LILA: Language-Informed Latent Actions

Anonymous Author(s)
Affiliation
Address
email

Abstract: We introduce Language-Informed Latent Actions (LILA), a framework for learning natural language interfaces in the context of human-robot collaboration. LILA falls under the shared autonomy paradigm: in addition to providing discrete language inputs, humans are given a low-dimensional controller – e.g., a 2 degree-of-freedom (DoF) joystick that moves left/right or up/down – for operating the robot. LILA learns to use language to *modulate* this controller, providing users with a control space that adapts based on the provided language: given an instruction like "place the cereal bowl on the tray," LILA might learn a 2-DoF control space where one dimension controls the distance from the robot's end-effector to the bowl, and the other dimension controls the robot's end-effector pose relative to the grasp point on the bowl. We evaluate LILA with real-world user studies pairing users with a 7-DoF Franka Emika Panda Arm, with the goal of completing a series of complex manipulation tasks with a lower dimensional control space conditioned on language. LILA models are not only more sample efficient and performant than imitation learning and end-effector control baselines, but also are qualitatively preferred by users.[1]

Keywords: Language for Shared Autonomy, Language & Robotics, Learned Latent Actions, Human-Robot Interaction

1 Introduction

Effective human-robot collaboration is rooted in the ability to strike the right balance between the capabilities of human users and the power of robotics. Finding this balance is important; nearly a million American adults live with physical disabilities, requiring assistance for everyday tasks like taking a bite of food or pouring a glass of milk [1]. One path forward is with *shared autonomy* [2, 3, 4], a framework that couples user input with automated robot assistance. Shared autonomy approaches tap into the complementary strengths of both humans and robots; humans continuously provide inputs, while the robot deftly ingests a stream of rich and diverse observations, assisting the human in fulfilling their desired objective. Such approaches have boundless promise for interpretability, safety, and sample efficiency, amongst others. Despite their promise, shared autonomy approaches are missing rich, intuitive ways of *specifying* and *disambiguating* tasks – a bridge that can be built by tapping into the potential of natural language.

Consider a kitchen or dining setting where a *high-dimensional* (high-DoF) robot (such as a wheelchair-mounted manipulator) aids a human user who may not be physically able to perform tasks requiring fine-grained manipulation. While one option for the human is to manually teleoperate the arm by controlling individual "modes" (separate degrees-of-freedom) of the robot's end-effector, past work has shown this to be unintuitive, slow, and frustrating [3, 5]. However, *learned latent actions* [6, 7] is a promising new paradigm that seeks to address the problem of building intuitive low-dimensional (low-DoF) controllers for high-DoF robot manipulators by learning *task-specific control spaces* via dimensionality reduction. While these approaches are reliable and sample efficient, they are limited by their single task nature. Existing latent action controllers fail to disambiguate user intent in multi-task settings, proving unable to assist in realizing the human's goal a human wishes to achieve.

[1] Additional visualizations, videos, and supplemental experiments can be found at the following webpage: https://sites.google.com/view/lila-corl21.

Submitted to the 5th Conference on Robot Learning (CoRL 2021). Do not distribute.

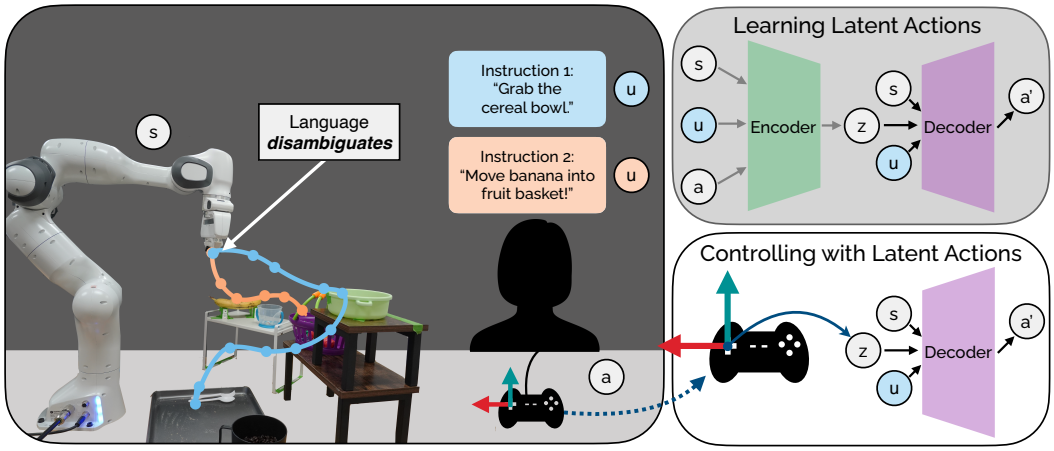

Figure 1: [**Left**] Our breakfast buffet environment with several diverse manipulation tasks. By providing both natural language input and low-dimensional joystick control [**Middle**], users *disambiguate* between different tasks while retaining the ability to maneuver through the environment. This is enabled by our language-informed latent actions (LILA) models [**Right**].

Language is an expressive way for humans to communicate intent. A user can easily express their desire to *"grab the cereal bowl"* or their preference to *"slowly pour the milk into the glass till it's half full"*. Prior work integrates language in robotics settings via full autonomy [8, 9, 10, 11, 12]. Unfortunately, these approaches are not silver bullets: they are often plagued with poor sample efficiency, inability to recover from failure, and poor generalization to novel states – many issues that shared autonomy methods (including learned latent actions) seek to address.

We propose Language-Informed Latent Actions (LILA), a framework for incorporating language within the broader paradigm of learned latent actions. Key to LILA is the principle that language *modulates* a user's low-level controller based on their provided utterance; given the utterance *"grab the cereal bowl"*, our assistive robot exposes a semantically meaningful, low-dimensional (2-DoF) control space where one dimension (one joystick axis) may control the distance from the robot's end-effector to the cereal bowl, whereas the other might control the angle of the end-effector relative to the bowl such that the robot's gripper can obtain a solid grasp of the object. Future utterances modulate the controller in similar ways – "pour the milk into the cup" might result in one joystick dimension controlling the angle at which the milk jug is tilted, whilst the other controls the height of the jug relative to the cup. Language not only is a natural means for a human to communicate their intent to the robot, but it helps *disambiguate* across a wide variety of tasks as well.

In addition to providing a formalism for integrating natural language within the learned latent actions framework, and a robust, sample-efficient model architecture that allows for reliable task execution, we perform a *comprehensive user study* across 10 users who use natural language and our learned LILA controllers to complete a variety of diverse manipulation tasks in a simplified assistive "buffet" setting. Our results show that not only are LILA models more reliable, performant, and sample efficient than fully autonomous (imitation learning) and end-effector control baselines, but they are qualitatively preferred by users as well.

2 Related Work

We build LILA within a shared autonomy framework [2, 13], applied to assistive teleoperation [14, 15]. We additionally build off work at the intersection of language and robotics [16, 17, 18].

Shared Autonomy & Assistive Teleoperation. Shared autonomy casts robot control as a collaborative process between humans and robots [2, 4, 13, 19]. While other work focuses on methods that actively assist users by "blending" human inputs with possibly task-agnostic policies [20, 21], in this work, we focus on assistive teleoperation, where a user is provided a low-dimensional controller (e.g., a joystick, sip-and-puff device) to directly control a high-dimensional robot manipulator. Using these controllers to control the end-effector via inverse kinematics or operational space control [22] is incredibly difficult, requiring frequent mode-switching to control specific robot DoFs, failing to provide users with fine-grained control [3, 23]. Instead, we adopt *learned latent actions*

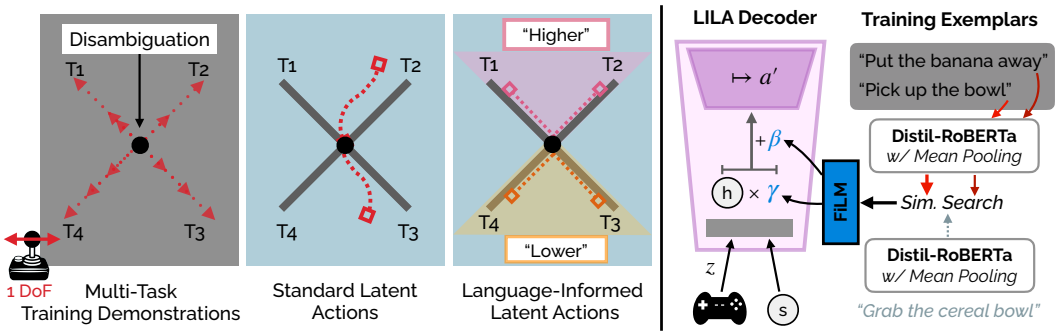

Figure 2: [**Left**] Navigating toward four points on a cross with a 1-DoF latent action, where there is a point of disambiguation. Standard LA fails at this task, while LILA accurately reaches the corners with the help of language. [**Right**] LILA decoder architecture. We embed a user utterance using a pretrained language model, then identify the closest exemplar in the training set via similarity search. We feed the embedding for this exemplar through a FiLM layer that injects linguistic information into the core of a standard latent action decoder.

[6, 7, 24, 25] a recent framework that uses conditional auto-encoders [26] to learn task-specific latent action spaces from demonstrations. These latent spaces match the dimensionality of the low-dimensional control interface, and provide intuitive, semantically meaningful control. However, existing models fail to differentiate between tasks when presented with similar states, hindering the ability to perform diverse behaviors in a workspace (e.g., manipulating a jug of milk in different ways – pouring, placing in the fridge, etc.). In this work, we use language for *disambiguation*; users naturally specify their intent with language, conditioning latent action models to produce intuitive control spaces that align with user objectives.

Language-Informed Robotics. A wealth of work looks at different ways to incorporate language for robotics, spanning approaches that map language to planning primitives [8, 27, 28, 29, 30], imitation learning from demonstrations paired with language instructions [10, 31, 32, 33], and reinforcement learning from language instructions paired with corresponding reward functions [11, 34, 35]. Other approaches use language in more nuanced ways, for example, to learn language-conditional reward functions directly [36, 37, 38], or within more adaptive frameworks, where language is used to correct or define new behavior [39, 40]. This list is not exhaustive, nor are the explanations complete; we present further discussion – including approaches that combine language with gestures and other modalities – in the Supplemental. However, all these approaches fall broadly under the category of full autonomy: after providing an instruction, human users cede control over to the robot policy, which then takes the actions necessary to perform a task.

While robots trained with these approaches can perform diverse tasks and generalize to new instructions, it is not without cost. Paramount is sample efficiency; imitation learning approaches often require hundreds to thousands of demonstrations to learn navigation tasks [32, 41], and reinforcement learning approaches can require millions of episodes of experience to learn robust policies [11, 42]. Whereas coarse behaviors are easy to learn, learning to recover from slight deviations from the training data, or to perform precise motions in a sequence is incredibly difficult. By casting our approach LILA within a shared autonomy framework instead, we intelligently offload these parts that are harder for robots – but easier and intuitive for humans – onto the user, who provides the necessary control inputs.

3 Framing Language for Assistive Teleoperation

Consider our running example (Fig. 1): you – who may not be physically able to perform actions like reaching or lifting – are equipped with a high-dimensional robot manipulator and a low-dimensional controller (e.g., a 2-DoF joystick). You have simple tastes – breakfast consists of a bowl of cereal with milk. How do you achieve this goal?

With LILA, you lead with language – *"grab the cereal bowl"* – which is fed to a learned model that *modulates*, or restructures, your control interface. Pressing **`up`** on the joystick now moves the robot arm towards the cereal bowl, while pressing **`right`** aligns the end-effector angle to make the grasp. You press a button closing the gripper around the cereal bowl, then hold **`down`** on the joystick

Task Name	Success	Example User Study Input	Mapped Training Data
Pick Banana	100%	*yellow in purple*	→ `pick up the yellow banana and place it into the purple basket`
Pick Fruit Basket	100%	*bring basket to center of pan*	→ `place the basket onto the tray`
Pick Cereal	100%	*go to the left side of the cream bowl, go down, grab the cereal bowl, and place it on the try*	→ `grab the cereal bowl and put it on the tray`
Pour Bowl	67%	*pick up the cup of marbles and pour them into the cereal bowl*	→ `pick and pour the cup of white balls into the bowl of cereal`
Pour Cup	100%	*pick up the clear cup with marbles in it and pour it in the black mug with the coffee beans in it*	→ `pick up the cup and pour the contents in the mug`

Table 1: Example utterances provided by user study participants paired with the retrieved exemplar per the method described in §4.2. Success rate refers to the percentage of the time a user study utterance (over all utterances in the study) was grounded to the correct task via our retrieval method. As each participant only attempted 2 tasks, success rate can fluctuate significantly, as is the case with the **Pour Bowl** task.

to bring it back to your tray (e.g. blue trajectory in Fig. 1). You speak – *"pour some milk into my cereal"* and your controls adapt; holding **down** has new meaning, bringing the robot closer to the cup of milk *and* angles the end-effector to grasp it. Tilting **right** begins to pour the milk, and so on.

Formalism. We formulate a user's task (on a *per-user basis*) as a fully-observable, language-augmented, Markov Decision Process (MDP) $\mathcal{M}$ defined by the tuple $(\mathcal{S}, \mathcal{U}, \mathcal{A}, T, R, \gamma)$ similar to prior work in language-conditional robotics [9, 31, 39]. Let $u \in \mathcal{U}$ denote a user's language utterance provided at the start of each task, where $\mathcal{U}$ is the full set of language utterances a user could provide to a robot. Let $s \in \mathcal{S} \subset \mathbb{R}^n$ be the robot's state, and $a \in \mathcal{A} \subset \mathbb{R}^m$ be the robot's action: taking action a in state s results in a next state s' according to the transition function $T(s, a)$. Given the language utterance u, the user implicitly defines a reward function $R(s, a, u) \in \mathbb{R}$; the human and robot collaboratively maximize this reward subject to discount factor $\gamma \in [0, 1]$.

In this work, we experiment exclusively with the Franka Emika Panda robot arm, a 7-DoF manipulator. States are taken to be the robot's joint states ($n = 7$) while actions are provided as continuous joint velocities ($m = 7$). Language utterances u are typed into a text console, but future work could easily extend this to use off-the-shelf speech recognition systems.

Problem Statement. This MDP forms the basis of a shared autonomy task wherein a human is equipped with a low-dimensional control interface for the robot. Let $z \in \mathcal{Z} \subset \mathbb{R}^d$ where $d << m$ be the human's control input to the robot, such as the $d = 1$ DoF controller in Fig. 2. Previous work on learning latent actions for assistive teleoperation [7, 25] learn a decoder $\phi(z, s) : \mathcal{Z} \times \mathcal{S} \rightarrow \mathcal{A}$ that maps user low-dimensional inputs $z \in \mathcal{Z}$ and current state $s \in \mathcal{S}$ to a high-dimensional action $a \in \mathcal{A}$. However, in situations such as the *disambiguation point* that may arise when task trajectories overlap (Fig. 2), too low a control input dimension d may lead to collapse in the latent action space.

We aim to learn a decoder $\phi(z, s, u) : \mathcal{Z} \times \mathcal{S} \times \mathcal{U} \rightarrow \mathcal{A}$ that incorporates the user's control input *and* initial language utterance u, and predicts the robot action a_u that maximizes reward. Thus, we propose using language as an expressive *conditioning* variable to enable latent actions to be useful in multi-task environments.

4 LILA – Language-Informed Latent Actions

LILA builds off the *learned latent actions* paradigm for assistive teleoperation. Specifically, we extend the original conditional auto-encoder models (CAE) from [6] to incorporate language and address the disambiguation problem discussed above.

4.1 Extending Learned Latent Actions with Language

We assume a dataset of demonstrations associated with a corresponding language utterance. We split each demonstration into triples (s, u, a) and use these to learn a conditional auto-encoder, consisting of a language-conditional encoder $\psi : \mathcal{S} \times \mathcal{A} \times \mathcal{U} \rightarrow \mathcal{Z}$ that maps language and states to a latent z and a decoder $\phi : \mathcal{Z} \times \mathcal{S} \times \mathcal{U} \rightarrow \mathcal{A}$ that attempts to reconstruct the original action a. We minimize the mean-squared error between the predicted action and the original action:

$$L_{\phi,\psi} = \frac{1}{N} \sum_{i=1}^{N} (\phi(\psi(s_i, a_i, u_i), s_i, u_i) - a_i)^2 \tag{1}$$

Both the encoder ψ and decoder ϕ are neural networks optimized with stochastic gradient descent; we discuss how we integrate language (including the use of pretrained language models) in §4.2.

4.2 Integrating Language

Large, pretrained language models such as BERT, T5, and GPT-3 [43, 44, 45] have revolutionized NLP, providing language representations that capture sentence-level semantics. Inspired by the success of using these models for encoding language in the context of robotics and reinforcement learning tasks [35, 46, 47], we take a similar approach. For encoding language, we use a distilled variant of the RoBERTa-Base model [48], from `Sentence-Transformers` [49, 50]. This model is fine-tuned on a large corpus of paraphrased sentences, allowing it to pick up on longer-term, sentence-level (utterance length) semantics. We generate sentence embeddings by performing mean-pooling over token embeddings for an utterance, as is done in prior work [35, 46]. We incorporate these embeddings into our latent actions models using feature-wise linear modulation (FiLM) layers [51], which incorporates linguistic information into learned features h by mapping language to parameters (γ, β) of an affine transformation $h' = \gamma * h + \beta$ that is fed to later layers of the network Fig. 2. The Supplemental provides additional details around the implementation (with code).

Nearest-Neighbor Retrieval at Inference. A major concern for work in language-conditioned robotics is the ability to generalize to novel language inputs not seen at training. While it may be unreasonable to expect generalization to completely new tasks, for user-facing systems with clear task distribution (as in our work), there is an implicit expectation that any language-informed system is capable of handling moderate variations or paraphrases of utterances from the training set.

To do this, adding linguistically diverse data has been the gold standard [41, 47]; however, a new class of approaches have emerged that sidestep additional data requirements by tapping into the potential of pretrained language models [40, 46]. These approaches frame language interpretation at inference (when interfacing with real users) as a *retrieval* problem: each new user utterance u' is embedded (with the same pretrained model as above) and then used to query a nearest neighbors store containing all training exemplars; once the nearest neighbor u_i has been identified, it replaces u' as an input to the LILA decoder.

The key benefit of such an approach is that there is no mismatch between train and test language inputs; all "test inputs" are drawn solely from the training set. This does mean, however, that user utterances that describe new tasks, or are otherwise unachievable also get mapped to language seen at training. While this limits generalization, it again highlights the benefits of operating within the shared autonomy paradigm (doubly so, considering the cost of a mistake in an assistive domain like the one we consider in this work): if, while providing control inputs to the robot, a user feels the robot is not acting in alignment with the user's desired objective, they can always stop execution.

5 User Study

We evaluate LILA with a real-world user study on a 7-DoF Franka Emika Panda Arm, on a series of complex manipulation tasks. Each user is provided a 2-DoF joystick for control. We compare against a non-learning, End-Effector (EE) control baseline where users "mode switch," controlling the velocity of 2 axes of the End-Effector pose at a given time – [(X, Y), (Z, Roll), (Pitch, Yaw)]. We also compare against a fully autonomous Imitation Learning (IL) strategy where users solely provide language inputs, and the robot attempts to perform a task without additional input. A baseline that is ostensibly missing is a *non-language* variant of the latent actions model, to demonstrate the necessity

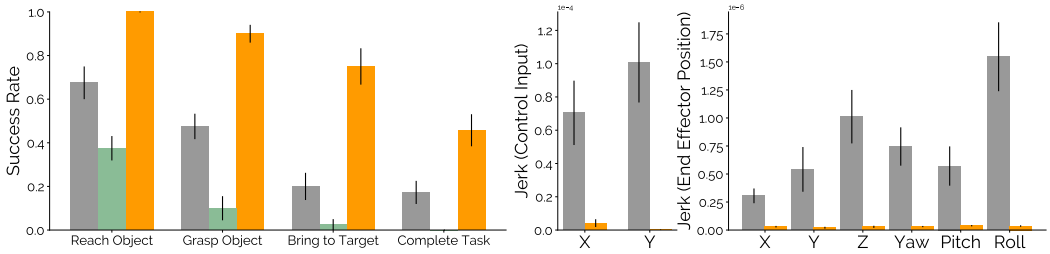

Figure 3: Quantitative Results. We average success rate [**Left**] across all sub-tasks for each control method, and find that LILA is significantly ($p > 0.05$) more performant. However, the steep drop in performance when completing the full task shows the difficulty of fine-grained control. We also calculate jerk as an indicator of controller smoothness, for both user control inputs [**Middle**] and end-effector position [**Right**]. Averaged across tasks and users, we find LILA leads to significantly smoother control for users than end-effector control.

of language. However, upon evaluating this baseline, we found it to be unintuitive, and unable to make progress on any task – showing another input such as language is necessary. We explore this further with experiments and formal arguments in the Supplemental.[2]

Environment. Fig. 1 shows our "breakfast buffet" setting, a scaled down version of an assistive feeding domain. We perform a within-subjects study where each user is assigned 2 of the following 5 tasks to perform with each control method: 1) **`Place Banana`**: placing the banana in the purple fruit basket, 2) **`Place Basket`**: grasping the purple basket by the handles, and dropping it on the tray, 3) **`Place Bowl`**: grasping the green cereal bowl by its edge, and moving it to the tray, 4) **`Pour Bowl`**: pouring the blue cup of marbles (a proxy for actual milk) into the cereal bowl (positioned on the tray), and 5) **`Pour Cup`**: pouring the blue cup of marbles into the yellow coffee cup. Fig. 1 shows idealized example trajectories for the **`Place Bowl`** (blue) and **`Place Banana`** (orange) tasks.

These tasks vary in difficulty, with many requiring precise grasping and dexterous manipulation (e.g., pouring). Quantitatively, we evaluate partial success based on how many of the following 4 subtasks users are able to complete: 1) **`Reaching`**: touching the desired object, 2) **`Grasping`**: executing a successful grasp, 3) **`Bring to Target`**: successfully transporting the manipulated object to the target location, and 4) **`Task Completion`**: whether users are able to finish the given task (placing an object without excessive force, in an acceptable orientation, or pouring all marbles into the desired receptacle).

Demonstration Collection. Both LILA and IL models require learning from (language, demonstration) pairs for each of the 5 tasks. We collect task demonstrations kinesthetically, by manually backdriving the robot arm, as in prior work [6, 25]. We trace out the trajectory necessary and record joint states at a fixed frequency. We initially collected 15 demonstrations per task for each method. However, on manually testing the IL model, we found it incapable of performing even rudimentary reaching behaviors. To give IL the best chance, we collected twice the number of demonstrations (30 per task; 150 total), requiring an extra 2 hours of labor. Further details, including videos of our demonstration collection process for each method, are included in the Supplemental.

Crowdsourcing Language Annotation. To realize our goal of building a *natural* language interface for human-robot collaboration, we collect language annotations for each task by crowdsourcing utterances. In order to capture the diverse ways users may refer to the objects and actions our tasks entail without any additional information (simulating a real user walking into our environment for the first time), we recruited 30 workers on Amazon Mechanical Turk and showed them a video of a recorded task demonstration, asking them to provide *"a short instruction that you would want to provide the robot to complete this task independently in the box below."*. However, this procedure alone resulted in some annotations containing "spam", noise, or extremely out-of-domain text. To address this issue systematically, without introducing our own bias, we filter data to identify workers who provided "atypical" annotations, defined by measuring the cosine distance between the sentence embedding (per the pretrained language model from §4.2) of an annotator's provided sentence, and the *average* sentence embedding aggregated over all annotators for the given task. We

[2]Visualizations of this omitted non-language baseline and a summary of the formal argument can be found here: `https://sites.google.com/view/lila-corl21/home/rebuttal-no-language-baseline`

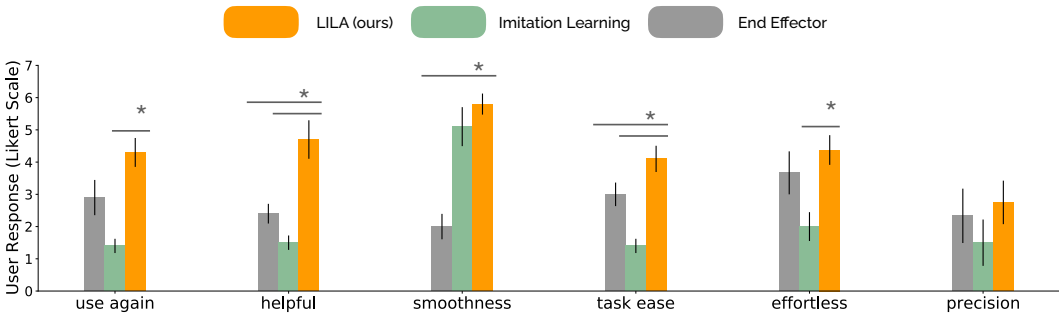

Figure 4: Qualitative Results. Using a 7-point Likert scale, we ask users to evaluate each of the 3 control methods for different properties. With high significance ($p > 0.05$), we find that LILA outperforms both Imitation Learning and End Effector Control baselines on several metrics, including degree of helpfulness the control method provided and ease in completing tasks.

used annotations only from the 15 "best" annotators under this metric as our ground-truth language utterances for training our models. Further details on our data collection and filtering procedure are in the Supplemental, as well as example "spam" and noisy annotations that were filtered out. Table 1 provides examples of crowdworker utterances from our final dataset (rightmost column).

Participants & Procedure. We conducted our study with a participant pool of 10 university students (5 female and 5 male, age range 23.2 ± 1.87) with a Franka Emika Panda Arm (7-DoF). Four subjects had prior experience teleoperation a robot arm.[3] We conduct a within-subjects study, where each participant completed 2 tasks, chosen randomly, with each of the three methods. Users were given 2 trials to complete each tasks, and an allotted 3 minutes per control strategy to practice. Users were also given a sheet describing controller inputs and details for each control method, which we include in the Supplemental. For Imitation Learning and LILA controllers, which require language inputs, participants provided a natural language utterance which was entered into the model – participants were allowed to verify the proctor entered their input accurately. This user-provided language utterance is used as the query in the nearest-neighbor retrieval described in §4.2; the retrieval set consists of all training utterances collected via the crowdsourcing procedure above. In addition to tracking quantitative success rates (normalized, based on progress relative to each of the 4 defined subtasks), time taken per task, and detailed logs of the robot behavior, at the end of each study, users were asked to fill out a qualitative survey evaluating each control method. We present both quantitative and qualitative results in the following paragraphs.

Quantitative Results. Fig. 3 summarizes our objective results. We evaluate both full- and partial-task success rates for each task across all control methods, in addition to computing smoothness metrics directly on the logged user inputs and robot actions. Intuitively, smoothness is a measure for intuitiveness (when measured on user 2-DoF joystick inputs), and ease of use (when measured on the robot's end-effector pose). We measure smoothness by calculating jerk, the time-derivative of acceleration. Our results show that LILA significantly ($p > 0.05$) outperforms both methods across all sub-tasks, and is also smoother to use both in control input space (2-DoF input) and end-effector space (6-DoF). However, the relative drop in performance of LILA for the final sub-task "Complete Task" shows the room for improvement in fine-grained control, such as pouring motions. Additionally of note is the poor performance of Imitation Learning. To explore this fully, we perform an ablation, and show that sample inefficiency is a likely cause – especially since we are in a low-data regime. These results and arguments can be found in the Supplemental.[4]

Qualitative Results. Fig. 4 summarizes our subjective results. We administered a 7-point Likert scale survey after users finished performing tasks with each of the three different methods; this survey included questions such as the the perceived helpfulness of the controller in completing the tasks (*helpful*) and whether the participant would use the control method again (*use again*). The survey results show that LILA outperforms both Imitation Learning and End Effector controls across most qualitative metrics, with statistically significant results ($p > 0.05$) marked with an $*$. We

[3] Due to the COVID-19 Pandemic and university restrictions, only those with pre-authorized access could participate in the study. See the COVID-19 considerations document in the Supplemental for more details.

[4] Additional experiment videos, and a summary of why imitation learning results are poor can be found here: https://sites.google.com/view/lila-corl21/home/rebuttal-imitation-learning-baseline

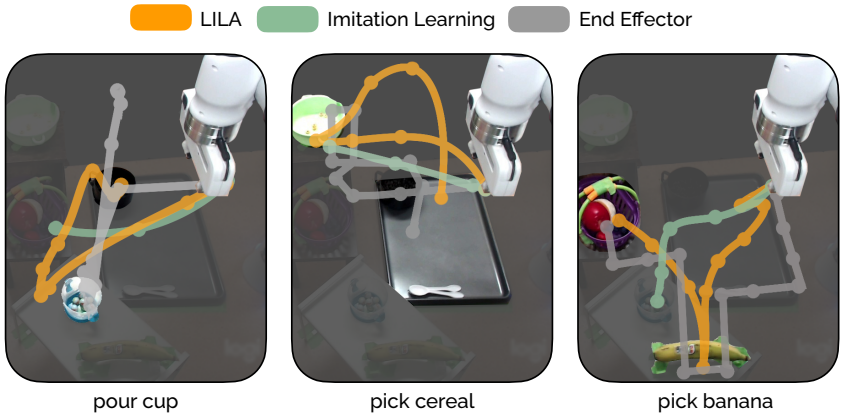

Figure 5: Trajectories from one user comparing three different control methods 3 out of our 5 tasks – **`Pour Cup`**, **`Pick Cereal`**, and **`Pick Banana`**. LILA provides smooth actions that immediately approach the target object for a task, while End-Effector control shows more rigid motions that result in users diverging from their intended paths. While imitation learning also enables smooth motions, it often fails shortly after reaching objects, hence the shortened trajectories.

additionally visualize samples of the observed end-effector trajectories by individual users (collected during our study) for 3 of our 5 tasks in Fig. 5. We observe that across all tasks, LILA results in smoother end-effector trajectories than end-effector control, while Imitation Learning comes close to the target object but is unable to complete the entire trajectory for the task.

6 Discussion

Summary. We present Language-Informed Latent Actions (LILA) a framework that marks the first step in combining the expressiveness and naturalness of language for *specifying* and *disambiguating* tasks within the context of assistive teleoperation. Our user study results show that when compared to fully autonomous, imitation learning approaches, LILA is more sample-efficient and performant, training on half the number of task demonstrations, but obtaining significantly higher success rates. Compared with no-learning end-effector control methods, we again show LILA's effectiveness at obtaining high success rates, but also demonstrate its ability to produce intuitive low-dimensional control spaces from language input. Qualitatively, we find that users prefer LILA to alternative methods across the board, opening the door for additional work on language & latent actions.

Limitations and Future Work. Currently, LILA is geared to use language as a mechanism for *task disambiguation* – in its current form, it has no mechanism for generalizing to completely unseen tasks or language specifications. We believe that the ability to disambiguate with language, and the integration of language within the latent actions framework is a strong research contribution, and hope that future work looks to dynamic states – perhaps by leveraging visual latent actions [25] – and to adapting to new utterances and tasks dynamically [52, 40]. Furthermore, while users found LILA intuitive and natural, they found themselves wanting to further modulate the robot's behavior with language instructions *during the course of execution*. Many users, upon seeing the robot make slight deviations from a desired path would instinctively provide spoken corrections – *"a little to the right"*, *"no, grasp it by the handle!"* – indicating a desire for *multi-resolution* language control. Future work will look into this as well.

Shared Autonomy and LILA. LILA fits within the shared autonomy paradigm, where the role of language has been underexplored. With LILA (and shared autonomy approaches in general), humans retain *agency* – they are responsible for robot motion, and if the robot moves in a way that is not safe, or does not align with their objectives, they stop providing control (and possibly reset, give a new instruction, or drop into a more complex control mode). Behavior is *interpretable* – the latent actions model, *critically informed by language*, produces intuitive control spaces that humans can quickly grasp. Finally, language is *natural* – users specify their objectives as they would if speaking to another person, and the robot uses that language to shape their control space. These properties – preserving agency, maintaining interpretability, and leveraging the expressive and natural features of language for specifying objectives – are critical for widespread human-robot collaboration, and we hope this work is a first step towards achieving that goal.

References

[1] D. M. Taylor. *Americans With Disabilities: 2014*. US Census Bureau, 2018.

[2] A. D. Dragan and S. S. Srinivasa. A policy-blending formalism for shared control. *International Journal of Robotics Research (IJRR)*, 32:790–805, 2013.

[3] B. D. Argall. Autonomy in rehabilitation robotics: an intersection. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:441–463, 2018.

[4] S. Javdani, H. Admoni, S. Pellegrinelli, S. S. Srinivasa, and J. A. Bagnell. Shared autonomy via hindsight optimization for teleoperation and teaming. *International Journal of Robotics Research (IJRR)*, 37:717–742, 2018.

[5] L. V. Herlant, R. M. Holladay, and S. S. Srinivasa. Assistive teleoperation of robot arms via automatic time-optimal mode switching. In *ACM/IEEE International Conference on Human Robot Interaction (HRI)*, pages 35–42, 2016.

[6] D. P. Losey, K. Srinivasan, A. Mandlekar, A. Garg, and D. Sadigh. Controlling assistive robots with learned latent actions. In *International Conference on Robotics and Automation (ICRA)*, pages 378–384, 2020.

[7] H. J. Jeon, D. P. Losey, and D. Sadigh. Shared autonomy with learned latent actions. In *Robotics: Science and Systems (RSS)*, 2020.

[8] S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. J. Teller, and N. Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2011.

[9] D. Arumugam, S. Karamcheti, N. Gopalan, L. L. S. Wong, and S. Tellex. Accurately and efficiently interpreting human-robot instructions of varying granularities. In *Robotics: Science and Systems (RSS)*, 2017.

[10] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Computer Vision and Pattern Recognition (CVPR)*, pages 6077–6086, 2018.

[11] K. M. Hermann, F. Hill, S. Green, F. Wang, R. Faulkner, H. Soyer, D. Szepesvari, W. Czarnecki, M. Jaderberg, D. Teplyashin, M. Wainwright, C. Apps, D. Hassabis, and P. Blunsom. Grounded language learning in a simulated 3d world. *arXiv preprint arXiv:1706.06551*, 2017.

[12] S. Stepputtis, J. Campbell, M. Phielipp, S. Lee, C. Baral, and H. B. Amor. Language-conditioned imitation learning for robot manipulation tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[13] D. Gopinath, S. Jain, and B. D. Argall. Human-in-the-loop optimization of shared autonomy in assistive robotics. *IEEE Robotics and Automation Letters (RA-L)*, 2:247–254, 2016.

[14] M. T. Ciocarlie and P. K. Allen. Hand posture subspaces for dexterous robotic grasping. *International Journal of Robotics Research (IJRR)*, 28:851–867, 2009.

[15] S. Jain and B. Argall. Probabilistic human intent recognition for shared autonomy in assistive robotics. *ACM Transactions on Human-Robot Interaction (THRI)*, 9:1–23, 2019.

[16] 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. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2019.

[17] C. Matuszek. Grounded language learning: Where robotics and NLP meet. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.

[18] S. Tellex, N. Gopalan, H. Kress-Gazit, and C. Matuszek. Robots that use language. *Annual Review of Control, Robotics, and Autonomous Systems*, 3(1):25–55, 2020.

[19] B. A. Newman, R. M. Aronson, S. Srinivasa, K. Kitani, and H. Admoni. Harmonic: A multimodal dataset of assistive human-robot collaboration. *arXiv preprint arXiv:1807.11154*, 2018.

[20] S. Reddy, A. D. Dragan, and S. Levine. Shared autonomy via deep reinforcement learning. In *Robotics: Science and Systems (RSS)*, 2018.

[21] C. B. Schaff and M. R. Walter. Residual policy learning for shared autonomy. In *Robotics: Science and Systems (RSS)*, 2020.

[22] O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3:43–53, 1987.

[23] W. J. Wilson, C. W. Hulls, and G. S. Bell. Relative end-effector control using cartesian position based visual servoing. *IEEE Transactions on Robotics (T-RO)*, 12:684–696, 1996.

[24] M. Li, D. P. Losey, J. Bohg, and D. Sadigh. Learning user-preferred mappings for intuitive robot control. In *International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[25] S. Karamcheti, A. Zhai, D. P. Losey, and D. Sadigh. Learning visually guided latent actions for assistive teleoperation. In *Learning for Dynamics & Control Conference (L4DC)*, 2021.

[26] C. Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.

[27] T. Kollar, S. Tellex, D. Roy, and N. Roy. Toward understanding natural language directions. In *Human-Robot Interaction*, pages 259–266, 2010.

[28] C. Matuszek, E. Herbst, L. Zettlemoyer, and D. Fox. Learning to parse natural language commands to a robot control system. In *International Symposium on Experimental Robotics (ISER)*, 2012.

[29] Y. Artzi and L. Zettlemoyer. Weakly supervised learning of semantic parsers for mapping instructions to actions. *Transactions of the Association for Computational Linguistics (TACL)*, 1:49–62, 2013.

[30] S. Karamcheti, E. C. Williams, D. Arumugam, M. Rhee, N. Gopalan, L. L. S. Wong, and S. Tellex. A tale of two draggns: A hybrid approach for interpreting action-oriented and goal-oriented instructions. In *First Workshop on Language Grounding for Robotics @ ACL*, 2017.

[31] V. Blukis, N. Brukhim, A. Bennett, R. A. Knepper, and Y. Artzi. Following high-level navigation instructions on a simulated quadcopter with imitation learning. In *Robotics: Science and Systems (RSS)*, 2018.

[32] X. E. 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 *Computer Vision and Pattern Recognition (CVPR)*, 2019.

[33] C. Lynch and P. Sermanet. Grounding language in play. *arXiv preprint arXiv:2005.07648*, 2020.

[34] D. S. Chaplot, K. M. Sathyendra, R. K. Pasumarthi, D. Rajagopal, and R. Salakhutdinov. Gated-attention architectures for task-oriented language grounding. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2018.

[35] F. Hill, S. Mokra, N. Wong, and T. Harley. Human instruction-following with deep reinforcement learning via transfer-learning from text. *arXiv preprint arXiv:2005.09382*, 2020.

[36] J. MacGlashan, M. Babes-Vroman, M. desJardins, M. Littman, S. Muresan, S. Squire, S. Tellex, D. Arumugam, and L. Yang. Grounding English commands to reward functions. In *Robotics: Science and Systems (RSS)*, 2015.

[37] J. Fu, A. Korattikara, S. Levine, and S. Guadarrama. From language to goals: Inverse reinforcement learning for vision-based instruction following. In *International Conference on Learning Representations (ICLR)*, 2019.

[38] D. Bahdanau, F. Hill, J. Leike, E. Hughes, S. A. Hosseini, P. Kohli, and E. Grefenstette. Learning to understand goal specifications by modelling reward. In *International Conference on Learning Representations (ICLR)*, 2019.

[39] J. D. Co-Reyes, A. Gupta, S. Sanjeev, N. Altieri, J. DeNero, P. Abbeel, and S. Levine. Guiding policies with language via meta-learning. In *International Conference on Learning Representations (ICLR)*, 2019.

[40] S. Karamcheti, D. Sadigh, and P. Liang. Learning adaptive language interfaces through decomposition. In *EMNLP Workshop for Interactive and Executable Semantic Parsing (IntEx-SemPar)*, 2020.

[41] 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 *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[42] M. Chevalier-Boisvert, D. Bahdanau, S. Lahlou, L. Willems, C. Saharia, T. H. Nguyen, and Y. Bengio. Babyai: A platform to study the sample efficiency of grounded language learning. In *International Conference on Learning Representations (ICLR)*, 2019.

[43] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Association for Computational Linguistics (ACL)*, pages 4171–4186, 2019.

[44] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

[45] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[46] A. Marzoev, S. Madden, M. Kaashoek, M. J. Cafarella, and J. Andreas. Unnatural language processing: Bridging the gap between synthetic and natural language data. *arXiv preprint arXiv:2004.13645*, 2020.

[47] A. Ku, P. Anderson, R. Patel, E. Ie, and J. Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[48] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[49] N. Reimers and I. Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

[50] N. Reimers and I. Gurevych. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[51] E. Perez, F. Strub, H. D. Vries, V. Dumoulin, and A. C. Courville. Film: Visual reasoning with a general conditioning layer. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2018.

[52] S. I. Wang, S. Ginn, P. Liang, and C. D. Manning. Naturalizing a programming language via interactive learning. In *Association for Computational Linguistics (ACL)*, 2017.