OCT 13 2020

TEXAS
The University of Texas at Austin

NLP + ACTION

PRESENTERS: PRIYANKA MANDIKAL, BINGYE LI

NLP + Action

- Trends in language understanding
  - Plain Text Corpora → NLP + Vision → NLP + Vision + Action

> 2015 > 2018

no grounding static datasets active perception

Source	An admitting privilege i to carry out a diagnosis
Reference	Le privilège d'admissic d'un hôpital, d'admettr diagnostic ou un traiten
RNNenc-50	Un privilège d'admissi centre médical d'un dia
RNNsearch-50	Un privilège d'admissi centre médical pour eff soins de santé à l'hôpita
Google Translate	Un privilège admettre centre médical pour eff que travailleur de soins

"construction worker in orange safety vest is working on road."

- Language + Embodied AI
- Instruction following in realistic situated environments – egocentric RGB cameras, agent actions

NLP + Action

1. ALFRED, A Benchmark for Interpreting Grounded Instructions for Everyday Tasks, CVPR 2020
   - ❖ Egocentric vision + Natural language instructions → Action sequences for household tasks

2. Grounding Language in Play, arXiv 2020
   - ❖ Teleoperated play + Natural language instructions → Continuous robotic control

OCT 13 2020

TEXAS
The University of Texas at Austin

ALFRED:
A Benchmark for Interpreting Grounded Instructions for Everyday Tasks

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, Dieter Fox

PRESENTER: BINGYE LI

Introduction

ALFRED is a benchmark for learning a mapping from natural language instructions and egocentric vision to sequences of actions for household tasks.

	— Language —		— Virtual Environment —				— Inference —	
	# Human Annotations	Granularity	Visual Quality	Movable Objects	State Changes	Vis. Obs.	Navigation	Interaction
TACoS [43]	17k+	High&Low	Photos	✗	✗	–	–	–
R2R [3]; Touchdown [14]	21k+; 9.3k+	Low	Photos	✗	✗	Ego	Graph	✗
EQA [15]	✗	High	Low	✗	✗	Ego	Discrete	✗
Matterport EQA [55]	✗	High	Photos	✗	✗	Ego	Discrete	✗
IQA [20]	✗	High	High	✗	✓	Ego	Discrete	Discrete
VirtualHome [42]	2.7k+	High&Low	High	✓	✓	3rd Person	✗	Discrete
VSP [58]	✗	High	High	✓	✓	Ego	✗	Discrete
ALFRED	25k+	High&Low	High	✓	✓	Ego	Discrete	Discrete + Mask

- Include both high-level goal and low-level natural language instructions.
- Include object and state interactions.
- Enable discretized, grid-based movement rather than topological graph navigation.
- Require spatially located interaction masks instead of choosing from a set of object classes.

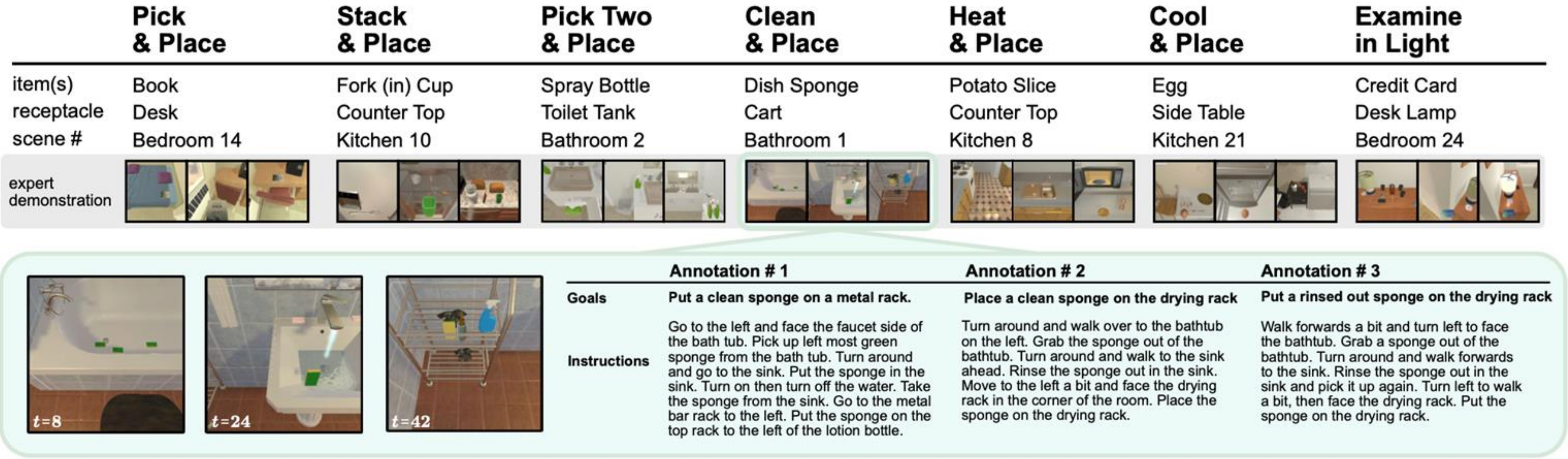
Related Work

- Vision & Language Navigation:
  - Navigation in static environment
  - No object interactions and state changes
- Vision & Language Task Completion
  - Based on simpler block worlds and fully observable scenes
  - AI2-THOR, an interactive 3D environment for visual AI, where AI agents can navigate in the scenes and interact with objects to perform tasks.

Related Work

- Embodied Question Answering
  - Question answering using templated language or static scenes
  - No task completion
- Instruction Alignment
  - Learning visual correspondence from recorded videos
  - Not in an interactive setting
- Robotics Instruction Following
  - Consider different tasks individually
  - Limited to fewer scenes and objects

ALFRED Dataset



ALFRED includes 25,743 English language directives describing 8,055 expert demonstrations averaging 50 steps each, resulting in 428,322 image-action pairs.

Expert demonstrations

- Expert demonstrations are composed of an agent’s egocentric visual observations of the environment, actions taken at each timestep, and ground-truth interaction masks.
- Navigation actions: MoveAhead, RotateRight, RotateLeft, LookUp, and LookDown.
- Manipulation actions: Pickup, Put, Open, Close, ToggleOn, ToggleOff, and Slice.

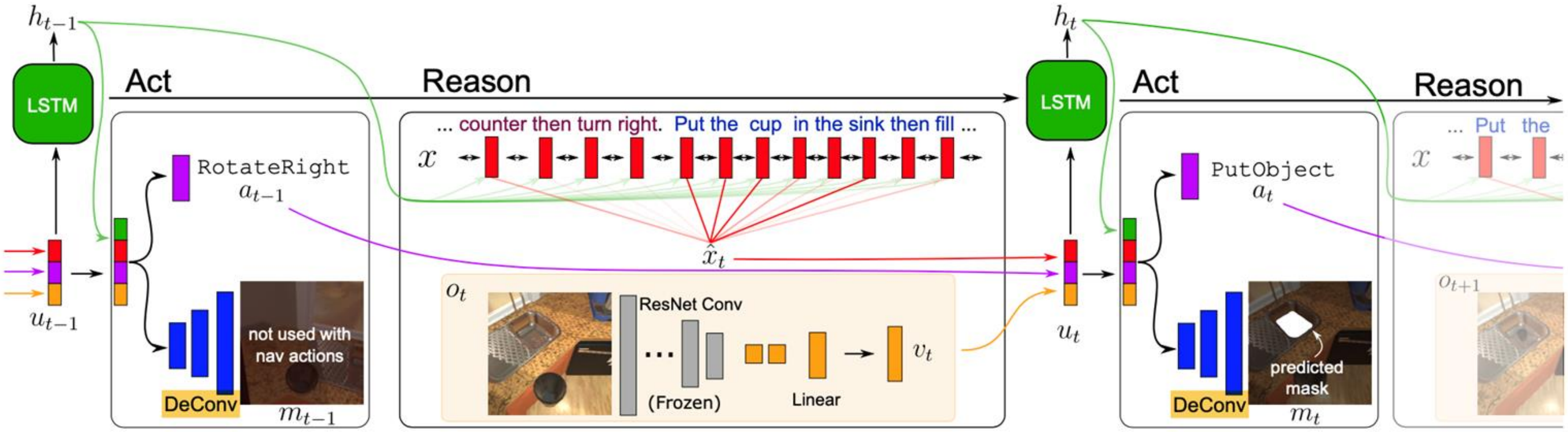
Language directives

- Language directives include a high-level goal together with low-level instructions.
- AMT workers write low-level, step-by-step instructions for each highlighted sub-goal segment.
- Ex: “*Walk to the coffee maker on the right.*”
- They also write a high-level goal that summarizes what the robot should accomplish during the expert demonstration.
- *Ex: “Rinse off a mug and place it in the coffee maker.”*

Sequence-to-Sequence Model

- A bidirectional-LSTM generates a representation of the language input
- A CNN encodes the visual input
- A decoder LSTM infers a sequence of low-level actions while attending over the encoded language
- At each timestep, the model produces the expert action and associated interaction mask for manipulation actions.

Sequence-to-Sequence Model

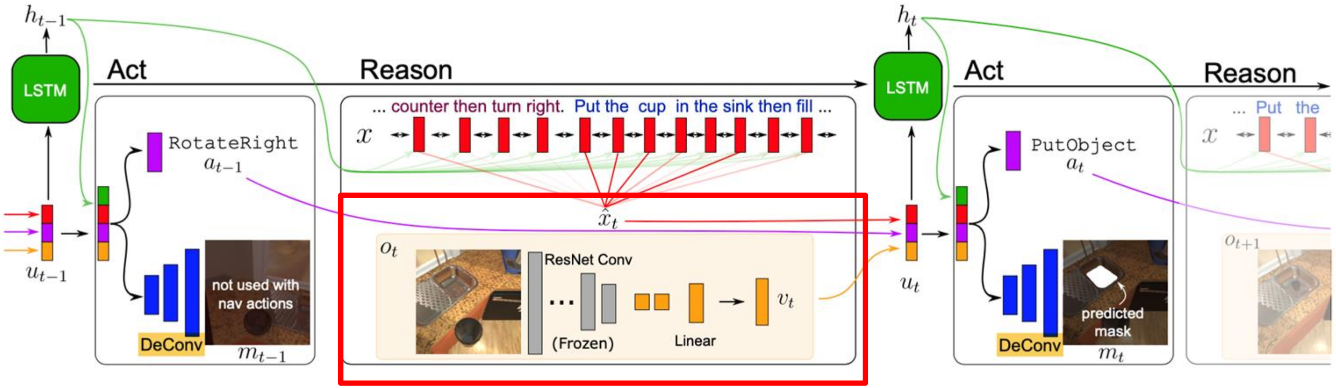


Language encoding

- a natural language goal $\overline{G} = \langle g_1, g_2, \ldots g_{L_g} \rangle$
- step-by-step instructions $\overline{S} = \langle s_1, s_2 \ldots s_{L_s} \rangle$
- Construct a single input sequence $\overline{X} = \langle g_1, g_2, \ldots g_{L_g}, \texttt{<SEP>}, s_1, s_2 \ldots s_{L_s} \rangle$
- Fed the sequence into a bidirectional LSTM encoder to produce an encoding

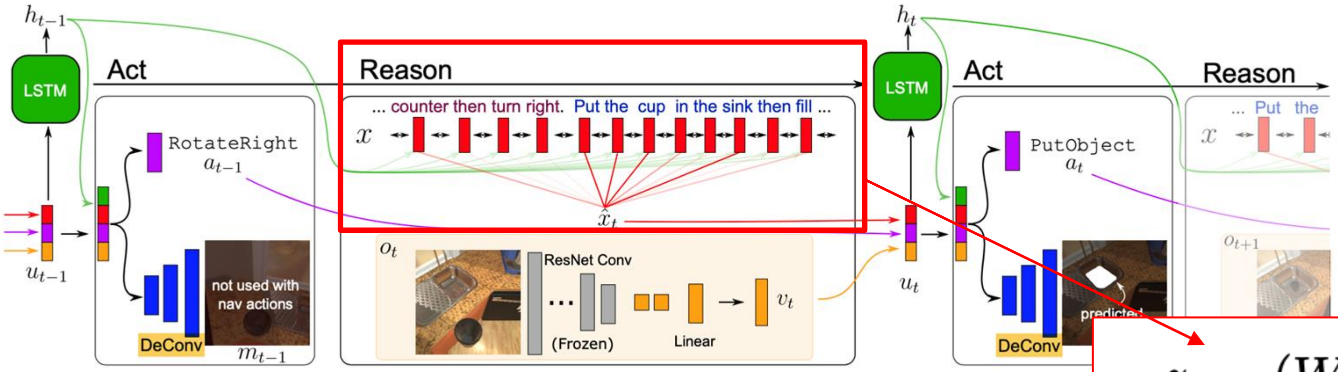
$$x = \{x_1, x_2, \ldots, x_{L_g + L_s}\}$$

Visual Encoding



Each visual observation o_t is encoded with a frozen ResNet-18 CNN followed by two more 1×1 convolution layers and a fully-connected layer.

Attention over language



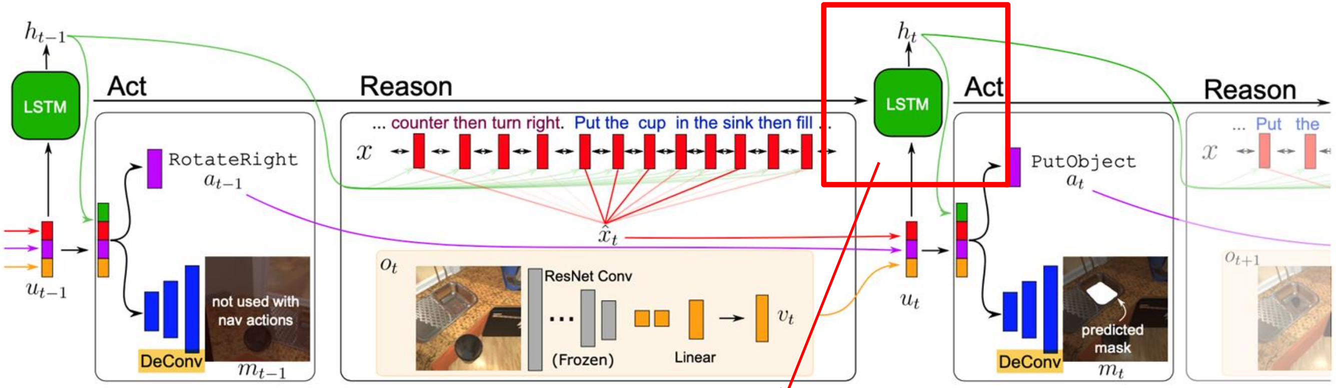
perform soft-attention on the language features x

$$z_t = (W_x h_{t-1})^\top x,$$

$$\alpha_t = \text{Softmax}(z_t),$$

$$\hat{x}_t = \alpha_t^\top x$$

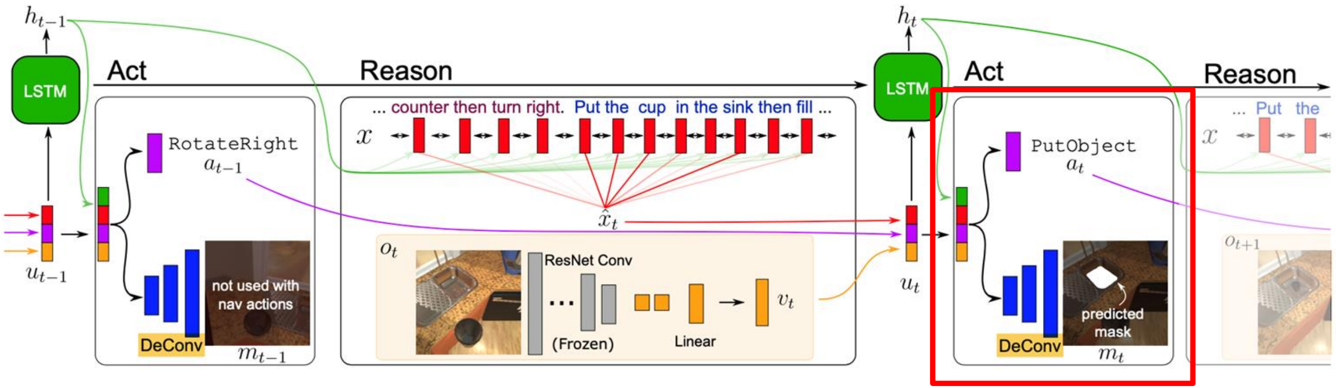
Action decoding



$$u_t = [v_t; \hat{x}_t; a_{t-1}],$$

$$h_t = \mathrm{LSTM}\left(u_t, h_{t-1}\right)$$

Action and mask prediction



Action loss: softmax cross entropy

Mask loss: binary cross entropy rebalanced for sparsity

$$a_t = \text{argmax}\left(W_a\left[h_t; u_t\right]\right)$$

$$m_t = \sigma\left(\textbf{deconv}\left[h_t; u_t\right]\right)$$

Progress Monitors

Progress prediction helps learn the utility of each state in the process of achieving the overall task.

$p_t = \sigma \left(W_p \left[h_t ; u_t \right] \right) \in [0, 1]$ normalized time-stamp value

Sub-goal prediction encourages the agent to coarsely track its progress through the language directive.

$c_t = \sigma \left(W_c \left[h_t ; u_t \right] \right) \in [0, 1]$ normalized number of completed sub-goals

Evaluation Metrics

- Task Success
- Goal-Condition Success
- Path Weighted Metrics
- Sub-Goal Evaluation

Task Success

1 if the object positions and state changes correspond correctly to the task goal-conditions at the end of the action sequence, and 0 otherwise.

“Put a hot potato slice on the counter”

succeed if any potato slice object has changed to the heated state and is resting on any counter top surface.

Goal-Condition Success

The goal-condition success of a model is the ratio of goal-conditions completed at the end of an episode to those necessary to have finished a task.

“Put a hot potato slice on the counter”

- *a potato must be sliced*
- *a potato slice should become heated*
- *a potato slice should come to rest on a counter top.*
- *the same potato slice that is heated should be on the counter top.*

Path Weighted Metrics

Both two metrics have a path weighted version, that considers the length of the expert demonstration.

The path weighted score P_s for metric s is given as $p_s = s \times \frac{L^*}{max(L^*, \hat{L})}$

where $\hat{L}$ is the number of actions the model took in the episode, and L^* is the number of actions in the expert demonstration.

Goal-Condition Success

The ability of a model to accomplish the next sub-goal conditioned on the preceding expert sequence.

Analysis

	Validation				Test			
	Seen		*Unseen*		*Seen*		*Unseen*	
Model	Task	Goal-Cond	Task	Goal-Cond	Task	Goal-Cond	Task	Goal-Cond
NO LANGUAGE	0.0 (0.0)	5.9 (3.4)	0.0 (0.0)	6.5 (4.7)	0.2 (0.0)	5.0 (3.2)	0.2 (0.0)	6.6 (4.0)
NO VISION	0.0 (0.0)	5.7 (4.7)	0.0 (0.0)	6.8 (6.0)	0.0 (0.0)	3.9 (3.2)	0.2 (0.1)	6.6 (**4.6**)
GOAL-ONLY	0.1 (0.0)	6.5 (4.3)	0.0 (0.0)	6.8 (5.0)	0.1 (0.1)	5.0 (3.7)	0.2 (0.0)	6.9 (4.4)
INSTRUCTIONS-ONLY	2.3 (1.1)	9.4 (6.1)	0.0 (0.0)	**7.0** (4.9)	2.7 (1.4)	8.2 (5.5)	**0.5** (**0.2**)	7.2 (**4.6**)
SEQ2SEQ	2.4 (1.1)	9.4 (5.7)	**0.1** (0.0)	6.8 (4.7)	2.1 (1.0)	7.4 (4.7)	**0.5** (**0.2**)	7.1 (4.5)
+ PM PROGRESS-ONLY	2.1 (1.1)	8.7 (5.6)	0.0 (0.0)	6.9 (5.0)	3.0 (1.7)	8.0 (5.5)	0.3 (0.1)	**7.3** (4.5)
+ PM SUBGOAL-ONLY	2.1 (1.2)	9.6 (5.5)	0.0 (0.0)	6.6 (4.6)	3.8 (1.7)	8.9 (5.6)	**0.5** (**0.2**)	7.1 (4.5)
+ PM Both	**3.7** (**2.1**)	**10.0** (**7.0**)	0.0 (0.0)	6.9 (**5.1**)	**4.0** (**2.0**)	**9.4** (**6.3**)	0.4 (0.1)	7.0 (4.3)
HUMAN	-	-	-	-	-	-	91.0 (85.8)	94.5 (87.6)

~8% goal-condition success rate (partially complete tasks)

	Validation				Test			
	Seen		*Unseen*		*Seen*		*Unseen*	
Model	Task	Goal-Cond	Task	Goal-Cond	Task	Goal-Cond	Task	Goal-Cond
NO LANGUAGE	0.0 (0.0)	5.9 (3.4)	0.0 (0.0)	6.5 (4.7)	0.2 (0.0)	5.0 (3.2)	0.2 (0.0)	6.6 (4.0)
NO VISION	0.0 (0.0)	5.7 (4.7)	0.0 (0.0)	6.8 (6.0)	0.0 (0.0)	3.9 (3.2)	0.2 (0.1)	6.6 (**4.6**)
GOAL-ONLY	0.1 (0.0)	6.5 (4.3)	0.0 (0.0)	6.8 (5.0)	0.1 (0.1)	5.0 (3.7)	0.2 (0.0)	6.9 (4.4)
INSTRUCTIONS-ONLY	2.3 (1.1)	9.4 (6.1)	0.0 (0.0)	**7.0** (4.9)	2.7 (1.4)	8.2 (5.5)	**0.5** (**0.2**)	7.2 (**4.6**)
SEQ2SEQ	2.4 (1.1)	9.4 (5.7)	**0.1** (0.0)	6.8 (4.7)	2.1 (1.0)	7.4 (4.7)	**0.5** (**0.2**)	7.1 (4.5)
+ PM PROGRESS-ONLY	2.1 (1.1)	8.7 (5.6)	0.0 (0.0)	6.9 (5.0)	3.0 (1.7)	8.0 (5.5)	0.3 (0.1)	**7.3** (4.5)
+ PM SUBGOAL-ONLY	2.1 (1.2)	9.6 (5.5)	0.0 (0.0)	6.6 (4.6)	3.8 (1.7)	8.9 (5.6)	**0.5** (**0.2**)	7.1 (4.5)
+ PM Both	**3.7** (**2.1**)	**10.0** (**7.0**)	0.0 (0.0)	6.9 (**5.1**)	**4.0** (**2.0**)	**9.4** (**6.3**)	0.4 (0.1)	7.0 (4.3)
HUMAN	-	-	-	-	-	-	91.0 (85.8)	94.5 (87.6)

- Vision and language modalities are necessary to accomplish the tasks.
- The NO LANGUAGE model finishes some goal-conditions by interacting with familiar objects seen during training.
- The NO VISION model similarly finishes some goal-conditions by following low-level language instructions for navigation and memorizing interaction masks for common objects.

	Validation				Test			
	Seen		*Unseen*		*Seen*		*Unseen*	
Model	Task	Goal-Cond	Task	Goal-Cond	Task	Goal-Cond	Task	Goal-Cond
NO LANGUAGE	0.0 (0.0)	5.9 (3.4)	0.0 (0.0)	6.5 (4.7)	0.2 (0.0)	5.0 (3.2)	0.2 (0.0)	6.6 (4.0)
NO VISION	0.0 (0.0)	5.7 (4.7)	0.0 (0.0)	6.8 (6.0)	0.0 (0.0)	3.9 (3.2)	0.2 (0.1)	6.6 (**4.6**)
GOAL-ONLY	0.1 (0.0)	6.5 (4.3)	0.0 (0.0)	6.8 (5.0)	0.1 (0.1)	5.0 (3.7)	0.2 (0.0)	6.9 (4.4)
INSTRUCTIONS-ONLY	2.3 (1.1)	9.4 (6.1)	0.0 (0.0)	**7.0** (4.9)	2.7 (1.4)	8.2 (5.5)	**0.5** (**0.2**)	7.2 (**4.6**)
SEQ2SEQ	2.4 (1.1)	9.4 (5.7)	**0.1** (0.0)	6.8 (4.7)	2.1 (1.0)	7.4 (4.7)	**0.5** (**0.2**)	7.1 (4.5)
+ PM PROGRESS-ONLY	2.1 (1.1)	8.7 (5.6)	0.0 (0.0)	6.9 (5.0)	3.0 (1.7)	8.0 (5.5)	0.3 (0.1)	**7.3** (4.5)
+ PM SUBGOAL-ONLY	2.1 (1.2)	9.6 (5.5)	0.0 (0.0)	6.6 (4.6)	3.8 (1.7)	8.9 (5.6)	**0.5** (**0.2**)	7.1 (4.5)
+ PM Both	**3.7** (**2.1**)	**10.0** (**7.0**)	0.0 (0.0)	6.9 (**5.1**)	**4.0** (**2.0**)	**9.4** (**6.3**)	0.4 (0.1)	7.0 (4.3)
HUMAN	-	-	-	-	-	-	91.0 (85.8)	94.5 (87.6)

- Providing only high-level, underspecified goal language is insufficient to complete the tasks but is enough to complete some goal-conditions.
- Using just low-level, step-by-step instructions, performs similarly to using both high- and low-levels.

Model	Validation Seen Task	Validation Seen Goal-Cond	Validation Unseen Task	Validation Unseen Goal-Cond	Test Seen Task	Test Seen Goal-Cond	Test Unseen Task	Test Unseen Goal-Cond
NO LANGUAGE	0.0 (0.0)	5.9 (3.4)	0.0 (0.0)	6.5 (4.7)	0.2 (0.0)	5.0 (3.2)	0.2 (0.0)	6.6 (4.0)
NO VISION	0.0 (0.0)	5.7 (4.7)	0.0 (0.0)	6.8 (6.0)	0.0 (0.0)	3.9 (3.2)	0.2 (0.1)	6.6 (**4.6**)
GOAL-ONLY	0.1 (0.0)	6.5 (4.3)	0.0 (0.0)	6.8 (5.0)	0.1 (0.1)	5.0 (3.7)	0.2 (0.0)	6.9 (4.4)
INSTRUCTIONS-ONLY	2.3 (1.1)	9.4 (6.1)	0.0 (0.0)	**7.0** (4.9)	2.7 (1.4)	8.2 (5.5)	**0.5** (**0.2**)	7.2 (**4.6**)
SEQ2SEQ	2.4 (1.1)	9.4 (5.7)	**0.1** (0.0)	6.8 (4.7)	2.1 (1.0)	7.4 (4.7)	**0.5** (**0.2**)	7.1 (4.5)
+ PM PROGRESS-ONLY	2.1 (1.1)	8.7 (5.6)	0.0 (0.0)	6.9 (5.0)	3.0 (1.7)	8.0 (5.5)	0.3 (0.1)	**7.3** (4.5)
+ PM SUBGOAL-ONLY	2.1 (1.2)	9.6 (5.5)	0.0 (0.0)	6.6 (4.6)	3.8 (1.7)	8.9 (5.6)	**0.5** (**0.2**)	7.1 (4.5)
+ PM Both	**3.7** (**2.1**)	**10.0** (**7.0**)	0.0 (0.0)	6.9 (**5.1**)	**4.0** (**2.0**)	**9.4** (**6.3**)	0.4 (0.1)	7.0 (4.3)
HUMAN	-	-	-	-	-	-	91.0 (85.8)	94.5 (87.6)

- The two progress monitoring signals are marginally helpful, increasing the success rate by ~1% to ~2%.
- They also lead to more efficient task completion, as indicated by the consistently higher path weighted scores.

Sub-Goal Ablations - **Validation**

	Model	Goto	Pickup	Put	Cool	Heat	Clean	Slice	Toggle	Avg.
Seen	No Lang	28	22	71	**89**	**87**	64	19	90	59
	S2S	49	32	80	87	85	**82**	23	97	67
	S2S + PM	**51**	**32**	**81**	88	85	81	**25**	**100**	**68**
Unseen	No Lang	17	9	31	75	86	13	8	4	30
	S2S	21	20	**51**	**94**	88	21	**14**	**54**	45
	S2S + PM	**22**	**21**	46	92	**89**	**57**	12	32	**46**

- Visual semantic navigation (Goto, Pickup) is considerably harder in unseen environments.
- Simple sub-goals like Cool, and Heat are achieved at a high success rate of ~90% because these tasks are mostly object-agnostic.

Rank	Submission	Created	Unseen Success Rate	Seen Success Rate	Seen PLWSR	Unseen PLWSR	Seen GC	Unseen GC	Seen PLW GC Success Rate	Unseen PLW GC Success Rate
1	Hierarchical Attention Model *Van-Quang Nguyen and Takayuki...*	08/01/2020	**0.0445**	**0.1239**	**0.0820**	**0.0224**	**0.2068**	**0.1234**	**0.1879**	**0.0944**
2	Baseline Seq2Seq + Progress M... *Singh, Bhambri, Kim, Choi (Gl...*	07/22/2020	0.0150	0.0541	0.0251	0.0070	0.1232	0.0808	0.0827	0.0520
3	Baseline + ImprovedMask *personal*	07/29/2020	0.0066	0.0385	0.0207	0.0032	0.1018	0.0798	0.0757	0.0551
4	baseline *DeepblueAI*	07/10/2020	0.0059	0.0365	0.0202	0.0019	0.0851	0.0714	0.0533	0.0410
5	Baseline Seq2Seq+PM (both) *Shridhar et. al (UW)*	03/28/2020	0.0039	0.0398	0.0202	0.0008	0.0942	0.0703	0.0627	0.0426
5	baseline *personal*	07/05/2020	0.0039	0.0359	0.0161	0.0006	0.0889	0.0690	0.0618	0.0421
7	baseline v2 *DeepblueAI*	07/17/2020	0.0033	0.0150	0.0071	0.0008	0.0605	0.0695	0.0383	0.0401
8	Baseline_v2 *Personal*	08/04/2020	0.0026	0.0033	0.0007	0.0005	0.0453	0.0690	0.0316	0.0413
9	test *personal*	08/20/2020	0.0020	0.0027	0.0009	0.0006	0.0463	0.0690	0.0333	0.0458
9	Baseline *personal*	08/04/2020	0.0020	0.0027	0.0009	0.0006	0.0463	0.0690	0.0333	0.0458

Conclusion

- Result: A baseline model based on recent embodied vision-and-language tasks performs poorly on ALFRED
- Challenges: long-horizon task planning, visual semantic navigation, object detection, referring expression grounding, and action grounding
- Goal: Shrink the gap between research benchmarks and real-world applications

OCT 13 2020

TEXAS
The University of Texas at Austin

GROUNDING LANGUAGE IN PLAY

Pierre Sermanet, Corey Lynch

Robotics at Google

PRESENTER: PRIYANKA MANDIKAL

Motivation

- ❖ Children learn language in the context of rich, sensorimotor experience
  - ➔ language acquisition is embodied

- ❖ Infants contribute actions while care-takers contribute relevant words
  - ➔ language acquisition is highly-social

Main Problem

- ❖ Assuming real humans play a critical role in robot language acquisition, what is the most efficient way to go about it?
- ❖ How can we scalably pair robot experience with relevant human language to bootstrap instruction following?

Problem Setting

- Control a robotic arm within a physics simulator
- Manipulate objects in the environment
- Conditioned on an external natural language instruction

Challenges

- High-dimensional continuous sensory inputs and actuators

- Even simple instruction following is notoriously hard

  E.g. “Sweep the block into the drawer”

  - Relate language to low-level perception (What does a block look like? What is a drawer?)
  - Perform visual reasoning (What does it mean for block to be in drawer?)
  - Solve a complex sequential decision problem (What commands do I send to my arm to “sweep”)

- Complex task specification → long-horizon robotic object manipulation from natural language instructions

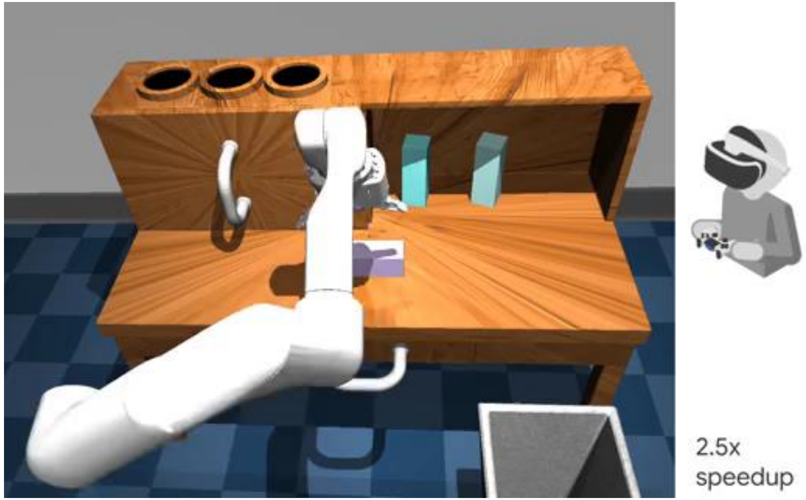
Related Work

- Robot learning from general sensors
  - Imitation learning: requires many human demonstrations
  - Reinforcement Learning: hand-designed reward functions
- Task-agnostic control: Single agent must reach any goal on command
  - Model-based control: Learn model through interactions and then plan; exploration issues
  - Goal-relabeling: used in both IL and RL ; this paper
- Covering state space: Exploration vs Tele-operated play
- Instruction following: restricted env and simplified actuators
  “learning to follow *natural* language is still not the standard in instruction following research” → restricted vocab and grammar

Prior Work

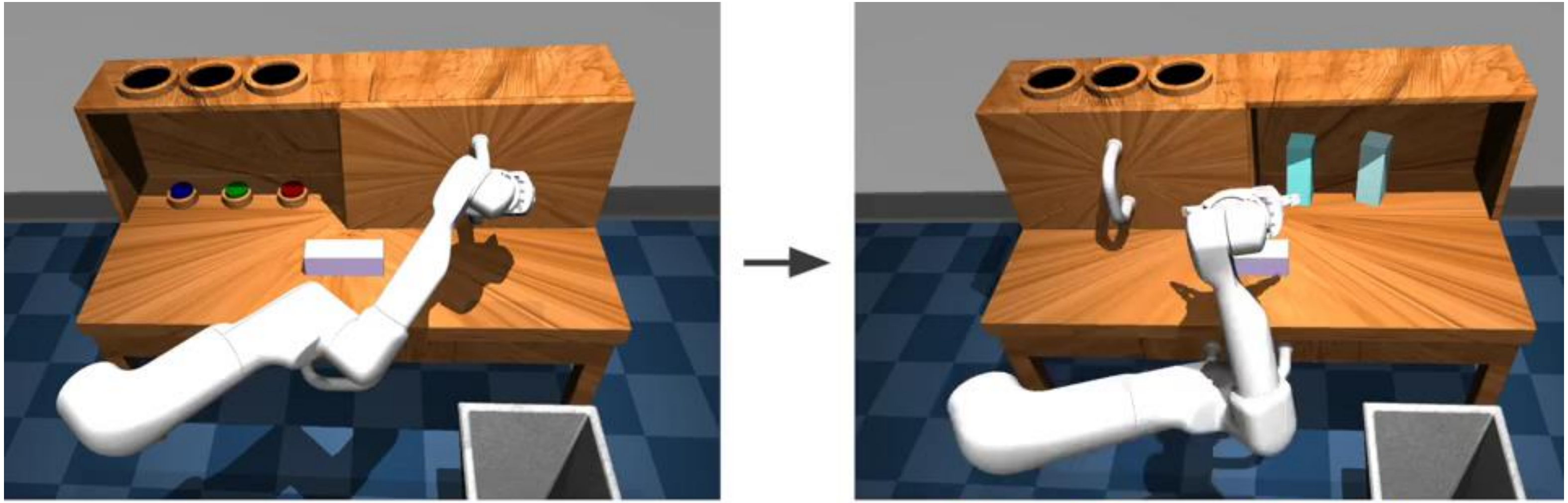
Learning from Play (LfP): Lynch et. al. CORL 2019

- Learning general-purpose skills from onboard sensors
- Tele-operated “play” data → relabeled imitation learning → goal-directed policy
- Limitation: tasks need to be specified using a *goal image* → impractical in real-world environments



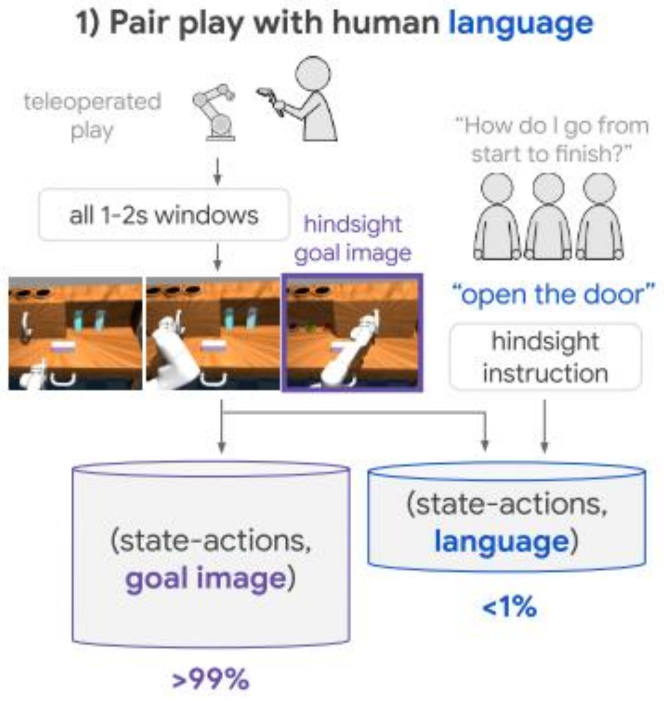
Prior Work

Learning from Play



Goal → Single Play-LMP policy

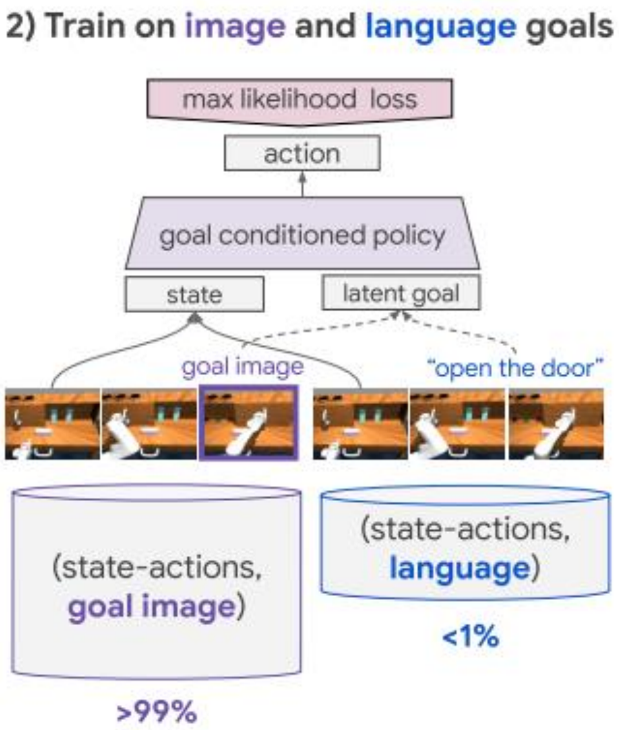
Overview



1a. Cover the space with teleoperated play

1b. Pair play with human language (Hindsight Instruction Pairing)

Overview

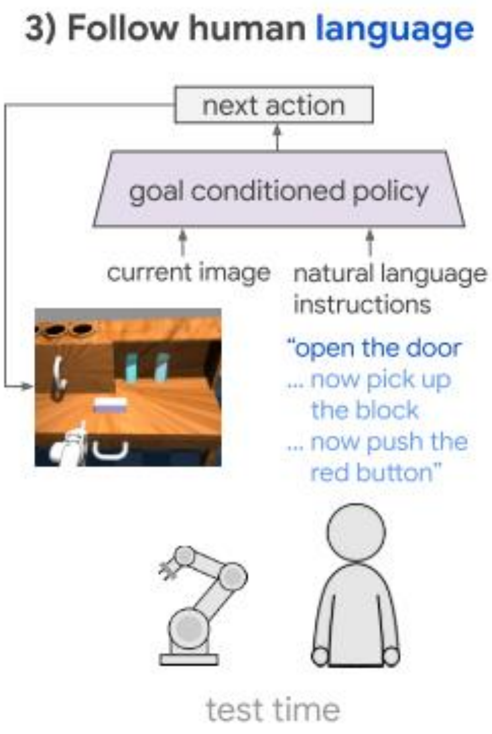


2. Multicontext imitation learning

- train a single policy to solve image or language goals

- highly data efficient

Overview



3. Condition on human language at test time

Preliminaries

- Relabeled Imitation Learning
  – Goal conditioned learning – train a single agent to reach any goal
  – Goal conditioned behavior cloning → relabel collected data

$$\mathcal{L}_{\mathrm{GCBC}} = \mathbb{E}_{(\tau,\, s_g) \sim \mathcal{D}_R} \left[\sum_{t=0}^{|\tau|} \log \pi_\theta \left(a_t | s_t, s_g \right) \right]$$

- Teleoperated Play
  – adds diversity to the dataset (fully cover state space)

- Learning from Play (LfP)
  – combines relabeled imitation learning with teleoperated play

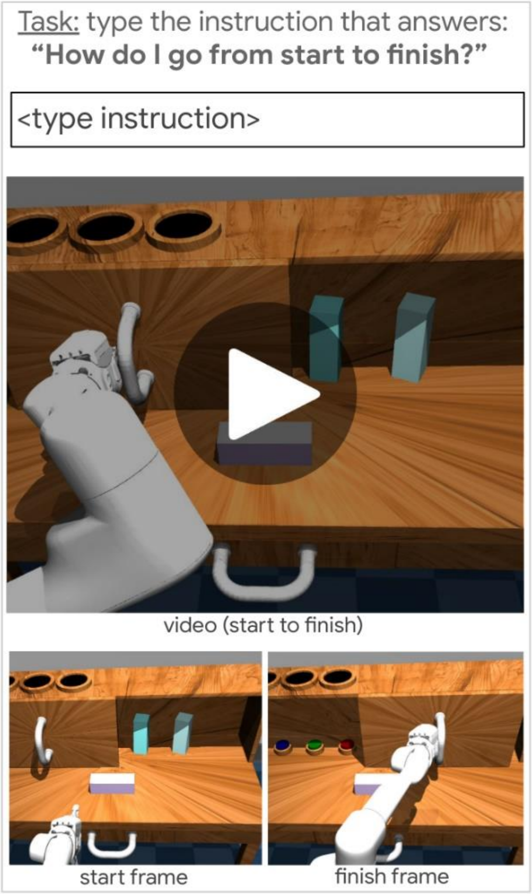
$$\mathcal{L}_{\mathrm{LfP}} = \mathbb{E}_{(\tau,\, s_g) \sim D_{\mathrm{play}}} \left[\sum_{t=0}^{|\tau|} \log \pi_\theta \left(a_t | s_t, s_g \right) \right]$$

APPROACH

Hindsight Instruction Pairing

- Sample any robot behavior from play, then collect an optimal instruction
- After-the-fact natural lang instructions, operator actions not affected by instructions → more diverse play and instruction dataset
- No restrictions on vocabulary or grammar

“pick the object and then lift it up.”
“pull the drawer.”
“drag the object into the drawer”
“drop the object,and again pickup the object high”
“close the drawer”
“do nothing.”



Hindsight Instruction Pairing

- Assumes access to D_{play} consisting of hindsight goal image samples

- From $D_{\text{play}} \rightarrow D_{(\text{play,lang})}$ consisting of hindsight instruction samples

Algorithm 3 Pairing robot sensor data with natural language instructions.

Input: D_{play}, a relabeled play dataset holding (τ, s_g) pairs.
Input: $D_{(\text{play,lang})} \leftarrow \{\}$.
Input: get_hindsight_instruction(): human overseer, providing after-the-fact natural language instructions for a given τ.
Input: K, number of pairs to generate, $K << |D_{\text{play}}|$.

for $0...K$ **do**
  # Sample random trajectory from play.
  $(\tau, \_) \sim D_{\text{play}}$
  # Ask human for instruction making τ optimal.
  l = get_hindsight_instruction(τ)
  Add $(\tau,\ l)$ to $D_{(\text{play,lang})}$
end for

Instruction Samples

Task	Natural language instructions
open sliding door	"move the door all the way to the right" "slide the door to the right" "move the sliding door all the way to the right and let go"
close sliding door	"‘Grasp the door handle, then slide the door to the left" "move the door all the way to the left"
open drawer	"‘open the cabinet drawer" "open the drawer and let go"
close drawer	"‘close the drawer and let go" "‘close the drawer"
grasp flat	"Pick up the block" "grasp the object and lift it up" "grasp the object and move your hand up"
grasp lift	"Pick up the object from the drawer and drop it on the table." "hold the block and place it on top of the table"
grasp upright	"Pick the object and lift it up" "grasp the object and lift"
knock	"push the block forward" "push the object towards the door"
pull out shelf	"Drag the block from the shelf towards the drawer" "pick up the object from the shelf and drop it on the table"

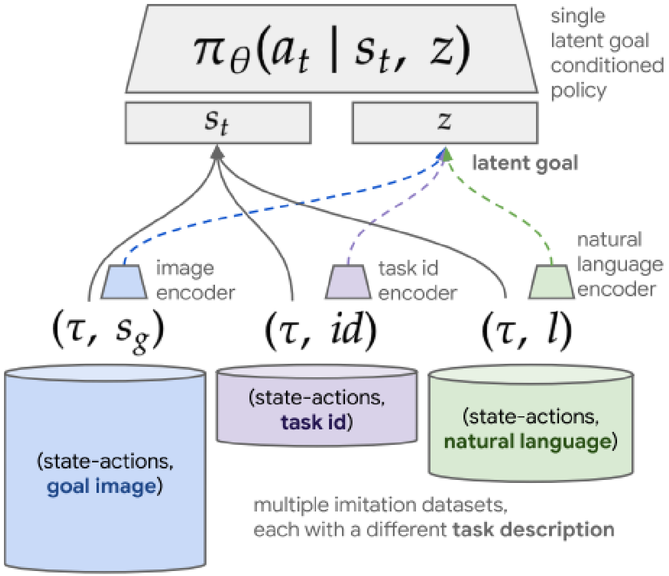
Task	Natural language instructions
put in shelf	"grasp the object and place it inside the cabinet shelf" "Pick the object, move it into the shelf and then drop it."
push red	"go press the red button" "Press the red button"
push green	"press the green button" "push the green button"
push blue	"push the blue button" "press down on the blue button."
rotate left	"Pick up the object, rotate it 90 degrees to the left and drop it on the table" "rotate the object 90 degrees to the left"
rotate right	"turn the object to the right" "rotate the block90 degrees to the right"
sweep	"roll the object into the drawer" "drag the block into the drawer"
sweep left	"Roll the block to the left" "close your fingers and roll the object to the left"
sweep right	"roll the object to the right" "Push the block to the right."

Multicontext Imitation Learning (MCIL)

- Generalization of contextual imitation to multiple heterogenous contexts

- Multiple imitation learning datasets, each with a different way of describing tasks and different cost of collection
  - E.g. goal image, task id, natural language, video demonstration, etc

- Trains
  - A single *latent goal* conditioned policy $\pi_\theta(a_t | s_t, z)$ over *all* datasets simultaneously
  - A set of encoders, one per dataset; each maps task description → *shared* latent space



Multicontext Imitation Learning (MCIL)

Training Procedure:

- At each training step, for each dataset:
  - sample a minibatch of trajectory-context pairs
  - encode the contexts in the latent space
- Contextual imitation objective (per dataset)

$$\mathcal{L}_{\text{context}}(D, h) = \mathbb{E}_{(\tau,\, c) \sim D}\left[\sum_{t=0}^{|\tau|} \log \pi_\theta\left(a_t | s_t, f_\theta(c)\right)\right]$$

- Full MCIL objective → averaged over all datasets

$$\mathcal{L}_{\text{MCIL}} = \frac{1}{|\mathcal{D}|}\sum_{k}^{|\mathcal{D}|} \mathcal{L}_{\text{context}}(D_k, h_k)$$

Algorithm 1 Multicontext imitation learning

Input: $\mathcal{D} = \{D^0, ..., D^K\}$, $D^k = \{(\tau_i^k, c_i^k)\}_{i=0}^{D^k}$, One dataset per context type (e.g. goal image, language instruction, task id), each holding pairs of (demonstration, context).
Input: $\mathcal{F} = \{f_\theta^0, ..., f_\theta^K\}$, One encoder per context type, mapping context to shared latent goal space, e.g. $z = f_\theta^k(c^k)$.
Input: $\pi_\theta(a_t|s_t, z)$, Single latent goal conditioned policy.
Input: Randomly initialize parameters $\theta = \{\theta_\pi, \theta_{f^0}, ..., \theta_{f^K}\}$
while True **do**
  $\mathcal{L}_{\text{MCIL}} \leftarrow 0$
  # Loop over datasets.
  for $k = 0...K$ **do**
    # Sample a (demonstration, context) batch from this dataset.
    $(\tau^k, c^k) \sim D^k$
    # Encode context in shared latent goal space.
    $z = f_\theta^k(c^k)$
    # Accumulate imitation loss.
    $\mathcal{L}_{\text{MCIL}} \mathrel{+}= \sum_{t=0}^{|\tau^k|} \log \pi_\theta(a_t|s_t, z)$
  end for
  # Average gradients over context types.
  $\mathcal{L}_{\text{MCIL}} \mathrel{*}= \frac{1}{|\mathcal{D}|}$
  # Train policy and all encoders end-to-end.
  Update θ by taking a gradient step w.r.t. $\mathcal{L}_{\text{MCIL}}$
end while

Multicontext Imitation Learning (MCIL)

- Advantages
    - Being context-agnostic → enables highly efficient training
        - Learn the majority of control from the cheapest data source
        - Learn general task conditioning from a small number of labelled examples
          E.g. Natural language instructions < 1% of collected robot experience!
    - Broadly useful beyond this paper

LangLfP

- Special case of MCIL
- Two datasets

$$\mathcal{D} = \{D_{\text{play}}, D_{(\text{play},\text{lang})}\}$$

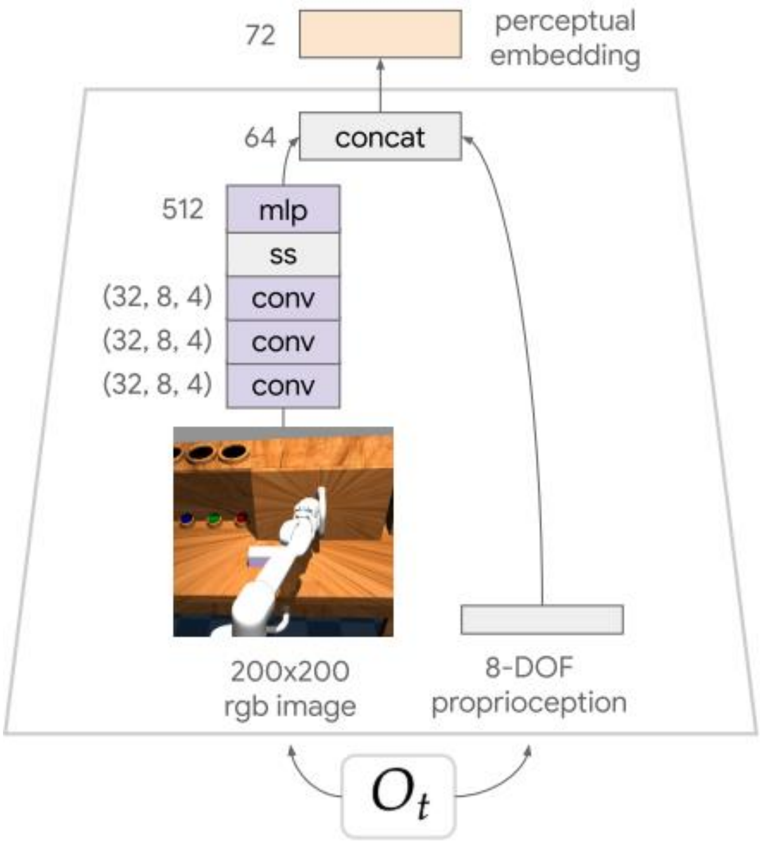
- Tasks
  - hindsight goal image
  - hindsight instructions

- Encoders: $\mathcal{F} = \{g_{\text{enc}}, s_{\text{enc}}\}$
  - Maps image goal and instructions → shared *visuo-lingual goal* space

- *Learns perception, language understanding and control end-to-end*

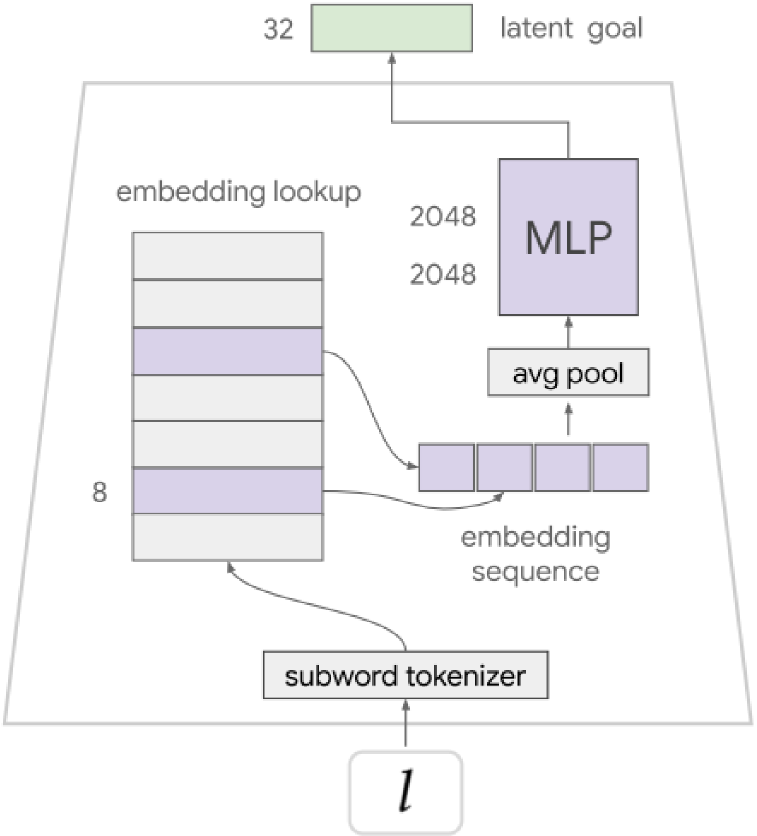
LangLfP: Perception Module

- Observations:
  - High-dim image (200x200x3)
  - Proprioceptive sensor readings i.e. robot joint angles and locations in Cartesian coordinate space



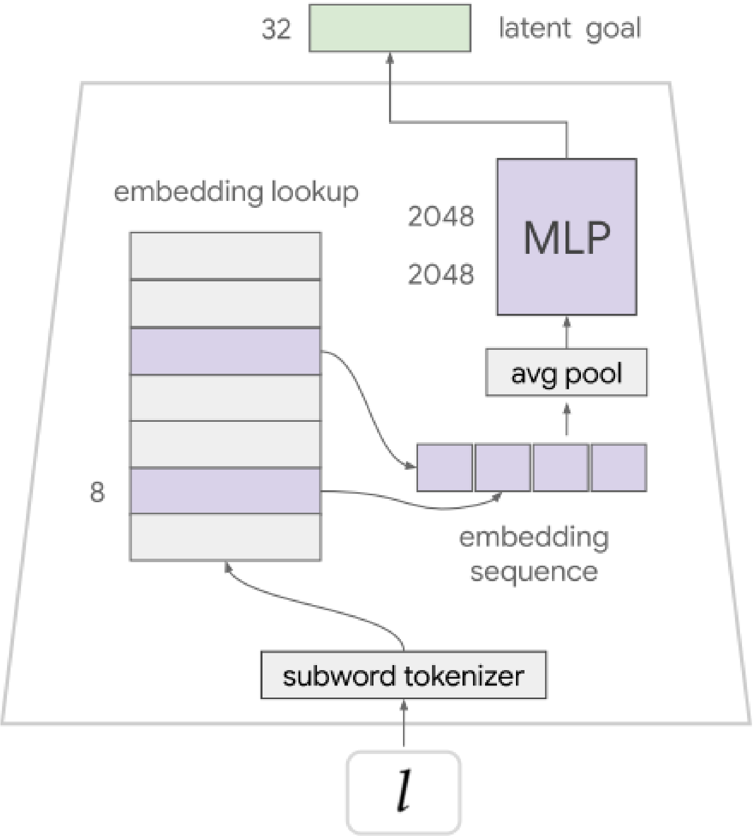
LangLfP: Language Module

- Two approaches:
  - From scratch (LangLfP)
  - Transfer Learning (TransferLangLfP)



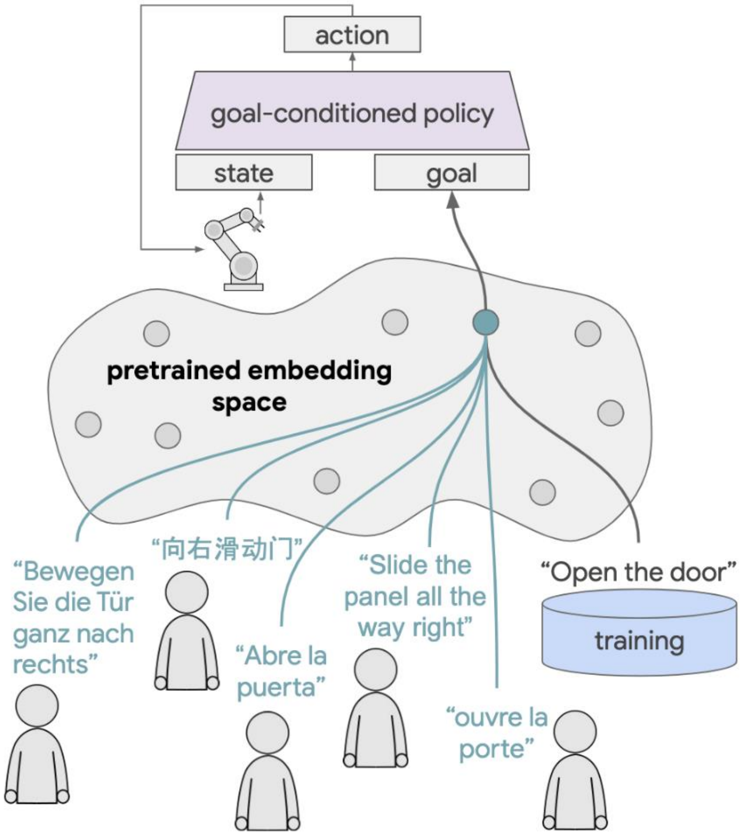
LangLfP: Language Module from Scratch

- Tokenize raw text into subwords
- Retrieve subword embeddings from a lookup table
- Summarize embeddings into a point in *z* space
- Embedding fed to 2-layer MLP



TransferLangLfP: Language Module via Transfer Learning

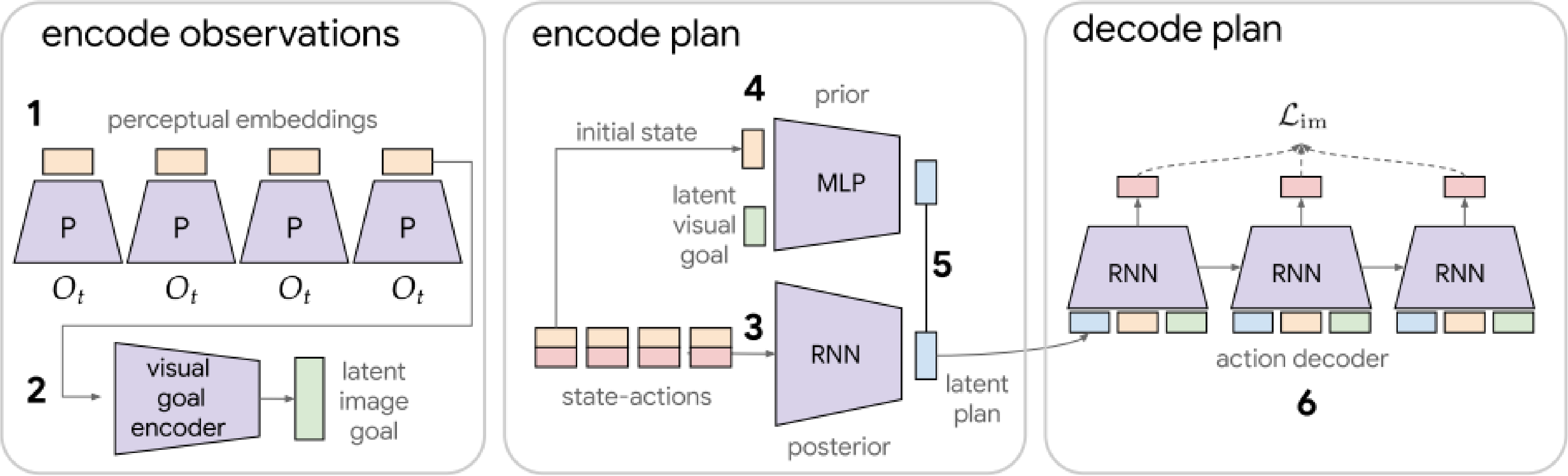
- Pretrained embeddings from Multilingual Universal Sentence Encoder (MUSE)
- Maps sentences → 512-D vector
- Benefits
  - Serves as a strong prior if there is a semantic match between source and target domains
  - Encodes word similarity → follow out-of-distribution instructions in zero shot



LangLfP: Control Module

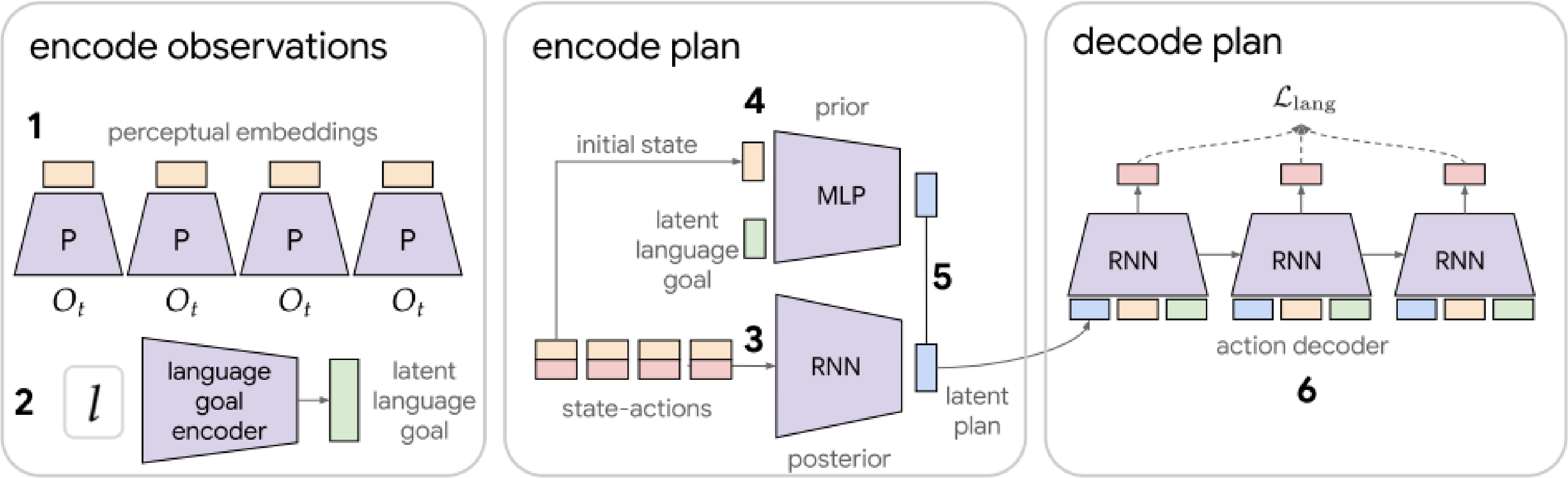
- Implement multicontext control policy: $\pi_\theta(a_t|s_t, z)$
- Use Latent Motor Plans (from LfP paper)
  - Goal directed imitation architecture
  - Uses latent variables to model multimodality
  - Seq2seq CVAE that auto-encodes contextual demos through a latent “plan” space
  - Decoder: goal-conditioned policy
  - Refer to LfP for more details

Multicontext LMP: Goal Image

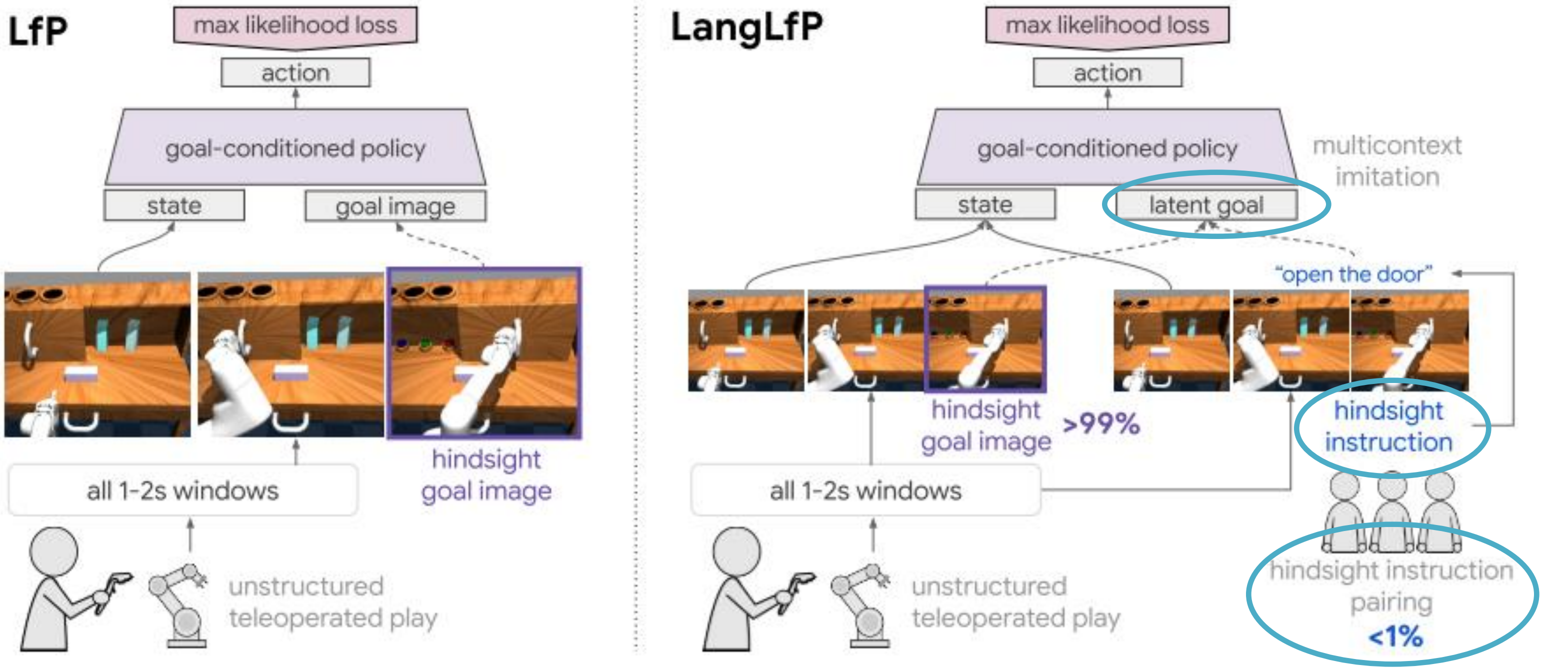


The University of Texas at Austin
WHAT STARTS HERE CHANGES THE WORLD

Multicontext LMP: Language

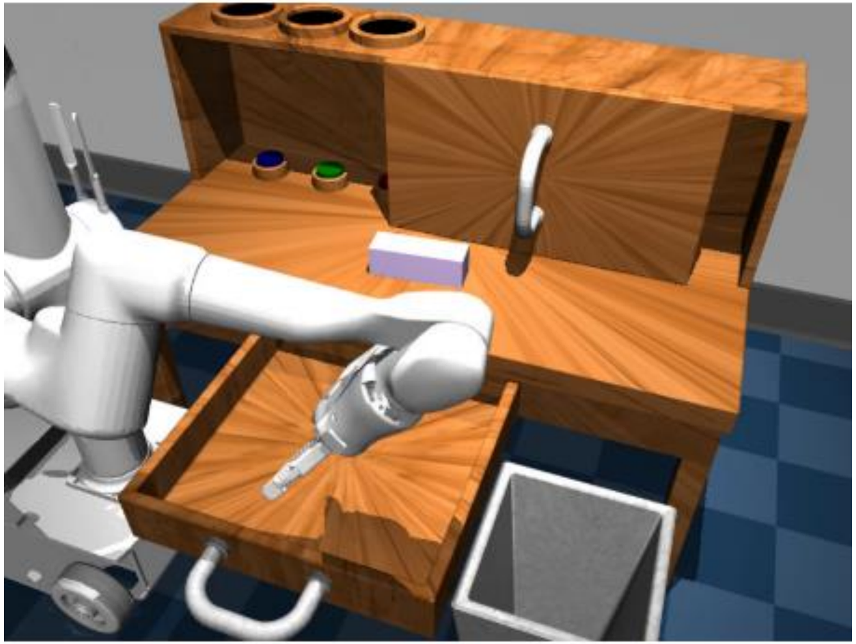


LfP vs LangLfP



Experimental Setup: Environment

- Situated robot in a 3D environment
- 8-DOF robot arm and parallel gripper
- RGB video sensors
- Proprioceptive sensors
- Goal: Agent must perform high-frequency, closed-loop continuous control to solve user-described manipulation tasks



Experimental Setup: Methods

- LangBC – language, but no play – multi-task demos – $D_{(demo,lang)}$
- LfP – play, but no language – D_{play}
- LangLfP – play and language – D_{play} and $D_{(play,lang)}$
- Restricted LangLfP – LangLfP restricted to size of $D_{(demo,lang)}$
- TransferLangLfP – LangLfP using MUSE embeddings - $D_{(play,lang)}$

→ 2 sets of experiments – pixel and state

Experiments: Ask-Me-Anything (AMA)

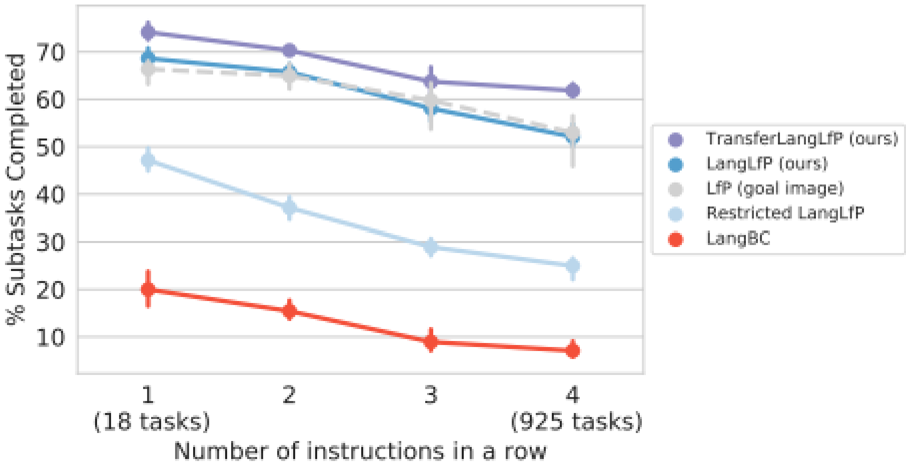
- Multi-stage instruction following
- Derived from Multi-18 → 18 evaluation tasks described in LfP. E.g. open sliding door, sweep, close sliding door, etc
- Consider all valid N-stage transitions between the 18 tasks ➔ Chain-2, Chain-3, Chain-4 manipulation benchmarks
- Multi-18 can be seen as a subset of this extended set

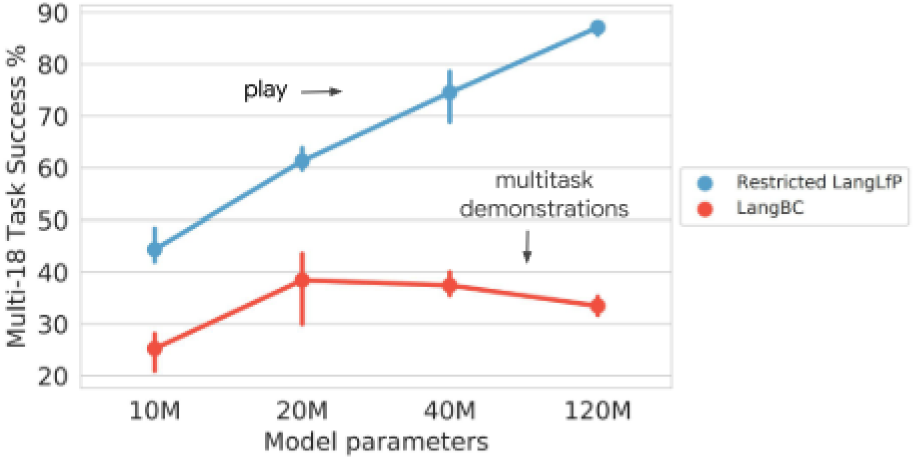
Experiments: Ask-Me-Anything (AMA)

Method	Input	Training source	Task conditioning	Multi-18 Success (18 tasks)	Chain-4 Success (925 long-horizon tasks)
LangBC	pixels	predefined demos	text	20.0% ±3.0	7.1% ±1.5
Restricted LangLfP	pixels	play	text	47.1% ±2.0	25.0% ±2.0
LfP	pixels	play	goal image	66.4% ±2.2	53.0% ±5.0
LangLfP (ours)	pixels	play	text	68.6% ±1.7	52.1% ±2.0
TransferLangLfP (ours)	pixels	**play**	**text**	**74.1%** ±1.5	**68.6%** ±1.6
LangBC	states	predefined demos	text	38.5% ±6.3	13.9% ±1.4
Restricted LangLfP	states	play	text	88.0% ±1.4	64.2% ±1.5
LangLfP (ours)	states	play	text	88.5% ±2.9	63.2% ±0.9
TransferLangLfP (ours)	states	**play**	**text**	**90.5%** ±0.8	**71.8%** ±1.6

- LangLfP ~ LfP, but is more scalable in terms of task conditioning
- TransferLangLfP > LangLfP and original LfP
- RestrictedLangLfP > LangBC; Restricted LangLfP can transition well between tasks; LangBC fails to recover from compounding errors

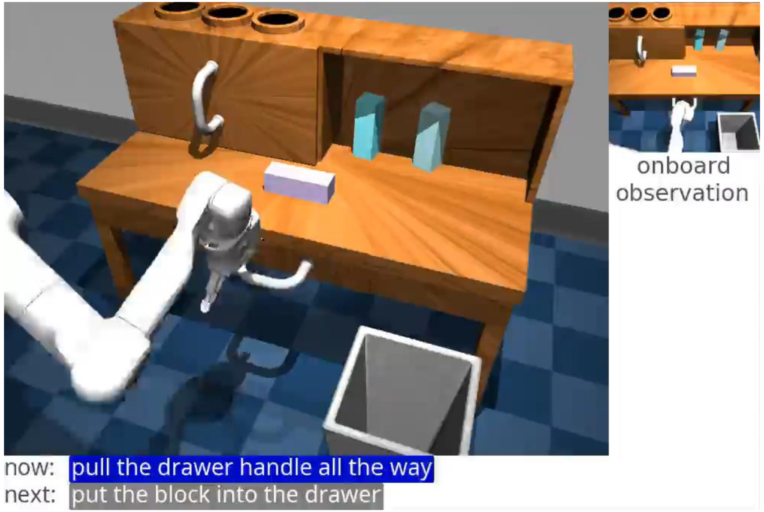
Experiments: Ask-Me-Anything (AMA)





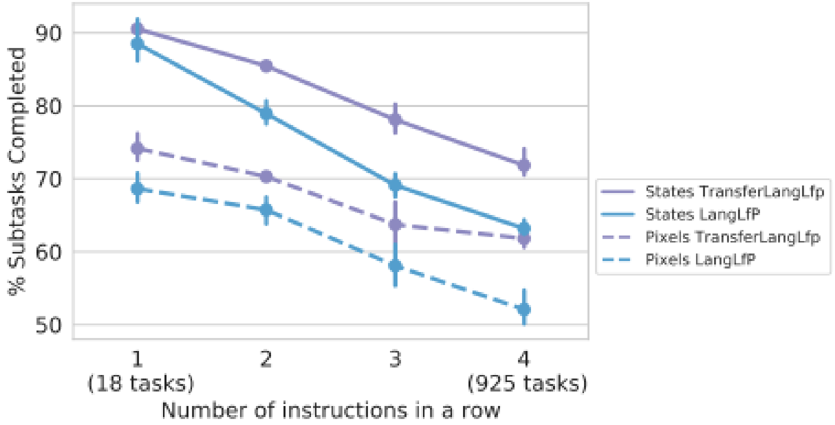
- ❖ As model capacity increases, play model can capitalize on increased strength because of diversity in dataset
- ❖ LangBC constrained to predefined behaviors

Results: Ask-Me-Anything (AMA)



Experiments: Knowledge Transfer

- TransferLangLfP outperforms LangLfP
- ➔ evidence that world knowledge in large corpora is beneficial for downstream robotic manipulation tasks



Experiments: Knowledge Transfer

Out-of-distribution instructions:

- Synonyms
  - "*Drag the block from the shelf*" → "*Retrieve the brick from the cupboard*"
  - OOD-syn eval set: 14k OOD samples across 18 tasks
  - TransferLangLfP generalizes *substantially*!
- 16 different languages
  - OOD-16-lang eval set: Translate (Multi-18 + OOD-syn) 240k samples across 18 tasks

Method	**OOD-syn (∼15k tasks)**	**OOD-16-lang (∼240k tasks)**
Random Policy	0.0% ± 0.0	0.0% ± 0.0
LangLfP	37.6% ± 2.3	27.94% ± 3.5
TransferLangLfP	**60.2%** ± 3.2	**56.0%** ± 1.4

- TransferLangLfP > LangLfP
- LangLfP resorts to producing max likelihood play actions

Results: Knowledge Transfer

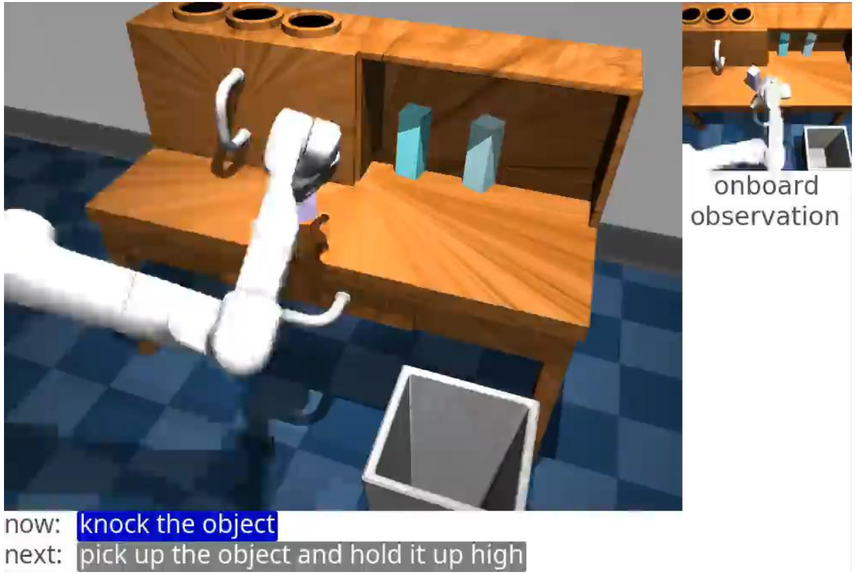
Results: TransferLangLfP vs LangLfP

TransferLangLfP

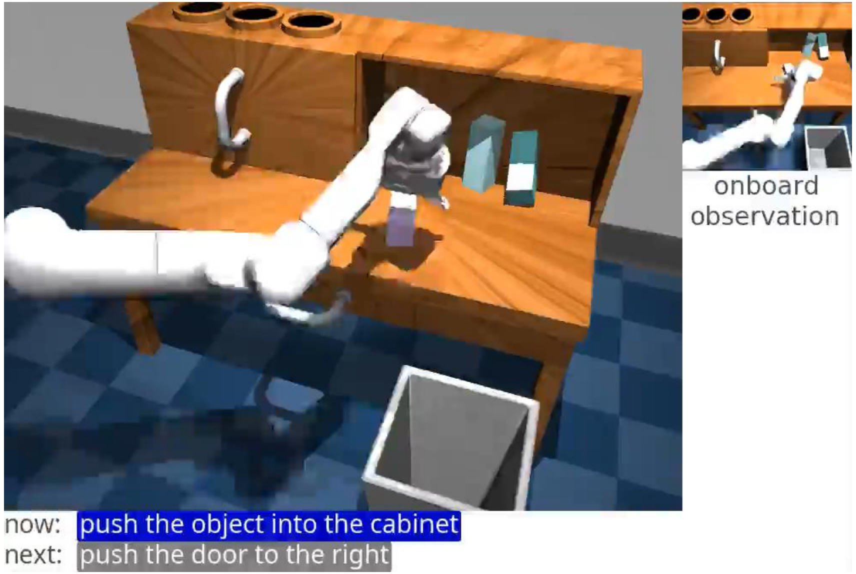
LangLfP

Limitations

Agent times out before task completion



Compounding error → awkward arm configurations



Future Work

- Current → goal-directed imitation, lacks autonomous policy improvement

  Future → Imitation + RL for autonomous policy improvement, not restricted to human actions

- Current → Single env

  Future → Large play corpora, generalization to new rooms and objects

Summary

- Introduced LangLfP, an extension of LfP trained both on relabeled goal image play and play paired with human language instructions
- Multicontext Imitation Learning → reduce the cost of language pairing
- Single policy trained with LangLfP can solve many 3D robotic manipulation tasks over a long horizon from onboard sensors via human language
- Simple technique for knowledge transfer; 16 different languages

Discussion

- Transfer learning on 16 langauges
  - For LangLfP, could have translated instructions to English before feeding into the model

- How do ALFRED and LangLfP compare with each other?
  - ALFRED
    - Pros: mobile robot, larger env diversity, large # of obj state changes (door open/close, lights on/off, bread whole/sliced, tap on/off, vase intact/broken …) → rich lang vocabulary
    - Cons: No physical realism, no fine motor control
  - LangLfP
    - Pros: contact-rich physics env, 8-DOF motor control
    - Cons: Fixed robot, limited state space, limited obj state changes (restricted to pick-up, door open/close, lights on/off) → limited action vocabulary