This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2021.3139667, IEEE Robotics and Automation Letters



# LANCON-LEARN: Learning with Language to Enable Generalization in Multi-Task Manipulation

Andrew Silva[1], Nina Moorman, William Silva, Zulfiqar Zaidi, Nakul Gopalan, and Matthew Gombolay

***Abstract*—Robots must be capable of learning from previously solved tasks and generalizing that knowledge to quickly perform new tasks to realize the vision of ubiquitous and useful robot assistance in the real world. While multi-task learning research has produced agents capable of performing multiple tasks, these tasks are often encoded as one-hot goals. In contrast, natural language specifications offer an accessible means both for (1) users to describe a set of new tasks to the robot and (2) robots to reason about the similarities and differences among tasks through language-based task embeddings. Until now, multi-task learning with language has been limited to navigation based tasks and has not been applied to continuous manipulation tasks, requiring precision to grasp and move objects. We present LANCON-LEARN, a novel attention-based approach to language-conditioned multi-task learning in manipulation domains to enable learning agents to reason about relationships between skills and task objectives through natural language and interaction. We evaluate LANCON-LEARN for both reinforcement learning and imitation learning, across multiple virtual robot domains along with a demonstration on a physical robot. LANCON-LEARN achieves up to a 200% improvement in zero-shot task success rate and transfers known skills to novel tasks faster than non-language-based baselines, demonstrating the utility of language for goal specification.**

***Index Terms*—Deep Learning Methods, Imitation Learning, Reinforcement Learning**

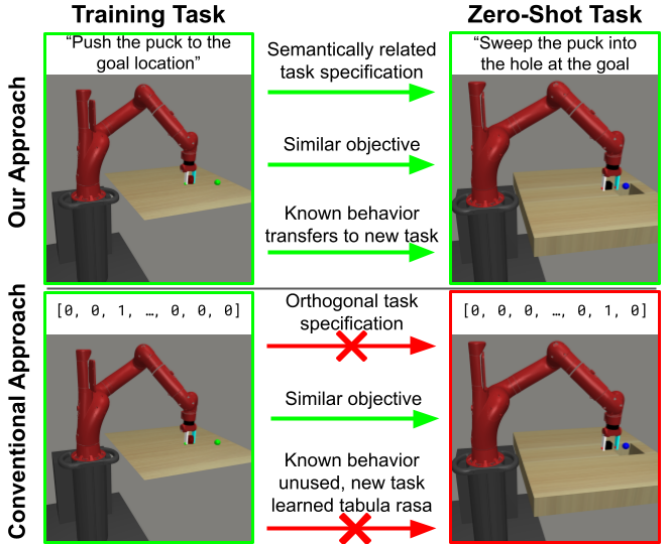



Fig. 1: LANCON-LEARN applied to multi-task learning with (left) high-level language commands embedded into semantically meaningful vectors versus (right) traditional, one-hot encodings requiring the robot to learn from scratch.

## I. INTRODUCTION

MULTI-TASK learning offers the promise of intelligent agents by enabling robots to accomplish multiple tasks specified by end-users. To program these agents with policies that map the state of the world to the correct action to accomplish a given task, users are typically expected to either define reward functions, in the case of reinforcement learning (RL), or to provide demonstrations or corrective feedback, in the case of imitation learning (IL). Despite the progress in multi-task learning, much work still centers around defining tasks as an orthonormal space (i.e., tasks are "one-hot" encoded in an orthonormal set [1], [2], [3]). However, defining tasks as an orthonormal space prohibits robots from identifying and leveraging semantic similarities across tasks. Further, one-hot encodings do not allow for flexibility in growing the agent's capacity to new problems, as the size of the orthonormal space is fixed a priori. While some prior work relaxes the assumption of defining the number of tasks a priori by leveraging a continuous embedding space [4], [5], [6], [7], these approaches do not incorporate human feedback, knowledge, or specification, thereby ignoring useful prior knowledge. What is critically needed is a methodology for specifying robot tasks that is both flexible to the addition of new tasks *and* conveys prior knowledge about the goals or objectives of the tasks, i.e. through natural language.

Natural language affords a unique opportunity to leverage task-specific semantic knowledge and is a convenient medium for many users to provide task objectives for robots, being more intuitive than a one-hot encoding. Leveraging language embeddings overcomes the limitations of learned task-embeddings by distilling natural language into semantically rich embeddings; however, prior work in mapping language to task abstractions has primarily considered navigation problems [8], [9], [10], [11], [12]. In contrast to prior work, we consider a set of diverse manipulation skills in a multi-task setup [2], involving a higher-dimension action space in which small mistakes are more likely to lead to complete task failures, thus complicating the learning process. Furthermore, our learning domain consists of tasks with randomized elements (e.g., objects spawn in random locations), making even one task in our work a multi-task learning problem by prior definitions [13]. Our approach enables us to tackle such diverse tasks with a single agent

Manuscript received: September 9, 2021; Revised November, 25, 2021; Accepted December, 19, 2021.

This paper was recommended for publication by Tamim Asfour upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by a gift from Konica Minolta and the Office of Naval Research under grant N00014-19-1-2076

[1]Georgia Institute of Technology, Atlanta, GA 30332, USA. Corresponding email: andrew.silva@gatech.edu

Digital Object Identifier (DOI): see top of this page.

2377-3766 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on January 06,2022 at 17:51:53 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2021.3139667, IEEE Robotics and Automation Letters



because language-commands provide significantly more task-relevant information than conventional approaches to goal-specification for learning agents.

In this paper, we present Language Conditioned Learning (LANCON-LEARN), a novel approach for specifying goals to multi-task learning agents by embedding natural language sequences into semantically task-relevant goal embeddings (i.e., text-based natural language commands mapped to real-valued vectors for use by neural network policies). The key to our success is the application of attention to a language-specified goal for the activation of task-relevant skills, eliciting goal-conditioned behavior by attending to salient components of a language command. We leverage the insight that language sequences contain semantically useful information to relate task objectives in a meaningful way, where prior work instead relies on uninformative, one-hot encodings or task-specific embeddings. Leveraging a language model to translate text-based goal specifications into real-valued vectors for input as goal embeddings, LANCON-LEARN generalizes to unseen tasks where conventional methods fail, leveraging prior knowledge and the relationships between known and novel goals. We demonstrate the power of our approach using both RL and IL. Our IL setup enables end-users to naturally teach robots multiple tasks using language and corrective feedback, showing four-fold improvement in zero-shot skill transfer compared to baselines. Through reinforcement, we offer an alternative in which the robot learns from a language command and a defined reward function, if corrective feedback from a human teacher is impractical [14], achieving over three-fold improvement in zero-shot skill transfer relative to baselines. We set a new state-of-the-art for zero-shot task success and few-shot knowledge transfer in Meta-World [15], achieving up to 100% success-rates on novel tasks with no prior experience.

**Our Contributions**:

- Introduce LANCON-LEARN, a novel approach to multi-task learning leveraging prior knowledge from language rather than naively learning from one-hot encodings.
- Demonstrate our technique in two distinct paradigms (IL and RL), yielding significant performance improvements.
- Validate the efficacy of LANCON-LEARN on zero-shot learning, showing up to 200% improvement on task success relative to baseline approaches [2], [16].
- Demonstrate LANCON-LEARN on transfer learning tasks, showing that our approach enables multi-task robots to leverage language to quickly outperform baselines and achieve 100% success on new tasks.

## II. RELATED WORK

**Language-guided Learning –** Early approaches to language-guided behavior needed to consider grounding language to symbols in the environment, often using parallel data to acquire language [17], [18], [19], [20]. These approaches, as well as more modern research [8], [21], [22], [23], [24], [25], [26], [27], [28], typically require a large corpus of commands or observations with granular instructions, as well as a detailed alignment of commands to states. Contemporary work demonstrates the utility of language commands for multi-task RL [29], while our work expands language-conditioning to encompass both IL and RL. For a review of recent crossover between language and task learning, we refer readers to [30].

The instruction-following literature includes several examples of language-based learning agents, centered on navigation [31], [32], [33], [10], [9], [34], [12], [35] or simple manipulation tasks [11], [36], [22]. While navigation requires coarse control in two dimensions, manipulation with realistic physics requires fine-grained control in three dimensions in addition to precise gripper control. Furthermore, prior approaches are primarily concerned with mapping language to a goal *location* rather than mapping language to a goal *location and skill*. Our work therefore targets both a broader set of skills and a more complex, demanding action-space.

Finally, research has framed the task-learning problem as one of program synthesis [37] or program-guided learning [38], [39], [40], [41]. Such work relies on low-level control for specific movement or sub-goal specification, rather than abstract, high-level commands. Our work presents an abstraction of such low-level instruction, leveraging only high-level language commands for robot learning to complete tasks without directly considering sub-goals or constraints on task execution. Our approach opens up opportunities in future work for more natural human-robot interactions [42] while providing a useful goal embedding in the process.

**Multi-task Reinforcement Learning –** Multi-task RL is a subfield of reinforcement learning concerned with teaching a single agent to solve multiple tasks. Whereas single-task RL optimizes for a single reward function, multi-task robots must learn robust policies that satisfy multiple objectives. While there are advantages to a multi-task learning setup (e.g., additional data to learn state representations, useful initializations, or multiple views of the same task [43], [44], [45]), there are often practical challenges such as catastrophic forgetting and conflicting gradients when learning multiple tasks [46].

One popular approach to multi-task RL is to employ hierarchical [23], [47] or modular neural networks [31], [48], [2]. Both approaches attempt to learn different sub-components of a larger network which can each specialize in a sub-task. In this work, we build on the notion of "soft-modularization" [2], which is an approach aimed at improving the generality of modular neural networks. Instead of attempting to learn discrete sub-tasks or modules, we learn a general set of modules alongside attention mechanisms. The attention mechanisms are conditioned on goals and activate necessary modules by re-weighting state representations between behavior modules.

Prior work unifying natural language and multi-task learning is primarily centered around low-level instructions [49], [50], [21] or other meta-data [29] for task specifications. These approaches require large datasets of annotated text and tedious alignment between language and specific motions. In this work, we contribute a novel approach to specifying goals for multi-task learning agents which circumvents the need for fine-grained, low-level instruction, yet still enables learning agents to leverage the power of natural language for identifying relationships between tasks.

**Language Modeling –** Our work is motivated by the insight that language contains useful, task-relevant information. Word-embedding techniques, e.g. GloVe [51], learn to associate

2377-3766 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on January 06,2022 at 17:51:53 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2021.3139667, IEEE Robotics and Automation Letters



individual tokens with neighboring tokens, learning the "meaning" of words. Recurrent neural network models, such as the long-short-term-memory network (LSTM) [52] have since been used to learn longer sequence-level representations (e.g., sentences). Modern approaches, including transformer models like BERT [53], have advanced the state of the art to consider context, rather than simply words in a vacuum. In our work, we leverage word embedding models and modern transformers.

## III. PRELIMINARIES

**Markov Decision Processes –** We represent each task in our multi-task learning setup as a Markov Decision Process (MDP), which is a 6-tuple of $\langle S, A, P, R, G, \gamma \rangle$ where $S$ is the set of all possible states and $A$ denotes the action space for the domain. $P : S \times A \times S' \rightarrow [0, 1]$ denotes the environment dynamics or the transition function, providing the probability of arriving in a new state $s'$ after taking action $a$ in state $s$. $R : S \times A \times S' \rightarrow \mathbb{R}$ is the reward function, and $\gamma$ is a discount factor to determine the weight of future value for each state. Our multi-task robot must find a policy, $\pi : S \times G \rightarrow [0, 1]^{|A|}$, that samples actions given states and task goals to maximize the long-term expected reward. Our agent estimates the value of current states $s$ under the policy, $\pi$, by the value function $V^{\pi}(s, g)$, which the policy seeks to maximize as given by $V^{\pi}(s, g) = \mathbb{E}_{a \sim \pi(s,g), s' \sim P(\cdot|s,a)} R(s, \pi(s, g)) + \gamma V^{\pi}(s', g)$.

**Dataset Aggregation (DAgger) –** To directly learn how to complete multiple tasks from a teacher, we consider the dataset aggregation (DAgger) approach to IL [54]. By imitating an expert, we can learn skills faster than with RL and demonstrate a path to multi-task learning from demonstration. Within DAgger, the learning agent guides the learning process by performing exploration and taking the best action in each state. The learning agent intermittently queries the teacher for action labels for each state the learner has encountered. As we are learning over $N$-dimensional continuous actions, we employ a mean-squared error loss, $L_{\pi}(\theta) = \sum_{n=0}^{N} (y_{n,t} - \pi_{\theta}(s_{n,t}, g_{n,t}))^2$, with respect to the teacher's actions $y$, state, $s$, and goal, $g$, at time $t$.

**Soft Actor-Critic –** To alleviate the burden on human teachers, we also explore multi-task RL, which requires comparatively little effort from humans. To learn $\pi$ in our multi-task RL problems, we leverage Soft Actor-Critic (SAC) [55]. In this framework, the agent must learn to maximize both expected task-return and entropy over available actions throughout the policy rollout. The full policy update, modified for our multi-task learning setup, is given by $J_{\pi}(\theta) = \mathbb{E}_{a_t \sim \pi_{\theta}, s_t, g_t \sim \mathcal{D}}[\alpha \log(\pi_{\theta}(a_t|s_t, g_t)) - Q_{\omega}(s_t, g_t, a_t)]]$.

## IV. APPROACH

### A. Soft-modular Architecture

Specifying goals in LANCON-LEARN using natural language, we leverage a soft-modular [2] neural network architecture. This architecture learns a policy directly over task information and activates sub-components of the policy according to goal embeddings, as shown in Fig. 2. Each step in a domain, the agent receives state and goal information from the environment and returns an action. The state information is embedded for use by the two central components of the architecture: the behavior modules and the attention mechanisms. Similarly, the goal information is embedded for use by the attention mechanisms of the architecture. Rather than use a conventional neural network with language goals concatenated to state-data as input to the network [56], we leverage a soft-modular architecture based upon empirical experimentation discussed in Sec. V.c.

The behavior modules are a set of successive parallel linear layers that learn to respond to task information to produce actions. Crucially, these layers only receive the state embedding, and are therefore encouraged to learn general representations and behaviors. Every module passes a hidden representation onto all successive modules, which receive a weighted sum of outputs from the previous layer. The output of the modules is weighted by the attention mechanisms of the architecture. Between each set of modules in the architecture, all representations are re-weighted by an attention mechanism before being passed on to the next set of modules. The agent thus learns a set of general-purpose modules that act over input states, and recombines these modules according to attention mechanisms activated by goals.

### B. Goal-label Prediction

To enhance LANCON-LEARN's ability to reason about the similarities and differences among language specifications and to cluster related commands, we incorporate an auxiliary supervised learning task into our RL setup. We obtain the multi-modal embedding by combining the state and goal embeddings, which is passed to a new network head that predicts which goal the agent is currently targeting. The probability of the current goal, Pr[$g$], is 1 if $g$ is the agent's current goal, and 0 otherwise. This process is visualized in Fig. 2, where $L_{AUX}$ is the cross-entropy loss for our auxiliary task, as shown in Eq. 1, where $s_t$ is the current environment state, $\phi$ and $\theta$ are the language model and policy parameters, respectively, and $\hat{g}$ is the goal-prediction embedding.

$$L_{AUX}(\theta, \phi) = -\text{Pr}[g] \log(\text{Pr}[g|\hat{g}, s_t, \mathcal{L}_{\phi}(g), \pi_{\theta}]) \quad (1)$$

Empirically, we find that the auxiliary task improves language-grounding for RL agents but is unnecessary for IL agents, likely due to the relative density and informativeness of the corrective feedback signal in IL. In RL, this additional task helps learn more robust embeddings for downstream tasks.

During training, both the language model, $\mathcal{L}$, and the policy, $\pi$, may be updated according to both the SAC objective and the auxiliary task. We treat the weighting between the objectives as a hyperparameter. We note that the SAC objective is contingent on both $\mathcal{L}_{\phi}$ and $\pi_{\theta}$, as the policy depends on language embeddings over the goal, $g$, produced by $\mathcal{L}_{\phi}$, as shown in Eq. 2.

$$J_{\pi,\mathcal{L}}(\theta, \phi) = \mathbb{E}_{a_t \sim \pi_{\theta}}[\mathbb{E}_{s_t, g_t \sim \mathcal{D}}[\alpha \log(\pi_{\theta}(a_t|s_t, \mathcal{L}_{\phi}(g_t))) - Q_{\omega}(s_t, \mathcal{L}_{\phi}(g_t), a_t)]] \quad (2)$$

Every $\lambda$ steps in the environment, where $\lambda$ is a hyperparameter, our multi-task robot updates according to the SAC and auxiliary objectives jointly. The complete gradient is a

2377-3766 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on January 06,2022 at 17:51:53 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2021.3139667, IEEE Robotics and Automation Letters



**Algorithm 1** Multi-Task Training Loop

1: **Initialize:** Policy $\pi_\theta$, Language model $\mathcal{L}_\phi$
Replay Buffer $\beta \leftarrow \varnothing$, Step index $t \leftarrow 0$
2: **Given:** Labeled language dataset $D_L$
3: **while** Training: **do**
4: Sample new task from the environment and corresponding natural language goal, $g \in G$, from $D_L$
5: $z_g \leftarrow \mathcal{L}_\phi(g)$
6: **for** $i = 1$ to length of episode **do**
7: $s_t \leftarrow$ environment_step
8: $a_i \sim \pi_\theta(\cdot|s_t, z_g)$
9: $\beta \leftarrow \beta + \{s_t, a_t, g\}$
10: **if** i % update_interval == 0 **then**
11: Update $\theta, \phi$ via Eq. 3 using IL or RL with minibatch from $\beta$
12: **end if**
13: **end for**
14: **end while**
15: **Return:** $\pi_\theta$, $\mathcal{L}_\phi$

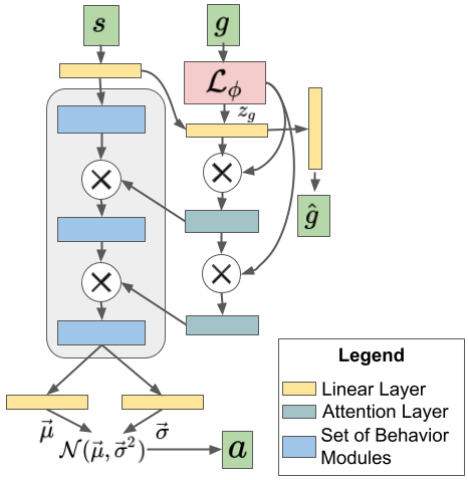



Fig. 2: Our architecture. State data are sent to behavior modules to learn general skills. Language goals are embedded and sent to attention mechanisms that re-weight data between modules to recombine skills into goal-specific behaviors. The model is trained via either IL or RL.

weighted combination of gradients from Eq.s 1 and 2 with respect to both sets of parameters, as in Eq. 3.

$$\nabla_{\phi,\theta} = w_1 \nabla_\theta J_{\pi,\mathcal{L}}(\theta,\phi) + w_2 \nabla_\theta L_{AUX}(\theta,\phi) + w_3 \nabla_\phi J_{\pi,\mathcal{L}}(\theta,\phi) + w_4 \nabla_\phi L_{AUX}(\theta,\phi) \quad (3)$$

### C. Multi-task Training Algorithm

To make use of language specifications as goals in LANCON-LEARN, we begin by collecting a set of language commands that the agent will use. In our work, we gather five sequences for each task. The LANCON-LEARN learning process is presented in Algorithm 1. At the beginning of each episode, the agent receives a new goal, $g$, from the environment (Line 4). The goal, $g$, is then passed to the agents language model, $\mathcal{L}_\phi$, producing a goal embedding, $z_g$, which the agent receives alongside state information, $s$, for the duration of the episode (Line 5). By using $z_g$ rather than $g$ directly, learning updates are now conditioned on the policy, $\pi_\theta$, and the language model, $\mathcal{L}_\phi$. As output, the agent produces the mean, $\mu$, and standard deviation, $\sigma$ of a normal distribution, one for each dimension in which the end-effector can move. Actions, $a \in A$, consist of changes in end-effector position, which are drawn from $\mathcal{N}(\mu, \sigma^2)$ during training. In evaluation mode, the agent directly uses the predicted mean $\mu$.

At pre-defined update intervals, we update the parameters of the agent's policy, $\theta$ and language model, $\phi$. We employ either DAgger [54] (using a trained agent as the oracle labeler) for IL, or SAC [55] for RL via the environment's reward functions. We note that any policy learning algorithm or framework could be used in our setup simply by changing the update mechanism in Line 11 of Algorithm 1.

## V. EXPERIMENTS AND RESULTS

We empirically demonstrate LANCON-LEARN on a virtual and real, physical robot. For this demonstration, we use a Rethink Robotics Sawyer, a seven degree-of-freedom robot arm. For all tasks, the agent is trained in simulation and the real-robot execution demonstrates robust plans that work across shifting dynamics (from simulation to the real world).

We first test LANCON-LEARN on unseen tasks (i.e., tasks that were not used during training). These test tasks require the robot to manipulate novel objects not encountered during training. Training language agents versus conventional one-hot goal encoding agents across ten tasks, we test how well the agents are able to generalize their learned skills to novel tasks with no additional training (i.e. zero shot transfer), using IL and multi-task RL. Finally, we conduct a qualitative investigation of the underlying goal embedding space using t-SNE projection of language commands [57].

### A. Baselines

We compare the efficacy of language embeddings as goal representations across multiple conditions, allowing us to compare common approaches from prior work and serving as an ablation study for our approach.

- LANCON-LEARN *(Ours)* – Goal sequences are embedded using GloVe [51] and then condensed with a bi-directional LSTM. The first and final hidden states of the LSTM are concatenated and given as a goal embedding.
- *Ours\Auxiliary* – Ours without the auxiliary goal-prediction task. We omit this comparison for IL, as the auxiliary task had no effect in the IL setup.
- *Ours\GloVe* – Ours with random rather than pre-trained word embeddings.
- *Ours\LSTM* – Goal sequences are embedded as the average GloVe embedding across words in the goal specification, losing word ordering information.
- *BERT*– Goal sequences are passed through a BERT [53] model that has been pre-trained for sequence classification[1], and the goal is the hidden state for $\langle CLS \rangle$.
- *One-Hot*– Goals are provided as one-hot encodings.
- *Random*– Goals are differentiable, randomly-initialized embeddings. Here, we benchmark against real-valued vectors with no a priori meaning [16].

[1] https://huggingface.co/bert-base-uncased

2377-3766 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on January 06,2022 at 17:51:53 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2021.3139667, IEEE Robotics and Automation Letters



| | Goal Specification | Set of Training Tasks | Push Puck With Obstacle | Topdown Button Press With Obstacle | Handle Push | Soccer | Coffee Push | Coffee Button |
|---|---|---|---|---|---|---|---|---|
| IL (DAgger) | **Ours** | **66%** | **42%** | 90% | **74%** | **70%** | **50%** | 100% |
| | Ours\LSTM | 62.6% | 24% | 82% | 62% | 52% | 26% | 100% |
| | Ours\GloVe | 53.9% | 0% | **100%** | 46% | 8% | 0% | 100% |
| | BERT | 41.9% | 0% | **100%** | 20% | 0% | 0% | 100% |
| | Random | 49.3% | 0% | 44% | 8% | 14% | 10% | 100% |
| | One-Hot | 58.9% | 0% | 8% | 52% | 14% | 30% | 100% |
| RL (SAC) | **Ours** | **38.1%** | **39%** | **100%** | 1% | **34%** | **21%** | **100%** |
| | Ours\Auxiliary | 16% | 0% | 40% | **40%** | 0% | 0% | **100%** |
| | Ours\LSTM | 36.2% | 11% | 57% | 36% | 2% | 8% | 92% |
| | Ours\GloVe | 28.3% | 4% | 16% | 14% | 0% | 0% | **100%** |
| | BERT | 35.2% | 12% | 81% | 0% | 16% | 5% | 91% |
| | Random | 35.4% | 0% | 68% | 5% | 0% | 0% | **100%** |
| | One-Hot | 36.3% | 4% | 25% | 0% | 0% | 20% | 40% |

TABLE I: Success rates on training and unseen (zero-shot) testing tasks in Meta-World training with IL or RL.

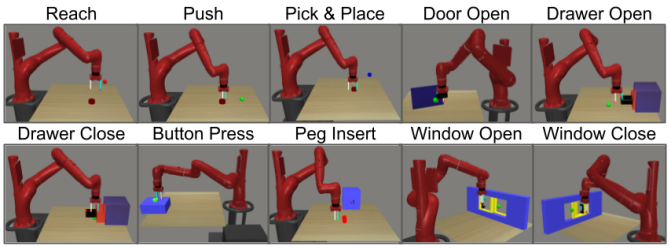



Fig. 3: The 10 tasks in our Meta-World training setup. Initial positions and goal locations (colored spheres) are randomized.

### B. Domains

For our simulated domain, we employ Meta-World [15], a multi-task and meta-learning domain that offers a wide array of robot manipulation task distributions performed by a simulated Sawyer robot. In this domain, we evaluate all agents on their ability to learn a single multi-task policy for ten training tasks in the "Multi-Task 10" setup (Fig. 3). Importantly, there is not a high degree of overlap in these tasks (i.e., unique skills must be learned for each task), forcing our agents to learn diverse skills and to ground diverse language commands. For a complete list of tasks, we refer the reader to [15]. We leverage an extension of Meta-World[2] that expands the state space to include position, velocity, and orientation for all objects in the domain (padded to a fixed-size regardless of task), and provides simplified reward functions for each task. The action space in Meta-World is a 4D continuous vector representing changes in $\mathbb{R}^3$ position and a gripper status control. Success metrics are defined for each task in terms of the difference between task-dependent object positions and the goal position. We train all agents for 15000 episodes across ten tasks, cycling to a new task each episode, repeating each experiment five times with different random seeds. Both initial and goal positions are randomized for each task, making even one task in our work a multi-task learning problem by prior definitions [13].

### C. Architecture Choice

Using the soft-modular network architecture described in Sec. IV, we obtained training task performance of 38.1%, 36.3%, and 35.4% success rates for *Ours*, *One-Hot*, and *Random* agents, respectively. In contrast, a standard deep network with a concatenated goal-state input achieved 0%, 8%, and 0%, respectively. As such, we proceed with soft-modular architectures (Sec. IV).

[2] https://github.com/hartikainen/metaworld/tree/reward-tweaks-rebase

### D. Zero-Shot Transfer Results

We present success rates for all IL and RL agents in Table I, where we compare both training task success rates as well as a selection of unseen tasks. Unseen tasks were selected as a set tasks that have skill-overlap with training tasks, meaning we could expect the agent to achieve more than a 0% success rate (e.g., the agent learns to push an object in training, and so we expect it can push a ball in the "soccer" task).

Our unseen task results show that LANCON-LEARN provides a significant boost to out-of-distribution task success for multi-task learning agents. When applied to unseen tasks, LANCON-LEARN provides a clear advantage compared to conventional goal-specification techniques, achieving up to or over 200% improvement on tasks such as "Push Puck with Obstacle" or "Soccer". This finding reinforces the value of language specification for high-level goals. Rather than simply passing a task index with no context or learning a goal-specific embedding that cannot generalize to unseen objectives, language specifications provide a useful prior and help to prime multi-task robots for task-execution on new, unseen tasks. We observe that the agent with the auxiliary task is as good or better than the agent without the auxiliary task on all but one of the six zero-shot transfer tasks.

While we observe that the *Random* and *One-Hot* agents find occasional success in button-press tasks, these agents regularly fail at adaptions of the "push" task. On closer inspection, we find that these agents treat all unseen tasks as button-press tasks, showing no understanding of the true task objective. LANCON-LEARN, on the other hand, shows significantly higher performance on an array of skills.

When tasks are specified with useful prior knowledge (e.g., using GloVe or BERT) the agent succeeds across all unseen tasks, adapting to the task based on the language command the agent is provided, with our approach completing up to 74% of novel push tasks and 100% of novel button-press tasks. We explore these results further in Sec. VI.

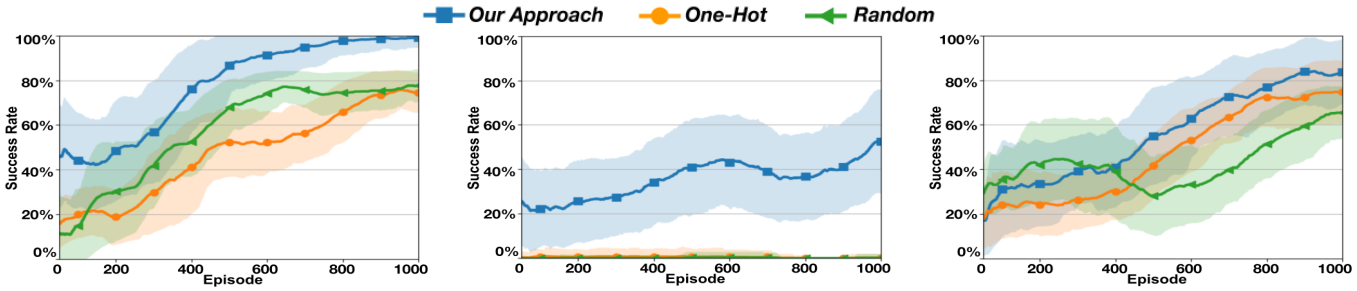
### E. Meta-World Transfer Results

Having shown LANCON-LEARN outperforms baselines in zero-shot skill transfer, we next seek to demonstrate the benefits of our approach for transferring knowledge to rapidly acquire novel skills. We compare LANCON-LEARN to the baseline one-hot encoding and random embedding approaches using three tasks not seen at all during training: "Sweep Into," "Push

2377-3766 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on January 06,2022 at 17:51:53 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2021.3139667, IEEE Robotics and Automation Letters






(a) "Sweep Into" (b) "Push Puck With Obstacle" (c) "Handle Push"

Fig. 4: Task success rate over five runs of 1000 episodes in the knowledge-transfer tasks. LANCON-LEARN leverages prior knowledge about tasks to enable the agent to transfer known skills and quickly solve the tasks.

Puck With Obstacle," and "Handle Push" from Meta-World, as our testbed, as these all involve skills learned in training (i.e., pushing) but applied in new ways. Using such tasks, we can observe how quickly our agents transfer known skills to novel situations. For these tasks, we will only consider RL, as IL provides an over-simplification of the task and washes out any benefits of prior-knowledge. We define the reward function as a sparse reward of 10 on task success and a step penalty of $-0.01$. In this experiment, we compare, *Ours*, *One-Hot*, and *Random* goal-specification approaches.

We present results from the knowledge transfer experiment in Fig. 4, where we plot task success rates over episodes of training. LANCON-LEARN provides a substantial advantage to a multi-task learning agent, as our approach starts off with a high success rate and is then able to convincingly outperform both baselines across the three tasks. We observe that our agent often begins with over twice the success rate of prior approaches, up to 50% compared to 20%, and always achieves a higher final success rate, up to 100%. While a one-hot encoding or random embedding provides no useful information, prior knowledge from language embeddings guides the agent in transferring known skills to new tasks.

### F. Analysis of Embedding Space

To perform our investigation, we compute the t-SNE [57] projections of all 250 language command embeddings in our dataset both before and after training with LANCON-LEARN, passing the language commands through the language encoder and the first layer of the policy network. Each task in the Meta-World domain has five associated language commands in our work, and there are 50 total tasks (giving us the total 250 points). All points are colored according to their associated task. Fig. 5 shows these t-SNE projections.

In these figures, we can see many closely related skills (e.g., "Button", "Reach", and "Push") are clustered together. Despite being initialized somewhat randomly and with little overlap, the learned LSTM encoder has produced embeddings that are highly similar after training. In a similar vein, we see that some commands that started off nearby, such as the "Drawer Open" and "Door Open" commands, have been pushed farther apart, reflecting the inherent differences between the skills required to complete each task. While the underlying language for these two commands is very similar (e.g. "Open the door" and "Open the drawer"), the skills required to complete the tasks are very different. The language model has therefore learned that, despite their similar language, the embeddings for these tasks should be highly separated. We also conducted an analysis for the *One-Hot* model, revealing a randomly-distributed plot with no meaningful clusters around skills or task objectives before or after training, shown in Fig. 6.

### G. Deployment to Sawyer

In our real robot testbed, we deploy the learned policies to a Sawyer robot. To apply trajectories across the domain gap, we calibrate the simulator's dynamics to the real Sawyer and then playback simulated trajectories on the real robot. This setup allows us to deploy our agents to the real world despite their actions being calibrated for the simulator's dynamics.

With our framework in place, we deploy the trajectories from a learned agent to the real Sawyer robot across the "Close Drawer," "Topdown Button Press," and "Close Window" tasks. Each episode, the object and goal positions move to new locations, and we randomly sample from the five different language sequences for each episode. A visual sequence of our robot deployment is shown in Fig. 7, and videos of our experiments are included in the supplementary.

## VI. DISCUSSION

LANCON-LEARN improves human-robot interaction for untrained end-users and multi-task learning agents' abilities to generalize to unseen tasks using known skills with both IL and RL methods in manipulation settings. We observe that language embeddings as goals for learning agents help to achieve this end being both more intuitive to specify (e.g. saying "Push the button" instead of a providing a one-hot encoding, $[0, 0, ..., 1, 0]$) and encapsulating prior knowledge within the goal embedding itself.

In our experiments, we trained several different multi-task agents using different goal specification techniques using IL and RL. We observe that out-of-distribution test task performance was significantly improved by using language embeddings as goals. LANCON-LEARN agents were able to map high-level concepts in language down to task execution, transferring known skills such as "Push" from training domains into novel testing domains with up to 100% success.

Further, prior knowledge from language embeddings enables the rapid acquisition of new skills while conventional one-hot

2377-3766 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on January 06,2022 at 17:51:53 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2021.3139667, IEEE Robotics and Automation Letters



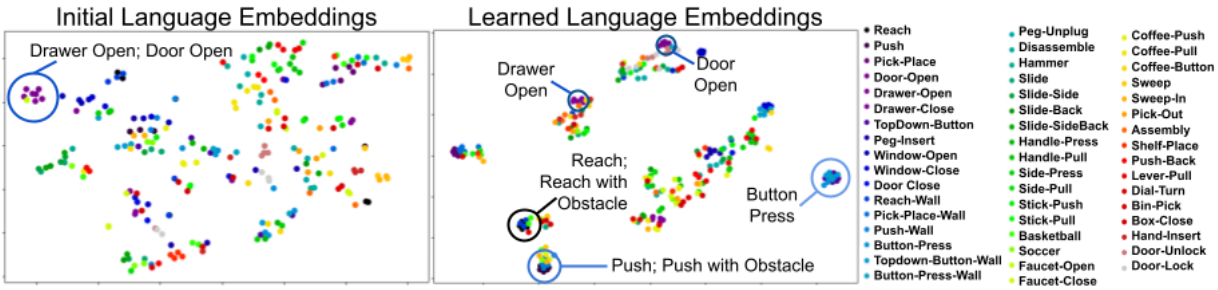



Fig. 5: t-SNE of language commands before and after training the LANCON-LEARN agent.

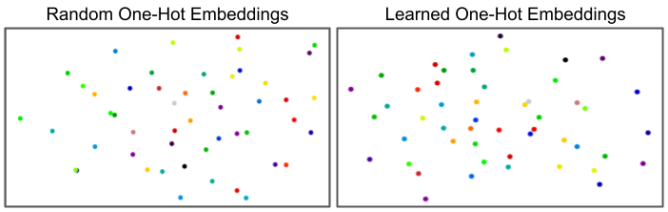



Fig. 6: Initial and final t-SNE for a *One-Hot* model

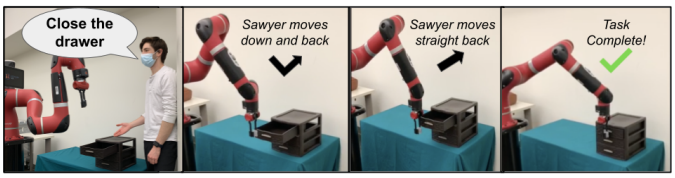



Fig. 7: We show that our agent is capable of producing trajectories that can be applied to a real Sawyer robot, here completing the "Close Drawer" task.

encoding approaches must re-learn skills from scratch. One-hot encodings are unable to provide any notion of similarity between distant indices in a large vector of zeros, thereby showing much worse generalization to unseen tasks. As we see in Table I and Fig. 4, language-conditioned agents yield higher performance than baselines (i.e., one-hot and random agents) across all tasks we tested.

When considering language goals, we observe that language-based agents benefit from some form of pre-training. *Ours\GloVe* is routinely worse than other language approaches on novel tasks (Table I), suggesting an under-tuned language model. Unseen words, such as "ball" in the "Soccer" task, are completely unknown to the *Ours\GloVe* model, and so the model provide a sequence embedding over untrained (i.e. random) word embeddings. Using GloVe [51] embeddings circumvents this out-of-distribution issue.

In our knowledge-transfer experiments, we show that LANCON-LEARN provides a useful warm-start to multi-task RL agents by fine-tuning a pre-trained agent on three new, related tasks. With a sparse reward function contingent on success, we observe that a multi-task agent that can leverage a language prior is able to outperform a conventional one-hot encoding agent. Our agent, using language sequences as high level goals, solves the tasks in just a few hundred episodes, exceeding baseline performance. This result highlights an exciting prospect for multi-task learning in robotics, as we achieve significant improvements to knowledge-transfer by providing language commands. While our current deployment requires sufficient similarity between a given language command for a transfer task and those for training tasks, future work may explore language-enabled approaches to compositionality of learned skills, providing avenues for more generally reusable robot skills in more varied domains.

## VII. CONCLUSION

We presented LANCON-LEARN for learning multi-task manipulation skills using high-level language commands for efficient generalization to new skills. While prior research has explored one-hot goals, such encodings lack meaningful prior knowledge and do not generalize to unseen tasks. We demonstrated that language enables useful goal specifications to multi-task learning agents across a variety language representations. From mean GloVe embeddings to BERT hidden-states, our experiments demonstrate that language representations of goals provide useful task-specific knowledge. LANCON-LEARN enables up to 100% success rates on unseen tasks (e.g., "Sweep Into") and affords rapid acquisition of new skills by relating known skills and grounded-language to new objectives.

## REFERENCES

[1] T.-L. Vuong, D.-V. Nguyen, T.-L. Nguyen, C.-M. Bui, H.-D. Kieu, V.-C. Ta, Q.-L. Tran, and T.-H. Le, "Sharing experience in multitask reinforcement learning," in *Proc. International Joint Conference on Artificial Intelligence*, 2019, pp. 3642–3648.

[2] R. Yang, H. Xu, Y. WU, and X. Wang, "Multi-Task Reinforcement Learning with Soft Modularization," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 4767–4777.

[3] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn, "Multi-task reinforcement learning without interference," *Optimization Foundations for Reinforcement Learning Workshop at NeurIPS*, 2019.

[4] R. Paleja, A. Silva, L. Chen, and M. Gombolay, "Interpretable and personalized apprenticeship scheduling: Learning interpretable scheduling policies from heterogeneous user demonstrations," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[5] A. Achille, M. Lam, R. Tewari, A. Ravichandran, S. Maji, C. C. Fowlkes, S. Soatto, and P. Perona, "Task2vec: Task embedding for meta-learning," in *Proc. IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6430–6439.

[6] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray *et al.*, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.

[7] S. Vandenhende, S. Georgoulis, B. De Brabandere, and L. Van Gool, "Branched multi-task networks: deciding what layers to share," *arXiv preprint arXiv:1904.02920*, 2019.

[8] P. Anderson, A. Shrivastava, J. Truong, A. Majumdar, D. Parikh, D. Batra, and S. Lee, "Sim-to-real transfer for vision-and-language navigation," in *Proc. of Conference on Robot Learning*, 2020.

[9] Y. Jiang, S. Gu, K. Murphy, and C. Finn, "Language as an abstraction for hierarchical deep reinforcement learning," *arXiv preprint arXiv:1906.07343*, 2019.

2377-3766 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on January 06,2022 at 17:51:53 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2021.3139667, IEEE Robotics and Automation Letters




[10] X. Wang, Q. Huang, A. Celikyilmaz, J. Gao, D. Shen, Y.-F. Wang, W. Y. Wang, and L. Zhang, "Reinforced Cross-Modal Matching and Self-Supervised Imitation Learning for Vision-Language Navigation," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jun. 2019, pp. 6622–6631.

[11] D. Misra, A. Bennett, V. Blukis, E. Niklasson, M. Shatkhin, and Y. Artzi, "Mapping instructions to actions in 3d environments with visual goal prediction," *arXiv preprint arXiv:1809.00786*, 2018.

[12] V. Blukis, R. A. Knepper, and Y. Artzi, "Few-shot object grounding and mapping for natural language robot instruction following," *arXiv preprint arXiv:2011.07384*, 2020.

[13] A. Li, L. Pinto, and P. Abbeel, "Generalized hindsight for reinforcement learning," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 7754–7767.

[14] S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza, "Power to the people: The role of humans in interactive machine learning," *Ai Magazine*, vol. 35, no. 4, pp. 105–120, 2014.

[15] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine, "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning," in *Proc. Conference on Robot Learning*, 2019.

[16] Y. Burda, H. Edwards, A. Storkey, and O. Klimov, "Exploration by random network distillation," *arXiv preprint arXiv:1810.12894*, 2018.

[17] K. Barnard and D. Forsyth, "Learning the semantics of words and pictures," in *Proc. International Conference on Computer Vision*, vol. 2. IEEE, 2001, pp. 408–415.

[18] S. Branavan, D. Silver, and R. Barzilay, "Learning to win by reading manuals in a monte-carlo framework," *Journal of Artificial Intelligence Research*, vol. 43, pp. 661–704, 2012.

[19] D. Chen and R. Mooney, "Learning to interpret natural language navigation instructions from observations," in *Proc. AAAI Conference on Artificial Intelligence*, vol. 25, no. 1, 2011.

[20] S. Tellex, T. Kollar, S. Dickerson, M. Walter, A. Banerjee, S. Teller, and N. Roy, "Understanding natural language commands for robotic navigation and mobile manipulation," in *Proc. AAAI Conference on Artificial Intelligence*, vol. 25, no. 1, 2011.

[21] S. Stepputtis, J. Campbell, M. Phielipp, S. Lee, C. Baral, and H. Ben Amor, "Language-conditioned imitation learning for robot manipulation tasks," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 13 139–13 150.

[22] J. Abramson, A. Ahuja, A. Brussee, F. Carnevale, M. Cassin, S. Clark, A. Dudzik, P. Georgiev, A. Guy, T. Harley *et al.*, "Imitating interactive intelligence," *arXiv preprint arXiv:2012.05672*, 2020.

[23] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, "Neural module networks," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 39–48.

[24] N. Gopalan, E. Rosen, G. Konidaris, and S. Tellex, "Simultaneously learning transferable symbols and language groundings from perceptual data for instruction following," *Robotics: Science and Systems*, 2020.

[25] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, L. Zettlemoyer, and D. Fox, "ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[26] P. Goyal, S. Niekum, and R. J. Mooney, "Pixl2r: Guiding reinforcement learning using natural language by mapping pixels to rewards," in *Proc. Conference on Robot Learning*, 2020.

[27] C. Lynch and P. Sermanet, "Language conditioned imitation learning over unstructured data," *Robotics: Science and Systems*, 2021.

[28] S. Nair, E. Mitchell, K. Chen, B. Ichter, S. Savarese, and C. Finn, "Learning language-conditioned robot behavior from offline data and crowd-sourced annotation," *arXiv preprint arXiv:2109.01115*, 2021.

[29] S. Sodhani, A. Zhang, and J. Pineau, "Multi-task reinforcement learning with context-based representations," in *Proc. International Conference on Machine Learning*, vol. 139, 18–24 Jul 2021, pp. 9767–9779.

[30] J. Luketina, N. Nardelli, G. Farquhar, J. Foerster, J. Andreas, E. Grefenstette, S. Whiteson, and T. Rocktäschel, "A Survey of Reinforcement Learning Informed by Natural Language," *arXiv:1906.03926 [cs, stat]*.

[31] A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra, "Embodied question answering," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1–10.

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

[33] K. Nguyen, D. Dey, C. Brockett, and B. Dolan, "Vision-Based Navigation With Language-Based Assistance via Imitation Learning With Indirect Intervention," in *Conference on Computer Vision and Pattern Recognition*. IEEE, Jun. 2019, pp. 12 519–12 529.

[34] A. Suhr, C. Yan, J. Schluger, S. Yu, H. Khader, M. Mouallem, I. Zhang, and Y. Artzi, "Executing instructions in situated collaborative interactions," *arXiv preprint arXiv:1910.03655*, 2019.

[35] L. Zhou and K. Small, "Inverse reinforcement learning with natural language goals," *Proc. AAAI Conference on Artificial Intelligence*, vol. 35, no. 12, pp. 11 116–11 124, May 2021.

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

[37] Y. Yang, J. P. Inala, O. Bastani, Y. Pu, A. Solar-Lezama, and M. Rinard, "Program synthesis guided reinforcement learning," *arXiv preprint arXiv:2102.11137*, 2021.

[38] M. Denil, S. G. Colmenarejo, S. Cabi, D. Saxton, and N. de Freitas, "Programmable agents," 2017.

[39] B. Gangopadhyay, H. Soora, and P. Dasgupta, "Hierarchical program-triggered reinforcement learning agents for automated driving," *arXiv preprint arXiv:2103.13861*, 2021.

[40] J. Oh, S. Singh, H. Lee, and P. Kohli, "Zero-shot task generalization with multi-task deep reinforcement learning," in *Proc. International Conference on Machine Learning*, 2017, pp. 2661–2670.

[41] E. A. Brooks, J. Rajendran, R. L. Lewis, and S. Singh, "Reinforcement learning of implicit and explicit control flow in instructions," 2021.

[42] S. Tellex, N. Gopalan, H. Kress-Gazit, and C. Matuszek, "Robots that use language," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 25–55, 2020.

[43] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 70, 06–11 Aug 2017, pp. 1126–1135.

[44] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," *arXiv preprint arXiv:1803.02999*, 2018.

[45] L. Chen, R. Paleja, M. Ghuy, and M. Gombolay, "Joint goal and strategy inference across heterogeneous demonstrators via reward network distillation," in *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 659–668. [Online]. Available: https://doi.org/10.1145/3319502.3374791

[46] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks," *Proc. of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[47] B. Hayes and B. Scassellati, "Autonomously constructing hierarchical task networks for planning and human-robot collaboration," in *International Conference on Robotics and Automation*. IEEE, 2016.

[48] C. Rosenbaum, I. Cases, M. Riemer, and T. Klinger, "Routing networks and the challenges of modular and compositional computation," *arXiv preprint arXiv:1904.12774*, 2019.

[49] J. Andreas, D. Klein, and S. Levine, "Modular multitask reinforcement learning with policy sketches," in *Proc. International Conference on Machine Learning*, 2017, pp. 166–175.

[50] P. Goyal, S. Niekum, and R. J. Mooney, "Using natural language for reward shaping in reinforcement learning," *arXiv preprint arXiv:1903.02020*, 2019.

[51] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543.

[52] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

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

[54] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proc. International Conference on Artificial Intelligence and Statistics*, 2011, pp. 627–635.

[55] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. International Conference on Machine Learning*, 2018.

[56] S. Branavan, L. S. Zettlemoyer, and R. Barzilay, "Reading between the lines: Learning to map high-level instructions to commands." Association for Computational Linguistics, 2010.

[57] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of machine learning research*, vol. 9, no. 11, 2008.


2377-3766 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on January 06,2022 at 17:51:53 UTC from IEEE Xplore. Restrictions apply.