PhD Proposal:
Using Natural Language to Aid Task Specification in Sequential Decision Making Problems

Prasoon Goyal

Outline

- Introduction
- Related Work
- Completed Work:
  - Using Natural Language for Reward Shaping in Reinforcement Learning (IJCAI 2019)
  - Guiding Reinforcement Learning by Mapping Pixels to Rewards (CoRL 2020)
  - Zero-shot Task Adaptation using Natural Language (arXiv, 2021)
- Proposed Work: Short-term
  - Neurosymbolic Model
  - Policy Adaptation
- Proposed Work: Long-term
  - Policy Regularization
  - Bayesian Inference
  - Supervised Attention

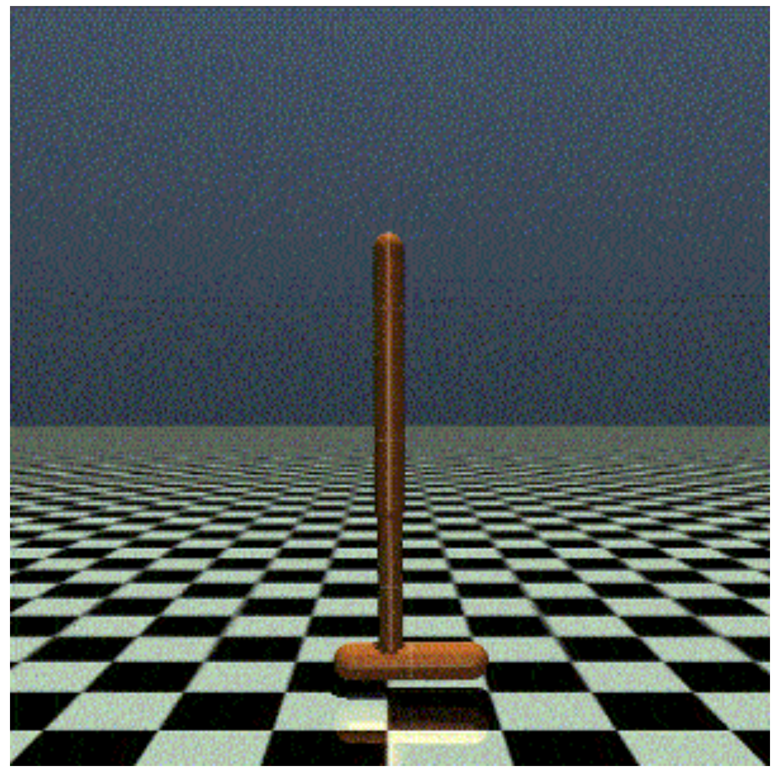
Outline

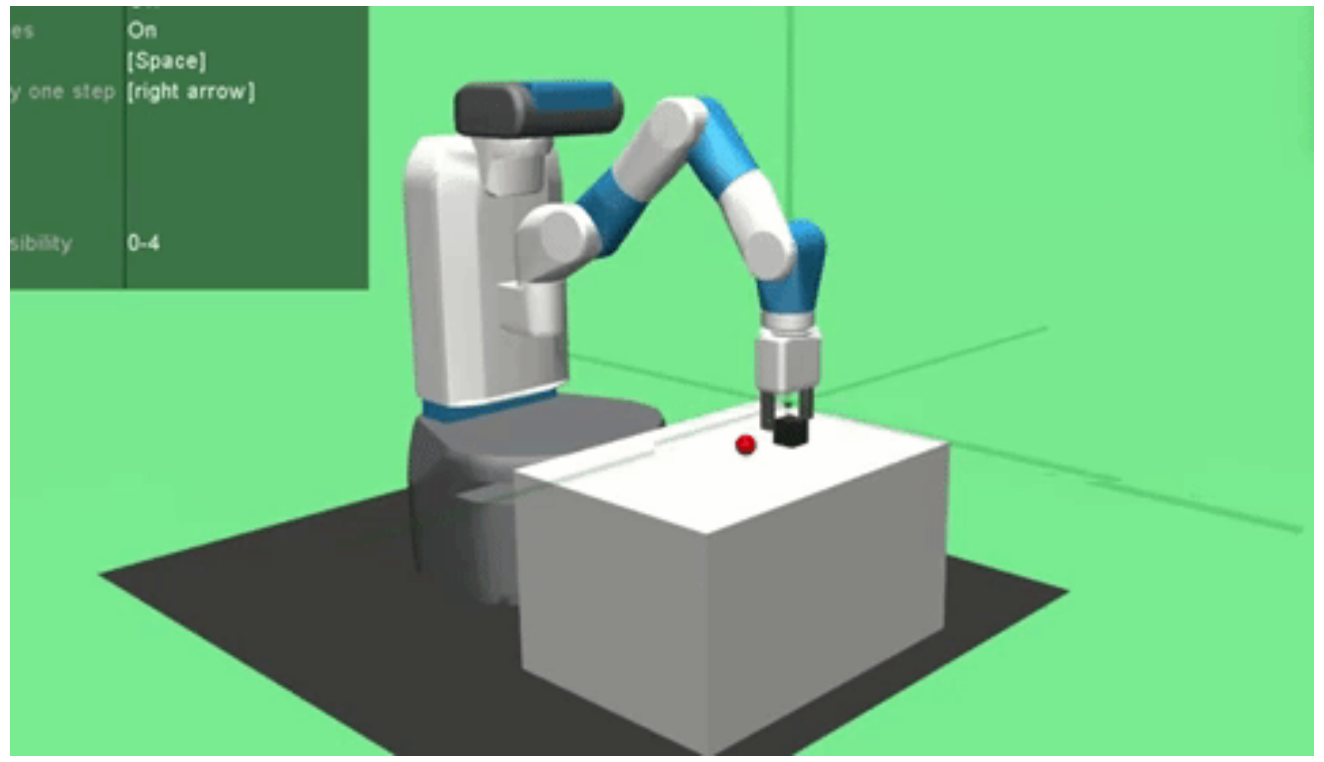
- Introduction
- Related Work
- Completed Work:
  - Using Natural Language for Reward Shaping in Reinforcement Learning (IJCAI 2019)
  - Guiding Reinforcement Learning by Mapping Pixels to Rewards (CoRL 2020)
  - Zero-shot Task Adaptation using Natural Language (arXiv, 2021)
- Proposed Work: Short-term
  - Neurosymbolic Model
  - Policy Adaptation
- Proposed Work: Long-term
  - Policy Regularization
  - Bayesian Inference
  - Supervised Attention

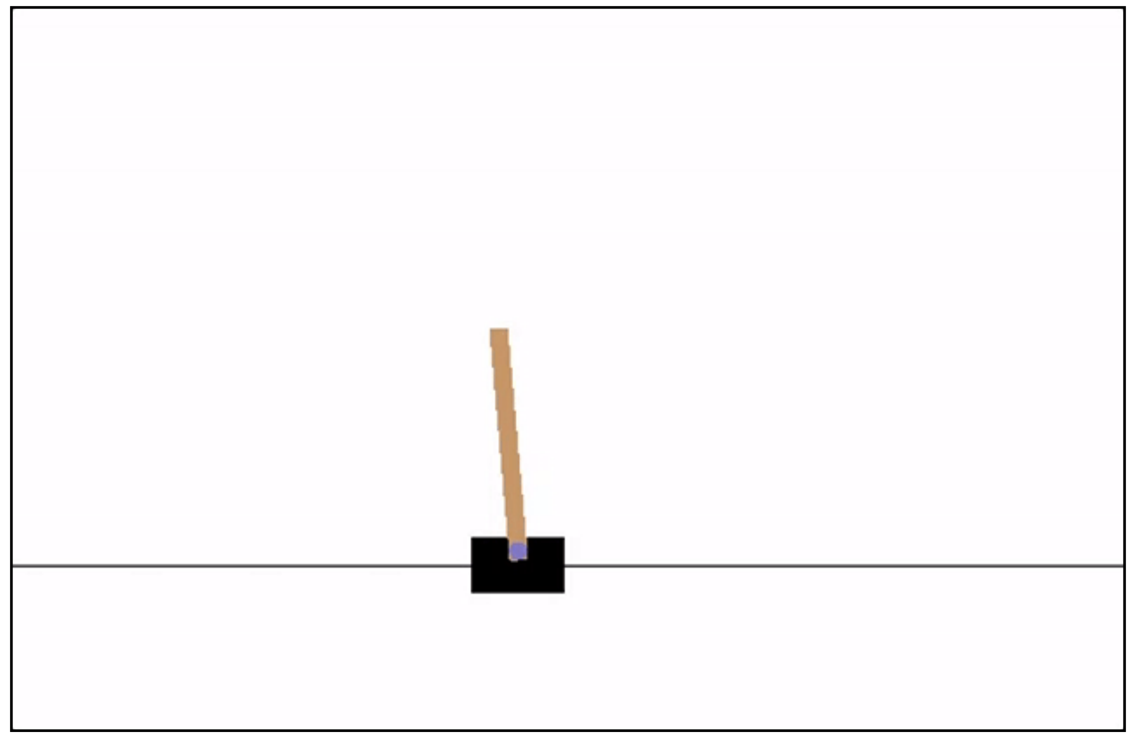
Introduction

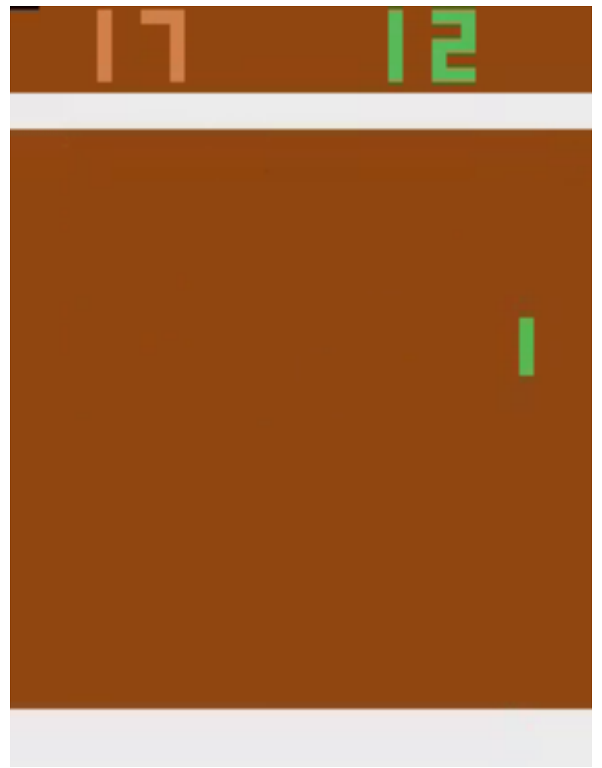
Reinforcement learning (RL) and imitation learning (IL):

Successfully applied in a lot of domains…









but still far from being applicable to real-world tasks.

=> Can we use natural language as an auxiliary learning signal?

Introduction

Language can be used to communicate different kinds of information:

- goals: “Make a sandwich”
- hints: “Plates are in the cabinet above the dishwasher”
- preferences: “Use hummus instead of cheese”
- feedback: “Make it a little less crispy”

Introduction

Language can be used to communicate different kinds of information:
- goals: “Make a sandwich”
- hints: “Plates are in the cabinet above the dishwasher”
- preferences: “Use hummus instead of cheese”
- feedback: “Make it a little less crispy”

Language can be provided by end users.

Introduction

Language can be used to communicate different kinds of information:
- goals: “Make a sandwich”
- hints: “Plates are in the cabinet above the dishwasher”
- preferences: “Use hummus instead of cheese”
- feedback: “Make it a little less crispy”

Language can be provided by end users.

We propose approaches that use natural language to reduce the burden of task design on the end user.

Sequential Decision Making

Reinforcement Learning

Task specification:
Designing reward functions

Imitation Learning

Task specification:
Providing demonstrations

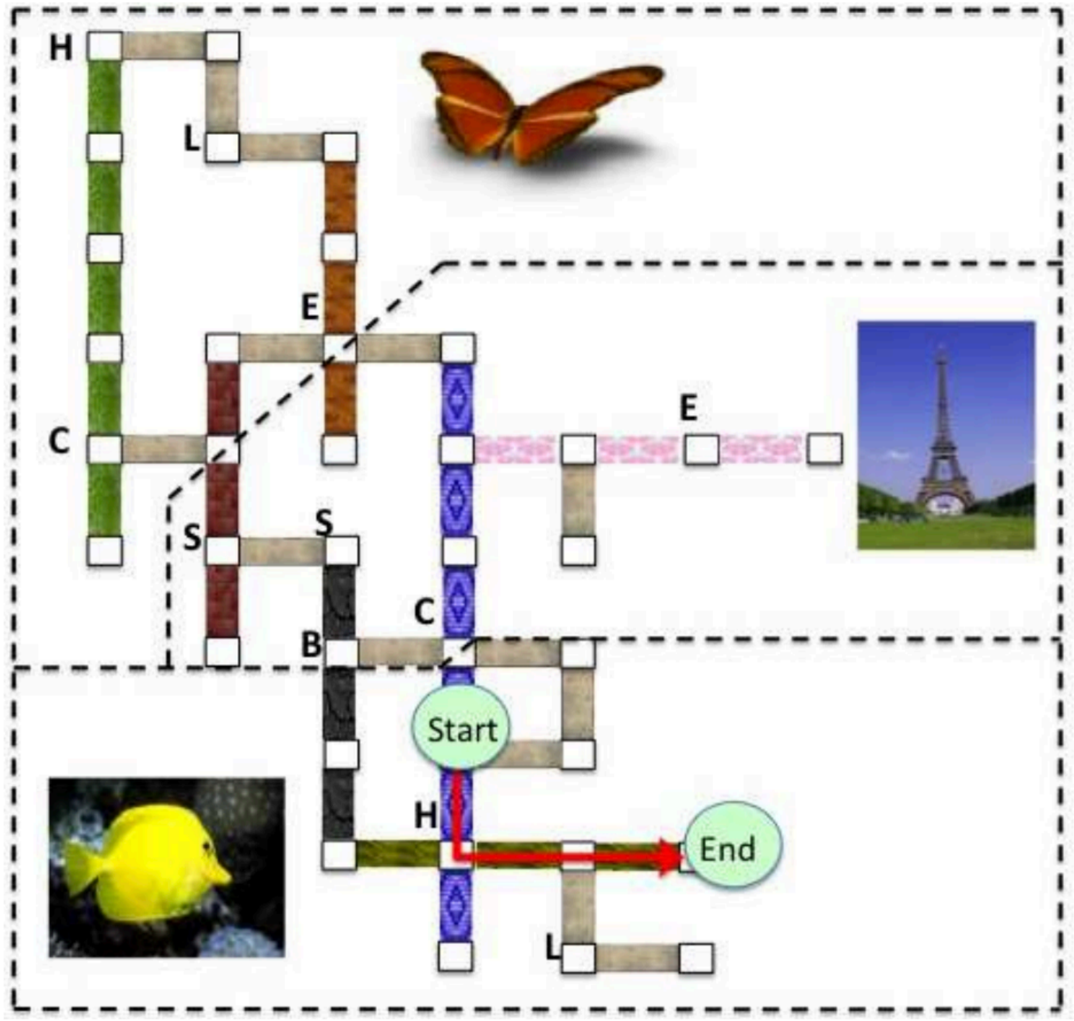
Use natural language to aid these!

Outline

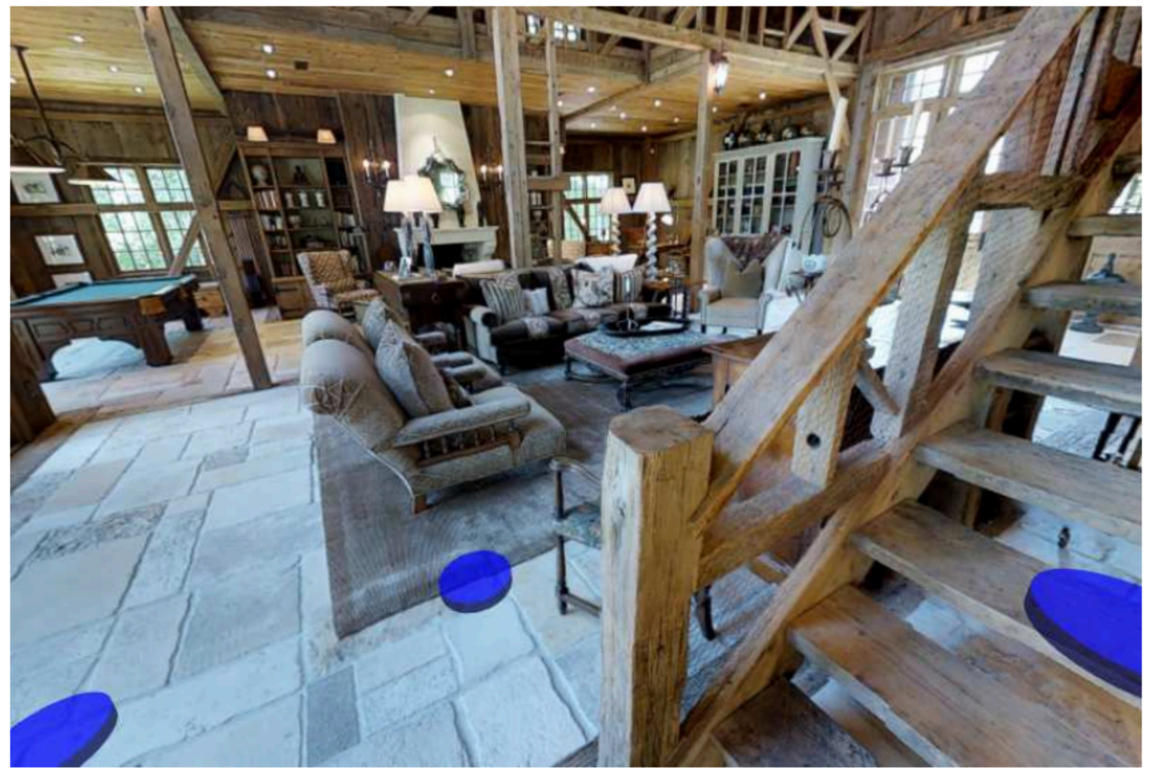
- Introduction
- Related Work
- Completed Work:
  - Using Natural Language for Reward Shaping in Reinforcement Learning (IJCAI 2019)
  - Guiding Reinforcement Learning by Mapping Pixels to Rewards (CoRL 2020)
  - Zero-shot Task Adaptation using Natural Language (arXiv, 2021)
- Proposed Work: Short-term
  - Neurosymbolic Model
  - Policy Adaptation
- Proposed Work: Long-term
  - Policy Regularization
  - Bayesian Inference
  - Supervised Attention

Related Work

Instruction-following



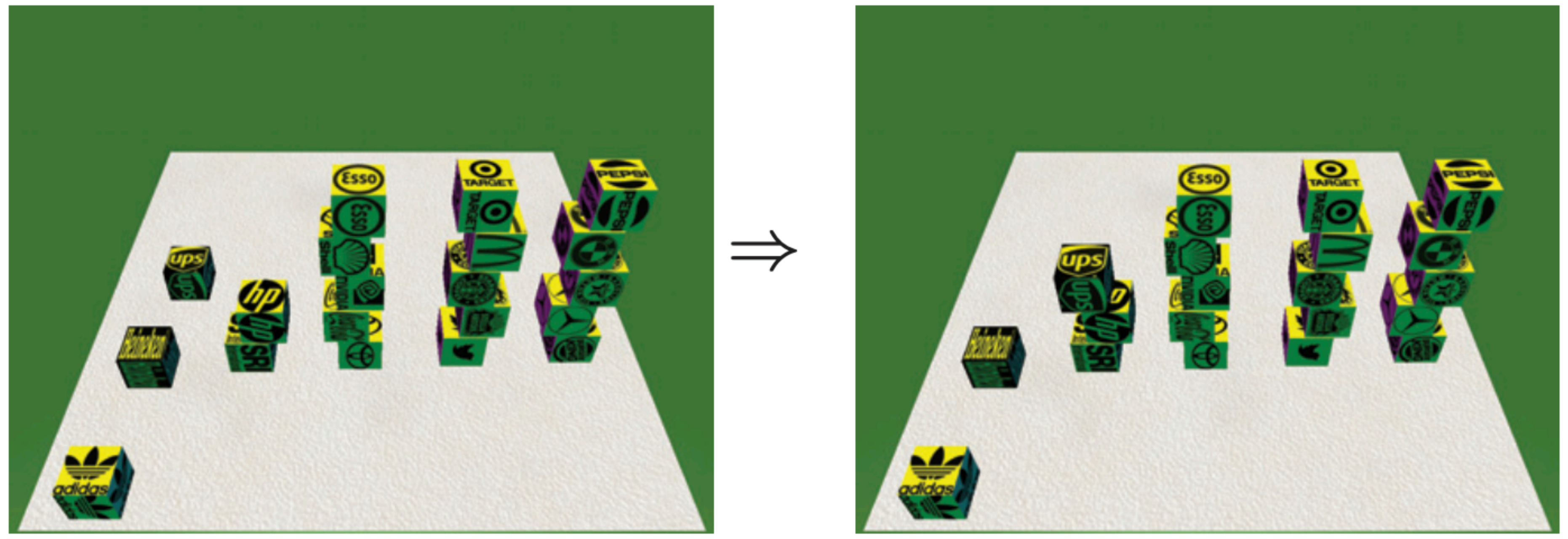
Chen and Mooney, 2011



Instruction: Head upstairs and walk past the piano through an archway directly in front. Turn right when the hallway ends at pictures and table. Wait by the moose antlers hanging on the wall.

Anderson et al., 2018

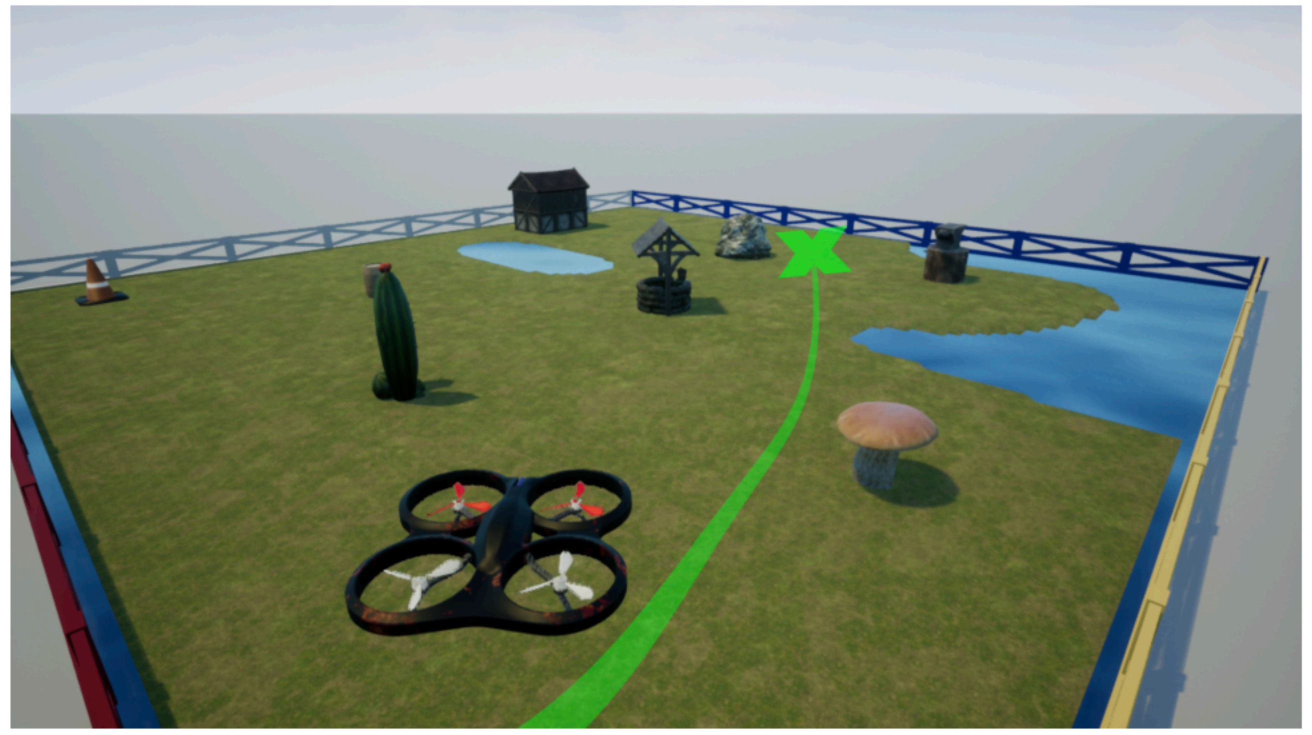
"On the (new) fourth tower, mirror Nvidia with UPS."



Bisk et al., 2018

Goal: "Rinse off a mug and place it in the coffee maker"

Shridhar et al., 2020



Go to the right side of the rock

Blukis et al., 2018

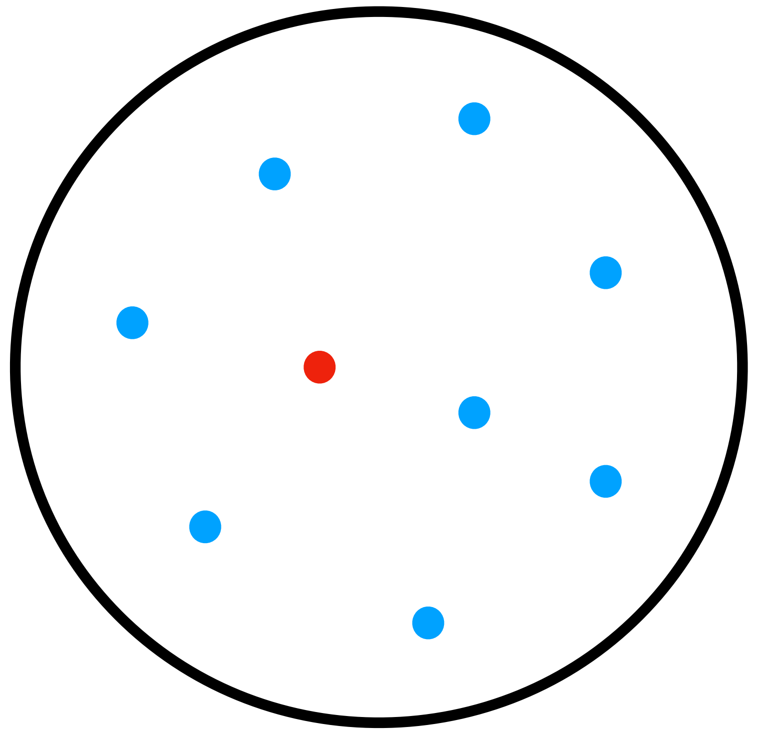
Commands from the corpus
- Go to the first crate on the left and pick it up.
- Pick up the pallet of boxes in the middle and place them on the trailer to the left.
- Go forward and drop the pallets to the right of the first set of tires.
- Pick up the tire pallet off the truck and set it down

Tellex et al., 2011

Related Work

Meta-Learning and Few-shot Learning

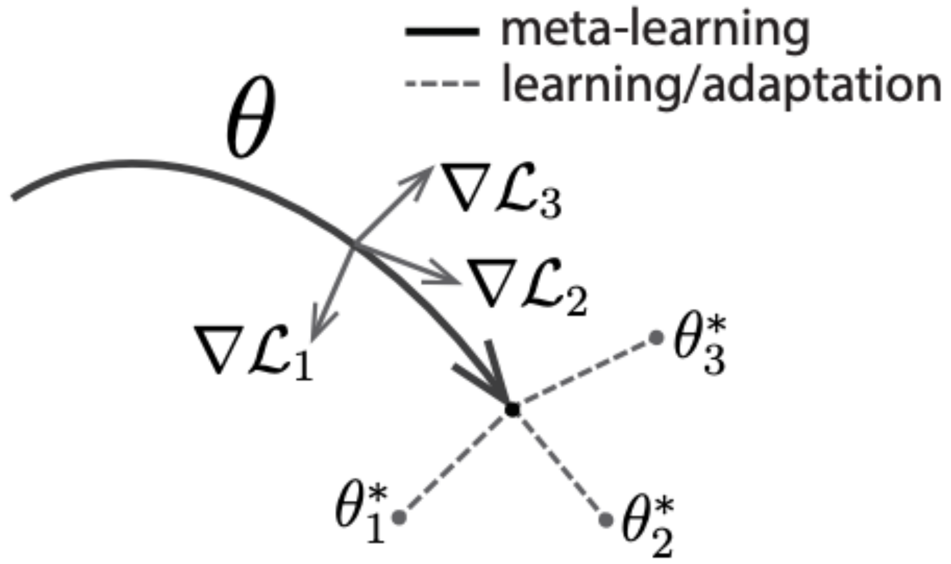
- Training task
- Test task



Lampert et al., 2009

Goal: Use the data from training tasks to
- extract useful features,
- build models (pretraining),
- learn a training routine,
- …

and use for the test task to
- learn from fewer datapoints, => Few-shot learning
- learn a more robust model,
- converge to a solution faster,
- …



Finn et al., 2017

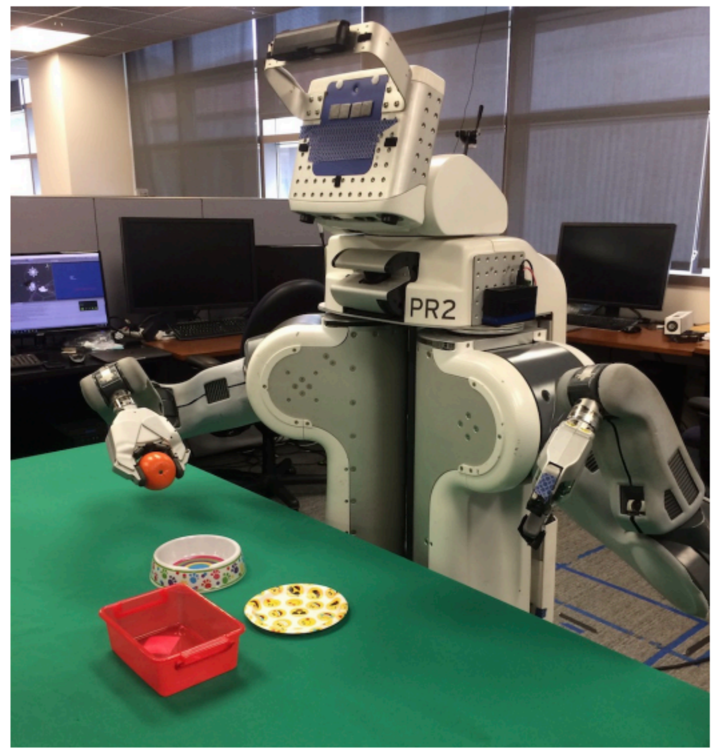


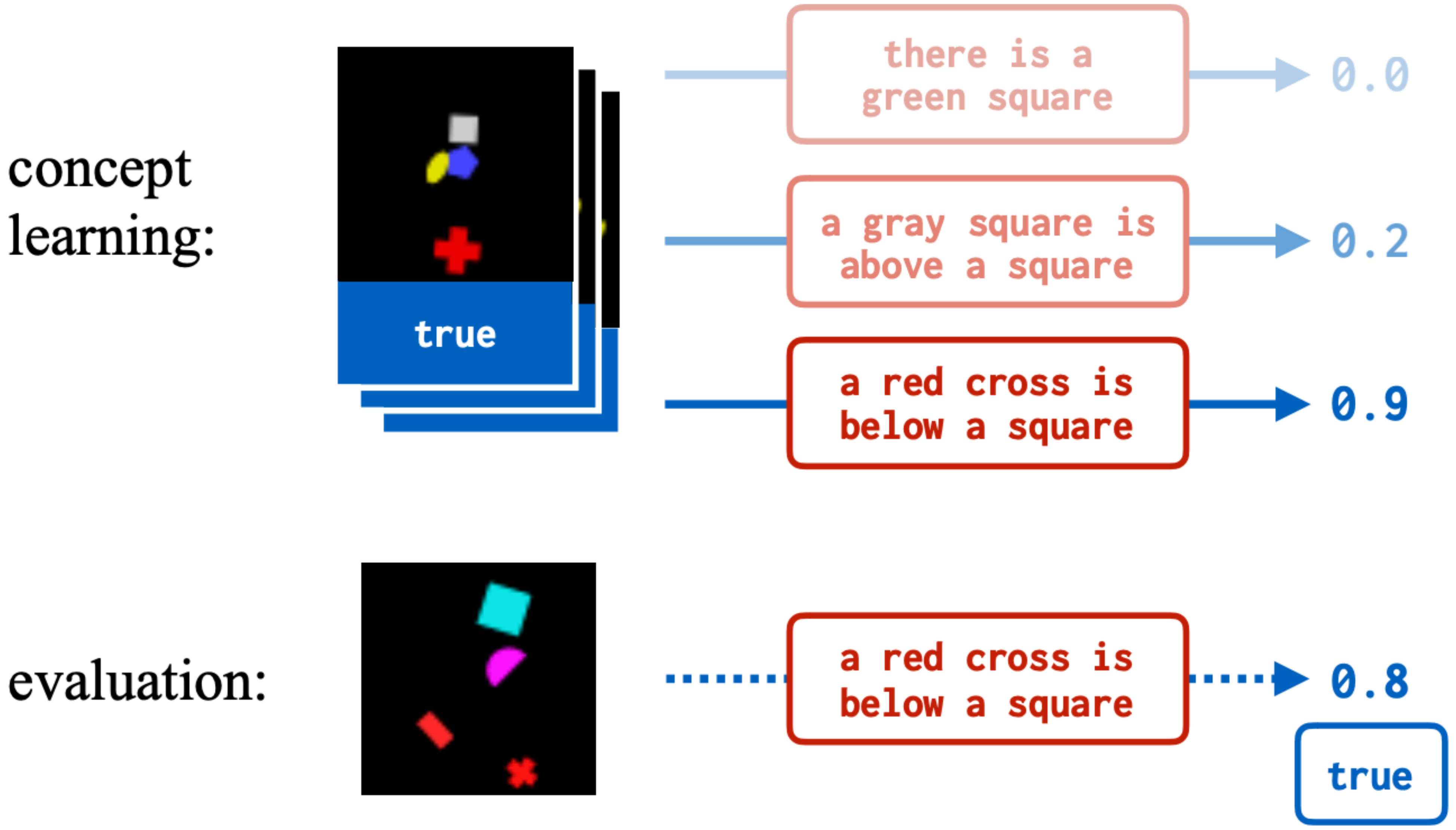
Figure 1: The robot learns to place a new object into a new container from a single demonstration.

Finn et al., 2017

Related Work

Language to Aid Learning

Narasimhan et al., 2018



Andreas et al., 2017

Outline

- Introduction
- Related Work
- Completed Work:
  - Using Natural Language for Reward Shaping in Reinforcement Learning (IJCAI 2019)
  - Guiding Reinforcement Learning by Mapping Pixels to Rewards (CoRL 2020)
  - Zero-shot Task Adaptation using Natural Language (arXiv, 2021)
- Proposed Work: Short-term
  - Neurosymbolic Model
  - Policy Adaptation
- Proposed Work: Long-term
  - Policy Regularization
  - Bayesian Inference
  - Supervised Attention

Sequential Decision Making

Reinforcement Learning

Task specification:
Designing reward functions

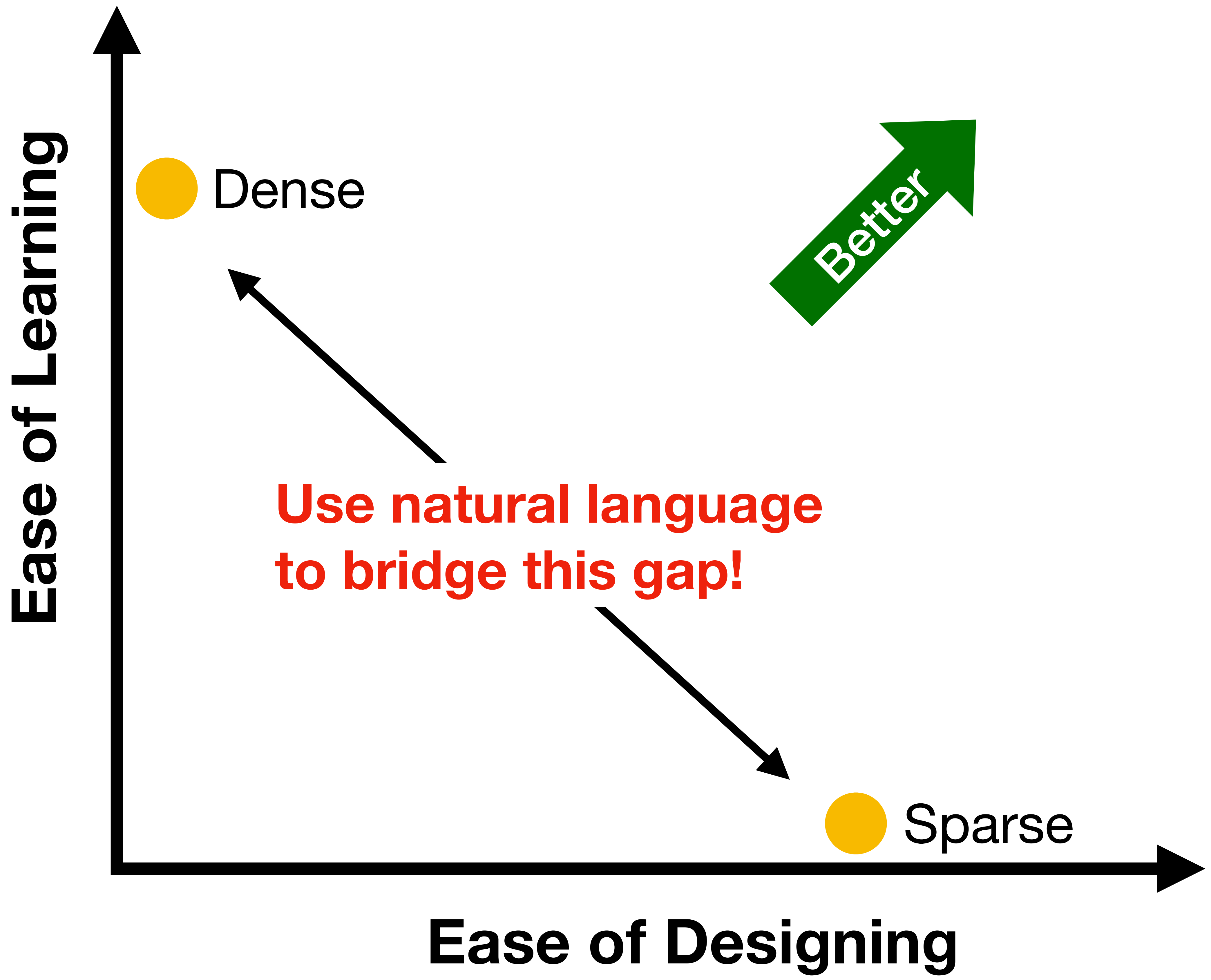
Imitation Learning

Task specification:
Providing demonstrations

Use natural language to aid these!

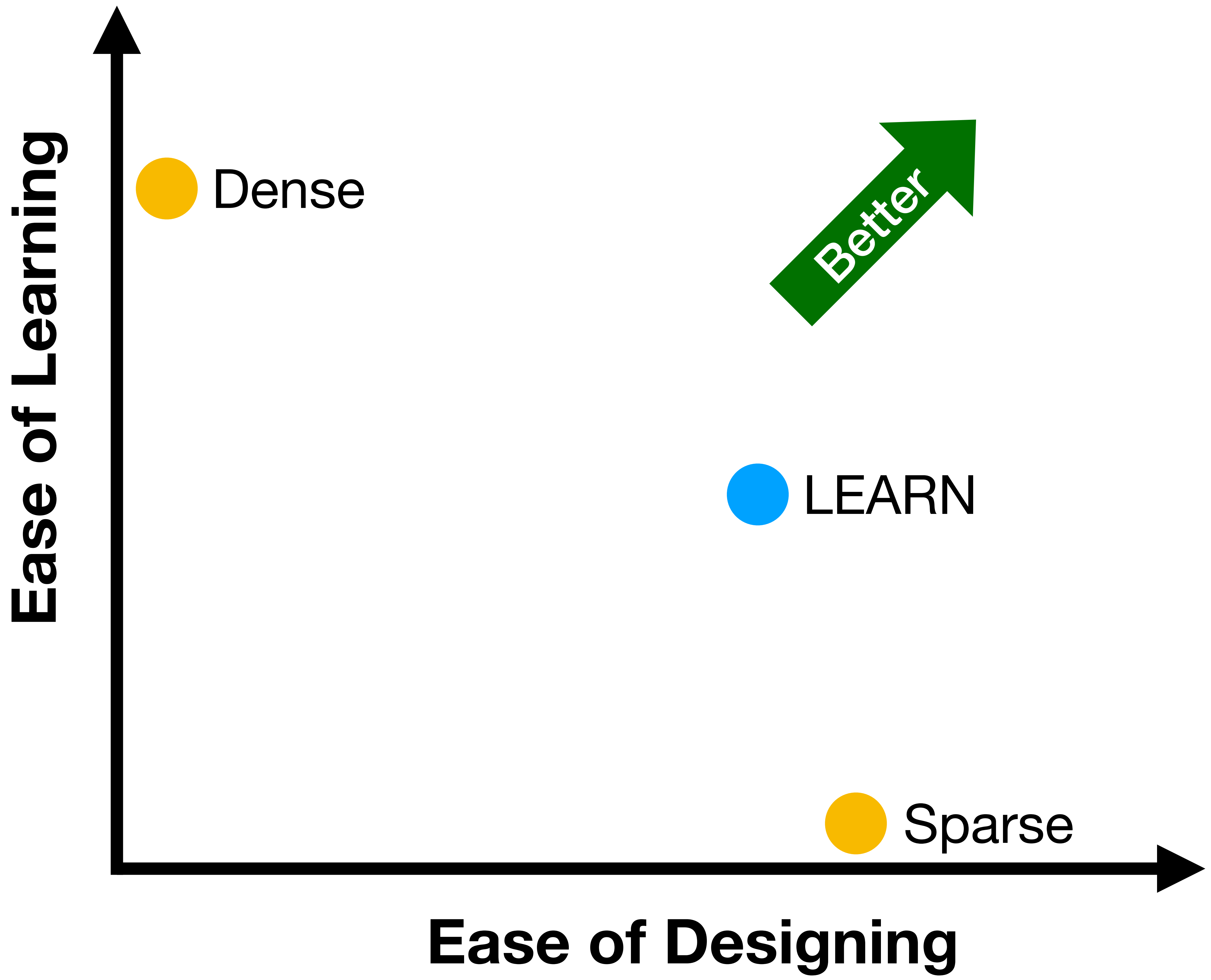
LanguagE-Action Reward Network (LEARN)

Motivation



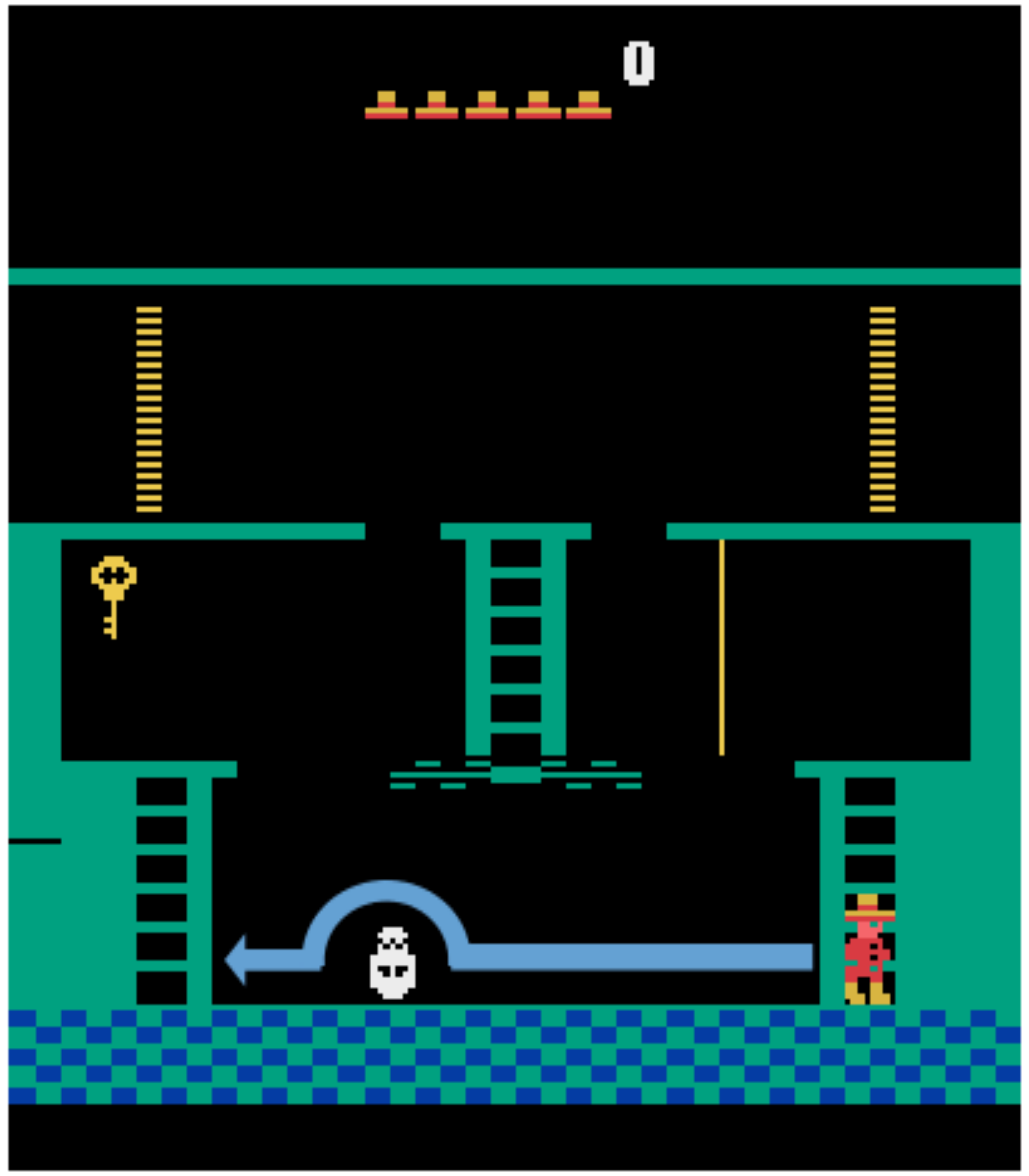
LanguagE-Action Reward Network (LEARN)

Motivation



LanguagE-Action Reward Network (LEARN)

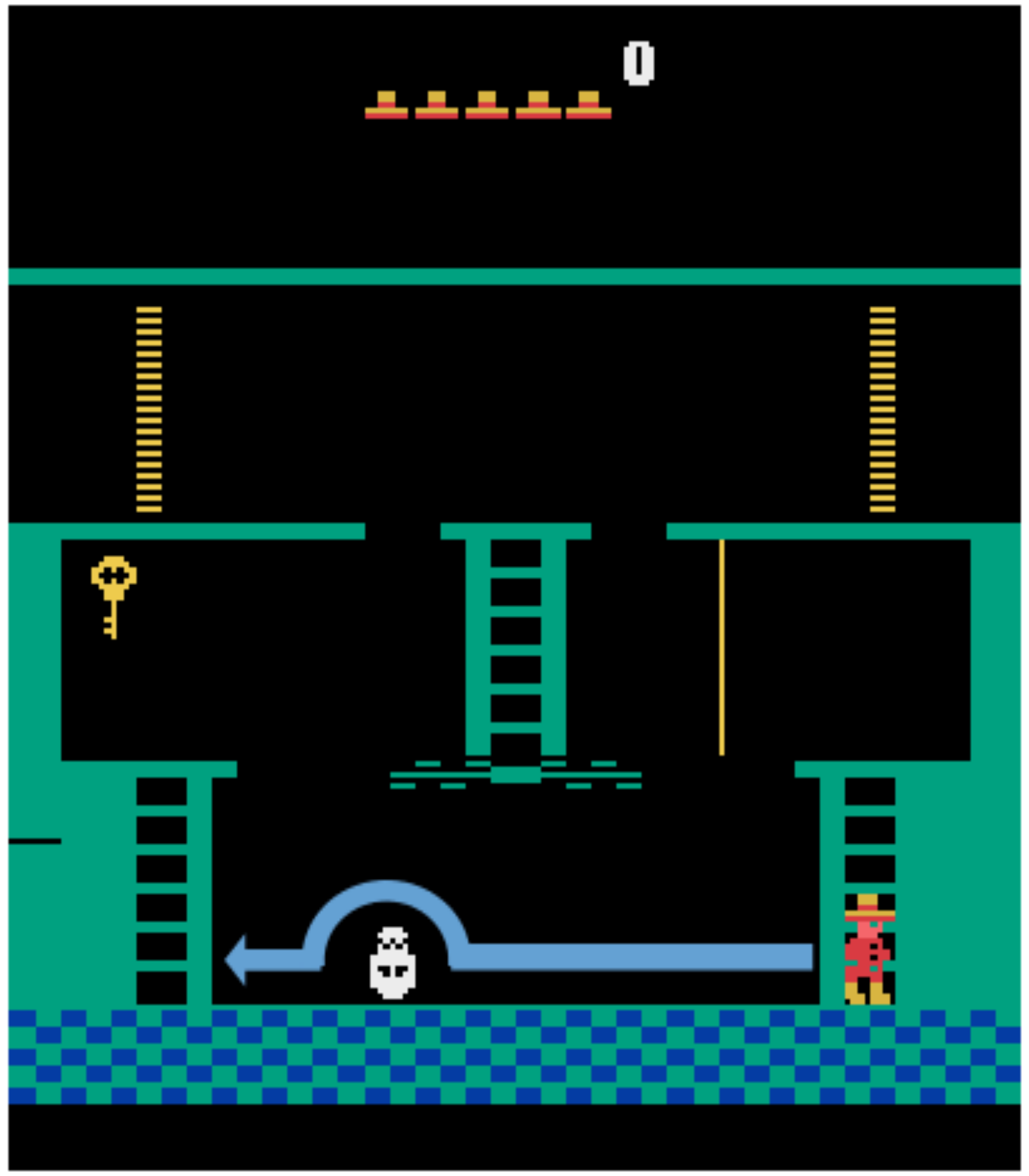
Motivation



[Bellemare et al., 2013]

LanguagE-Action Reward Network (LEARN)

Motivation



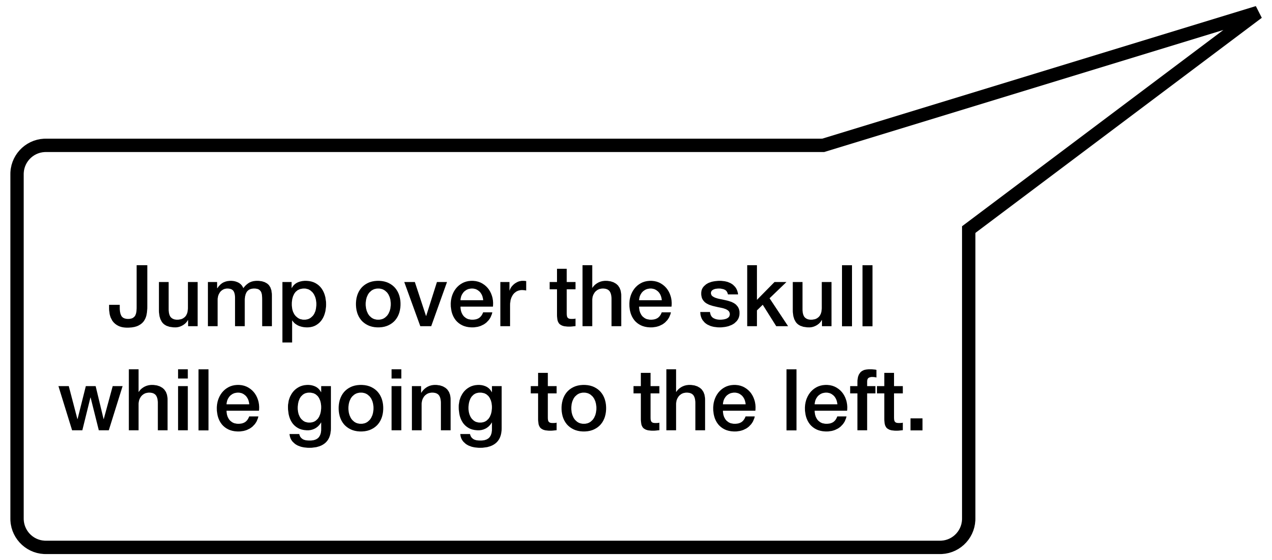
Jump over the skull while going to the left.

Can we use natural language to provide intermediate rewards to the agent?

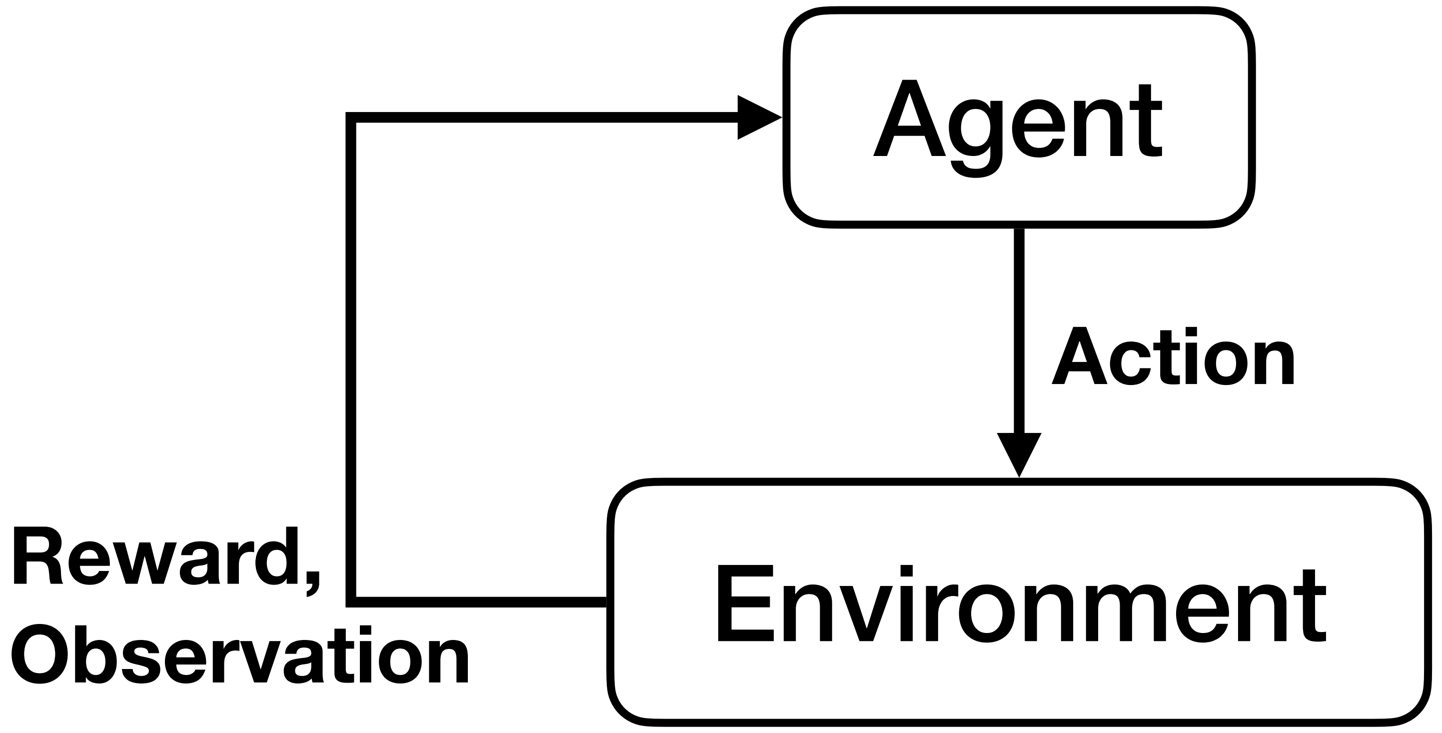
[Bellemare et al., 2013]

LanguagE-Action Reward Network (LEARN)

Approach

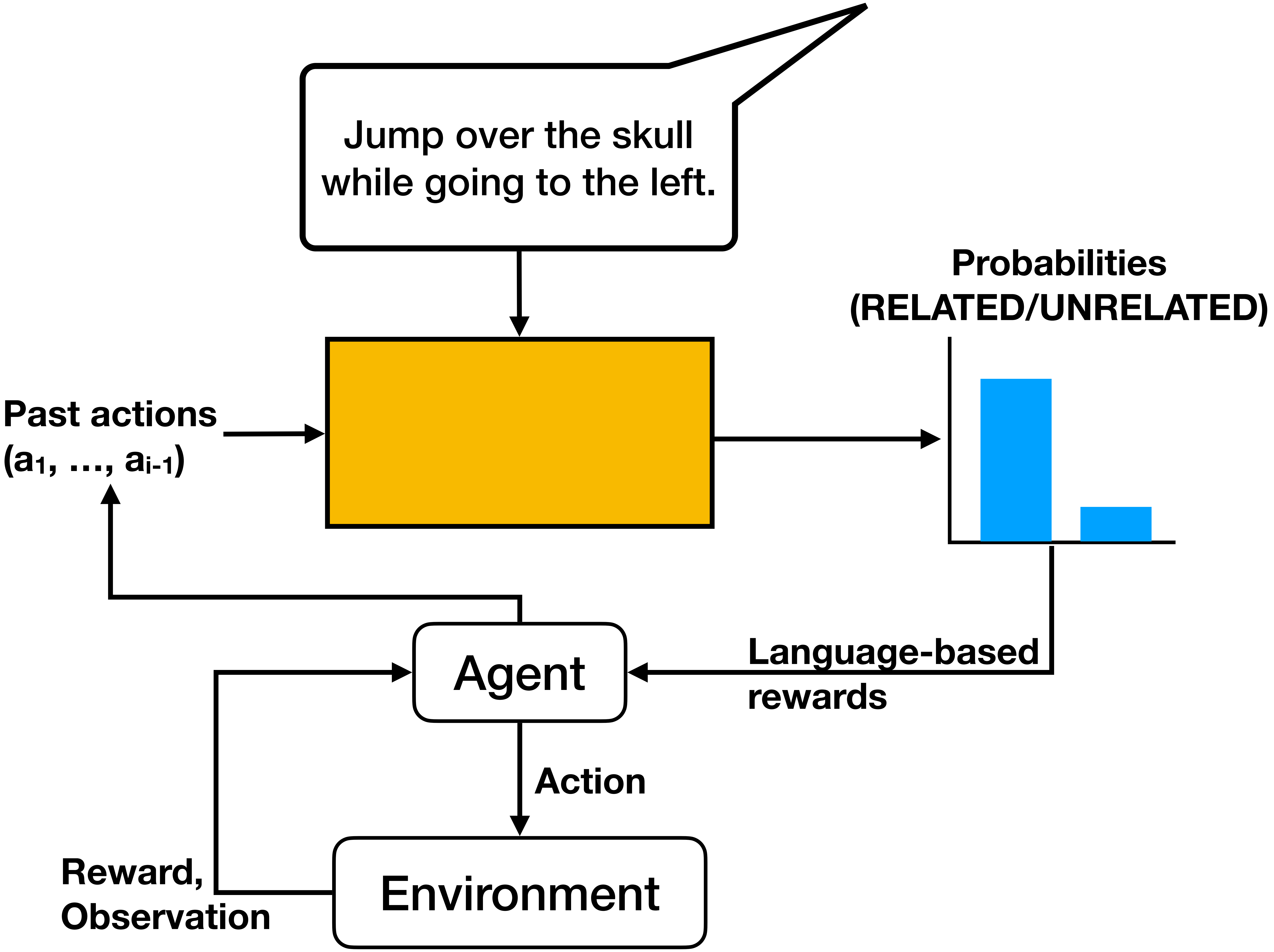


- Standard RL setup, plus a natural language command describing the task.



LanguagE-Action Reward Network (LEARN)

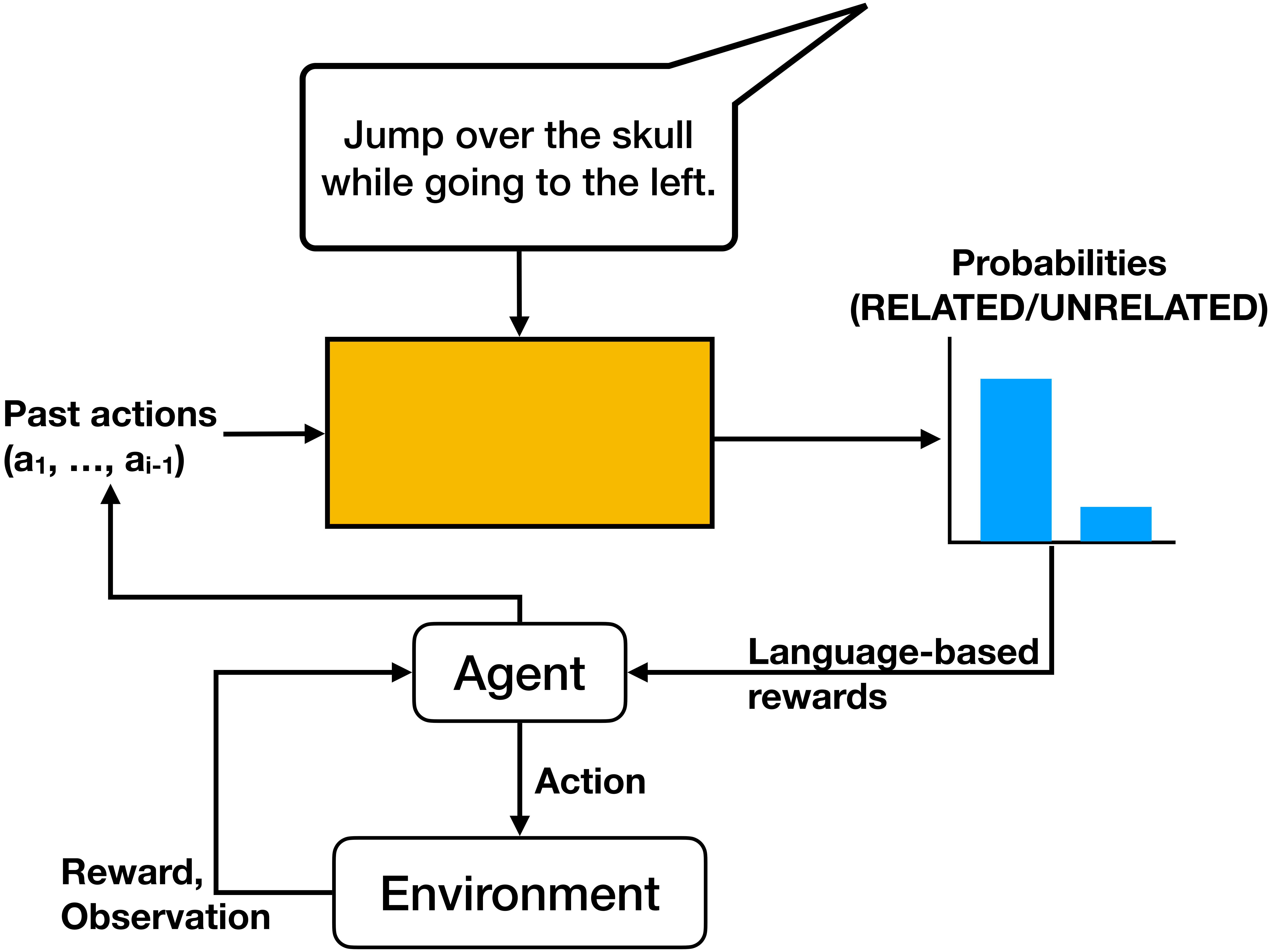
Approach



- Standard RL setup, plus a natural language command describing the task.
- Use the agent's past actions and the command to generate additional rewards.

LanguagE-Action Reward Network (LEARN)

Approach



- Standard RL setup, plus a natural language command describing the task.
- Use the agent's past actions and the command to generate additional rewards.

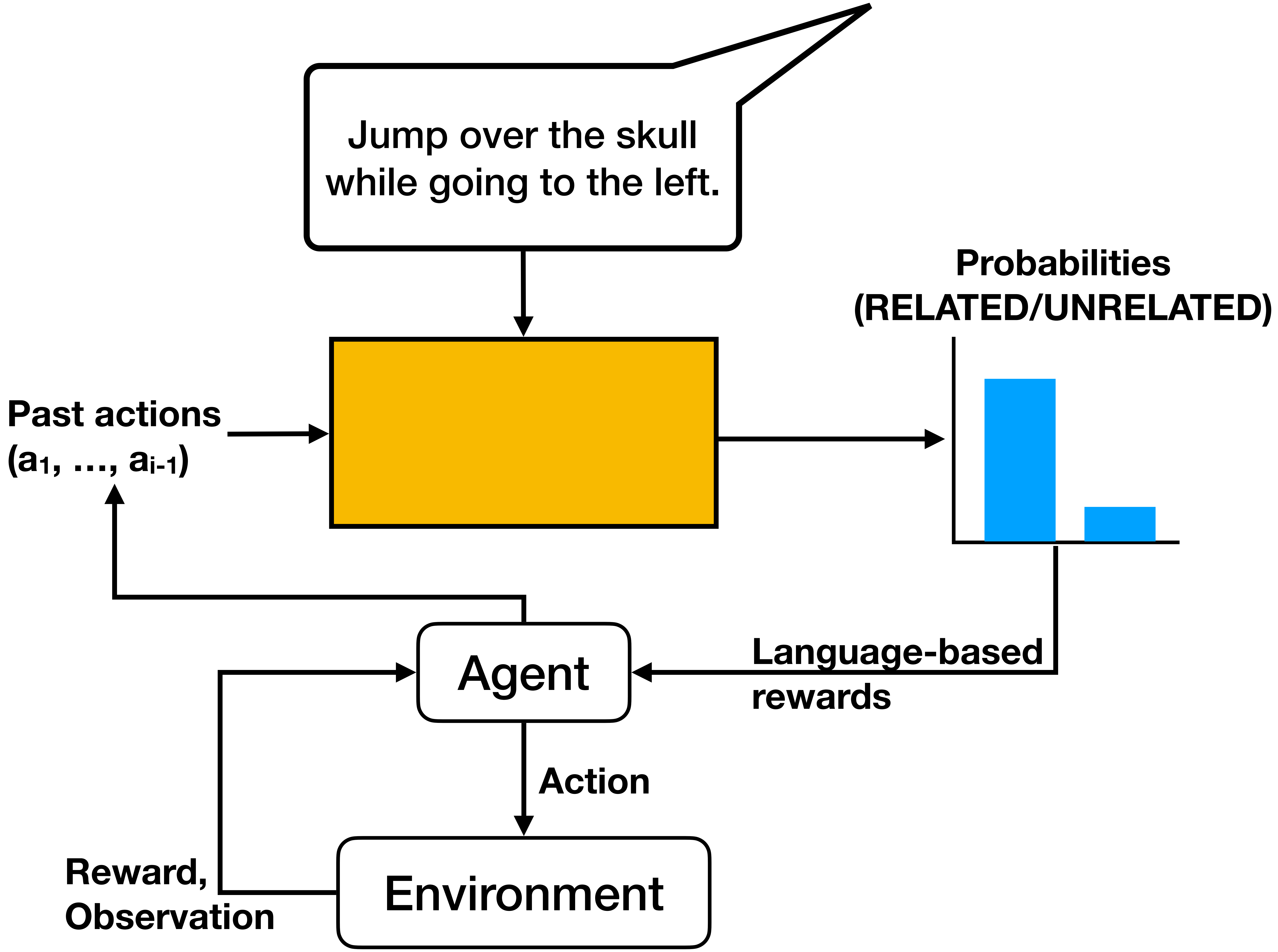
For example,

Past actions		**Reward**
`LLLJLLL`	→	High
`RRRUULL`	→	Low

[L: Left, R: Right, U: Up, J: Jump]

LanguagE-Action Reward Network (LEARN)

Approach



- Standard RL setup, plus a natural language command describing the task.
- Use the agent's past actions and the command to generate additional rewards.

For example,

Past actions		Reward
4441444	→	High
3332244	→	Low

[4: Left, 3: Right, 2: Up, 1: Jump]

LanguagE-Action Reward Network (LEARN)

Approach

Problem: Given a sequence of actions (e.g. 4441444) and a command (e.g. “Jump over the skull while going to the left”), are they related?

Using the sequence of actions, generate an *action-frequency vector*:

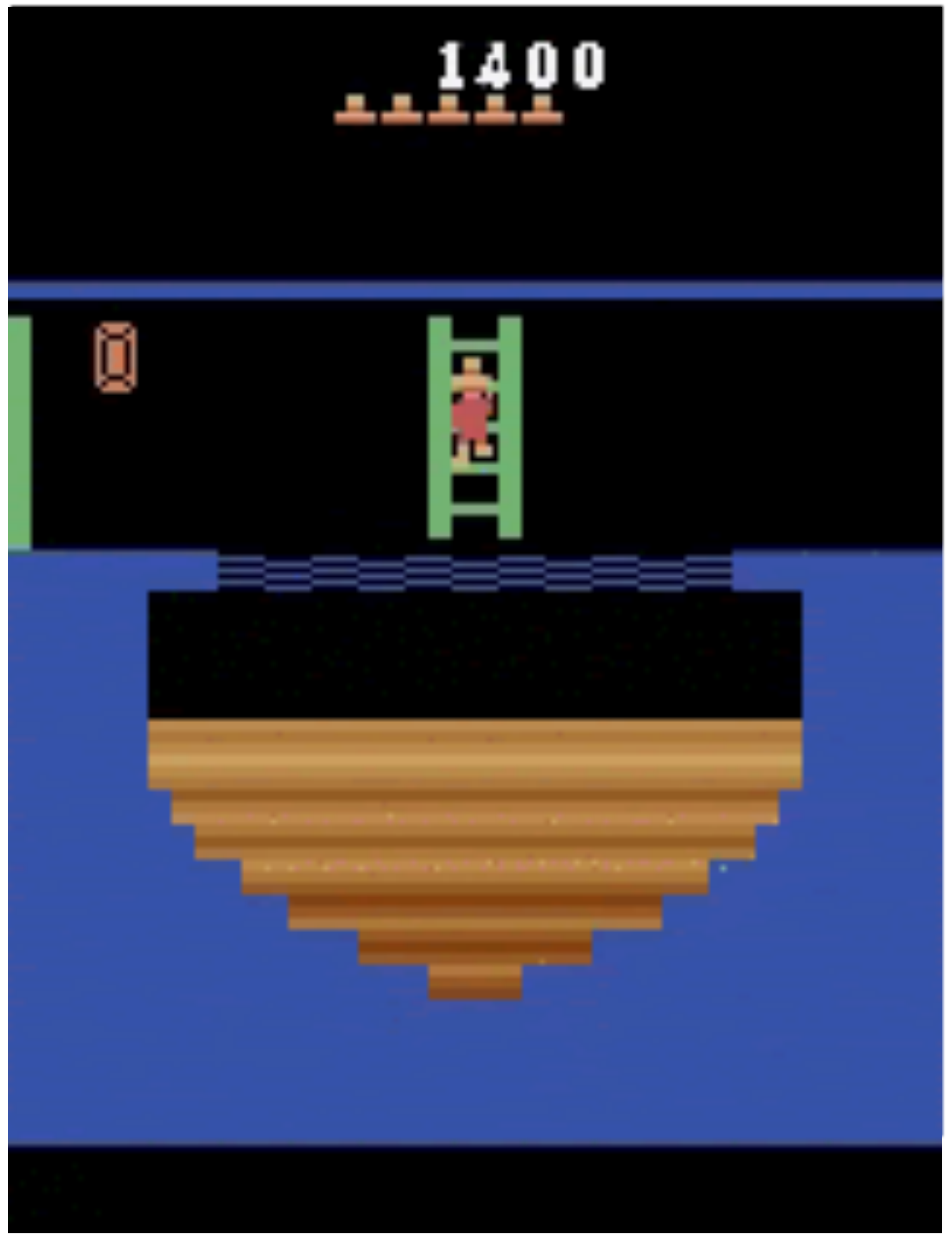
ϵ	⇒	[0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00]
4	⇒	[0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00]
42	⇒	[0.00	0.00	0.50	0.00	0.50	0.00	0.00	0.00]
422	⇒	[0.00	0.00	0.67	0.00	0.33	0.00	0.00	0.00]

Train a neural network — LanguagE Action Reward Network (LEARN) — that takes in the action-frequency vector and the command to predict whether they are related or not.

LanguagE-Action Reward Network (LEARN)

Data Collection

Clip 1:



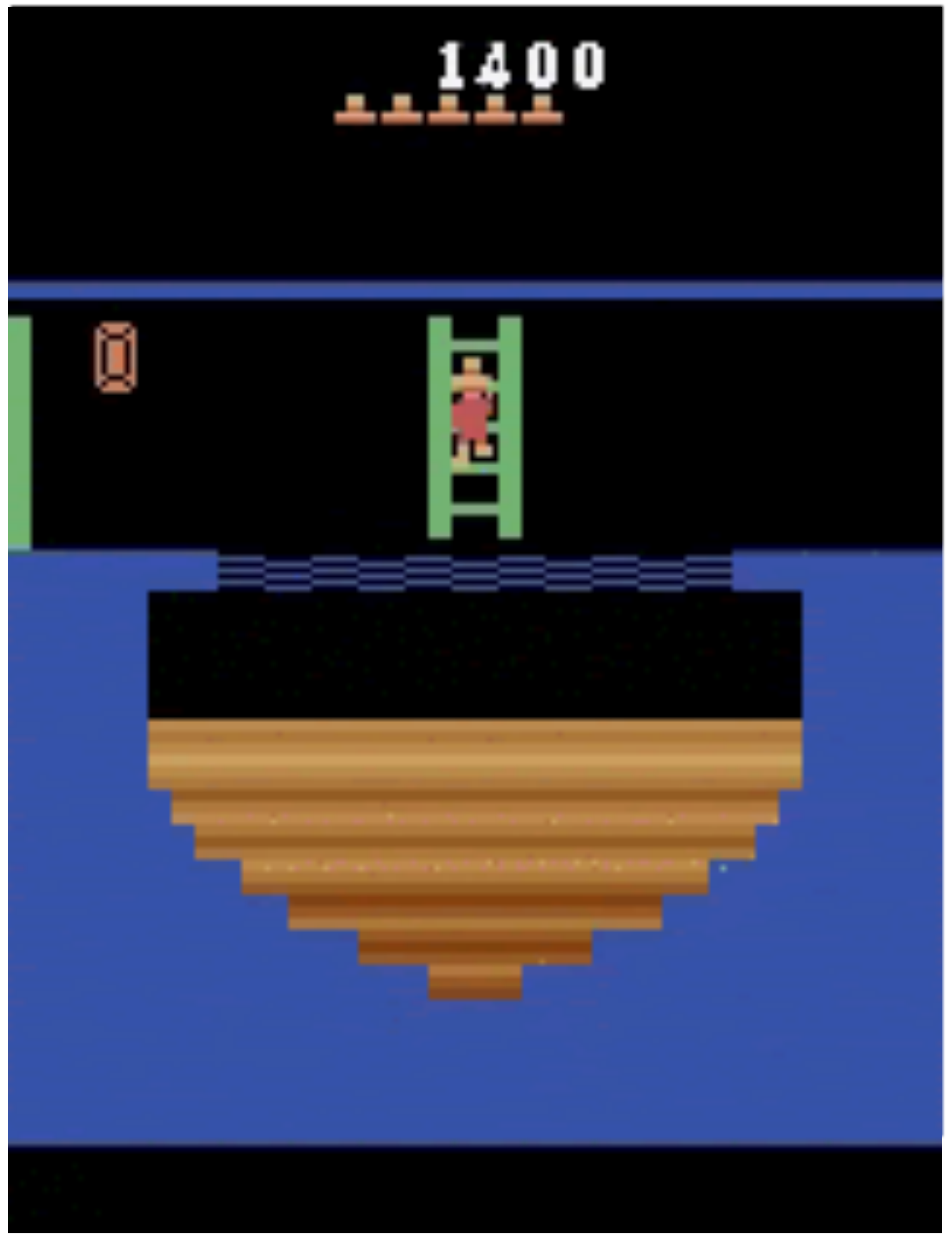
Please enter the description below:

[Kurin et al., 2017]

LanguagE-Action Reward Network (LEARN)

Data Collection

Clip 1:



Please enter the description below:

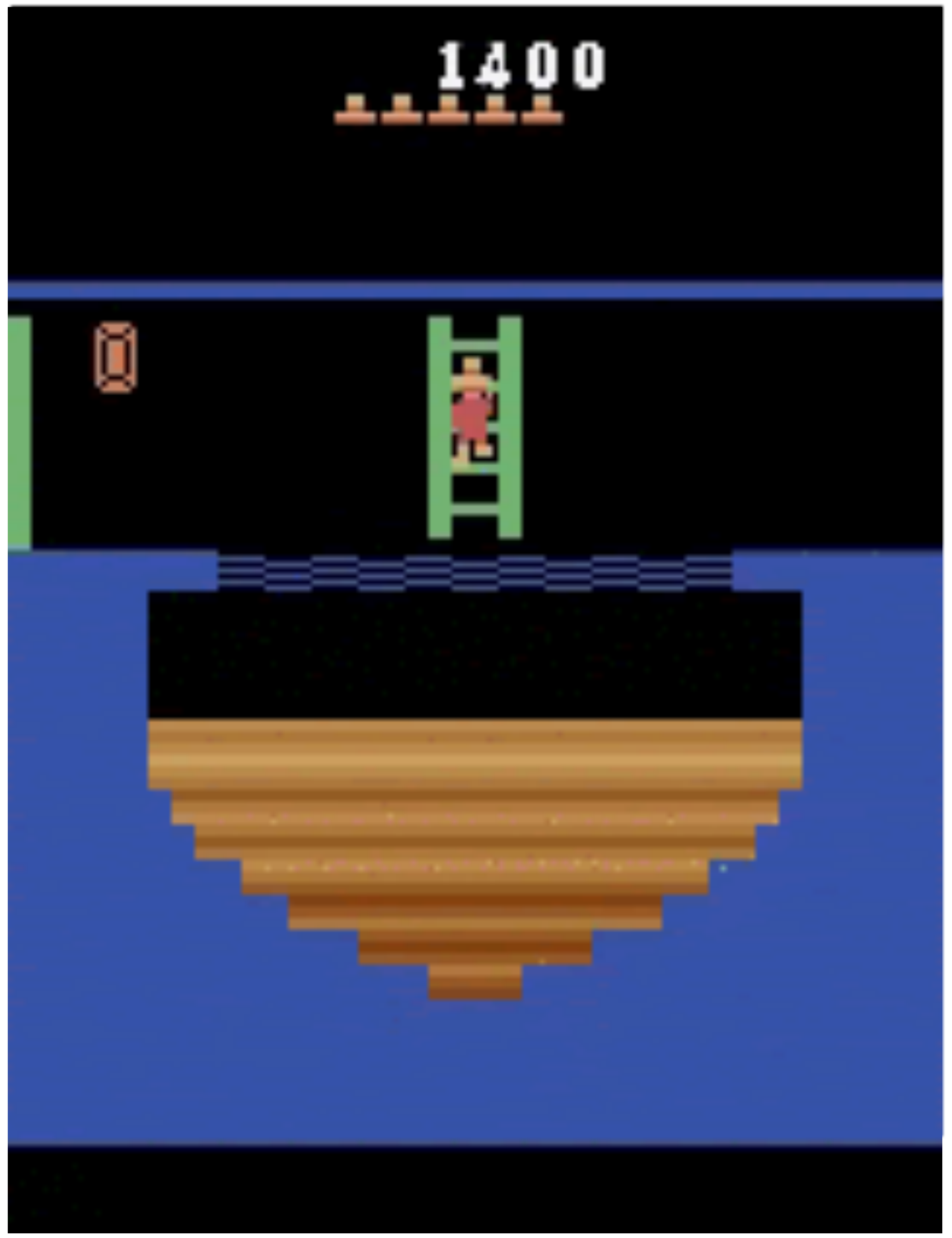
1.	wait
2.	using the ladder on standing
3.	going slow and climb down the ladder
4.	move down the ladder and walk left
5.	go left watch the trap and move on
6.	climbling down the ladder
7.	ladder dwon and running this away
8.	stay in place on the ladder.
9.	go down the ladder
10.	go right and climb up the ladder
11.	just jump and little move to right side
12.	run all the way to the left.
13.	go left jumping once
14.	go left
15.	move right and jump over green creature then go down the ladder
16.	hop over to the middle ledge
17.	wait for the two skulls and dodge them in the middle
18.	walk to the left and then jump down
19.	jump to collected gold coin and little move
20.	wait for the platform to materialize then walk and leap to your right to collect the coins.

[Kurin et al., 2017]

LanguagE-Action Reward Network (LEARN)

Data Collection

Clip 1:



Please enter the description below:

Ill-formed

Spelling errors

1.	wait
2.	using the ladder on standing
3.	going slow and climb down the ladder
4.	move down the ladder and walk left
5.	go left watch the trap and move on
6.	climbling down the ladder
7.	ladder dwon and running this away
8.	stay in place on the ladder.
9.	go down the ladder
10.	go right and climb up the ladder
11.	just jump and little move to right side
12.	run all the way to the left.
13.	go left jumping once
14.	go left
15.	move right and jump over green creature then go down the ladder
16.	hop over to the middle ledge
17.	wait for the two skulls and dodge them in the middle
18.	walk to the left and then jump down
19.	jump to collected gold coin and little move
20.	wait for the platform to materialize then walk and leap to your right to collect the coins.

[Kurin et al., 2017]

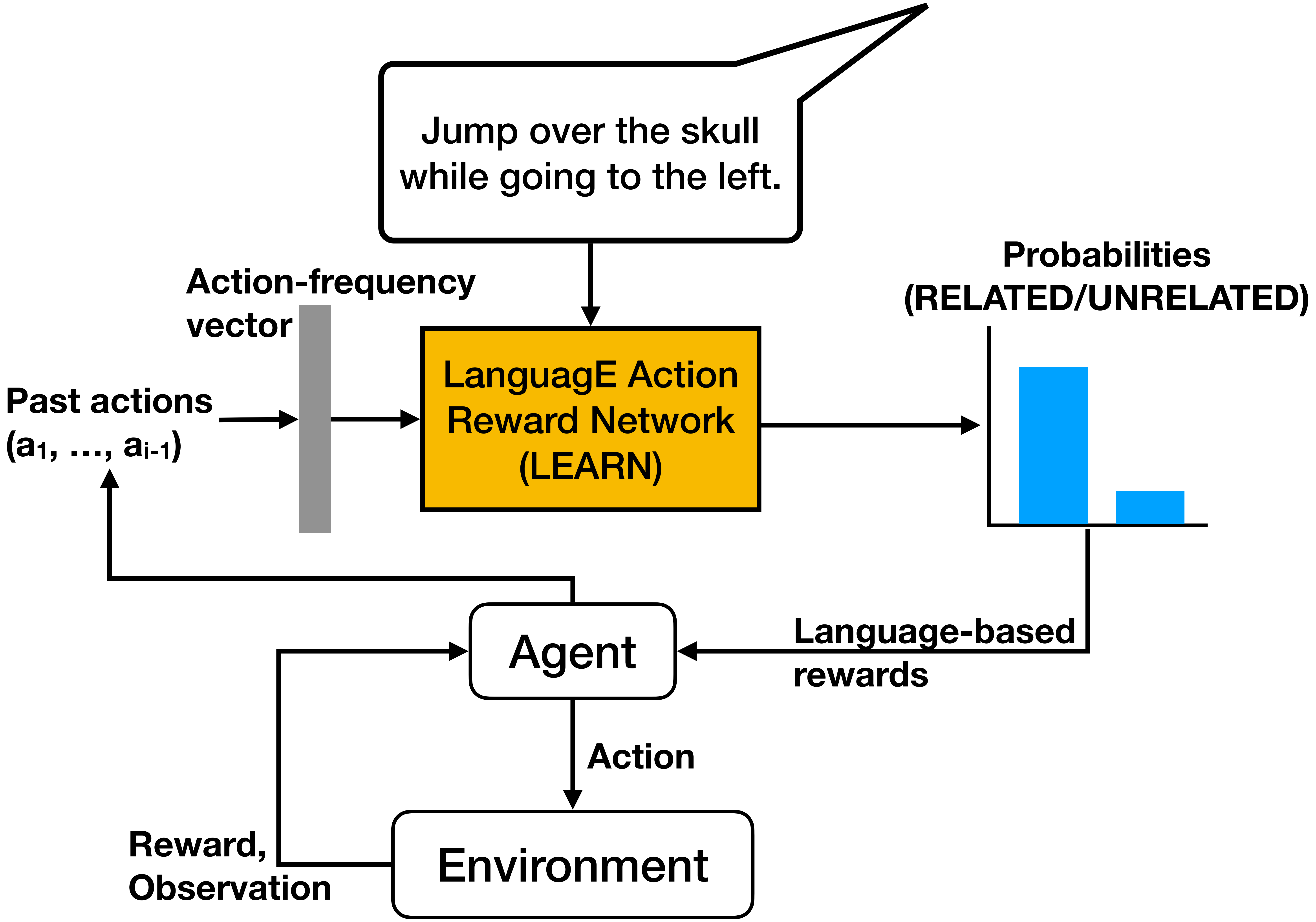
LanguagE-Action Reward Network (LEARN)

Training the Model

Supervised Learning:
- Binary classification: Related vs Unrelated
- Positive examples: Action-frequency vectors and corresponding language
  Negative examples: Random pairs

LanguagE-Action Reward Network (LEARN)

Putting it all together…

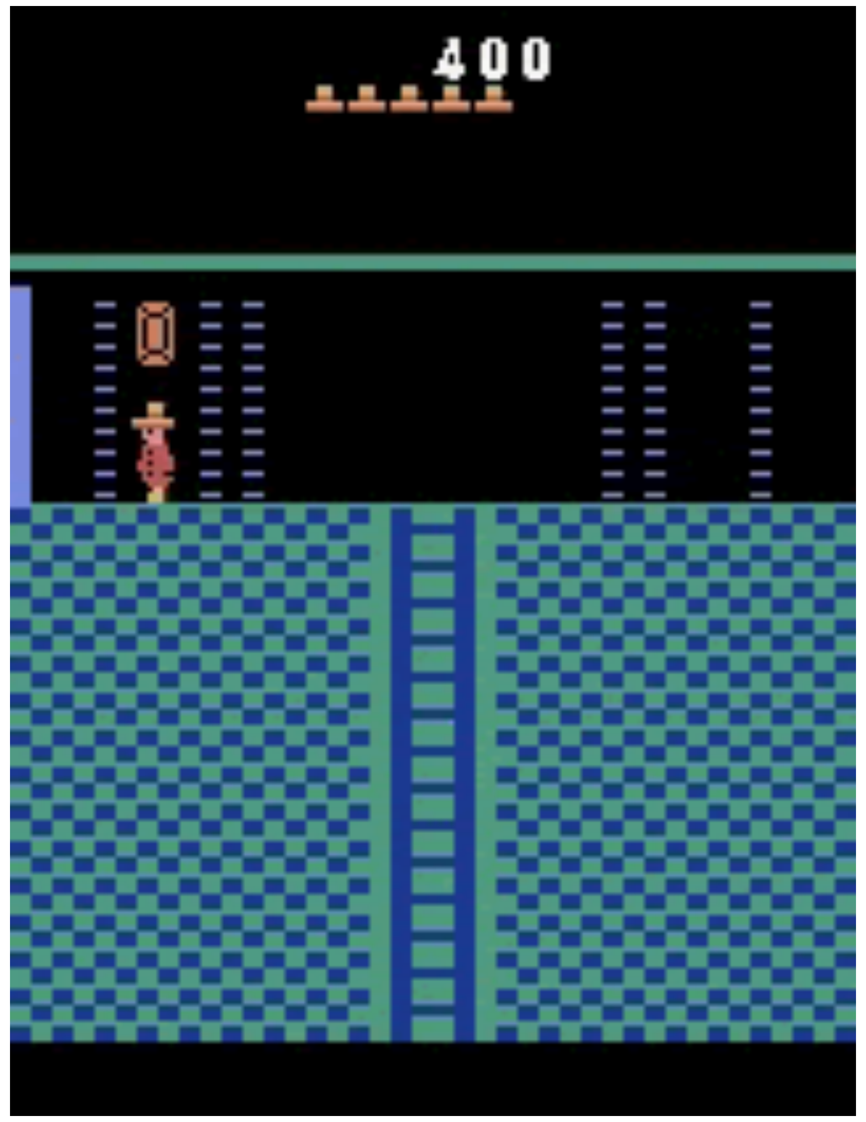


- Using the agent's past actions, generate an action-frequency vector.
- LEARN: Scores the relatedness between the action-frequency vector and the language command.
- Use the relatedness scores as language-based rewards. Defined using a potential function => optimal policy does not change [Ng et al., 1999].

LanguagE-Action Reward Network (LEARN)

Experiments

15 tasks, with natural language descriptions collected using Amazon Mechanical Turk.

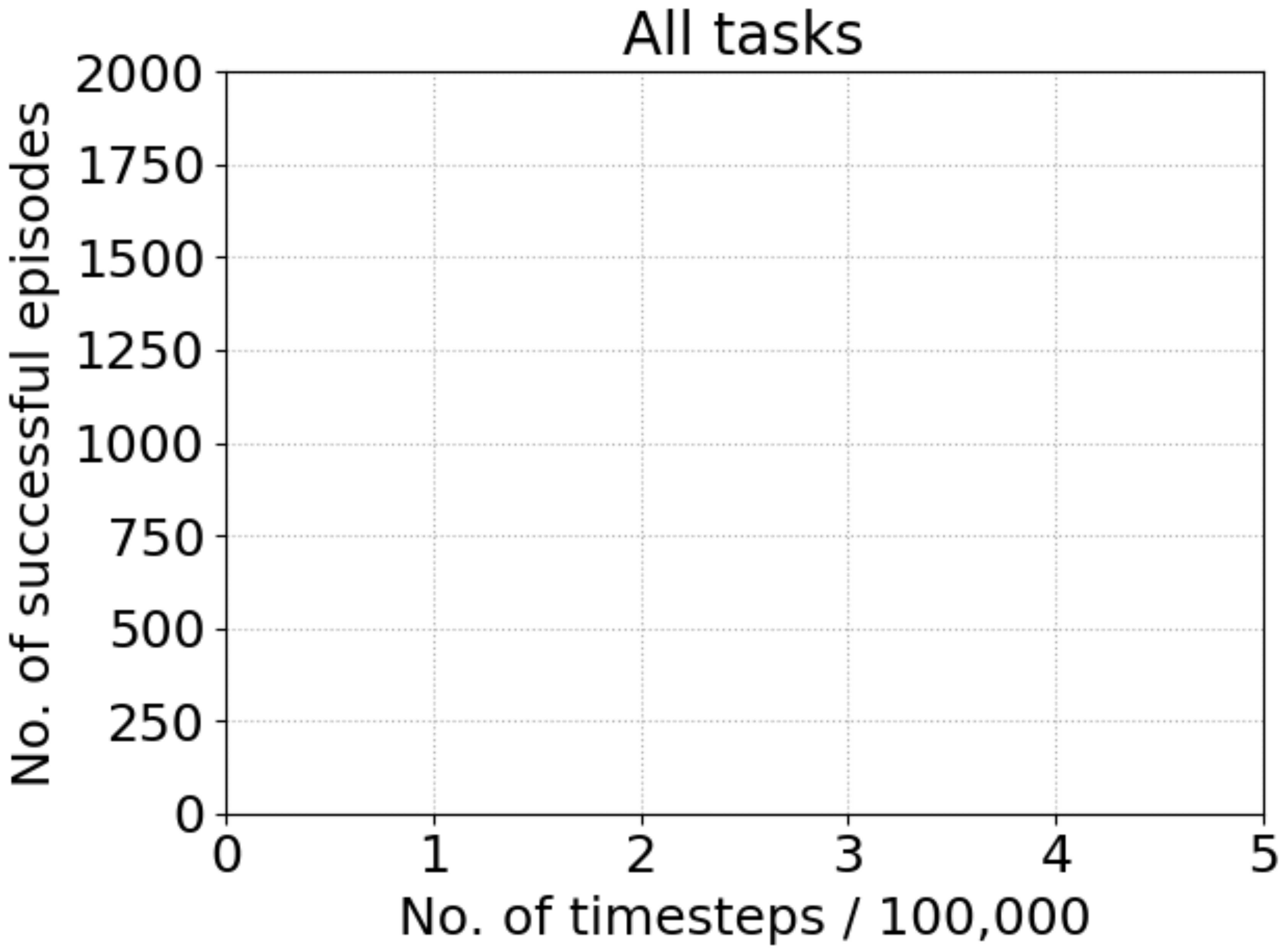


- Jump to take bonus walk right and left the climb downwards in ladder.
- Jump Pick Up The Coin And Down To Step The Ladder
- jump up to get the item and go to the right

LanguagE-Action Reward Network (LEARN)

Results

- Compared RL training using PPO algorithm with and without language-based reward.

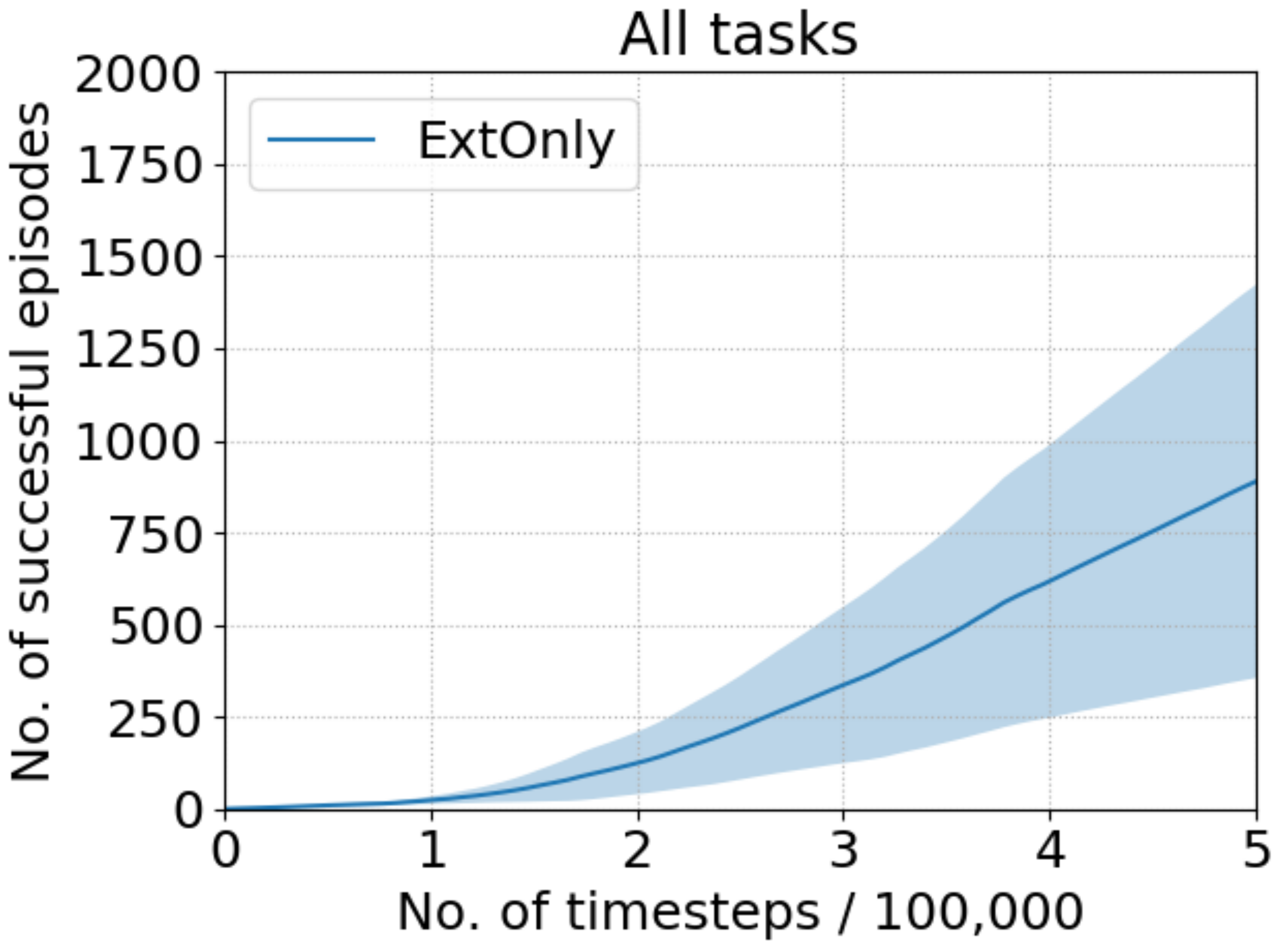


LanguagE-Action Reward Network (LEARN)

Results

- Compared RL training using PPO algorithm with and without language-based reward.

- ExtOnly: Reward of 1 for reaching the goal, reward of 0 in all other cases.



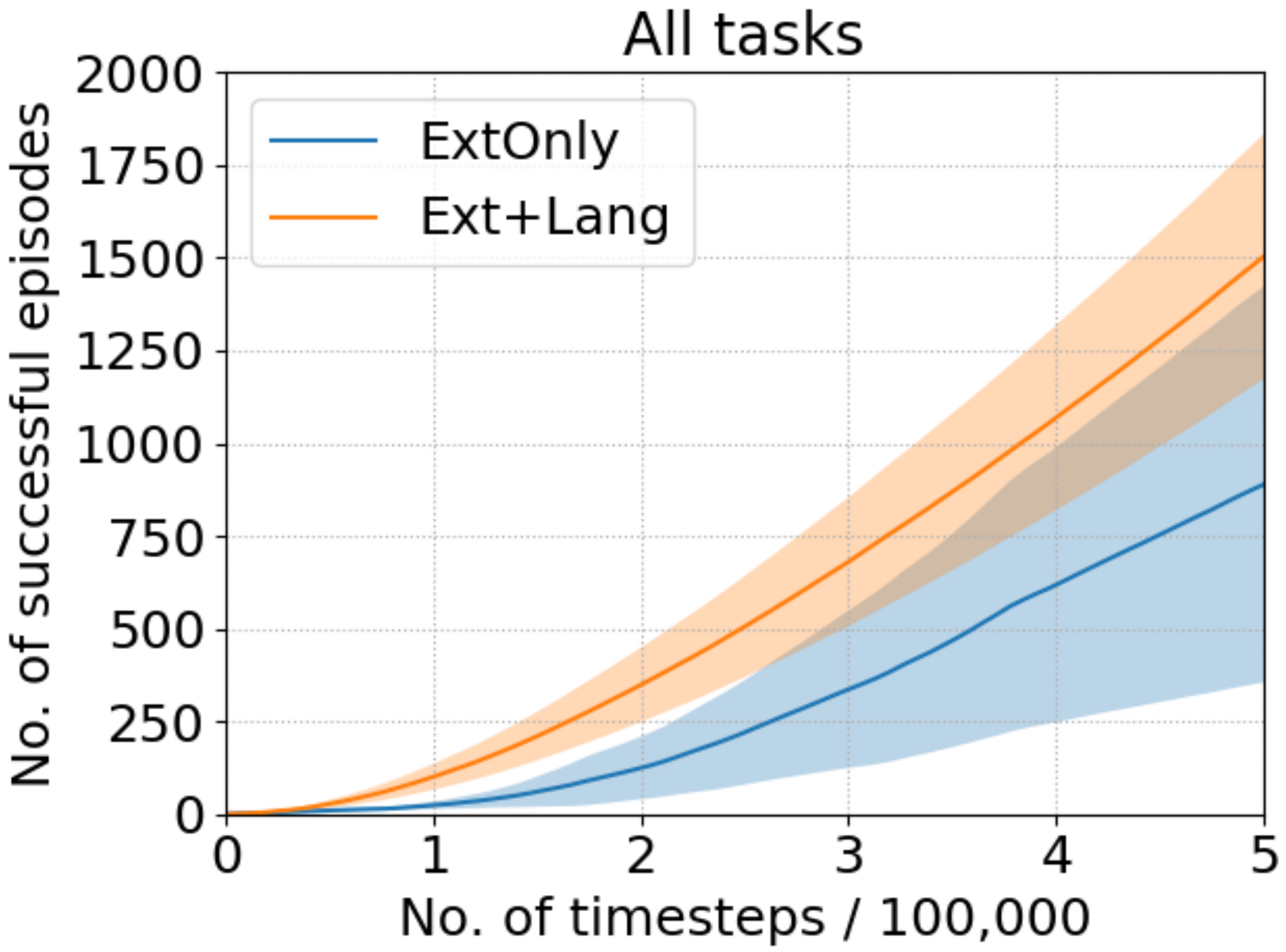
LanguagE-Action Reward Network (LEARN)

Results

- Compared RL training using PPO algorithm with and without language-based reward.

- ExtOnly: Reward of 1 for reaching the goal, reward of 0 in all other cases.

- Ext+Lang: Extrinsic reward plus language-based intermediate rewards.



Outline

- Introduction
- Related Work
- Completed Work:
    - Using Natural Language for Reward Shaping in Reinforcement Learning (IJCAI 2019)
    - Guiding Reinforcement Learning by Mapping Pixels to Rewards (CoRL 2020)
    - Zero-shot Task Adaptation using Natural Language (arXiv, 2021)
- Proposed Work: Short-term
    - Neurosymbolic Model
    - Policy Adaptation
- Proposed Work: Long-term
    - Policy Regularization
    - Bayesian Inference
    - Supervised Attention

Pixels and Language to Rewards (PixL2R)

Motivation

LEARN results in efficient policy learning, but

Pixels and Language to Rewards (PixL2R)

Motivation

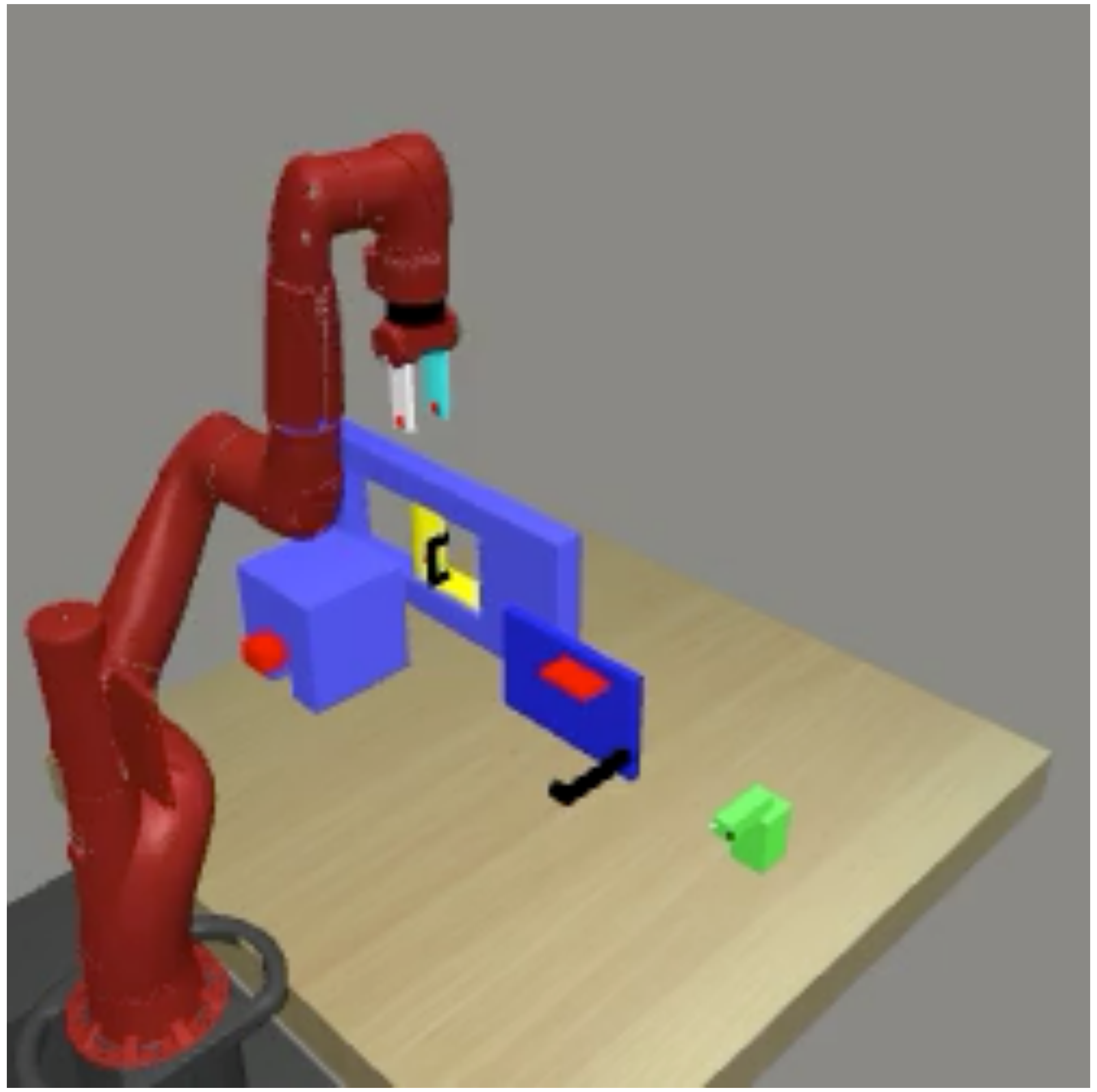
LEARN results in efficient policy learning, but
- the action-frequency vector is undefined for continuous action spaces
- discards temporal information in action sequences
- does not use state information

Pixels and Language to Rewards (PixL2R)

MetaWorld Domain

Robot table-top manipulation:

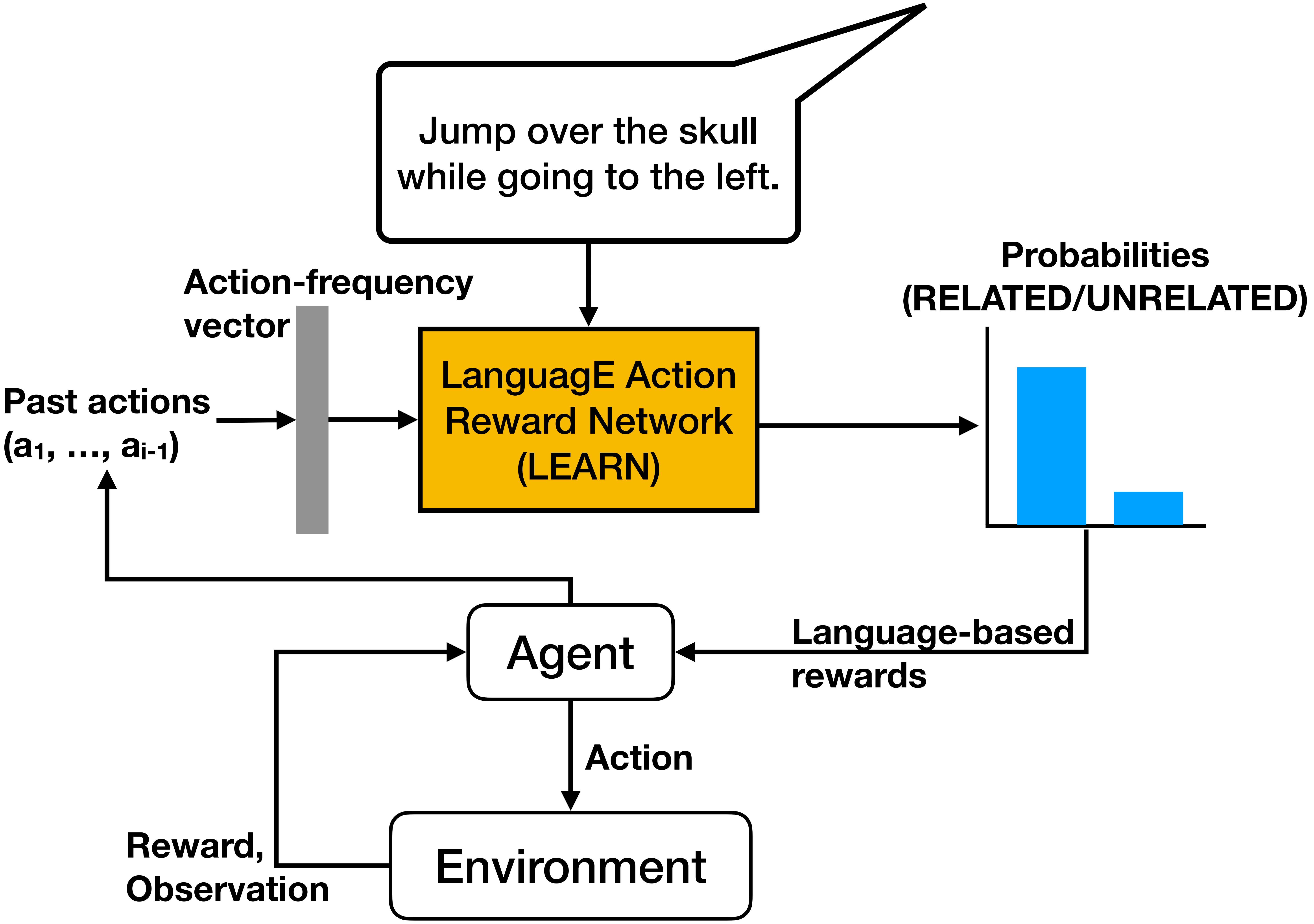
- Multiple objects in the scene
- Goal: Interact with a pre-selected object



[Yu et al., 2019]

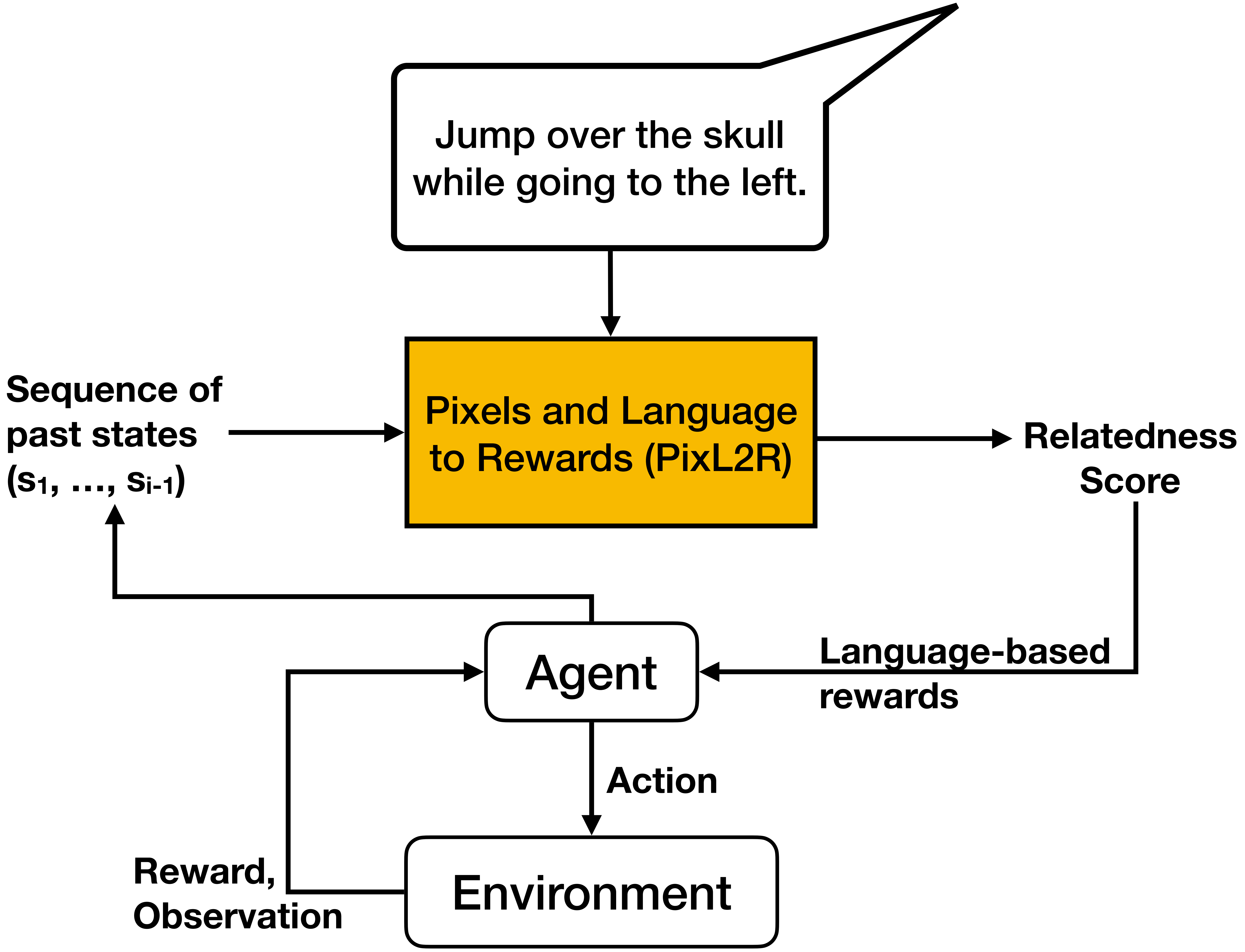
Pixels and Language to Rewards (PixL2R)

Approach



Pixels and Language to Rewards (PixL2R)

Approach



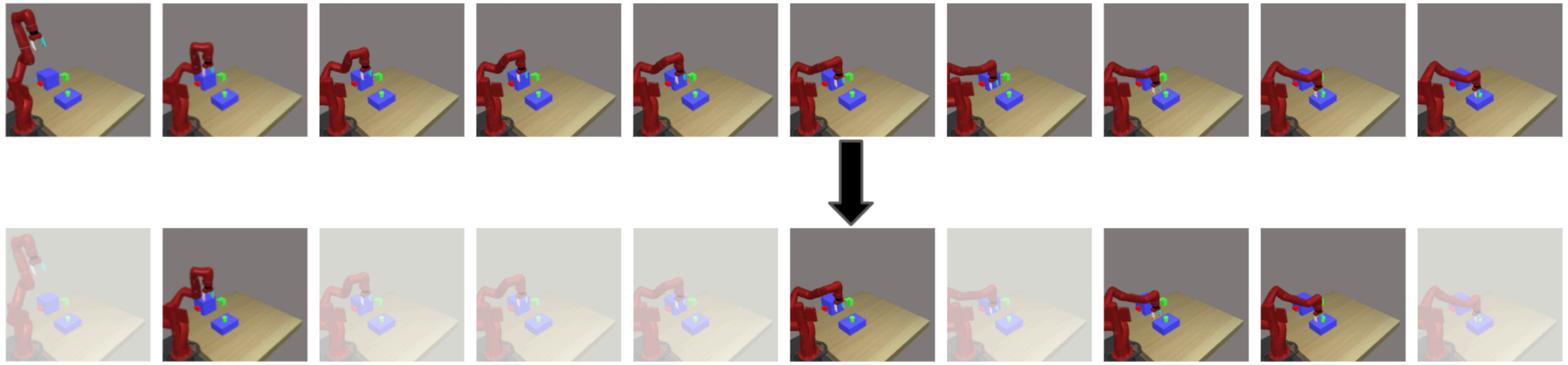
Pixels and Language to Rewards (PixL2R)

Data Augmentation

Pixels and Language to Rewards (PixL2R)

Data Augmentation

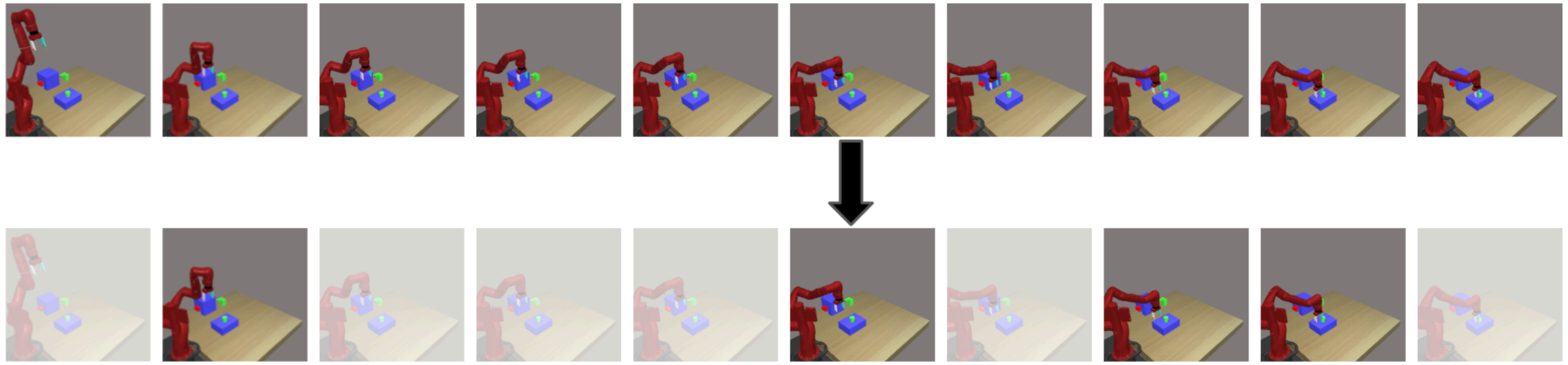
- Frame dropping



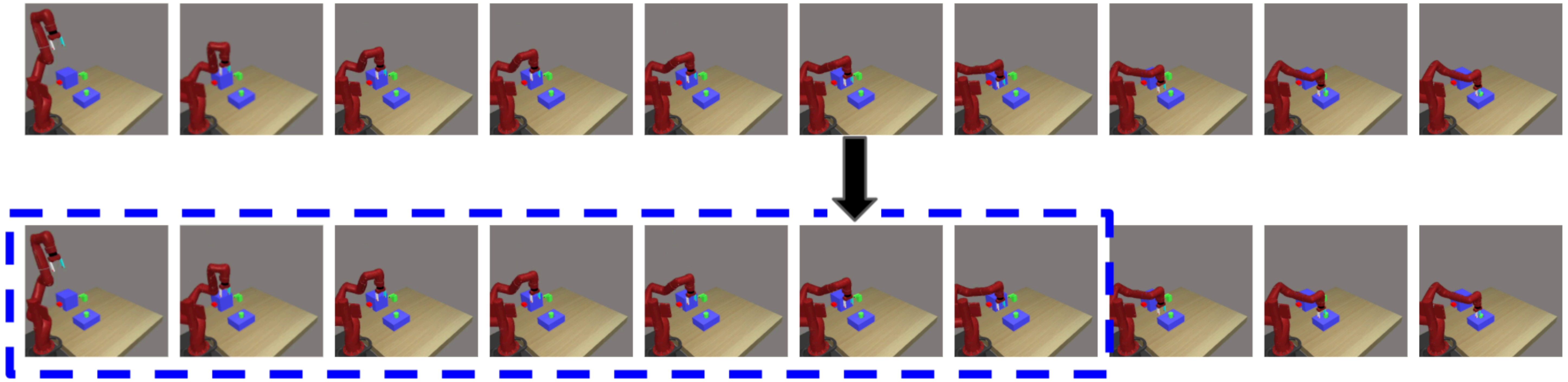
Pixels and Language to Rewards (PixL2R)

Data Augmentation

- Frame dropping



- Partial trajectories



Pixels and Language to Rewards (PixL2R)

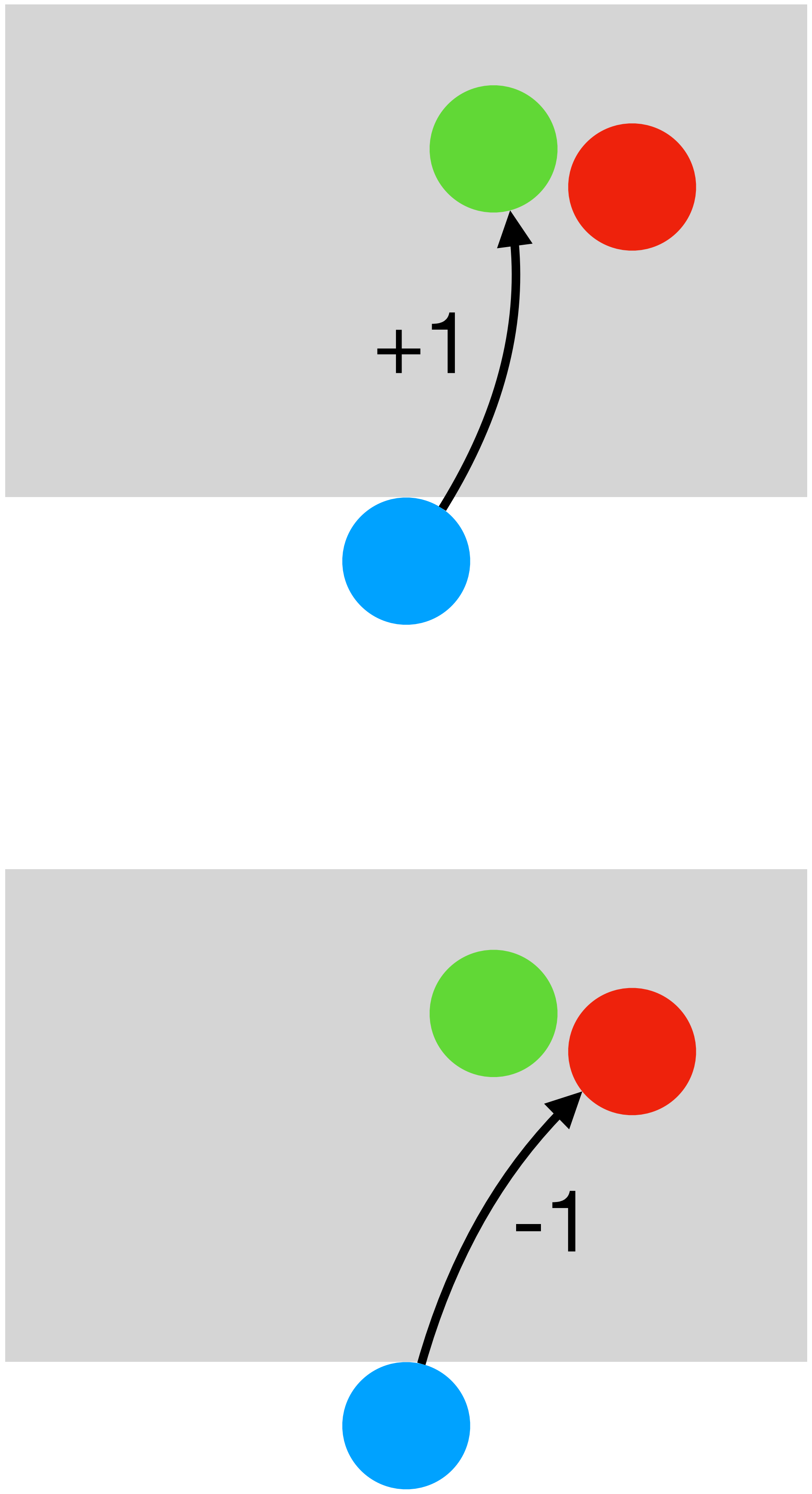
Training Objective

Classification:

Starting position

Correct object

Incorrect object



Pixels and Language to Rewards (PixL2R)

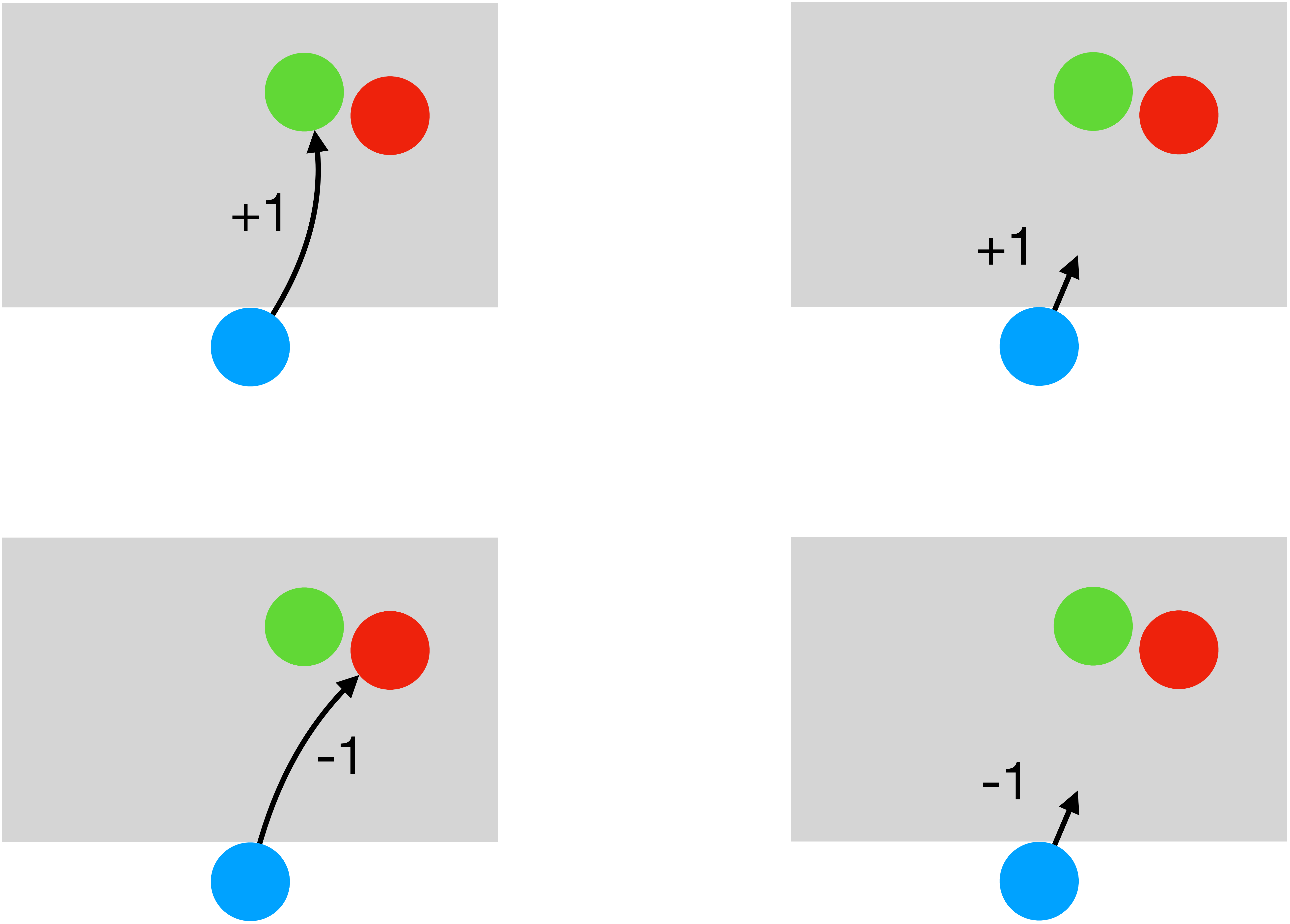
Training Objective

Classification:

Starting position

Correct object

Incorrect object

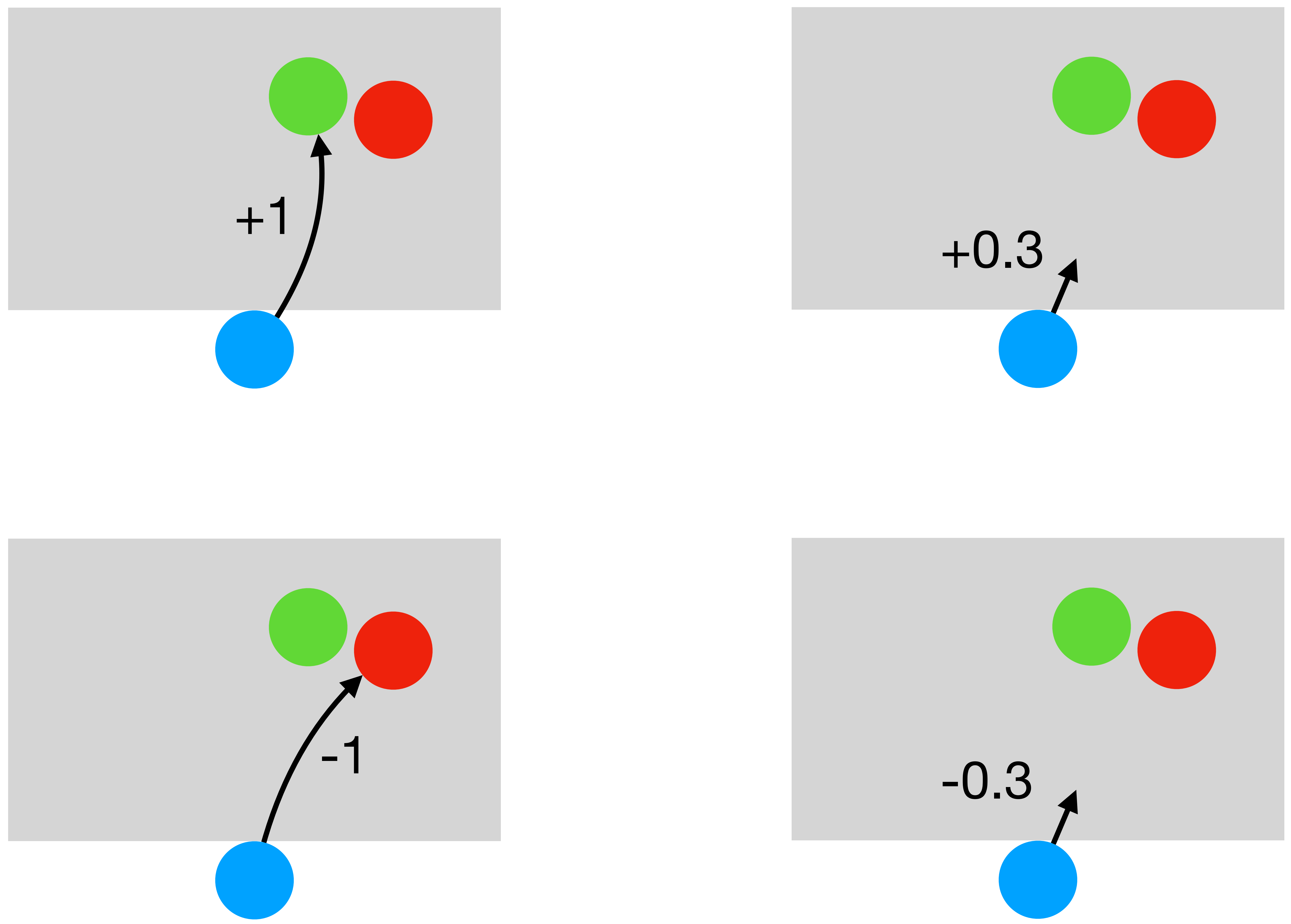


Pixels and Language to Rewards (PixL2R)

Training Objective

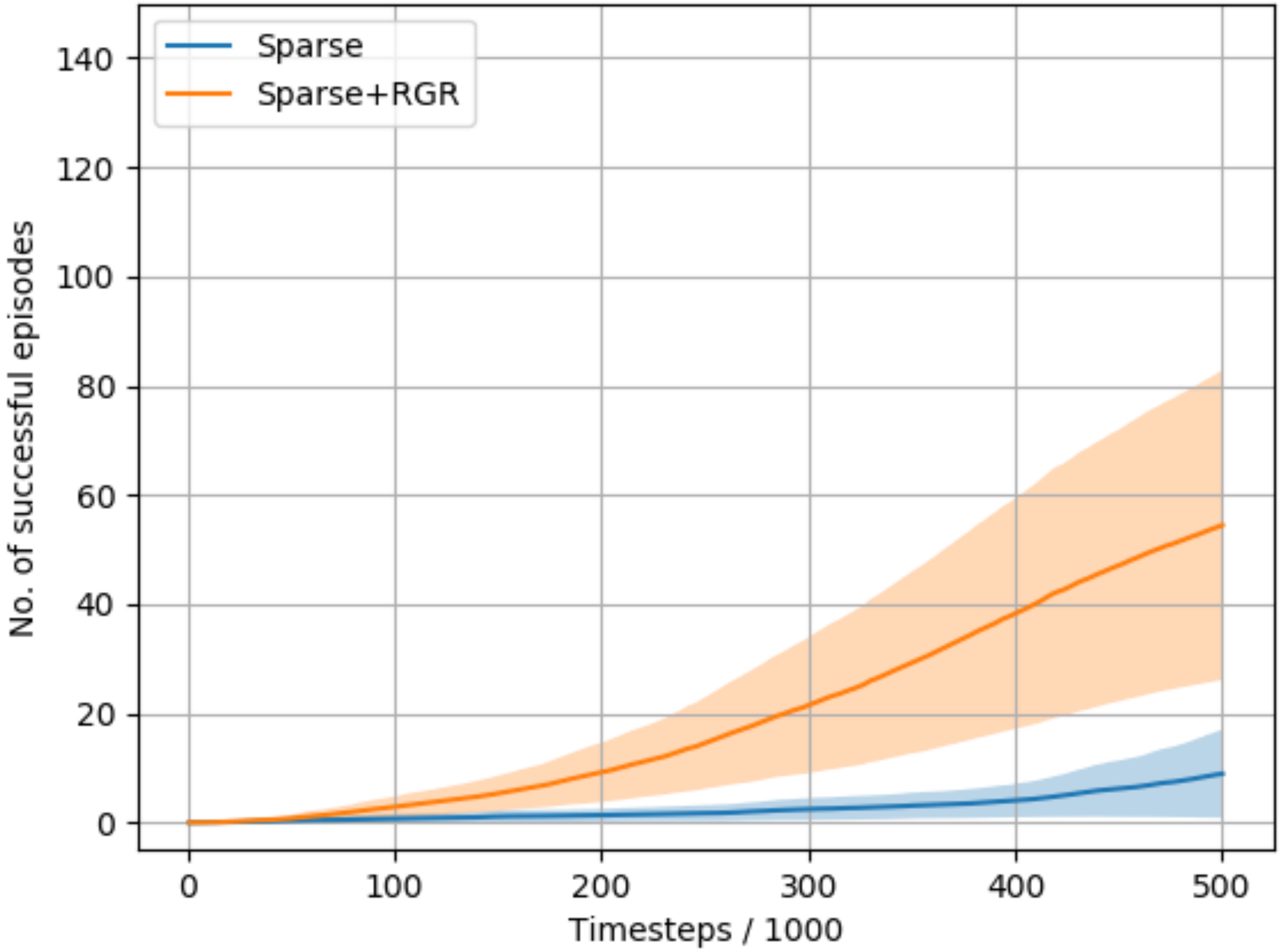
Regression:

- Starting position
- Correct object
- Incorrect object



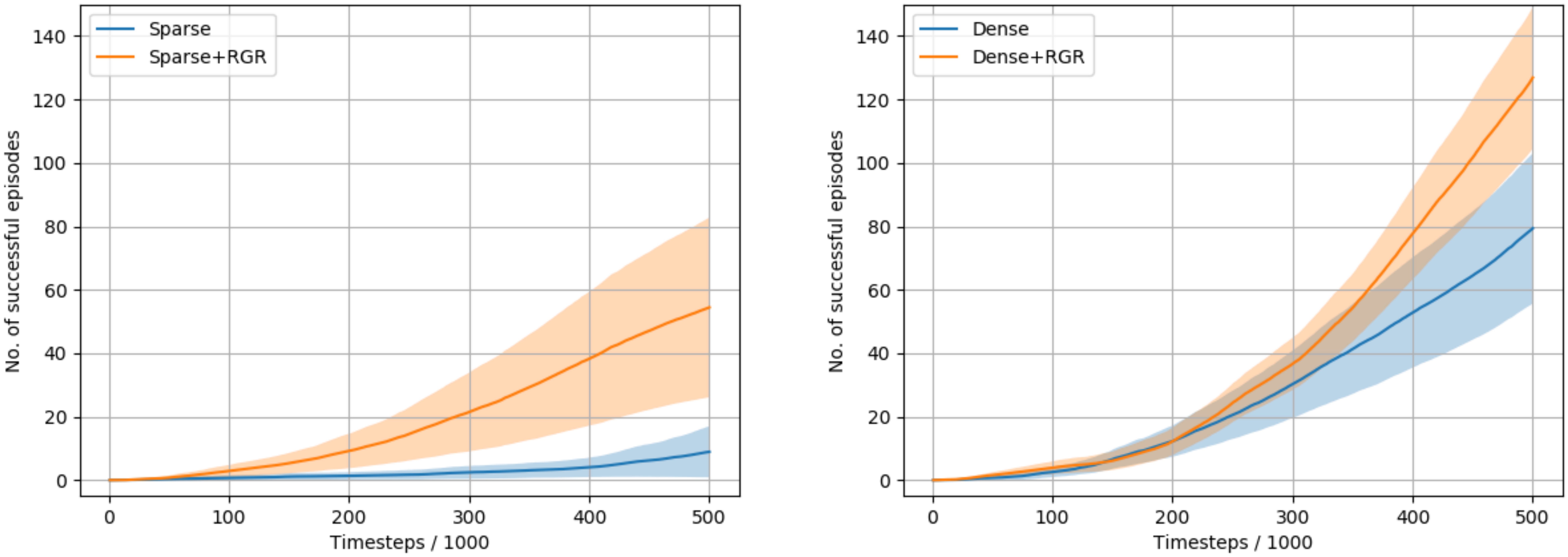
Pixels and Language to Rewards (PixL2R)

Results



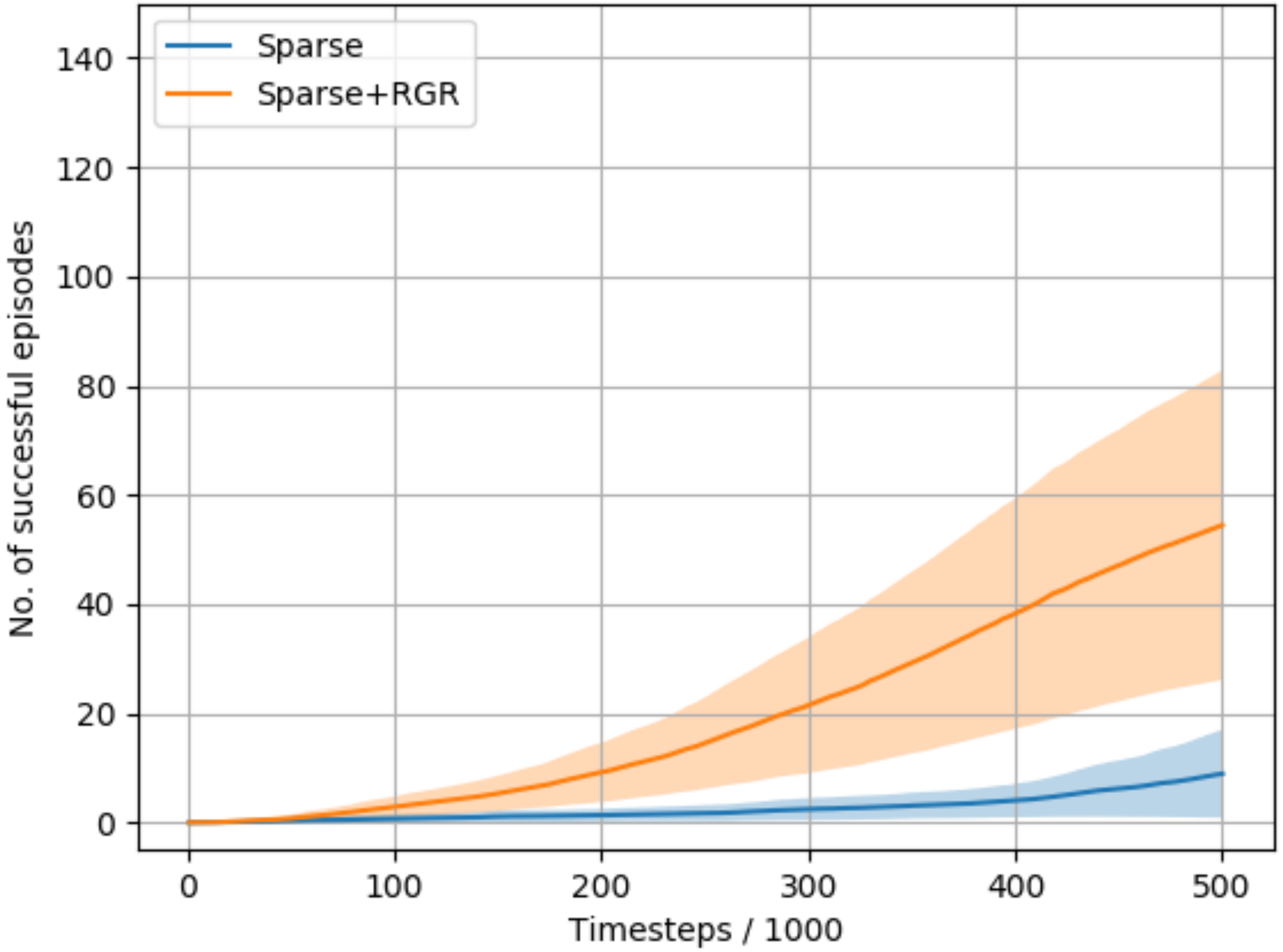
Pixels and Language to Rewards (PixL2R)

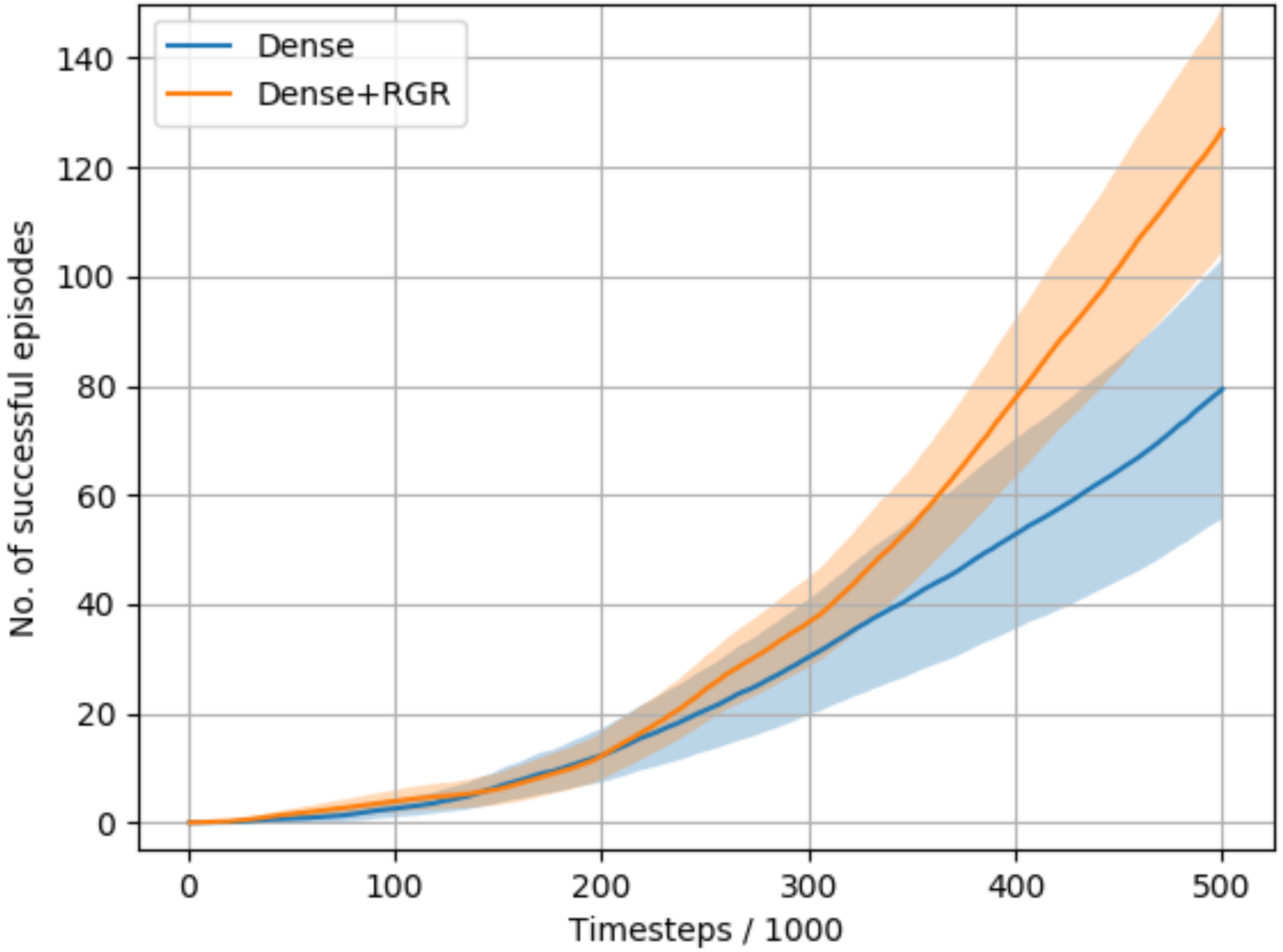
Results



Pixels and Language to Rewards (PixL2R)

Results

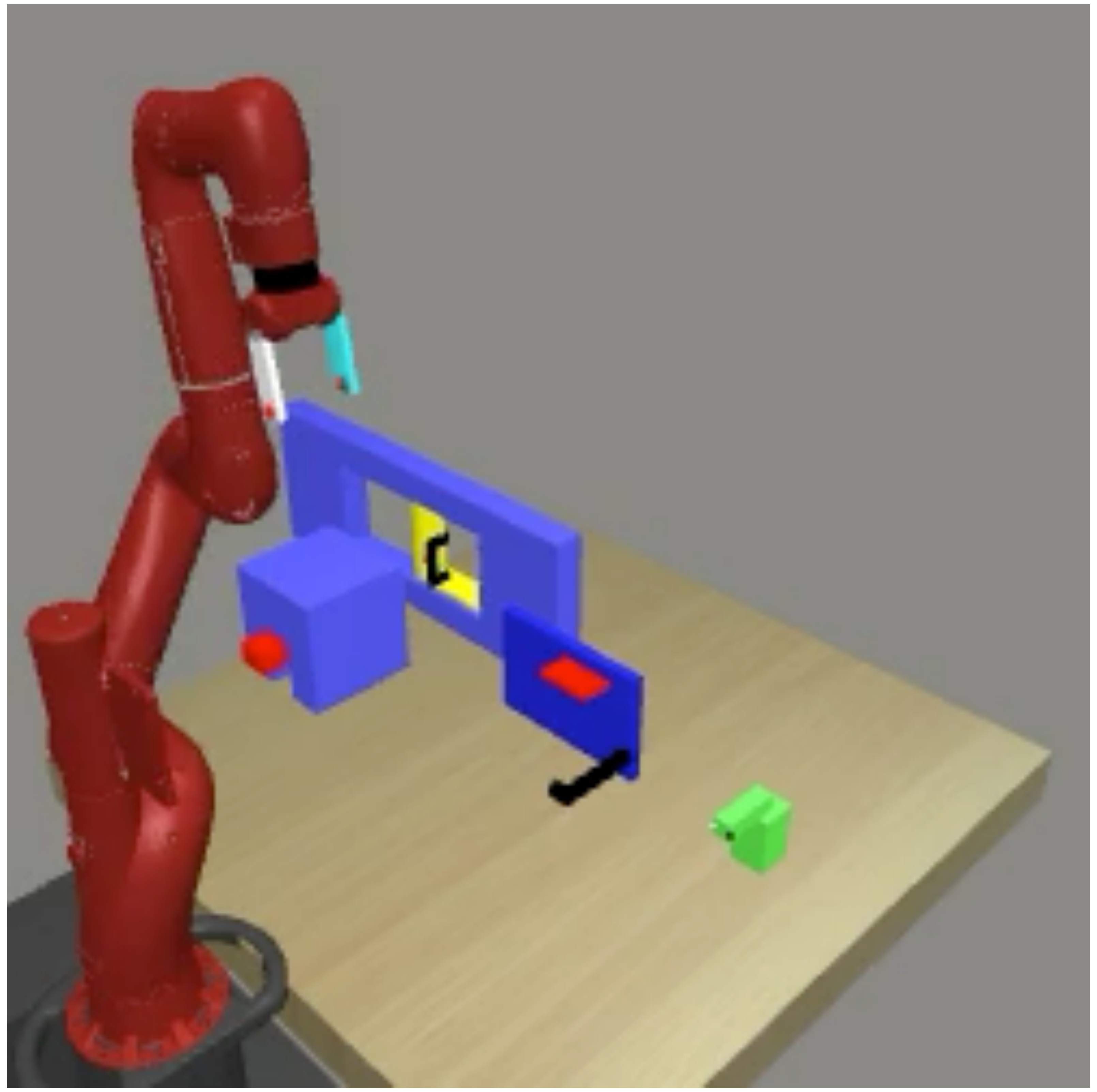




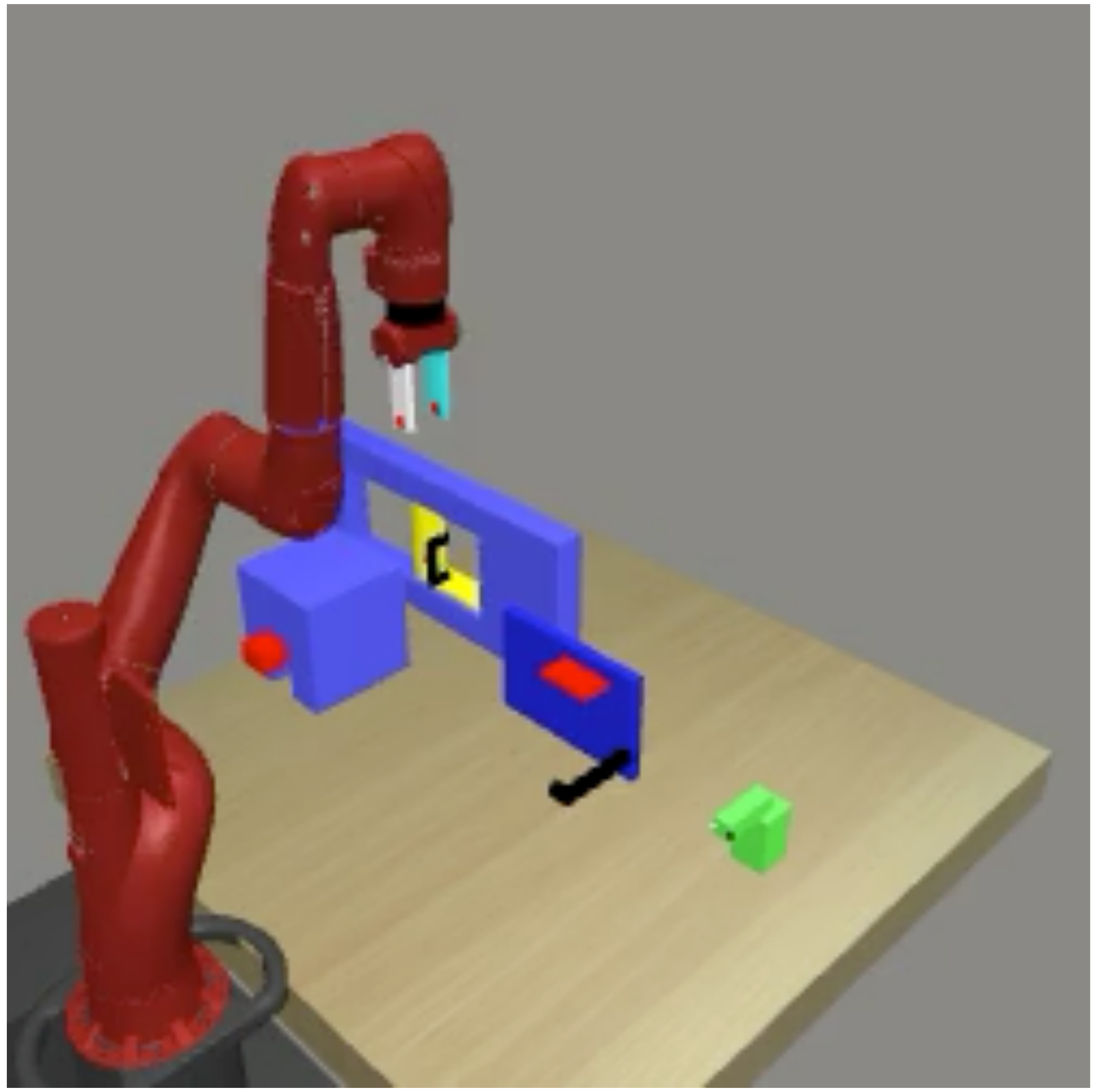
New RL training paradigm: Coarse dense rewards designed by hand + Language-based rewards

Pixels and Language to Rewards (PixL2R)

Results



Final policy **without** language-based rewards



Final policy **with** language-based rewards

Outline

- Introduction
- Completed Work:
  - Using Natural Language for Reward Shaping in Reinforcement Learning (IJCAI 2019)
  - Guiding Reinforcement Learning by Mapping Pixels to Rewards (CoRL 2020)
  - Zero-shot Task Adaptation using Natural Language (arXiv, 2021)
- Proposed Work: Short-term
  - Neurosymbolic Model
  - Policy Adaptation
- Proposed Work: Long-term
  - Policy Regularization
  - Bayesian Inference
  - Supervised Attention
- Related Work

Sequential Decision Making

Reinforcement Learning

Task specification:
Designing reward functions

Imitation Learning

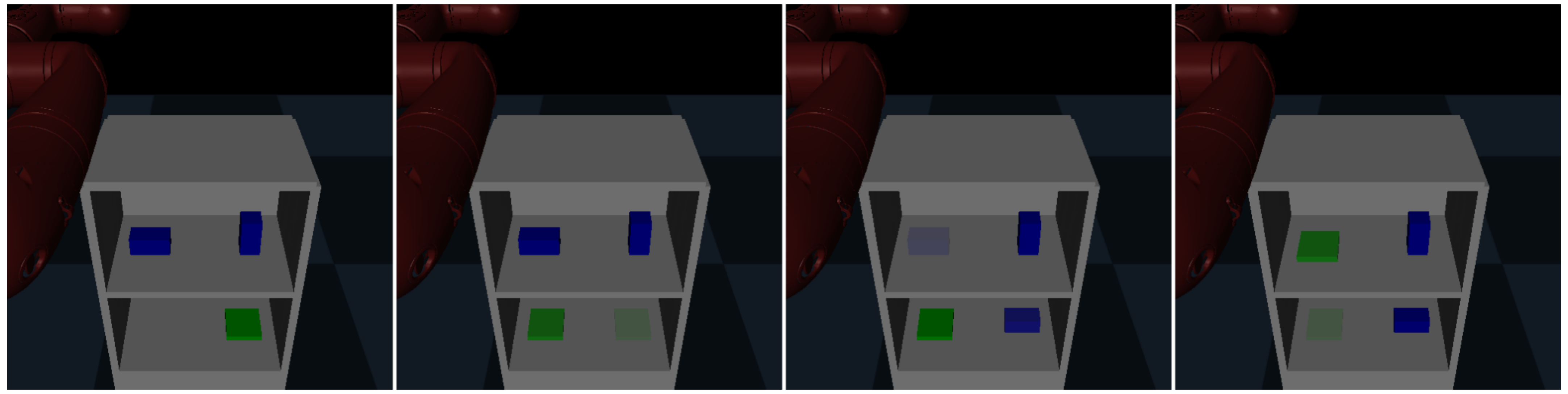
Task specification:
Providing demonstrations

Use natural language to aid these!

Language-Aided Reward and Value Adaptation (LARVA)

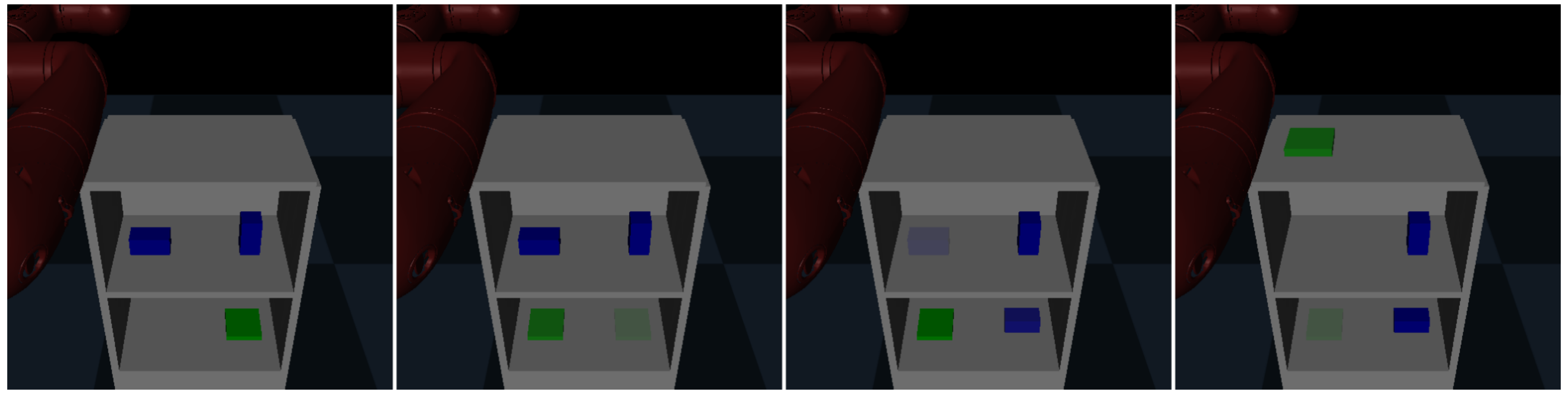
Problem Setting

Source task



Language describing the difference between source and target tasks

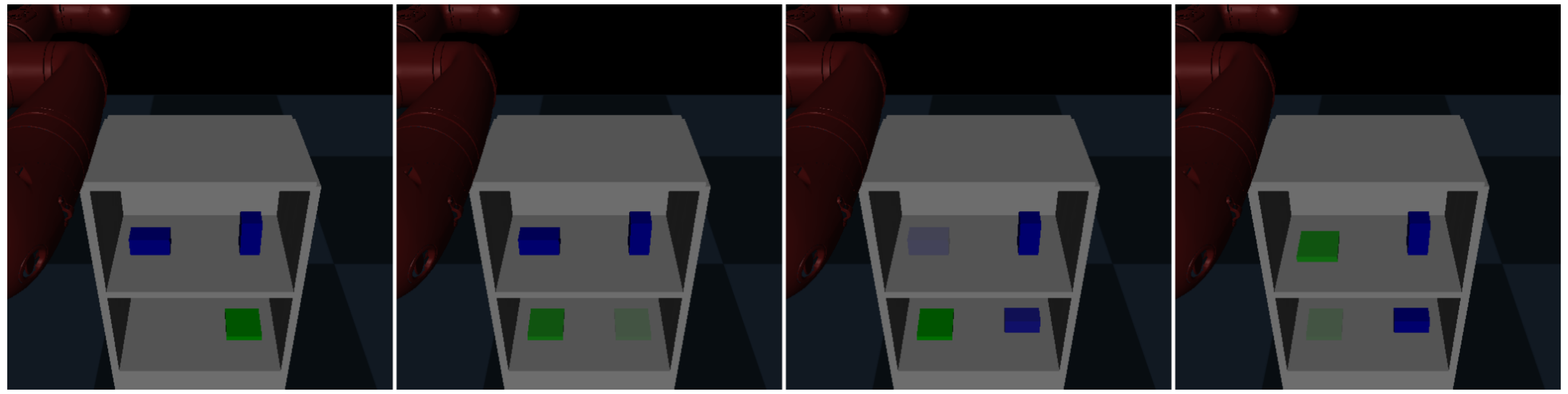
Target task



Language-Aided Reward and Value Adaptation (LARVA)

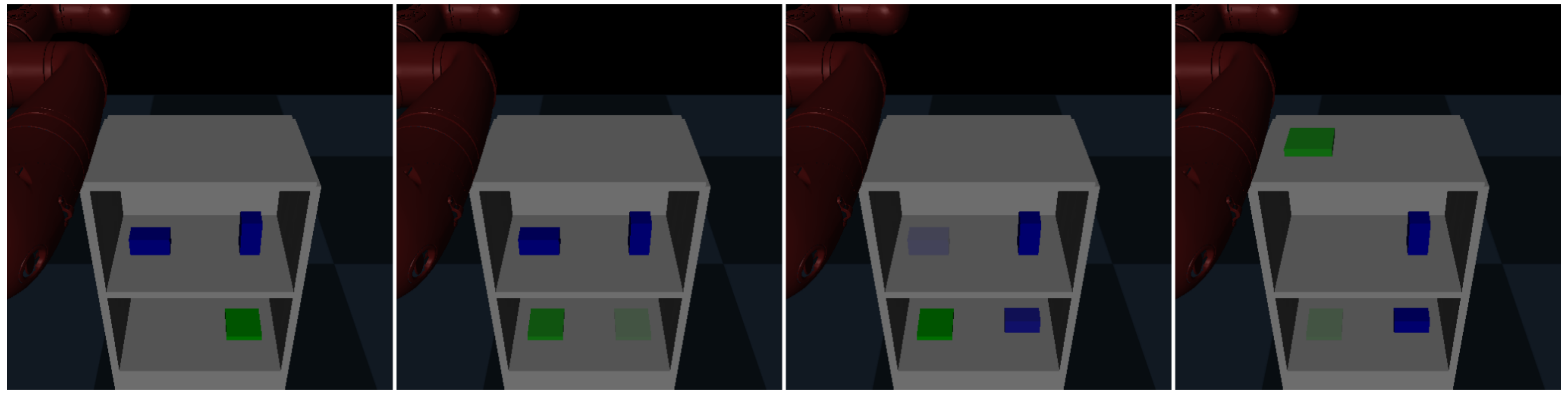
Problem Setting

Source task



Language describing the difference between source and target tasks

Target task

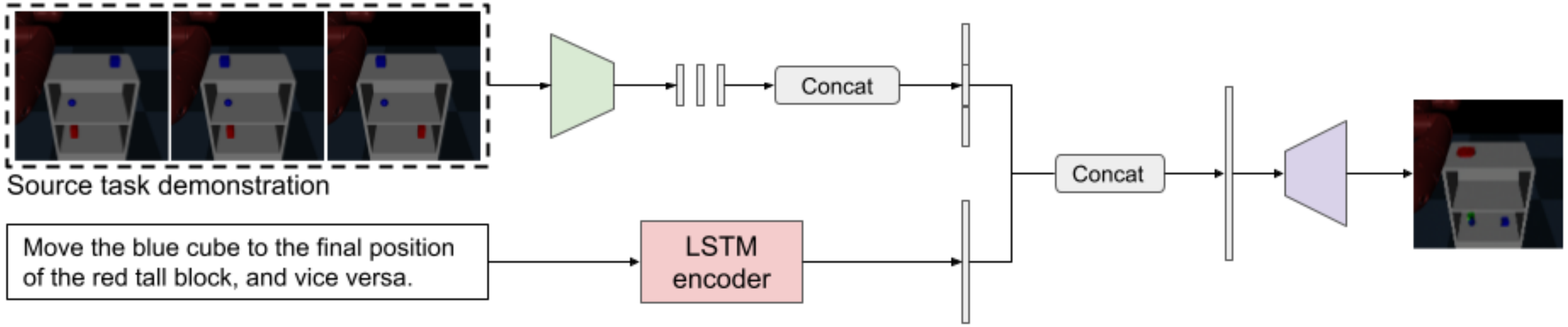


GOAL:
Learn the target task *without any demonstrations*.

Language-Aided Reward and Value Adaptation (LARVA)

Approach

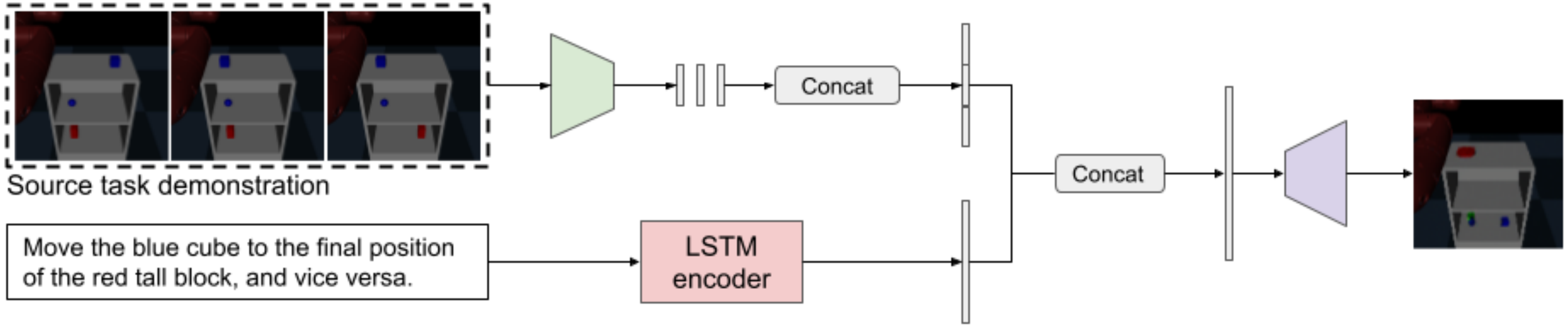
Target Goal Prediction:



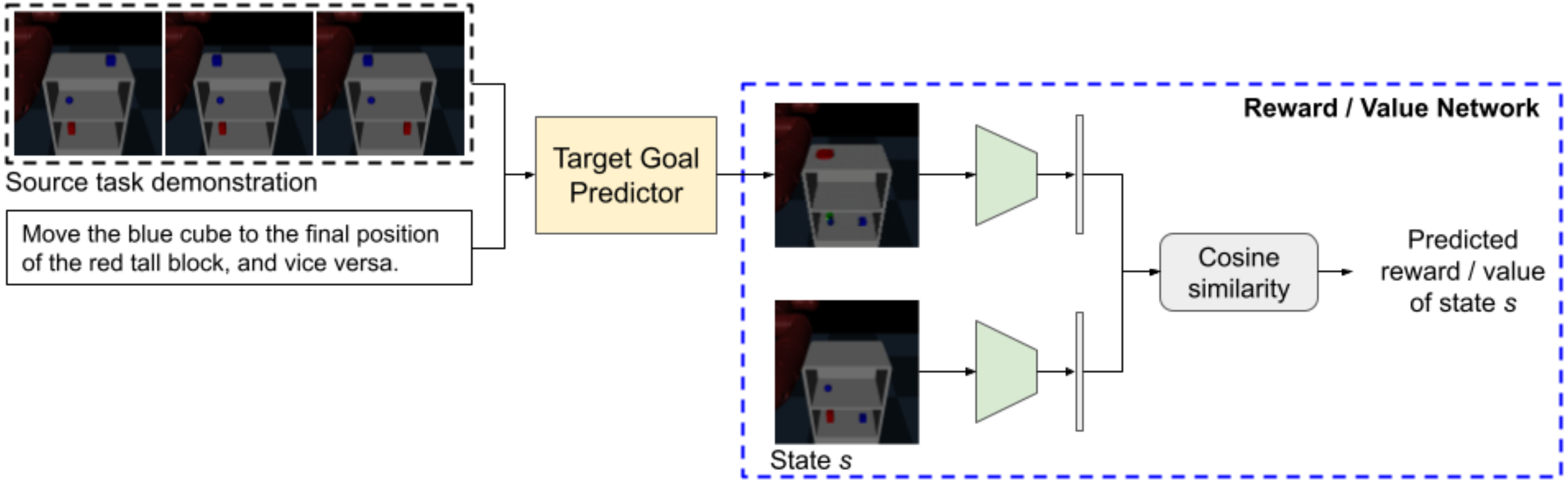
Language-Aided Reward and Value Adaptation (LARVA)

Approach

Target Goal Prediction:



Reward/Value Prediction:



Language-Aided Reward and Value Adaptation (LARVA)

Approach

Training data: (source demo, language, target goal, target reward / value function)

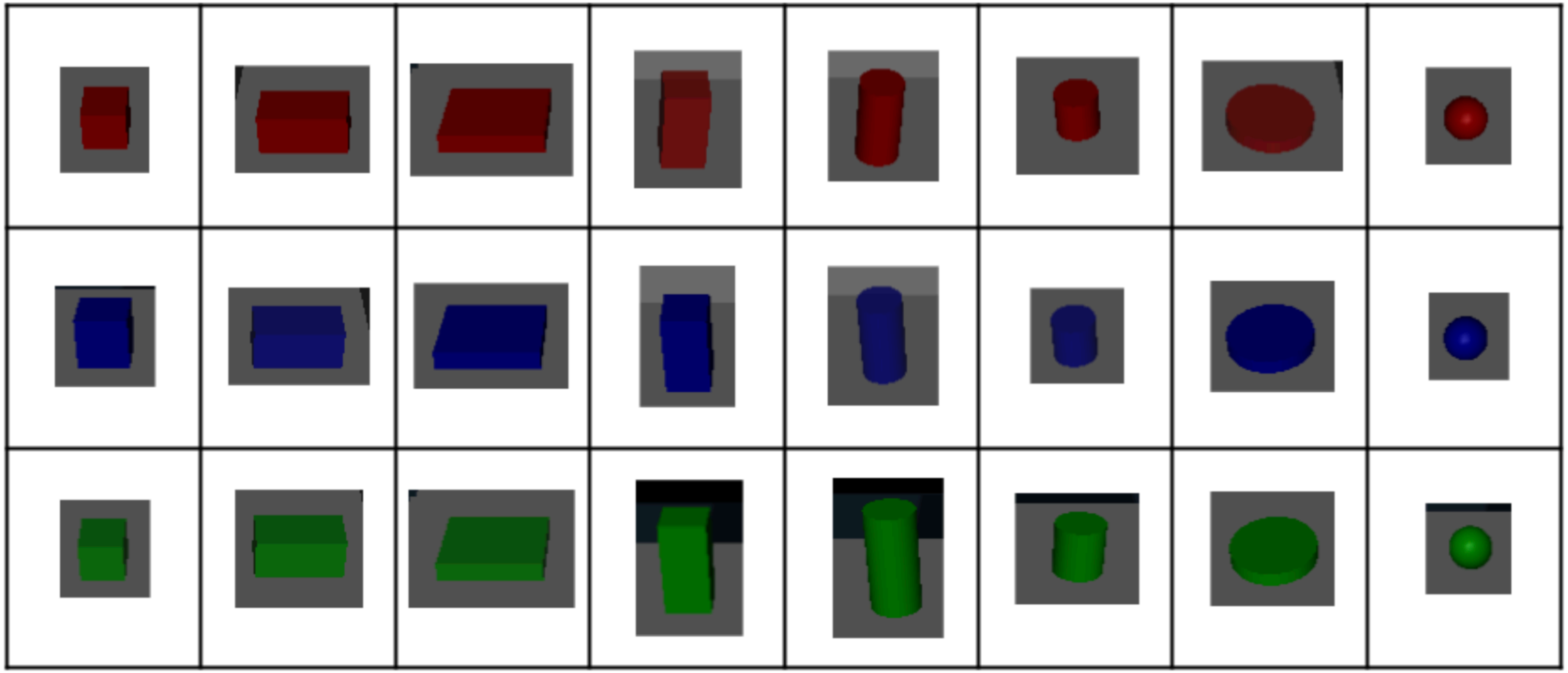
Loss functions:

- Reward / Value prediction: Mean-squared error
- Target goal prediction: Mean-squared error

Language-Aided Reward and Value Adaptation (LARVA)

Experiments

Objects:



Language-Aided Reward and Value Adaptation (LARVA)

Experiments

6 types of adaptations:

Same object, different place location	A ➝ P	A ➝ Q
Different object, same place location	A ➝ P	B ➝ P
Move two objects, with swapped final locations	A ➝ P B ➝ Q	A ➝ Q B ➝ P
Delete a step	A ➝ P B ➝ Q C ➝ R	B ➝ Q C ➝ R
Insert a step	B ➝ Q C ➝ R	A ➝ P B ➝ Q C ➝ R
Modify a step	A ➝ P B ➝ Q C ➝ R	A ➝ P D ➝ S C ➝ R

Language-Aided Reward and Value Adaptation (LARVA)

Experiments

Language Data:

- Template-based
- Paraphrases from Amazon Mechanical Turk

Evaluation Metric:

- Success rate: percentage of datapoints where the true goal state for the target task gets the highest reward / value by the model.

Language-Aided Reward and Value Adaptation (LARVA)

Experiments

Experiment	Success rate (%)	
	Synthetic	Natural

Language-Aided Reward and Value Adaptation (LARVA)

Experiments

Experiment		Success rate (%)	
		Synthetic	Natural
1.	LARVA; reward prediction	97.8	75.7
2.	LARVA; value prediction	97.7	73.3

Language-Aided Reward and Value Adaptation (LARVA)

Experiments

	Experiment	Success rate (%)	
		Synthetic	Natural
1.	LARVA; reward prediction	97.8	75.7
2.	LARVA; value prediction	97.7	73.3
3.	LARVA; no target goal supervision	20.0	2.7

Language-Aided Reward and Value Adaptation (LARVA)

Experiments

	Experiment	Success rate (%)	
		Synthetic	Natural
1.	LARVA; reward prediction	97.8	75.7
2.	LARVA; value prediction	97.7	73.3
3.	LARVA; no target goal supervision	20.0	2.7
4.	LARVA; no language	20.7	22.3
5.	LARVA; no source demonstration	4.2	3.3

Language-Aided Reward and Value Adaptation (LARVA)

Experiments

	Experiment	Success rate (%)	
		Synthetic	Natural
1.	LARVA; reward prediction	97.8	75.7
2.	LARVA; value prediction	97.7	73.3
3.	LARVA; no target goal supervision	20.0	2.7
4.	LARVA; no language	20.7	22.3
5.	LARVA; no source demonstration	4.2	3.3
6.	NN without decomposition	1.8	1.0

Language-Aided Reward and Value Adaptation (LARVA)

Experiments

	Experiment	Success rate (%)	
		Synthetic	Natural
1.	LARVA; reward prediction	97.8	75.7
2.	LARVA; value prediction	97.7	73.3
3.	LARVA; no target goal supervision	20.0	2.7
4.	LARVA; no language	20.7	22.3
5.	LARVA; no source demonstration	4.2	3.3
6.	NN without decomposition	1.8	1.0
7.	LARVA: Compostionality – red box	87.6	62.4
8.	LARVA: Compostionality – blue cylinder	89.4	65.9

Outline

- Introduction
- Related Work
- Completed Work:
  - Using Natural Language for Reward Shaping in Reinforcement Learning (IJCAI 2019)
  - Guiding Reinforcement Learning by Mapping Pixels to Rewards (CoRL 2020)
  - Zero-shot Task Adaptation using Natural Language (arXiv, 2021)
- Proposed Work: Short-term
  - Neurosymbolic Model
  - Policy Adaptation
- Proposed Work: Long-term
  - Policy Regularization
  - Bayesian Inference
  - Supervised Attention

Task Adaptation: Neurosymbolic Model

Motivation

Most natural tasks can be described using a sequence of actions applied on some objects…

such that there is modularity, i.e., the same action can be applied to different objects.

=> Motivates a neurosymbolic approach

Task Adaptation: Neurosymbolic Model

Motivation

Neural Production Systems
[Goyal et al., 2021]

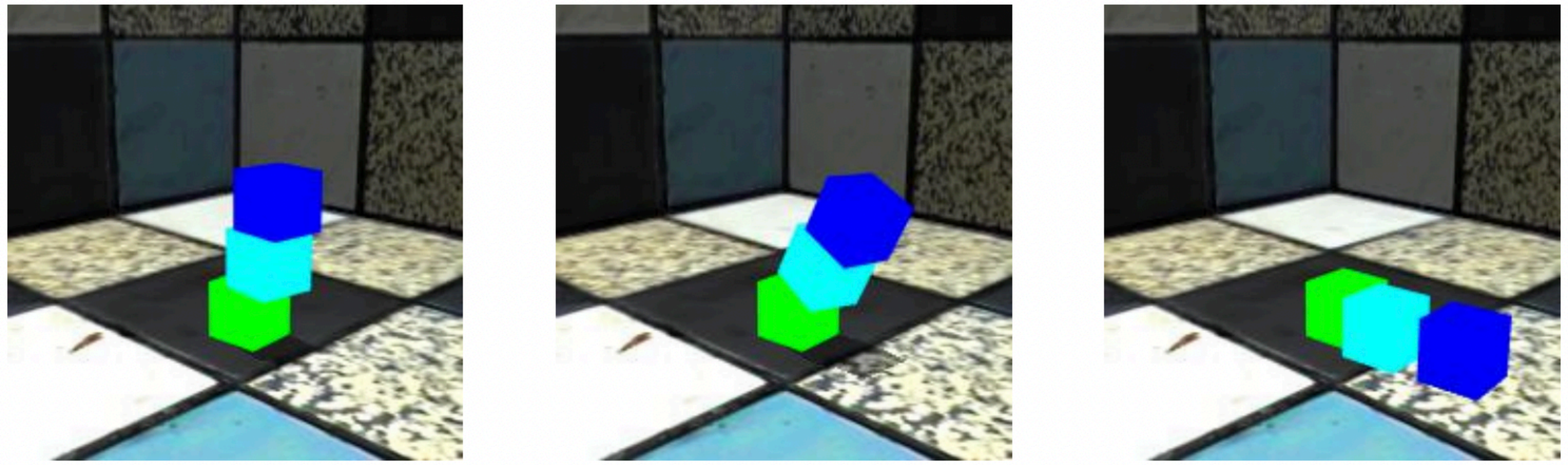
Not me!

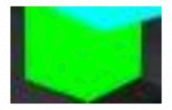
Task Adaptation: Neurosymbolic Model

Motivation

Neural Production Systems
[Goyal et al., 2021]

Not me!



	Rule 1	Rule 2
	0.84	0.16
	0.62	0.38
	0.29	0.71

Task Adaptation: Neurosymbolic Model

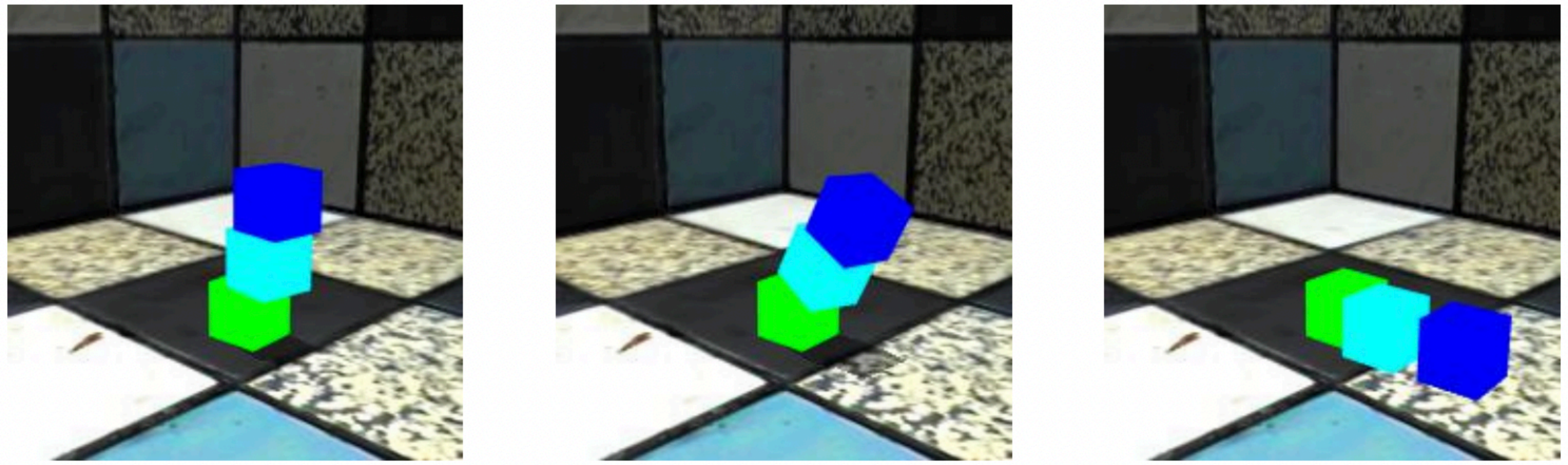
Motivation

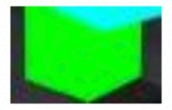
Neural Production Systems
[Goyal et al., 2021]

Not me!

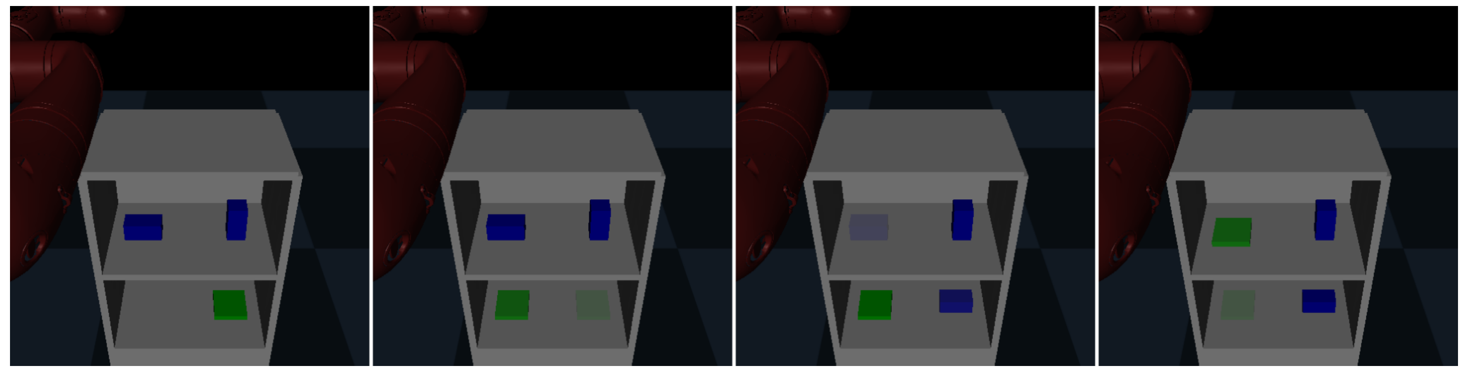
Entities: Vectors
Rules: Neural Networks

Learned from data



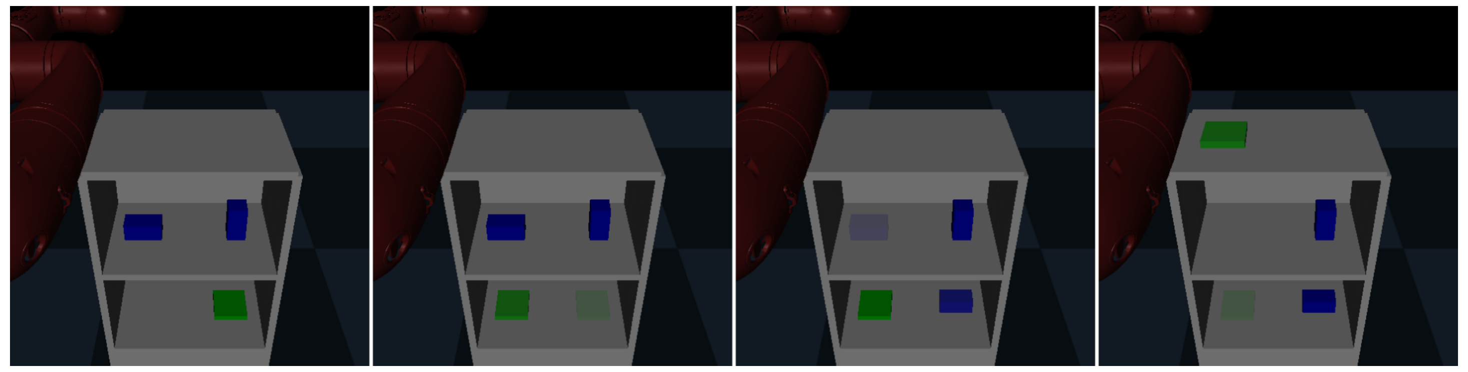
	Rule 1	Rule 2
	0.84	0.16
	0.62	0.38
	0.29	0.71

Task Adaptation: Neurosymbolic Model — Training



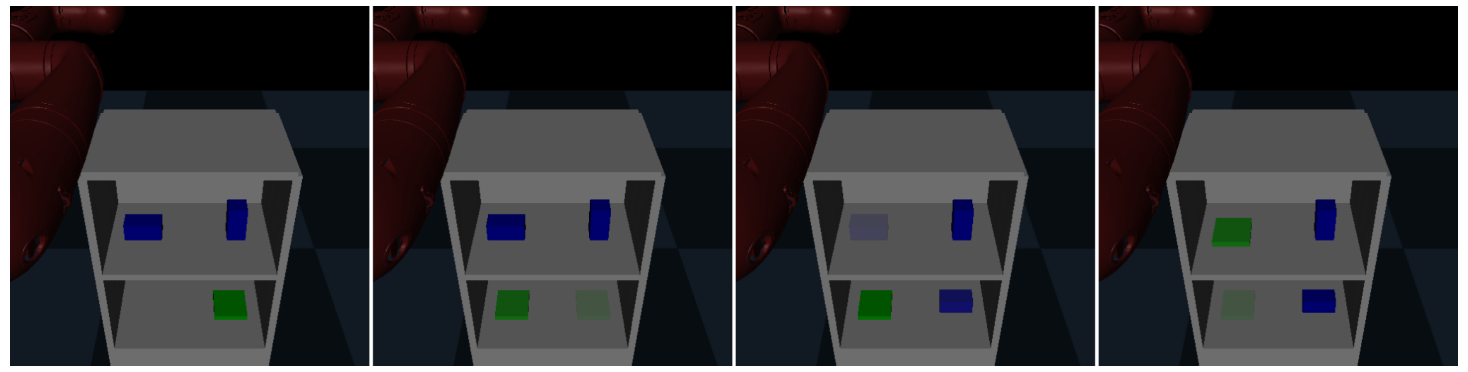
Source demo

In the third step, move the green flat block from bottom left to top left.



Target demo

Task Adaptation: Neurosymbolic Model — Training



Source demo

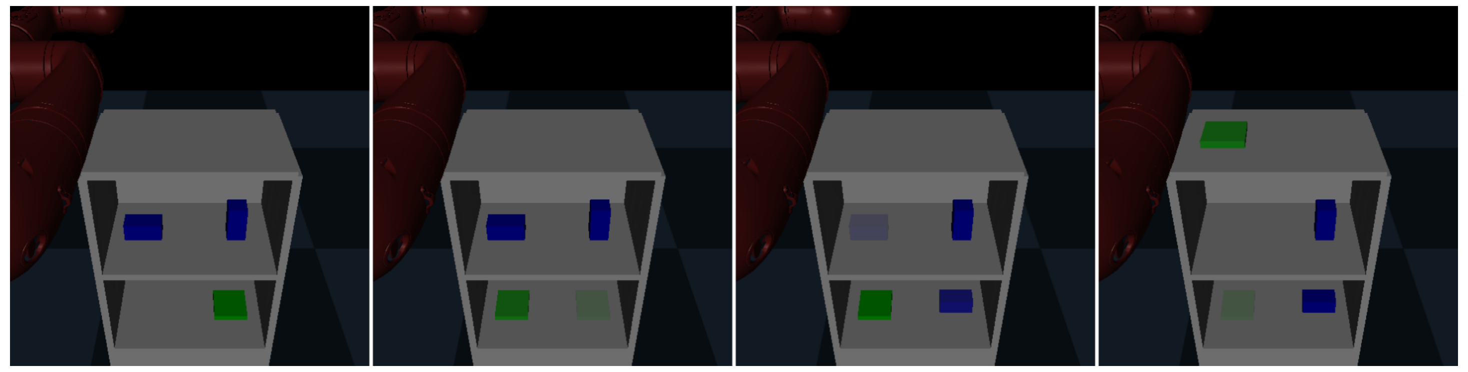
E1: Blue, Long, Mid-left
E2: Blue, Tall, Mid-right
E3: Green, Flat, Bottom-right

E1: Blue, Long, Mid-left
E2: Blue, Tall, Mid-right
E3: Green, Flat, Bottom-left

E1: Blue, Long, Mid-left
E2: Blue, Tall, Bottom-right
E3: Green, Flat, Bottom-left

E1: Blue, Long, Mid-left
E2: Blue, Tall, Bottom-right
E3: Green, Flat, Mid-left

In the third step, move the green flat block from bottom left to top left.



Target demo

E1: Blue, Long, Mid-left
E2: Blue, Tall, Mid-right
E3: Green, Flat, Bottom-right

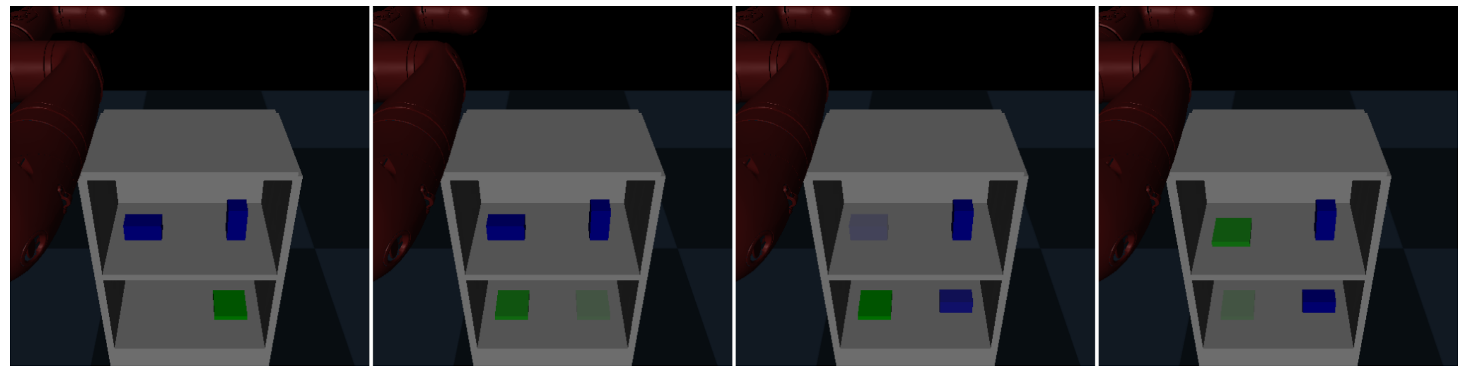
E1: Blue, Long, Mid-left
E2: Blue, Tall, Mid-right
E3: Green, Flat, Bottom-left

E1: Blue, Long, Mid-left
E2: Blue, Tall, Bottom-right
E3: Green, Flat, Bottom-left

E1: Blue, Long, Mid-left
E2: Blue, Tall, Bottom-right
E3: Green, Flat, Top-left

Task Adaptation: Neurosymbolic Model — Training

In the third step, move the green flat block from bottom left to top left.



Source demo

E1: Blue, Long, Mid-left
E2: Blue, Tall, Mid-right
E3: Green, Flat, Bottom-right

↓ To_BottomLeft(E3)

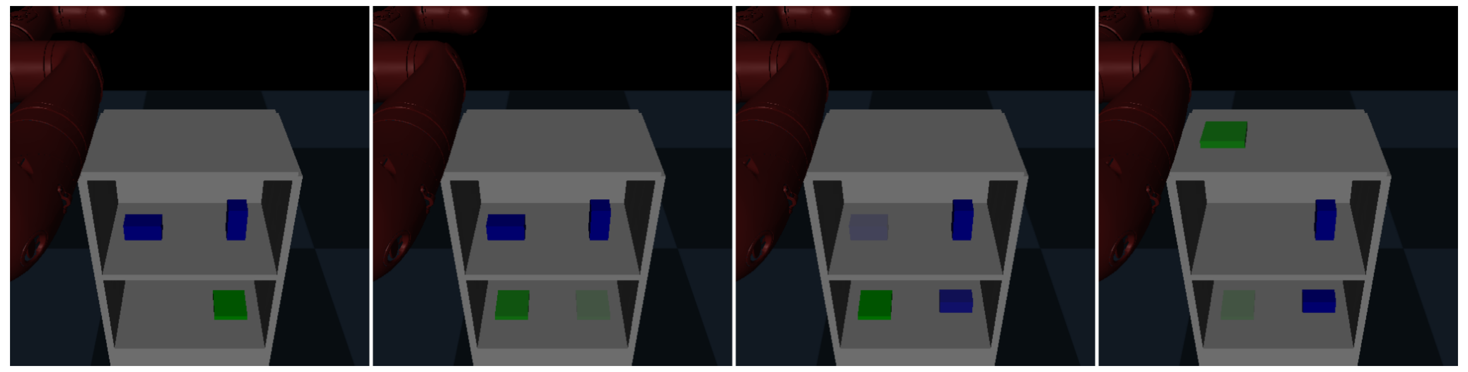
E1: Blue, Long, Mid-left
E2: Blue, Tall, Mid-right
E3: Green, Flat, Bottom-left

↓ To_BottomRight(E2)

E1: Blue, Long, Mid-left
E2: Blue, Tall, Bottom-right
E3: Green, Flat, Bottom-left

↓ To_MidLeft(E3)

E1: Blue, Long, Mid-left
E2: Blue, Tall, Bottom-right
E3: Green, Flat, Mid-left



Target demo

E1: Blue, Long, Mid-left
E2: Blue, Tall, Mid-right
E3: Green, Flat, Bottom-right

↓ To_BottomLeft(E3)

E1: Blue, Long, Mid-left
E2: Blue, Tall, Mid-right
E3: Green, Flat, Bottom-left

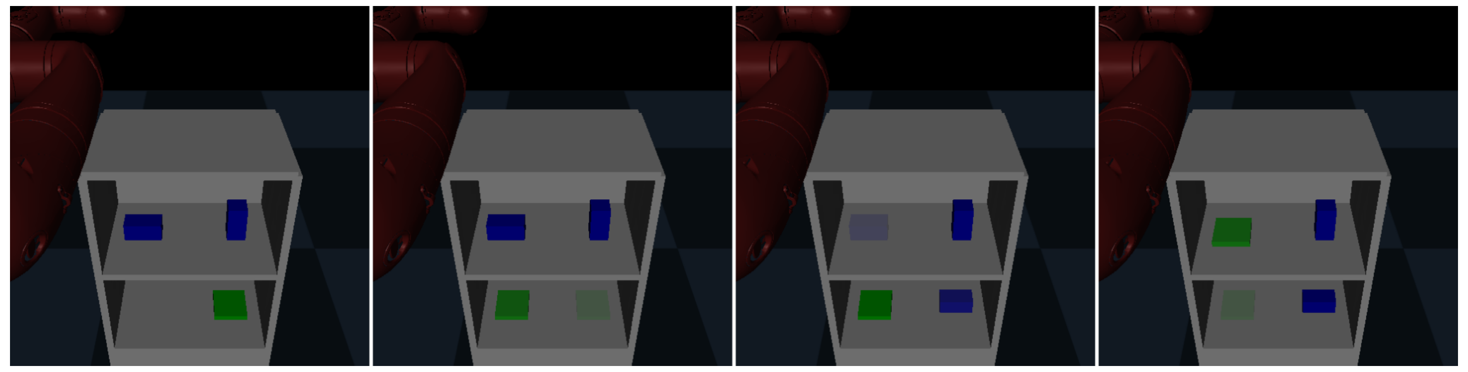
↓ To_BottomRight(E2)

E1: Blue, Long, Mid-left
E2: Blue, Tall, Bottom-right
E3: Green, Flat, Bottom-left

↓ To_TopLeft(E3)

E1: Blue, Long, Mid-left
E2: Blue, Tall, Bottom-right
E3: Green, Flat, Top-left

Task Adaptation: Neurosymbolic Model — Training



Source demo

E1: Blue, Long, Mid-left
E2: Blue, Tall, Mid-right
E3: Green, Flat, Bottom-right

↓ To_BottomLeft(E3)

E1: Blue, Long, Mid-left
E2: Blue, Tall, Mid-right
E3: Green, Flat, Bottom-left

↓ To_BottomRight(E2)

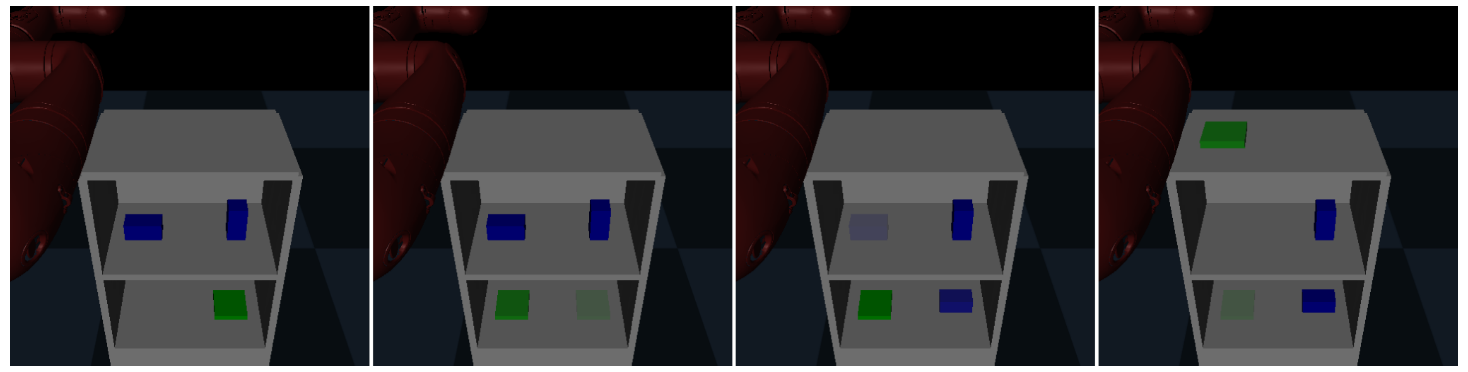
E1: Blue, Long, Mid-left
E2: Blue, Tall, Bottom-right
E3: Green, Flat, Bottom-left

↓ To_MidLeft(E3)

E1: Blue, Long, Mid-left
E2: Blue, Tall, Bottom-right
E3: Green, Flat, Mid-left

In the third step, move the green flat block from bottom left to top left.

Adaptation Model



Target demo

E1: Blue, Long, Mid-left
E2: Blue, Tall, Mid-right
E3: Green, Flat, Bottom-right

↓ To_BottomLeft(E3)

E1: Blue, Long, Mid-left
E2: Blue, Tall, Mid-right
E3: Green, Flat, Bottom-left

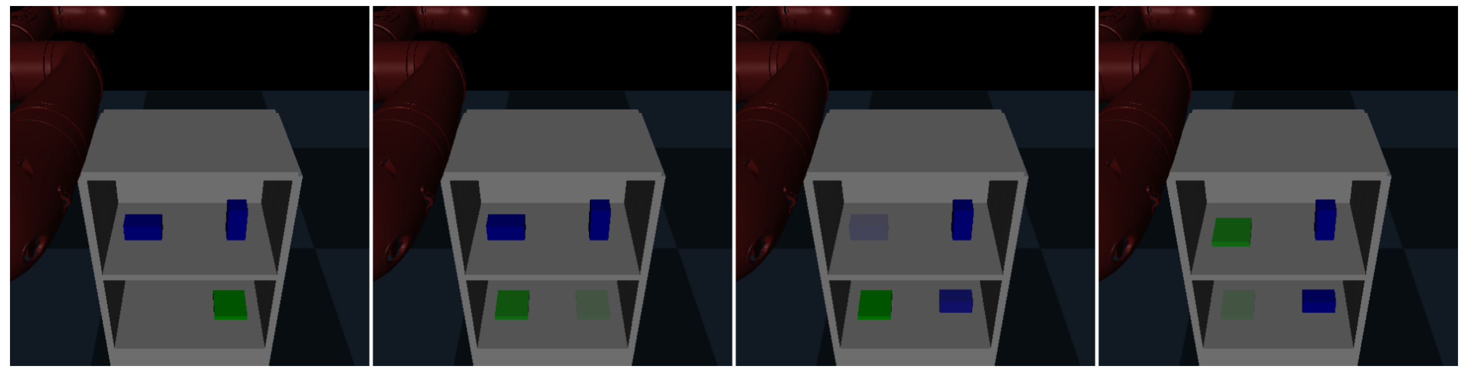
↓ To_BottomRight(E2)

E1: Blue, Long, Mid-left
E2: Blue, Tall, Bottom-right
E3: Green, Flat, Bottom-left

↓ To_TopLeft(E3)

E1: Blue, Long, Mid-left
E2: Blue, Tall, Bottom-right
E3: Green, Flat, Top-left

Task Adaptation: Neurosymbolic Model — Training



Source demo

E1: Blue, Long, Mid-left
E2: Blue, Tall, Mid-right
E3: Green, Flat, Bottom-right

To_BottomLeft(E3)

E1: Blue, Long, Mid-left
E2: Blue, Tall, Mid-right
E3: Green, Flat, Bottom-left

To_BottomRight(E2)

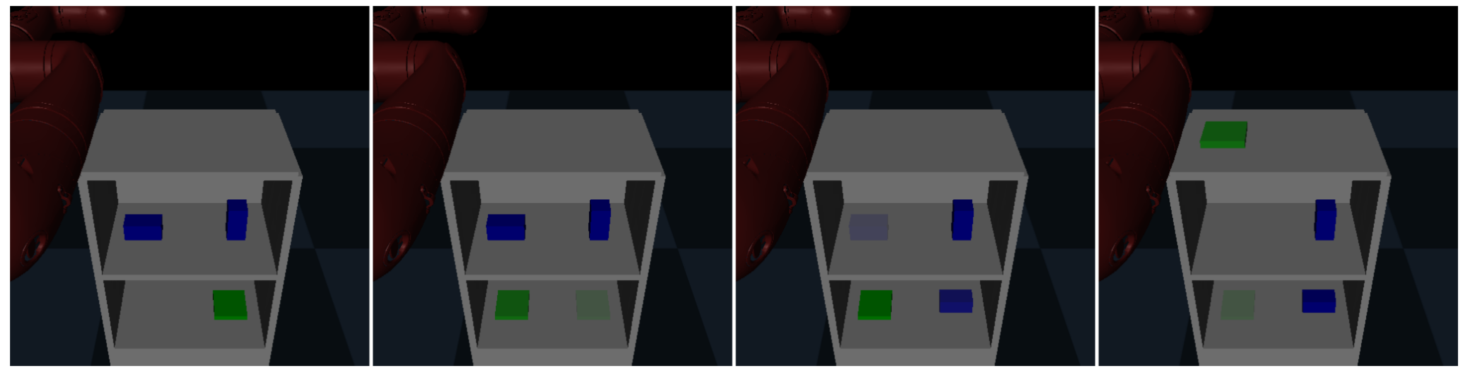
E1: Blue, Long, Mid-left
E2: Blue, Tall, Bottom-right
E3: Green, Flat, Bottom-left

To_MidLeft(E3)

E1: Blue, Long, Mid-left
E2: Blue, Tall, Bottom-right
E3: Green, Flat, Mid-left

In the third step, move the green flat block from bottom left to top left.

Adaptation Model



Target demo

E1: Blue, Long, Mid-left
E2: Blue, Tall, Mid-right
E3: Green, Flat, Bottom-right

To_BottomLeft(E3)

E1: Blue, Long, Mid-left
E2: Blue, Tall, Mid-right
E3: Green, Flat, Bottom-left

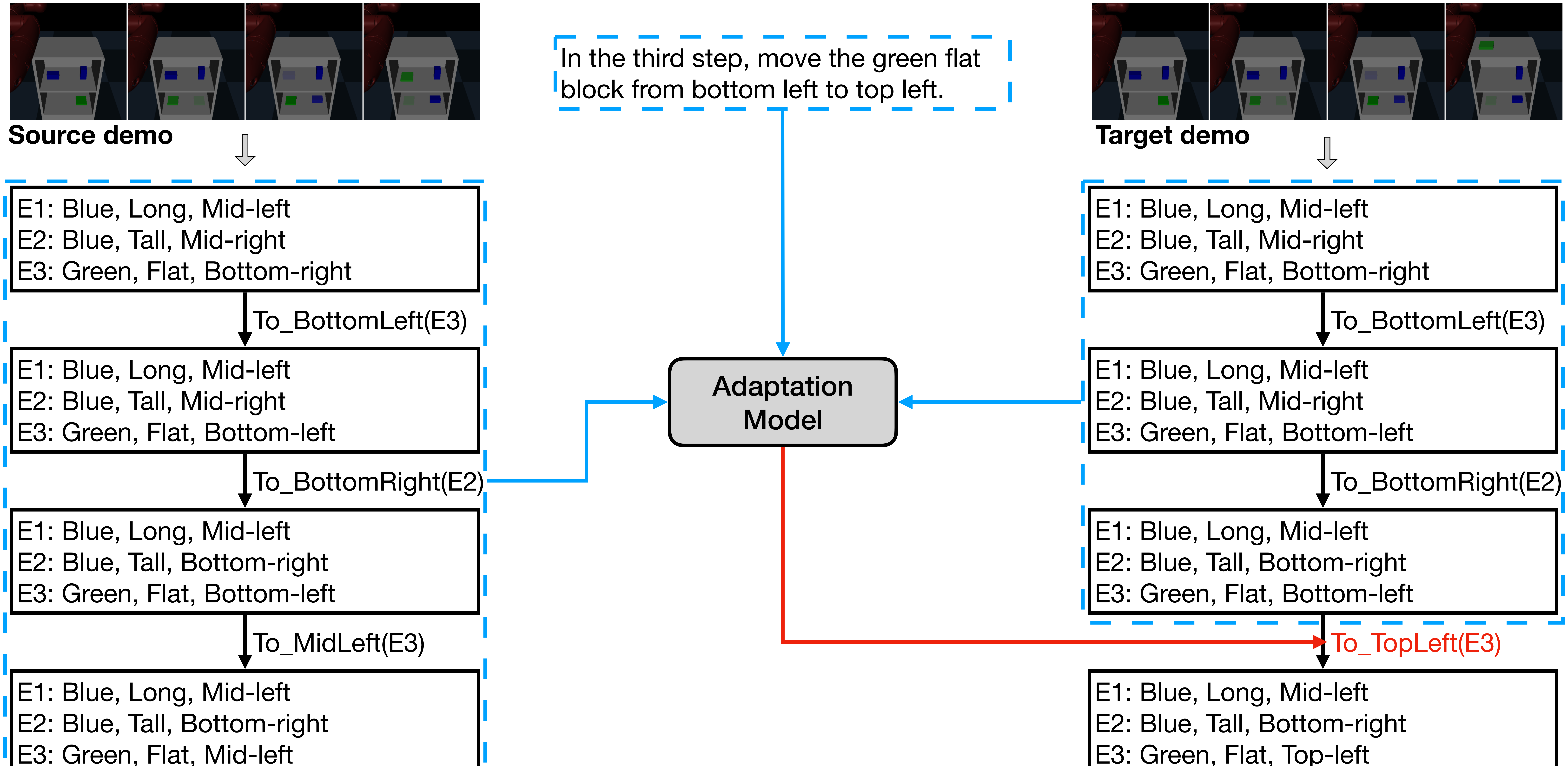
To_BottomRight(E2)

E1: Blue, Long, Mid-left
E2: Blue, Tall, Bottom-right
E3: Green, Flat, Bottom-left

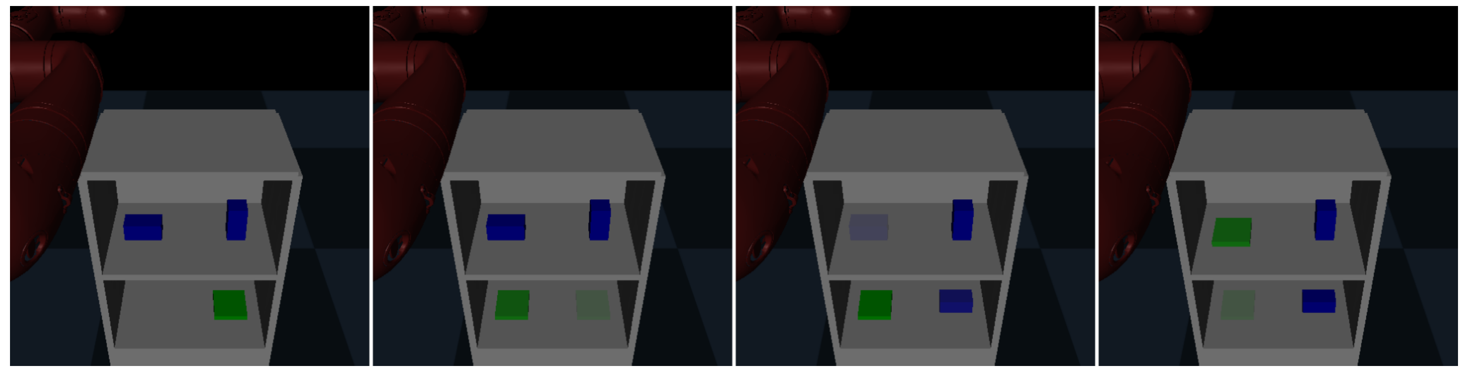
To_TopLeft(E3)

E1: Blue, Long, Mid-left
E2: Blue, Tall, Bottom-right
E3: Green, Flat, Top-left

Task Adaptation: Neurosymbolic Model — Training



Task Adaptation: Neurosymbolic Model — Inference



Source demo

E1: Blue, Long, Mid-left
E2: Blue, Tall, Mid-right
E3: Green, Flat, Bottom-right

To_BottomLeft(E3)

E1: Blue, Long, Mid-left
E2: Blue, Tall, Mid-right
E3: Green, Flat, Bottom-left

To_BottomRight(E2)

E1: Blue, Long, Mid-left
E2: Blue, Tall, Bottom-right
E3: Green, Flat, Bottom-left

To_MidLeft(E3)

E1: Blue, Long, Mid-left
E2: Blue, Tall, Bottom-right
E3: Green, Flat, Mid-left

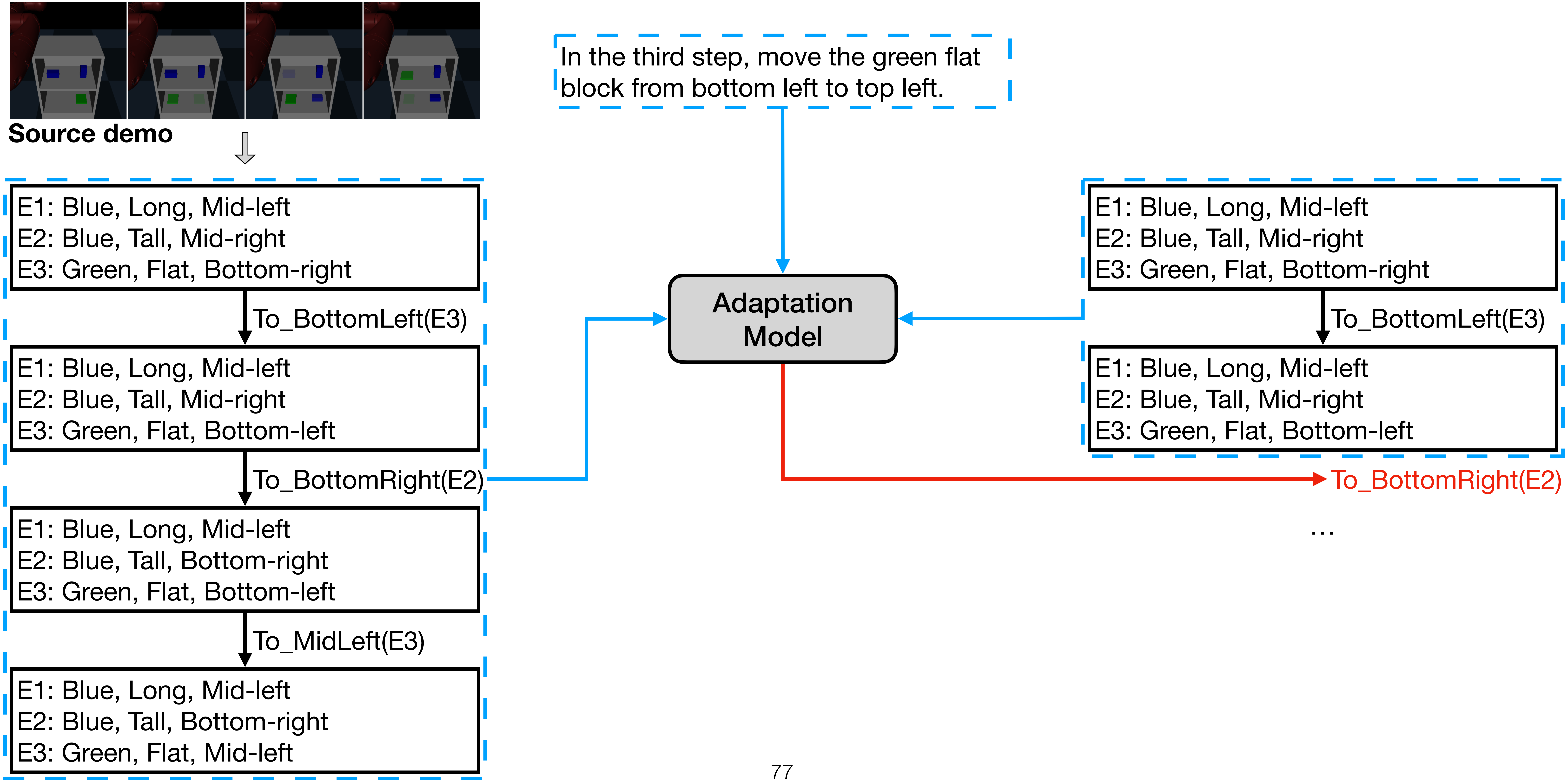
In the third step, move the green flat block from bottom left to top left.

Adaptation Model

E1: Blue, Long, Mid-left
E2: Blue, Tall, Mid-right
E3: Green, Flat, Bottom-right

To_BottomLeft(E3)

Task Adaptation: Neurosymbolic Model — Inference



Task Adaptation: Neurosymbolic Model

Approach

- The proposed model can be used to predict a state-only demonstration for the target task.
  => Use IRL, e.g., Generative Adversarial Imitation from Observation (GAIfO) to learn a policy.

[Torabi et al., 2018]

Task Adaptation: Neurosymbolic Model

Approach

- The proposed model can be used to predict a state-only demonstration for the target task.
  => Use IRL, e.g., Generative Adversarial Imitation from Observation (GAIfO) to learn a policy.

- For continuous control, segment the demonstrations.
  => Production rules may need to be augmented with a vector that controls the shape of the trajectory between keyframes.

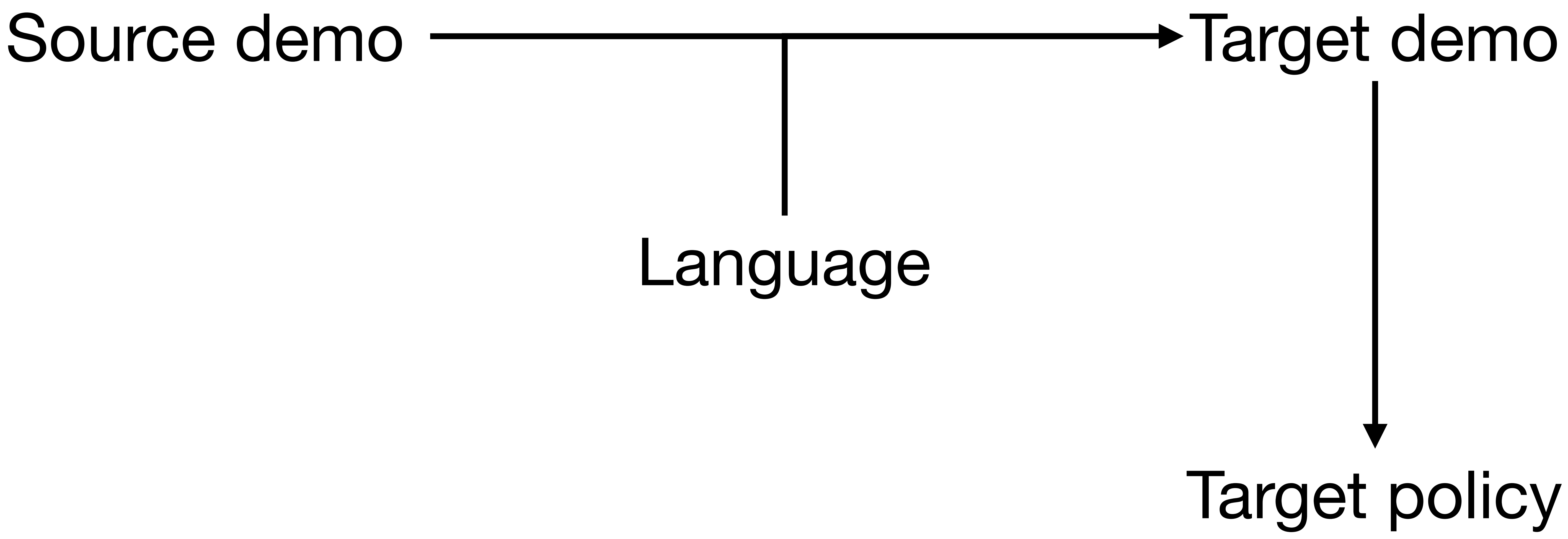
[Torabi et al., 2018]

Outline

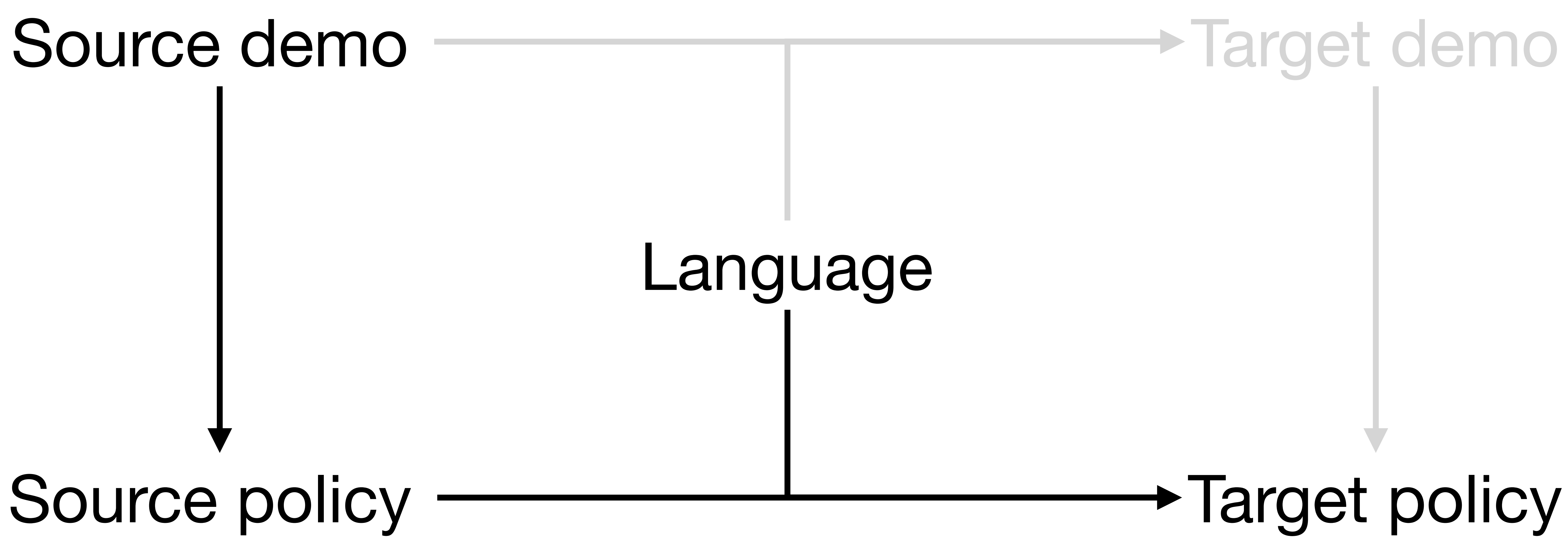
- Introduction
- Related Work
- Completed Work:
  - Using Natural Language for Reward Shaping in Reinforcement Learning (IJCAI 2019)
  - Guiding Reinforcement Learning by Mapping Pixels to Rewards (CoRL 2020)
  - Zero-shot Task Adaptation using Natural Language (arXiv, 2021)
- Proposed Work: Short-term
  - Neurosymbolic Model
  - Policy Adaptation
- Proposed Work: Long-term
  - Policy Regularization
  - Bayesian Inference
  - Supervised Attention

Policy Adaptation

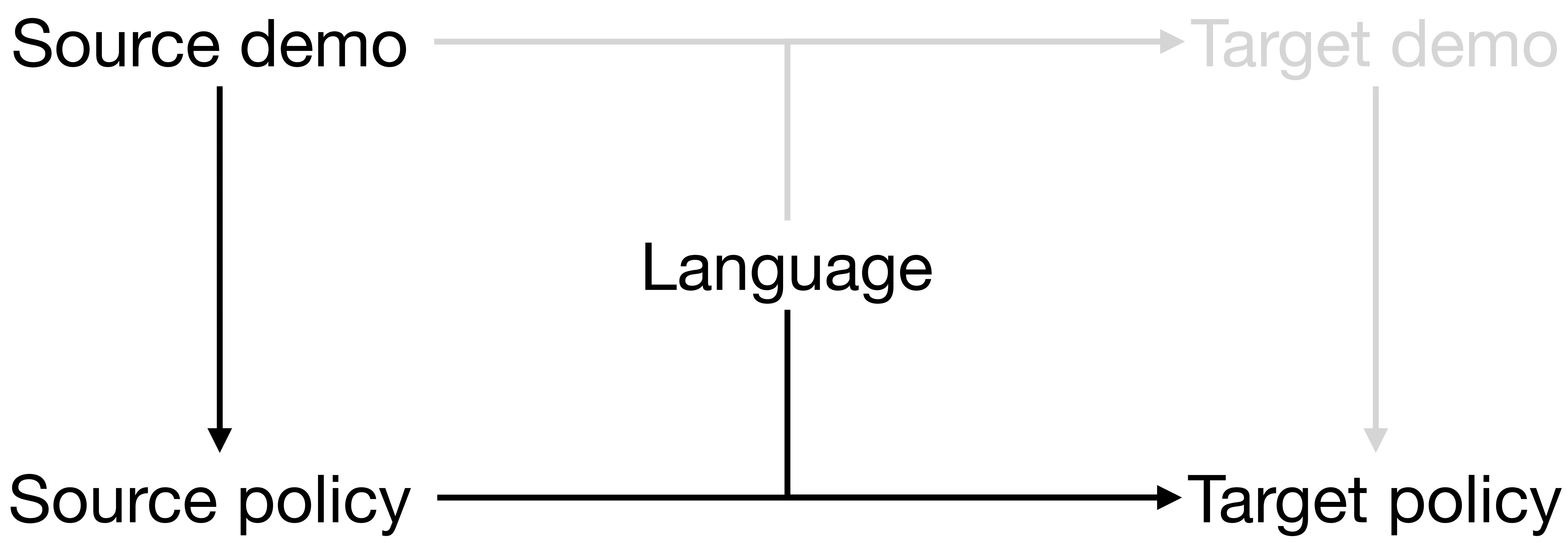
Motivation



Policy Adaptation
Motivation



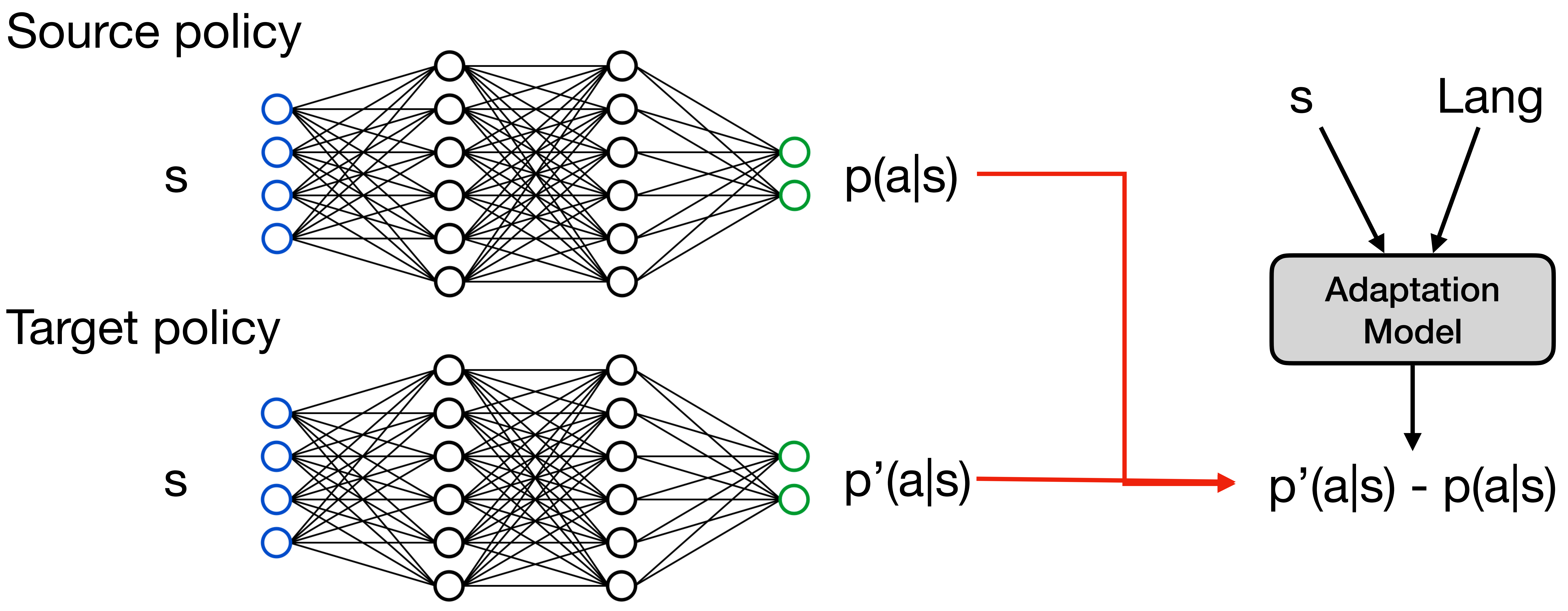
Policy Adaptation
Motivation



Dynamics for the source and target tasks must be identical.

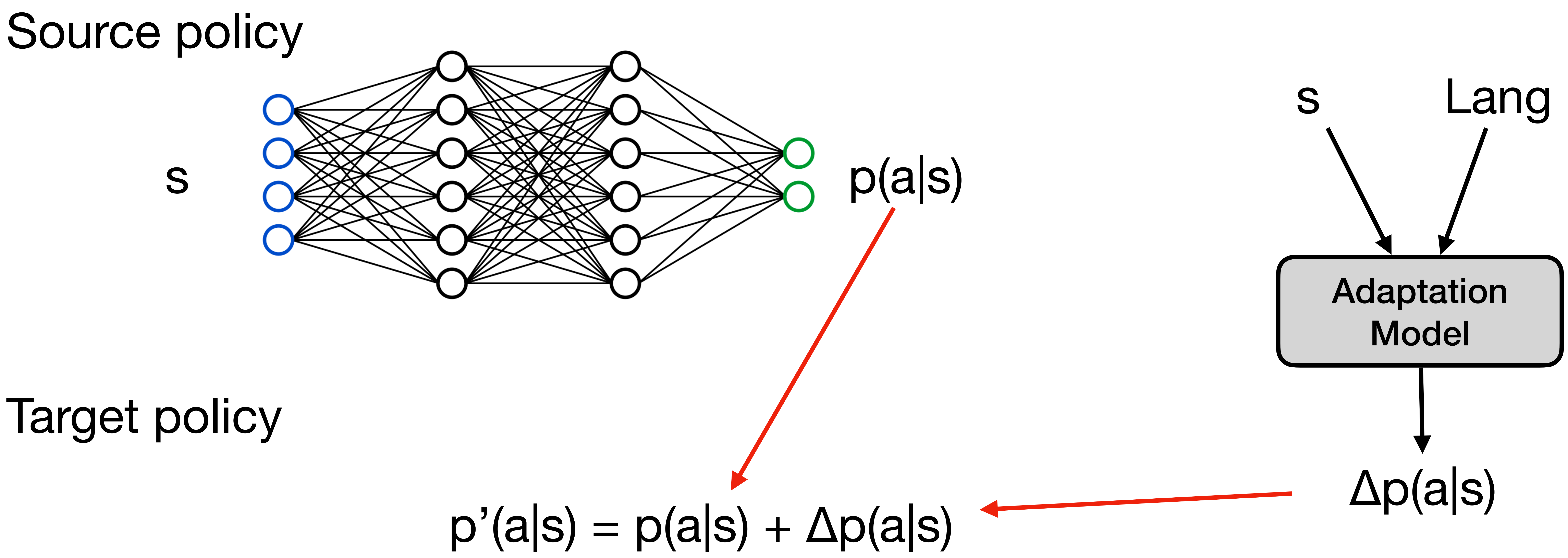
Policy Adaptation — Training

Approach



Policy Adaptation — Inference

Approach



Summary so far…

GOAL: Use language to reduce the burden of task design on the user.

<u>Language to generate rewards for RL</u>

- LEARN: Framework that predicts the relatedness between past actions and language, which can be used as intermediate rewards.

- PixL2R: Extends LEARN framework to predict relatedness between states and language.

=> Leads to faster policy training, both in sparse and dense reward settings.

<u>Language for task adaptation in IL</u>

- LARVA: Framework to predict the target reward or value function, given a source demo and a description of how the source and target tasks differ.

- Neurosymbolic model: Learn an adaptation model to predict production rules + entities for target task.

- Policy adaptation using language

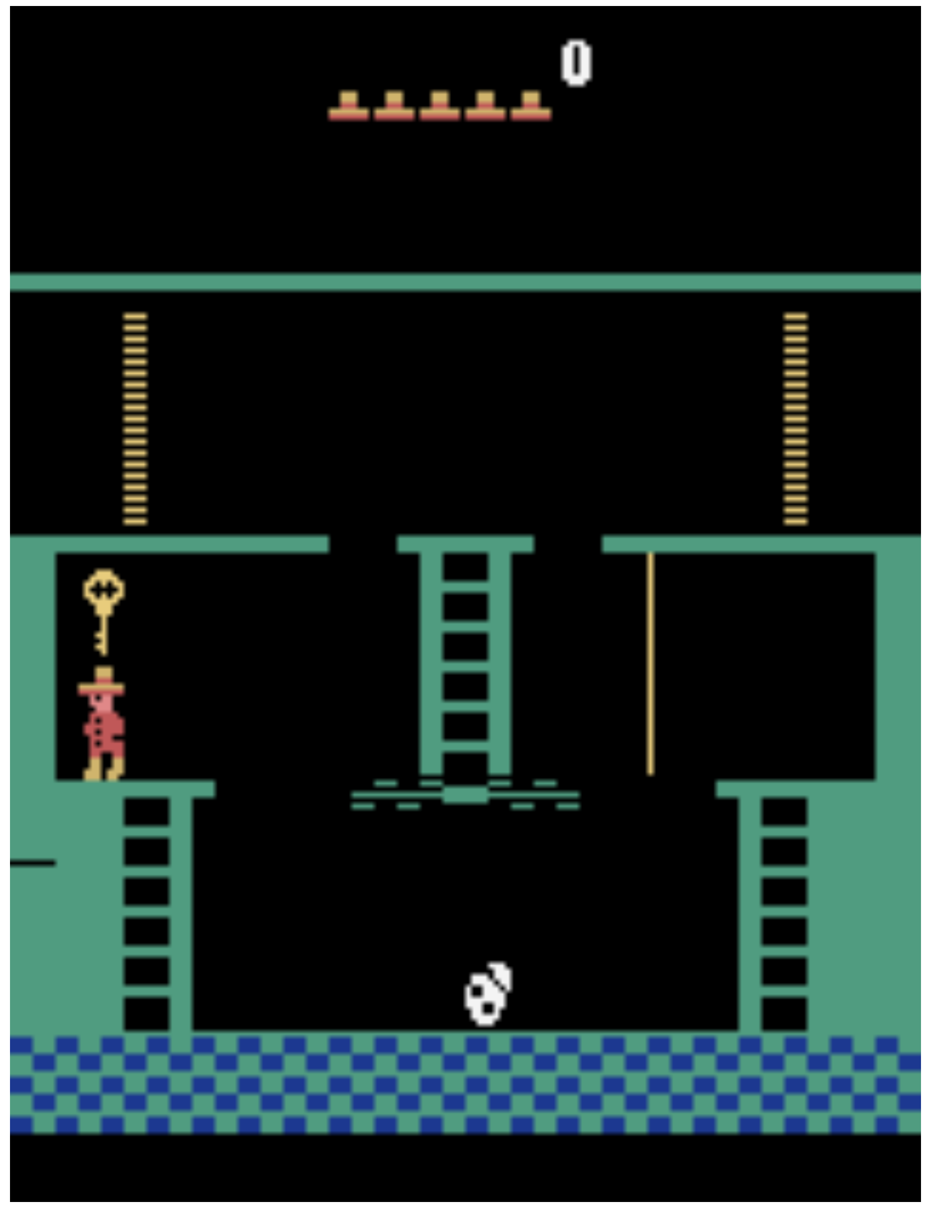
=> Enables learning from demonstrations of related tasks.

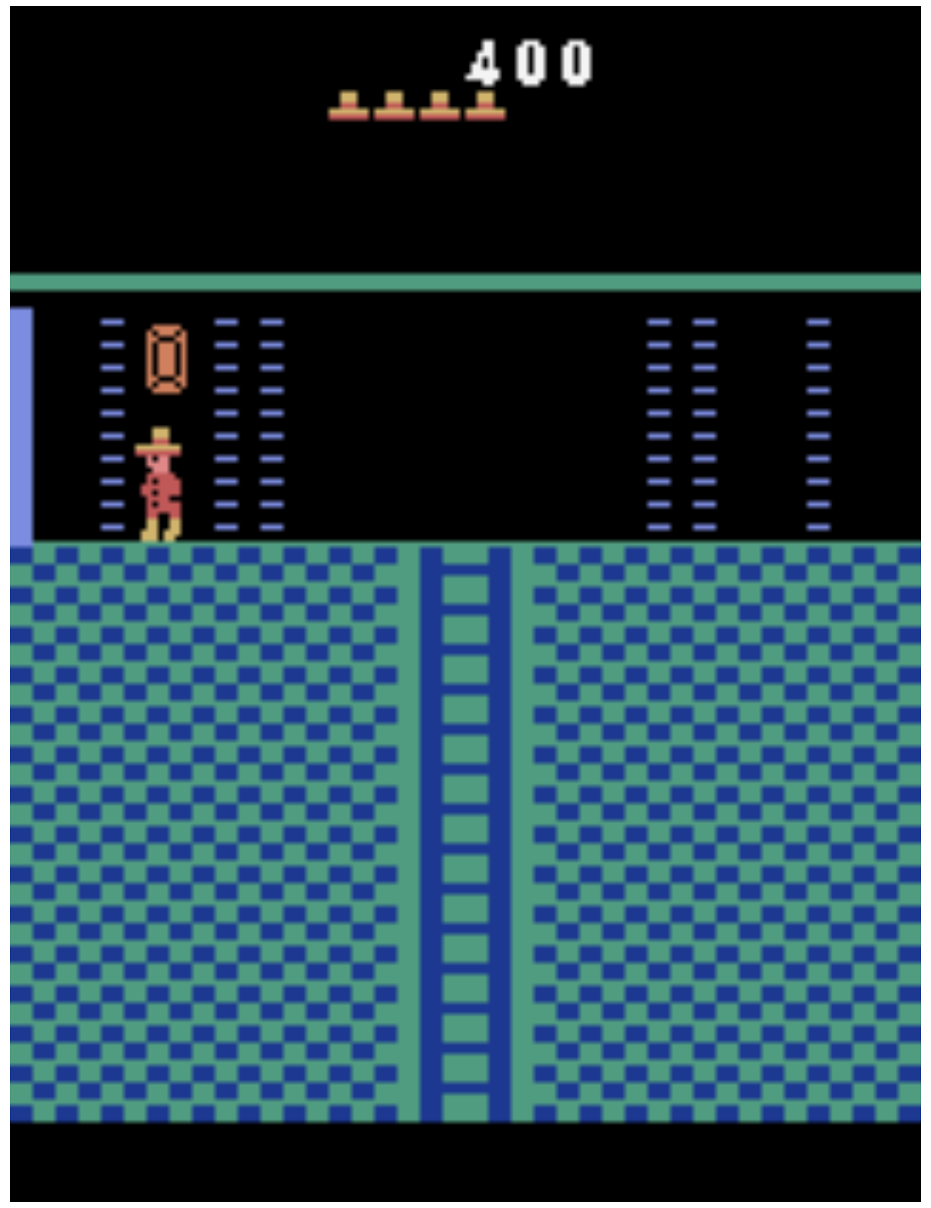
Outline

- Introduction
- Related Work
- Completed Work:
  - Using Natural Language for Reward Shaping in Reinforcement Learning (IJCAI 2019)
  - Guiding Reinforcement Learning by Mapping Pixels to Rewards (CoRL 2020)
  - Zero-shot Task Adaptation using Natural Language (arXiv, 2021)
- Proposed Work: Short-term
  - Neurosymbolic Model
  - Policy Adaptation
- Proposed Work: Long-term
  - Policy Regularization
  - Bayesian Inference
  - Supervised Attention

Long Term Future Directions

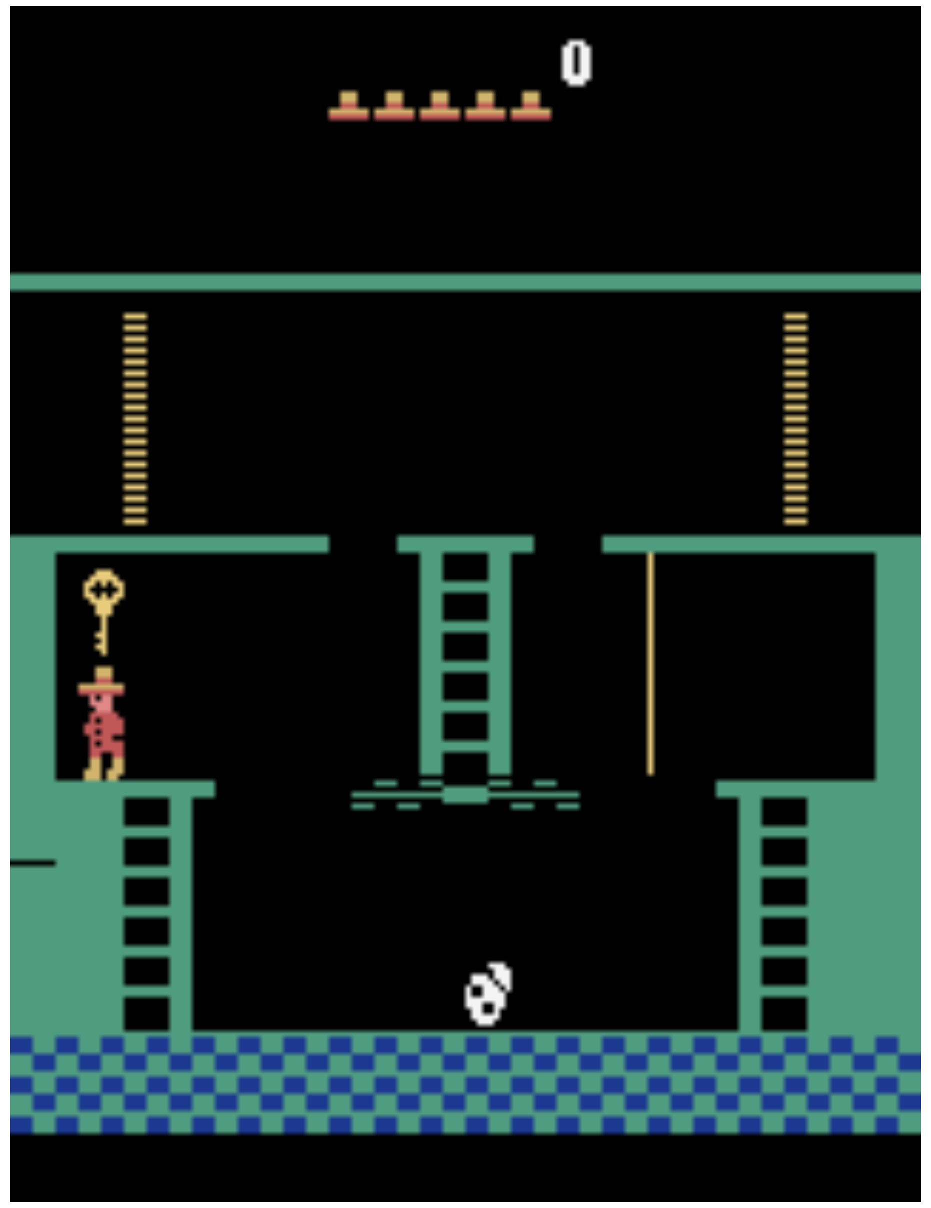
Policy Regularization



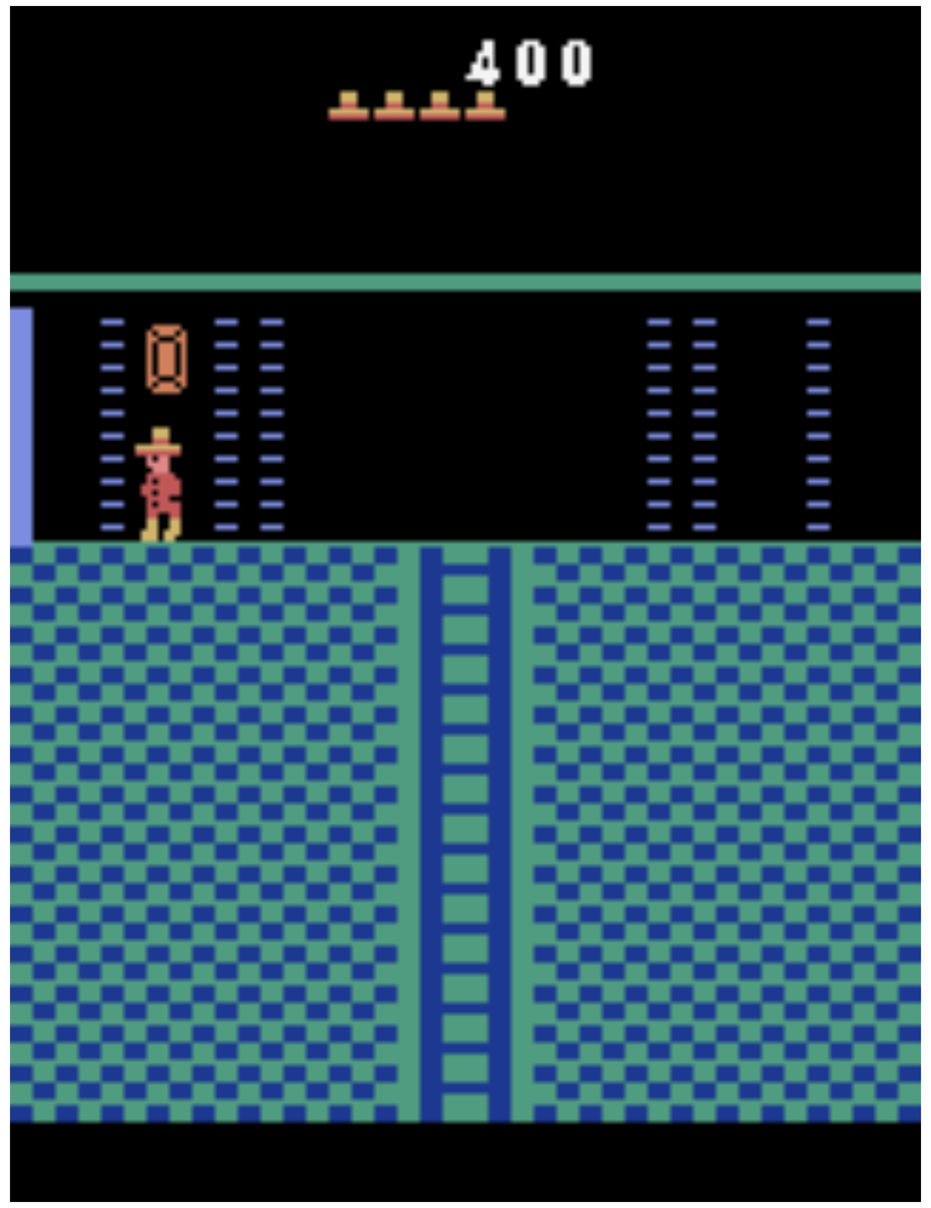


Long Term Future Directions

Policy Regularization



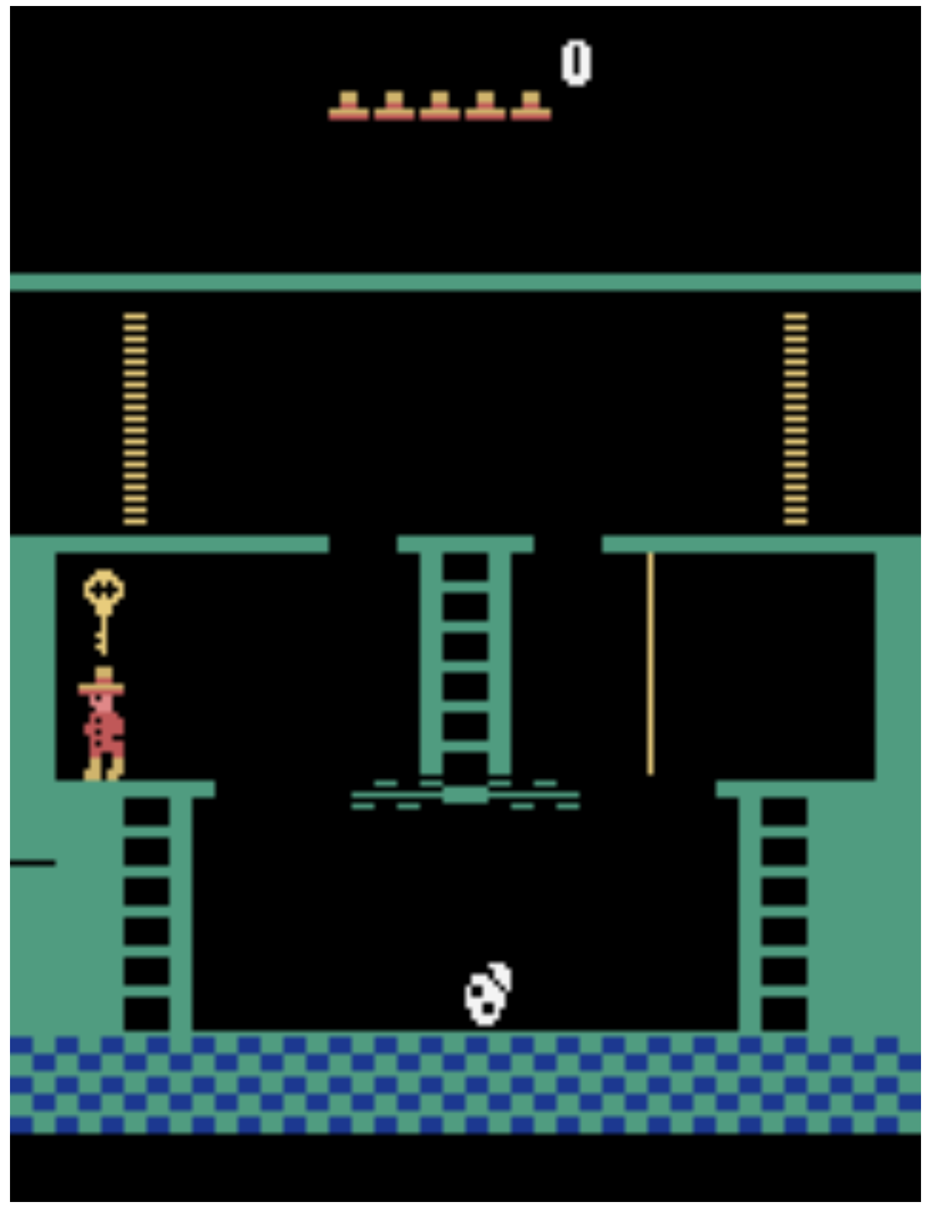
Jump to collect
the key



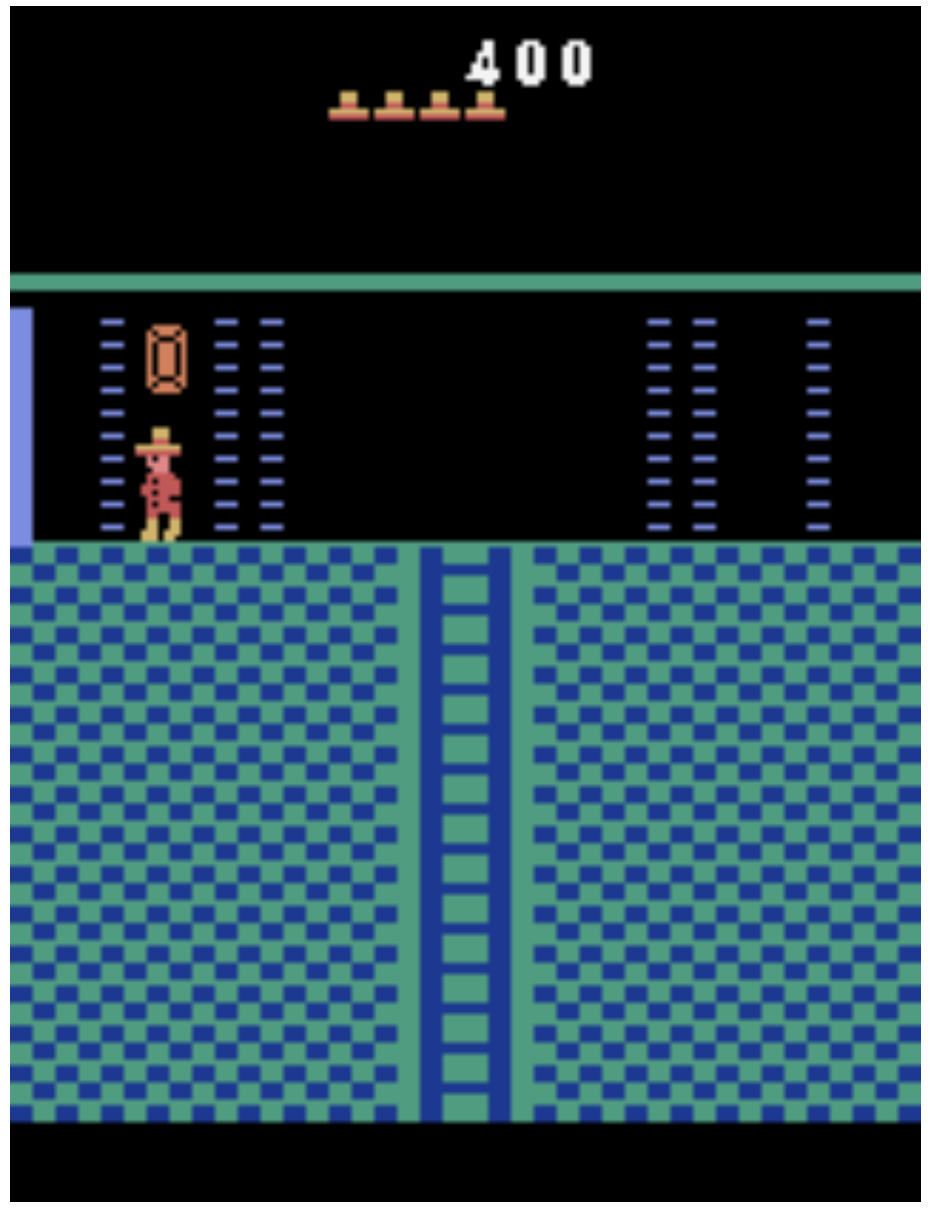
Jump to collect
the orb

Long Term Future Directions

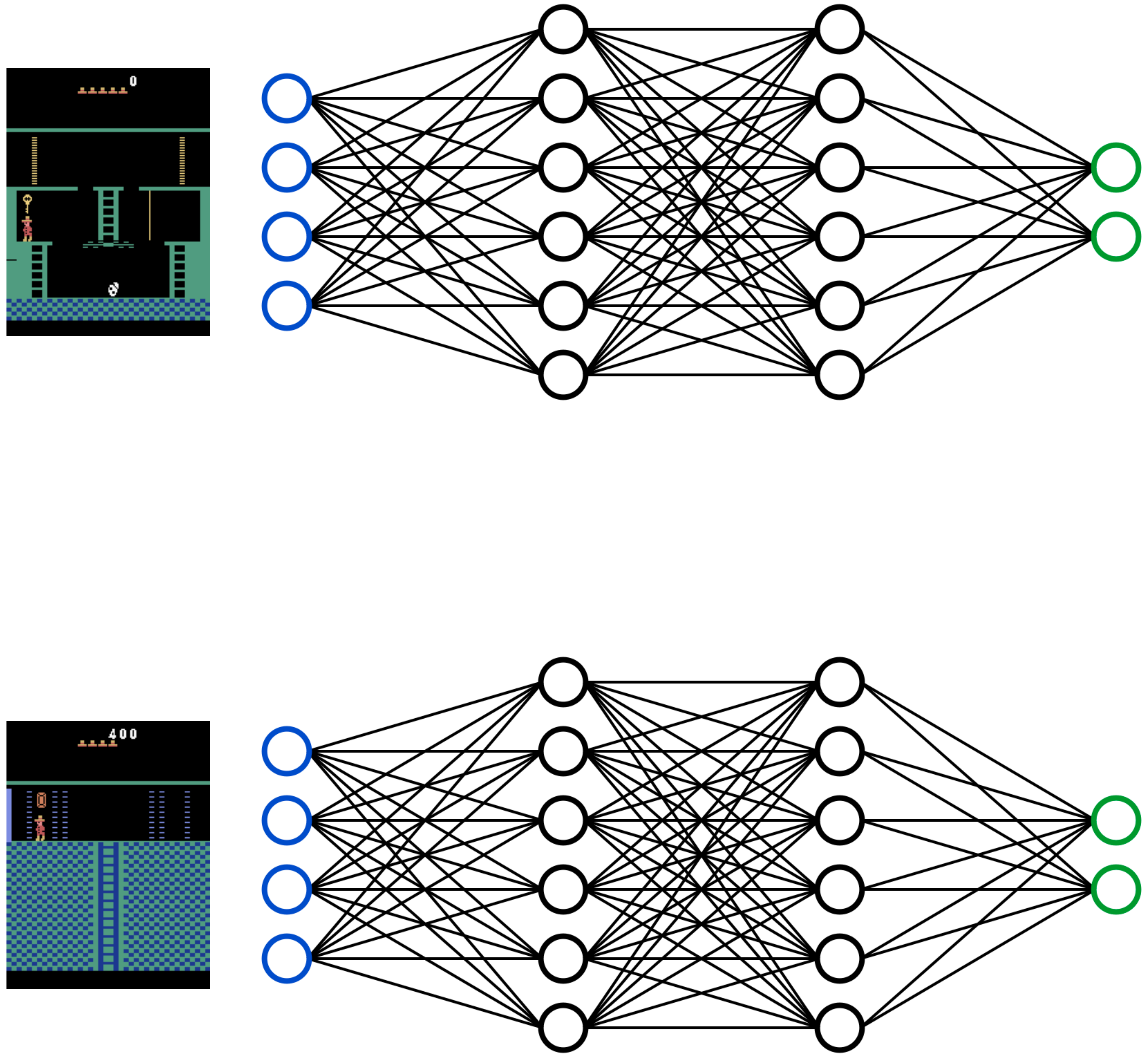
Policy Regularization



Jump to collect
the key

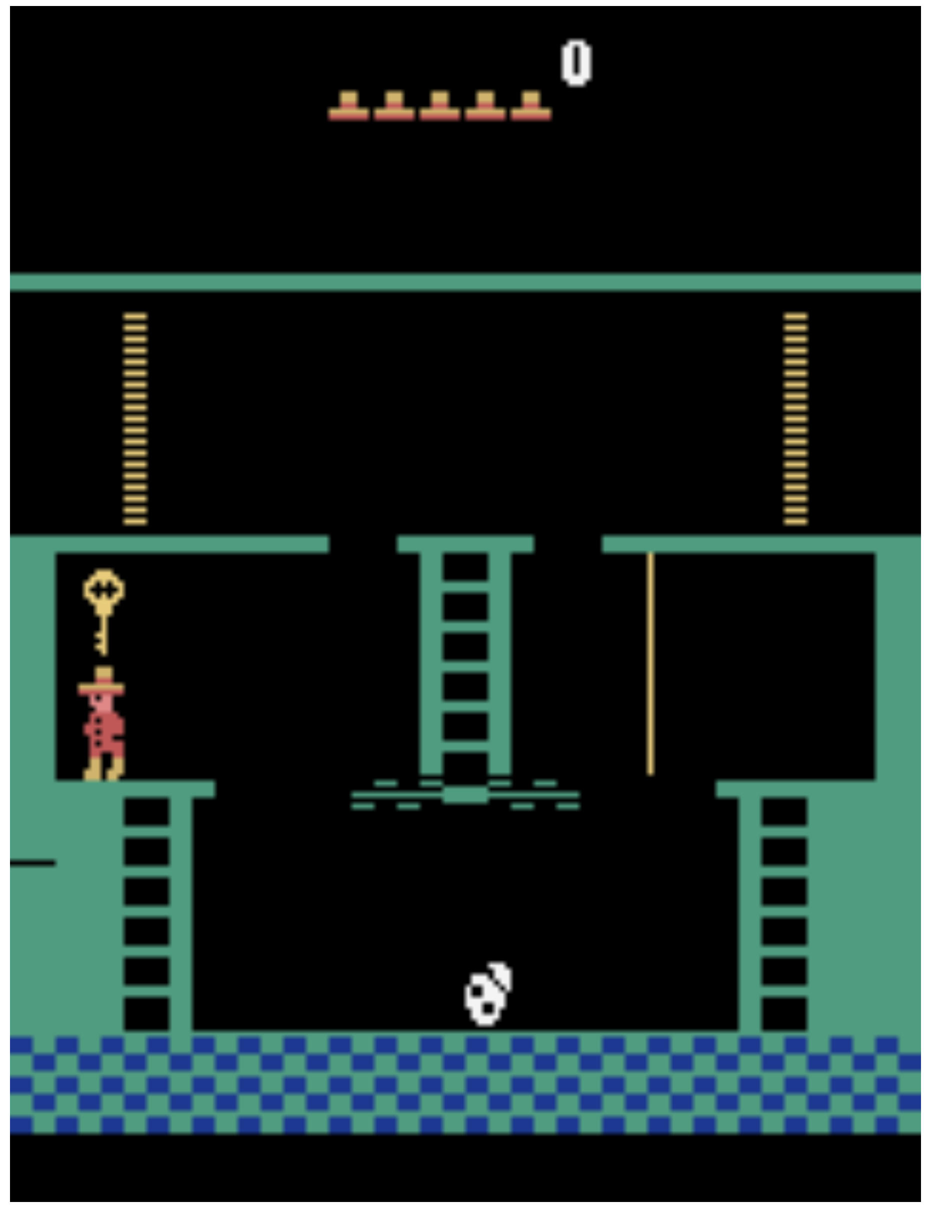


Jump to collect
the orb

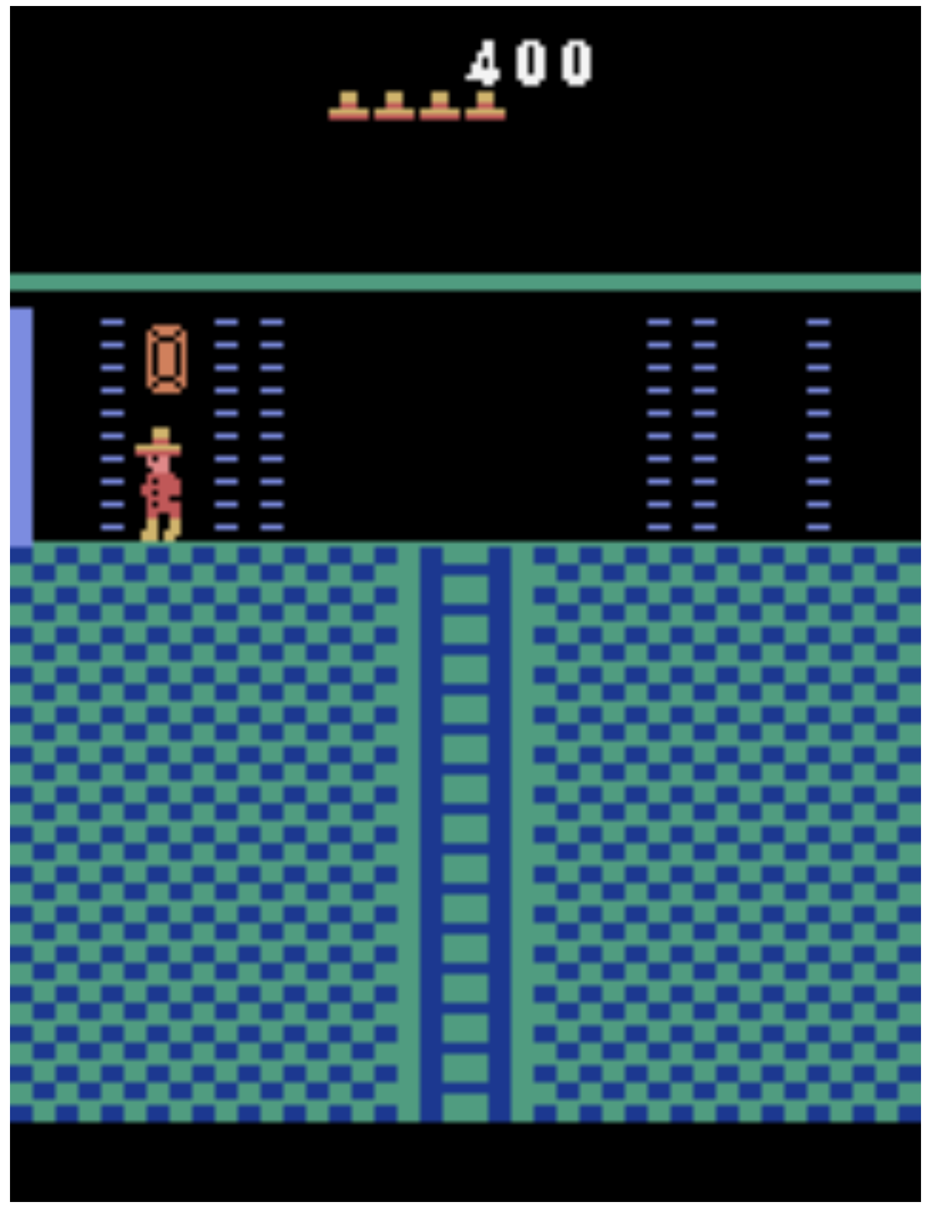


Long Term Future Directions

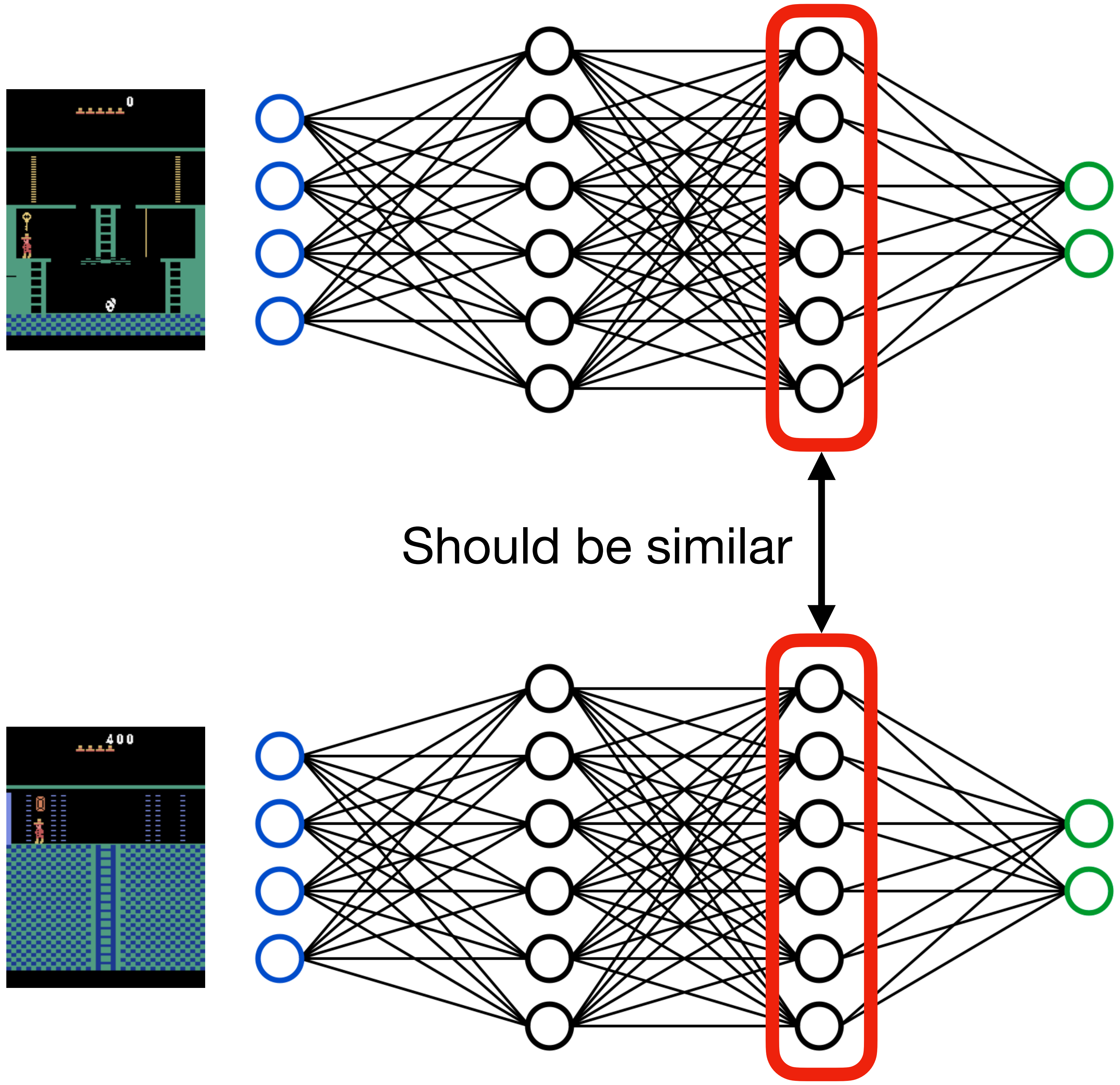
Policy Regularization



Jump to collect
the key



Jump to collect
the orb



Outline

- Introduction
- Related Work
- Completed Work:
  - Using Natural Language for Reward Shaping in Reinforcement Learning (IJCAI 2019)
  - Guiding Reinforcement Learning by Mapping Pixels to Rewards (CoRL 2020)
  - Zero-shot Task Adaptation using Natural Language (arXiv, 2021)
- Proposed Work: Short-term
  - Neurosymbolic Model
  - Policy Adaptation
- Proposed Work: Long-term
  - Policy Regularization
  - Bayesian Inference
  - Supervised Attention

Long Term Future Directions

Bayesian Inference

Bayesian IRL :
[Ramachandran et al., 2007]

$$p(R|\mathcal{D}) = p(\mathcal{D}|R)p(R)$$

Bayesian Instruction-following :
[MacGlashan et al., 2015]

$$p(R|\mathcal{L}) = p(\mathcal{L}|R)p(R)$$

Long Term Future Directions

Bayesian Inference

Bayesian IRL :
[Ramachandran et al., 2007]

$$p(R|\mathcal{D}) = p(\mathcal{D}|R)p(R)$$

Bayesian Instruction-following :
[MacGlashan et al., 2015]

$$p(R|\mathcal{L}) = p(\mathcal{L}|R)p(R)$$

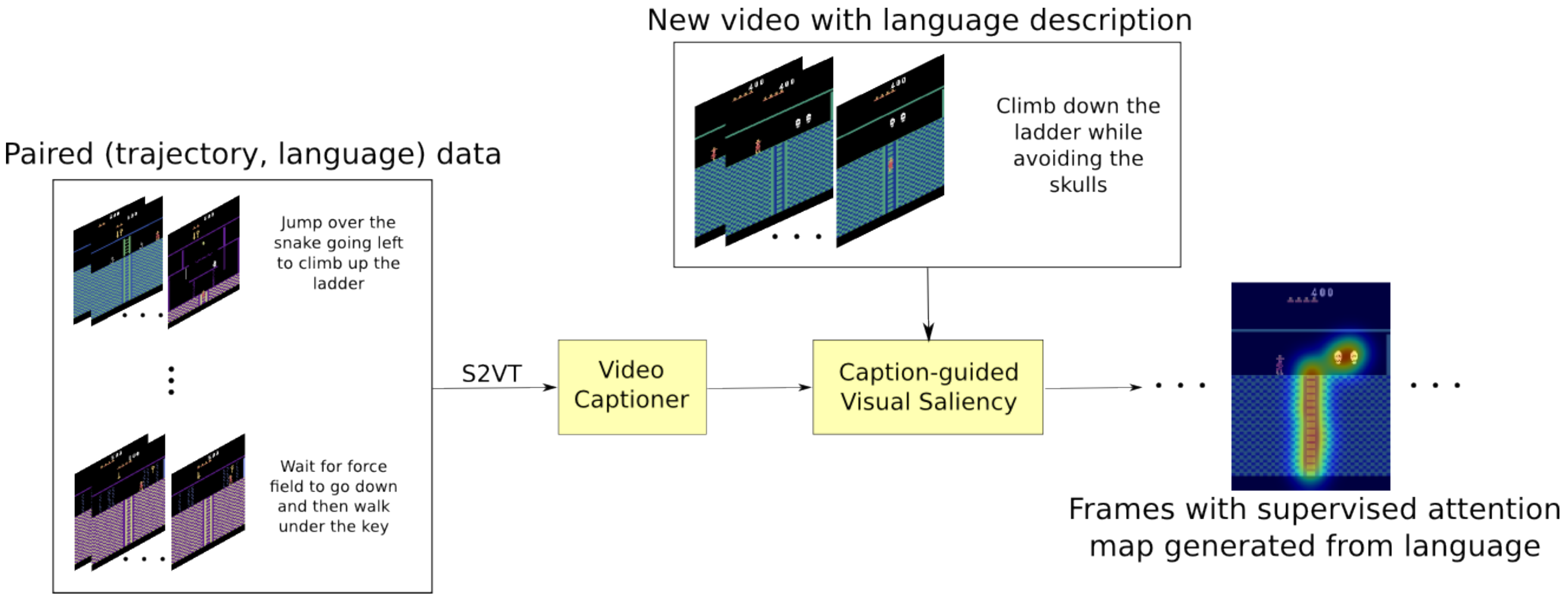
$$p(R|\mathcal{D}, \mathcal{L}) = p(\mathcal{D}, \mathcal{L}|R)p(R) = p(\mathcal{D}|R)p(\mathcal{L}|R)p(R)$$

Outline

- Introduction
- Related Work
- Completed Work:
  - Using Natural Language for Reward Shaping in Reinforcement Learning (IJCAI 2019)
  - Guiding Reinforcement Learning by Mapping Pixels to Rewards (CoRL 2020)
  - Zero-shot Task Adaptation using Natural Language (arXiv, 2021)
- Proposed Work: Short-term
  - Neurosymbolic Model
  - Policy Adaptation
- Proposed Work: Long-term
  - Policy Regularization
  - Bayesian Inference
  - Supervised Attention

Long Term Future Directions

Supervised Attention



[Venugopalan et al., 2015; Ramanishka et al., 2017]

Questions?