p, \quad \forall\ l \in L^{(p)} \quad (2)$$

Given $p$, the *maximum* number of candidates that can be output for a trajectory is $k = 1/p$. The *minimum* number of candidates, meanwhile, can be zero, if there are no candidate instructions satisfying this hurdle probability.

We will investigate in Section V-C the effects of these candidate instruction selection strategies on relabeled instruction accuracy, augmented dataset size, and downstream policy performance.

### C. Language Conditioned Policies with Behaviour Cloning

Given a dataset $\mathcal{D} = [\mathcal{D}_A, \mathcal{D}_C]$ of robot trajectories and corresponding augmented language instructions, we can train a language-conditioned control policy with Behavior Cloning (BC). While instruction augmented offline datasets can be used by any downstream language-conditioned policy learning method such as offline RL or BC, we limit our work to the conceptually simpler BC in order to focus our analysis on the importance of instruction augmentation.

## IV. Experimental Setup

We first describe the setup for our experimental validation, including the environments, the configuration of the robot, and the datasets that we use in our experiments. Since our aim is to study the benefits of our proposed instruction augmentation approach, we describe other alternative methods for augmenting the dataset for comparison. Finally, we detail our evaluation protocol, which involves testing the degree to which policies learned with different types of instruction augmentation generalize to *novel previously unseen* instructions.

### A. Environment, Robot, and Datasets

We implement DIAL in a challenging real-world robotic manipulation setting based on the kitchen environments described by Ahn et al. [1]. We focus on the practically-motivated setting where a dataset of teleoperated demonstrations is available, collected for downstream imitation learning [1, 24]. A mobile manipulator robot with a parallel-jaw gripper, an over-the-shoulder RGB camera, and a 7 DoF arm is placed in an office kitchen to interact with common objects using concurrent [46] continuous closed-loop control from pixels. We collect a large-scale dataset of over 80,000 robot trajectories via human teleoperation ($\mathcal{D}_B$), where teleoperators receive 551 structured commands motivated by common manipulation skills and objects in a kitchen environment, following prior work [1]. Afterwards, we leverage crowd-sourced human annotators to label 2,800 robot trajectories with two hindsight instructions each, resulting in a total of 5,600 unique episodes with crowdsourced captions ($\mathcal{D}_A$). Human annotators are shown the first and last frame of the episode and asked to provide a free-form text description describing how a robot should be commanded to go from the start to the end. Visualizations of the mobile manipulator, dataset collection procedure, and example scenes are shown in Figure 4 and further detailed in Appendix B.

## B. Instruction Augmentation

We consider various methods of instruction augmentation which each result in different relabeled datasets that are then used for downstream policy learning.

*a) DIAL implementations:* We implement DIAL with a CLIP model that is fine-tuned on $\mathcal{D}_A$ with the procedure described in Section III-A. After fine-tuning CLIP, we source 18,719 candidate instruction labels ($L$) from the combination of $\mathcal{D}_A$ and a corpus of GPT-3 proposals of potential language instructions. To perform instruction augmentation that relabels dataset $\mathcal{D}_B$ of 80,000 robot trajectories that do not contain crowd-sourced annotations with $L$, we follow Section III-B to implement two variations of DIAL: **Top-$k$ selection** and **Min-$p$ selection**.

The version of DIAL with **Top-$k$ selection** applies a fixed number $k$ instruction augmentations for every episode in the source dataset based on cosine similarity distances. By changing $k$, we produce three instruction augmented datasets: 80,000 relabeled demonstrations ($k$ = 1), 240,000 relabeled demonstrations ($k$ = 3), and 800,000 relabeled demonstrations ($k$ = 10). The version of DIAL with **Min-$p$ selection** is more conservative and only performs instruction augmentation when confidence from CLIP is above some threshold $p$. By changing $p$, we produce three instruction augmented datasets: 128,422 relabeled demonstrations ($p$ = 0.1), 38,516 relabeled demonstrations ($p$ = 0.2), and 17,013 relabeled demonstrations ($p$ = 0.3). Additional details can be found in Appendix C.

*b) Non-visual instruction augmentation methods:* We consider three instruction augmentation methods that do *not* utilize any visual information. First, we implement a "Gaussian Noise" baseline that adds random noise to existing crowd-sourced instructions' language embeddings. Second, we design a "Word-level Synonyms" baseline that replaces individual words in existing instructions with sampled synonyms from a predefined list. Finally, we introduce a "LLM-proposed Instructions" baseline that replaces entire instructions with alternative instructions as proposed by GPT-3. Implementation details for these baselines can be found in Appendix E.

## C. Policy Training

Using these various instruction augmented datasets, we train vision-based language-conditioned behavioral cloning policies with the RT-1 architecture [7]. One main difference from RT-1 is that instead of utilizing USE [10] as the task representation for language conditioning, we instead use the language encoder of the fine-tuned CLIP model that was used for instruction augmentation in Section III-B; full details are described further in Appendix G. Nonetheless, we treat the behavioral cloning policy as an independent component of our method and focus on studying instruction augmentation methods; we do not explore different policy architectures or losses in this work.

| **Category** | **Instruction Samples** |
|---|---|
| *Spatial* | `['knock down the right soda', 'raise the left most can', 'raise bottle which is to the left of the can']` |
| *Rephrased* | `['pick up the apple fruit', 'liftt the fruit' [sic], 'lift the yellow rectangle']` |
| *Semantic* | `['move the lonely object to the others', 'push blue chip bag to the left side of the table', 'move the green bag away from the others']` |

TABLE I: Samples from the 60 novel evaluation instructions we consider. 34 *Spatial* tasks focus on instructions involving reasoning about spatial relationships, such as specifying an object's initial position relative to other objects in the scene. 16 *Rephrased* tasks are linguistic re-phrasings of the original 551 foresight tasks, such as referring to sodas and chips by their colors instead of their brand name. 10 *Semantic* tasks describe skills not contained in the original dataset, such as moving objects away from all other objects, since the original dataset only contains trajectories of moving objects towards other objects. A full list is provided in Table VII.

## D. Evaluation

In contrast to many prior works [3, 6] on instruction following, we focus our evaluation only on *novel instructions unseen during training*. To source these novel instructions, we crowd-source instructions and prompt GPT-3 for evaluation task suggestions, and then filter out any instructions already contained in either the crowd-sourced language instructions in $\mathcal{D}_A$, the original set of 551 structured teleoperator commands $\mathcal{D}_B$, or the instruction augmentated dataset $\mathcal{D}_C$; in total, we sample 60 novel evaluation instructions. We organize these evaluation instructions into three categories to allow for more detailed analysis of qualitative policy performance; examples are shown in Table I and a full list is provided in Table VII.

## V. Experimental Results

In our experiments, we investigate whether DIAL can improve the policy performance on the unseen tasks described in Section IV-D when starting from fully or partially labeled source datasets. We ablate on the types of instruction augmentations described in Section IV-B, and analyze the importance of accuracy when augmenting instructions with DIAL.

## A. Does DIAL improve performance on unseen tasks?

We investigate whether DIAL can enable language-conditioned behavior cloning policies to successfully perform

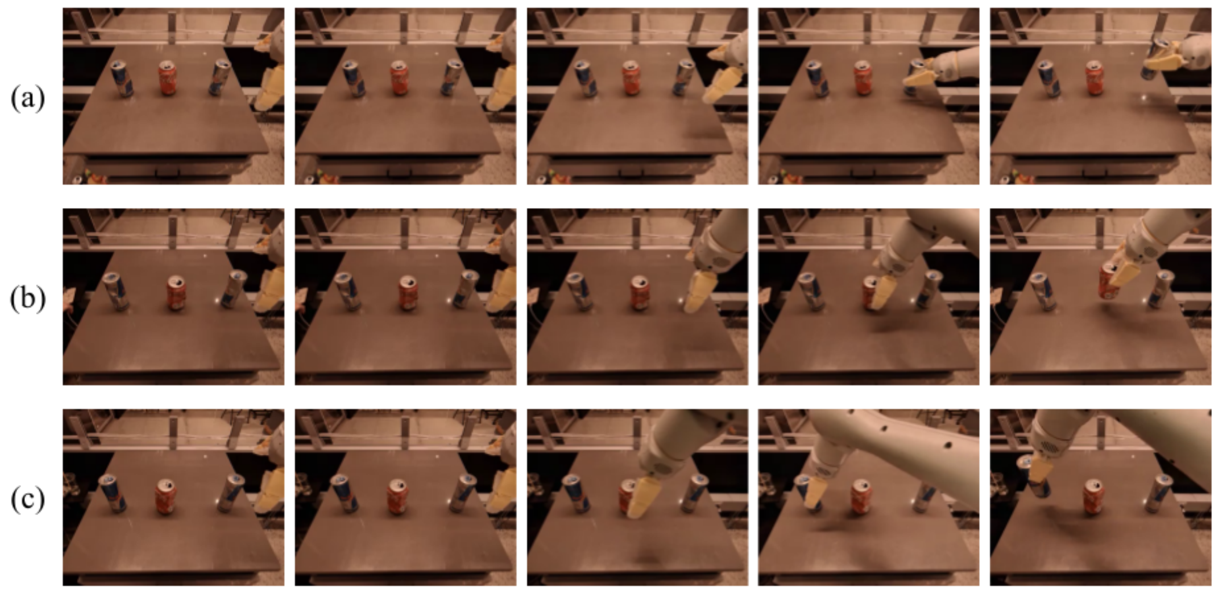


Fig. 5: Given the same starting scene, DIAL follows the instructions of (a) `pick can which is on the right of the table`, (b) `pick the can in the middle`, and (c) `pick can which is on the left of the table`. Non-DIAL methods do not adjust their behaviors based on language commands, demonstrating a lack of spatial understanding.

novel instructions. For training various control policies, we consider three training datasets: $\mathcal{D}_A$ contains 5,600 episodes which contain crowd-sourced hindsight language instructions, $\mathcal{D}_B$ contains 80,000 episodes which contain structured commands given to teleoperators, and $\mathcal{D}_C$ contains 38,516 episodes with instructions predicted by DIAL with Min-$p = 0.2$ starting from $\mathcal{D}_A$ and $\mathcal{D}_B$. We refer to training on only $\mathcal{D}_A$ as the Interactive Language (IL) [32] setting, training on only $\mathcal{D}_B$ as the RT-1 [7] setting, and training on both $\mathcal{D}_A$ and $\mathcal{D}_B$ as the RT-1 + IL setting. Then, we refer training on $\mathcal{D}_C$ (either with or without $\mathcal{D}_A$ and $\mathcal{D}_B$) as DIAL. This experiment is practically motivated by the setting where large amounts of unstructured trajectory data are available but hindsight labels are expensive to collect, such as robot play data [14, 31, 32].

After policy training, we evaluate on task instructions not contained in $\mathcal{D}_A$, $\mathcal{D}_B$, or $\mathcal{D}_C$. Table II demonstrates that DIAL is able to solve over 40% more challenging novel tasks across the three evaluation categories compared to either RT-1 and/or IL, which do not use the instruction augmented data $\mathcal{D}_C$.

An example is shown in Figure 5, where DIAL successfully understands the spatial concepts of “left”, “middle”, and “right”. Such spatial concepts are especially important to identify object instances in scenes with duplicate objects: while baseline methods ignore the language instruction and instead repeat the same motions or randomly select a target object, DIAL is able to consistently target the correct objects. In addition to *Spatial* tasks, DIAL is also able to outperform baseline policies at *Semantic* tasks that focus on semantic skills not contained in the original foresight instructions. We show more examples of evaluation successes in Figure 8.

| | **Dataset Properties** | | | **Evaluation on Novel Instructions** | | | |
|---|---|---|---|---|---|---|---|
| Method | $\mathcal{D}_A$ | $\mathcal{D}_B$ | $\mathcal{D}_C$ | *Spatial* | *Rephrased* | *Semantic* | Overall |
| IL [32] | ✓ | | | 30.0% | 40.0% | 7.7% | 27.5% |
| RT-1 [7] | | ✓ | | 38.0% | 40.0% | 15.4% | 33.8% |
| RT-1 + IL | ✓ | ✓ | | 46.0% | 60.0% | 15.4% | 42.5% |
| | ✓ | | ✓ | 58.0% | 46.7% | 15.4% | 47.5% |
| **DIAL** (ours) | | ✓ | ✓ | 50.0% | 37.5% | 10.0% | 36.7% |
| | ✓ | ✓ | ✓ | **68.0%** | **66.7%** | **30.8%** | **60.0%** |

TABLE II: Comparing the performance of language-conditioned policies trained on different types of labeled datasets. $\mathcal{D}_A$ contains 5,600 episodes with crowd-sourced language instructions and is representative of the Interactive Language (IL) [32] setting. $\mathcal{D}_B$ contains 80,000 episodes with structured teleoperator commands and is representative of the RT-1 [7] setting. DIAL additionally creates an augmented $\mathcal{D}_C$ with 38,516 relabeled instructions. We find that DIAL is able to significantly performance on novel evaluation instructions, especially in the IL setting where $\mathcal{D}_B$ is not available.

## *B. How does DIAL compare to other instruction augmentation methods?*

We compare DIAL to non-visual instruction augmentation strategies outlined in Section IV-B. For this comparison, we apply the baseline instruction augmentation methods on both episodes with crowd-sourced annotations ($\mathcal{D}_A$) and on episodees with structured teleoperator commands ($\mathcal{D}_B$) to produce different instruction aug-

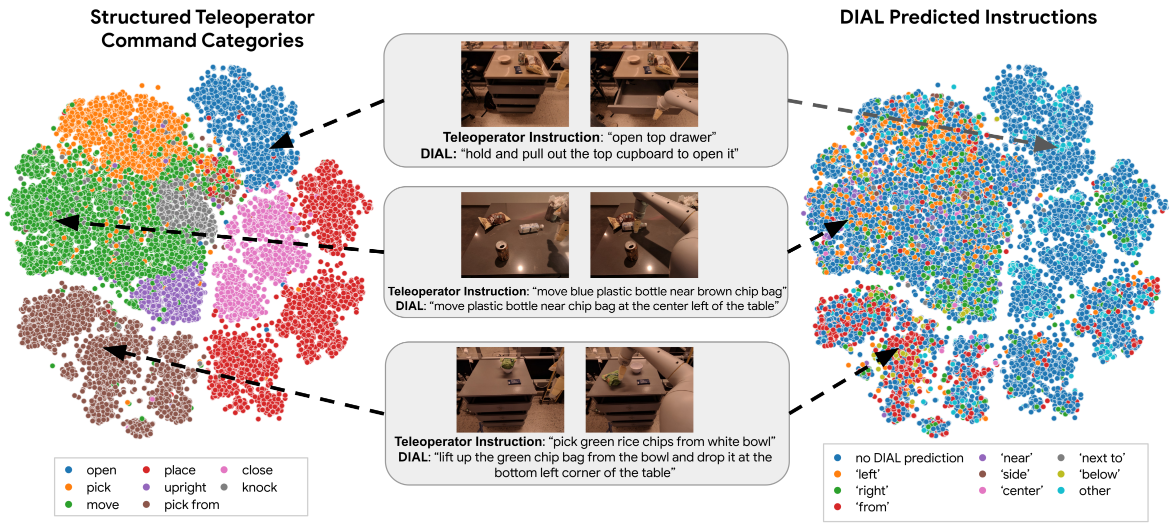


Fig. 6: Comparing the task diversity of structured commands provided to teleoperators and DIAL instruction predictions. The visualization shows a t-SNE for 30,000 trajectories with their first and last frames embedded via a fine-tuned CLIP model. On the left, embeddings are colored based on the skill categories of structured teleoperator commands. On the right, embeddings are colored based on particular keywords that are present in the DIAL predicted instructions. While large clusters of episodes may all correspond to the same teleoperator command, DIAL predictions may highlight more nuanced semantic concepts.

| | **Evaluation on Novel Instructions** | | | |
|---|---|---|---|---|
| **Instruction Augmentation** | *Spatial* Tasks | *Rephrased* Tasks | *Semantic* Tasks | Overall |
| None | 46.0% | 60.0% | 15.4% | 42.5% |
| Gaussian Noise | 36.0% | 40.0% | 23.1% | 33.8% |
| Word-level Synonyms | 28.0% | 46.7% | 7.7% | 27.5% |
| LLM-proposed Instructions | 28.0% | 46.7% | 23.1% | 30.0% |
| **DIAL** (ours) | **68.0%** | **66.7%** | **30.8%** | **60.0%** |

TABLE III: Evaluating language-conditioned BC policies trained on datasets with different types of instruction augmentation. Each policy performs 80 evaluations over 60 novel task instructions. DIAL is consistently most performant, especially on *Spatial* Tasks requiring visual scene understanding.

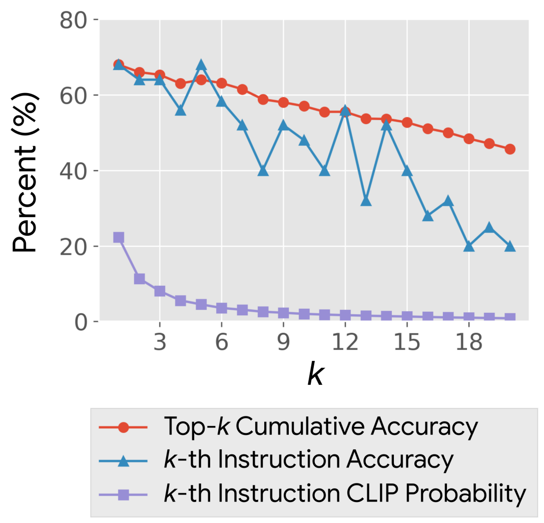


Fig. 7: The factual accuracy of the top 20 instruction augmentation predictions of 50 sampled episodes relabeled by a fine-tuned CLIP model. For $k \in [1, 20]$, we measure the accuracy of the $k$-th instruction, the cumulative accuracy of the top $k$ instructions, and CLIP's confidence score for the $k$-th instruction. While top instructions are often accurate, they become increasingly inaccurate along with CLIP's confidence.

mented datasets ($\mathcal{D}_C^{Gaussian}, \mathcal{D}_C^{Synonyms}, \mathcal{D}_C^{LLM}, \mathcal{D}_C^{DIAL}$). Afterwards, we train separate language-conditioned policies policies: the "None" model trains on $\{\mathcal{D}_A, \mathcal{D}_B\}$, while the remaining models train on $\{\mathcal{D}_A, \mathcal{D}_B, \mathcal{D}_{C'}\}$ with $\mathcal{D}_{C'}$ being each of their respective instruction augmented datasets. Table III indicates that DIAL significantly outperforms other baseline instruction augmentation methods. For both the *Spatial* Tasks and *Rephrased* tasks, we observe that these baseline instruction augmentation methods without visual grounding resulted in worse performance compared to the no instruction augmentation.

### *C. How sensitive is DIAL to hyperparameters and instruction prediction accuracy?*

We study the trade-off between increasing the amount of instruction augmentation and potentially relabeling with incorrect or irrelevant instructions. By varying the hyperparameters of Top-$k$ prediction and Min-$p$ prediction, the two instruction prediction variations of DIAL discussed in Section IV-B, we can indirectly influence the size the potential label inaccuracy of the relabeled datasets. To measure how instruction augmentation accuracy changes as we increase $k$,

| DIAL Version | Dataset Properties | | Evaluation on Novel Instructions | | | |
|---|---|---|---|---|---|---|
| Prediction Method | Relabeled Episodes | Relabeled Accuracy | *Spatial* Tasks | *Rephrased* Tasks | *Semantic* Tasks | Overall |
| Top-$k$, $k = 1$ | $80,000$ | $68.0\%$ | $62.0\%$ | $40.0\%$ | $23.1\%$ | $50.0\%$ |
| Top-$k$, $k = 3$ | $240,000$ | $65.3\%$ | $62.0\%$ | $40.0\%$ | $15.4\%$ | $48.8\%$ |
| Top-$k$, $k = 10$ | $800,000$ | $57.0\%$ | $37.5\%$ | $50.0\%$ | $20.0\%$ | $35.0\%$ |
| Min-$p$, $p = 0.10$ | $128,422$ | $61.9\%$ | $44.0\%$ | $46.7\%$ | $23.1\%$ | $40.0\%$ |
| Min-$p$, $p = 0.20$ | $38,516$ | $68.8\%$ | $\mathbf{68.0}\%$ | $\mathbf{66.7}\%$ | $\mathbf{30.8}\%$ | $\mathbf{60.0}\%$ |
| Min-$p$, $p = 0.30$ | $17,013$ | $76.0\%$ | $62.0\%$ | $53.3\%$ | $46.2\%$ | $56.3\%$ |

TABLE IV: Comparing DIAL with Top-$k$ prediction against DIAL with Min-$p$ prediction. By increasing $k$ or decreasing $p$, augmented datasets become larger but increasingly inaccurate. We provide analysis of the relationship between instruction accuracy and CLIP confidence in Figure 7 and Section V-C.

we ask human labelers to rate whether proposed instruction augmentation are factually accurate descriptions of a given episode. We show an example of the top 10 predicted instruction augmentations in an episode in Figure 12.

In Figure 7, we sample 50 episodes and ask human labelers to assess the predicted instruction accuracy as we increase the number of predictions produced by CLIP. While the initial predictions are often correct, the later predictions are often factually inaccurate. The top-20th instruction prediction is only factually accurate $20.0\%$ of the time.

When applying these different relabeled datasets to downstream policy learning, we find in Table IV that Min-$p$ instruction prediction, a more conservative approach than Top-$k$ prediction, performs the best across all evaluation instructions. We also find that finetuning CLIP is quite important for instruction augmentation, which is detailed in Appendix A2.

## VI. Conclusion, Limitations, and Future Work

In this work, we introduced DIAL, a method that uses VLMs to label offline datasets for language-conditioned policy learning. Scaling DIAL to a real world robotic manipulation domain, we perform a large-scale study of over 1,300 evaluations and find that DIAL is able to outperform baselines by $40\%$ on a challenging set of 60 novel evaluation instructions unseen during training. We compare DIAL against instruction augmentation methods that do not consider visual context, and also ablate the source datasets we use for instruction augmentation. Finally, we study the interplay between larger augmented datasets and lowered instruction accuracy; we find that control policies are able to utilize relabeled demonstrations even when some labels are inaccurate, suggesting that DIAL is able to provide a cheap and automated option to extract additional semantic knowledge from offline control datasets.

*Limitations and Future Work:* Although DIAL seems to improve policy understanding on many novel concepts not contained in the original training dataset, it sometimes fails, especially when evaluating tasks that may require new motor skills. cIn addition, since both our crowd-source annotators and VLMs only have access to the first and final states of an episode, they do not capture skills involving temporal coherence nor is aware of *how* these instructions are accomplished. A natural next step is to apply DIAL to full video episodes. Another interesting direction is to view DIAL as goal-conditioning and attempting visual goals during training or evaluation. Moreover, on-policy or RL variations of DIAL may be able to effectively explore the task representation space autonomously.

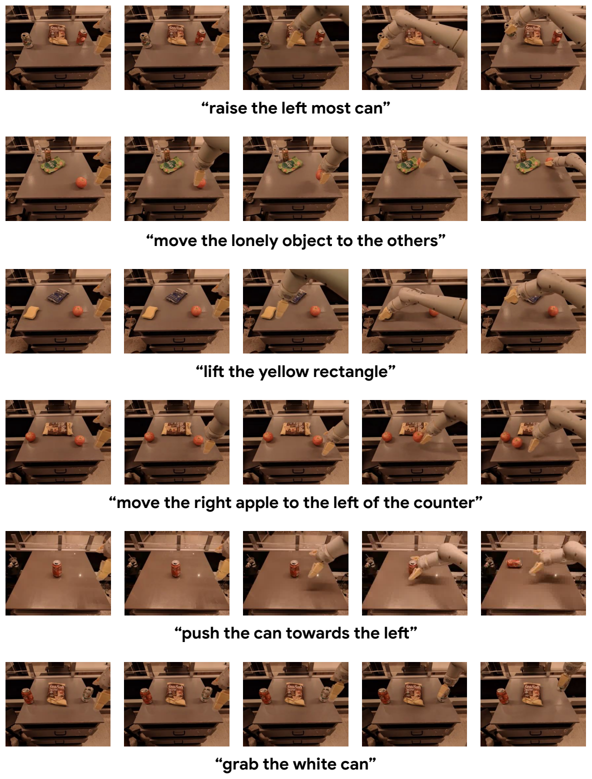

"raise the left most can"
"move the lonely object to the others"
"lift the yellow rectangle"
"move the right apple to the left of the counter"
"push the can towards the left"
"grab the white can"

Fig. 8: DIAL successfully completes various novel evaluation instructions requiring understanding of concepts such as relative spatial references, colors, and alternative phrasings. These concepts were not present in the structured teleoperator commands used for the original training demonstrations.

## Acknowledgement

The authors would like to thank Kanishka Rao, Debidatta Dwibedi, Pete Florence, Yevgen Chebotar, Fei Xia, and Corey Lynch for valuable feedback and discussions. We would also like to thank Emily Perez, Dee M, Clayton Tan, Jaspiar Singh, Jornell Quiambao, and Noah Brown for navigating the ever-changing challenges of data collection and robot policy evaluation at scale. Additionally, Tom Small designed informative animations to visualize DIAL. Finally, we would like to thank the large team that built [8] and [1], upon which we develop DIAL.

## References

[1] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. In *6th Annual Conference on Robot Learning*, 2022. URL https://openreview.net/forum?id=bdHkMjBJG_w.

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*, 2022.

[3] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3674–3683, 2018.

[4] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

[5] Dzmitry Bahdanau, Felix Hill, Jan Leike, Edward Hughes, Arian Hosseini, Pushmeet Kohli, and Edward Grefenstette. Learning to understand goal specifications by modelling reward. *arXiv preprint arXiv:1806.01946*, 2018.

[6] Valts Blukis, Yannick Terme, Eyvind Niklasson, Ross A Knepper, and Yoav Artzi. Learning to map natural language instructions to physical quadcopter control using simulated flight. *arXiv preprint arXiv:1910.09664*, 2019.

[7] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale. In *arXiv preprint arXiv:2212.06817*, 2022.

[8] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[9] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

[10] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, 2018.

[11] Harris Chan, Yuhuai Wu, Jamie Kiros, Sanja Fidler, and Jimmy Ba. Actrce: Augmenting experience via teacher's advice for multi-goal reinforcement learning. *arXiv preprint arXiv:1902.04546*, 2019.

[12] Yevgen Chebotar, Karol Hausman, Yao Lu, Ted Xiao, Dmitry Kalashnikov, Jake Varley, Alex Irpan, Benjamin Eysenbach, Ryan Julian, Chelsea Finn, et al. Actionable models: Unsupervised offline reinforcement learning of robotic skills. *arXiv preprint arXiv:2104.07749*, 2021.

[13] Geoffrey Cideron, Mathieu Seurin, Florian Strub, and Olivier Pietquin. Self-educated language agent with hindsight experience replay for instruction following. 2019.

[14] Zichen Jeff Cui, Yibin Wang, Nur Muhammad, Lerrel Pinto, et al. From play to policy: Conditional behavior generation from uncurated robot data. *arXiv preprint arXiv:2210.10047*, 2022.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[17] Felix Duvallet, Thomas Kollar, and Anthony Stentz. Imitation learning for natural language direction following through unknown environments. In *2013 IEEE International Conference on Robotics and Automation*, pages 1047–1053. IEEE, 2013.

[18] Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. *arXiv preprint arXiv:2206.08853*, 2022.

[19] Evelyn Fix and Joseph Lawson Hodges. Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review/Revue Internationale de Statistique*, 57(3):238–247, 1989.

[20] Justin Fu, Anoop Korattikara, Sergey Levine, and Sergio Guadarrama. From language to goals: Inverse reinforcement learning for vision-based instruction following. *arXiv preprint arXiv:1902.07742*, 2019.

[21] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022.

[22] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. *arXiv preprint arXiv:2201.07207*, 2022.

[23] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022.

[24] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022.

[25] Yiding Jiang, Shixiang Shane Gu, Kevin P Murphy, and Chelsea Finn. Language as an abstraction for hierarchical deep reinforcement learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[26] Leslie Pack Kaelbling. Learning to achieve goals. In *IJCAI*, volume 2, pages 1094–8. Citeseer, 1993.

[27] Dmitry Kalashnikov, Jacob Varley, Yevgen Chebotar, Benjamin Swanson, Rico Jonschkowski, Chelsea Finn, Sergey Levine, and Karol Hausman. Mt-opt: Continuous multi-task robotic reinforcement learning at scale. *arXiv preprint arXiv:2104.08212*, 2021.

[28] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. *arXiv preprint arXiv:2209.07753*, 2022.

[29] Jelena Luketina, Nantas Nardelli, Gregory Farquhar, Jakob Foerster, Jacob Andreas, Edward Grefenstette, Shimon Whiteson, and Tim Rocktäschel. A survey of reinforcement learning informed by natural language. *arXiv preprint arXiv:1906.03926*, 2019.

[30] Corey Lynch and Pierre Sermanet. Language conditioned imitation learning over unstructured data. *arXiv preprint arXiv:2005.07648*, 2020.

[31] Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In *Conference on robot learning*, pages 1113–1132. PMLR, 2020.

[32] Corey Lynch, Ayzaan Wahid, Jonathan Tompson, Tianli Ding, James Betker, Robert Baruch, Travis Armstrong, and Pete Florence. Interactive language: Talking to robots in real time. *arXiv preprint arXiv:2210.06407*, 2022.

[33] 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. *arXiv preprint arXiv:2108.03298*, 2021.

[34] Oier Mees, Jessica Borja-Diaz, and Wolfram Burgard. Grounding language with visual affordances over unstructured data. *arXiv preprint arXiv:2210.01911*, 2022.

[35] Oier Mees, Lukas Hermann, and Wolfram Burgard. What matters in language conditioned robotic imitation learning. *arXiv preprint arXiv:2204.06252*, 2022.

[36] Suraj Nair, Eric Mitchell, Kevin Chen, Silvio Savarese, Chelsea Finn, et al. Learning language-conditioned robot behavior from offline data and crowd-sourced annotation. In *Conference on Robot Learning*, pages 1303–1315. PMLR, 2022.

[37] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

[38] Matthias Plappert, Marcin Andrychowicz, Alex Ray, Bob McGrew, Bowen Baker, Glenn Powell, Jonas Schneider, Josh Tobin, Maciek Chociej, Peter Welinder, et al. Multi-goal reinforcement learning: Challenging robotics environments and request for research. *arXiv preprint arXiv:1802.09464*, 2018.

[39] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[40] Frank Röder, Manfred Eppe, and Stefan Wermter. Grounding hindsight instructions in multi-goal reinforcement learning for robotics. *arXiv preprint arXiv:2204.04308*, 2022.

[41] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[42] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport:

What and where pathways for robotic manipulation. In *Conference on Robot Learning*, pages 894–906. PMLR, 2022.

[43] Andrew Silva, Nina Moorman, William Silva, Zulfiqar Zaidi, Nakul Gopalan, and Matthew Gombolay. Lanconlearn: Learning with language to enable generalization in multi-task manipulation. *IEEE Robotics and Automation Letters*, 7(2):1635–1642, 2021.

[44] Simon Stepputtis, Joseph Campbell, Mariano Phielipp, Stefan Lee, Chitta Baral, and Heni Ben Amor. Language-conditioned imitation learning for robot manipulation tasks. *Advances in Neural Information Processing Systems*, 33:13139–13150, 2020.

[45] Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R Walter, Ashis Gopal Banerjee, Seth Teller, and Nicholas Roy. Approaching the symbol grounding problem with probabilistic graphical models. *AI magazine*, 32(4):64–76, 2011.

[46] Ted Xiao, Eric Jang, Dmitry Kalashnikov, Sergey Levine, Julian Ibarz, Karol Hausman, and Alexander Herzog. Thinking while moving: Deep reinforcement learning with concurrent control. *arXiv preprint arXiv:2004.06089*, 2020.

[47] Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Vikas Sindhwani, et al. Transporter networks: Rearranging the visual world for robotic manipulation. In *Conference on Robot Learning*, pages 726–747. PMLR, 2021.

## APPENDIX

### A. Additional Experiments

We present failure examples, study the importance of both pre-training and fine-tuning, and also explore whether VLMs used for DIAL could also be utilized as a task representation.

*1) Failure Examples:* In addition to the successful example trajectories visualized in Figure 5 and Figure 8, we also show some examples of failure cases in Figure 9.

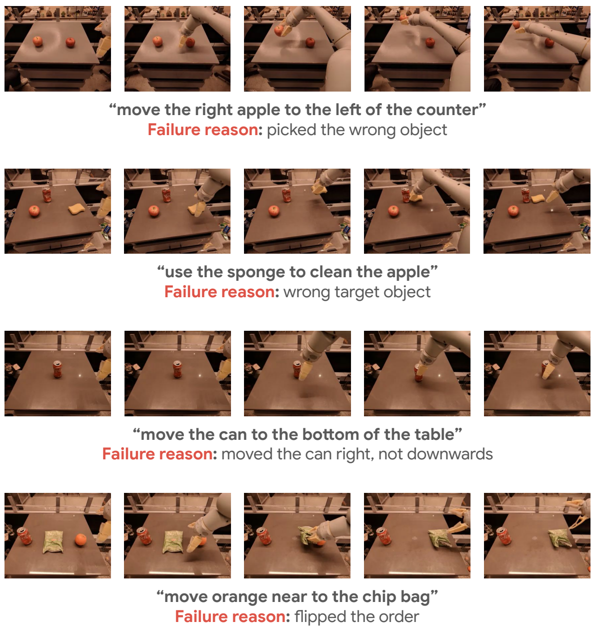

"move the right apple to the left of the counter"
Failure reason: picked the wrong object
"use the sponge to clean the apple"
Failure reason: wrong target object
"move the can to the bottom of the table"
Failure reason: moved the can right, not downwards
"move orange near to the chip bag"
Failure reason: flipped the order

Fig. 9: Samples of evaluation failures for DIAL. Errors are due to a combination of motor control and task confusion.

*2) How important is VLM fine-tuning for DIAL?:* Whereas Section V-C studies different prediction mechanisms for a fine-tuned CLIP model (FT-CLIP), we are also interested in comparing different CLIP models altogether. The main FT-CLIP model used in DIAL is initialized from the pretrained OpenAI CLIP weights and then fine-tuned on $\mathcal{D}_A$ from Section III-A, which begs the question: are both a strong pretrained initialization and subsequent fine-tuning necessary for strong instruction labeling performance? To answer this question, we perform instruction augmentation with (1) the frozen pretrained OpenAI CLIP model and (2) a fine-tuned CLIP model that starts from a random weight initialization instead of from the OpenAI pretrained weights.

As we see in Figure 10, predicted instruction accuracy for both of these models is significantly lower than the main FT-CLIP model. The poor performance of the frozen pretrained CLIP model (1) suggests that the particular embodied captioning task required by DIAL is likely quite out of distribution for the pre-training used by the OpenAI CLIP model, so some amount of domain data is required. On the other hand, the poor performance of training CLIP from scratch on robot demonstration data (2) suggests that internet-scale pre-training is required to start from a reasonable prior; there may not be sufficient robot domain data in the datasets we consider to fully train CLIP from scratch.

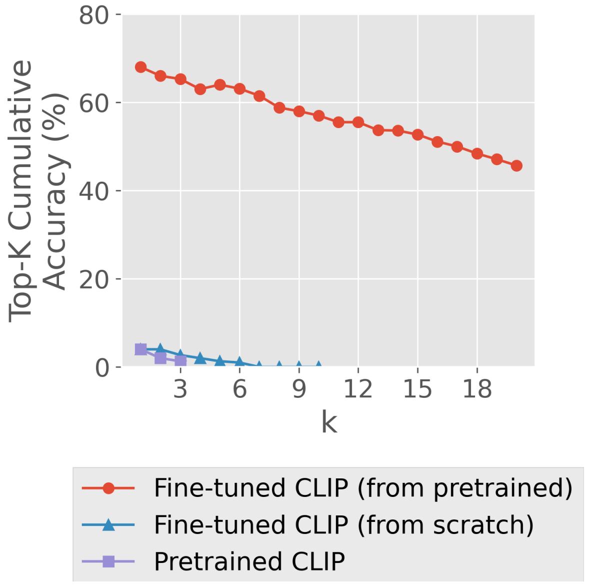


Fig. 10: Estimated top-$K$ cumulative instruction labeling accuracy using CLIP variants. The fine-tuned CLIP model from OpenAI checkpoint [39] performed significantly better than frozen CLIP model and the fine-tuned model from random initial weights.

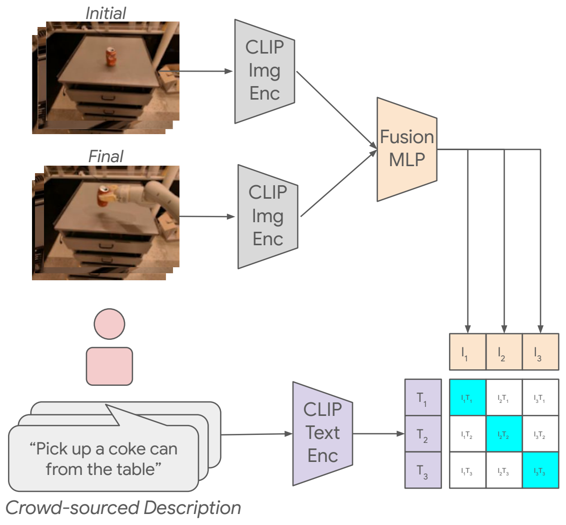


Fig. 11: CLIP architecture for fine-tuning using the contrastive loss. The image embeddings of the initial and final observation is concatenated and passed through an MLP to produce an episode embedding to dot product with the text embedding.

*3) Is a VLM good at relabeling also a good task representation?:* We study whether a VLM fine-tuned for instruction augmentation can also act as a better task representation for conditioning a policy in the form of a more powerful language embedding. Across the various groundtruth and relabeled datasets we focus on, we find that fine-tuned CLIP (FT-CLIP) is the most effective task representation, as seen in Table V. FT-CLIP is a good representation not only for freeform language instructions like those contained in the fine-tuning dataset $\mathcal{D}_A$, but also for structured metadata labels used to collect the demonstrations in $\mathcal{D}_B$. Thus, we utilize FT-CLIP's language encoder as the main task representation for language conditioning in all control policies we train, besides for policies that explicitly denote otherwise, such as in Table V.

| | | **Evaluation on Novel Instructions** | | | |
|---|---|---|---|---|---|
| Dataset | Task Encoder | *Spatial* | *Rephrased* | *Semantic* | Overall |
| $\mathcal{D}_A$ | USE | 22.5% | 50.0% | 0.0% | 21.7% |
| $\mathcal{D}_A$ | FT-CLIP | 30.0% | 40.0% | 7.7% | 27.5% |
| $\mathcal{D}_A$, $\mathcal{D}_B$ | CLIP | 45.0% | 40.0% | 10.0% | 40.0% |
| $\mathcal{D}_A$, $\mathcal{D}_B$ | FT-CLIP | 46.0% | 60.0% | 15.4% | 42.5% |
| DIAL, $k = 1$ | USE | 50.0% | 50.0% | 20.0% | 43.3% |
| DIAL, $k = 1$ | FT-CLIP | 62.0% | 40.0% | 23.1% | 50.0% |

TABLE V: Comparing downstream policy performance when improving the task representation from USE [10] to Pretrained OpenAI CLIP (CLIP) [39] to fine-tuned CLIP (FT-CLIP), as described in Section III-A. We find that the FT-CLIP representation is the best task representation in all dataset settings: training on crowd-sourced language annotations ($\mathcal{D}_A$), training on structured teleoperator commands along with crowd-sourced language annotations $\{\mathcal{D}_A, \mathcal{D}_B\}$, and using DIAL $\{\mathcal{D}_A, \mathcal{D}_B, \mathcal{D}_C\}$ with Top-$k$ with $k = 1$ (DIAL, $k = 1$).

## B. Dataset Details

Following the procedure in [7], we collect a large dataset of robot trajectories via teleoperation by uniformly sampling from one of the structured commands shown in Table VI and sending those commands to teleoperators that operate a mobilee manipulator robot in the real world. After teleoperators discard unsafe or failed demonstrations, we save a dataset of 80,000 successful robot demonstration trajectories ($\mathcal{D}_B$). Then, we send 2,800 of the episodes from $\mathcal{D}_B$ to be labeled by a pool of human labelers, who see the first and last frame of the episode and are tasked with providing a natural language description of how a robot could be commanded to from the first frame to the last frame. We request two independent language instruction annotations for each episode, so we obtain a total of 5,600 episode-instruction pairs, which we save as $\mathcal{D}_A$. Finally, we produce various relabeled datasets via instruction augmentation, that we detail in Section IV-B for different prediction methods for DIAL and Section V-B for different non-visual instruction augmentation baselines.

## C. DIAL Implementation Details

We implement DIAL with a CLIP model that is fine-tuned on 5,600 annotated episodes ($\mathcal{D}_A$) with the procedure described in Section III-A. The architecture of the CLIP model is shown in Figure 11. The initial and final state observation images are embedded using the CLIP image encoder $I_{enc}$, and the resulting embeddings are concatenated and passed through a 200 hidden dimension single-layer MLP to produce the final episode embedding. We use the CLIP text encoder $T_{enc}$ to embed the crowd-sourced annotations to produce the corresponding text embeddings. To fine-tune, we loaded the CLIP encoder weights from the `ViT-B/32` OpenAI checkpoint. We use a batch size of 64 and train for 100,000 iterations, but select the best checkpoint based on text prediction accuracy on a randomly held-out test set of 10% of the training dataset $\mathcal{D}_A$. We fine-tune both encoders $I_{enc}$ and $T_{enc}$ as well as the fusion MLP.

After fine-tuning CLIP, we source 18,719 candidate instruction labels ($L$) from $\mathcal{D}_A$ and a corpus of GPT-3 proposals of potential language instructions. The GPT-3 proposals are generated by using the prompt shown in Listing 1 to iterate over the 551 instructions used to collect teleoperated demonstrations. We note that that Listing 1 generates diverse instructions that may not be accurate for a given episode. Listing 1 is purposefully tuned to produce "hallucinated" descriptions that can add semantic properties in the proposed instructions that may or not be correct (for example, "pick up the orange" might be augmented into "retrieve the orange from the sink" or "raise the orange next to the vase"). The motivation behind this design decision is that GPT-3 predictions can be a lot less conservative when being used downstream by DIAL, since the CLIP model will ideally filter out irrelevant instructions. In contrast, the prompt in Listing 2 is used for producing Sentence-Level Synonyms, which should ideally always be factually equivalent to the original instruction.

Next, to relabel 80,000 robot trajectories that do not contain crowd-sourced annotations ($\mathcal{D}_B$) with $L$, we follow Section III-B to implement two variations of DIAL: **Top-$k$ selection** and **Min-$p$ selection**. For these two variations, we use $k = \{1, 3, 10\}$ and $p = \{0.1, 0.2, 0.3\}$.

## D. DIAL Hyperparameters

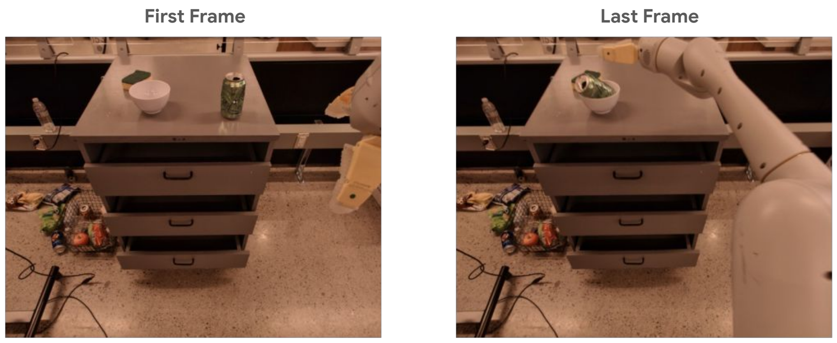
In Section V-C, we examined DIAL's sensitivity to hyperparameters, including top-K and min-P sampling, which influence dataset diversity and instruction accuracy (label noise). Such tradeoffs are common in data augmentation, where the benefits of increased augmentation must be balanced with the risks of too much label noise. These hyperparameter choices are highly domain-specific, and in our case we optimize this balance by selecting hyperparameters based on measuring offline VLM prediction accuracy. Notably, hyperparameter tuning in DIAL is substantially more cost-effective than manual full dataset labeling by humans, enabling scalable policy learning improvement from generated instructions. We plan to include further discussion on hyperparameter selection and future work on studying these tradeoffs in more detail.

## E. Instruction Augmentation Baselines

*a) Gaussian Noise:* Given an instruction $l$, we add Gaussian noise to the language embedding produced by the

| Structured Teleoperator Command Categories | Count | Description | Example Instruction |
|---|---|---|---|
| Pick `Object` | 17 | picking objects on a counter | pick water bottle |
| Move `Object` Near `Object` | 342 | moving an object near another | move pepsi can near rxbar blueberry |
| Place `Object` Upright | 8 | placing an elongated object vertically upright | place coke can upright |
| Knock `Object` Over | 8 | picking an object and laying it sideways on the counter | knock redbull can over |
| Open / Close `Drawer` | 6 | opening or closing a counter drawer | open the top drawer |
| Place `Object` into `Receptacle` | 85 | pick an object on the table and put it in a container or drawer | place brown chip bag into white bowl |
| Pick `Object` from `Receptacle` and Place on the Counter | 85 | pick an object out of a container or drawer and place it on the counter | pick green jalapeno chip bag from paper bowl and place on counter |
| Total | 551 | | |

TABLE VI: Teleoperators receive an instruction sampled from a total of 551 unique structured commands covering 7 different manipulation skills. The 80,000 episodes in $\mathcal{D}_B$ each contain exactly one of these structured teleoperator commands.



| Instruction Augmentation Prediction by CLIP | *p* | Accurate? |
|---|---|---|
| #1: pick up the green can and place it in the bowl which is at the left side of the table | 0.2244 | ✅ |
| #2: lift green can from table and place it in white cup | 0.1408 | ✅ |
| #3: pick up the green can which is close to the water bottle and place it in the bowl | 0.1209 | ❌ |
| #4: place green can into the plastic white bowl | 0.0699 | ✅ |
| #5: pick the green can from the bottom right of the table and place it into the white bowl | 0.0664 | ✅ |
| #6: pick up the silver can and place it in the white bowl | 0.0429 | ❌ |
| #7: bring the blue can and place it into white paper bowl | 0.0417 | ❌ |
| #8: pick up the green can from the bottom left side of the table | 0.0388 | ❌ |
| #9: pick up the green can from the bottom side of the table and drop it into bowl | 0.0339 | ✅ |
| #10: pick up the red bull can and drop it in the white bowl | 0.0243 | ❌ |

Fig. 12: The top 10 proposed instruction augmentations for a single episode with original structured teleoperator command `place green can in white bowl`. In some cases, the predicted captions provide additional semantic information such as describing the location of the can or the material of the bowl. As seen in Figure 7, the probability CLIP assigns to each the candidates quickly drops off past a few top predictions. Setting Min-$p = 0.2$ would only take the first instruction augmentation prediction, while setting $k = 3$ would take the top three predictions, including an incorrect prediction (#3).

CLIP text encoder $T_{enc}$, directly obtaining the augmentation in the latent space $\tilde{z}_l$:

$$\tilde{z}_l = T_{enc}(l) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma) \in \mathbb{R}^d \tag{3}$$

In our implementation, we choose $\sigma = 0.05$ and perform the Gaussian noise augmentation dynamically to the 512-dimension CLIP $T_{enc}$ embedding resulting from passing in the original language instruction to the CLIP text encoder.

*b) Word-level Synonyms:* We replace *individual words* in existing instructions with sampled synonyms from a predefined list. The mapping between words present in the original structured 551 instructions and possible synonyms is shown in Listing 3.

*c) Sentence-level Synonyms:* We replace entire instructions with alternative instructions as proposed by GPT-3. We pre-compute valid sentence-level synonyms by using the prompt shown in Listing 1 to iterate over the 551 instructions used to collect teleoperated demonstrations.

### *F. Augmented Dataset Details*

As noted in Section IV-B, DIAL uses CLIP to score predictions from 18,719 candidate text labels $L$, which are a union of 9,393 crowd-sourced instructions ($L_{CC}$) from the original $\mathcal{D}_A$ dataset as well as 9,336 instruction candidates ($L_{GPT3}$) from GPT-3 curated with the prompt in Listing 1. During the instruction augmentation process, we find that CLIP selects 3,675 unique instructions from these 18,719 candidates $L$. Additionally, 86.3% of selected instructions were from $L_{CC}$ and 13.7% from $L_{GPT3}$. These characteristics show that CLIP generally prefers human-like instructions while sometimes incorporating GPT-3 variations. We note that these prediction characteristics may result from the fact that our relabeling model CLIP was finetuned solely on crowd-sourced labels, and that the desired output distribution is unclear (is it a positive or negative property to prefer crowd-sourced annotations?). Furthermore, it's possible that the true distribution of accurate and desireable language descriptions was already well-covered by the human annotations; so if GPT-3 proposed an instruction already contained in the human-sourced $L_{CC}$, we would deduplicate that candidate from $L$ and consider the instruction to be human-sourced. To enhance diversity, future work could leverage improved VLMs and fine-tune on more diverse labels, including LLM-generated captions.

## G. Language-Conditioned Policy Training

The policies used in this work are trained using the RT-1 [7] architecture on a large dataset of human-provided demonstrations. The policies receive natural language descriptions in the form of a 512-dimensional VLM embedding and a short history of images and outputs discrete action tokens which are then transformed to continuous action outputs. Apart from the experiments explicitly denoted as using USE [10] and frozen CLIP [39] language embeddings in Table V, all policies trained in this work utilize fine-tuned CLIP (FT-CLIP) language embeddings as described in Section III-A.

Note that the exact policy architecture is not the main focus of this work, so we utilize the exact same policy training procedure across each experiment and only vary the instruction augmented datasets that the policies are trained on.

## H. Evaluation Instructions

We utilize an evaluation setup focusing solely on novel instructions unseen during training. To source these novel instructions, we 1) crowd-source instructions from a different set of humans than the original dataset labelers and 2) prompt GPT-3 with Listing 2 to produce reasonable tasks that might be asked of a home robot manipulating various objects on a kitchen counter. Then, we normalize all instructions by removing punctuation, removing non-alphanumeric symbols, converting all instructions to lower case, and removing leading and ending spaces. Afterwards, we filter out any instructions already contained in either the instruction augmentation process in Section III-B or in the original set of 551 foresight tasks in Table VI. Finally, as seen in Table VII, we organize them into various semantic categories to allow for more detailed analysis of quantitative policy performance.

Listing 1: GPT-3 Prompt for Proposing Candidate Tasks.

```
For the following tasks for a helpful home robot, rephrase
them to imagine different variations of the task. These
variations include different types of objects, different
locations, different obstacles, and different strategies
for how the task should be accomplished.

3 rephrases for: pick mountain dew
Answer: lift the mountain dew on the left side of the desk,
grab the mountain dew soda next to the water, pick the
farthest green soda can

4 rephrases for: move your arm to the right side of the desk
Answer: bring your arm to the right of the counter, move
right slightly, go far to the rightmost part of the table,
reorient your gripper to point right

10 rephrases for: bring me the yogurt
Answer: retrieve the yogurt, bring the white snack, pick up
the yogurt cup from the far right, lift the yogurt snack
from the left, bring back the yogurt near the chip bag, lift
the yogurt from the top of the counter, bring the yogurt
closest to the apple, grab the yogurt, lift the close left
yogurt on the bottom left, retrieve the yogurt on the
bottom of the table

10 rephrases for: <INSTRUCTION_TO_AUGMENT>
Answer:
```

Listing 2: GPT-3 Prompt for "Sentence-level Synonyms".

```
You are a helpful home robot in an office kitchen. You are
able to manipulate household objects in a safe and efficient
manner. Here are some tasks you are able to accomplish in
various environments:

5 tasks in a sink with a sponge, brush, plate, and a cup:
move sponge near the cup, fill up the cup with water, clean
the plate with the brush, pick up the plate, put the cup
on the plate

3 tasks in a storage room with a box, a ladder, and a
hammer: lift the hammer, push the ladder, put the hammer
in the box

10 tasks on a table with an apple, a coke can, a sponge, and
an orange: pick up the apple, pick up the coke can, use the
sponge to clean the apple, use the sponge to clean the coke
can, put the apple down, put the coke can down, pick up the
orange, peel the orange, eat the orange, throw away the peel

10 tasks on a table with <OBJECT_1>, <OBJECT_2>, and
<OBJECT_3>:
```

Listing 3: Synonym Mapping for “Word-level Synonyms”.

```
SYNONYM_MAP = {
    'rxbar blueberry': [
        'rxbar blueberry', 'blueberry rxbar',
        'the blueberry rxbar', 'the rxbar blueberry'
    ],
    'rxbar chocolate': [
        'rxbar chocolate', 'chocolate rxbar',
        'the chocolate rxbar', 'the rxbar chocolate'
    ],
    'pick': ['pick', 'pick up', 'raise', 'lift'],
    'move': [
        'move', 'push', 'move', 'displace', 'guide',
        'manipulate', 'bring'
    ],
    'knock': ['knock', 'push over', 'flick', 'knockdown'],
    'place': ['place', 'put', 'gently place', 'gently put'],
    'open': ['open', 'widen', 'pull', 'widely open'],
    'close': ['close', 'push close', 'completely close'],
    'coke': [
        'coke', 'coca cola', 'coke', 'coca cola',
        'the coke', 'a coke', 'a coca cola', 'the coca cola'
    ],
    'green': [
        'green', 'bright green', 'grass colored', 'lime',
        'a green', 'the green', 'a lime', 'the lime',
        'the bright green', 'a bright green'
    ],
    'blue ': ['blue ', 'dark blue ', 'the blue ', 'a blue '],
    'pepsi': ['pepsi', 'blue pepsi', 'pepsi', 'a pepsi',
        'the pepsi'],
    '7up': ['7up', 'white 7up', '7up', '7-up', '7up',
        'a 7up', 'the 7up'],
    'redbull': [
        'redbull', 'red bull', 'energy drink',
        'redbull energy', 'redbull soda',
        'the redbull', 'a redbull', 'a red bull',
        'the red bull'
    ],
    'blueberry': ['blueberry', 'blue berry'],
    'chocolate': ['chocolate', 'brown chocolate'],
    'brown': ['brown', 'coffee colored',
                'the brown', 'a brown'],
    'jalapeno': ['jalapeno', 'spicy', 'hot', 'fiery'],
    'rice': ['rice'],
    'chip': ['chip', 'snack', 'chips'],
    'plastic': ['plastic'],
    'water': ['water', 'water', 'agua'],
    'bowl': ['bowl', 'half dome', 'chalice'],
    'togo': ['togo', 'to-go', 'to go'],
    'box': ['box', 'container', 'paper box'],
    'upright': ['upright', 'right side up', 'correctly'],
    'near': ['near', 'close to', 'nearby',
            'very near', 'very close to'],
    'can': ['can', 'soda can', 'aluminum can'],
    'rxbar': ['rxbar', 'snack bar', 'granola bar',
        'health bar', 'granola'],
    'apple': [
        'apple', 'red apple', 'the apple', 'the red apple',
        'an apple', 'a red apple', 'small apple',
        'the small apple'
    ],
    'orange': [
        'orange', 'the orange', 'orange fruit', 'an orange',
        'a small orange', 'a large orange'
    ],
    'sponge': [
        'sponge', 'yellow sponge', 'the yellow sponge',
        'a yellow sponge', 'a sponge', 'the sponge'
    ],
    'bottle': ['bottle', 'plastic bottle',
                'recycleable', 'clear'],
}
```

| **Category** | **Instruction Samples** |
|---|---|
| *Spatial* | ['grab the bottle on the left of the table', 'grab the can which is on the right side of the table', 'grab the chip on the left', 'grab the chip on the right', 'grab the right most apple', 'knock down the right soda', 'lift the apple which is on the left side of the table', 'lift the apple which is on the right side of the table', 'lift the chips on the left side', 'lift the chips on the right side', 'lift the left can', 'move the left soda to the can on the right side of the table', 'move the soda can which is on the right toward the chip bag', 'pick can which is on the left of the table', 'pick can which is on the right of the table', 'pick chip bag on the left', 'pick chip bag on the right', 'pick the can in the middle', 'pick the left coke can', 'pick the left fruit', 'pick the leftmost chip bag', 'pick the object on the right side of the table', 'pick the right coke can', 'pick the right object', 'pick the rightmost chip bag', 'pick up the left apple', 'pick up the left object', 'pick up the right can', 'pick up the right object', 'push the left side apple to the brown chips', 'raise bottle which is to the left of the can', 'raise the blue tin', 'raise the left most can', 'raise the thing which is on the left of the counter'] |
| *Rephrased* | ['grab and lift up the green bag', 'grab the blue pepsi', 'grab the white can', 'knock over the water', 'lift the orange soda', 'lift the yellow rectangle', 'liftt the fruit', 'move green packet near the red apple', 'move orange near to the chip bag', 'pick up the apple fruit', 'push green chips close to the coke', 'upright the lime green can', 'put the apple next to the candy bar', 'retrieve the can from the left side of the coffee table', 'set the apple down next to the chocolate bar', 'take the can from the left side of the counter'] |
| *Semantic* | ['move the can to the bottom of the table', 'move the green bag away from the others', 'move the lonely object to the others', 'move the right apple to the left of the counter', 'push blue chip bag to the left side of the table', 'push the can towards the left', 'push the can towards the right', 'push the left apple to the right side', 'use the sponge to clean the coke can', 'use the sponge to clean the apple'] |

TABLE VII: Novel evaluation instructions sourced from humans or GPT-3, grouped by category. Spatial tasks focus on tasks involving Spatial relationships, Rephrased tasks contain tasks that directly map to a foresight skill, and Semantic tasks describe semantic concepts not contained in the relabeled or original datasets. In total, there are 60 instructions across the three categories.