EC^2 : Emergent Communication for Embodied Control

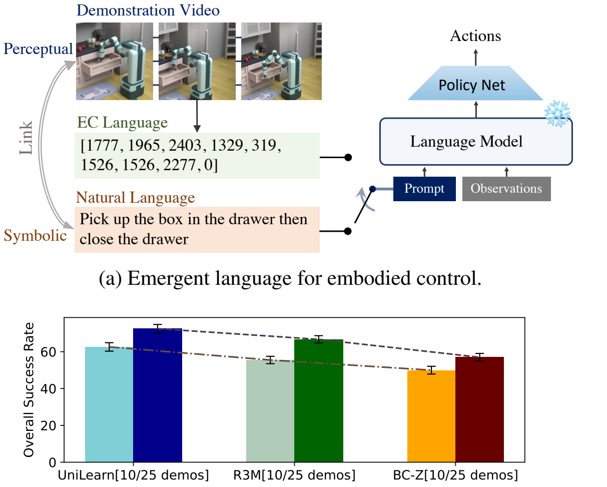
Yao Mu[1], Shunyu Yao [2], Mingyu Ding [1], Ping Luo[1], Chuang Gan[3,4]
[1]The University of Hong Kong [2] Princeton University [3]UMass Amherst, [4]MIT-IBM Watson AI Lab
ymu, myding, pluo@cs.hku.hk shunyuy@princeton.edu ganchuang@csail.mit.edu

Abstract

Embodied control requires agents to leverage multi-modal pre-training to quickly learn how to act in new environments, where video demonstrations contain visual and motion details needed for low-level perception and control, and language instructions support generalization with abstract, symbolic structures. While recent approaches apply contrastive learning to force alignment between the two modalities, we hypothesize better modeling their complementary differences can lead to more holistic representations for downstream adaption. To this end, we propose ***E****mergent* ***C****ommunication for* ***E****mbodied* ***C****ontrol (EC^2), a novel scheme to pre-train video-language representations for few-shot embodied control. The key idea is to learn an unsupervised "language" of videos via emergent communication, which bridges the semantics of video details and structures of natural language. We learn embodied representations of video trajectories, emergent language, and natural language using a language model, which is then used to finetune a lightweight policy network for downstream control. Through extensive experiments in Metaworld and Franka Kitchen embodied benchmarks,* ***EC^2*** *is shown to consistently outperform previous contrastive learning methods for both videos and texts as task inputs. Further ablations confirm the importance of the emergent language, which is beneficial for both video and language learning, and significantly superior to using pre-trained video captions. We also present a quantitative and qualitative analysis of the emergent language and discuss future directions toward better understanding and leveraging emergent communication in embodied tasks.*

1. Introduction

We study the problem of few-shot embodied control, where an embodied agent needs to execute language instructions or follow video demonstrations given only a few examples in a new environment. Such a capability of fast adaption is key to practical robotic deployment, as it is non-scalable and expensive to collect extensive action-labeled trajectories for each new application scenario. Instead, agents need to leverage pre-trained video and language representations to quickly learn how to act, and generalize across high-level concepts (e.g. object and verb types) as well as low-level visual features.



(a) Emergent language for embodied control.

(b) Performance comparison on overall success rate across all tasks.

Figure 1. Overview of EC^2. The key idea of EC^2 is to build the link between perceptual grounding and symbolic concept via emergent communication (EC) language. We learn to extract embodied representation via language model utilizing emergent language or natural language as prompt and visual observation as input for downstream embodied control tasks. The actions are generated by a lightweight policy network containing a few MLP [51] layers that map the learned frozen embodied representation into action space. Extensive experiments show that EC^2 outperforms existing methods in Metaworld [60] and Franka Kitchen [17] benchmarks.

How can we pre-train multi-modal representations of videos and texts for downstream embodied control in different domains? Inspired by the success of image-text models such as CLIP [47], recent work has investigated contrastive representation learning approaches using paired video-language data, with application toward robotic manipulation and control tasks [24, 38]. However, simply aligning video and language pairs in the embedding space omits the difference between these two modalities: while videos contain more visual and motion details, language abstracts away key structures underlying the task. For example, a video of opening a door contains details of approaching the handle, pressing the handle, pulling the door open, and so on, while the corresponding language description could be as simple as "open a door" — more detailed language descriptions such as "press down the handle" is usually not available in existing datasets and very expensive to manually label. But for downstream control purposes, a video and its abstract language description could present complementary benefits — while the former is expressive in details needed for low-level control, the latter provides the structure for generalization across domains, tasks, and skills. Leveraging such modality differences presents opportunities to better incorporate language and videos for embodied representation learning, marrying their respective benefits rather than forcing them to be aligned.

One perspective to view the video-language difference is through the emergence of language [39, 55]: language derives meaning from its use [57]. It is a communication tool humans develop to collaboratively solve embodied tasks, thus abstracting perception experiences like videos into symbolic concepts. Emergent communication in machine learning [9, 26, 28, 30, 56] aims to similarly learn an "emergent language" of perception in an unsupervised and domain-adaptive way: as Figure 2(a) shows, a speaker and a listener network play a referential game, where a speaker maps some visual stimuli (e.g. image, video) into a message of discrete tokens, and a listener uses the message to choose the speaker reference out of detractors. By jointly optimizing for game success, the communication protocol emerges structural and semantic properties resembling language, yet its discrete tokens might convey more fine-grained concepts than natural language useful for distinguishing stimuli. Thus, an emergent language can act as a bridge between natural language and videos, featuring a structure similar to the former while preserving the semantics of the latter.

Inspired by recent work that links emergent and natural language [58], we propose **E**mergent **C**ommunication for **E**mbodied **C**ontrol (EC^2), a three-phase scheme to learn embodied representations for downstream few-shot control (Figure 1(a-c)). First, we learn an emergent language of demonstration videos without using any language labels, which could potentially capture domain-specific concepts not directly available in paired natural language captions. Second, we learn an embodied representation by using a language model to predict trajectory segments conditional on trajectory contexts and language annotations, both natural and emergent. Rather than forcing video and language representations to be aligned, such a sequence modeling approach learns to represent natural language (for compositional generalization) and emergent language (for low-level perception and control) jointly with video trajectories, where the emergent language serves as a bridge between language and videos. Third, we train a lightweight policy network on top of the embodied representation to few-shot adapt to downstream embodied control.

We demonstrate that emergent language can help embodied control tasks and enable data-efficient few-shot downstream imitation learning via extensive experimental results across two existing benchmark simulation environments (Franka-Kitchen [18] and MetaWorld [60]). EC^2 outperforms existing methods and achieves state-of-the-art performance on both language and video instruction following tasks. Our study also shows that although emergent language and natural language are utilized in a parallel manner in the pre-training phase, the word embeddings of natural language can still be predicted by linear regression models from emergent language. We further investigate the impact of emergent language and natural language correlation on the performance of downstream tasks and show that the performance in downstream tasks is best when the learned EC contains more information than natural language, i.e., when the accuracy of predicting natural language from emergent language is higher than that of predicting emergent language from natural language. Overall, on the basis of these results, we believe that EC^2 has great potential to become a promising embodied control framework.

To sum up, our work makes the following contributions. (1) we build an embodied representation pre-training framework with emergent communication (EC) via a language model, which incorporates both the abstract, compositional structure of language and visuals with low-level motion details. (2) we develop a few-shot embodied control system, which transfers the pre-trained language model to downstream tasks as a frozen module and quickly adapts lightweight policy networks with a few expert data to generate actions. (3) we demonstrate that the learned representation with EC can benefit few-shot embodied control tasks and extensive experiments show that EC^2 outperforms existing methods in Metaworld [60] and Franka Kitchen [17] benchmarks.

2. Related Work

2.1. Emergent Communication

The recent advances in NLP [7, 14, 49] benefits from huge text corpora and are striving to find statistical regularities on large amounts of data via big models like GPT [7, 48, 49] and BERT [14]. This narrow focus was leaving aside language emergence and language grounding (in some other modalities, e.g., sight, with more complex inputs than text). In contrast, human language acquisition [13] starts with a cumulative series of interactions with other people [3, 8] grounded in the physical and social world [1, 19, 54] and does not rely on passive and complex corpora like

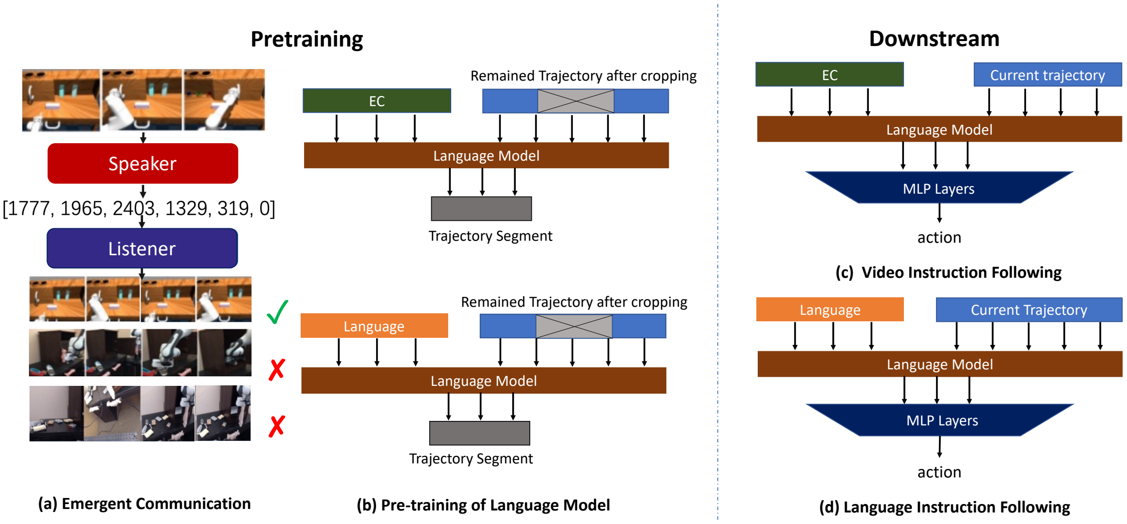


Figure 2. The overall framework of EC^2. **Pre-training:** **(a)** We first pre-train a speaker (🟥) and listener (🟪) via emergent communication game for emergent language generation and a language model (🟫) to extract embodied representations. **(b)** We generate the corresponding emergent language (🟩) for each demonstration video in the multi-modal dataset. Predicting the cropped part of a latent trajectory is used as an auxiliary task to help pre-training the language model. The language model takes emergent language or natural language as prompt and the cropped trajectory (⊠) as input and aims to predict the cropped part of the trajectory (◼). The loss for the prediction using emergent language as prompt and natural language as prompt is calculated separately. **Few-shot downstream policy learning:** In downstream imitation learning tasks, both the EC speaker and the pre-trained language model(🟫) are frozen. The language model extracts effective embodied features with current trajectory (🟦 contains current state and historical information) as input. **(c)** For tasks with language instruction, we use natural language as the prompt, and **(d)** for tasks with video instruction, we use emergent language as the prompt. The output features of the language model are mapped into actions to mimic expert behavior by a few simple MLP [51] layers (▼).

Wikipedia. Emergent Communication (EC) is one promising direction toward this motivation, where a communication protocol is shaped through multi-agent interactions with perceptual grounding like a human [43]. As a typical example, the referential image game involves a speaker creating a sequence of discrete tokens based on an input image, and a listener tasked with selecting the correct input from a group of distractors using the message. Both networks are optimized jointly using success signals from the game [58]. By studying games like this type, researchers are interested in the emergence of desirable properties resembling natural language, such as game success generalization and compositionality, which holds great promises towards more functional [56, 57] and generalizable [27] language agents.

2.2. Embodied Control under Instruction

Embodied control has achieved much success in learning grasping and pick-place tasks from low-dimensional state [2, 4, 10, 25, 36, 42, 52]. Deep learning makes embodied control directly learn from high dimensional observations becoming feasible [46, 50, 61]. Recent advances aim to build a learning-based embodied control system with deep neural networks that are flexibly conditioned on either a demonstration video [5, 59] of a human or a language instruction [35, 53] via multi-modal representation learning. With the guidance of instruction, the embodied agent aims to generalize to new scenes [44], novel objects [5, 16, 22, 59, 62], novel object configurations [40], and novel goal configurations [12, 15, 20]. For instance, BC-Z [24] has developed a flexible imitation learning system that can learn from demonstrations and interventions, based on various forms of task information, such as pre-trained embeddings of natural language or human performance videos. R3M [38] pre-trains visual representations on diverse human video-language data by encouraging alignment between the two modalities. However, since videos often convey more detailed information than words, this framework may lose valuable teaching information that can help robots perform complex tasks. In this work, we aim to build an instruction-conditioned embodied control framework without forcing alignment on video and language and consider the video demonstrations and language instructions as "parallel sentences" during the pre-training process to learn useful information for downstream tasks.

3. Method

As shown in Figure 2, we first pre-train a speaker and listener via an emergent communication game for emergent language generation and a language model to extract embodied representations by predicting the cropped part of a latent trajectory. The language model takes the emergent language or natural language as prompt and the masked trajectory as input and aims to predict the masked part of the trajectory separately. Then, we transfer the learned speaker and language model into downstream embodied control tasks as frozen modules. The output of the language model extracts embodied representation from instruction prompt and current observations and is mapped into actions by the downstream policy network, which contains a few task-specific MLP [51] layers.

3.1. Problem Statement

The goal for embodied control is to learn a policy π_B conditioned on the instruction, either language instruction or video instruction, that reproduces the expert behavior on the desired task specified by the instruction. Usually, the demonstrations of experts are presented in the form of state-action trajectories, with each pair indicating the action to take at the state being visited. Let τ_E denote a trajectory sampled from expert policy $\pi_E(s|c)$: $\tau_E = \left[(s_0, a_0), (s_1, a_1), \ldots, (s_n, a_n)\right]$. The instruction is encoded into the latent vector c. To learn the behavior policy $\pi_B(s|c)$, the demonstrated actions are usually utilized as the target label for each state, and the mapping $a \in \pi_B(s|c)$ from states to actions is learned in a supervised manner with given instructions.

3.2. Emergent Communication Language Generation for demonstration video

Our idea is to automatically generate synthetic language that aligns with video demonstration through emergent communications, which connect perceptual, language, and action for pre-training. As shown in Figure 2(c), we consider a typical speaker-listener referential game [29, 30, 33] on a set of N video features $\mathcal{D}_I = \{I_1, \cdots, I_N\}$. At each training step, the speaker takes one video feature I_i and generates a discrete message $M_i \in [V]^T$ using the Gumbel-Softmax trick [23], where V is the vocabulary size, and T is the sequence length limit. For simplicity, denote $m = M_i$ so that $m_t = M_{i,t}$ denotes the t-th token of the message M_i. The generated discrete message is defined as "emergent language", and the process of emergent language generation by the speaker can be formulated as

$$\begin{aligned} \mathbf{hs}_0 &= I_i, \quad \mathbf{hs}_t = \mathrm{GPT}_{\text{spk}}\left(m_{t-1}, \mathbf{hs}_{t-1}\right)\ (t>0), \\ m_0 &= [\text{CLS}], \quad m_t = \mathrm{Softmax}\left(\mathrm{MLP}_{\text{spk}}(\mathbf{hs}_t)\right)\ (t>0). \end{aligned} \tag{1}$$

Here $\mathbf{hs}_t$ denotes speaker hidden states. Next, the listener takes the message m, and tries to guess the right video I_i out of a set of K confounding videos $C_i = \{I_{j_1}, \cdots, I_{j_K}\} \subset \mathcal{D}_I - \{I_i\}$. Note that different from the EC games on images, in order to learn temporal information, except for irrelevant videos, we also construct several additional candidates in C_i by performing temporal augmentation on the original video, such as reverse order and random disorder. The listener also uses a GRU [11] layer to turn the message m into a hidden vector $\mathbf{hl}_T$

$$\mathbf{hl}_0 = \mathbf{0}, \quad \mathbf{hl}_t = \mathrm{GPT}_{\text{lsn}}\left(m_t, \mathbf{hl}_{t-1}\right)\ (t>0). \tag{2}$$

Based on $\mathbf{hl}_T$, the listener assigns a score for each candidate video based on inverse square error [31], then selects the image by Softmax sampling across the scores. The speaker and listener are jointly optimized by minimizing the cross-entropy loss of video selection:

$$\begin{aligned} \text{score}(\mathbf{I}) &= \left\|\mathbf{hl}_T - \mathrm{MLP}_{\text{lsn}}(\mathbf{I})\right\|_2^{-2}, \\ p(\text{ guess } = \mathbf{I}) &= \mathrm{softmax}(\text{score}(\mathbf{I})) \quad \left(\mathbf{I} \in \{\mathbf{I}_i\} \cup C_i\right), \\ \mathcal{L}_{EC} &= -\mathbb{E}_{\mathbf{I}_i, C_i} \mathbb{E}_{\mathbf{M}_i} \log p\left(\text{ guess } = \mathbf{I}_i\right). \end{aligned} \tag{3}$$

The speaker can be employed to generate emergent language $\mathcal{D}_M = \{M_1, \cdots, M_N\}$ based on input videos.

3.3. Pre-training of Language Model with Emergent Language

We conduct a trajectory completion task to pre-train the GPT-like language model without action labels, as shown in Figure 2(a)(b). Firstly, we random sample a sequence of observations $\hat{o} = \{o_1, o_2, \ldots, o_N\}$ stored in the dataset and map it into latent trajectory τ_{whole} by encoder g_θ.

$$\tau_{\text{whole}} = g_\theta(\hat{o}) \tag{4}$$

Then we crop a random segment τ_{seg} from the whole latent trajectory τ_{whole} and the remained trajectory is denoted as τ_{rem}. The language model f_ϕ takes τ_{rem} as input and uses either emergent language e generated by the speaker or natural language l as a prompt to predict the cropped segment τ_{seg}. The language model is optimized by jointly minimizing L_{EC} and L_{Lang}

$$L_{\text{EC}} = \left(\tau_{\text{seg}} - f_\theta(e, \tau_{\text{rem}})\right)^2 \tag{5}$$

$$L_{\text{Lang}} = \left(\tau_{\text{seg}} - f_\theta(l, \tau_{\text{rem}})\right)^2 \tag{6}$$

The loss for the prediction using emergent language as prompt and natural language as prompt is calculated separately.

EC^2 uses the same language model for both emergent language generation and trajectory completion pretext task for framework thriftiness, and both the $L_{\text{EC-pre}}$ and $L_{\text{EC-gen}}$ are used to update the language model jointly. To ensure that the speaker's gradient is not truncated, we use an MLP [51] layer to map from the logits of the word distribution to a 512-dimensional latent vector instead of selecting a word embedding in the dictionary according to the token. For natural language, we use the Byte-Pair Encoding (BPE) tokenizer [6] to encode it into tokens, which are widely used by OpenAI for tokenization when pre-training the GPT model, including GPT [48], GPT-2 [49], RoBERTa [34], and BART [32]. We also learn a corresponding dictionary of word embedding. Although the emergent language and natural language are used separately, they can learn correlated information via the pretext trajectory completion task.

3.4. Few-shot Policy Learning for Downstream Embodied control

To transfer the pre-trained language model into the downstream imitation learning tasks, the language model $f_\theta(\cdot,\cdot)$ take the current trajectory $\tau_{cur} = \{s_{t-T},\dots,s_{t-1},s_t\}$ and the emergent language e or natural language instruction l as prompt, then output the predicted features m_t. Finally, the output m_t of the language model is mapped to specific action a_t by task-specific MLP layers π_ψ. The language model f_θ is frozen during downstream imitation learning. The action under learned behavior policy is sampled by $\hat{a}_t \sim \pi_\psi(a_t|m_t)$ and the expert demonstration action a_t are used as the label for imitation learning. For the video instruction following task, we generate emergent language by the speaker and use it as the prompt of the language model. For language instruction following tasks, the natural language is tokenized by the BPE tokenizer, and select the corresponding word embedding in the dictionary as the prompt as language model via the inferred token. With a trajectory with length H, the task-specific MLP layers are optimized by minimizing $\mathcal{L}_{\text{IL}}(\psi)$.

$$\mathcal{L}_{\text{IL}}(\psi) = E_{\tau\sim\mathcal{B}}\left[-\frac{1}{H}\sum_{t}^{t+H-1}\log\pi_\psi(a_t|m_t)\right] \tag{7}$$

The MLP [51] layers are implemented with hidden sizes [256, 256] with batch normalization [21] and are trained with a learning rate of 0.001 and a batch size of 32 for 20000 iterations.

4. Experiments

4.1. Pre-training Dataset

We use the LOReL [37] real robot dataset, which is collected by Franka Emika Panda mounted over an IKEA desk. The LOReL dataset contains 3000 episodes (150000 frames) of reinforcement learning that trains policies for different behaviors on the IKEA desk using online RL. The language instruction is annotated by leveraging crowdsourcing, specifically Amazon Mechanical Turk. Human annotators are asked to describe the behavior if any, that the robot is doing and to phrase it as a command without any pre-specified template. The LOReL dataset contains 6000 annotations, two per episode, containing a total of 1699 unique instructions. The dataset contains different videos with the same language instruction to ensure diversity. We removed the episodes where annotators reported the robot as inactive or its actions unclear, making them incomprehensible.

4.2. Downstream Testing Environments

We utilize two standard robotic manipulation benchmarks, namely MetaWorld [60] and the Franka Kitchen environment [17], as our downstream testing environments. It is worth noting that these testing environments were not used during the training phase of EC^2. In MetaWorld, the agent is presented with a series of tasks, including assembling a ring on a peg, picking and placing a block between bins, pushing a button, opening a drawer, and hammering a nail. In Franka Kitchen, the agent aims to learn how to slide open the right door, open the left door, turn on the light, turn the stove top knob, and open the microwave. As shown in Figure 3, both environments offer image observations to the agent, as well as proprioceptive data, which includes the end-effector pose and joint positions. These data are concatenated with the encoded vision observations for each task, providing the agent with a comprehensive understanding of the current task. All tasks involve random environment variation, either by varying the position of the target object in MetaWorld or the positioning of the desk in Franka Kitchen. Additionally, we consider three views for each environment to evaluate the robustness of representations across viewpoints. Finally, for each environment, we also consider three demo sizes used in downstream embodied control tasks: $[5, 10, 25]$ in MetaWorld and Franka Kitchen.

4.3. Experimental Setup

For embodied control tasks, we take R3M and BC-Z as the key baseline. **R3M** [38] learns the state-of-the-art embodied representations pre-trained on diverse paired human video-language data with time-contrastive learning, which encourages states closer in time to be closer in embedding space and video-language alignment to encourage the embeddings to capture semantically relevant features. For sufficient comparison, we implement R3M with both the language instruction-conditioned version and video instruction-conditioned version, respectively, based on their official implementation. For embodied control with natural language instruction, we use the language encoder of the pre-trained model officially released by their author team to encode the natural language. For embodied control with demonstration video instruction, we use the officially released vision encoder to extract features of every frame in the video and map them into the instruction representation by one MLP layer. **BC-Z** [24] jointly learns the alignment between video representation and language representation together with policy learning across multiple embodied control tasks. Our evaluation methodology is inspired by [38, 41]. We focus on evaluating the embodied representation of instruction and observation for downstream policy learning with behavior cloning. We parameterize the downstream policy π as a two-layer MLP preceded by a

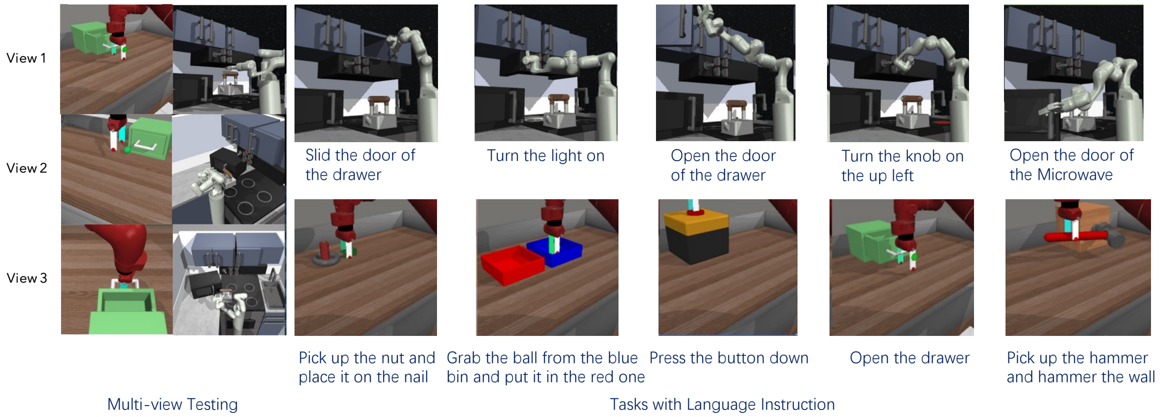

View 1
View 2
View 3
Slid the door of the drawer
Turn the light on
Open the door of the drawer
Turn the knob on the up left
Open the door of the Microwave
Pick up the nut and place it on the nail
Grab the ball from the blue bin and put it in the red one
Press the button down
Open the drawer
Pick up the hammer and hammer the wall
Multi-view Testing
Tasks with Language Instruction

Figure 3. **Evaluation Environments**: We consider a set of manipulation tasks including 5 tasks with a Sawyer from MetaWorld [60], 5 tasks from a Franka operating over a Kitchen [17]. We provide specific language instruction labels for every task and 25 demos per task for evaluation. We show an example image and corresponding language instruction for each task shown above. **Evaluation Viewpoints**: To robustly measure the efficacy of different visual representations, we consider three camera viewpoints per environment.

Benchmark	Num. Demos	Instruction	EC^2 (ours)	R3M [38]	BC-Z [24]
MetaWorld	10	Lang	**72.4** ± 2.4%	69.2 ± 2.0%	63.4 ± 2.1%
		Video	**76.0** ± 2.1% (**+3.6**)	71.8 ± 1.8% (+2.6)	64.8 ± 2.0% (+1.4)
	25	Lang	**82.6** ± 2.0%	77.4 ± 2.4%	74.1 ± 2.2%
		Video	**85.0** ± 2.2% (**+2.4**)	79.0 ± 2.1% (+1.6)	75.2 ± 1.0%(+1.1)
Franka Kitchen	10	Lang	**48.4** ± 2.5%	41.8 ± 2.5%	34.2 ± 2.3%
		Video	**53.6** ± 2.5%(**+5.2**)	45.7 ± 2.4% (+3.9)	37.5 ± 2.2% (+3.3)
	25	Lang	**59.8** ± 2.5%	56.0 ± 2.3%	38.6 ± 2.4%
		Video	**63.2** ± 2.6% (**+3.4**)	58.7 ± 2.0% (+2.7)	40.0 ± 2.0% (+1.4)

Table 1. Performance comparison on Success rate of embodied control with language/video instruction across different demo size. We mark the performance difference between complete the task with natural language instruction and video instruction as blue.

BatchNorm at the input. We train the agent for 20,000 steps, evaluate it online in the environment every 1000 steps, and report the best success rate achieved. For each task, we run 5 seeds of behavior cloning. The final success rate reported for one method on one task is the average over the 5 seeds × 3 camera viewpoints × 3 demo sizes, for a total of 45 runs (see more details in Appendix).

4.4. Results

We aim to answer 5 questions to get insight into embodied AI community: 1) Can emergent language help embodied control tasks with natural language instruction? 2) Can emergent language help the embodied control tasks with video instruction? 3) Can the joint training of emergent language and natural language in GPT-Like network architecture help each other? 4) Can emergent language provide more detailed and effective guidelines than video captions? 5) What is the relationship between model size, control performance, and the correlation between emergent language and natural language?

To answer the first two questions, we study if EC^2 enables more data-efficient policy learning with both natural language instructions and video instructions.

1) *Can emergent language help the embodied control tasks with natural language?*

We measure the success rate of downstream few-shot policy learning with different pre-training methods. In Franka Kitchen and MetaWorld, we only use no more than 25 demos to perform downstream few-shot policy learning and report the performance with both 10 and 25 demos separately in Table 1. Learning from scratch is struggled to perform well when working with small amounts of data since it can be challenging for the model to identify meaningful patterns or relationships with such limited data. While the methods with a pre-trained model, like EC^2, R3M, and BC-Z performs better than those learning from scratch. Through our study, as shown in Table 1, EC^2 exceeds existing methods with natural language instruction, which indicates that emergent language can help the embodied control tasks to improve the performance with natural language instruction. Across all the evaluations, EC^2 is overall able to learn these embodied tasks in an extremely low data regime (only 10 demos) with ≈60.4% overall success rate, despite never seeing any data from the target environments in training the representation, while R3M only achieves ≈55.5% overall success rate.

Benchmark	Num. Demos	Language Instruction		Video Instruction		
		EC^2	EC^2 (-EC)	EC^2	EC^2 (-NL)	EC^2 (+Cap,-EC)
MetaWorld	10	**72.4** $\pm$ 2.4%	69.0 $\pm$ 2.0%	**76.0** $\pm$ 2.1%	72.6 $\pm$ 2.4%	73.0 $\pm$ 2.2%
	25	**82.6** $\pm$ 2.0%	78.0 $\pm$ 2.4%	**85.0** $\pm$ 2.5%	82.0 $\pm$ 2.0%	73.2 $\pm$ 2.5%
Franka Kitchen	10	**48.4** $\pm$ 2.5%	43.0 $\pm$ 3.3%	**53.6** $\pm$ 2.5%	49.0 $\pm$ 2.5%	48.6 $\pm$ 2.4%
	25	**59.8** $\pm$ 2.5%	54.0 $\pm$ 2.6%	**63.2** $\pm$ 2.6%	61.3 $\pm$ 2.4%	49.0 $\pm$ 2.3%

Table 2. Ablation study on embodied control with language instruction and video instruction.

Model Size	#layer	#head	#embded
Base	8	16	512
Mini	6	6	192
Micro	4	4	128

Table 3. Hyper-parameters for different model sizes of EC^2. All the models are implemented based on GPT [48]-like network architecture.

Natural Language	Emergent language
"Open the right **drawer**"	[262, 847, 490, 536, **108**, **500**, 941, 838, 749, 444]
"Grasp in the middle **drawer**"	[356, 69, **108**, 259, 72, 616, **500**, 896, 623, 433]
"**Reach** down for **stapler**"	[204, 897, 285, 509, **701**, 545, **538**, 729, 902, 195]
"Pick up the **stapler** in the **drawer**"	[535, 67, **108**, 661, 84, **701**, 776, **500**, 377, 454]
"**Reach** to the plug"	[982, 691, 72, **538**, 97, 665, 148, 369, 102, 1003]
"**Reach** to the left **drawer**"	[786, **538**, 601, 874, 848, **500**, 315, **108**, 72, 914]

Table 4. Qualitative examples of emergent language and corresponding natural language. The word and tokens that have the same semantic meaning are marked with the same color.

2) *Can emergent language help the embodied control tasks with video instruction?*

As shown in Table1, EC^2 shows an obvious advantage over the exciting methods and achieves the best success rate ($\approx$64.8% with 10 demos and $\approx$74.1% with 25demos) with video instruction. As marked in blue in Table 1, EC^2 has better performance with video instruction than with language instruction, which indicates that EC^2 learns more effective information provided by a video demonstration. In contrast, R3M and BC-Z, which use contrast learning or regression to force the two modalities of language and video aligned, have little difference in performance under language instruction and video instruction.

We also investigate the task performance of EC^2 and baselines with different amounts of demos in policy learning. In Figure 4, we plot the average success rate of each method across each demo size. We observe that the performance improvement from EC^2 is consistent, outperforming the baselines across every environment and demo size.

Ablation Study: Also, in the data-efficient imitation learning setting, we ablate the different components of EC^2 training to answer question 3 and 4 and check if each module is indispensable.

3) *Can the joint training of emergent language and natural language with GPT-Like network help each other?*

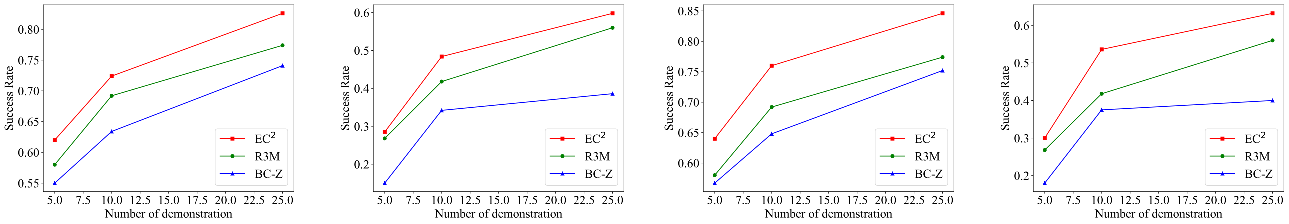
We conduct ablation studies to compare EC^2 and EC^2 (-EC) in language instruction following tasks and compare EC^2 and EC^2 (-NL) in video instruction following tasks. EC^2 (-EC) does not use emergent language to pre-train the language model under EC^2 framework, which predicts the masked trajectories with only the language instruction (human-labeled caption). EC^2 (-NL) does not use natural language to pre-train the language model under EC^2 framework, which predicts the masked trajectories with only the emergent language. As shown in the left-hand column in Table 2, without the help of emergent language, the performance decreased significantly, and EC^2 (-EC) shows no obvious advantages to the other methods that learn pre-trained models by video-language pre-training. Similarly, as shown in the right-hand column in Table 2, EC^2 also outperforms EC^2 (-NL), which demonstrates that the language pre-training helps the video instruction task with emergent language. Thus, the above ablation analysis shows that the joint training of emergent language and natural language can benefit each other.

4) *Can emergent language provide more detailed and effective guidelines than video captions?*

In EC^2 (+Cap, -EC), we replace the emergent language in EC^2 with the language caption given by a MARN [45] as a data augmentation, which is a pre-trained video caption model. As shown in Table 2, EC^2 shows an obvious advantage to EC^2 (+Cap, -NL), which demonstrates that emergent language provides more detailed and effective guidelines than video captions.

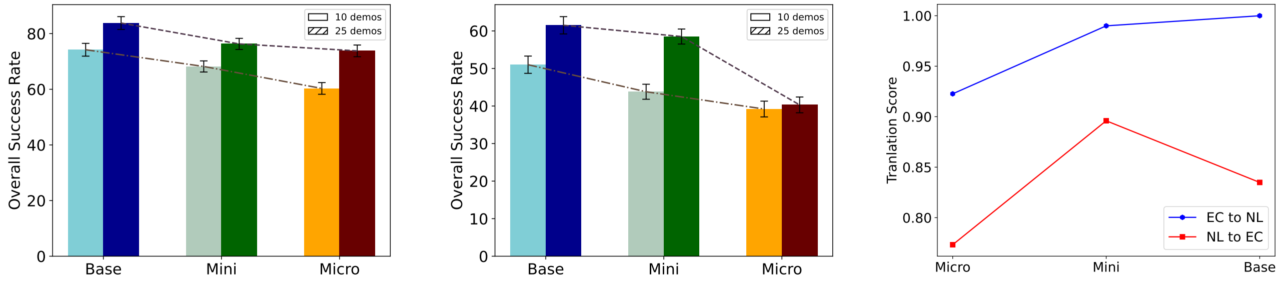
5) *What is the relationship between model size, control performance, and the correlation between emergent language and natural language?*

We first tested the overall average success rate of EC^2 with different model sizes (listed in Table 3) in the downstream tasks (Franka Kitchen and MetaWorld). As shown in Figure 5a and Figure 5b, the larger the size of the model, the more expressive it is and thus the better its performance. Then, we perform a linear correlation analysis on the word embeddings of the emergent language and the word embeddings of the matched natural language. We use the square of the coefficient of determination, denoted R^2 as the measure of predicting the performance of linear regression, which reflects the proportion of the variation in the dependent variable that is predictable from the independent variable. Consider a linear regression model $f(\cdot)$ which predicts y with



(a) MetaWorld[***language*** instruction] (b) Kitchen[***language*** instruction] (c) MetaWorld[***video*** instruction] (d) Kitchen[***video*** instruction]

Figure 4. Performance over different demo sizes used in downstream imitation learning. We report the success rate of EC^2 and baseline across different demo sizes. We see that the performance improvement from EC^2 is consistent across all demo sizes.



(a) Ablation on model size in MetaWorld (b) Ablation on model size in Franka Kitchen (c) Translation score across different model sizes

Figure 5. Ablation on model size and its relationship with the correlation of emergent language and Natural language. We show the ablation results on model size with 10 demos and 25 demos in downstream policy learning separately.

the input x, the coefficient of determination R is defined as

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{\sum_i (y_i - f(x_i))^2}{\sum_i (y_i - \bar{y})^2} \tag{8}$$

We evaluat the R^2 values of two linear regressions. One predicts natural language word embeddings from emergent language, and the other predicts emergent language word embeddings from natural language. As shown in Figure 5c, the R^2 of predicting natural language embeddings from the emergent language increase along with the model becoming larger, which corresponds to better performance. In contrast, the R^2 of predicting emergent language embeddings from the natural language is non-monotonous, along with the model size becoming larger. The reason is that emergent language captures more information that natural language does not contain when the model is large enough and has promising performance. In short, EC^2 has the best performance in embodied control tasks when emergent language contains more information than natural language, the performance is median when emergent language and language information are nearly equal, and the performance is worst when emergent language contains less information than language information. As shown in Table 4, we also provide qualitative examples of emergent language and its corresponding natural language to show that emergent language can capture key concepts from video. We find that the tokens "108" and "500" correspond to "drawer" in natural language and token "701" corresponds to "stapler." Except for nouns, emergent language can also get the concept of verbs, for example, token "538" corresponds to the verb "reach" in natural language. The current manual check analysis of emergent language is just a preliminary attempt, and more understanding can be promising for future work.

5. Conclusions and Discussions

This paper aims to build the link between perceptual grounding and symbolic concepts by emergent communication (EC) language for embodied control. To this end, we develop a novel **E**mbodied **C**ontrol framework with the help of **E**mergent **C**ommunication language (EC^2), which pre-trains a language model via masked trajectory complement pretext task condition on emergent language or natural language. The pre-trained language model is utilized to extract embodied representation from instructions and observations and is used as a frozen module for downstream data-efficient policy learning. Extensive experiments show that **EC^2** outperforms existing methods in Metaworld and Franka kitchen benchmarks. We believe that EC^2 will serve as a solid step toward the general decision-making model. **Limitations and future works:** more understanding about emergent language, training EC^2 on a larger and more diverse dataset, and applications on more practical setup can be promising directions of future works.

Acknowledgement. Ping Luo is partially supported by the National Key R&D Program of China No.2022ZD0161000 and the General Research Fund of HK No.17200622. Shunyu Yao was supported in part by the National Science Foundation under Grant No. 2107048.

References

[1] Muhannad Alomari, Paul Duckworth, David Hogg, and Anthony Cohn. Natural language acquisition and grounding for embodied robotic systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017. 2

[2] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009. 3

[3] Michelle E Barton. *Input and interaction in language acquisition*. Cambridge University Press, 1994. 2

[4] Aude Billard, Yann Epars, Sylvain Calinon, Stefan Schaal, and Gordon Cheng. Discovering optimal imitation strategies. *Robotics and autonomous systems*, 47(2-3):69–77, 2004. 3

[5] Alessandro Bonardi, Stephen James, and Andrew J Davison. Learning one-shot imitation from humans without humans. *IEEE Robotics and Automation Letters*, 5(2):3533–3539, 2020. 3

[6] Kaj Bostrom and Greg Durrett. Byte pair encoding is suboptimal for language model pretraining. *arXiv preprint arXiv:2004.03720*, 2020. 4

[7] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 2

[8] Jerome Bruner. The role of interaction formats in language acquisition. In *Language and social situations*, pages 31–46. Springer, 1985. 2

[9] Angelo Cangelosi and Domenico Parisi. Computer simulation: A new scientific approach to the study of language evolution. In *Simulating the evolution of language*, pages 3–28. Springer, 2002. 2

[10] Rawichote Chalodhorn, David B Grimes, Keith Grochow, and Rajesh PN Rao. Learning to walk through imitation. In *IJCAI*, volume 7, pages 2084–2090, 2007. 3

[11] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 4

[12] Sudeep Dasari and Abhinav Gupta. Transformers for one-shot visual imitation. *Conference on Robot Learning (CoRL)*, 2020. 3

[13] Jill G De Villiers, Jill De Villiers, Peter A De Villiers, and Peter A DeVilliers. *Language acquisition*. Harvard University Press, 1978. 2

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2

[15] Yan Duan, Marcin Andrychowicz, Bradly C Stadie, Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. One-shot imitation learning. *arXiv preprint arXiv:1703.07326*, 2017. 3

[16] Chelsea Finn, Tianhe Yu, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot visual imitation learning via meta-learning. In *Conference on Robot Learning*, pages 357–368. PMLR, 2017. 3

[17] Abhishek Gupta, Vikash Kumar, Corey Lynch, Sergey Levine, and Karol Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. *arXiv preprint arXiv:1910.11956*, 2019. 1, 2, 5, 6

[18] Abhishek Gupta, Vikash Kumar, Corey Lynch, Sergey Levine, and Karol Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. *Conference on Robot Learning*, 2019. 2

[19] Stevan Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3):335–346, 1990. 2

[20] De-An Huang, Suraj Nair, Danfei Xu, Yuke Zhu, Animesh Garg, Li Fei-Fei, Silvio Savarese, and Juan Carlos Niebles. Neural task graphs: Generalizing to unseen tasks from a single video demonstration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8565–8574, 2019. 3

[21] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015. 5

[22] Stephen James, Michael Bloesch, and Andrew J Davison. Task-embedded control networks for few-shot imitation learning. In *Conference on Robot Learning*, pages 783–795. PMLR, 2018. 3

[23] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016. 4

[24] 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 *Conference on Robot Learning*, pages 991–1002. PMLR, 2022. 1, 3, 5, 6

[25] S.M. Khansari-Zadeh and Aude Billard. Learning stable non-linear dynamical systems with gaussian mixture models. *IEEE Transaction on Robotics*, 27(5):943–957, 2011. 3

[26] Simon Kirby. Natural language from artificial life. *Artificial life*, 8(2):185–215, 2002. 2

[27] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017. 3

[28] Angeliki Lazaridou and Marco Baroni. Emergent multi-agent communication in the deep learning era. *arXiv preprint arXiv:2006.02419*, 2020. 2

[29] Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. Emergence of linguistic communication from referential games with symbolic and pixel input. *arXiv preprint arXiv:1804.03984*, 2018. 4

[30] Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. Multi-agent cooperation and the emergence of (natural) language. *arXiv preprint arXiv:1612.07182*, 2016. 2, 4

[31] Jason Lee, Kyunghyun Cho, Jason Weston, and Douwe Kiela. Emergent translation in multi-agent communication. *arXiv preprint arXiv:1710.06922*, 2017. 4

[32] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019. 5

[33] Fushan Li and Michael Bowling. Ease-of-teaching and language structure from emergent communication. *Advances in neural information processing systems*, 32, 2019. 4

[34] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 4

[35] Corey Lynch and Pierre Sermanet. Grounding language in play. *arXiv preprint arXiv:2005.07648*, 2020. 3

[36] Katharina Mülling, Jens Kober, Oliver Kroemer, and Jan Peters. Learning to select and generalize striking movements in robot table tennis. *The International Journal of Robotics Research*, 32(3):263–279, 2013. 3

[37] Suraj Nair, Eric Mitchell, Kevin Chen, Silvio Savarese, Chelsea Finn, et al. Learning language-conditioned robot behavior from offline data and crowd-sourced annotation. In *Conference on Robot Learning*, pages 1303–1315. PMLR, 2022. 5

[38] 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. 1, 3, 5, 6

[39] Martin A. Nowak and D C Krakauer. The evolution of language. *Proceedings of the National Academy of Sciences of the United States of America*, 96 14:8028–33, 1999. 2

[40] Tom Le Paine, Sergio Gómez Colmenarejo, Ziyu Wang, Scott Reed, Yusuf Aytar, Tobias Pfaff, Matt W Hoffman, Gabriel Barth-Maron, Serkan Cabi, David Budden, et al. One-shot high-fidelity imitation: Training large-scale deep nets with rl. *arXiv preprint arXiv:1810.05017*, 2018. 3

[41] Simone Parisi, Aravind Rajeswaran, Senthil Purushwalkam, and Abhinav Gupta. The (un)surprising effectiveness of pre-trained vision models for control. *international conference on machine learning*, 2022. 5

[42] Peter Pastor, Heiko Hoffmann, Tamim Asfour, and Stefan Schaal. Learning and generalization of motor skills by learning from demonstration. In *2009 IEEE International Conference on Robotics and Automation*, pages 763–768. IEEE, 2009. 3

[43] Shivansh Patel, Saim Wani, Unnat Jain, Alexander G Schwing, Svetlana Lazebnik, Manolis Savva, and Angel X Chang. Interpretation of emergent communication in heterogeneous collaborative embodied agents. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15953–15963, 2021. 3

[44] Deepak Pathak, Parsa Mahmoudieh, Guanghao Luo, Pulkit Agrawal, Dian Chen, Yide Shentu, Evan Shelhamer, Jitendra Malik, Alexei A Efros, and Trevor Darrell. Zero-shot visual imitation. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 2050–2053, 2018. 3

[45] Wenjie Pei, Jiyuan Zhang, Xiangrong Wang, Lei Ke, Xiaoyong Shen, and Yu-Wing Tai. Memory-attended recurrent network for video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8347–8356, 2019. 7

[46] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. Technical report, Carnegie-Mellon University, 1989. 3

[47] 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. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 1

[48] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. *Technical report, OpenAI*, 2018. 2, 4, 7

[49] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 2, 4

[50] Rouhollah Rahmatizadeh, Pooya Abolghasemi, Ladislau Bölöni, and Sergey Levine. Vision-based multi-task manipulation for inexpensive robots using end-to-end learning from demonstration. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 3758–3765. IEEE, 2018. 3

[51] Hassan Ramchoun, Youssef Ghanou, Mohamed Ettaouil, and Mohammed Amine Janati Idrissi. Multilayer perceptron: Architecture optimization and training. 2016. 1, 3, 4, 5

[52] Stefan Schaal, Jan Peters, Jun Nakanishi, and Auke Ijspeert. Learning movement primitives. In *Robotics research. the eleventh international symposium*, pages 561–572. Springer, 2005. 3

[53] Simon Stepputtis, Joseph Campbell, Mariano Phielipp, Stefan Lee, Chitta Baral, and Heni Ben Amor. Language-conditioned imitation learning for robot manipulation tasks. *arXiv preprint arXiv:2010.12083*, 2020. 3

[54] Paul Vogt. The physical symbol grounding problem. *Cognitive Systems Research*, 3(3):429–457, 2002. 2

[55] Kyle Wagner, James A. Reggia, Juan Uriagereka, and Gerald S. Wilkinson. Progress in the simulation of emergent communication and language. *Adaptive Behavior*, 11:37 – 69, 2003. 2

[56] Kyle Wagner, James A Reggia, Juan Uriagereka, and Gerald S Wilkinson. Progress in the simulation of emergent communication and language. *Adaptive Behavior*, 11(1):37–69, 2003. 2, 3

[57] Ludwig Wittgenstein. Philosophical investigations. *New York: Macmill an Publish*, 1958. 2, 3

[58] Shunyu Yao, Mo Yu, Yang Zhang, Karthik R Narasimhan, Joshua B Tenenbaum, and Chuang Gan. Linking emergent and natural languages via corpus transfer. *arXiv preprint arXiv:2203.13344*, 2022. 2, 3

[59] Tianhe Yu, Chelsea Finn, Annie Xie, Sudeep Dasari, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot im-

itation from observing humans via domain-adaptive meta-learning. *Robotics: Science and Systems*, 2018. 3

[60] 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*, pages 1094–1100. PMLR, 2020. 1, 2, 5, 6

[61] Tianhao Zhang, Zoe McCarthy, Owen Jow, Dennis Lee, Xi Chen, Ken Goldberg, and Pieter Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5628–5635. IEEE, 2018. 3

[62] Allan Zhou, Eric Jang, Daniel Kappler, Alex Herzog, Mohi Khansari, Paul Wohlhart, Yunfei Bai, Mrinal Kalakrishnan, Sergey Levine, and Chelsea Finn. Watch, try, learn: Meta-learning from demonstrations and reward. *International Conference on Learning Representations (ICLR)*, 2020. 3