Augmenting Robotic Capabilities through Natural Language

by
Albert Yu

PhD Proposal

The University of Texas at Austin

Abstract

Augmenting Robotic Capabilities through Natural Language

Albert Yu, PhD student
The University of Texas at Austin, 2025

SUPERVISORS: Raymond Mooney, Roberto Martín-Martín

Despite rapid advances in language and vision models, current robots still lag far behind human physical capabilities due to the relative scarcity of real-world data compared to online text and images. How can we leverage abundant language data to advance robotic capabilities? Language provides semantic structure that facilitates the understanding of diverse data, improving sample efficiency in scarce data regimes. It also provides a natural communicative medium when interacting with and learning from humans.

To leverage the first benefit of language, we first take inspiration from how humans teach each other in video tutorials, through simultaneous video and language streams, to more efficiently teach robots new skills. We then show that language can bridge wide visual sim2real gaps, enabling robots to learn tasks with just a few real-world demonstrations by leveraging knowledge from imperfect simulation data. To leverage the second benefit of language, we explore how bidirectional dialog can enable robots to solve complex manipulation tasks by communicating to and collaborating with a wide distribution of human collaborators in the real-world. We develop a robotic framework that requests and proactively offers help through mixed-initiative, free-form dialog, enabling the robot to adapt to changing human preferences and each agent's physical capabilities to be strategically utilized. Finally, we discuss avenues of

future work, such as how human-robot collaboration can be facilitated through dialog-based replanning, how both agents can improve through bidirectional feedback, and how language-based guidelines extracted from manuals can enable robots to behave more safely and learn more quickly.

Table of Contents

Chapter 1: Introduction

1.1 The Purpose of Robotics

Robots augment human physical capabilities. Autonomous vehicles help reduce traffic collisions and enable a future where drivers can perform other tasks during their commuting time. Factory robots increase manufacturing production and warehouse throughput while reducing worker fatigue. Rescue and relief robots respond to natural and anthropogenic disasters, traveling to zones unsafe for humans to reach. Household robots promise to enable the elderly and disabled to perform cleaning, cooking, assistive feeding, and furniture assembly tasks, while saving everyone's time on undesirable tasks.

1.2 Robots in Unstructured Environments

Robots are generally deployed to two classes of domains—structured and unstructured, to perform two broad categories of tasks—manipulation and navigation. Across these four domain-task category pairs, current robotic capabilities vary widely. Robots are generally seen as most competent in structured environments (e.g., factory floors, where their motions can be programmed and executed fairly consistently and reliably) on locomotion and navigation tasks (including autonomous driving) where they must avoid obstacles while moving in the correct direction.

Manipulation tasks in unstructured environments seem more challenging for robots to learn because the primary task is to avoid certain obstacles while making contact to grasp and manipulate relevant objects on the scene. Unlike in structured environments, unstructured (also known as open-world) settings such as households contain infinite variations of object and scene configurations, making it quite difficult to program a robot to perform manipulation reliably and safely.

1.3 Data

For a robot to perform intelligently in unstructured settings, it must adapt to a dynamically changing world. Instead of pre-programmed motions, data becomes necessary for robots to learn patterns of acting intelligently, performing real-world tasks, and recovering from failures. While fields such as natural language and computer vision can leverage the entire internet's text and images, there is a scarcity of readily-available data that tells a robot the exact motor commands it should execute when it encounters some observation in the world for a task it hopes to achieve.

Despite their potential in structured environments, the physical capabilities of manipulation robots in unstructured environments still lag behind humans. This is ironic given the purpose of robotics is to augment human capabilities (Section 1.1). The longstanding data scarcity in robotics is a major reason for the shortcomings of current robotics. Data hungry algorithms and architectures that serve well in data-abundant fields may not be the easiest path forward for robotics, given the expensiveness of collecting real-world data and training large models. How might we confront these issues?

1.4 What Language Can Contribute to Robotics

To act in the world, the robot must first perceive the world through sight and/or touch. After acting, the robot must observe the physical change it has caused to decide its next action. This perception-action loop was the focus for robotic control for several decades. However, as brought up in Section 1.3, paired perception and action data for robotics is scarce and expensive, hindering the skills robots can learn. Other sources of data must be leveraged.

For several decades, natural language processing (NLP) was constrained to applications like translation, semantic parsing, and entity recognition. For robots to augment human physical capabilities, they must understand human need and intent, which are most easily specified through natural language. Robots that coexist

with and serve humans must understand language, a modality not included in the traditional perception-action paradigm.

Furthermore, language brings additional benefits to robotics. Individual words are inherently more abstract, conceptual, and meaning-based than, for instance, individual image pixels. While images provide precision to guide robot movements, language provides the ability for robots to extract semantic meaning from scarce data, associate rules-based behavior to specific visual inputs, and enable object-based reasoning in complex scenes and planning for long-horizon tasks.

Most importantly, freeform language data is plentiful, as evidenced by the success of LLMs since 2022 trained on the textual soup of the internet. This wealth of language-based knowledge has enabled incredible breakthroughs in automated programming and writing and common-sense reasoning and world knowledge, but challenges remain in bringing the generalization power of LLMs to physical tasks.

1.5 Proposal Overview

Given these advantages of natural language, in this proposal, we investigate two core lines of work to leverage natural language to expand robotic capabilities. First, language is a compressed store of meaning through which we can more efficiently learn from scarce robotics data. Second, language is a rich communicative medium through which humans and robots can collaborate and adapt to each other by expressing their intentions, preferences, and capabilities.

Along the first line of work, we explore how semantic meaning captured in language can be exploited to enable efficient learning for few-shot generalization to both new tasks and new domains. We take inspiration from how humans teach each other in video tutorials, through simultaneous video and language streams, to more efficiently teach robots new skills. We show that providing natural language instructions along with a single visual demonstration greatly improves sample efficiency when learning novel tasks compared to previous methods that teach robots with only one

modality. However, sometimes it is expensive for humans to teach robots through multiple simultaneous modalities in the real-world. Researchers have tried to leverage cheap, abundant simulation data to train robots, but the sim2real domain gap often degrades performance of simulation-trained policies. We show that language can help shape the learning of visual representations to better bridge large visual differences between sim and real, even when the sim2real gap is large and involves hard-to-simulate deformable objects. This enables robots to generalize to real-world domains with just a few real-world demonstrations.

Along the second line of work, we explore how language can enable smooth human-robot collaboration to accomplish complex manipulation tasks that the robot cannot easily learn during deployment. We argue that mixed-initiative dialog, which enables either the robot or human to start a conversation thread, can greatly facilitate human-robot collaboration, improving the adaptability of each agent to the other's capabilities. We develop a robotic framework capable of collaborating with a wide range of real human participants through bidirectional, mixed-initiative, free-form dialog. Our method achieves 50% more successful trials than the LLM baseline on long-horizon mobile manipulation tasks and was preferred by more than 75% of the 18 participants.

Finally, we discuss avenues of future work that attempt to unify these two lines of work. We propose a framework for using language-based guidelines from manuals to enable robots to behave more safely and learn more quickly, which uses both core advantages of language—language rules represent a compact distillation of prior experience, advice, and know-how from the human, and language rules are written by the human to communicate all the detailed preferences a human has for robot behavior. We also propose additional problem settings to enhance human-robot collaboration through mixed-initiative dialog: to address information asymmetry for task completion, to replan, and to provide bidirectional motion feedback so both agents can accomplish tasks together that neither can perform alone.

Chapter 2: Background and Related Work

For robots to perform useful manipulation tasks in unstructured environments, we established in Section 1.3 that robots need to learn intelligent patterns of behavior from data. There are two main praadigms for doing so: imitation learning and reinforcement learning.

2.1 Imitation Learning

In imitation learning (Pomerleau, 1988; Hussein et al., 2017), also known as behavioral cloning (BC), we assume access to an expert teacher that provides demonstrations of behaviors that the robot should imitate, conditioned on the observation o_t (i.e., RGB image, robot xyz end-effector position). Each expert demonstration (usually a human teleoperation of the robot) is collected as a sequence of observation-action tuples indexed by the timestep t: $[(o_t, a_t), ...]$, where a_t represents the action taken from observation o_t. a_t represents the robot command (i.e., desired change in xyz position of its end effector).

The goal of imitation learning is to train a policy $\pi_\theta : o_t \mapsto a_t$ that maps observations to actions, where $o_t \in \mathbb{R}^{d_o}$ and $a_t \in \mathbb{R}^{d_a}$. More commonly, $\pi_\theta : o_t \mapsto P(a_t|o_t)$, a probability distribution over possible actions. We want to learn the parameters θ of π, namely the weights of the neural network representing π, such that the error ε is minimized between the policy-predicted action $\hat{a}_t$, and the expert demonstration action a_t: $\varepsilon = ||a_t - \hat{a}_t||_2^2$, where $\hat{a}_t \sim \pi(\cdot|o_t)$. There are two losses to minimize ε: either ℓ_2 or log-likelihood. Let $\mathcal{D}$ be the set of demonstrations in our dataset, each of which is a trajectory τ_i.

When using ℓ_2 loss, the goal is to find the optimal policy parameters :

$$\theta = \underset{\theta}{\operatorname{argmin}} \sum_{\tau_i \in \mathcal{D}} \sum_{(o_{t,i}, a_{t,i}) \sim \tau_i} ||a_{t,i} - \pi_\theta(\cdot|o_{t,i})||_2^2 \tag{2.1}$$

Where the outer sum is over all expert trajectories in our dataset, and the inner sum is over all observation-action transitions in our trajectory.

When using log-likelihood loss, the goal is to find optimal policy parameters:

$$\theta = \underset{\theta}{\operatorname{argmin}} \sum_{\tau_i \in \mathcal{D}} \sum_{(o_{t,i}, a_{t,i}) \sim \tau_i} -\log \pi_\theta(a_{t,i}|o_{t,i}) \tag{2.2}$$

Essentially, this seeks to maximize the product of the policy's probabilities of predicting all actions $a_{t,i}$ in expert demo τ_i. This is the same optimization problem as maximizing the sum of log probabilities of the policy, which is equivalent to minimizing the negative sum of log probabilities.

By following either of these objectives, we train a policy π_θ to follow the expert's actions in observations seen by the expert, and rely on the interpolative power of neural networks to be robust to small deviations in the observation from the training distribution.

2.2 Reinforcement Learning

Often, expert demonstrations are hard to obtain. For instance, human expert-level teleoperation is difficult on multi-legged robots. Additionally, imitation learning yields policies upperbounded by the expert's performance. Sometimes these considerations make reinforcement learning (RL) better for training robots. Instead of expert demonstrations, RL requires a reward function (i.e. performance metric) that measures how good or bad a state-action pair is toward achieving the task.

RL operates in a Markov-Decision Process (MDP), defined as a tuple $\mathcal{M} = (S, A, R, S', \mathcal{P})$, where S is the set of start states from where actions A can be taken, S' is the set of next states the agent lands in after taking an action, $\mathcal{P} : S \times A \mapsto P(S')$ is the conditional probability distribution over next states after taking action $a \in A$ from state $s \in S$. At each timestep, the agent receives scalar reward $R(s, a)$, where R is a function mapping states and actions to a scalar.

The policy randomly explores the state-action space and finds a sequence of actions to maximize the sum of discounted rewards:

$$\mathbb{E}_{(s_t,a_t,s_{t+1})}\left[\sum_{t=0}^{\infty}\gamma^t R(s_t,a_t)\right] \tag{2.3}$$

Let $\pi(\cdot|s_t)$ represent the action predicted by the policy from state s_t. We define two functions that correspond to the "goodness" of a state when acting under a policy π. The first is the value function, or the expected sum of discounted rewards that the policy collects from state s_0 to termination.

$$V^\pi(s_0) = \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t R\big(s_t,\pi(\cdot|s_t)\big)\right] \tag{2.4}$$

$$= R(s_0,\pi(\cdot|s_0)) + \gamma\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t R\big(s_{t+1},\pi(\cdot|s_{t+1})\big)\right] \tag{2.5}$$

$$= R\big(s_0,\pi(\cdot|s_0)\big) + \gamma V^\pi(s_1) \tag{2.6}$$

The second is the quality function, or the expected reward from state s_0, taking action a_0, with all future actions dictated by policy π.

$$Q(s_0,a_0) = R(s_0,a_0) + \gamma Q^\pi(s_1,a_1) \tag{2.7}$$

Both V and Q can be defined recursively, as seen above, which is key to learning these functions using one of many RL algorithms for the policy to learn to perform optimal actions. We refer the reader to surveys detailing different classes of RL algorithms (Arulkumaran et al., 2017; Ghasemi et al., 2024).

Multiple works have explored using language to shape the reward function R for an RL agent (Nair et al., 2021; Goyal et al., 2019, 2020; Fan et al., 2022; Ma et al., 2023, 2024a). Some researchers have specifically explored using language models (LLMs) to generate code to tune a reward function when training robotic RL policies (Ma et al., 2024b; Yu et al., 2023a).

2.3 Multitask Learning

So far, we have discussed training our policy π to perform a specific task, either through imitation learning or RL. However, in unstructured environments, we want robots to be able to perform a wide range of tasks. Under the current tools discussed, we would need to train a single-task policy from scratch for each new task we want the robot to perform. This is unscalable and prevents us from leveraging data for one task in learning a related second task. To overcome these problems, multitask learning (Wilson et al., 2007a; Taylor and Stone, 2009) is necessary.

Earlier in the section, our policy was conditioned on only the state or observation $\pi : S \mapsto P(A)$. However, if we are trying to learn a multitask policy π, then the same state may demand different actions, depending on what the task is. Thus, we additionally condition π on a task conditioning vector C, so $\pi : S \times C \mapsto P(A)$. C can be as simple as a one-hot vector, goal image embedding, or natural language instruction embedding.

Work in multi-task learning suggests that training on a wide range of tasks, instead of the single target task, helps the robot learn shared perceptual representations across the different tasks, improving generalization (Kalashnikov et al., 2021; Yu et al., 2019). The most straightforward way to condition multi-task policies is through one-hot vectors (Ebert et al., 2021; Kalashnikov et al., 2021; Walke et al., 2022; Yu et al., 2021b). Multi-task robotic policies have also been studied in other settings and contexts, such as hierarchical goal-conditioned policies (Gupta et al., 2022a), probabilistic modeling techniques (Wilson et al., 2007b), distillation and transfer learning (Parisotto et al., 2015; Teh et al., 2017; Xu et al., 2020; Rusu et al., 2015), data sharing (Espeholt et al., 2018; Hessel et al., 2019), gradient-based techniques (Yu et al., 2020), policy modularization (Andreas et al., 2017; Devin et al., 2017) and task modularization (Yang et al., 2020a). However, one-hot conditioning fails to leverage similarity information between related tasks. We propose an approach to address this issue in Section 3.

2.4 Vision and Language

Researchers have explored how to ground natural language in visual concepts or objects by developing models that can understand both visual and textual modalities. Vision-language research first found success with problems like visual question-answering (VQA; Agrawal et al. (2015); Marino et al. (2019)), image captioning (Socher et al., 2014; Kiros et al., 2014; Mao et al., 2014; Vinyals et al., 2014), and video summarization (Venugopalan et al., 2015; Yao et al., 2015) in the mid 2010s. The first attempts at training models for the inverse problems of text-to-image generation (Mansimov et al., 2015; Reed et al., 2016) and text-to-video generation (Pan et al., 2017; Li et al., 2017) came shortly after.

Transformers arose as a neural architecture for pure language tasks but soon unified the processing of both images and text with the first vision-language transformers (Lu et al., 2019; Sun et al., 2019; Li et al., 2019; Tan and Bansal, 2019; Su et al., 2019; Chen et al., 2019), some of which have been applied to robotic control (Shridhar et al., 2021; Zeng et al., 2020; Shridhar et al., 2022; Cui et al., 2022; Zeng et al., 2022; Brohan et al., 2023, 2022). In addition to architectures, researchers have also examined learning better vision-language joint representations (Radford et al., 2021; Zhai et al., 2021; Zhu et al., 2023) to improve robotic control (Nair et al., 2022; Shridhar et al., 2021, 2022).

These prior works in vision-language research form the technical basis of enabling language-conditioned, vision-based robotics in the real-world. However, visual representations learned from these prior approaches yield robotic policies that largely fail to generalize to large domain shifts. We address this problem in Section 4.

2.5 Natural Language for Robot Task Specification

As mentioned in Section 2.3, multitask policies can take in as context C a natural language embedding of the task instruction, making it a language-conditioned multitask policy (Jang et al., 2021; Lynch and Sermanet, 2021; Mees et al., 2021, 2022;

Shao et al., 2020; Sodhani et al., 2021; Silva et al., 2021; Karamcheti et al., 2021; Garg et al., 2022). Pretrained language embedding spaces normally preserve a notion of semantic similarity through distance—that is, two strings similar in meaning will be encoded into two language embeddings close in distance. This means that language-conditioned multitask policies have two benefits: robustness (slight rewordings of the language instruction do not meaningfully change the language embedding) and generalizability to new tasks (a new, related task will have a language embedding close to those of semantically related tasks that the policy has already trained on). However, teaching robots only through language can lead to a lot of ambiguities, which we address in Section 3.

LLMs have also been used as task planners (Huang et al., 2022; Ahn et al., 2022; Chen et al., 2022; Raman et al., 2023; Choi et al., 2025; Luo et al., 2023), as code generators that dictate a robotic policy's behavior (Liang et al., 2022; Li et al., 2024; Huang et al., 2023), and as part of a hierarchical policy where the higher level produces language and the lower level produces fine-grained robot actions (Shi et al., 2025, 2024; Belkhale et al., 2024).

2.6 Natural Language for Human-Robot Interaction

While language can serve as a monologic means to teach or instruct robots, it can also serve as a medium for dialog to enable Human-Robot Interaction (HRI). Some systems integrate LLMs as task planners or delegators (Wang et al., 2024a; Mandi et al., 2023; Feng et al., 2024) for tasks like real-world cooking (Wang et al., 2024a) and object sorting (Mandi et al., 2023), where task delegations are communicated through dialog. Other systems implement a leader-follower paradigm in simulated worlds, where the leader instructs the follower in natural language (Suhr et al., 2022; Kojima et al., 2021; Team et al., 2022; Gao et al., 2023). Another line of work empowers the robot to ask humans for clarifications (Ren et al., 2023), request assistance (Bennetot et al., 2020; Knepper et al., 2013; Veloso et al., 2015), or inform humans of their

observations (Chen et al., 2010; Mutlu et al., 2006; Cascianelli et al., 2018). We argue that these prior works do not exploit the full flexibility of language as a communicative medium and present a new framework for bidirectional dialog and HRI in Section 5.

Chapter 3: Using Both Demonstrations and Language Instructions to Efficiently Learn Robotic Tasks

3.1 Introduction

We mentioned the two core lines of work in this proposal for leveraging natural language to improve robotic capabilities in Section 1.5. This chapter describes our initial work along the first area of leveraging language as a store of meaning to generalize to new robotic tasks from scarce data, and it was published at ICLR 2023 (Yu and Mooney, 2022).

Say we have a household robot, and we want to teach it new tasks. What is the best way to do so? Looking at ourselves, we humans often learn complex tasks through multiple concurrent modalities, such as simultaneous visual and linguistic (speech/captioning) streams of a video tutorial. One might reasonably expect robotic policies to also benefit from multi-modal task specification. However, previous work in multitask policies condition only on a single modality during evaluation: one-hot embeddings, language embeddings, or demonstration/goal-image embeddings. Each has limitations.

One-hot encodings for each task (Kalashnikov et al., 2021; Ebert et al., 2021) suffice for learning a repertoire of training tasks but generalize poorly to novel tasks, since one-hot embedding spaces do not leverage semantic similarity between tasks to more rapidly learn additional tasks. Conditioning policies on goal-images (Nair et al., 2017, 2018; Nasiriany et al., 2019) or training on video demonstrations (Smith et al., 2020; Young et al., 2020) often suffer from ambiguity, especially when there are large differences between the environment of the demonstration and the environment the robot is in, hindering the understanding of a demonstration's true intention. Language-conditioned policies (Blukis et al., 2018, 2019; Mees et al., 2021, 2022)

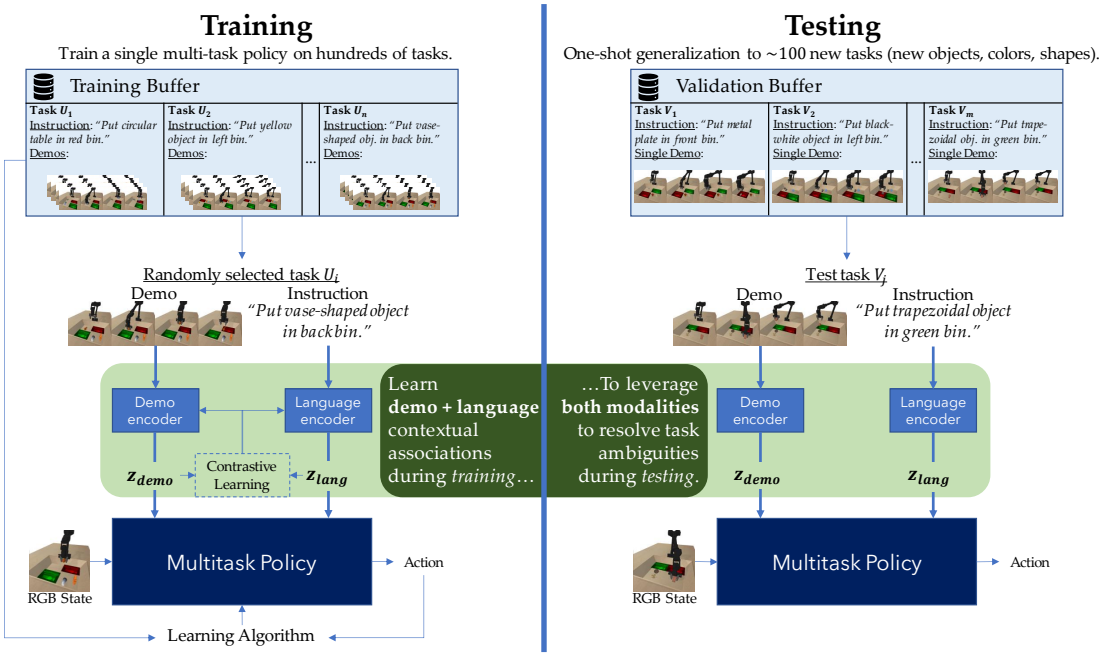

Figure 3.1: **DeL-TaCo Overview.** Unlike prior multitask methods that condition on a single task specification modality, DeL-TaCo simultaneously conditions on both language and demonstrations during training and testing to resolve any ambiguities in either task specification modality, enabling better generalization to novel tasks and significantly reducing teacher effort for specifying new tasks.

often face greater ambiguity challenges, since humans specify similar tasks in very linguistically dissimilar ways at different levels of granularity, sometimes with novel nouns and verbs not seen during training.

We posit that current unimodal task representations are often too inefficient and ambiguous for novel task specification. In these tasks, current task-conditioning methods would need either a large number of diverse demonstrations to disambiguate the intended task, or a long, very detailed, fine-grained language instruction. Both are difficult for novice users to provide. We argue that conditioning the policy on *both* a demonstration *and* language not only ameliorates the ambiguity issues with language-only and demonstration-only specifications, but is *far cheaper to provide*.

We propose DeL-TaCo (Figure 3.1), a new task embedding scheme comprised of two component modalities that contextually complement each other: demonstra-

tions of the target task and corresponding language descriptions. With bimodal task embeddings, ambiguity is bidirectionally resolved: instructions disambiguate intent in demonstrations, and demonstrations help ground novel noun and verb tokens by conveying what to act on, and how. To summarize, we (1) propose DeL-TaCo for training and integrating demonstrations and language into joint task embeddings for few-shot novel task specification, and (2) show that DeL-TaCo significantly lowers teacher effort in novel task-specification and improves generalization performance over previous unimodal task-conditioning methods.

3.2 Related Work

Our work is most related to BC-Z (Jang et al., 2021), which trains a video demonstration encoder to predict the pretrained embeddings of corresponding language instructions, while jointly training a multi-task imitation learning policy conditioned on *either* the instruction *or* demonstration embeddings. Lynch and Sermanet (2021); Mees et al. (2021) similarly learn a similar policy conditioned on either language or goal images. However, during testing, these policies are conditioned on only one of the two modalities, whereas ours uses both modalities during training and testing, which we show improves generalization and reduces human teacher effort on a broad category of tasks.

3.3 Problem Setting

3.3.1 Multi-task Imitation Learning

We define a set of n tasks $\{T_i\}_{i=1}^n$ and split them into training tasks U and test tasks V, where (U, V) is a bipartition of $\{T_i\}_{i=1}^n$. For each task T_i, we assume access to a set of m expert trajectories $\{\tau_{ij}\}_{j=1}^m$ and a single language description l_i. Given continuous state space $\mathcal{S}$, continuous action space $\mathcal{A}$, and task embedding space $\mathcal{Z}$, the goal is to train a Markovian policy $\pi : \mathcal{S} \times \mathcal{Z} \to \Pi(\mathcal{A})$ that maps the current state and task embeddings to a probability distribution over the continuous action space.

During training, we assume access to a buffer $\mathcal{D}_{\text{train}}$ of trajectories for only the tasks in U and their associated natural language descriptions. We define each trajectory as a fixed-length sequence of state-action pairs $\tau_{ij} = \left[\left(s_{0,j}^{(i)}, a_{0,j}^{(i)}\right), \left(s_{1,j}^{(i)}, a_{1,j}^{(i)}\right), ...\right]$, where j is the trajectory index for task $T_i \in U$ with task embedding z_i. We use behavioral cloning (BC) to update the parameters of π to maximize the log probability of $\pi\left(a_{t,j}^{(i)} | s_{t,j}^{(i)}, z_i\right)$, though our framework is agnostic to the learning algorithm and would work for RL approaches as well.

During evaluation, we assume access to a buffer $\mathcal{D}_{\text{val}}$ of trajectories for only the tasks in V and their associated natural language descriptions. Unlike $\mathcal{D}_{\text{train}}$ where we have m demonstrations for each task, in $\mathcal{D}_{\text{val}}$ we have just a single demonstration for each task. For all test tasks $T_i \in V$, we rollout the policy for a fixed number of timesteps by taking action $a_t \sim \pi(a|s_t, z_i)$. The z_i for all test tasks is computed beforehand and held constant throughout each test trajectory.

3.3.2 Task Encoder Networks

To obtain the task embedding z_i, we have two encoders (either trained jointly with policy π, or frozen from a pretrained model): a *demonstration encoder*, $f_{demo} : \tau_{ij} \mapsto z_{demo,i}$ mapping trajectories of task T_i to demonstration embeddings, and a *language encoder*, $f_{lang} : l_i \mapsto z_{lang,i}$ mapping task instruction strings l_i to language embeddings. Previous work has explored using language embedding $z_{lang,i}$ *or* goal image/demonstration embedding $z_{demo,i}$ as the task embedding z_i, but DeL-TaCo uses the bimodal task embedding $z_i = [z_{demo,i}, z_{lang,i}]$ during training and testing.

3.4 Method

3.4.1 Architecture

Demonstration and Language Encoders. The encoder f_{demo} is a CNN network trained from scratch. Following Jang et al. (2021), we input the demonstration as an array of $m \times n$ frames (in raster-scan order) from the trajectory for

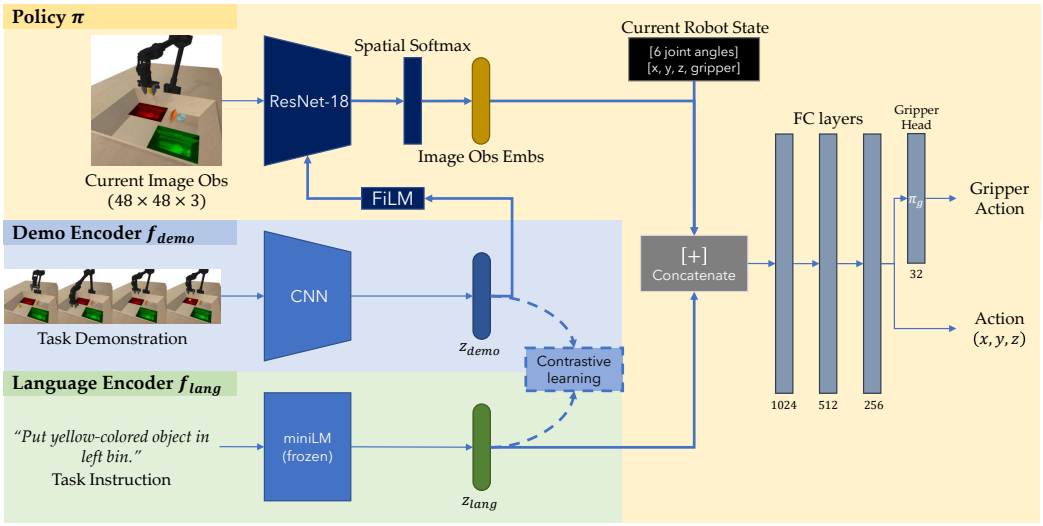

Figure 3.2: **Method Architecture.** DeL-TaCo uses three main networks: the policy π, a demonstration encoder f_{demo}, and a language encoder f_{lang}. During both training and testing, the policy is conditioned on the demonstration and language embeddings for the task.

faster processing. (We use $(m, n) = (1, 2)$ or $(2, 2)$ in our experiments.) We freeze a pretrained miniLM (Wang et al., 2020) as the encoder f_{lang}, where $z_{lang,i}$ is simply the average of all miniLM-embedded tokens in l_i (we found this works better than taking the [CLS] token embedding).

Policy Network. We use a ResNet-18 (He et al., 2015) as the visual backbone for the policy π, followed by spatial softmax (Finn et al., 2016) and linear layers.

Task Conditioning Architecture. BC-Z (Jang et al., 2021) inputs the task embedding into the ResNet backbone via FiLM (Perez et al., 2018) layers, which apply a learned affine transformation to the intermediate image representations after each residual block. BC-Z's task embeddings are either from demonstrations *or* language. Since our policy conditions on both, the main architectural decision was finding the best way to feed task embeddings from multiple modalities into the policy.

Empirically, a simple approach performed best. The demonstration embeddings z_{demo} are fed into the policy's ResNet backbone via FiLM, while the language

task embeddings z_{lang} and robot proprioceptive state (6 joint angles, end-effector xyz coordinates, and gripper open/close state) are concatenated to the output of the spatial softmax layer. Our full network architecture is shown in Figure 3.2.

3.4.2 Training and Losses

During each training iteration, we sample a size k subset of training tasks $M = \{T_{m_1}, ..., T_{m_k}\} \subset U$. Given a trajectory τ_{ij} for task T_{m_i} and corresponding natural language instruction l_i, we compute the demonstration embeddings $z_{emb,m_i} = f_{demo}(\tau_{ij})$ and language embeddings $z_{lang,m_i} = f_{lang}(l_i)$. We collect the embeddings of tasks in M in matrices $Z_{demo} = [z_{demo,m_1}, ..., z_{demo,m_k}]$ and $Z_{lang} = [z_{lang,m_1}, ..., z_{lang,m_k}]$.

To train the demonstration encoder, Jang et al. (2021) use a cosine distance loss to directly regress demonstration embeddings to their associated language embeddings. However, this causes demonstration embeddings to be essentially equivalent to the associated language embeddings for each task, undercutting the value of passing both to our policy. To preserve information unique to each modality while enabling the language and demonstration embedding spaces to shape each other, we train with a CLIP-style (Radford et al., 2021) contrastive loss for our demonstration encoder:

$$\mathcal{L}_{demo}(Z_{demo}, Z_{lang}) = CrossEntropy\left(\frac{1}{\beta} Z_{demo}^{\top} Z_{lang}, I\right) \tag{3.1}$$

where I is the identity matrix and β is a tuned temperature scalar. For some trajectory of state-[xyz action, gripper action] pairs $x_{t,i,j} = \left(s_{t,j}^{(i)}, \left[a_{t,j}^{(i)}, g_{t,j}^{(i)}\right]\right)$ extracted from expert demonstration τ_{ij} for task T_{m_i}, we use a weighted combination of standard BC log-likelihood loss for the xyz actions and MSE loss for the gripper actions. We abbreviate z_l and z_d for z_{lang} and z_{demo}:

$$\mathcal{L}_{\pi}(\tau_{ij}) = \sum_{x_{t,i,j} \in \tau_{ij}} -\log \pi\left(a_{t,j}^{(i)} | s_{t,j}^{(i)}, z_{d,m_i}, z_{l,m_i}\right) + \alpha_g \left\| g_{t,j}^{(i)} - \pi_g\left(s_{t,j}^{(i)}, z_{d,m_i}, z_{l,m_i}\right) \right\|_2 \tag{3.2}$$

π outputs a distribution over xyz actions, and the gripper head π_g of the policy network outputs a scalar $\in [0, 1]$ for the gripper action trained on a tuned $\alpha_g > 0$.

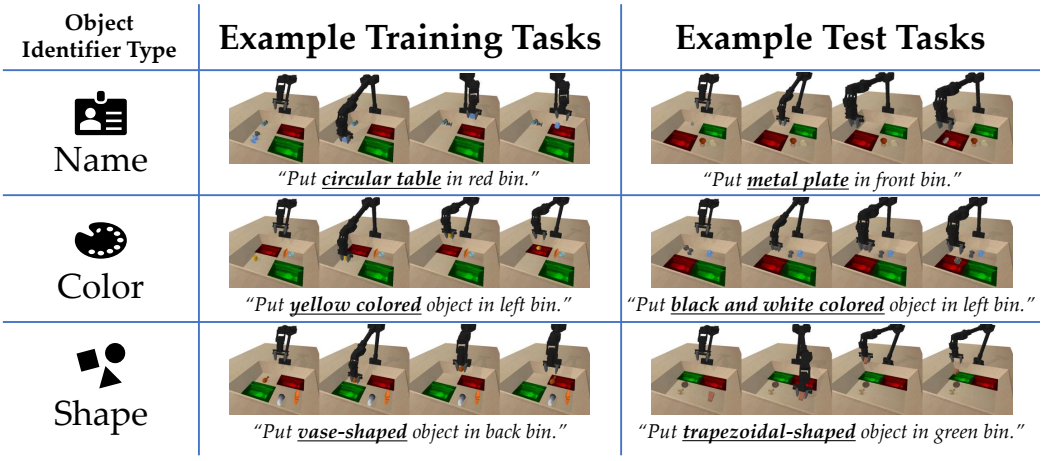

Figure 3.3: **Sample train+test tasks**, grouped by object identifier types (underlined in instructions).

Both f_{demo} and π networks are trained jointly with the following loss, for a tuned $\alpha_d > 0$:

$$\mathcal{L}(\pi, f_{demo}, f_{lang}) = \mathcal{L}_\pi(\tau_{ij}) + \alpha_d \mathcal{L}_{demo}(Z_{demo}, Z_{lang}) \tag{3.3}$$

Note $\mathcal{L}_\pi(\tau_{ij})$ is summed over all trajectories in the batch of training tasks M (double summation in Equation 3.2 ommitted for brevity). f_{lang} has no loss term because we freeze the pretrained language model and rely on its pretrained embedding space to shape the demonstration encoding space.

3.4.3 Evaluation

During evaluation, we want the robot to perform some novel task $T_v \in V$. Recall that $T_v \notin U$, our set of training tasks. From Section 3.3.1, we have access to a validation task buffer $\mathcal{D}_{\text{val}}$ with a single demonstration τ_v and a natural language instruction l_v of task T_v. We encode the demonstration with f_{demo} and the language with f_{lang} and pass both task embeddings to the policy.

3.5 Experiments

We empirically investigate the following questions: (1) Does DeL-TaCo improve generalization performance on novel tasks? (2) If so, how much teacher effort

does DeL-TaCo reduce over specifying tasks with either modality alone?

3.5.1 Setup

Environment. We develop a Pybullet (Coumans and Bai, 2007-2022) simulation environment with a WidowX 250 robot arm, 32 possible objects, and 2 containers. The action space is end-effector (x, y, z) delta positions, plus the binary gripper state (closed/opened). We subdivide the workspace into four quadrants. Two quadrants are randomly chosen to contain the two containers, and three of the 32 possible objects are dropped at random locations in the remaining two quadrants. RGB image observations are size $48 \times 48 \times 3$.

Task Objective. We design a set of pick-and-place tasks where the objective is to grasp the target object and place it in the target container. The scene contains three visually distinct objects (one is the target object) and two visually distinct containers (one is the target container). Thus, a robotic policy that disregards both the task demonstration and instruction and picks any random object and places it into any random container would succeed with 1-in-6 odds.

Language Instructions for Each Task. Figure 3.3 shows a selection of our training and testing tasks. Each task is specified through language with a single template-based instruction of the format "put [target object identifier] in [target container identifier]."

We make this environment more challenging by having task instructions refer to containers by either their color or quadrant position and objects by either their name, color, or shape. The multiple identifiers help simulate ambiguity that arises from informal human instructions, where different humans may refer to the same object or container through different attributes, enabling demonstrations and instructions to complement each other when the robot learns a new task. In total, there are 50 target object identifiers and 6 target container identifiers, giving us 300 pick-and-place tasks. We train and evaluate on a bipartition of these 300 tasks.

Table 3.1: Evaluation on Novel Objects, Colors, and Shapes. (p) = pretrained.

Demo Encoder	Lang. Encoder	Task Conditioning	Success ± SD (%)
–	–	One-hot (lower bound)	4.9 ± 1.7
–	–	One-hot Oracle (upper bound)	69.3 ± 7.4
–	miniLM (p)	Lang.-only	17.1 ± 2.2
CNN	–	Demo-only	20.8 ± 2.4
CNN	miniLM (p)	**DeL-TaCo (ours)**	$\mathbf{25.8 \pm 3.4}$
CNN	–	BC-Z; Demo-only	6.7 ± 2.3
CNN	miniLM (p)	MCIL; Demo-only + Lang.-only	7.5 ± 1.2

Data. Using a scripted policy, we collect 200 successful demonstrations per training task, and a single successful demonstration per test task. All demonstrations are 30 timesteps long. Our training buffer contains roughly 40,000 trajectories.

3.5.2 Generalization Performance on Novel Tasks

Table 3.1 shows our generalization performance on 102 test tasks when trained on 198 tasks. In this setting, the robot must not only know how to pick-and-place the 8/32 objects it has never seen during training, but must also understand novel instructions that refer to these objects by either their name, color, or shape.

We lower-bound the performance of our task conditioning methods by first running a one-hot conditioned policy, with the expectation that it performs worse than conditioning on language and/or demonstrations for the reasons mentioned in Section 3.1. As an upper-bound, we directly train a one-hot oracle on only the 102 evaluation tasks and evaluate on those same tasks. No other method in the table is trained on any evaluation tasks.

Next, we examine the performance of policies conditioned with only language, with only one demonstration, and with both (DeL-TaCo). The language-only policies do not involve training f_{demo}, and only the language instruction embeddings are fed into the policy via FiLM during training and testing. The demo-only policies involve a trained f_{demo}, but during training and testing, only the demonstration embedding z_{demo} is passed into the policy via FiLM. DeL-TaCo (ours) conditions on *both* demonstration *and* language during training and testing.

Table 3.2: Value of Language. Evaluation on Novel Objects, Colors, and Shapes.

Task Conditioning	Demo-only					DeL-TaCo (ours)
# demos per test-task finetuned on	0	10	25	50	100	0
Success Rate (%)	20.8	23.4	24.6	<u>26.1</u>	32.9	<u>25.8</u>
± SD (%)	±2.4	±1.8	±2.5	±2.6	±2.5	±3.4

DeL-TaCo achieves the highest performance, increasing the success rate of the second-best conditioning method, demo-only, from 20.8% to 25.8%. Both methods using demonstration embeddings outperform the language-conditioned policy perhaps because a visual demonstration is important in conveying the nature of the chosen object and how the robot should manipulate it.

Prior methods like <u>BC-Z</u> (Jang et al., 2021) perform worse than DeL-TaCo because its demo encoder is trained to directly regress z_{demo} to z_{lang}, hindering it from performing better than solely using z_{lang} during testing. <u>MCIL</u> (Lynch and Sermanet, 2021), also performs worse than DeL-TaCo because without any task encoder loss term, learning a well-shaped task embedding space is more difficult, hurting generalization performance.

3.5.3 How many demonstrations is language worth?

To answer our second question, we further finetune the demo-only policy on a variable number of test-task expert demonstrations. Results are shown in Table 3.2. The demo-only policy only starts to match and surpass DeL-TaCo (underlined) when it is finetuned on 50 demonstrations (underlined) *per evaluation task* (a total of around 5,000 demonstrations for all test tasks combined). This suggests that surprisingly, specifying a new task to DeL-TaCo with a single demonstration and language instruction performs as well as specifying a new task to a demo-only policy with a single demonstration *and finetuning it on 50 additional demonstrations of that task*. This showcases the immense value of language in supplementing demonstrations for novel task specification, significantly reducing the effort involved in teaching robots novel tasks over demonstration-only methods.

Chapter 4: Natural Language Can Help Bridge the Sim2Real Gap

4.1 Introduction

In Chapter 3, we saw how a simple language instruction could contain information equivalent to 50 demonstrations when training robots to perform new tasks. We also saw that more broadly, teaching robots new tasks is often better done with both demonstrations and language, rather than a single modality alone. In this chapter, we further extend the idea of using language as a store of meaning to learn from scarce data, but instead of few-shot generalization to new tasks, we demonstrate how language can enable few-shot generalization to new domains. This chapter represents work published in RSS 2024 (Yu et al., 2024).

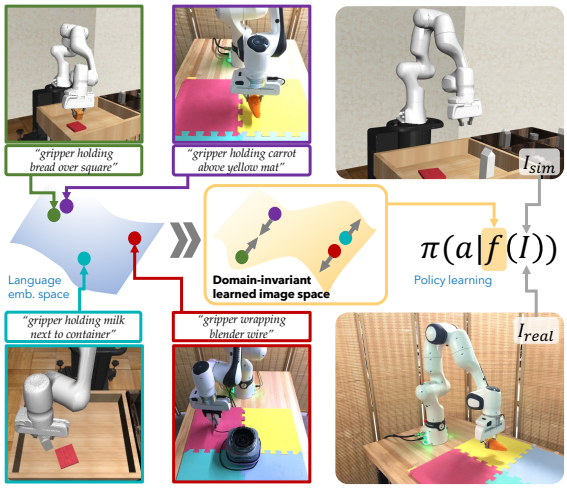

Figure 4.1: **Bridging the sim2real gap with language**. Robot images from simulation and the real world with similar language descriptions (*green & purple borders*) are mapped to similar features in language embedding space, while sim and real images with different language descriptions (*teal & red*) are mapped to faraway locations. We learn a policy conditioned on these image embeddings from both sim and real images (*right*).

Researchers have recently achieved significant success on household tasks with visual imitation learning (IL) (Schaal, 1999; Brohan et al., 2022). Some researchers are attempting to generalize visual IL to any target domain by collecting large, expensive datasets of demonstrations from many domains (Brohan et al., 2022, 2023; Padalkar et al., 2023). But can we instead transfer a policy trained on cheaply acquired, diverse simulation data to a real-world target task with just a few demonstrations?

We propose creating a domain-agnostic visual representation for policy training. Such a representation should enable the policy to use the simulation image-action data as an inductive bias to learn with few-shot real world data. This representation must allow the policy to tap into the right distribution of actions by being broad enough to capture the task-relevant semantic state from image observations, yet fine-grained enough to be conducive to low-level control. For instance, a sim and real image observation, both showing the robot gripper a few inches above a pan handle, should lie close together in the image embedding space to lead to similar actions, even if the two images have large differences in pixel space.

How might we acquire supervision for learning such a visual representation? Language is an ideal medium for providing it. Descriptions of task-relevant features in image observations, such as whether or not a gripper is close to a pan handle, serve as a unifying signal to align the representations of images between sim and real. We hypothesize that if a sim and real image have similar language descriptions (e.g., "the gripper is open and right above the pan handle"), then their underlying semantic states are also similar, and thus the actions the policy predicts conditioned on each image should also be semantically similar (e.g., moving downward to reach the pan handle). The pretrained embedding space of large language models (LLMs) offers a well-tuned signal that can be leveraged to measure the semantic similarity between real and sim images via their associated language descriptions (see Fig. 4.1). This simple insight allows us to learn a domain-agnostic visual representation to bridge the visual sim2real gap.

We introduce Lang4Sim2Real, a lightweight pretrain-finetune framework for transferring between any two domains that have large visual differences but contain data across a similar distribution of tasks. Lang4Sim2Real has the following main advantages over prior sim2real efforts: (1) alleviates the need for the engineering-intensive task of matching a sim environment to the real environment both visually and semantically, (2) enables sim2real transfer on tasks involving hard-to-simulate, deformable objects, and (3) bridges a wide sim2real gap that includes differences

in: camera point-of-view (1st vs 3rd person), friction and damping coefficients, task goals, robot control frequencies, and initial robot and object position distributions. To our knowledge, this is the first work that shows that using language to learn a domain-invariant visual representation can help improve the sample efficiency and performance of sim2real transfer.

4.2 Related Work

Vision-only pretraining improves performance on image-based robotic policies with objectives ranging from masked image modeling (Radosavovic et al., 2023), image reconstruction (Zhao et al., 2022; Gupta et al., 2022b; Seo et al., 2023), contrastive learning (Laskin et al., 2020; He et al., 2020), video frame temporal ordering (Jing et al., 2023), future frame prediction (Zhao et al., 2022), and image classification (Yuan et al., 2022; Wang et al., 2022) on internet-scale datasets (Deng et al., 2009; Grauman et al., 2021; Goyal et al., 2017; Damen et al., 2018). However, vision-only representations are typically not robust to wide sim2real domain shifts.

Prior work in **vision-language pretraining** was described in Section 2.4. Zhu et al. (2023) used language to align representations learned across multiple modalities including depth and audio. Instead of using language to bridge modalities, our approach uses language to bridge visual representations between domains.

While we approach **sim2real** through vision-language pretraining, domain randomization (Andrychowicz et al., 2020; Matas et al., 2018; Tobin et al., 2017) and system identification (Yu et al., 2017; Kaspar et al., 2020) remain popular approaches. However, these are engineering-intensive procedures we seek to avoid.

Several methods have been proposed to learn **domain invariant representations** in pixel-space with GANs (James et al., 2019; Bousmalis et al., 2017; Ho et al., 2021; Rao et al., 2020) or with semantic segmentation and depth maps (Müller et al., 2018; Ai et al., 2023). However, these are high-dimensional and computationally expensive representations, in the case of GANs.

4.3 Problem Setting

In this work, we address the problem of few-shot visual imitation-learning (IL): learning a visuomotor manipulation policy in the real world based on a few real-world demonstrations. We cast sim2real as a $k+1$ multi-task IL problem: k tasks from simulation and the target task (with a few demonstrations) in the real world. In general terms, we assume a *source domain* in which data can be acquired cheaply and a *target domain* where data is expensive to collect.

In our setting, we consider access to two datasets across two domains: $\mathcal{D}^{\text{s}}$, which spans multiple tasks in the source domain, and $\mathcal{D}^{\text{t}}_{\text{target}}$, demonstrations of the target task in the target domain. Thus, we assume that $|\,\mathcal{D}^{\text{s}}\,| >> |\,\mathcal{D}^{\text{t}}_{\text{target}}\,|$, due to how expensive target domain data collection is (such as in the real world). We make two simple assumptions about the similarity of the two domains. First, we assume the source and target tasks are all of the same general structure, such as multi-step pick-and-place task compositions, but with different objects and containers across different subtasks. Second, to train a common policy for both domains, we assume the domains share state and action space dimensionality.

All of our datasets are in the form of expert trajectories. Each trajectory, $\tau = \{(I_t, s_t, [a_t, l_t], l_{task})\}$, is a sequence of tuples containing an image observation, I_t ($128\times$ 128 RGB), robot proprioceptive state, s_t (end effector position and joint angles), and a language instruction of the task, l_{task}. Note that l_{task} is the same over all timesteps of all trajectories in a given task. $[a_t, l_t]$ denotes that a trajectory may optionally also include robot actions (in which case we consider the trajectory a full demonstration) and/or a language description of the image I_t. In the following sections, we identify with $\tau[L]$ a trajectory with language descriptions l_t, but no actions a_t. Similarly, $\tau[A]$ is a full demonstration with actions, a_t, but no language descriptions, l_t. The language labels for images (l_t) can be automatically generated from a programmatic function that maps image observations to language scene descriptions depending on the relative position between the robot and the objects in the scene. During pretraining, we use

$\tau[L]$ image-language (I_t, l_t) pairs from $\mathcal{D}^{\text{s}} \cup \mathcal{D}^{\text{t}}_{\text{target}}$. During policy learning, we use $\tau[A]$ data: $(I_t, s_t, a_t, l_{task})$ tuples from $\mathcal{D}^{\text{s}} \cup \mathcal{D}^{\text{t}}_{\text{target}}$.

4.4 Method

In our method, we adopt the common *pretrain-then-finetune* learning paradigm (see Fig. 4.2). First, we pretrain an image backbone encoder on cross-domain language-annotated image data (Sec. 4.4.1). Then, we freeze this encoder and train a policy network composed of trainable adapter modules and a policy head to perform BC on action-labeled data from both domains (Sec. 4.4.2).

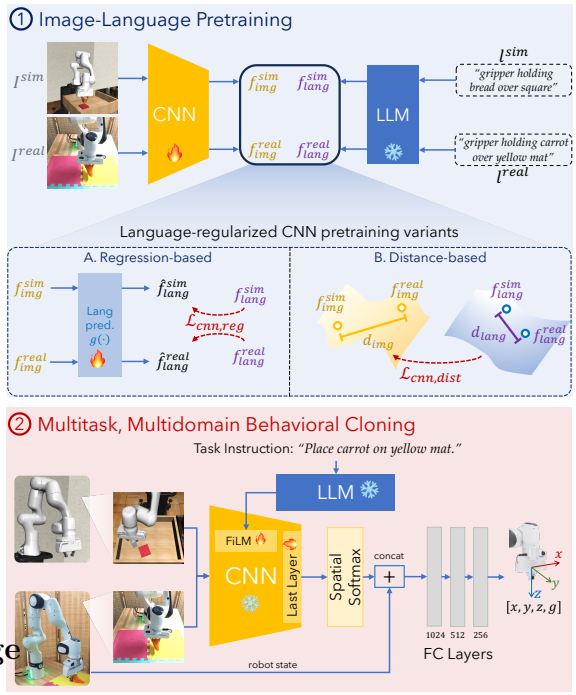

Figure 4.2: **Method.** (i) Top: Image-Language Pretraining. (ii) Bottom: During multitask, multidomain BC, we freeze our pretrained f_{cnn}, add adapter modules and a policy head, then train the resulting multitask language-conditioned policy on $\mathcal{D}^{\text{s}} \cup \mathcal{D}^{\text{t}}_{\text{target}}$.

4.4.1 Cross-Domain Image-Language Pretraining

We first automatically label trajectories with language either during data collection with heuristics, or in hindsight with off-the-shelf-based object detectors. After this data is collected, our first step in Lang4Sim2Real involves learning a domain-invariant representation that leverages simulation data for few shot IL. For that, we need to learn an image observation encoder, $f_{cnn} : I_t \rightarrow \mathbb{R}^{d_{cnn}}$, that preserves the semantic similarity of scenes in images between the two domains. For instance, if both image I^s from $\mathcal{D}^{\text{s}}$ (sim) and image I^t from $\mathcal{D}^{\text{t}}_{\text{target}}$ (real world) show

the robot's gripper open and a few inches above the object to grasp, even if from different viewing angles, then we want their image embeddings to be close together in the learned image encoding space. This will facilitate policy learning later, as the policy will need to draw from a similar distribution of actions for similar scene semantics, which are now already mapped into similar visual features.

Theoretically, off-the-shelf pretrained vision-language models (VLMs) (Radford et al., 2021; Nair et al., 2022) should already possess these properties as they were trained on a massive distribution of image and language data. However, in the context of robot manipulation, pretrained VLMs tend to encode all observations of the trajectory into a very narrow region of the embedding space without sufficient distinction for task-relevant, semantic aspects of the image such as the location of the gripper in relation to the manipulated objects. This renders them unsuitable without additional finetuning for our application (see Sec. 4.6).

In Lang4Sim2Real, we propose an alternative approach to obtain a visual representation that preserves the semantic similarity of scenes in images between the two domains. We train a ResNet-18 (He et al., 2015) from scratch as our image encoder using image-language tuples (I^s, l^s) from $\mathcal{D}^{\text{s}}$ and (I^t, l^t) from $\mathcal{D}^{\text{t}}_{\text{target}}$. We denote this vision language pretraining dataset as $\mathcal{D}_{VL} = \{(I^d, l^d) : (I^d, l^d) \in \mathcal{D}^{\text{s}} \cup \mathcal{D}^{\text{t}}_{\text{target}}\}$, where d is either the source or target domain. The images are observations collected during 100 demonstrations from each of the tasks in $\mathcal{D}^{\text{s}}$ and 25-100 demonstrations from $\mathcal{D}^{\text{t}}_{\text{target}}$, totaling around 10k images per domain. We assume that the two sets of language descriptions in $\mathcal{D}^{\text{s}}$ and $\mathcal{D}^{\text{t}}_{\text{prior}}$ are similarly distributed; otherwise, language may not help learn domain-invariant features between $\mathcal{D}^{\text{s}}$ and $\mathcal{D}^{\text{t}}$.

To effectively leverage language as a bridge between visually different domains, we need a well-tuned (frozen) language model, $f_{lang} : l \rightarrow \mathbb{R}^{d_{lang}}$, to map strings to $d_{lang}-$dimensional language embeddings. We use off-the-shelf miniLM (Wang et al., 2020), since prior work (Mees et al., 2022) has demonstrated its effectiveness for language-conditioned control policies.

We propose two image-language pretraining variants in Lang4Sim2Real to obtain a visual representation based on language supervision (see Fig. 4.2(i)A-B):

4.4.1.1 Language-Regression

Our first variant is a straightforward use of language supervision to shape the image embedding space: predicting the language embedding of the description, l^d, given the embedding of the corresponding image, I^d. Let $g : \mathbb{R}^{d_{cnn}} \to \mathbb{R}^{d_{lang}}$ be a single linear layer (language predictor in Fig. 4.2(i)(A)) trained to minimize the following loss:

$$\mathcal{L}_{cnn,reg}(\mathcal{D}_{VL}) = \left\| g\left(f_{cnn}(I^d)\right) - f_{lang}(l^d) \right\|_2^2 \tag{4.1}$$

This effectively makes the pretrained image encoder reflect the LLM embedding space.

4.4.1.2 Language-Distance Learning

We also experiment with a second variant of image-language pretraining that provides a softer form of language supervision. We posit that the pairwise distances between corresponding two language embeddings are what convey semantic meaning, not the exact values of the language embeddings themselves. Thus, we design an objective to regress the image embedding distances between a pair of images from the two domains to their corresponding language distance:

$$\mathcal{L}_{cnn,dist}(\mathcal{D}_{VL}) = \left\| f_{cnn}^{\top}(I^s) f_{cnn}(I^t) - d\big(l^s, l^t\big) \right\|_2^2 \tag{4.2}$$

where the language distance function we use, $d : l \times l \to \mathbb{R}$ is BLEURT (Sellam et al., 2020), a learned string similarity score commonly used in NLP. We found BLEURT provided a richer signal than dot products or ℓ-2 distances.

4.4.2 Multitask, Multidomain Behavioral Cloning

Our second step in Lang4Sim2Real involves learning a multi-domain, multi-task, language-conditioned BC policy (see Fig. 4.2(ii)) conditioned on our learned

domain-invariant visual representation from Section 4.4.1. During this phase of policy learning, we freeze all but the last layer of f_{cnn}, insert trainable FiLM adapter modules (Perez et al., 2018) to process the language instruction embeddings, and a fully-connected policy head to process the image feature, $f_{cnn}(I_t)$, and proprioceptive state, s_t. We train the resulting policy π with BC loss to predict the mean and standard deviation of a multivariate Gaussian action distribution. The policy is trained on $k+1$ tasks: k from $\mathcal{D}^{\text{s}}$ (thousands of trajectories per task) and 1 from $\mathcal{D}^{\text{t}}_{\text{target}}$ (≤ 100 trajectories, see Sec. 4.5).

4.5 Experimental Setup

We evaluate Lang4Sim2Real on sim2real settings, where the few shot IL is defined in the real world and we use simulation to address the data scarcity. We aim to use language to bridge a wide sim2real gap with differences in control frequency, task goals, visual observation appearance, objects, and initial positions.

Evaluation Metrics. Task success rate is calculated through ten evaluation trials (with different initial object positions and orientations) for each of two seeds per task, for a total of 20 trials per table entry. For multi-step tasks, we also measure partial credit—the number of consecutive subtasks completed from the start.

Environments. For each of our tasks, we design simulation environments in Robosuite (Zhu et al., 2020; Todorov et al., 2012). In both simulation and real, we use a 7-DOF Franka Emika Panda arm and use a common action space (Khatib, 1987) consisting of the continuous xyz delta displacement and a continuous gripper closure dimension. Robot proprioception is 22-dimensional and RGB images are 128×128.

Tasks. For each task suite, we collect data from simulated domain $\mathcal{D}^{\text{s}}$ and real target domain $\mathcal{D}^{\text{t}}$. All demonstrations in sim and real are collected with a scripted policy. Sim trajectories range from 200-320 timesteps long, at 50 Hz, while real trajectories run at 2 Hz and range from 18-45 timesteps. Our three task suites are simple stacking, multi-step long-horizon pick and place, and wrapping deformable,

hard-to-simulate wires around a central object. See the visual sim2real gap in Fig. 4.3.

In the first and third tasks, we collect and train on 400 demonstrations per task (1600 total) as our $\mathcal{D}^{s}$ simulation data, while we have 1400 demonstrations per task (5600 total) for the second task. We train and evaluate with 25, 50, or 100 $\mathcal{D}^{t}_{\text{target}}$ demonstrations.

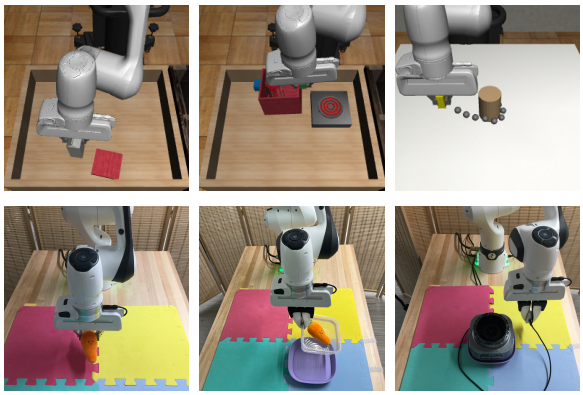

Figure 4.3: Top row: Simulation; bottom row: real-world. Columns from Left to Right: Stack Object, Multi-step Pick and Place, and Wrap Wire tasks.

For the second task, the robot must first put an object into a container, and then put that container onto another container. The objects and containers are different in sim and real. The third task is to wrap a wire around the blender by at least 5/6ths of a full revolution. In simulation, we approximate the wire with a chain of spheres connected by free joints.

Table 4.1: `sim2real`: Performance by number of real world trajectories

Method	Action-labeled Data		Stack Object			Multi-step Pick and Place						Wrap Wire		
	Sim	Real	Success Rate (%)			Success Rate (%)			Subtasks Completed			Success Rate (%)		
	$\mathcal{D}^{s}$	$\mathcal{D}^{t}_{\text{target}}$	25	50	100	25	50	100	25	50	100	25	50	100
No Pretrain ($\mathcal{D}^{t}$)	–	✓	20	30	45	0	30	35	0.45	1.05	1.05	20	15	45
No Pretrain ($\mathcal{D}^{s}+\mathcal{D}^{t}$)	✓	✓	35	20	55	45	25	55	1.15	1.0	1.4	25	20	20
MMD	✓	✓	25	35	**80**	20	10	35	0.8	0.9	1.1	5	10	20
Domain Random.	✓	✓	40	60	40	10	10	25	0.7	0.6	0.7	0	0	0
ADR+RNA	✓	✓	35	30	35	15	25	40	0.85	0.8	1.3	0	10	0
Lang Reg. (ours)	✓	✓	40	**75**	**80**	**60**	**80**	**90**	**1.45**	**1.8**	**1.9**	**45**	**40**	45
Lang Dist. (ours)	✓	✓	**60**	45	**80**	55	70	75	1.35	1.65	1.6	30	25	**75**
Stage Classif.	✓	✓	40	60	60	50	60	50	**1.45**	1.55	1.5	30	**40**	50
CLIP (frozen)	✓	✓	25	5	15	10	15	40	0.3	0.45	1.0	35	35	30
R3M (frozen)	✓	✓	30	45	65	15	60	55	0.7	1.4	1.5	5	25	25

Baselines. To evaluate the effectiveness of Lang4Sim2Real, we consider two sets of baselines: non-pretrained baselines where the CNN is initialized from scratch, and baselines with pre-trained visual encoders. For the non-pretrained baselines, we examine training with only $\mathcal{D}^{t}$ data, and training with both $\mathcal{D}^{s}$ and $\mathcal{D}^{t}$ data. This

enables us to understand the benefits of our proposed training procedure. We also compare to popular prior sim2real baselines: **MMD** (Tzeng et al., 2014), **Domain randomization** (Tobin et al., 2017) of the colors, textures, and physics of the $\mathcal{D}^s$ environment, and Automatic Domain Randomization with Random Network Adversary (**ADR+RNA**) (OpenAI et al., 2019). For the pretrained baselines, we consider two strong foundation models as the visual backbone, **CLIP** (Radford et al., 2021) and **R3M** (Nair et al., 2022).

4.6 Experimental Results

What is the impact of our pretraining approach? Lang4Sim2Real nearly doubles the success rate of both non-pretrained baselines (first row-group in Table 4.1) in most task suites, demonstrating the importance of our visual pretraining procedure versus simply training a policy on all the data at once.

How do our two image-language pretraining variants compare? We compare our two pretraining variants introduced in Sections 4.4.1.1 and 4.4.1.2. Language regression performs better on average.

What is the effect of language in learning shared representations? We ablate the effect of language during pretraining as the "stage classification" row in Table 4.1, where the pretraining task is to predict the stage index of an image instead the language embedding or embedding distance. Language provides a measurable benefit in all task suites, especially in multi-step pick-and-place, perhaps because Lang4Sim2Real leverages similarities in language descriptions between the first and second steps of the pick-and-place task.

How does our method compare to prior works in sim2real and vision-language pretraining? Lang4Sim2Real outperforms all of the prior sim2real baselines we tested against (second row-group in Table 4.1), which collectively do relatively poorly in most settings, highlighting the difficulty of the sim2real problem in our setup. Our method outperforms both vision-language, internet-scale pretrained

baselines (fourth row-group) across the board. When trained on increasing amounts of real-world data, both R3M and CLIP tend to plateau at 65% and 40% respectively, while our method achieves up to 90%. This suggests that CLIP and R3M do not scale as well as our method, despite being pretrained on internet-scale data while our method was pretrained on images from just a few hundred sim and real trajectories. These results, especially on the wire wrap task, show that Lang4Sim2Real is able to bridge wide sim2real gaps even with deformable objects.

Chapter 5: Mixed-Initiative Dialog for Human-Robot Collaborative Manipulation

5.1 Introduction

In Chapters 3 and 4, we saw how language, as a store of semantic meaning, enables a surprising amount of few-shot generalization to new tasks *and* new domains. In this chapter (based on work currently under review (Yu et al., 2025)), we pivot to our second line of work mentioned in Section 1.5 to explore how language can enable flexible communication paradigms for human-robot collaboration. Why is collaboration important, and how can effective communication expand physical robotic capabilities?

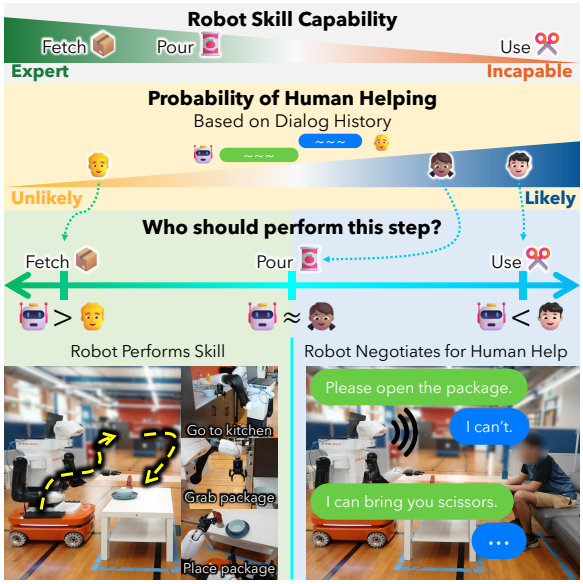

Figure 5.1: We present MICoBot, a system for human-robot collaboration where both agents can initiate and carry out physical and verbal actions. MICoBot uses both the robot's capability and the likelihood of human helping (inferred from previous dialog history) to determine whether the robot is better suited than the human to perform the skill. If it is, it attempts the skill itself. If not, it negotiates for human help.

Imagine preparing for a dinner party with a friend. Your friend might excel at mixing drinks while you focus on cooking the main dish. You are also better at decorating, while both of you reluctantly negotiate over less desirable tasks like cleaning.

Now, imagine a helper robot in place of the friend. Current robots are not fully autonomous for many household tasks, but they offer broad capabilities with varying levels of reliability that can be leveraged through collaboration with humans.

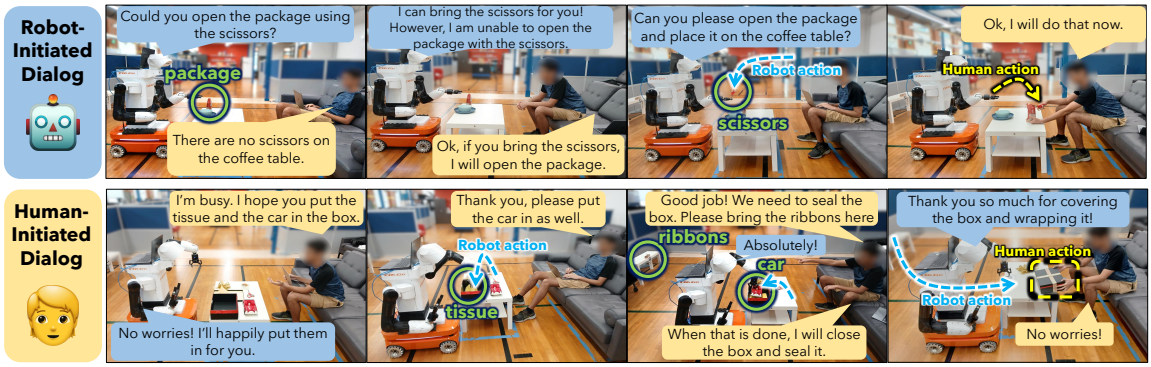

Figure 5.2: MICoBot supports *both* robot-initiated (top row) *and* human-initiated (bottom row) task-directed speech2speech dialog, where both agents discuss who is best suited to perform steps in a long-horizon task.

To be an effective partner, such a robot must communicate in physically grounded natural language, decide when to take initiative or defer to the human, negotiate task allocation based on strengths and preferences, and adapt to changing contexts. These ingredients are essential not only for collaborative household robots, but also for coding assistants, chatbots, and AI agents.

Long-horizon tasks, such as preparing for a party, require dynamic, bidirectional collaboration across control, initiative, and communication. In particular, the ability to both take initiative and yield control is central to effective human–AI teamwork. However, current AI systems (e.g., chatbots) typically rely on one-directional, human-initiated interactions (Ouyang et al., 2022; Achiam et al., 2023), while prior human–robot interaction (HRI) approaches often assume fixed collaboration plans and full human compliance (Selvaggio et al., 2021). Such assumptions limit flexibility and fail to account for the diverse preferences, capabilities, and strengths of different human partners. We argue that effective human–robot collaboration requires a paradigm shift toward mixed-initiative dialog as the communicative medium, enabling both agents to initiate, negotiate, and respond to proposals in natural language.

To enable this paradigm shift, we introduce MICoBot (Mixed-Initiative Collaborative roBot), the first system that supports mixed-initiative dialog for seamless human–robot collaboration in the physical world. MICoBot allocates task steps to the

most suitable agent (see Fig. 5.1) to maximize overall success, minimize human effort, and respect human-initiated requests. It achieves this by engaging in mixed-initiative dialog and negotiation to decide step allocation (see Fig. 5.2), while coordinating the physical and verbal actions required to execute the plan.

We validate MICoBot through real-world user studies, where 18 participants collaborated with a TIAGo mobile manipulator on three household tasks. Our approach improves success rate by **50%** over a pure LLM baseline and is preferred by over **75%** of participants. In summary, our contributions are: (1) **A new problem setting** that integrates mixed-initiative natural language dialog with mixed-initiative human–robot interaction. (2) **A novel optimization framework** for task allocation that balances human and robot effort with success through a unified metric. (3) **A hierarchical robotic system, MICoBot**, that enables mixed-initiative speech-to-speech human–robot collaboration and flexibly adapts to diverse real human collaborators in physically grounded, long-horizon tasks.

5.2 Related Work

Mixed-initiative dialog (Carbonell, 1970; Allen et al., 1999; Chu-Carroll, 2000) refers to communication with freeflowing questions and answers from both parties. In the NLP field, the dominant chatbot paradigm adopted by large language models (LLMs) largely eschews mixed-initiative interaction: humans pose substantive questions, and the chatbot responds to these requests (Ouyang et al., 2022; Achiam et al., 2023). Recent work has sought to make dialog systems more goal-directed and persuasive toward some goal, such as soliciting donations (Wu et al., 2025; Deng et al., 2023a; Yu et al., 2023b; Chen et al., 2023; Deng et al., 2024) or clarifying ambiguous human requests (Qian et al., 2021; Deng et al., 2023b; Chen et al., 2024). However, none of these systems addressed mixed-initiative dialog in grounded, real-world collaborative manipulation tasks.

In human-robot interaction (HRI), researchers have developed **human-robot**

collaboration systems that interact through language but are restricted to **single-initiative dialog** (see Section 2.6). Some works in HRI have explored **mixed-initiative collaborative systems without dialog**, only with physical actions (Few et al., 2006; Natarajan et al., 2024; Bishop et al., 2020; Rosero et al., 2021; Paleja et al., 2024; Jiang and Arkin, 2015; Baraglia et al., 2016). These prior works overlook the critical role of communication in effective collaboration.

Prior works have also studied **human-robot optimal task allocation** by maximizing productive time and minimizing idle agents (Vats et al., 2022; Yu et al., 2021a) or maintaining safety (Faccio et al., 2024; Singh et al., 2023). These solutions assume availability of all agents. In contrast, MICoBot can adapt to the specific human's willingness to help by estimating its availability based on previous dialog.

5.3 Problem Setting

MDP Formulation. We study how human-robot collaborative manipulation can be facilitated through mixed-initiative dialog. We assume that both agents can observe the state of the world, $s \in \mathcal{S}$, and perform actions, $a \in \mathcal{A} = \mathcal{A}_p \cup \mathcal{A}_v$, comprised of a physical action space, $\mathcal{A}_p$ (e.g., move objects, open them, etc.), that directly affect the physical state of the environment s, and a free-form, natural language verbal action space, $\mathcal{A}_v$, which is directly observed by the other agent but does not change the physical state. We model the problem as a Markov Decision Process (MDP) from the robot's point of view (see Fig. 5.3), where on each environment step, the robot performs some action, $a_R \in \mathcal{A}_{p,R} \cup \mathcal{A}_{v,R}$ and receives an observation $o = [I, a_{v,H}, s_{proprio}]$ consisting of an RGB-D image I, an optional verbal action from the human partner $a_{v,H}$, and the robot's proprioceptive state $s_{proprio}$. Within each environment step, the human may

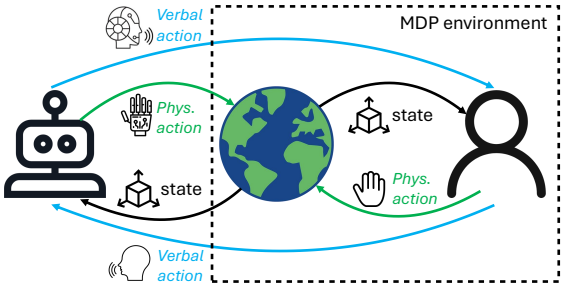

Figure 5.3: Proposed MDP for Mixed-Initiative Collaboration.

perform a series of actions, $a_H \in \mathcal{A}_{p,H} \cup \mathcal{A}_{v,H}$, in its own physical and verbal action space after perceiving the world state and robot's previous dialog, $a_{v,R}$.

Physical and Verbal Action Spaces. The physical and verbal action spaces, $\mathcal{A}_p$ and $\mathcal{A}_v$, are shared between both agents. Each element of these action spaces is a parameterized action primitive represented by the pair, $a_{p/v} = (\omega_{p/v}, \theta_{p/v})$. ω_p is the type of the physical action primitive (`open`, `pick-and-place`, etc.) and θ_p are the corresponding parameters (e.g., what object to open or pick and where to place it). We assume that humans are fully competent in executing all steps of a collaborative household manipulation task but may be unwilling or unavailable to perform some or all required actions. Their behavior can range from indifferent (never acting) to overly proactive (completing the entire task without robot involvement).

In contrast, robots often have limited manipulation capabilities and may be unable to execute more complex actions, in which case it uses verbal actions to communicate with the human. ω_v is the type of the verbal action primitive (`ask_human_for_help`, `respond_to_human`, etc.), and θ_v are the corresponding parameters defining the context of the verbal primitive (e.g., what step the robot needs help on). While the types of verbal actions are limited, each generates freeform and open-vocabulary language. MICoBot first selects an abstract verbal action from this space, then translates it into a natural language utterance to negotiate with the human—conveying its requests and the assistance it requires for successful collaboration. This involves reasoning over asymmetric human and robot physical capabilities to devise collaboration strategies that maximize task success while minimizing human effort.

Collaborative Task Definition and Problem Statement. We assume the collaborative task is defined by a task plan of length T, represented as a sequence of unassigned **physical** action primitives, $[a_{p,0}, ..., a_{p,T-1}]$, such as [(`pick-and-place`(`box`, `table`), ..., `close`(`box`)], obtained from the task instructions or off-the-shelf task planner. To complete the manipulation task while minimizing human effort, the system must allocate steps of the plan between the two agents—negotiating with the

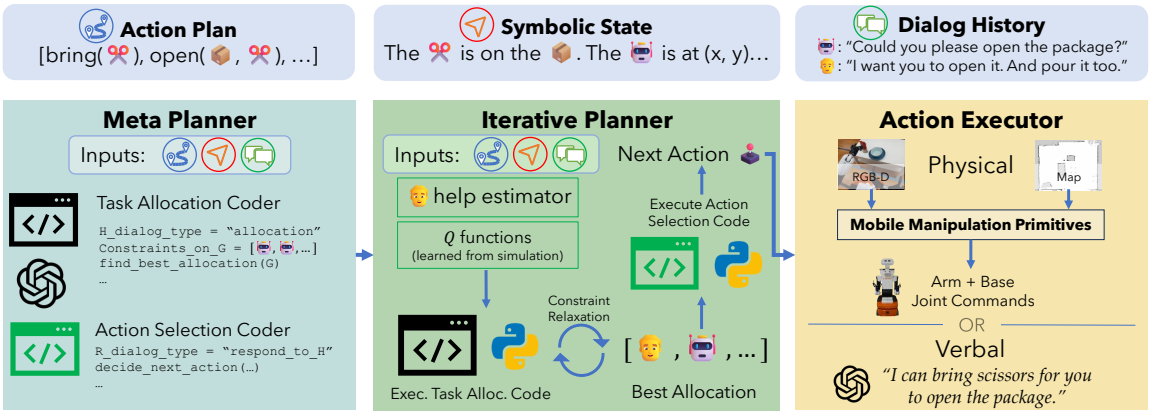

Figure 5.4: MICoBot consists of 3 decision-making modules: a meta-planner that produces a collaborative strategy expressed through adaptive planning code, an iterative planner that executes the code and optimizes our objective (Eq. 5.1) to decide the next primitive action, and an action executor that outputs the low-level pose trajectory or verbal utterance to say to the human.

human through robot-initiated dialog to suggest assignments, adapting to human preferences through human-initiated dialog, and ultimately executing its assigned physical actions. At each step t, the system must compute the best allocation of the remaining steps of the plan, $G = [g_t, ..., g_{T-1}]$, where $\forall t, g_t \in \{H, R\}$. The optimal allocation G^* maximizes the expected task success probability while minimizing total human effort. The optimization also incorporates constraints conveyed through the mixed-initiative dialog history, such as task allocation requests or proposed task splits. The resulting allocation G^* determines whether the robot executes the current step (R) or negotiates for human help (H).

5.4 MICoBot: Mixed-Initiative Collaborative Robot

5.4.1 Collaborative Task Allocation as Optimization.

A helpful physical collaborator must aim for task success with minimal human effort while adhering to human preferences expressed in dialog. We formulate collaborative task allocation as a constrained optimization problem. To simplify, we combine success probability and effort into a single Q-value inspired by temporal distances in

RL (Myers et al., 2025).

We assume each task step is executed by a multi-task policy π that performs continuous low-level control at a fixed control frequency. We define the reward as $r = -1$ per control time step, terminating when the skill completes or times out. A well-trained Q-function, $Q : o_t \times a_t = (\omega_t, \theta_t) \mapsto \mathbb{R}$ with a discount factor of 1, thus represents the **negative expected number of timesteps** until skill completion from a given state. We assign each agent a distinct Q-function: Q_R for the robot and Q_H for the human. These agent-specific Q-functions thus provide a unified, interpretable cost metric for comparing step allocations, jointly capturing both execution time (effort) and likelihood of success.

However, directly optimizing step allocation with Q_H and Q_R diverges from realistic human-robot collaboration in three ways: (1) human and robot effort are valued equally, when human effort is more valuable; (2) the human is assumed to always comply with robot-initiated requests, overlooking their willingness and availability; and (3) human-initiated requests or preferences are not respected. To address (1), we introduce a *human-effort ratio*, α, valuing human effort to robot effort. To address (2), human Q-values are adjusted with an inferred probability, $p_{H,t}$, of the human agreeing to perform action $a_{H,t} = \omega_t(\theta_t)$ when asked. To address (3), we enforce constraints, $C_1, \ldots, C_n$, extracted from human-initiated dialog—such as explicit requests to perform specific steps themselves or delegate them to the robot. Altogether, we propose the following objective to find the optimal task allocation G^*:

$$\begin{aligned} \max_{g_t,\ldots,g_T} \quad & \sum_{t}^{T-1} \left(\mathbb{1}_{g_t=H} \cdot \frac{\alpha}{p_{H,t}} + \mathbb{1}_{g_t=R} \right) Q_{g_t}(s_t, a_t), \\ s.t. \quad & C_1, \ldots, C_n \text{ are satisfied} \end{aligned} \tag{5.1}$$

that minimizes expected time-to-success and human effort.

	Pour Package in Bowl $n = 6$		**Assemble Toy Car** $n = 6$		**Pack Gift Box** $n = 6$		**Average** $n = 18$	
	MICoBot	LLM	MICoBot	LLM	MICoBot	LLM	MICoBot (ours)	LLM
Entire Task Success Rate (%, ↑)	**100**	83	**67**	0	**67**	0	$\mathbf{77.8 \pm 15.7}$	27.8 ± 39.3
% of task steps completed (↑)	**100**	93	**94**	31	**88**	50	$\mathbf{93.8 \pm 5.1}$	58.2 ± 26.0
% of steps performed by Human	27	29	60	5	35	21	40.5 ± 14.2	18.2 ± 9.7

Table 5.1: Comparison between MICoBot (ours) and the LLM baseline across three real-world tasks.

5.4.2 MICoBot Framework

MICoBot is a three-level framework (Fig. 5.4) that includes 1) a meta-planner (implemented as GPT-4o) that processes human dialog and generates a collaborative strategy expressed in code, 2) an iterative planner that updates planning state variables and allocates and decides the next action to perform by executing the code, and 3) an action executor that carries out the action primitive, either through low-level physical actions or by formulating a dialog utterance to the human.

Q Functions. To quantify Eq. 5.1, MICoBot requires accurate Q-functions that capture each agent's expected effort and success probability on each task step. To collect data to learn the robot's Q-function (Q_R), we use the OmniGibson simulator (Li et al., 2022), recording both completion times and success rate. We train a supervised network as Q_R that predicts the expected timesteps for an action primitive a to succeed from a given symbolic state o. When estimating the human's Q-function (Q_H), we simply obtain time estimates for each step from an LLM predicting how long a human needs to execute action $a_t = \omega_t(\theta_t)$, plus a travel time estimate based on human-object distances.

Human Helpfulness Estimator. To adapt to temporally changing human sentiment, MICoBot estimates the probability of human assistance at the current t-th timestep, $p_{H,t}$, using an LLM-based sentiment analysis over prior dialog.

5.5 Experimental Evaluation

We evaluate MICoBot in the real-world on a Tiago mobile manipulator working with 18 unique human participants on household tasks. A successful robotic collaborator must achieve task success (primary metric) while minimizing human effort (secondary metric). We also report subjective user satisfaction measures.

Environment. In the real-world, we perform our experiments in a mock apartment with a kitchen and living room area with commonplace furniture. In all of our tasks, the robot and human work together on opposite sides of a coffee table. Simulating a household setting, the participant mainly sits on the couch, doing their personal (i.e., non-task-related) work. The human can be as inactive or proactive as they wish. Each human user study consists of two 20-30 minute trials, in which they collaborate with both our method and a pure LLM baseline, ordered randomly.

Skills. To perform long-horizon household tasks, the robot has access to several mobile-manipulation action primitives relating to pick-and-place, pouring, and folding. To initiate and respond within mixed-initiative dialog, the robot uses open-vocabulary verbal action primitives to ask the human for help on a step, propose to split up a few steps with the human, accept/reject human requests based on its capability, and respond to the human for all other queries.

Baselines. Because multiple components of our method are powered by LLMs, we compare our approach to a pure LLM baseline (**LLM**) given the same information as our meta-planner: symbolic state, dialog history, task plan, α human-robot effort tradeoff ratio, and a list of the robot's skills. The LLM primarily optimizes for task success and secondarily minimizes human effort.

Tasks. We perform user studies on 3 real-world tasks, each with 6 users for a total of 18 unique participants. **(1) Pour package into bowl**: Fetch the bowl, package, and scissors, cut open the package, and pour it into the bowl. **(2) Assemble toy car**: bring the car parts, wheels, and drill from the shelf to the coffee table, drill in the wheels, switch the drill bit, and finally drill in the windows and seats. **(3)**

Pack gift box: fold the gift box, put tissue wrapping paper and a toy car in the box, close the lid, wrap ribbons, and tape down a gift bow. Each task is 5-8 mobile manipulation steps long and requires varying degrees of human involvement.

5.6 Experimental Analysis

(1) Does our method achieve the best trade-off between task success and minimizing human effort? In our real-world user study (Table 5.1), MICoBot achieves a 78% task success rate compared to 28% for the LLM baseline (statistically significant with p-value 0.007 under Fisher's exact test). Additionally, MICoBot achieves a 94% task step completion rate compared to the baseline's 58% (statistically significant with p-value 0.002 under the Wilcoxon-signed-rank test). MICoBot understood its own limitations (through affordance functions trained in simulation), and was hence better at leveraging human assistance effectively on the steps it was ill-suited to perform. The LLM baseline tended to prioritize minimizing human effort over task completion by allocating the robot multiple steps it was incapable of, since the LLM lacked an understanding of the robot's affordances. MICoBot uses roughly double the amount of human effort (41% vs 18%) to achieve nearly triple the success rate of the LLM baseline, demonstrating a better trade-off between maximizing task success and minimizing human effort.

(2) How do users feel about working with our system? An A/B blind preference test shows that 78% of users preferred our method over the LLM baseline. Our method also significantly outperformed the baseline in user scores on overall satisfaction, communicative ability, and capability in asking for a suitable amount of help (statistically significant under the Wilcoxon-signed-rank test with p-values ranging from 0.007 to 0.024; see Figure 5.5). In contrast, the LLM baseline often failed to ask when it needed help and was unwilling to reject human requests it could not fulfill, leading to over-promises and task failures.

(3) Is mixed-initiative dialog critical to our method's performance?

In real-world user studies, MICoBot engaged in 2.4 dialog initiative shifts per trial, compared to the LLM baseline's 1.1. This enabled MICoBot to boost human acceptance of help requests from 55% to 86%. The LLM baseline made far fewer help requests per trial (0.9 vs. MICoBot's 2.9) and achieved a smaller acceptance increase (70% to 75%). This demonstrates mixed-initiative dialog is critical to collaborative discussion and task success.

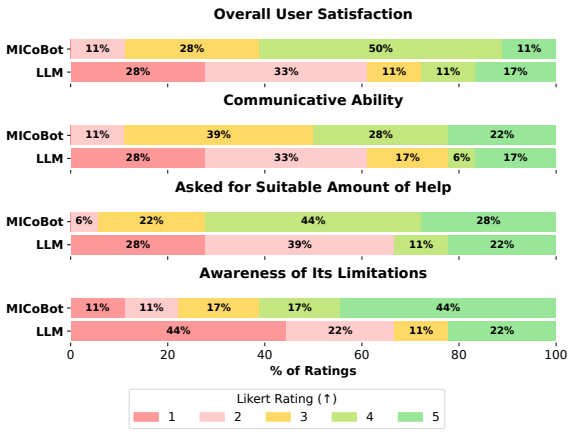

Figure 5.5: Our method substantially outperforms the LLM baseline in user ratings averaged over all $n = 18$ participants.

Chapter 6: Proposed Future Work

We discussed in Sections 3 and 4 ways to use language for semantic understanding in robotics tasks, enabling a greater degree of few-shot generalization to new tasks and domains. In Section 5, we discussed an entirely separate way to use language—for freeform, flexible, bidirectional communication that enables robots to accomplish long-horizon tasks with human collaboration. In our future work, we seek to expand upon and combine these two separate lines of work.

6.1 Steering Policies and Accelerating Learning with Language Rules and Guidance

When we bring helper robots into our homes, it is crucial for the robot to operate under a set of rules that we specify—for example, to avoid running over family members, avoid manipulating glass and other brittle objects, and generally try to conserve water and electricity when cleaning and using appliances. It would be very helpful if we could enumerate these desired behaviors in natural language, hand over this manual to the robot, and train the robot to follow this manual.

As a problem setting, let G represent this manual (equivalently, "guidebook" or "rulebook"). G contains natural language guidelines that specify the robot's objectives, prohibited behaviors, and information about the expected dynamics of the environment. We hope to leverage these guidelines to enable robots to learn faster, behave more safely, and improve human satisfaction. The guidelines in G range in importance from critical (e.g., "never cause physical harm to the human") to recommendations of less importance (e.g., "try to move smoothly from one place to another"), to purely informational guidelines (e.g., "the toaster becomes hot when you turn the dial past 0").

Learning from rules offers to bridge the two uses of language that we explored

in Sections 3, 4, and 5. Along the first thread of leveraging language's pretrained embedding space, rules offers a compact, distilled representation of good and bad behaviors, language knowledge, and know-how that can be leveraged to help the robot learn faster. Along the second thread of using language as a communicative medium between humans and robots, a rules manual is an extensive document communicating detailed intent, preferences, directives, and knowledge to the robot.

Enabling robots to follow a complex system of ranked rules requires us to develop a cost function to quantify the compliance of robot behavior. This makes RL a natural learning paradigm to teach the robot good behaviors. In RL however, exploration is difficult because policies randomly explore regions around the current behavioral policy to discover optimal trajectories for achieving a task, which can be computationally intractable for long-horizon tasks and continuous state and action spaces.

To address the problem of exploration inefficiency and guideline compliance, we propose factorizing the policy into two levels: a high-level policy $\pi_{hi} : o_t \times l_{task} \mapsto l_{act}$ that predicts an action expressed in language tokens every k timesteps, where $l_{act} \in \mathbb{R}^{n \times d_l}$, and a low-level policy $\pi_{lo} : o_t \times l_{task} \times l_{act} \mapsto a_t$, which predicts a low-level robot action (i.e., xyz delta positions) $a_t \in \mathbb{R}^{d_a}$ every timestep.

The goal of π_{lo} is to predict a series of low-level actions a_t that follow the intermediate-level action l_{act} expressed in language tokens. The job of π_{hi} is thus to predict l_{act} that induces π_{lo} to make progress toward the goal expressed in l_{task}. Having an intermediate action representation l_{act} enables (1) better interpretability, allowing us to judge whether the policy is following guidelines, and (2) enables more semantically diverse exploration, since perturbations in l_{act} token space leads to more meaningful behavioral policy differences than perturbations in low-level action space.

6.1.1 Prior work

6.1.1.1 Constitutional AI

Before several LLMs were popularly released in late 2022, researchers worked to prevent them from emitting hateful, discriminatory, or violent content. One of these methods was Constitutional AI (Bai et al., 2022), where researchers developing Claude wrote a list of 16 principles, referred to as a *constitution*. To encourage their LLM to adhere to this constitution and produce safe responses, they iteratively prompted the LLM to (1) provide a response r to an original prompt p, (2) determine whether r had following a randomly sampled principle from the constitution, and (3) if not, self-suggest modified responses r' to the original prompt p to follow that principle. This yielded LLM responses that were more likely to comply with the constitution, and these prompt and modified response pairs (p, r') were then used to finetune the model.

In a sense, our proposal can be summarized as "Constitutional AI for Robotics," with one key difference. In constitutional AI, all principles are listed as equally important. However, in the physical world, some principles (such as minimizing harm to the human) are much more important than others (such as minimizing damage to objects on the scene). This means that rules within rulebooks should be ordered by priority and importance (Censi et al., 2019).

6.1.1.2 Language as Intermediate Action Representation

Our proposed high-level policy architecture is inspired from RT-H (Belkhale et al., 2024) and YAY-Robot (Shi et al., 2024), both of which use a high-level policy that passes a language action for the low-level policy to execute. This intermediate language action is a convenient interface for a human to provide on-the-fly verbal feedback to correct the robot's behavior in real time. However, these works only perform imitation learning and do not consider the main problem of following robot rules or using the rules and knowledge in a guidebook to help accelerate learning and

RL exploration.

6.1.1.3 Symbolic Specifications

When a robot needs to learn to follow rules and guidelines, a key question is how these rules can be represented in a manner compatible with the policy architecture and learning algorithm. Much work has been done on task specification in symbols. A popular representation is Linear Temporal Logic (LTL), which, as its name suggests, can represent multi-step task specifications that can change with time, such as "Go to three nearby restaurants and survey the prices, then buy the cheapest entree above 800 calories that you saw and bring it to me." Researchers have used LTL for task specification and generalization to new tasks (Liu et al., 2023b,a, 2024b,a; Hsiung et al., 2021). Hu et al. (2023) proposed a spinoff on LTL that is more compact and code-like, RoboEval Temporal Logic (RTL). However, these works focus on task specification through symbolic and logical expressions and do not study how a guidebook of multiple logical expressions can shape robot behavior during learning.

6.1.1.4 Constrained RL

The standard objective in RL is to maximize the expected sum of discounted rewards (Equation 2.3). However, a policy that aims to maximize expected reward does not guarantee safe behavior. Constrained Policy Optimization (CPO) (Achiam et al., 2017) proposes modifying the standard RL objective by adding cost functions, the analog of reward functions but indicating how undesirable a state-action pair is. CPO can then learn value functions that are based on the discounted sum of these cost functions. These cost-value functions must be maintained under some constant, predetermined threshold over the entire trajectory. The RL agent must then maximize reward while visiting states that satisfy the cost-value function threshold constraints.

A separate concept called shielding was also proposed to encourage safe RL (Alshiekh et al., 2017). There are two shielding paradigms: a shield that precedes the

policy whose responsibility is to provide a list of safe actions for the policy to choose from, or a shield that follows the policy whose responsibility is to remediate unsafe actions from the policy to make them safe to execute in the real world.

More relevant to our proposed work is training RL policies with explicit natural language constraints. Yang et al. (2020b) learns a 2D binary mask, of the same dimension as the state space, representing the states that the agent cannot visit. Wang et al. (2024b); Lou et al. (2024) adopt CPO by comparing the cosine similarity of language embeddings of language constraints and textual observations. However, these works deal with simplified gridworld domains where a mask over the entire state space is feasible to predict at each timestep, or in simple 2D pointmass domains, and they only deal with negative rules pertaining to which states the agent should not move to. We hope to learn from a much wider range of guidelines—not just negative constraints on which states shouldn't be visited, but also guidelines on what behaviors are recommended, and dynamics information about the environment. We are also interested in enabling faster RL exploration when following this guidebook, which these prior works were unable to demonstrate.

6.1.1.5 Formal Methods

The field of formal methods (Woodcock et al., 2009) provides tools for analyzing and ensuring guarantees of each component of a program. These principles could be applied to our setting of rule following robots, especially if the policy has an intermediate planning stage that was written in code or symbolic form.

6.1.2 Types of Guidelines

What are the types of guidelines in a guidebook we provide our robot? We propose some broad categories we want our robot to comply with and make use of during the learning process.

First, we define three guidelines types: constant, conditional, and dynamics.

A guideline is constant if it always applies under all situations (e.g., "Never collide with the furniture", or "Prioritize choosing shorter paths"). A guideline is conditional if it applies under a specific state condition (e.g., "If the lights are off, turn them on"). Finally, a dynamics guideline is one that describes a certain aspect of how the system evolves over time, such as "if the cabinet is opened, objects can be placed inside."

Second, we split the constant and conditional guidelines by their importance: critical and non-critical. Critical guidelines must be followed and complied with at all times, while non-critical guidelines can be ignored in service of obeying critical guidelines. The notion of prioritization over rules is common in unstructured environments. For instance, we may want a self-driving car to avoid abrupt lane changes to ensure a smooth ride—a non-critical guideline. However, if there were a sudden obstacle blocking its lane, we would prefer the car swerve out of its lane temporarily to avoid it—a critical guideline (Censi et al., 2019).

6.1.3 Proposed Method

6.1.3.1 Measuring Rule Following

How would we ensure that the policy follows the rules and guidelines stipulated in the guidebook G? We can leverage the fact that the high-level action l_{act} produced by the high-level policy π_{hi} is a token-by-token prediction of what the robot intends to do—making it of the same modality as the guidebook G written in natural language.

Recall that π_{hi} predicts a probability distribution over language tokens, so we can sample it to get candidate string predictions for l_{act}. We want to shape these probability distributions to achieve better compliance with rules most relevant to the current state s_t. We would like to place more probability mass on performing a language action l_{act} that complies with guidelines and decrease the probability mass on performing language actions that do not. To do this, we propose modifying the predicted distribution of l_{act} via a compliance score which evaluates how likely the proposed l_{act} complies with all relevant rules in G at the current state s_t.

Let $w(l_i)$ = compliance score of $l_{act,i}$, one of the candidate l_{act} predictions of π_{hi}. Say we have access to a rule retriever $F : G \times s_t \mapsto R$, which gives a subset $R = \{r_j\}_{j=0}^{k-1}$ of the top-k most relevant guidelines G in the current state s_t. F can be implemented as a RAG-based system (Lewis et al., 2020). We define the compliance score as follows:

$$w(l_i) = \sum_{r_j \in R} P(\text{sat}(r_j)|l_i, s_t)\sigma\big(P(\text{rel}(R)|s_t)\big)_j \tag{6.1}$$

where $P(\text{sat}(r_j)|l_i, s_t)$ is the probability that rule r_j is satisfied if action l_i is executed by π_{lo} from state s_t, and $P(\text{rel}(R)|s_t)$ is a vector where element j is the probability that rule r_j is relevant given the current state s_t. The $\sigma(\cdot)_j$ indicates the jth element of the vector after the softmax operation.

The relevance scores are important because rules often vary in importance as a function of the current state s_t. For instance, the rule "all dirty plates should go in the dishwasher," is most relevant after the robot has grasped a dirty plate. Other rules like "do not move close to the wall" are more relevant the closer the robot is to the wall.

Thus Equation 6.1 measures how well the proposed language action l_i satisfies the rules in R, weighted by the relative relevance of the rules each other. Intuitively, under this definition, satisfying a more relevant rule contributes more heavily to the compliance score than satisfying a less relevant rule. $w(l_i) \in [0, 1]$, where $w(l_i) = 1$ indicates perfect compliance to all top-k retrieved rules in R.

Finally, we can update the predicted logits from $\pi_{hi}(l_i|s_t)$ by pushing them toward the softmax distribution induced by $w(l_i)$. This moves more probability mass toward sampled l_{acts} from π_{hi} that are more compliant to the relevant rules in the rulebook G.

As initial steps, we plan to use an off-the-shelf VLM for $P(\text{rel}(r_i)|s_t)$ and an LLM for $P(\text{sat}(r_i)|l_i)$, but for improved performance, we may need to consider training

these from in-domain data as binary classifiers or state-based heuristics. We also plan to initially treat all guidelines as equally important. After this works decently, as a subsequent step, we will explore extracting and inferring the relative importance of the rules from G as an additional multiplicative term in Equation 6.1. We can also leverage dynamics guidelines from G to more accurately estimate the satisfaction probability of rule r_j when following language act l_i from state s_t.

6.1.3.2 Data Collection

We seek to train the robot to perform some task expressed in natural language l_{task} while following guidebook G. We assume access to two datasets. The first, $\mathcal{D}_{task}$, consists of a small number of target task demonstrations $\tau_i = \big[(s_0, a_0), ..., (o_{T-1}, a_{T-1})\big]$. We also assume that there are language segmentations of each expert demonstration, where each expert demonstration is divided into chunks $[t_{start}, t_{end}]$ that denote the timesteps between which the robot is performing a movement with the language description l_{act}. The language segmentation is either provided by a human oracle, or a heuristic captioning method similar to Section 4.

The second dataset $\mathcal{D}_{play}$ is of the same format, except that it contains multiple tasks that are somewhat related to l_{task}. This provides diversity of paired language and action behaviors, useful for increasing the range of language inputs that π_{lo} is able to follow and language outputs π_{hi} is able to generate.

6.1.3.3 Training

In the first phase, we learn $\pi_{lo} : s_t \times l_{act} \mapsto a_t$ with imitation learning on task data $\mathcal{D}_{task}$ to get a good behavioral initialization. In the second phase, we learn $\pi_{hi} : s_t \times l_{task} \mapsto l_{act}$, also with imitation learning, but this time on both task and play data $\mathcal{D}_{task} \cup \mathcal{D}_{play}$. This enables π_{hi} to produce a broader distribution of language actions.

In the third phase, we aim to improve the overall performance of our system

beyond its imitation learning performance through RL. However, we want the exploration to be rule-guided to learn faster. In this phase, we first perform RL finetuning on π_{hi} (while π_{lo} is frozen), and then perform RL finetuning on π_{lo} (while π_{hi} is frozen).

When finetuning π_{hi}, we use a standard loss $\mathcal{L}_{RL}$ (such as an advantage-based loss in PPO (Schulman et al., 2017)), plus an auxiliary term that seeks to maximize probability weight on language actions (l_{act}) that have high rule compliance. Thus the losses of π_{hi} and π_{lo} are:

$$\mathcal{L}_{hi} = \mathcal{L}_{RL} - \lambda\big(\log \pi_{hi}(l_{act}|s_t, l_{task}) + \log w(l_{act})\big) \tag{6.2}$$

$$\mathcal{L}_{lo} = \mathcal{L}_{RL} \tag{6.3}$$

The second term in $\mathcal{L}_{hi}$ allows us to maximize the probability weight on l_{act} that comply with rules with a high compliance score, $w(l_{act})$. This term is weighed by a scalar hyperparameter λ. Training the two policies with RL in stages allows us to first achieve large scale semantically meaningful exploration (e.g., "move toward the apple" vs "close gripper around cup handle"), and then achieve fine-grained exploration (i.e., within a gaussian ball around the behavior policy's actions).

6.2 Proposed Experiments and Metrics

We propose to first develop and experimentally validate our proposed method in minibehavior (Jin et al., 2023), a gridworld environment that supports simple discrete symbols for articulated objects like furniture. We may also consider other safety-focused benchmarks (Achiam and Amodei, 2019; Yang et al., 2020b; Zhou et al., 2024). We then hope to experiment within a proper physics-based robot simulator, such as robosuite Zhu et al. (2020), before finally bringing it into a real-world, real-robot setting.

The primary baselines are prior works in performing RL from natural language constraints (Yang et al., 2020b; Wang et al., 2024b; Lou et al., 2024). The primary

metric is success rate at convergence. Secondary metrics can potentially include the behavioral compliance of the trained policies to the guidebook as well as the number of environment samples needed to achieve some performance threshold. It would also be interesting to evaluate the generalization performance of the policy to changes in the rulebook G, such as when rules are added or removed.

6.3 Long-term Future Work

In this section, we propose longer-term problems to work on that extend the human-robot collaborative manipulation and bidirectional dialog setting described in Section 5. In our earlier work described in that section, we focused on a setting where both the human and robot have a shared understanding of where all the task-relevant objects are as well as the high-level steps to perform the task. We assumed the human is perfectly competent at every step of the task but not necessarily willing to help the robot. Additionally, the mixed-initiative dialog focused on ways to negotiate and propose allocating parts of the task to each agent. For more seamless human-robot collaboration in unstructured household settings, we can expand the problem setting along a number of fronts.

6.3.1 Information Asymmetry

We assumed in Section 5 that there was a capability asymmetry where the robot could only perform a subset of the steps with varying success rates, whereas the human could perform all of them perfectly competently. However, in a more realistic setting, both agents not only have different capabilities but also have different levels of knowledge about the world and different ideas for how to go about performing the task.

For instance, the human sitting at their desk may ask the robot to go to the kitchen area to find a fruit candy snack, pour some into a plate, and bring the plate to the desk. The robot goes to the pantry and finds that there are no such

fruit snacks. After some suggestions communicated remotely from the human about potential other places in the kitchen area where the fruit snack package may be, and unsuccessful attempts at locating the package at those places, the robot and human conclude that the package must have been finished already. The human then asks the robot what related snacks are available in the pantry. The robot scans over the hundreds of visible items packed in the pantry and chooses a handful of snacks most related to the human's original request.

Here, the key technical challenges are to decide (1) when the robot needs additional information from the human to succeed or be more efficient at the task, (2) what the robot needs to know, and how to formulate dialog to ask the human for this information, and (3) how much information to respond to a human's request for information (such as only providing the most relevant list of snacks instead of telling the human about every snack in the pantry).

6.3.2 Replanning from Dialog

In the preceding example, we saw how the robot needed to adjust its plan from getting a fruit snack to some other snack that the human specified later in the course of dialog. Querying and receiving new information is not the only occasion for the robot to replan in the world. Sometimes the human may change their mind midway through a robot task execution and interrupt the robot to modify the current task.

Other times, when a human refuses a critical step, the robot must replan, selecting the best new plan completion candidate to maximize partial task completion success and meet as many user expectations as possible. For instance, if the robot realizes that it is incapable of opening the fruit snack package because scissors are needed, and the human is unwilling to help the robot open the package, the robot will need to decide whether to (i) bring the scissors, plate, and unopened snack package to the human, or (ii) look for a snack package with a fruity flavor that is already opened that the robot can directly pour into a plate and bring to the human.

Often, the robot may fail at executing a skill. To recover from its own failures, replanning is also needed—it may need to ask the human for specific help on the skill it failed at, or it may need to consider self-recovery behaviors.

The technical challenges in this problem setting are (1) being able to translate the dialog history with the human into an edit of the plan, which is a sequence of skill-parameter pairs executable with the robot's library of skills, (2) proposing and ranking candidate plan completions after issues during execution (e.g., human refusal to help), based on their relevance, similarity, and feasibility compared to the original plan, and (3) deciding whether to ask the human for help after an execution failure, or how to recover from the failure itself.

6.3.3 Bi-directional Low-level Feedback

Prior work has explored humans providing natural language feedback to improve the behavior of robotic agents. However, not only do robotic collaborators need to follow human feedback, but the *human* may need to heed *feedback from the robot* so that the robot can continue helping the human.

For instance, if the human were in the kitchen with the robot and the human cuts the package with scissors, they may place the package back on the table for the robot to pick up, but in a spot too far for the robot to reach. The robot must then recognize that it cannot proceed from the current state, determine what distribution of initial states would facilitate success on its next skill, and formulate a natural language utterance asking the human to perform an adjustment to the state (e.g., "please move the package within 12 inches of my right gripper") to enable the current state to be brought within the distribution of high-success states for the robot.

We also hope to eventually tackle real-time simultaneous human-robot collaboration not just on tasks where the human is 100% competent, but also in cases where neither the robot nor human working alone can accomplish the task, and simultaneous physical collaboration is needed. For instance, moving a heavy table from one

room to another is usually a two-person job. What would it take for a human-robot team to be able to perform the same task? Both agents must simultaneously provide and follow low-level feedback to successfully coordinate as they lift the table so that it is roughly level on both sides, carefully turn and move it through narrow doorways and hallways, and delicately place it back on the floor without injuring the human.

Chapter 7: Conclusion

The purpose of robotics is to augment human physical capabilities. To enable robots to help humans beyond structured settings and to deploy them into unstructured environments like homes, robots must learn to act intelligently based on behaviors extracted from large amounts of data. However, real robot data is scarce and expensive to collect. We argue that natural language is an abundant and powerful data modality to augment robotic capabilities for two reasons: (1) language is a store of semantic meaning important for a robot to generalize to new domains and tasks, and (2) language forms the basis of human-robot communication and collaboration.

We demonstrated several promising results that leverage both benefits of language. To leverage the first benefit, we demonstrated that simultaneous language and demonstration task conditioning greatly improved sample efficiency when generalizing to new tasks, and that providing a single language instruction was as important to final performance as finetuning on 50 test task demonstrations (Section 3). We also showed that language can bridge wide sim2real gaps, including those involving deformable objects, by providing a common grounding between visually dissimilar but semantically similar images in simulation and real, boosting the performance and sample-efficiency of sim2real policies (Section 4). To leverage the second benefit, we showed that mixed-initiative dialog greatly improves human-robot collaboration on mobile manipulation tasks by enabling the robot to use freeform dialog to negotiate with the human on what steps each agent should accomplish (Section 5).

For near-term future work, we hope to bridge these two threads of work by using natural language to guide exploration and push the robot toward user-specified behaviors and guidelines. This leverages language in both of these threads: (1) pretrained language spaces determine which rules are most relevant given the current state, and whether the predicted language action complies with the guidelines, and

(2) language is used as a human-robot communicative medium, through which the human expresses its exact preferences and constraints, and the robot acts in accordance with everything the human has communicated.

For long-term future work, we propose a number of problem extensions to the mixed-initiative dialog framework that enhance human-robot communication beyond task allocation to also support bidirectionally relaying information needed for task completion, for replanning, and for low-level movement feedback. We hope that work along these fronts will ultimately enable collaborative robots that can work seamlessly and simultaneously with humans on tasks that neither agent can perform alone.

Works Cited

Josh Achiam and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning. 2019. URL https://api.semanticscholar.org/CorpusID:208283920.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Joshua Achiam, David Held, Aviv Tamar, and P. Abbeel. Constrained policy optimization. *ArXiv*, abs/1705.10528, 2017. URL https://api.semanticscholar.org/CorpusID:10647707.

Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Devi Parikh, and Dhruv Batra. Vqa: Visual question answering. *International Journal of Computer Vision*, 123:4 – 31, 2015. URL https://api.semanticscholar.org/CorpusID:3180429.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. Do as I can and not as I say: Grounding language in robotic affordances. In *arXiv preprint arXiv:2204.01691*, 2022.

Bo Ai, Zhanxin Wu, and David Hsu. Invariance is key to generalization: Examining the role of representation in sim-to-real transfer for visual navigation. *arXiv preprint arXiv:2310.15020*, 2023.

J.E. Allen, C.I. Guinn, and E. Horvtz. Mixed-initiative interaction. *IEEE Intelligent Systems and their Applications*, 14(5):14–23, 1999. doi: 10.1109/5254.796083.

Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. *ArXiv*, abs/1708.08611, 2017. URL `https://api.semanticscholar.org/CorpusID:3132647`.

Jacob Andreas, Dan Klein, and Sergey Levine. Modular multitask reinforcement learning with policy sketches. In *International Conference on Machine Learning (ICML)*, 2017.

OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34:26–38, 2017. URL `https://api.semanticscholar.org/CorpusID:4884302`.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, John Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Chris Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, E Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, J Landau, Kamal Ndousse, Kamile Lukosiute, Liane

Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noem'i Mercado, Nova Dassarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, T. J. Henighan, Tristan Hume, Sam Bowman, Zac Hatfield-Dodds, Benjamin Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom B. Brown, and Jared Kaplan. Constitutional ai: Harmlessness from ai feedback. *ArXiv*, abs/2212.08073, 2022. URL `https://api.semanticscholar.org/CorpusID:254823489`.

Jimmy Baraglia, Maya Cakmak, Yukie Nagai, Rajesh Rao, and Minoru Asada. Initiative in robot assistance during collaborative task execution. In *HRI*, pages 67–74, 2016. doi: 10.1109/HRI.2016.7451735.

Suneel Belkhale, Tianli Ding, Ted Xiao, Pierre Sermanet, Quon Vuong, Jonathan Tompson, Yevgen Chebotar, Debidatta Dwibedi, and Dorsa Sadigh. Rt-h: Action hierarchies using language, 2024. URL `https://arxiv.org/abs/2403.01823`.

Adrien Bennetot, Vicky Charisi, and Natalia Díaz-Rodríguez. Should artificial agents ask for help in human-robot collaborative problem-solving?, 2020. URL `https://arxiv.org/abs/2006.00882`.

Justin Bishop, Jaylen Burgess, Cooper Ramos, Jade B. Driggs, Tom Williams, Chad C. Tossell, Elizabeth Phillips, Tyler H. Shaw, and Ewart J. de Visser. Chaopt: A testbed for evaluating human-autonomy team collaboration using the video game overcooked!2. In *2020 Systems and Information Engineering Design Symposium (SIEDS)*, pages 1–6, 2020. doi: 10.1109/SIEDS49339.2020.9106686.

Valts Blukis, Dipendra Misra, Ross A. Knepper, and Yoav Artzi. Mapping navigation instructions to continuous control actions with position-visitation prediction. In *Conference on Robot Learning (CoRL)*, 2018.

Valts Blukis, Yannick Terme, Eyvind Niklasson, Ross A. Knepper, and Yoav Artzi. Learning to map natural language instructions to physical quadcopter control using simulated flight. In *Conference on Robot Learning (CoRL)*, 2019.

Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3722–3731, 2017.

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

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Jaime R. Carbonell. Ai in cai: An artificial-intelligence approach to computer-assisted instruction. *IEEE Transactions on Man-Machine Systems*, 11(4):190–202, 1970. doi: 10.1109/TMMS.1970.299942.

Silvia Cascianelli, Gabriele Costante, Thomas A Ciarfuglia, Paolo Valigi, and Mario L Fravolini. Full-gru natural language video description for service robotics applications. *IEEE RA-L*, 3(2):841–848, 2018.

Andrea Censi, Konstantin Slutsky, Tichakorn Wongpiromsarn, Dmitry S. Yershov, Scott Pendleton, James Guo Ming Fu, and Emilio Frazzoli. Liability, ethics, and culture-aware behavior specification using rulebooks. *2019 International Conference on Robotics and Automation (ICRA)*, pages 8536–8542, 2019. URL `https://api.semanticscholar.org/CorpusID:67856233`.

Boyuan Chen, Fei Xia, Brian Ichter, Kanishka Rao, Keerthana Gopalakrishnan, Michael S. Ryoo, Austin Stone, and Daniel Kappler. Open-vocabulary queryable scene representations for real world planning. In *arXiv preprint arXiv:2209.09874*, 2022.

David L. Chen, Joohyun Kim, and Raymond J. Mooney. Training a multilingual sportscaster: Using perceptual context to learn language. *Journal of Artificial Intelligence Research*, 37:397–435, 2010. URL `http://www.cs.utexas.edu/users/ai-lab?chen:jair10`.

Maximillian Chen, Xiao Yu, Weiyan Shi, Urvi Awasthi, and Zhou Yu. Controllable mixed-initiative dialogue generation through prompting. *arXiv preprint arXiv:2305.04147*, 2023.

Maximillian Chen, Ruoxi Sun, Sercan O. Arık, and Tomas Pfister. Learning to clarify: Multi-turn conversations with action-based contrastive self-training, 2024. URL `https://arxiv.org/abs/2406.00222`.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *European Conference on Computer Vision*, 2019. URL `https://api.semanticscholar.org/CorpusID:216080982`.

Jae-Woo Choi, Hyungmin Kim, Hyobin Ong, Youngwoo Yoon, Minsu Jang, Jaehong Kim, et al. Reactree: Hierarchical task planning with dynamic tree expansion using llm agent nodes. 2025.

Jennifer Chu-Carroll. MIMIC: An adaptive mixed initiative spoken dialogue system for information queries. In *Sixth Applied NLP Conference*, pages 97–104, Seattle, Washington, USA, April 2000. ACL. doi: 10.3115/974147.974161. URL `https://aclanthology.org/A00-1014/`.

Erwin Coumans and Yunfei Bai. Bullet and pybullet, physics simulation for games, visual effects, robotics and reinforcement learning. `https://pybullet.org`, May 2007-2022.

Yuchen Cui, Scott Niekum, Abhinav Gupta, Vikash Kumar, and Aravind Rajeswaran. Can foundation models perform zero-shot task specification for robot manipulation? *Learning for Dynamics and Control Conference*, 2022.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The epic-kitchens dataset. In *European Conference on Computer Vision (ECCV)*, 2018.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Yang Deng, Wenqiang Lei, Wai Lam, and Tat-Seng Chua. A survey on proactive dialogue systems: Problems, methods, and prospects. *arXiv preprint arXiv:2305.02750*, 2023a.

Yang Deng, Lizi Liao, Liang Chen, Hongru Wang, Wenqiang Lei, and Tat-Seng Chua. Prompting and evaluating large language models for proactive dialogues: Clarification, target-guided, and non-collaboration, 2023b. URL `https://arxiv.org/abs/2305.13626`.

Yang Deng, Wenxuan Zhang, Wai Lam, See-Kiong Ng, and Tat-Seng Chua. Plug-and-play policy planner for large language model powered dialogue agents, 2024. URL `https://arxiv.org/abs/2311.00262`.

Coline Devin, Abhishek Gupta, Trevor Darrell, Pieter Abbeel, and Sergey Levine. Learning modular neural network policies for multi-task and multi-

robot transfer. In *International Conference on Robotics and Automation (ICRA)*, 2017.

Frederik Ebert, Yanlai Yang, Karl Schmeckpeper, Bernadette Bucher, Georgios Georgakis, Kostas Daniilidis, Chelsea Finn, and Sergey Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. *arXiv preprint arXiv:2109.13396*, 2021.

Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International Conference on Machine Learning*, pages 1407–1416. PMLR, 2018.

Maurizio Faccio, Irene Granata, and Riccardo Minto. Task allocation model for human-robot collaboration with variable cobot speed. *Journal of Intelligent Manufacturing*, 35:793–806, 2024. doi: 10.1007/s10845-023-02073-9. URL `https://link.springer.com/article/10.1007/s10845-023-02073-9`.

Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. *Advances in Neural Information Processing Systems*, 35:18343–18362, 2022.

Xueyang Feng, Zhi-Yuan Chen, Yujia Qin, Yankai Lin, Xu Chen, Zhiyuan Liu, and Ji-Rong Wen. Large language model-based human-agent collaboration for complex task solving, 2024. URL `https://arxiv.org/abs/2402.12914`.

Douglas A. Few, David J. Bruemmer, and Miles C. Walton. Improved human-robot teaming through facilitated initiative. In *ROMAN 2006 - The 15th IEEE International Symposium on Robot and Human Interactive Communication*, pages 171–176, 2006. doi: 10.1109/ROMAN.2006.314413.

Chelsea Finn, Xin Yu Tan, Yan Duan, Trevor Darrell, Sergey Levine, and Pieter Abbeel. Deep spatial autoencoders for visuomotor learning. In *International Conference on Robotics and Automation (ICRA)*, 2016.

Qiaozi Gao, Govind Thattai, Suhaila Shakiah, Xiaofeng Gao, Shreyas Pansare, Vasu Sharma, Gaurav Sukhatme, Hangjie Shi, Bofei Yang, Desheng Zheng, Lucy Hu, Karthika Arumugam, Shui Hu, Matthew Wen, Dinakar Guthy, Cadence Chung, Rohan Khanna, Osman Ipek, Leslie Ball, Kate Bland, Heather Rocker, Yadunandana Rao, Michael Johnston, Reza Ghanadan, Arindam Mandal, Dilek Hakkani Tur, and Prem Natarajan. Alexa arena: A user-centric interactive platform for embodied ai, 2023. URL `https://arxiv.org/abs/2303.01586`.

Divyansh Garg, Skanda Vaidyanath, Kuno Kim, Jiaming Song, and Stefano Ermon. Lisa: Learning interpretable skill abstractions from language, 2022. URL `https://arxiv.org/abs/2203.00054`.

Majid Ghasemi, Amir Hossein Moosavi, and Dariush Ebrahimi. A comprehensive survey of reinforcement learning: From algorithms to practical challenges. 2024. URL `https://api.semanticscholar.org/CorpusID:274422589`.

Prasoon Goyal, Scott Niekum, and Raymond Mooney. Using natural language for reward shaping in reinforcement learning. 2019. URL `https://arxiv.org/abs/1903.02020`.

Prasoon Goyal, Scott Niekum, and Raymond Mooney. Pixl2r: Guiding reinforcement learning using natural language by mapping pixels to rewards. 2020. URL `https://arxiv.org/abs/2007.15543`.

Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database

for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850, 2017.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, et al. Ego4d: Around the world in 3,000 hours of egocentric video. *arXiv preprint arXiv:2110.07058*, 2021.

Abhishek Gupta, Corey Lynch, Brandon Kinman, Garrett Peake, Sergey Levine, and Karol Hausman. Demonstration-bootstrapped autonomous practicing via multi-task reinforcement learning. *arXiv preprint arXiv:2203.15755*, 2022a.

Agrim Gupta, Stephen Tian, Yunzhi Zhang, Jiajun Wu, Roberto Martín-Martín, and Li Fei-Fei. Maskvit: Masked visual pre-training for video prediction. *arXiv preprint arXiv:2206.11894*, 2022b.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

Matteo Hessel, Hubert Soyer, Lasse Espeholt, Wojciech Czarnecki, Simon Schmitt, and Hado van Hasselt. Multi-task deep reinforcement learning with popart. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3796–3803, 2019.

Daniel Ho, Kanishka Rao, Zhuo Xu, Eric Jang, Mohi Khansari, and Yunfei Bai. Retinagan: An object-aware approach to sim-to-real transfer. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10920–10926. IEEE, 2021.

Eric Hsiung, Hiloni Mehta, Junchi Chu, Xinyu Liu, Roma Patel, Stefanie Tellex, and George Dimitri Konidaris. Generalizing to new domains by mapping natural language to lifted ltl. *2022 International Conference on Robotics and Automation (ICRA)*, pages 3624–3630, 2021. URL `https://api.semanticscholar.org/CorpusID:238634478`.

Zichao Hu, Francesca Lucchetti, Claire Schlesinger, Yash Saxena, Anders Freeman, Sadanand Modak, Arjun Guha, and Joydeep Biswas. Deploying and evaluating llms to program service mobile robots. *IEEE Robotics and Automation Letters*, 9:2853–2860, 2023. URL `https://api.semanticscholar.org/CorpusID:265294597`.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Noah Brown, Tomas Jackson, Linda Luu, Sergey Levine, Karol Hausman, and Brian Ichter. Inner monologue: Embodied reasoning through planning with language models. In *arXiv preprint arXiv:2207.05608*, 2022.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models, 2023. URL `https://arxiv.org/abs/2307.05973`.

Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. In *ACM Computing Surveys*, 2017.

Stephen James, Paul Wohlhart, Mrinal Kalakrishnan, Dmitry Kalashnikov, Alex Irpan, Julian Ibarz, Sergey Levine, Raia Hadsell, and Konstantinos Bousmalis. Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12627–12637, 2019.

Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. BC-z: Zero-shot task generalization with robotic imitation learning. In *5th Annual Conference on Robot Learning*, 2021. URL `https://openreview.net/forum?id=8kbp23tSGYv`.

Shu Jiang and Ronald C. Arkin. Mixed-initiative human-robot interaction: Definition, taxonomy, and survey. In *2015 IEEE International Conference on Systems, Man, and Cybernetics*, pages 954–961, 2015. doi: 10.1109/SMC.2015.174.

Emily Jin, Jiaheng Hu, Zhuoyi Huang, Ruohan Zhang, Jiajun Wu, Li Fei-Fei, and Roberto Martín-Martín. Mini-behavior: A procedurally generated benchmark for long-horizon decision-making in embodied ai. *arXiv preprint 2310.01824*, 2023.

Ya Jing, Xuelin Zhu, Xingbin Liu, Qie Sima, Taozheng Yang, Yunhai Feng, and Tao Kong. Exploring visual pre-training for robot manipulation: Datasets, models and methods. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11390–11395. IEEE, 2023.

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

Siddharth Karamcheti, Megha Srivastava, Percy Liang, and Dorsa Sadigh. Lila: Language-informed latent actions. In *5th Annual Conference on Robot Learning*, 2021. URL `https://arxiv.org/pdf/2111.03205`.

Manuel Kaspar, Juan D Muñoz Osorio, and Jürgen Bock. Sim2real transfer for reinforcement learning without dynamics randomization. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4383–4388. IEEE, 2020.

Oussama Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, 1987.

Ryan Kiros, Ruslan Salakhutdinov, and Richard S. Zemel. Multimodal neural language models. In *International Conference on Machine Learning*, 2014. URL `https://api.semanticscholar.org/CorpusID:12365096`.

Ross A. Knepper, Todd Layton, John Romanishin, and Daniela Rus. Ikeabot: An autonomous multi-robot coordinated furniture assembly system. In *2013 IEEE International Conference on Robotics and Automation*, pages 855–862, 2013. doi: 10.1109/ICRA.2013.6630673.

Noriyuki Kojima, Alane Suhr, and Yoav Artzi. Continual learning for grounded instruction generation by observing human following behavior, 2021. URL `https://arxiv.org/abs/2108.04812`.

Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*, pages 5639–5650. PMLR, 2020.

Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. *ArXiv*, abs/2005.11401, 2020. URL `https://api.semanticscholar.org/CorpusID:218869575`.

Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, Mona Anvari, Minjune Hwang, Manasi Sharma, Arman Aydin, Dhruva Bansal, Samuel Hunter, Kyu-Young Kim, Alan Lou, Caleb R Matthews, Ivan Villa-Renteria, Jerry Huayang Tang, Claire Tang, Fei Xia, Silvio Savarese,

Hyowon Gweon, Karen Liu, Jiajun Wu, and Li Fei-Fei. BEHAVIOR-1k: A benchmark for embodied AI with 1,000 everyday activities and realistic simulation. In *CoRL*, 2022. URL https://openreview.net/forum?id=_8DoIe8G3t.

Chengshu Li, Jacky Liang, Andy Zeng, Xinyun Chen, Karol Hausman, Dorsa Sadigh, Sergey Levine, Li Fei-Fei, Fei Xia, and Brian Ichter. Chain of code: Reasoning with a language model-augmented code emulator. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 28259–28277. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/li24ar.html.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *ArXiv*, abs/1908.03557, 2019. URL https://api.semanticscholar.org/CorpusID:199528533.

Yitong Li, Martin Renqiang Min, Dinghan Shen, David Edwin Carlson, and Lawrence Carin. Video generation from text. In *AAAI Conference on Artificial Intelligence*, 2017. URL https://api.semanticscholar.org/CorpusID:8672818.

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

Jason Liu, Ziyi Yang, Ifrah Idrees, Sam Liang, Benjamin Schornstein, Stefanie Tellex, and Ankit Shah. Grounding complex natural language commands for temporal tasks in unseen environments. In *Conference on Robot Learning*, 2023a. URL https://api.semanticscholar.org/CorpusID:257102641.

Jason Xinyu Liu, Ziyi Yang, Ifrah Idrees, Sam Liang, Benjamin Schornstein, Stefanie Tellex, and Ankit Shah. Lang2ltl: Translating natural language commands to temporal robot task specification. *ArXiv*, abs/2302.11649, 2023b. URL https://api.semanticscholar.org/CorpusID:267931252.

Jason Xinyu Liu, Ankit Shah, George Konidaris, Stefanie Tellex, and David Paulius. Lang2LTL-2: Grounding spatiotemporal navigation commands using large language and vision-language models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024a.

Jason Xinyu Liu, Ankit Shah, Eric Rosen, Mingxi Jia, George Dimitri Konidaris, and Stefanie Tellex. Skill transfer for temporal task specification. *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2535–2541, 2024b. URL https://api.semanticscholar.org/CorpusID:271798955.

Xingzhou Lou, Junge Zhang, Ziyan Wang, Kaiqi Huang, and Yali Du. Safe reinforcement learning with free-form natural language constraints and pre-trained language models. In *Adaptive Agents and Multi-Agent Systems*, 2024. URL https://api.semanticscholar.org/CorpusID:266999399.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Neural Information Processing Systems*, 2019. URL https://api.semanticscholar.org/CorpusID:199453025.

Xusheng Luo, Shaojun Xu, and Changliu Liu. Obtaining hierarchy from human instructions: an llms-based approach. In *CoRL 2023 Workshop on Learning Effective Abstractions for Planning (LEAP)*, 2023.

Corey Lynch and Pierre Sermanet. Language conditioned imitation learning over unstructured data. *Robotics: Science and Systems*, 2021. URL https://arxiv.org/abs/2005.07648.

Yecheng Jason Ma, William Liang, Vaidehi Som, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. Liv: Language-image representations and rewards for robotic control. *arXiv preprint arXiv:2306.00958*, 2023.

Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models, 2024a. URL https://arxiv.org/abs/2310.12931.

Yecheng Jason Ma, William Liang, Hungju Wang, Sam Wang, Yuke Zhu, Linxi Fan, Osbert Bastani, and Dinesh Jayaraman. Dreureka: Language model guided sim-to-real transfer. In *Robotics: Science and Systems (RSS)*, 2024b.

Zhao Mandi, Shreeya Jain, and Shuran Song. Roco: Dialectic multi-robot collaboration with large language models, 2023. URL https://arxiv.org/abs/2307.04738.

Elman Mansimov, Emilio Parisotto, Jimmy Ba, and Ruslan Salakhutdinov. Generating images from captions with attention. *CoRR*, abs/1511.02793, 2015. URL https://api.semanticscholar.org/CorpusID:9996719.

Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, and Alan Loddon Yuille. Explain images with multimodal recurrent neural networks. *ArXiv*, abs/1410.1090, 2014. URL https://api.semanticscholar.org/CorpusID:3527896.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3190–3199, 2019. URL https://api.semanticscholar.org/CorpusID:173991173.

Jan Matas, Stephen James, and Andrew J Davison. Sim-to-real reinforcement learning for deformable object manipulation. In *Conference on Robot Learning*, pages 734–743. PMLR, 2018.

Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks, 2021. URL `https://arxiv.org/abs/2112.03227`.

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

Matthias Müller, Alexey Dosovitskiy, Bernard Ghanem, and Vladlen Koltun. Driving policy transfer via modularity and abstraction. *arXiv preprint arXiv:1804.09364*, 2018.

Bilge Mutlu, Jodi Forlizzi, and Jessica Hodgins. A storytelling robot: Modeling and evaluation of human-like gaze behavior. In *2006 6th IEEE-RAS International Conference on Humanoid Robots*, pages 518–523, 2006. doi: 10.1109/ICHR.2006.321322.

Vivek Myers, Chongyi Zheng, Anca Dragan, Sergey Levine, and Benjamin Eysenbach. Learning temporal distances: Contrastive successor features can provide a metric structure for decision-making, 2025. URL `https://arxiv.org/abs/2406.17098`.

Ashvin Nair, Dian Chen, Pulkit Agrawal, Phillip Isola, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Combining self-supervised learning and imitation for vision-based rope manipulation. In *International Conference on Robotics and Automation (ICRA)*, 2017. URL `https://arxiv.org/abs/1703.02018`.

Ashvin Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, and Sergey Levine. Visual reinforcement learning with imagined goals. In *Advances*

in Neural Information Processing Systems (NeurIPS), 2018. URL https://arxiv.org/abs/1807.04742.

Suraj Nair, Eric Mitchell, Kevin Chen, Brian Ichter, Silvio Savarese, and Chelsea Finn. Learning language-conditioned robot behavior from offline data and crowd-sourced annotation. In *5th Annual Conference on Robot Learning*, 2021. URL https://arxiv.org/pdf/2109.01115.

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

Soroush Nasiriany, Vitchyr Pong, Steven Lin, and Sergey Levine. Planning with goal-conditioned policies. 2019.

Manisha Natarajan, Chunyue Xue, Sanne van Waveren, Karen Feigh, and Matthew Gombolay. Mixed-initiative human-robot teaming under suboptimality with online bayesian adaptation, 2024. URL https://arxiv.org/abs/2403.16178.

OpenAI, Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, Jonas Schneider, Nikolas Tezak, Jerry Tworek, Peter Welinder, Lilian Weng, Qiming Yuan, Wojciech Zaremba, and Lei Zhang. Solving rubik's cube with a robot hand, 2019.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *NeurIPS*, 35:27730–27744, 2022.

Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan,

et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023.

Rohan Paleja, Michael Munje, Kimberlee Chang, Reed Jensen, and Matthew Gombolay. Designs for enabling collaboration in human-machine teaming via interactive and explainable systems, 2024. URL `https://arxiv.org/abs/2406.05003`.

Yingwei Pan, Zhaofan Qiu, Ting Yao, Houqiang Li, and Tao Mei. To create what you tell: Generating videos from captions. *Proceedings of the 25th ACM international conference on Multimedia*, 2017. URL `https://api.semanticscholar.org/CorpusID:5039505`.

Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov. Actor-mimic: Deep multitask and transfer reinforcement learning. *arXiv preprint arXiv:1511.06342*, 2015.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI*, 2018.

Dean Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In *Conference on Neural Information Processing Systems (NeurIPS)*, 1988.

Kun Qian, Ahmad Beirami, Satwik Kottur, Shahin Shayandeh, Paul Crook, Alborz Geramifard, Zhou Yu, and Chinnadhurai Sankar. Database search results disambiguation for task-oriented dialog systems, 2021. URL `https://arxiv.org/abs/2112.08351`.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training. In *Conference on Robot Learning*, pages 416–426. PMLR, 2023.

Shreyas Sundara Raman, Vanya Cohen, David Paulius, Ifrah Idrees, Eric Rosen, Ray Mooney, and Stefanie Tellex. Cape: Corrective actions from precondition errors using large language models. In *2nd Workshop on Language and Robot Learning: Language as Grounding*, 2023.

Kanishka Rao, Chris Harris, Alex Irpan, Sergey Levine, Julian Ibarz, and Mohi Khansari. Rl-cyclegan: Reinforcement learning aware simulation-to-real. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11157–11166, 2020.

Scott E. Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *International Conference on Machine Learning*, 2016. URL `https://api.semanticscholar.org/CorpusID:1563370`.

Allen Z. Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, Zhenjia Xu, Dorsa Sadigh, Andy Zeng, and Anirudha Majumdar. Robots that ask for help: Uncertainty alignment for large language model planners, 2023. URL `https://arxiv.org/abs/2307.01928`.

Andres Rosero, Faustina Dinh, Ewart J. de Visser, Tyler Shaw, and Elizabeth Phillips. Two many cooks: Understanding dynamic human-agent team communication and perception using overcooked 2, 2021. URL `https://arxiv.org/abs/2110.03071`.

Andrei Rusu, Sergio Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Krikpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy distillation. *arXiv preprint arXiv:1511.06295v2*, 2015.

Stefan Schaal. Is imitation learning the route to humanoid robots? *Trends in cognitive sciences*, 3(6):233–242, 1999.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *ArXiv*, abs/1707.06347, 2017. URL `https://api.semanticscholar.org/CorpusID:28695052`.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*, 2020.

Mario Selvaggio, Marco Cognetti, Stefanos Nikolaidis, Serena Ivaldi, and Bruno Siciliano. Autonomy in physical human-robot interaction: A brief survey. *IEEE RA-L*, 6(4):7989–7996, 2021.

Younggyo Seo, Danijar Hafner, Hao Liu, Fangchen Liu, Stephen James, Kimin Lee, and Pieter Abbeel. Masked world models for visual control. In *Conference on Robot Learning*, pages 1332–1344. PMLR, 2023.

Lin Shao, Toki Migimatsu, Qiang Zhang, Karen Yang, and Jeannette Bohg. Concept2robot: Learning manipulation concepts from instructions and human demonstrations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2020.

Lucy Xiaoyang Shi, Zheyuan Hu, Tony Z. Zhao, Archit Sharma, Karl Pertsch, Jianlan Luo, Sergey Levine, and Chelsea Finn. Yell at your robot: Improving on-the-fly from language corrections, 2024. URL `https://arxiv.org/abs/2403.12910`.

Lucy Xiaoyang Shi, Brian Ichter, Michael Equi, Liyiming Ke, Karl Pertsch, Quan Vuong, James Tanner, Anna Walling, Haohuan Wang, Niccolo Fusai, Adrian Li-Bell, Danny Driess, Lachy Groom, Sergey Levine, and Chelsea Finn. Hi robot: Open-ended instruction following with hierarchical vision-language-action models, 2025. URL `https://arxiv.org/abs/2502.19417`.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Proceedings of the 5th Conference on Robot Learning (CoRL)*, 2021.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. *Conference on Robot Learning*, 2022.

Andrew Silva, Nina Moorman, William Silva, Zulfiqar Zaidi, Nakul Gopalan, and Matthew Gombolay. Lancon-learn: Learning with language to enable generalization in multi-task manipulation. In *IEEE Robotics and Automation Letters*, 2021.

Sukriti Singh, Anusha Srikanthan, Vivek Mallampati, and Harish Ravichandar. Concurrent constrained optimization of unknown rewards for multi-robot task allocation, 2023. URL `https://arxiv.org/abs/2305.15288`.

Laura Smith, Nikita Dhawan, Marvin Zhang, Pieter Abbeel, and Sergey Levine. Avid: Learning multi-stage tasks via pixel-level translation of human videos. In *Robotics Science and Systems (RSS)*, 2020.

Richard Socher, Andrej Karpathy, Quoc V. Le, Christopher D. Manning, and A. Ng. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics*, 2:207–218, 2014. URL `https://api.semanticscholar.org/CorpusID:2317858`.

Shagun Sodhani, Amy Zhang, and Joelle Pineau. Multi-task reinforcement learning with context-based representations. *arXiv preprint arXiv:2102.06177*, 2021.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. *ArXiv*,

abs/1908.08530, 2019. URL https://api.semanticscholar.org/CorpusID:201317624.

Alane Suhr, Claudia Yan, Charlotte Schluger, Stanley Yu, Hadi Khader, Marwa Mouallem, Iris Zhang, and Yoav Artzi. Executing instructions in situated collaborative interactions, 2022. URL https://arxiv.org/abs/1910.03655.

Chen Sun, Austin Myers, Carl Vondrick, Kevin P. Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7463–7472, 2019. URL https://api.semanticscholar.org/CorpusID:102483628.

Hao Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *Conference on Empirical Methods in Natural Language Processing*, 2019. URL https://api.semanticscholar.org/CorpusID:201103729.

Matthew E. Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *J. Mach. Learn. Res.*, 10:1633–1685, 2009. URL https://api.semanticscholar.org/CorpusID:17216004.

DeepMind Interactive Agents Team, Josh Abramson, Arun Ahuja, Arthur Brussee, Federico Carnevale, Mary Cassin, Felix Fischer, Petko Georgiev, Alex Goldin, Mansi Gupta, Tim Harley, Felix Hill, Peter C Humphreys, Alden Hung, Jessica Landon, Timothy Lillicrap, Hamza Merzic, Alistair Muldal, Adam Santoro, Guy Scully, Tamara von Glehn, Greg Wayne, Nathaniel Wong, Chen Yan, and Rui Zhu. Creating multimodal interactive agents with imitation and self-supervised learning, 2022. URL https://arxiv.org/abs/2112.03763.

Yee Whye Teh, Victor Bapst, Wojciech Marian Czarnecki, John Quan, James Kirkpatrick, Raia Hadsell, Nicolas Heess, and Razvan Pascanu. Distral: Robust multitask reinforcement learning. *arXiv preprint arXiv:1707.04175*, 2017.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance, 2014.

Shivam Vats, Oliver Kroemer, and Maxim Likhachev. Synergistic scheduling of learning and allocation of tasks in human-robot teams, 2022. URL `https://arxiv.org/abs/2203.07478`.

Manuela Veloso, Joydeep Biswas, Brian Coltin, and Stephanie Rosenthal. Cobots: robust symbiotic autonomous mobile service robots. In *IJCAI, 2015*, page 4423–4429. AAAI Press, 2015. ISBN 9781577357384.

Subhashini Venugopalan, Marcus Rohrbach, Jeff Donahue, Raymond J. Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence – video to text. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4534–4542, 2015. URL `https://api.semanticscholar.org/CorpusID:4228546`.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and D. Erhan. Show and tell: A neural image caption generator. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3156–3164, 2014. URL `https://api.semanticscholar.org/CorpusID:1169492`.

Homer Walke, Jonathan Yang, Albert Yu, Aviral Kumar, Jedrzej Orbik, Avi Singh, and Sergey Levine. Don't start from scratch: Leveraging prior data to

automate robotic reinforcement learning. In *Proceedings of the 6th Conference on Robot Learning (CORL)*, 2022.

Che Wang, Xufang Luo, Keith Ross, and Dongsheng Li. Vrl3: A data-driven framework for visual deep reinforcement learning. *Advances in Neural Information Processing Systems*, 35:32974–32988, 2022.

Huaxiaoyue Wang, Kushal Kedia, Juntao Ren, Rahma Abdullah, Atiksh Bhardwaj, Angela Chao, Kelly Y Chen, Nathaniel Chin, Prithwish Dan, Xinyi Fan, Gonzalo Gonzalez-Pumariega, Aditya Kompella, Maximus Adrian Pace, Yash Sharma, Xiangwan Sun, Neha Sunkara, and Sanjiban Choudhury. Mosaic: A modular system for assistive and interactive cooking, 2024a. URL `https://arxiv.org/abs/2402.18796`.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers, 2020.

Ziyan Wang, Meng Fang, Tristan Tomilin, Fei Fang, and Yali Du. Safe multi-agent reinforcement learning with natural language constraints. *ArXiv*, abs/2405.20018, 2024b. URL `https://api.semanticscholar.org/CorpusID:270123483`.

Aaron Wilson, Alan Fern, Soumya Ray, and Prasad Tadepalli. Multi-task reinforcement learning: a hierarchical bayesian approach. In *International Conference on Machine Learning*, 2007a. URL `https://api.semanticscholar.org/CorpusID:6225453`.

Aaron Wilson, Alan Fern, Soumya Ray, and Prasad Tadepalli. Multi-task reinforcement learning: a hierarchical bayesian approach. In *Proceedings of the 24th international conference on Machine learning*, pages 1015–1022, 2007b.

Jim Woodcock, Peter Gorm Larsen, Juan Bicarregui, and John S. Fitzgerald. Formal methods: Practice and experience. *ACM Comput. Surv.*, 41:19:1–19:36, 2009. URL https://api.semanticscholar.org/CorpusID:195345947.

Shirley Wu, Michel Galley, Baolin Peng, Hao Cheng, Gavin Li, Yao Dou, Weixin Cai, James Zou, Jure Leskovec, and Jianfeng Gao. Collabllm: From passive responders to active collaborators. In *ICML*, 2025.

Zhiyuan Xu, Kun Wu, Zhengping Che, Jian Tang, and Jieping Ye. Knowledge transfer in multi-task deep reinforcement learning for continuous control. *arXiv preprint arXiv:2010.07494*, 2020.

Ruihan Yang, Huazhe Xu, Yi Wu, and Xiaolong Wang. Multi-task reinforcement learning with soft modularization. *arXiv preprint arXiv:2003.13661*, 2020a.

Tsung-Yen Yang, Michael Y Hu, Yinlam Chow, Peter J. Ramadge, and Karthik Narasimhan. Safe reinforcement learning with natural language constraints. *ArXiv*, abs/2010.05150, 2020b. URL https://api.semanticscholar.org/CorpusID:222291341.

L. Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Joseph Pal, H. Larochelle, and Aaron C. Courville. Describing videos by exploiting temporal structure. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4507–4515, 2015. URL https://api.semanticscholar.org/CorpusID:623318.

Sarah Young, Dhiraj Gandhi, Shubham Tulsiani, Abhinav Gupta, Pieter Abbeel, and Lerrel Pinto. Visual imitation made easy, 2020.

Albert Yu and Raymond J Mooney. Using both demonstrations and language instructions to efficiently learn robotic tasks. *arXiv preprint arXiv:2210.04476*, 2022.

Albert Yu, Adeline Foote, Raymond Mooney, and Roberto Martín-Martín. Natural language can help bridge the sim2real gap. In *Robotics: Science and Systems (RSS), 2024*, 2024.

Albert Yu, Chengshu Li, Luca Macesanu, Arnav Balaji, Ruchira Ray, Raymond Mooney, and Roberto Mart'in-Mart'in. Mixed-initiative dialog for human-robot collaborative manipulation. *ArXiv*, abs/2508.05535, 2025. URL `https://api.semanticscholar.org/CorpusID:280546031`.

Tian Yu, Jing Huang, and Qing Chang. Optimizing task scheduling in human-robot collaboration with deep multi-agent reinforcement learning. *Journal of Manufacturing Systems*, 60:487–499, 2021a. ISSN 0278-6125. doi: https://doi.org/10.1016/j.jmsy.2021.07.015. URL `https://www.sciencedirect.com/science/article/pii/S0278612521001527`.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2019. URL `https://arxiv.org/abs/1910.10897`.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *arXiv preprint arXiv:2001.06782*, 2020.

Tianhe Yu, Aviral Kumar, Yevgen Chebotar, Karol Hausman, Sergey Levine, and Chelsea Finn. Conservative data sharing for multi-task offline reinforcement learning. *Advances in Neural Information Processing Systems*, 34, 2021b.

Wenhao Yu, Jie Tan, C Karen Liu, and Greg Turk. Preparing for the unknown: Learning a universal policy with online system identification. *arXiv preprint arXiv:1702.02453*, 2017.

Wenhao Yu, Nimrod Gileadi, Chuyuan Fu, Sean Kirmani, Kuang-Huei Lee, Montse Gonzalez Arenas, Hao-Tien Lewis Chiang, Tom Erez, Leonard Hasenclever, Jan Humplik, Brian Ichter, Ted Xiao, Peng Xu, Andy Zeng, Tingnan Zhang, Nicolas Heess, Dorsa Sadigh, Jie Tan, Yuval Tassa, and Fei Xia. Language to rewards for robotic skill synthesis, 2023a. URL `https://arxiv.org/abs/2306.08647`.

Xiao Yu, Maximillian Chen, and Zhou Yu. Prompt-based monte-carlo tree search for goal-oriented dialogue policy planning, 2023b. URL `https://arxiv.org/abs/2305.13660`.

Zhecheng Yuan, Zhengrong Xue, Bo Yuan, Xueqian Wang, Yi Wu, Yang Gao, and Huazhe Xu. Pre-trained image encoder for generalizable visual reinforcement learning. *Advances in Neural Information Processing Systems*, 35:13022–13037, 2022.

Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Vikas Sindhwani, et al. Transporter networks: Rearranging the visual world for robotic manipulation. In *Proceedings of the 4th Conference on Robot Learning (CoRL)*, 2020.

Andy Zeng, Maria Attarian, Brian Ichter, Krzysztof Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael Ryoo, Vikas Sindhwani, Johnny Lee, Vincent Vanhoucke, and Pete Florence. Socratic models: Composing zero-shot multimodal reasoning with language. *arXiv*, 2022.

Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. *CoRR*, abs/2111.07991, 2021. URL `https://arxiv.org/abs/2111.07991`.

Tony Zhao, Siddharth Karamcheti, Thomas Kollar, Chelsea Finn, and Percy Liang. What makes representation learning from videos hard for control? 2022. URL `https://api.semanticscholar.org/CorpusID:252635608`.

Qinhong Zhou, Sunli Chen, Yisong Wang, Haozhe Xu, Weihua Du, Hongxin Zhang, Yilun Du, Josh Tenenbaum, and Chuang Gan. Hazard challenge: Embodied decision making in dynamically changing environments. *ArXiv*, abs/2401.12975, 2024. URL `https://api.semanticscholar.org/CorpusID:267095203`.

Bin Zhu, Bin Lin, Munan Ning, Yang Yan, Jiaxi Cui, HongFa Wang, Yatian Pang, Wenhao Jiang, Junwu Zhang, Zongwei Li, et al. Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment. *arXiv preprint arXiv:2310.01852*, 2023.

Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020.