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

Abstract

In recent years, much progress has been made in learning robotic manipulation policies that follow natural language instructions. Such methods typically learn from corpora of robot-language data that was either collected with specific tasks in mind or expensively re-labeled by humans with rich language descriptions in hindsight. Recently, large-scale pretrained vision-language models (VLMs) like CLIP [38] or ViLD [21] 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 to make them useful even for tasks that are not reflected in their ground truth annotations? For example, if the original annotations contained simple task descriptions such as "pick up the 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 **D**ata-driven **I**nstruction **A**ugmentation for **L**anguage-conditioned control (DIAL): we utilize semi-supervised language labels leveraging the semantic understanding of CLIP to propagate knowledge onto large datasets of unlabeled demonstration data and then train language-conditioned policies on the augmented datasets. 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 96.5% of the 80,000 demonstrations do not contain crowd-sourced language annotations. DIAL enables imitation learning policies to acquire new capabilities and generalize to 60 novel instructions unseen in the original dataset. We provide examples of DIAL policy evaluations at https://instructionaugmentation.github.io

1 Introduction

Recent advances in decision making have combined data-driven policies with language models to enable control policies that respond to natural language instructions, an important capability for practical adoption of general robots in the real world. A popular method used to learn such policies is behavioral cloning (BC) [24, 32], which has been applied to robotic datasets [2] with diverse language instructions [31]. Similar to trends observed in other fields of machine learning, much of the progress in language-conditioned BC stems from the availability and scale of labeled robotic control datasets. Commonly, human teleoperation is used to collect robot demonstrations that are paired with language instruction annotations, which may also be provided by human labeling [29]. While the language instruction label for a demonstration provides useful supervision, the demonstration may illustrate many behaviors and semantic concepts beyond the original language instruction. A solution might be

[*]Equal contribution

NeurIPS 2022 Workshop on Robot Learning: Trustworthy Robotics, Virtual, Virtual

to ask humans for additional language labels by showing them the original episode [31], but this may be expensive and time-intensive. Is there a way to extract additional concepts beyond the original instruction label, preferably without additional human labeling? Similarly, is there a way to leverage behaviors in an demonstration without any initial language label?

One possibility is to leverage large-scale pretrained language models (LLMs) [8, 14] and vision-language models (VLMs) [3, 38], which have been able to pretrain on internet-scale data and then be applied to downstream domains. In robotics, they have been used as representations for perception [36, 41], as task representation for language [24, 29], or as planners [2, 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 [38] on a challenging real-world robotic manipulation setting with 80,000 teleoperated demonstrations and 5,600 crowd-sourced instruction labels, showing that DIAL allows policies to acquire understanding of skills not contained in the original task labels. Sample emergent capabilities of our method are shown in Figure 4, but we focus our analysis on a large quantitative evaluation of over 1,300 real world robot evaluations across 60 novel instructions. We find that our method outperforms instruction augmentation methods that aren't visually grounded and improves novel instruction performance when applied to either partially or fully labeled datasets.

2 Related Work

Language instruction following in robotics. Language-instruction following agents have been extensively explored with engineered symbolic representations [16, 44], with Reinforcement Learning (RL) [6, 19, 28], and with Behavior Cloning (BC) [4, 7, 24, 29]. Recent advances in deep learning with large amounts of data have led to works following natural language for robotic manipulations [2, 27, 33, 42, 43]. Latent Motor Policies (LMP) [30] learns hierarchical goal-conditioned policies. Subsequent Language from Play (LfP) [29] uses language goals provided by large dataset of hindsight human labels on robotic play data. Similarly, Interactive Language [31] 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.

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 [17, 35], as part of the agent architecture [36, 41], or as planners for long-horizon tasks [2, 22, 23]. MineCLIP [17] fine-tunes CLIP [38] 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 [35] 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 [40] using a binary cross entropy loss. CLIPort [41] uses a frozen CLIP vision and text encoders in combination with Transporter networks [48] for imitation learning. R3M [36] uses representations pretrained constrastively on Ego4D [20] human video datasets for robotic policy learning via imitation learning. For long-horizon language instructions, LLMs are used as planners both in a simulated [22] and real-world robotics [2]. 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.

Hindsight relabeling for goal-conditioned reinforcement learning. The relabeling approach for goal-conditioned reinforcement learning [37] originates from Hindsight Experience Replay (HER) [5], which relabels the desired goals in a trajectory with achieved goals (hindsight goal) in the same trajectories to generate positive examples in a sparse reward setting. Relabeling has later been applied to environments where the goals are images [11], task IDs [26], and language instructions [10, 12, 25]. Early works with templated language goals rely on environment simulators to provide hindsight labels [10, 25], and more recently [12, 39] use learned models to predict templated language instructions.

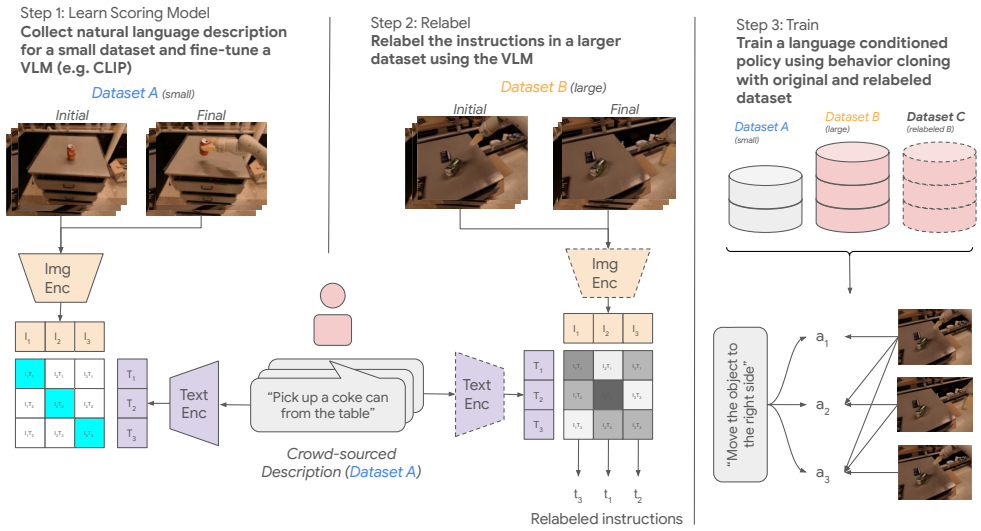


Figure 1: DIAL consists of three steps: (1) Contrastive fine-tuning of a vision-language model (VLM) such as CLIP [38] on small dataset of robot manipulation trajectories with crowd-sourced natural language annotation, (2) using the fine-tuned VLM (in dashed outline) to score and rank the relevance of crowd-sourced annotations against a larger dataset of trajectories to produce novel instruction labels, and (3) training a language-conditioned policy using behavior cloning on the original and relabeled dataset. See Section 3 for more details.

Our work leverages a pretrained VLM to produce unstructured natural language relabeling instructions that scale to real robot environments.

3 Data-driven Instruction Augmentation for Language-conditioned Control

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

3.1 Finetuning Vision-Language Model Representations on Offline Dataset

Given an offline dataset of N trajectories $[\tau_1, \ldots, \tau_N]$, $\tau_n = ([(s_0^n, a_0^n), (s_1^n, a_1^n), \ldots, (s_T^n)])$, we collect a corresponding natural language description 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 [34, 41], our instantiation of DIAL uses CLIP [38] 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) [15] 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_θ, 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 α across the states and texts:

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

3.2 Instruction Augmentation on Offline Datasets

We are also given a much larger offline dataset of $M \gg N$ trajectories $[\hat{\tau}_1, \ldots, \hat{\tau}_M]$, where $\hat{\tau}_m = ([(\hat{s}_0^m, \hat{a}_0^m), (\hat{s}_1^m, \hat{a}_1^m), \ldots, (\hat{s}_T^m)])$. These trajectories may be collected from human teleoperated demonstrations on a wide variety of tasks [2], or from unstructured robotic "play" data [30].

We assume that these trajectories do not have any associated instruction labels for these episodes. We denote this larger offline dataset as $\mathcal{D}_B = [\hat{\tau}_1, \ldots, \hat{\tau}_M]$.

We use the fine-tuned VLM model to propose natural language instructions $\tilde{l}^m$ for the 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 [8] proposals of possible tasks, which we denote as $\mathcal{D}_{GPT-3}$ (the details of this procedure will be covered in Section 4.2). 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 [18] 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 2 visualizes the three datasets generated.

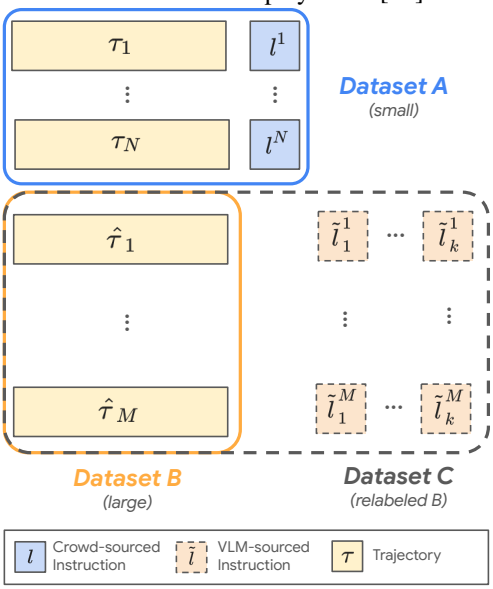


Figure 2: The construction of datasets: Dataset A ($\mathcal{D}_A$) (blue) consists of the N trajectories $\{\tau_n\}_{n=1}^N$ labeled with crowd-sourced instructions $\{l^n\}_{n=1}^N$ describing what the robot agent performed in the episode. Dataset B ($\mathcal{D}_B$) (yellow) consists of a much larger set of trajectories, $\{\hat{\tau}_m\}_{m=1}^M$ *without* crowd-sourced instructions. Dataset C ($\mathcal{D}_C$) (black, dashed) contains Dataset B trajectories relabeled with VLM-sourced hindsight instruction(s) $\{\tilde{l}_1^m, \ldots, \tilde{l}_k^m\}_{m=1}^M$.

There are several potential strategies for candidate instruction selection:

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.

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

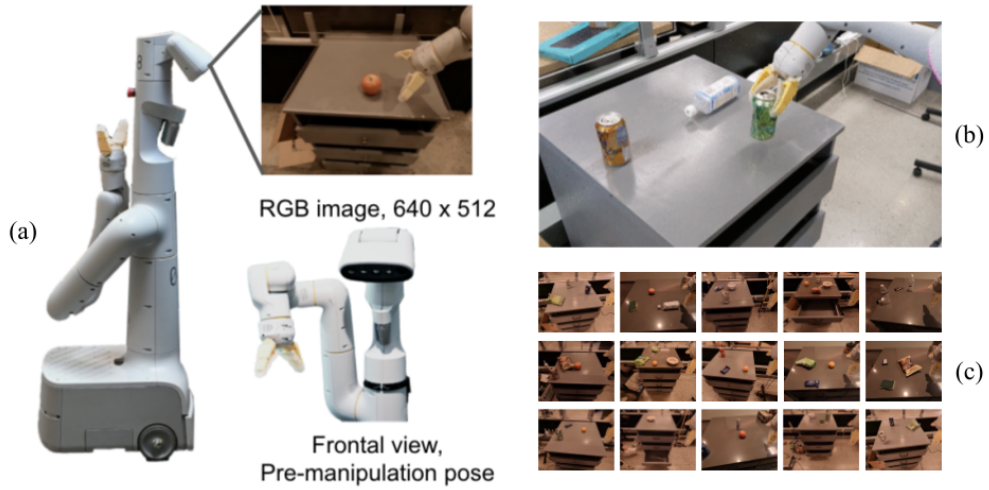


Figure 3: (a) An Everyday Robots mobile manipulator robot receives RGB images from an onboard camera and uses a 7 DoF arm with parallel-jaw grippers. (b) The robot performs a variety of manipulation tasks with kitchen objects and cabinet drawers. (c) A sample of scenes, which range across various countertops, drawers, and object arrangements in an office kitchen setting.

softmax over all the candidate instructions with temperature parameter α from CLIP:

$$P(\tilde{l}^m = l^n | (\hat{s}_0^m, \hat{s}_T^m)) = \frac{\exp(d(z_l^n, \hat{z}_s^m)/\alpha)}{\sum_{n'} \exp(d(z_l^{n'}, \hat{z}_s^m)/\alpha)} \tag{2}$$

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

$$P(\tilde{l}^m = l) \geq p, \quad \forall\, l \in L^{(p)} \tag{3}$$

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 5.3 the effects of these candidate instruction selection strategies on relabeled instruction accuracy, augmented dataset size, and downstream policy performance.

3.3 Learning 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.

4 Experimental Setup

We first describe the experimental conditions including the kitchen environment, physical robot features, and datasets collected. We then describe the various methods used to perform instruction augmentation on our datasets and the subsequent details on policy training. Finally, we detail our evaluation protocol involving unseen instructions.

4.1 Environment, Robot, and Datasets

We implement DIAL in a challenging real-world robotic manipulation setting in a kitchen environment similar to SayCan [2]. We focus on the practically-motivated setting where a dataset of teleoperated

demonstrations is available, collected for downstream imitation learning [2, 24]. An Everyday Robots robot [46], a mobile manipulator with RGB observations, is placed in an office kitchen to interact with common objects using concurrent [47] continuous closed-loop control from pixels, as shown in Figure 3. The robot uses parallel-jaw grippers, an over-the-shoulder RGB camera, and a 7 DoF arm. We collect a large-scale dataset of over 80,000 robot trajectories via human teleoperation ($\mathcal{D}_B$ in Section 3.2), where teleoperators perform 551 unique tasks motivated by common manipulation skills and objects in a kitchen environment [2]. Afterwards, we leverage crowd-sourced human annotators to label 2,800 robot trajectories with two possible hindsight instructions each, resulting in a total of 5,600 unique episodes with crowdsourced captions ($\mathcal{D}_A$ in Section 3.1). 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.

4.2 Instruction Augmentation

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

DIAL implementations We implement DIAL with a CLIP model that is finetuned on 5,600 annotated episodes ($\mathcal{D}_A$) with the procedure described in Section 3.1. After finetuning CLIP, we source 18,719 candidate instruction labels ($\mathcal{L}$) from the combination of $\mathcal{D}_A$ and a corpus of GPT-3 proposals of potential language instructions. To perform instruction augmentation that relabels a dataset of 80,000 robot trajectories that do not contain crowd-sourced annotations ($\mathcal{D}_B$) with $\mathcal{L}$, we follow Section 3.2 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 ($p = 0.3$). With increasing k or decreasing p, augmented datasets become larger but the relabeled instructions may become increasingly irrelevant or inaccurate, which is explored is explored further in Section 5.3. Additional details can be found in Appendix A.2.

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 "Sentence-level Synonyms" baseline that replaces entire instructions with alternative instructions as proposed by GPT-3. Additional implementation details for these baselines can be found in Appendix A.3.

4.3 Policy Training

Using these various instruction augmented datasets, we train vision-based language-conditioned behavior cloning policies similar to the formulation in BC-Z [24], as described in Section 3.3. Compared to BC-Z, we use a larger Transformer [45] based backbone instead of ResNet18 and use a CLIP language encoder instead of a Universal Sentence Encoder [9]. Nonetheless, we treat the behavior 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. We provide further details on our behavior cloning policy implementation in Appendix A.5.

4.4 Evaluation

In contrast to prior works [4, 7] 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 instruction augmentation process in Section 3.2 or in the original set of 551

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 1: 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 6.

foresight tasks in Section 4.1; in total, we sample 60 novel evaluation instructions. After sourcing these instructions, we organize them into semantic categories to allow for more detailed analysis of qualitative policy performance; examples are shown in Table 1 and a full list is provided in Table 6.

5 Experimental Results

In our experiments, we investigate whether DIAL can improve the policy performance on the unseen tasks described in Section 4.4 when starting from fully or partially labeled source datasets. We ablate on the types of instruction augmentations described in Section 4.2, and analyze the importance of accuracy when augmenting instructions with DIAL.

5.1 Does DIAL improve policy performance on unseen tasks?

We investigate whether instruction augmentation can enable language-conditioned behavior cloning policies to successfully perform novel instructions. We find that DIAL is able to solve challenging novel tasks that elude the baseline models, as shown in Table 2. An example is shown in Figure 4,

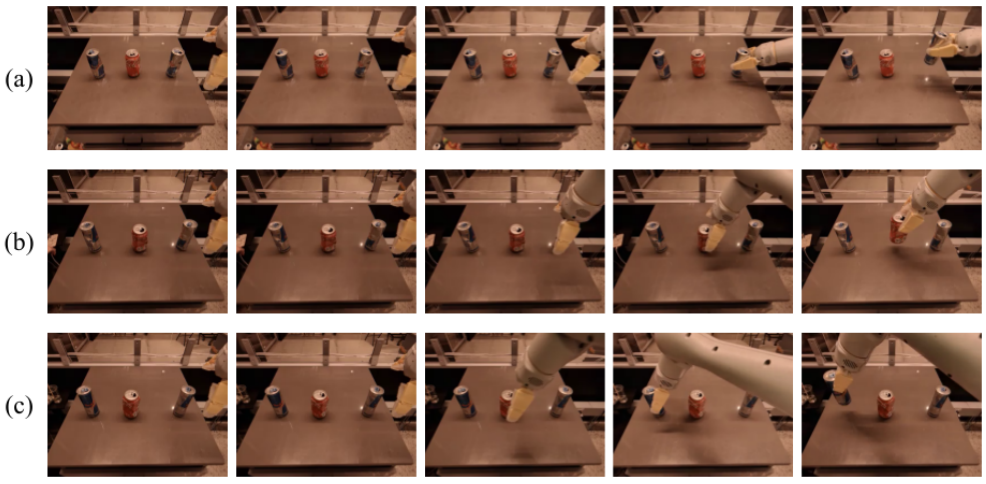


Figure 4: Given the same starting scene, DIAL correctly follows the instructions of (a) `pick can which is on the right of the table`, (b) `pick the can in the middle`, and (c) `pick can which is on the left of the table`. Non-DIAL methods don't adjust their behavior based on language commands, performing the same motions that show a lack of spatial understanding.

Instruction Augmentation	Evaluation on Novel Instructions			
	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%
Sentence-level Synonyms	28.0%	46.7%	23.1%	30.0%
DIAL (ours)	**68.0%**	**66.7%**	**30.8%**	**60.0%**

Table 2: Evaluating language-conditioned BC policies trained on various relabeled datasets produced from different types of instruction augmentation. Trained policies are evaluated for 80 evaluations each over 60 novel task instructions not present in the training datasets. DIAL with CLIP is able to produce the most performant policy, especially on *Spatial* Tasks requiring visual scene understanding.

DIAL Version	Dataset Properties		Evaluation on Novel Instructions			
Instruction Augmentation	Foresight Instructions	Crowd-sourced Labels	*Spatial* Tasks	*Rephrased* Tasks	*Semantic* Tasks	Overall
None	80,000	0	38.0%	40.0%	15.4%	33.8%
None	0	5,600	30.0%	40.0%	7.7%	27.5%
None	80,000	5,600	46.0%	60.0%	15.4%	42.5%
DIAL (ours)	80,000	0	50.0%	37.5%	10.0%	36.7%
DIAL (ours)	0	5,600	58.0%	46.7%	15.4%	47.5%
DIAL (ours)	80,000	5,600	**68.0%**	**66.7%**	**30.8%**	**60.0%**

Table 3: Comparing the performance of DIAL when applied to partially labeled datasets. DIAL is able to significantly improve novel instruction evaluation performance across all three evaluation categories, especially in the setting where Foresight Labels are not available.

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 5.

5.2 Does using DIAL for *partially* labeled datasets improve performance on unseen tasks?

The source trajectory dataset we utilize consists of a 5,600 trajectories dataset ($\mathcal{D}_A$) with crowd-sourced hindsight labels and a larger 80,000 trajectories dataset ($\mathcal{D}_B$) that does not have any crowd-sourced instructions. Even though the dataset $\mathcal{D}_B$ does not contain hindsight labels, it contains structured task information that was used to guide human demonstrators as to which task should be collected (for example, "pick coke can"); we refer to these structured commands as foresight instructions. In Section 5.1, we considered all information available during both instruction augmentation as well as policy training. However, does DIAL still improve policy performance in scenarios where the source dataset is only partially labeled? We study the setting where foresight instructions are not available as well as the setting where crowd-sourced labels are not available. As shown in Table 3, we find that in both cases DIAL significantly increases performance on novel evaluation instructions. This experiment is 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 [13, 30, 31].

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 4: 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 Appendix A.4 and Figure 7.

5.3 How sensitive is DIAL to hyperparameters and instruction prediction accuracy?

We study the tradeoff 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 4.2, 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, we ask human labelers to rate whether proposed instruction augmentation are factually accurate descriptions of a given episode. We show an example of predicted instruction augmentations in Figure 7 and show detailed analysis in Appendix A.4. When applying these different relabeled datasets to downstream policy learning, we find in Table 4 that Min-p instruction prediction, a more conservative approach than Top-k prediction, performs the best across all evaluation instructions.

6 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 large scale real world robotic manipulation domain, we find that DIAL is able to outperform baselines on a challenging set of 60 novel evaluation instructions unseen during training We compare DIAL against instruction augmentation methods that don't 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 Although DIAL seems to improve policy understanding on many novel concepts not contained in the original training dataset, there are still cases where DIAL fails, especially when evaluating tasks that may require new motor skills. In addition, there are domain and dataset specific design choices for DIAL such as the choice of instruction prediction parameters. We find that pretrained VLMs do not work well zero-shot on specific robotic domains, so DIAL currently finetunes with domain-specific robot data.

Future Work An interesting direction is to view DIAL as goal-conditioning and attempting visual goals during training or evaluation. In addition, on-policy or RL variations of DIAL may be able to effectively explore the task representation space autonomously.

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 [1] and [2], upon which we develop DIAL.

References

[1] Google paper on robotic transformer policies - available soon. *Available on arXiv soon*, 2022.

[2] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, D. Ho, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, E. Jang, R. J. Ruano, K. Jeffrey, S. Jesmonth, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, K.-H. Lee, S. Levine, Y. Lu, L. Luu, C. Parada, P. Pastor, J. Quiambao, K. Rao, J. Rettinghouse, D. Reyes, P. Sermanet, N. Sievers, C. Tan, A. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, S. Xu, and M. Yan. Do as i can and not as i say: Grounding language in robotic affordances. In *arXiv preprint arXiv:2204.01691*, 2022.

[3] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, et al. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*, 2022.

[4] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. 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.

[5] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. Pieter Abbeel, and W. Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

[6] D. Bahdanau, F. Hill, J. Leike, E. Hughes, A. Hosseini, P. Kohli, and E. Grefenstette. Learning to understand goal specifications by modelling reward. *arXiv preprint arXiv:1806.01946*, 2018.

[7] V. Blukis, Y. Terme, E. Niklasson, R. A. Knepper, and Y. Artzi. Learning to map natural language instructions to physical quadcopter control using simulated flight. *arXiv preprint arXiv:1910.09664*, 2019.

[8] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[9] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, et al. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, 2018.

[10] H. Chan, Y. Wu, J. Kiros, S. Fidler, and J. Ba. Actrce: Augmenting experience via teacher's advice for multi-goal reinforcement learning. *arXiv preprint arXiv:1902.04546*, 2019.

[11] Y. Chebotar, K. Hausman, Y. Lu, T. Xiao, D. Kalashnikov, J. Varley, A. Irpan, B. Eysenbach, R. Julian, C. Finn, et al. Actionable models: Unsupervised offline reinforcement learning of robotic skills. *arXiv preprint arXiv:2104.07749*, 2021.

[12] G. Cideron, M. Seurin, F. Strub, and O. Pietquin. Self-educated language agent with hindsight experience replay for instruction following. 2019.

[13] Z. J. Cui, Y. Wang, N. Muhammad, L. Pinto, et al. From play to policy: Conditional behavior generation from uncurated robot data. *arXiv preprint arXiv:2210.10047*, 2022.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[15] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[16] F. Duvallet, T. Kollar, and A. 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.

[17] L. Fan, G. Wang, Y. Jiang, A. Mandlekar, Y. Yang, H. Zhu, A. Tang, D.-A. Huang, Y. Zhu, and A. Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. *arXiv preprint arXiv:2206.08853*, 2022.

[18] E. Fix and J. L. Hodges. Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review/Revue Internationale de Statistique*, 57(3):238–247, 1989.

[19] J. Fu, A. Korattikara, S. Levine, and S. Guadarrama. From language to goals: Inverse reinforcement learning for vision-based instruction following. *arXiv preprint arXiv:1902.07742*, 2019.

[20] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. 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.

[21] X. Gu, T.-Y. Lin, W. Kuo, and Y. Cui. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2021.

[22] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. *arXiv preprint arXiv:2201.07207*, 2022.

[23] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022.

[24] 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.

[25] Y. Jiang, S. S. Gu, K. P. Murphy, and C. Finn. Language as an abstraction for hierarchical deep reinforcement learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[26] 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.

[27] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng. Code as policies: Language model programs for embodied control. *arXiv preprint arXiv:2209.07753*, 2022.

[28] J. Luketina, N. Nardelli, G. Farquhar, J. Foerster, J. Andreas, E. Grefenstette, S. Whiteson, and T. Rocktäschel. A survey of reinforcement learning informed by natural language. *arXiv preprint arXiv:1906.03926*, 2019.

[29] C. Lynch and P. Sermanet. Language conditioned imitation learning over unstructured data. *arXiv preprint arXiv:2005.07648*, 2020.

[30] C. Lynch, M. Khansari, T. Xiao, V. Kumar, J. Tompson, S. Levine, and P. Sermanet. Learning latent plans from play. In *Conference on robot learning*, pages 1113–1132. PMLR, 2020.

[31] 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. *arXiv preprint arXiv:2210.06407*, 2022.

[32] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. *arXiv preprint arXiv:2108.03298*, 2021.

[33] O. Mees, J. Borja-Diaz, and W. Burgard. Grounding language with visual affordances over unstructured data. *arXiv preprint arXiv:2210.01911*, 2022.

[34] O. Mees, L. Hermann, and W. Burgard. What matters in language conditioned robotic imitation learning. *arXiv preprint arXiv:2204.06252*, 2022.

[35] S. Nair, E. Mitchell, K. Chen, S. Savarese, C. 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.

[36] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

[37] M. Plappert, M. Andrychowicz, A. Ray, B. McGrew, B. Baker, G. Powell, J. Schneider, J. Tobin, M. Chociej, P. Welinder, et al. Multi-goal reinforcement learning: Challenging robotics environments and request for research. *arXiv preprint arXiv:1802.09464*, 2018.

[38] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[39] F. Röder, M. Eppe, and S. Wermter. Grounding hindsight instructions in multi-goal reinforcement learning for robotics. *arXiv preprint arXiv:2204.04308*, 2022.

[40] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[41] M. Shridhar, L. Manuelli, and D. Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on Robot Learning*, pages 894–906. PMLR, 2022.

[42] A. Silva, N. Moorman, W. Silva, Z. Zaidi, N. Gopalan, and M. Gombolay. Lancon-learn: Learning with language to enable generalization in multi-task manipulation. *IEEE Robotics and Automation Letters*, 7(2):1635–1642, 2021.

[43] S. Stepputtis, J. Campbell, M. Phielipp, S. Lee, C. Baral, and H. Ben Amor. Language-conditioned imitation learning for robot manipulation tasks. *Advances in Neural Information Processing Systems*, 33:13139–13150, 2020.

[44] S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. Teller, and N. Roy. Approaching the symbol grounding problem with probabilistic graphical models. *AI magazine*, 32 (4):64–76, 2011.

[45] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[46] X Development, LLC. Everyday Robots. http://www.everydayrobots.com, 2022. Accessed: 2022-06-15.

[47] T. Xiao, E. Jang, D. Kalashnikov, S. Levine, J. Ibarz, K. Hausman, and A. Herzog. Thinking while moving: Deep reinforcement learning with concurrent control. *arXiv preprint arXiv:2004.06089*, 2020.

[48] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani, et al. Transporter networks: Rearranging the visual world for robotic manipulation. In *Conference on Robot Learning*, pages 726–747. PMLR, 2021.

Dataset	Task Instruction Encoder	Evaluation on Novel Instructions			
		Spatial	Rephrased	Semantic	Overall
HS Only	USE	22.5%	50.0%	0.0%	21.7%
HS Only	FT CLIP	30.0%	40.0%	7.7%	27.5%
FS + HS	Pretrained CLIP	45.0%	40.0%	10.0%	40.0%
FS + HS	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 5: Comparing downstream policy performance when improving the task representation from USE [9] to Pretrained CLIP [38] to Finetuned CLIP (FT CLIP), as described in Section 3.1. We find that the FT CLIP representation is the best task representation in all dataset settings: training on crowd-sourced hindsight labels (HS), training on foresight instructions along with crowd-sourced hindsight labels (FS + HS), and using DIAL with Top-k with $k = 1$ (DIAL, $k = 1$).

A Appendix

A.1 Additional Experiments

A.1.1 Additional Qualitative Examples

In addition to the example trajectories visualized in Figure 4, we also show additional successful DIAL evaluations on novel instructions in Figure 5. Some of these scenarios are especially challenging, such as the setting with only a single coke can with instruction "push the can towards the left". Nonetheless, DIAL is not perfect, and we show some examples of failure cases in Figure 6.

A.1.2 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 Finetuned CLIP is the most effective task representation, as seen in Table 5. Finetuned CLIP is a good representation not only for freeform language instructions like those contained in the finetuning dataset $\mathcal{D}_A$, but also for structured metadata labels used to collect the demonstrations in $\mathcal{D}_B$.

A.2 DIAL Implementation Details

We implement DIAL with a CLIP model that is finetuned on 5,600 annotated episodes ($\mathcal{D}_A$) with the procedure described in Section 3.1. After finetuning CLIP, we source 18,719 candidate instruction labels ($\mathcal{L}$) 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 3 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 $\mathcal{L}$, we follow Section 3.2 to implement two variations of DIAL: **Top-k selection** and **Min-p selection**. For these two variations, we use $k = \{1, 3, 10\}$ and $p = \{0.1, 0.2, 0.3\}$.

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:
```

A.3 Instruction Augmentation Baselines

Gaussian Noise Given an instruction l, we add Gaussian noise to the language embedding produced by the CLIP text encoder T_{enc}, directly obtaining the augmentation in the latent space $\tilde{z}_l$:

$$\tilde{z}_l = T_{enc}(l) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma) \in \mathbb{R}^d \tag{4}$$

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.

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 2.

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.

A.4 Instruction Augmentation Accuracy

As described in Section 4.4, instruction prediction accuracy may decrease when increasing the number *k* of instruction augmentations. 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 correct often, the later predictions are often factually inaccurate. The top-20-th instruction prediction is only factually accurate 20.0% of the time. An example of the top 10 predictions of an episode is shown in Figure 8.

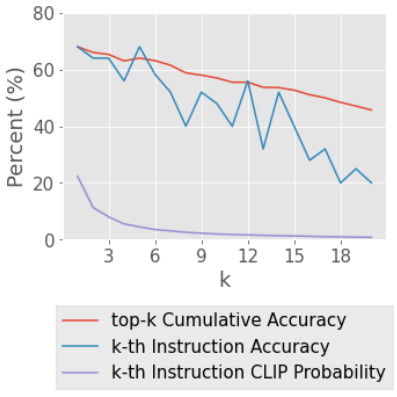


Figure 7: The accuracy of the top 20 instruction augmentation predictions of a sample of 50 episodes that have been relabeled by a Finetuned CLIP model in Section 4.2

A.5 Language-Conditioned Policy Training

The policies used in this work are trained using a Transformer architecture on a large dataset of human-provided demonstrations. The policy takes a natural language description 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. For more details on the policy, the data used to train it and the training procedure, see [1]. 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.

A.6 Evaluation Instructions

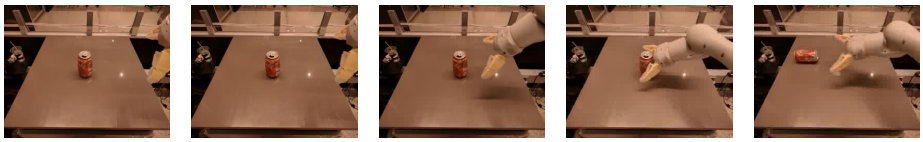
We utilize an evaluation setup focusing solely on novel instructions unseen during training. To source these novel instructions, we 1) crowd-source instructions from a different set of humans than the original dataset labelers and 2) prompt GPT-3 with Listing 3 to produce reasonable tasks that might 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 3.2 or in the original set of 551 foresight tasks in Section 4.1. Finally, as seen in Table 6, we organize them into various semantic categories to allow for more detailed analysis of quantitative policy performance.

“raise the left most can”

“move the lonely object to the others”

“lift the yellow rectangle”

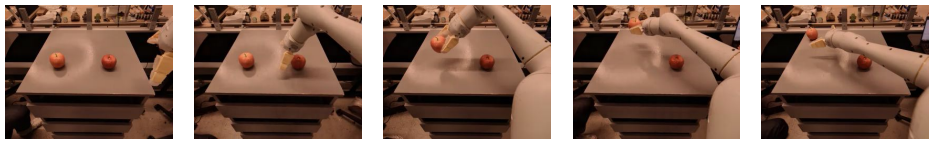
“move the right apple to the left of the counter”



“push the can towards the left”

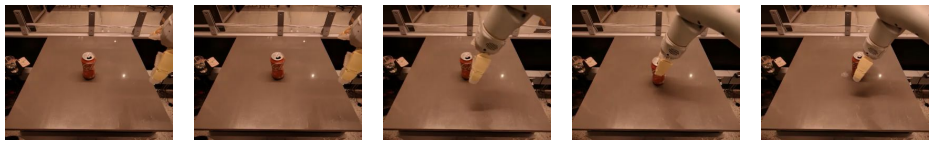
“grab the white can”

Figure 5: DIAL successfully completes various novel evaluation instructions as listed in Table 6.



“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



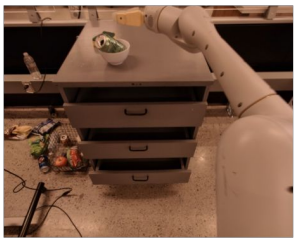
“move the can to the bottom of the table”
Failure reason: moved the can right, not downwards

“move orange near to the chip bag”
Failure reason: flipped the order

Figure 6: Samples of evaluation failures for DIAL. Errors are due to a combination of motor control and task confusion.

First Frame

Last Frame



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	❌

Figure 8: The top 10 proposed instruction augmentations for a single episode with original foresight instruction `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.

Listing 2: 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 6: 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.

Listing 3: 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>:
```