Interactive Language: Talking to Robots in Real Time

Corey Lynch, Ayzaan Wahid, Jonathan Tompson
Tianli Ding, James Betker, Robert Baruch, Travis Armstrong, Pete Florence

Robotics at Google

Abstract— We present a framework for building interactive, real-time, natural language-instructable robots in the real world, and we open source related assets (dataset, environment, benchmark, and policies). Trained with behavioral cloning on a dataset of hundreds of thousands of language-annotated trajectories, a produced policy can proficiently execute an order of magnitude more commands than previous works: specifically we estimate a 93.5% success rate on a set of 87,000 unique natural language strings specifying raw end-to-end visuo-linguo-motor skills in the real world. We find that the same policy is capable of being guided by a human via real-time language to address a wide range of precise long-horizon rearrangement goals, e.g. *"make a smiley face out of blocks"*. The dataset we release comprises nearly 600,000 language-labeled trajectories, an order of magnitude larger than prior available datasets. We hope the demonstrated results and associated assets enable further advancement of helpful, capable, natural-language-interactable robots. See videos at https://interactive-language.github.io.

I. INTRODUCTION

The goal of building a robot that can follow a diverse array of natural language instructions has been a longstanding goal of AI research, since at least the SHRDLU [1] experiments starting in the late 1960s. While recent research on this topic has been abundant [2]–[9], few efforts have actually produced a robot that (i) exists in the real world, and (ii) can capably respond to a large number of rich, diverse language commands. We expect that future research will continue to produce larger and more diverse sets of behaviors, either by sequencing raw skills together [10] or growing the number of raw skills themselves [11]. However, we are also interested in (iii), the capacity to follow *interactive* language commands, by which we mean that the robot reacts capably and in-the-moment to new natural language instructions provided during ongoing task execution. Although we might expect such a robot to be possible given current methods, natural language-interactable robots are frequently slow in practice, and often use blocking parameterized skills [7], [10] or simplifying self-resetting behaviors [9], [12] that prohibit this kind of live, real-time interaction.

In this paper, we demonstrate a framework for producing real-world, real-time-interactable, natural-language-instructable robots (Fig. 1, a) that by certain metrics operate at an order of magnitude larger scale than prior works. To accelerate further research in this setting, we accordingly provide our associated recipe, dataset, models, hardware environment description, simulated analogue environment, and a research benchmark for language conditioned manipulation (Fig. 1, c). In terms of scale, the produced robot policies can address 87,000 unique commands at an estimated 93.5% success rate (Fig. 1, b), with continuous 5Hz visuolinguomotor control, and are capable of chaining raw skills to reach hundreds of thousands of long horizon goals in its environment.
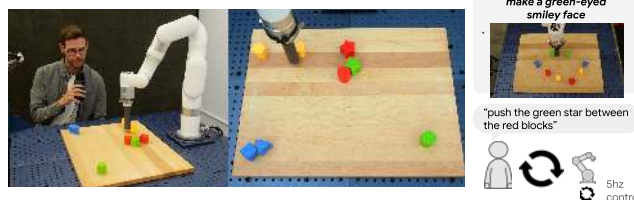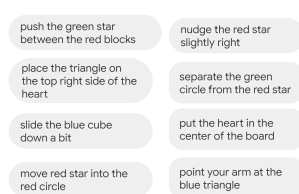
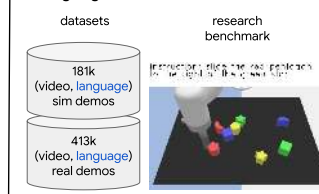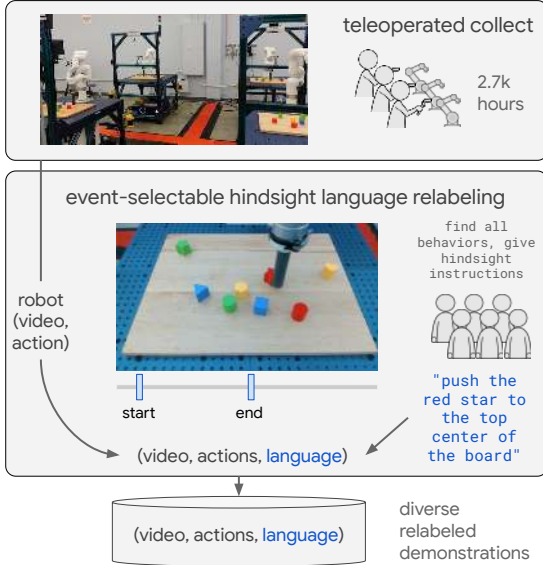**Fig. 1: Real-time language, diverse robot behaviors.** a) Over the course of 5 minutes, a human guides a robot to precisely rearrange objects a table into a desired shape, with real-time natural language as the only mechanism for specifying behaviors. b) We demonstrate a single robot that can capably address 87,000 behaviors specified entirely in natural language. c) We release Language-Table, a suite of human-collected datasets and a multi-task continuous control benchmark for open vocabulary visuolinguomotor learning.

This robot exists in an environment which we designed to provide a tractable yet difficult level of challenge (perception from pixels, feedback-rich control, multiple objects, ambiguous natural language instructions). We cast real time language guidance as a large scale imitation learning problem [11], [13], [14] (Figure 2). The learning algorithm recipe itself is intentionally simple, and instead the complexity of this effort was primarily in the data effort itself, for which we detail insights and techniques. We hope the dataset and benchmark may catalyze further work which may improve on our demonstrated sample complexity and performance.
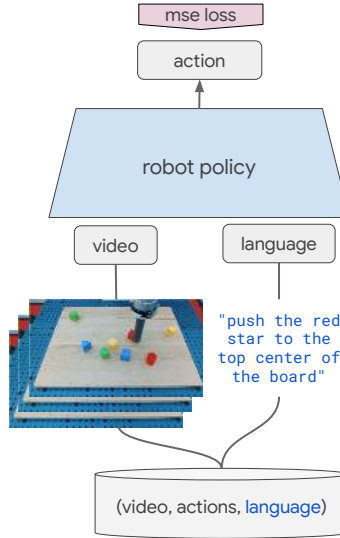
Beyond demonstrating diverse short-horizon skills, we also use these capabilities to study the nonobvious benefits of a real-time language robot. For one, we show that through occasional human natural-language feedback, the robot can accomplish complex long-horizon rearrangements such as *"put the blocks into a smiley face with green eyes"* that require multiple minutes of precise coordinated control (Figure 5, left). We also find that real-time language competency unlocks new capabilities like simultaneous, multi-robot instruction – in which a single human can guide multiple real-time robots through long-horizon tasks (Figure 5, right).

Contributions. Our primary contributions include (i) Interactive Language, a framework for producing real world robots that can capably receive interactive open vocabulary language condi-
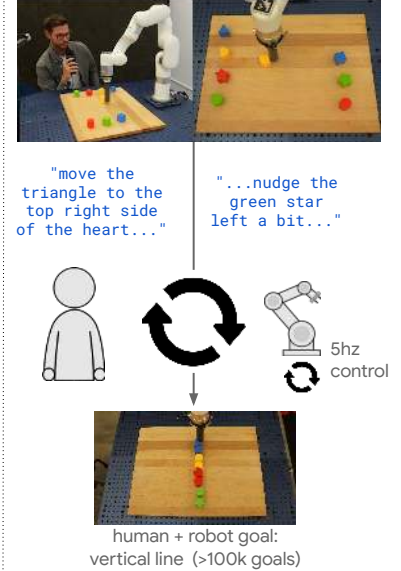
1) High-throughput teleoperation + hindsight language relabeling

teleoperated collect

2.7k hours

event-selectable hindsight language relabeling

find all behaviors, give hindsight instructions

robot (video, action)

start   end

"push the red star to the top center of the board"

(video, actions, language)

(video, actions, language)

diverse relabeled demonstrations

2) Language conditioned behavioral cloning (LCBC)

mse loss

action

robot policy

video   language

"push the red star to the top center of the board"

(video, actions, language)

3) Human + robot solve goals with real time language

"move the triangle to the top right side of the heart..."

"...nudge the green star left a bit..."

5hz control

human + robot goal: vertical line (>100k goals)

Fig. 2: Interactive Language, a large scale robot imitation learning framework for real-time language. Stage 1: First, high throughput robot data collection with multiple operators. Post-collection, relabel robot video and actions into language conditioned demonstrations using event-selectable hindsight relabeling. Stage 2: do simple language conditioned behavioral cloning. Stage 3: Human guides a single learned policy in real-time using natural language to accomplish hundreds of thousands of goals.

tioning in real-time[1] while performing continuous-control visuomotor manipulation. Interactive Language combines existing techniques, together with novel components like event-selectable hindsight relabeling, to define a simple and scalable recipe for learning large repertoires of natural-language-conditionable skills. (ii) We use this system to present and study the setting of *interactive language guidance*, showing that the combination of real-time language feedback and a low-level language-conditionable policy can address long-horizon manipulation goal states in a tabletop rearrangement setting. (iii) To facilitate future research in this domain, we release *Language-Table*, a dataset and simulated multitask imitation learning benchmark. With nearly 600,000 diverse demonstrations across simulation and the real world, Language-Table is, to our knowledge, the largest natural language conditioned imitation learning dataset of its kind by an order of magnitude (Table III).

II. RELATED WORK

From single-task imitation to multi-task and language conditioning. Imitation learning (see review [14]), the perspective we adopt in this work, provides a simple and stable way for robots to acquire behaviors from human expert demonstrations. While historically imitation learning has been applied to individual tasks from instrumented state [16]–[19], the desire for more general purpose robots has motivated study into policies capable of learning multiple skills at once from more generic on-board sensory observations like RGB pixels [20]–[22]. To condition multiple learned behaviors, prior setups have relied on discrete one-hot task identifiers [23], which can be difficult to scale to many tasks, or goal images [24]–[26], which can be impractical to provide in real world scenarios. Alternatively, a long history of

prior work in broader AI research [1]–[6], [11] has sought a more convenient form of specification in the form of natural language conditioning (survey [27]), with some results on physical robots [7]–[9], [12]. This focus has yielded many varied and impressive approaches to tackling the grounding problem [1], [28]—learning to relate language to one's embodied observations and actions. However, in both simulation and the real world, instruction-following robots rarely leverage the full capabilities of continuous control, instead employing simplified, parameterized action spaces [6], [7], [29], [30]. Furthermore, once provided, language conditioning is typically presumed fixed over robot execution [8]–[10], [12], with little opportunity for subsequent interaction by the instructor. Our work, in contrast, studies the first combination, to our knowledge, of real-time natural language guidance of a physical robot engaged in continuous visuomotor manipulation.

Interactively guiding robot behavior with language. Our work exists in a larger setting of humans modifying or correcting the behavior of autonomous agents [31], historically addressed in forms like teleoperation [32]–[34], kinesthetic teaching [35], or sparse human preference feedback [36]. Certain works have studied language as a means of correction, but typically do so under simplifying assumptions that we relax in the current work. For example, [37], [38], [39], and [40] study language corrections, but under the respective simplifying assumptions of hand-defined optimization for grounding, undivided operator attention, paired iterative corrections at training time, and presumed access to motion planners and task cost functions. Additionally, to the best of our knowledge, none of these works support multiple-Hz iterative specification over the course of execution. Closest to our approach is [11] and [30], which study language-interactive agents learned via imitation, but entirely in simulation and under varying degrees of actuation realism. In contrast to these prior studies,

---

[1] For the scope of this paper, by real-time we mean new language conditioning can occur in the "blink of an eye", i.e. approximately 3 Hz [15] or greater.

our work learns real-time natural language policies end-to-end from RGB pixels to continuous control outputs with a simple behavioral cloning objective [13], and applies them to contact-rich real-world manipulation tasks.

Scaling real world imitation learning. One of the largest bottlenecks in robot imitation is often simply the amount of diverse robot data made available to learning [9], [22], [23]. Many multi-task imitation learning frameworks determine the set of tasks to be learned upfront [7], [9], [10], [12], [14]. While this may simplify collection conceptually, it also often requires that reset protocols and success criteria be designed manually for each behavior. Another challenge particular to large scale multi-operator collections is that typically not all data can be considered optimal [41], [42], often requiring manual post-hoc success filtering [9], [10]. These per-task manual efforts have historically been difficult to scale to a large and diverse task setting, like the one studied in this work. We sidestep both these scaling concerns by instead having operators continuously teleoperate long-horizon behaviors, with no requirements on low level task segmentation or resets [11], [25], [43] and then leverage after-the-fact crowdsourced language annotation [8], [11]. In contrast to the "random window" relabeling explored in [11], we give annotators precise control over the start and end of behaviors they are annotating, which we find in practice better aligns relabeled training data to the actual commands given at test time.

III. PROBLEM SETUP

Our goal is to train a conditional policy, $\pi_\theta(a|s,\ l)$, parameterized by θ, which maps from observations $s \in \mathcal{S}$ and human-provided language $l \in \mathcal{L}$ to actions $a \in \mathcal{A}$ on a physical robot. In particular we are interested in *open-vocabulary language-conditioned visuomotor policies*, in which the observation space contains high-dimensional RGB images, e.g. $= \mathbb{R}^{H \times W \times C}$, and where language conditioning $\mathcal{L}$ has no predefined template, grammar, or vocabulary. We are also particularly interested in allowing humans to interject new language $\mathcal{L}$ at any time, at the natural rate of the visuo-linguo-motor, in which commanded l encodes a distribution of achievable goals $\mathcal{G}^{short} \in \mathcal{G}^{short}$ in the environment. Note that humans may generate a new language instruction l based on their own perception of the environment, $s^H \in \mathcal{S}^H$, which may differ substantially from the robot's $s \in \mathcal{S}$ (e.g. due to viewpoint, self-occlusion, limited observational memory, etc.). As in prior works [11], we treat natural-language-conditioned visuomotor skill learning as a contextual imitation learning problem [14]. As such, we acquire an offline dataset $\mathcal{D}$ containing pairs of valid demonstrations and the conditions they resolve $\{(\tau, l)_i\}_{i=0}^{\mathcal{D}}$. Each τ_i is a variable-length trajectory of robot observations and actions $\tau_i = [(s_0, a_0), (s_1, a_1), ..., (s_T)]$, and each l_i describes the full trajectory as a second-person command.

IV. INTERACTIVE LANGUAGE: METHODS AND ANALYSIS

First we introduce *Interactive Language*, summarized in Figure 2, a simple and generically applicable imitation learning framework for training real-time natural-language-interactable robots. Interactive Language combines a scalable method for collecting varied, real world language-conditioned demonstration datasets, with straightforward language conditioned behavioral cloning (LCBC).

	Has contact	Object/location-directed instructions	Compound instructions
Random window [8], [11]	86%	47%	16%
Event-selectable (ours)	91%	83%	< 1%
Real test instructions	89%	84%	< 1%

TABLE I: Which relabeling strategy aligns best with test-time language?

Real-World Data Collection	
Total robots	4
Total teleoperators	10
Total episodes	16.4k
Average episode length (minutes)	9.9
Total hours of collect time	2.7k
Hindsight Relabeling	
Total crowdsourced annotators	64
Total relabeled demonstrations obtained	299k
Total unique relabeled instructions	87k
Average relabeled demonstration length (seconds)	5.8
Total number of hours of relabeled demonstrations obtained	488
Total instruction hours / Collect hours	18.06%

TABLE II: Statistics: real-world collection and relabeling. This data snapshot went into training and is a subset of the full Language-Table data.

A. Data Collection

High throughput raw data collection. Interactive Language adopts purposefully minimal collection assumptions to maximize the flow of human demonstrated behavior to learning. Operators teleoperate a variety of long-horizon behaviors constantly, without low-level task definition, segmentation, or episodic resets. This strategy shares assumptions with "play" collection [25], but additionally guides collect towards temporally extended low-entropy states like lines, shapes, and complex arrangements. Each collect episode lasts ~10 minutes before a break, and is guided by multiple randomly chosen long-horizon prompts $p \in \mathcal{P}$ (e.g. "*make a square shape out of the blocks*"), drawn from the set of target long-horizon goals, which teleoperators are free to follow or ignore. We do not assume all of the data collected for each prompt p is optimal (each p is discarded after collecting). In practice, our collection includes many inevitable edge cases that might otherwise require data cleaning, e.g. solving for the wrong p or knocking blocks off table. We log all of these cases and incorporate them later on as training data. Concretely, this collect procedure yields a *semi-structured, optimality-agnostic* collection $\mathcal{D}_{collect} = \{\tau_i\}_{i=0}^{\mathcal{D}_{collect}}$. The purpose of $\mathcal{D}_{collect}$ is to provide a sufficiently diverse basis for crowdsourced hindsight language relabeling [8], [11], described next.

Event-selectable hindsight relabeling. We convert $\mathcal{D}_{collect}$ into natural language conditioned demonstrations $\mathcal{D}_{training} = \{(\tau, l)_i\}_{i=0}^{\mathcal{D}_{training}}$, using a new variant of hindsight language relabeling [11] we call "Event-Selectable Hindsight Relabeling" (Fig.2, left). Previous "random window" relabeling systems [8], [11] have at least two drawbacks: each random window is not guaranteed to contain "usefully describable" actions, and random window lengths must be determined upfront as a sensitive hyperparameter. We instead ask annotators to watch the full collect video, then find K coherent behaviors ($K = 24$ in our case). Annotators have the ability to mark the start and end frame of each behavior, and are asked to phrase their text descriptions as natural language commands. In Table I, we compare event-selectable relabeling to

prior "random window" relabeling on a subset of our training data. We find that while both strategies tend to describe contact-rich behaviors, our analysis suggests event-selectable relabeling yields more well-matched data: fewer complex compound instructions, and more compositionally directed instructions.

Throughput and bottleneck analysis. Here, we share some insights gained from scaling our robot collect and hindsight relabeling operation. See statistics on our collected data in Table II. We find, perhaps surprisingly, that the main bottleneck in our data operation is *not* robot teleoperation but rather the crowdsourced language annotation that follows, with 18.06% of the raw data having undergone annotation prior to model training (5.5x as much unlabeled collected data as annotated data). This is true even though there are 16x as many hindsight annotators as robots. Bottlenecks like this may be addressed by exploiting language-free co-training [11], or by simply continuing to horizontally scale crowdsourced annotators.
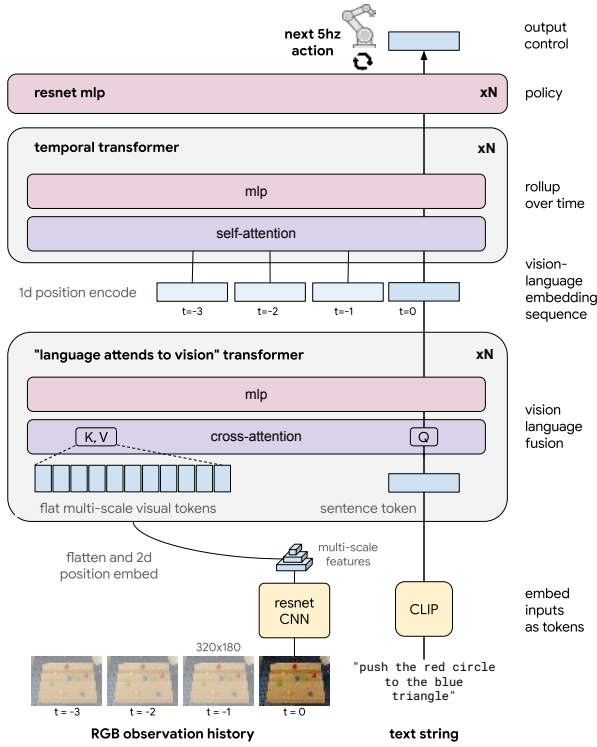

Fig. 3: LAVA: our transformer-based architecture for language conditioned visuomotor control.

B. Policy Learning

Transformer-based agent architecture. In Figure 3, we describe our transformer-based [44] neural network policy architecture, mapping from video and text to continuous actions, which we refer to as LAVA ("Language Attends to Vision to Act"). Each training example consists of $(s, a, l)_i \sim \mathcal{D}_{\text{training}}$, where $s \in \mathbb{R}^{\text{seqlen} \times 640 \times 320 \times 3}$ is RGB observation history. We ConvNet-process each frame in the video s to obtain multi-scale visual features (features at multiple resolutions). The first two layers are Imagenet-pretrained ResNet [45], [46]. l is embedded using a pretrained CLIP text encoder [47], which is finetuned on our in-domain data, but remains fixed during policy training. We fuse visual and lingual information using a "Language-Attends-to-Vision"

Dataset	# Traj. (k)	# Unique (k)	Physical Actions	Real	Available
Episodic Demonstrations					
BC-Z [9]	25	0.1	✓	✓	✓
SayCan [10]	68	0.5	✓	✓	✗
Playhouse [30]	1,097	779	✗	✗	✗
Hindsight Language Labeling					
BLOCKS [50], [51]	30	n/r	✗	✗	✓
LangLFP [11]	10	n/r	✓	✗	✗
LOREL [8], [52]	6	1.7	✓	✓	✓
CALVIN [53]	20	0.4	✓	✗	✓
Language-Table	**594**	**198**	✓	✓	✓
(*real*+*sim*)	(*413*+*181*)	(*119*+*79*)			

TABLE III: Comparison of human-guided, language-labeled trajectory datasets. Highlighted are the number of language-labeled trajectories and number of unique language labels (k=thousands) in real and sim, along with whether the data uses physical actions, real-world data, and if it is publicly available. n/r means not reported.

transformer block, which performs cross-attention with language acting as query, and flattened multi-scale visual tokens acting as keys and values. This operation is applied to each image, and the sequence output is fed to a temporal prenorm [48] transformer, which is average pooled and fed to a deep residual multi-layer perceptron (MLP), outputting the predicted next action a.

Training. We train our policies with a standard supervised language conditioned behavioral cloning (LCBC) objective. While we expect that more complex loss functions or policy classes may acquire even better results, all the policies we present were trained as deterministic policies with a simple mean squared error loss: $\min_\theta \sum_{(s,a,l)\sim\mathcal{D}_{\text{training}}} ||a - \pi_\theta(s, l)||_2^2$, e.g. as in [9], [49].

V. LANGUAGE-TABLE: DATASETS AND ENVIRONMENT

To facilitate further research in language-conditioned visuomotor learning, we release *Language-Table*, which consists of (i) a suite of datasets and (ii) a simulated multi-task language conditioned control environment and benchmark.

Dataset. Language-Table provides our human-relabeled $\mathcal{D}_{\text{training}}$ and the underlying human-teleoperated $\mathcal{D}_{\text{collect}}$, both in simulation and the real world. The $\mathcal{D}_{\text{training}}$ real and sim datasets are highlighted in Table III – an order of magnitude larger than comparable, previously-available datasets.

Environment and Benchmark. Language-Table's simulated environment resembles our real-world tabletop manipulation scenario, which consists of an xArm6 robot, constrained to move in a 2D plane with a cylindrical end-effector as in [54], in front of a smooth wooden board with a fixed set of 8 plastic blocks, comprising 4 colors and 6 shapes (Fig. 5). In both simulation and real collection, we use high-rate human teleoperation with a 3rd person view (line-of-sight in real). Actions are 2D delta Cartesian setpoints, from the previous setpoint to the new one. We batch collected training and inference data to 5hz observations and actions.

The Language-Table benchmark computes automated metrics for 5 task families, with 696 unique task variations. In addition to thresholded task success, a metric we find that better correlates with human-preferred performance is Success weighted by Path Length (SPL) [55], which trades off success rate against the efficiency of the path it took to succeed. We note that policy hyperparameters ordered by SPL in Language-Table have thus far

Short-Horizon Instruction	Success
(87k more...)	...
push the blue triangle to the top left corner	80.0%
separate the red star and the red circle	100.0%
nudge the yellow heart a bit right	80.0%
place the red star above the blue cube	90.0%
point your arm at the blue triangle	100%
push the group of blocks left a bit	100%
Average over 87k, CI 95%	**93.50% +- 3.42%**

TABLE IV: Real world: Evaluating a wide variety of short-horizon language conditionable skills. 95% Confidence interval on the average success of our single policy over 87k (Table II) unique natural language instructions.

been ordered similarly in real-world performance. This provides a degree of validation for the simulated benchmark's relevancy to real world robotics.

VI. POLICY RESULTS AND DISCUSSION

We present experiments aimed at answering the following questions: (1) How capably can the system follow a wide variety of short-horizon natural language conditioned commands? (2) How capably can these skills be composed through interactive language guiding to accomplish a wide variety of multi-step long-horizon compositional rearrangements? (3) What is the benefit of being able to provide *interactive* language feedback, compared to open-loop language plans? (4) Can one operator simultaneously guide several robots equipped with our policy? (5) Ablations: How does our transformer-based policy architecture compare to an existing visuo-linguo-motor baseline? How does our presented approach scale with varying amounts of data?

A. Real world: diverse short-horizon language conditionable skills

Ideally, we would be able to evaluate an Interactive Language policy on *any* short-horizon command a real human might give it, which is intractable in general. As a surrogate, we estimate a 95% confidence interval on average success over the 87,588 unique language instructions collected via crowdsourcing (20 randomly selected instructions, 10 trials each) available at time of analysis (Table II). To succeed, policies must ground object properties and compositional spatial concepts (e.g. "...*top right side* of the *yellow hexagon*" vs "*top right side* of the board"), and resolve difficult ambiguities (e.g. "*nudge* the cube left *a bit*"). We report results in Table IV, with examples in Figure 4. We see that Interactive Language obtains a 93.5% expected average success rate over all 87,588 instructions, 95% CI [90.08%, 96.92%]. To our knowledge, this is the largest set of language conditioned behaviors a real-world policy has been shown to capably address, demonstrating a solid base capacity for language conditioned visuomotor control.

B. Real world: long-horizon real-time language guidance

Long horizon goal reaching. Next we aim to see whether humans can guide Interactive Language policies through a wide range of multi-step compositional rearrangements. We define over 100,000 language-distinct compositional goal states on our tabletop from 11 high level families (e.g. make high-level shapes, sort by color, place all blocks in specific locations, arrange into lines, etc.), then sample 20 uniformly from all 11. See Figure 5 for examples of different goal states. We evaluate each long horizon
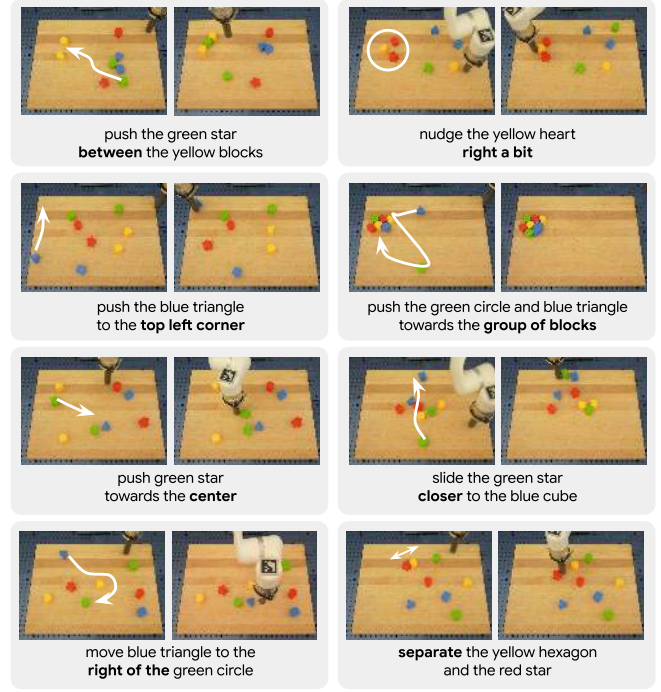

Fig. 4: Learning a wide variety of short-horizon open vocabulary behaviors. Interactive Language rollouts on a sample of the >87,000 crowdsourced natural language instructions we evaluate.

goal 3 times from randomly reset board states, yielding 60 total evaluations of a single policy. We report success rates in Table V. We see that our policy obtains an 85.0% expected average success rate on this diverse set of goals, 95% CI [69.35%, 100.00%]. These results are best appreciated by watching the supplementary videos. We note that reaching precise long horizon goals in the real world for even a single goal is a notoriously difficult problem for learning robots [43]. Even though our policies do not do so fully autonomously, we believe the fact that a real robot can address such a large and varied set of goals with real-time language feedback suggests a synergistic mode of future operation (at least until large improvements are made in the fully autonomous setting): robots learn a set of general-purpose low-level skills, and humans put them together in a familiar way using natural language, interrupting at any time to offer situation-specific corrections.

Open-loop vs real-time language feedback. Next, we attempt to quantify the benefit of being able to provide *real-time* language feedback, over the more common "open-loop" evaluation setting where the sequence of subgoals is decided up front [10]–[12], [43]. We hypothesize that many of the tasks in our environment might require several rounds of iterative and interactive specification, due to the stochastic nature of single-point-of-contact pushing. We perform the same evaluation as in the previous section, but the human operator commits up front to the set and order of commands they will provide. We present results for this ablation in Table V, finding that performance deteriorates from 85% to 25% when real-time language is removed. This indicates that for contact-rich tasks like the ones studied in this work, success depends heavily on sufficient *real-time feedback*—not only for the low-level policy, but also for the agent providing it instructions.

Multi-robot control via spoken language. Finally, we investigate a new competency afforded by Interactive Language:
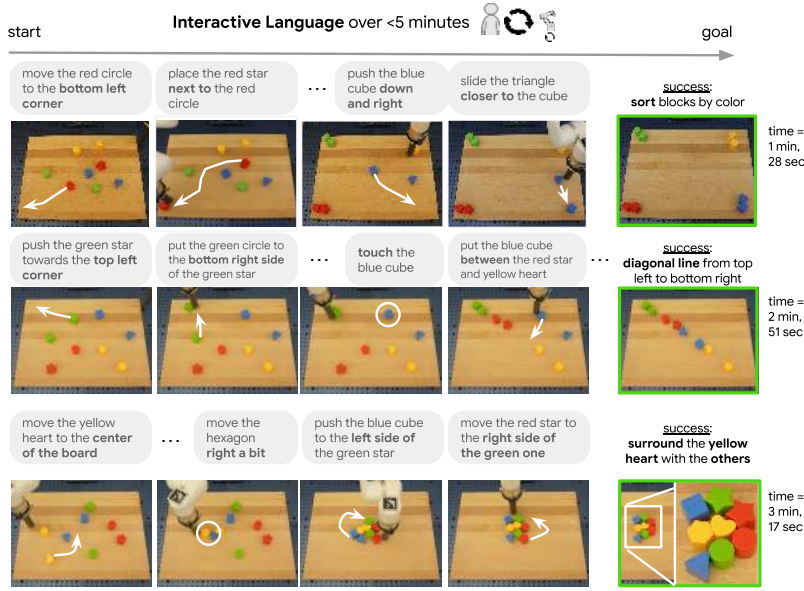
Fig. 5: Capabilities explored with Interactive Language. Left: **Long-horizon language guidance** allows a human to guide a single policy to achieve a wide variety of long-horizon precise rearrangement goals. Language is used to interject new subgoals on-the-fly, to offer real-time corrections of unsafe or undesirable behavior (e.g. "move the block away from the edge"), or to constrain the motions of the agent (e.g. "slide the triangle *slowly* left"). We evaluate policies on 11 goal families spanning hundreds of thousands of tasks. Right: **Simultaneous multi-robot control**. Real time language allows a single human operator to guide multiple robots at once through the same long-horizon task, without requiring undivided attention to any one robot.

Language interaction style	Average number of instructions provided	Long-horizon success %
Open-loop	6.5	25.0% +- 18.98%
Real-time (ours)	15	**85.0% +- 15.65%**

TABLE V: Real world: long-horizon goal reaching via real-time human language guidance. 95% Confidence interval on the average success of our single real-time policy over 11 families and 100k possible goals, as compared to an open-loop baseline.
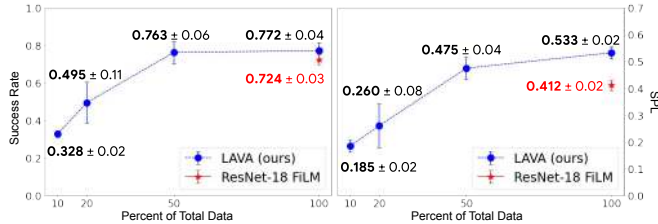
Fig. 6: Ablations in simulation. We compare our LAVA transformer architecture to a baseline ResNet-18 FiLM model from [9], as well as ablate the amount of data provided to training. We find the average success-weighted path length (SPL) to be a better indicator of qualitative performance than (unweighted) average success.

simultaneous multi-robot control. In Figure 5, see video as well, we see that four robots equipped with Interactive Language policies can be guided at the same time by one operator. This language guided multi-robot control is, as far as we know, a capability not demonstrated in the literature. Importantly, due to short-horizon skill competency, this shows that language can relax the assumption of undivided operator attention, which is common for prior ways of correcting online robot behavior [32], [34], [56].

C. Simulation: Architecture and data ablation

In Figure 6, we present results in simulation ablating (i) our transformer-based policy architecture LAVA against the FiLM-conditioned ResNet architecture in [9] and (ii) the amount

of data provided to policy training. We report average success and SPL [55] over the multi-task benchmark in Language-Table (see "Environment and Benchmark" in Section V), and all numbers are reported with confidence intervals over three seeded training runs. We see the presented architecture is indeed responsible for significant gains over prior work in SPL, a path-length-aware success metric we find correlates best with real world quality in our setup. When sweeping the amount of training data, we find that policy performance is seeing diminishing returns, but not yet plateauing across each doubling of data. While perhaps surprising given the scale of our collect, we believe that this result highlights the environment's complexity as well as the difficulty of open vocabulary visuomotor learning.

VII. CONCLUSION, LIMITATIONS, AND FUTURE WORK

We have presented and analyzed the Interactive Language framework and we provide a number of associated assets, notably the Language-Table dataset and environment. We believe the scale of the dataset assets, the recipe used to produce them, the scale of the demonstrated policy diversity, and the exploration of new capabilities, each offer benefit to the research community in further advancing capable, realtime-conditionable visuo-linguo-motor robots. While simple and scalable, our approach does have a number of limitations. The open problems in broader human-robot collaboration are numerous [57], including intention detection, non-verbal communication, physically collaborative task completion, etc. Our approach addresses only the setting of real-time language-guided manipulation. Future work may investigate applying Interactive Language to important domains like real-time assistive robots, which may benefit from more capable natural language interfaces [37]. We hope that our work can be useful as a basis for future research in capable, helpful robots with visuo-linguo-motor control.

REFERENCES

[1] T. Winograd, "Understanding natural language," *Cognitive psychology*, vol. 3, no. 1, pp. 1–191, 1972.

[2] S. R. Branavan, H. Chen, L. Zettlemoyer, and R. Barzilay, "Reinforcement learning for mapping instructions to actions," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, 2009, pp. 82–90.

[3] S. Tellex, T. Kollar, S. Dickerson, M. Walter, A. Banerjee, S. Teller, and N. Roy, "Understanding natural language commands for robotic navigation and mobile manipulation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 25, no. 1, 2011, pp. 1507–1514.

[4] D. Chen and R. Mooney, "Learning to interpret natural language navigation instructions from observations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 25, no. 1, 2011, pp. 859–865.

[5] C. Matuszek, E. Herbst, L. Zettlemoyer, and D. Fox, "Learning to parse natural language commands to a robot control system," in *Experimental robotics*. Springer, 2013, pp. 403–415.

[6] F. Hill, A. Lampinen, R. Schneider, S. Clark, M. Botvinick, J. L. McClelland, and A. Santoro, "Emergent systematic generalization in a situated agent," *arXiv preprint arXiv:1910.00571*, 2019.

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

[8] 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*. PMLR, 2022, pp. 1303–1315.

[9] 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*. PMLR, 2022, pp. 991–1002.

[10] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," *arXiv preprint arXiv:2204.01691*, 2022.

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

[12] 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*, vol. 33, pp. 13 139–13 150, 2020.

[13] D. A. Pomerleau, "Efficient Training of Artificial Neural Networks for Autonomous Navigation," *Neural Comput.*, vol. 3, 1991.

[14] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, J. Peters *et al.*, "An algorithmic perspective on imitation learning," *Foundations and Trends® in Robotics*, vol. 7, no. 1-2, pp. 1–179, 2018.

[15] K.-A. Kwon, R. J. Shipley, M. Edirisinghe, D. G. Ezra, G. Rose, S. M. Best, and R. E. Cameron, "High-speed camera characterization of voluntary eye blinking kinematics," *Journal of the Royal Society Interface*, vol. 10, no. 85, p. 20130227, 2013.

[16] D. A. Pomerleau, "Alvinn: An Autonomous Land Vehicle in a Neural Network," Carnegie-Mellon University, Tech. Rep., 1989.

[17] S. Schaal, A. Ijspeert, and A. Billard, "Computational approaches to motor learning by imitation," *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, vol. 358, no. 1431, pp. 537–547, 2003.

[18] S. M. Khansari-Zadeh and A. Billard, "Learning stable nonlinear dynamical systems with gaussian mixture models," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 943–957, 2011.

[19] S. Schaal, J. Peters, J. Nakanishi, and A. Ijspeert, "Learning movement primitives," in *Robotics research. the eleventh international symposium*. Springer, 2005, pp. 561–572.

[20] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[21] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.

[22] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel, "Deep imitation learning for complex manipulation tasks from virtual reality teleoperation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 5628–5635.

[23] R. Rahmatizadeh, P. Abolghasemi, L. Bölöni, and S. Levine, "Vision-based multi-task manipulation for inexpensive robots using end-to-end learning from demonstration," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 3758–3765.

[24] Y. Ding, C. Florensa, P. Abbeel, and M. Phielipp, "Goal-conditioned imitation learning," *Advances in neural information processing systems*, vol. 32, 2019.

[25] 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*. PMLR, 2020, pp. 1113–1132.

[26] 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.

[27] 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.

[28] K. M. Hermann, F. Hill, S. Green, F. Wang, R. Faulkner, H. Soyer, D. Szepesvari, W. M. Czarnecki, M. Jaderberg, D. Teplyashin *et al.*, "Grounded language learning in a simulated 3d world," *arXiv preprint arXiv:1706.06551*, 2017.

[29] X. Wang, Q. Huang, A. Celikyilmaz, J. Gao, D. Shen, Y.-F. Wang, W. Y. Wang, and L. Zhang, "Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6629–6638.

[30] D. I. A. Team, J. Abramson, A. Ahuja, A. Brussee, F. Carnevale, M. Cassin, F. Fischer, P. Georgiev, A. Goldin, T. Harley *et al.*, "Creating multimodal interactive agents with imitation and self-supervised learning," *arXiv preprint arXiv:2112.03763*, 2021.

[31] S. Tellex, R. Knepper, A. Li, D. Rus, and N. Roy, "Asking for help using inverse semantics," 2014.

[32] D. Rakita, B. Mutlu, and M. Gleicher, "An autonomous dynamic camera method for effective remote teleoperation," in *2018 13th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2018, pp. 325–333.

[33] M. Kelly, C. Sidrane, K. Driggs-Campbell, and M. J. Kochenderfer, "Hg-dagger: Interactive imitation learning with human experts," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8077–8083.

[34] J. Spencer, S. Choudhury, M. Barnes, M. Schmittle, M. Chiang, P. Ramadge, and S. Srinivasa, "Learning from interventions: Human-robot interaction as both explicit and implicit feedback," in *16th Robotics: Science and Systems, RSS 2020*. MIT Press Journals, 2020.

[35] P. Kormushev, S. Calinon, and D. G. Caldwell, "Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input," *Advanced Robotics*, vol. 25, no. 5, pp. 581–603, 2011.

[36] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," *Advances in neural information processing systems*, vol. 30, 2017.

[37] A. Broad, J. Arkin, N. Ratliff, T. Howard, B. Argall, and D. C. Graph, "Towards real-time natural language corrections for assistive robots," in *RSS Workshop on Model Learning for Human-Robot Communication*, 2016.

[38] S. Karamcheti, M. Srivastava, P. Liang, and D. Sadigh, "Lila: Language-informed latent actions," in *Conference on Robot Learning*. PMLR, 2022, pp. 1379–1390.

[39] J. D. Co-Reyes, A. Gupta, S. Sanjeev, N. Altieri, J. Andreas, J. DeNero, P. Abbeel, and S. Levine, "Guiding policies with language via meta-learning," *arXiv preprint arXiv:1811.07882*, 2018.

[40] P. Sharma, B. Sundaralingam, V. Blukis, C. Paxton, T. Hermans, A. Torralba, J. Andreas, and D. Fox, "Correcting robot plans with natural language feedback," *arXiv preprint arXiv:2204.05186*, 2022.

[41] 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.

[42] M. Yang, S. Levine, and O. Nachum, "Trail: Near-optimal imitation learning with suboptimal data," *arXiv preprint arXiv:2110.14770*, 2021.

[43] A. Gupta, V. Kumar, C. Lynch, S. Levine, and K. Hausman, "Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning," *arXiv preprint arXiv:1910.11956*, 2019.

[44] 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*, vol. 30, 2017.

[45] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[47] 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*. PMLR, 2021, pp. 8748–8763.

[48] R. Xiong, Y. Yang, D. He, K. Zheng, S. Zheng, C. Xing, H. Zhang, Y. Lan, L. Wang, and T. Liu, "On layer normalization in the transformer architecture," in *International Conference on Machine Learning*. PMLR, 2020, pp. 10 524–10 533.

[49] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel, "Deep Imitation Learning for Complex Manipulation Tasks from Virtual Reality Teleoperation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[50] Y. Bisk, D. Yuret, and D. Marcu, "Natural language communication with robots," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 751–761.

[51] Y. Bisk, K. J. Shih, Y. Choi, and D. Marcu, "Learning interpretable spatial operations in a rich 3d blocks world," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[52] B. Wu, S. Nair, F.-F. Li, and C. Finn, "Example-driven model-based reinforcement learning for solving long-horizon visuomotor tasks," in *5th Annual Conference on Robot Learning*, 2021. [Online]. Available: https://openreview.net/forum?id=_daq0uh6yXr

[53] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard, "Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks," *IEEE Robotics and Automation Letters*, 2022.

[54] P. Florence, C. Lynch, A. Zeng, O. A. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson, "Implicit behavioral cloning," in *Conference on Robot Learning*. PMLR, 2022, pp. 158–168.

[55] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva *et al.*, "On evaluation of embodied navigation agents," *arXiv preprint arXiv:1807.06757*, 2018.

[56] A. Bajcsy, D. P. Losey, M. K. O'Malley, and A. D. Dragan, "Learning from physical human corrections, one feature at a time," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 2018, pp. 141–149.

[57] B. Hayes and B. Scassellati, "Challenges in shared-environment human-robot collaboration," *learning*, vol. 8, no. 9, 2013.

[58] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," *GitHub Repository*, 2016.

[59] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[60] 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.

[61] 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.

[62] 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.

A. Additional real-world experiment details

Our real-world experiments use UFACTORY xArm6 robot arms with all state logged at 100 Hz. Observations are recorded from an Intel RealSense D415 camera, using RGB-only images at 640x360 resolution, logged at 30 Hz, which we resize to 320x180 before handing to robot policies. Policies use 320x180 single-camera RGB-only images, with no other observations besides language. The asynchronous observations and actions are batched to psuedo-synchronous 5 Hz pairs for training the policy, with camera latency (characterized at roughly 80 ms) accounted for when forming psuedo-synchronous training pairs. The cylindrical end-effector is made from a 6 inch long plastic PVC pipe sourced from McMaster-Carr (9173K515). The work surface is 24 x 18 inch smooth wood cutting board. The manipulated objects are from the Play22 Baby Blocks Shape Sorter toy kit (Play22). The 6DOF robot is constrained to move in a 2D plane above the table.

B. Language-Table: Datasets

Here we outline the various datasets available in Language-Table, across simulation and real.

1) Simulation-Raw-Collect: This dataset consists of 6 teleoperators teleoperating a robot in simulation, following long horizon prompts. See representative prompts in Table VIII. 8318 episodes were collected with an average length of 36.8 ±15 seconds, yielding a total of 85.5 hours of raw data.

2) Simulation-Relabeled: The Simulation-Raw-Collect data was sent to 64 crowdsourced annotators, who used the interface described in Appendix E to generate 181,020 hindsight relabeled trajectories, with 78,623 unique instructions. See representative instructions in Table VII.

3) Real-World-Raw-Collect: This dataset consists of 11 teleoperators alternating over four robots, following long horizon prompts. See representative prompts in Table VIII. 23498 total episodes were collected with an average length of 9.9 minutes ±5.6 seconds, yielding a total of 3865 hours of raw data. Note that 16417 episodes totaling 2701 hours went into the actual training of policies, and the remaining was collected after training the demonstrated policy, but before releasing the dataset.

4) Real-World-Relabeled: The Real-World-Raw-Collect data was sent to 64 crowdsourced annotators, who used the interface described in Appendix E to generate 414,798 total hindsight relabeled trajectories, with 119,959 unique instructions. See representative instructions in Table VII. Note that 298,782 relabeled trajectories went into training, with 87,140 unique instructions, and the remaining was collected post-training, but pre-release.

C. Language-Table: Environment

Our simulated environment is intended to roughly match our real world setup, and consists of a simulated 6DoF robot xArm6 implemented in PyBullet [58] equipped with a small cylindrical end effector. Third person perspective 320x180 RGB-only images from a simulated camera are used as visual input. On a board in front of the robot are 8 blocks: red crescent, red pentagon, blue crescent, blue cube, green cube, green star, yellow star, and yellow pentagon. Like in the real world, the arm is constrained to the 2D plane and the action space is the delta 2D cartesian setpoint of the end effector. We run all experiments from RGB and language input only, but the environment additionally exposes 26-dimensional state observations (2D position and 1D rotation angle for each block, 2D position of end effector). While our real world policies perform asynchronous inference and control at 5hz, the policies in Language-Table perform blocking control at 10hz. Despite this difference, and others like differences between real and simulated images, we found policy performance in Language-Table was highly correlated with policy performance in the real world.

D. Language-Table: Evaluation

We define five simulated evaluation task families (spanning 696 unique task conditions) in Language-Table, each with a hand-defined success criterion:

- *block2block*: Push a block to another block. Success is thresholded distance between source and target block. There are 56 unique task conditions (8 source blocks x 7 target blocks).
- *block2abs*: Push a block to an absolute location on the board: top left, top center, top right, center left, center, center right, bottom left, bottom center, bottom right. Success is thresholded distance between block and target location. There are 72 unique task conditions (8 blocks x 9 locations).
- *block2rel*: Push a block to a relative offset location: left, right, up, down, up and left, up and right, down and left, down and right. Success is the thresholded distance between the block and the invisible target offset location. There are 64 unique task conditions (8 blocks x 8 offset directions).
- *block2blockrel*: Push a block to a relative offset location of another block: left side, right side, top side, bottom side, top left side, top right side, bottom left side, bottom right side. Success is the thresholded distance between the source block and the invisible target offset location of the target block. There are 448 unique task conditions (8 source blocks x 7 target blocks x 8 offset directions).
- *separate*: Separate two blocks. Success is the thresholded distance between the two blocks. There are 56 unique task conditions (8 source blocks x 7 target blocks).

These task families were used to benchmark models in simulation, allowing us to find hyperparameters that transferred well to fully-real-world training (we note that there was no sim-to-real component in the training employed by this work). The language conditioning for the automated evaluation tasks are generated synthetically from predefined synonym sets for each task condition.

E. Event selectable hindsight relabeling details

Figure 7 depicts a mockup of the interface our crowdsourced workers used to do event selectable hindsight relabeling. We asked data labelers to first watch an entire long horizon video, then produce 12 medium horizon and 12 short horizon actions, where the definition is left to rater discretion. Labelers have control of temporal segmentation tools, allowing them to mark the beginning and end of each action, and they describe each action as an open vocabulary natural language instruction.

Method	Success Rate
Full LAVA + training recipe (ours)	**0.772** ± 0.044
No CLIP Finetuning	0.735 ± 0.023
No Temporal Fusion Module	0.732 ± 0.036
Half Batch Size (2048)	0.720 ± 0.038

TABLE VI: Results of experiments to ablate model architecture details in the Language-Table simulator. All results are reported over 3 seeds after 350k steps.
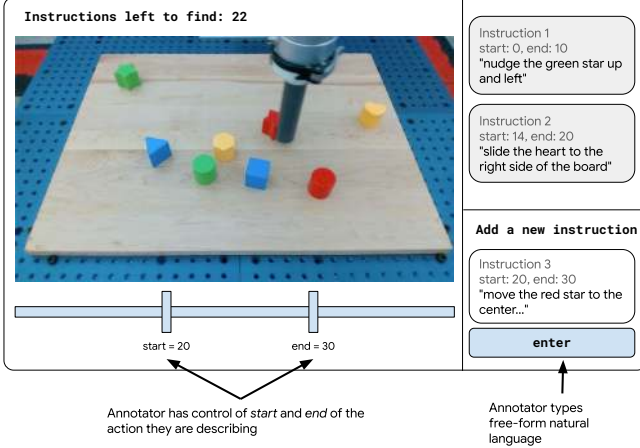
Fig. 7: Mockup of our event selectable hindsight relabeling interface.

F. Model architecture details

Here we describe LAVA ("Language Attends to Vision for Actions"), the transformer-based visuo-linguo-motor neural network architecture we use in this work. Internally, our architecture consists of a perception module, language module, vision-language fusion module, temporal fusion module, and policy output. We describe each below.

Perception module. Each training example consists of $(s, a, l)_i \sim \mathcal{D}_{\text{training}}$, where $s \in \mathbb{R}^{\text{seqlen} \times 320 \times 180 \times 3}$ is RGB observation history, and for the shown policies we used seqlen=4. We pass each frame in the video s through a convnet to obtain multi-scale visual feature descriptors (features at multiple layers). Our convnet consists of two Imagenet-pretrained ResNet [45], [46] layers and two additional learned convolutional layers with channel sizes 128 and 256 respectively and 2D max pooling between each layer. This yields a multi-scale feature pyramid for each image with $[H,W]$ of sizes [[112, 112], [56, 56], [28, 28], [14, 14], [7, 7]].

Language module. We use a pretrained CLIP text encoder [47], which is finetuned on our in-domain data, but remains fixed during policy training. We use a simple contrastive method for finetuning models pretrained on (image, language) pairs to domains where the observations are (video, language) pairs: generate (start frame s_0, goal frame s_g, language l) from all videos, and then during finetuning, pass concatenated image encodings $\text{concat}([z_0, z_g])$ through an MLP to get a single encoding z^{im} with the same dimensionality as encoded language z^{lang}. We preprocess text by stripping punctuation and extra spaces, but apply no additional preprocessing or augmentation. Cleaned text is passed through the CLIP embedder to get a sentence embedding with dimensionality 512.

Vision-Language Fusion Module. We fuse visual and lingual information using a "Language-Attends-to-Vision" transformer

block. For a single image position, this block takes as input (i) multi-scale pixel features (in our case the CNN features at zero-indexed layers 2, 3, 4 with H,W sizes [28, 28], [14, 14], [7, 7]) and (ii) a sentence embedding (in our case the 512-dimensional CLIP-encoded l). First, we map each layer n to $[H_n, W_n, \text{dmodel}]$ using a layer-specific MLP, then 2D position encode each feature map with 2D sinusoidal positional embeddings. We then flatten all the multi-scale features into one long visual token list (in our case with shape [1029, dmodel]). We project language to dmodel using an MLP, and apply dropout to both projected image and language features. We then iteratively fuse vision and language features, handing the sentence token as query and visual tokens as keys and values to a standard pre-norm decoder-only transformer [48] performing cross-attention, with only language on the residual path. Our vision-language transformer had 4 layers, with dmodel=128, 2 heads, feed forward width of 128, and dropout of 0.1.

Temporal Fusion Module. The output from applying our vision-language fusion module to each image in the $seqlen=4$ context history is a [seqlen,dmodel] sequence of vision-language embeddings. We apply 1D sinusoidal positional encoding to each element of the sequence, then feed the sequence to a standard pre-norm transformer performing self-attention, also outputting [seqlen,dmodel], which we average pool over the time dimension. Our temporal transformer had 2 layers, with dmodel=128, 2 heads, feed forward width of 128, and dropout of 0.1.

Policy Output. We hand the average-pooled dmodel embedding to a deep residual MLP with 2 blocks of residual width 1024. Each block has 3 MLP layers, the first two with width 256 and the final with width 1024. All MLPs have ReLU activation with normal initialization on kernel and bias. Finally we use a linear projection to the 2D action space.

G. Training details

We train our policies on a TPUv3 8x8 pod (64 TPUv3 chips) for approximately 500,000 steps or until training loss plateaus. At roughly 7.6 steps/second, policies finish training in 18 hours. All models are trained with Adam [59] with default TensorFlow momentum parameters, learning rate 1e-3, and a batch size of 4096. Action labels are normalized using statistics collected from training to $\mathcal{N}(0,1)$.

H. Ablations

We ablate the following training details. The results are reported in Table VI.

- **CLIP Finetuning.** We evaluate the importance of finetuning the CLIP language module on in-domain data. We use the pretrained weights for the ViT-B/32 model from [47] to encode language instructions without finetuning the text encoder on any Language-Table data. The text encoder still remains fixed during training. Although this results in only a few percent drop in the Language-Table sim, we observe a much stronger qualitative difference in model behavior on the real robot. This difference between sim and real could be explained by the Language-Table sim evaluation using a fixed set of templated instructions, while the real world evaluation uses more diverse language from human operators.

put all the blocks in a vertical line on the right of the board
group the blocks by color
make one horizontal line out of the red and blue blocks, then make a horizontal line out of the green and yellow blocks
make one horizontal line out of the blue and green blocks, then make a horizontal line out of the red and yellow blocks
put the: 0) green circle to top left, 1) red circle to top center, 2) green star to top right, 3) red star to center left, 4) blue triangle to center right, 5) yellow heart to bottom left, 6) yellow hexagon to bottom center, 7) blue cube to bottom right,
put 3 blocks in the bottom left corner, then the rest in the center left
make one horizontal line out of the red and green blocks, then make a vertical line out of the blue and yellow blocks
put the blocks in a diagonal line from the top left to bottom right
put the yellow and red blocks together in a group, then put the green and blue blocks together in a group
put the blue and green blocks together in the bottom center, then put the red and yellow blocks together in the center right
put the: 0) blue triangle to top left, 1) yellow hexagon to top center, 2) green star to top right, 3) blue cube to center left, 4) red circle to center right, 5) red star to bottom left, 6) yellow heart to bottom center, 7) green circle to bottom right
surround the yellow heart with the others
put all the blocks in the bottom right corner
make a "V"" shape out of all the blocks
put the red blocks in the center right, the yellow blocks in the bottom center, the green blocks in the bottom right corner, and the blue blocks in the top center
put the: 0) blue cube to top left, 1) yellow heart to top center, 2) blue triangle to top right, 3) red star to center left, 4) yellow hexagon to center right, 5) green star to bottom left, 6) green circle to bottom center, 7) red circle to bottom right
put the: 0) green circle to top left, 1) yellow heart to top center, 2) blue cube to top right, 3) red circle to center left, 4) blue triangle to center right, 5) green star to bottom left, 6) red star to bottom center, 7) yellow hexagon to bottom right
put all the blocks in the top center
put the red blocks in the top right corner, the green blocks in the center right, the blue blocks in the bottom center, and the yellow blocks in the center
put all the blocks in a vertical line on the center of the board

TABLE VIII: Representative examples of the prompts used to drive collection. These are discarded after collection.

slide the green circle into the top side of the yellow hexagon
slide the green star along with the yellow hexagon towards the center
move your arm near the bottom center
push the yellow heart closer to the yellow hexagon and blue triangle
move the blue triangle into group of blocks
push the red star upwards
place the yellow heart to the left side of the green star
place the green staryellow hexagon at the center of the board
nudge red star along with red circle a bit up
move the group of blocks to the centre of the board
move the arm left beside the red star
slide the blue triangle along with yellow hexagon slightly up
move the blue cube towards the center
push the red star along with the red circle towards the top center
push red star below the yellow heart
separate yellow hexagonn from the blue cube
move the red circle right and down a bit
slide the blue cube towards left
move the blue triangle along with the red circle slightly right
push the blue triangle to the bottom right of the blue cube

TABLE VII: Representative examples of crowdsourced instructions obtained via hindsight relabeling.

- **Temporal Fusion Module.** We evaluate the importance of using our transformer-based temporal fusion module. We encode each image in the $seqlen = 4$ context history by stacking the images channel-wise and encoding the images using a randomly initialized ConvNet. The multi-scale visual feature descriptors from this ConvNet are still fed into the "Language-Attends-to-Vision" transformer block, with no additional self-attention temporal fusion transformer.
- **Batch Size.** We evaluate the effect of batch size on our model by reducing our batch size in half to 2048. We see that this results in a performance drop after 350k steps.

I. Extended Related Work

Recent work has leveraged large language models (LLMs) to generate sequences of subgoals for language conditioned policies. These can be "open-loop" [10], [60], which lack an mechanism for replanning, or "closed-loop" [61], which generate up-to-date plans by prepending textual descriptions of the current scene to the prompting of the LLM planner. A limitation of both formulations is that when tasks that call for fine-grained spatial detail (like those examined in this work), it is difficult for LLMs to generate accurate subgoals from purely textual scene descriptions. Although visual language models (VLMs) like [62] suggest a promising direction, matching human levels of perception and cognition to effectively guide policies towards arbitrary goals remains a difficult open challenge. Our work is complementary in that our focus is instead on obtaining a large diverse set of short-horizon behaviors, and ones that can be interactively conditioned in real time. Combining autonomous long-horizon planning together with our demonstrated recipe for short-horizon behaviors is a strong candidate for future work.