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?
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 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.
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

Blukis et al., 2018

Anderson et al., 2018

Tellex et al., 2011

Shridhar et al., 2020

“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"
Related Work
Meta-Learning and Few-shot Learning

**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, \(\Rightarrow\) Few-shot learning
- learn a more robust model,
- converge to a solution faster,
- ...

Figure 1: The robot learns to place a new object into a new container from a single demonstration.
Related Work
Language to Aid Learning

Andreas et al., 2017
Narasimhan et al., 2018

concept learning:

evaluation:

true

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

Use natural language to aid these!

Imitation Learning

Task specification:
Providing demonstrations
LanguagE-Action Reward Network (LEARN)

Motivation

Use natural language to bridge this gap!
LanguagE-Action Reward Network (LEARN)

Motivation
LanguagE-Action Reward Network (LEARN)

Motivation

[Bellemare et al., 2013]
Can we use natural language to provide intermediate rewards to the agent?

Jump over the skull while going to the left.

[Bellemare et al., 2013]
LanguagE-Action Reward Network (LEARN)

Approach

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

Jump over the skull while going to the left.
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,

<table>
<thead>
<tr>
<th>Past actions</th>
<th>Reward</th>
</tr>
</thead>
<tbody>
<tr>
<td>LLLLJLLL</td>
<td>High</td>
</tr>
<tr>
<td>RRRUULL</td>
<td>Low</td>
</tr>
</tbody>
</table>

[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,

<table>
<thead>
<tr>
<th>Past actions</th>
<th>Reward</th>
</tr>
</thead>
<tbody>
<tr>
<td>4441444</td>
<td>High</td>
</tr>
<tr>
<td>3332244</td>
<td>Low</td>
</tr>
</tbody>
</table>

[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*:

\[
\begin{align*}
\epsilon & \Rightarrow [0.00 \ 0.00 \ 0.00 \ 0.00 \ 0.00 \ 0.00 \ 0.00 \ 0.00] \\
4 & \Rightarrow [0.00 \ 0.00 \ 0.00 \ 0.00 \ 1.00 \ 0.00 \ 0.00 \ 0.00] \\
42 & \Rightarrow [0.00 \ 0.00 \ 0.50 \ 0.00 \ 0.50 \ 0.00 \ 0.00 \ 0.00] \\
422 & \Rightarrow [0.00 \ 0.00 \ 0.67 \ 0.00 \ 0.33 \ 0.00 \ 0.00 \ 0.00]
\end{align*}
\]

Train a neural network — LanguageE 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

Please enter the description below:

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

[Kurin et al., 2017]
LanguagE-Action Reward Network (LEARN)

Data Collection

Clip 1:

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

[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.
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
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

Jump over the skull while going to the left.

Agent

Reward Network (LEARN)

Past actions \( (a_1, \ldots, a_{i-1}) \)

Action-frequency vector

Proportionalities (RELATED/UNRELATED)

Language-based rewards

Environment

Action

Reward, Observation

(LEARN)
Pixels and Language to Rewards (PixL2R)

Approach

Jump over the skull while going to the left.

Sequence of past states $(s_1, ..., s_{i-1})$ → Pixels and Language to Rewards (PixL2R) → Relatedness Score

Agent

Language-based rewards → Environment

Reward, Observation → Action
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**

![Diagram showing starting position, correct object, and incorrect object with rewards +1 and -1.](image-url)
Pixels and Language to Rewards (PixL2R)

Training Objective

Regression:

- Starting position
- Correct object
- Incorrect object
Pixels and Language to Rewards (PixL2R)

Results

Graph showing the number of successful episodes over timestep (in thousands) for different reward conditions.
Pixels and Language to Rewards (PixL2R)

Results

![Graph showing the number of successful episodes over time for Sparse and Sparse+RGR scenarios.](image1)

![Graph showing the number of successful episodes over time for Dense and Dense+RGR scenarios.](image2)
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

Use natural language to aid these!

Imitation Learning

Task specification:
Providing demonstrations
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

GOAL:
Learn the target task *without any demonstrations.*
Language-Aided Reward and Value Adaptation (LARVA)

Approach

Target Goal Prediction:

Move the blue cube to the final position of the red tall block, and vice versa.
Language-Aided Reward and Value Adaptation (LARVA)

Approach

**Target Goal Prediction:**
- Source task demonstration
- LSTM encoder

**Reward/Value Prediction:**
- Source task demonstration
- Target Goal Predictor

**Reward / Value Network:**
- Predicted reward / value of state $s$

Move the blue cube to the final position of the red tall block, and vice versa.
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:

<table>
<thead>
<tr>
<th>Type</th>
<th>Original State</th>
<th>Modified State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Same object, different place location</td>
<td>A → P</td>
<td>A → Q</td>
</tr>
<tr>
<td>Different object, same place location</td>
<td>A → P</td>
<td>B → P</td>
</tr>
<tr>
<td>Move two objects, with swapped final locations</td>
<td>A → P</td>
<td>A → Q</td>
</tr>
<tr>
<td></td>
<td>B → Q</td>
<td>B → P</td>
</tr>
<tr>
<td>Delete a step</td>
<td>A → P</td>
<td>B → Q</td>
</tr>
<tr>
<td></td>
<td>B → Q</td>
<td>C → R</td>
</tr>
<tr>
<td>Insert a step</td>
<td>B → Q</td>
<td>A → P</td>
</tr>
<tr>
<td></td>
<td>C → R</td>
<td>B → Q</td>
</tr>
<tr>
<td>Modify a step</td>
<td>A → P</td>
<td>A → P</td>
</tr>
<tr>
<td></td>
<td>B → Q</td>
<td>D → S</td>
</tr>
<tr>
<td></td>
<td>C → R</td>
<td>C → R</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Success rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Synthetic</td>
</tr>
</tbody>
</table>

59
## Language-Aided Reward and Value Adaptation (LARVA)

### Experiments

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Success rate (%)</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Synthetic</td>
<td>Natural</td>
</tr>
<tr>
<td>1. LARVA; reward prediction</td>
<td>97.8</td>
<td>75.7</td>
<td></td>
</tr>
<tr>
<td>2. LARVA; value prediction</td>
<td>97.7</td>
<td>73.3</td>
<td></td>
</tr>
</tbody>
</table>
## Experiments

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Success rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Synthetic</td>
</tr>
<tr>
<td>1. LARVA; reward prediction</td>
<td>97.8</td>
</tr>
<tr>
<td>2. LARVA; value prediction</td>
<td>97.7</td>
</tr>
<tr>
<td>3. LARVA; no target goal supervision</td>
<td>20.0</td>
</tr>
</tbody>
</table>
Language-Aided Reward and Value Adaptation (LARVA)

Experiments

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Success rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Synthetic</td>
</tr>
<tr>
<td>1. LARVA; reward prediction</td>
<td>97.8</td>
</tr>
<tr>
<td>2. LARVA; value prediction</td>
<td>97.7</td>
</tr>
<tr>
<td>3. LARVA; no target goal supervision</td>
<td>20.0</td>
</tr>
<tr>
<td>4. LARVA; no language</td>
<td>20.7</td>
</tr>
<tr>
<td>5. LARVA; no source demonstration</td>
<td>4.2</td>
</tr>
</tbody>
</table>
## Experiments

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Success rate (%)</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Synthetic</td>
<td>Natural</td>
<td></td>
</tr>
<tr>
<td>1. LARVA; reward prediction</td>
<td>97.8</td>
<td>75.7</td>
<td></td>
</tr>
<tr>
<td>2. LARVA; value prediction</td>
<td>97.7</td>
<td>73.3</td>
<td></td>
</tr>
<tr>
<td>3. LARVA; no target goal supervision</td>
<td>20.0</td>
<td>2.7</td>
<td></td>
</tr>
<tr>
<td>4. LARVA; no language</td>
<td>20.7</td>
<td>22.3</td>
<td></td>
</tr>
<tr>
<td>5. LARVA; no source demonstration</td>
<td>4.2</td>
<td>3.3</td>
<td></td>
</tr>
<tr>
<td>6. NN without decomposition</td>
<td>1.8</td>
<td>1.0</td>
<td></td>
</tr>
</tbody>
</table>
## Language-Aided Reward and Value Adaptation (LARVA)

### Experiments

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Success rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Synthetic</td>
</tr>
<tr>
<td>1. LARVA; reward prediction</td>
<td>97.8</td>
</tr>
<tr>
<td>2. LARVA; value prediction</td>
<td>97.7</td>
</tr>
<tr>
<td>3. LARVA; no target goal supervision</td>
<td>20.0</td>
</tr>
<tr>
<td>4. LARVA; no language</td>
<td>20.7</td>
</tr>
<tr>
<td>5. LARVA; no source demonstration</td>
<td>4.2</td>
</tr>
<tr>
<td>6. NN without decomposition</td>
<td>1.8</td>
</tr>
<tr>
<td>7. LARVA: Compostionality – red box</td>
<td>87.6</td>
</tr>
<tr>
<td>8. LARVA: Compostionality – blue cylinder</td>
<td>89.4</td>
</tr>
</tbody>
</table>
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!
Motivation

Neural Production Systems
[Goyal et al., 2021]

Not me!
Task Adaptation: Neurosymbolic Model

Motivation

Neural Production Systems
[Goyal et al., 2021]
Not me!

Entities: Vectors
Rules: Neural Networks
\{ Learned from data

<table>
<thead>
<tr>
<th>Rule 1</th>
<th>Rule 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.84</td>
<td>0.16</td>
</tr>
<tr>
<td>0.62</td>
<td>0.38</td>
</tr>
<tr>
<td>0.29</td>
<td>0.71</td>
</tr>
</tbody>
</table>
In the third step, move the green flat block from bottom left to top left.
In the third step, move the green flat block from bottom left to top left.
In the third step, move the green flat block from bottom left to top left.
In the third step, move the green flat block from bottom left to 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
In the third step, move the green flat block from bottom left to top left.

Adaptation Model

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

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 — Inference

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-right

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

Adaptation Model

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

To_BottomLeft(E3)
In the third step, move the green flat block from bottom left to top left.

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.
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

Source demo \(\rightarrow\) Target demo

Language

Target policy
Policy Adaptation

Motivation
Policy Adaptation

Motivation

Dynamics for the source and target tasks must be identical.
Policy Adaptation — Training

Approach

Source policy

Target policy

s

p(a|s)

p'(a|s)

s

Lang

Adaptation Model

p'(a|s) - p(a|s)
Policy Adaptation — Inference

Source policy

Target policy

\[ p'(a|s) = p(a|s) + \Delta p(a|s) \]
Summary so far…

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

**Language to generate rewards for RL**

- **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.

**Language for task adaptation in IL**

- **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

Should be similar
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]  
\[ p(R|D) = p(D|R)p(R) \]

Bayesian Instruction-following:  
[MacGlashan et al., 2015]  
\[ p(R|L) = p(L|R)p(R) \]
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

Paired (trajectory, language) data

Jump over the snake going left to climb up the ladder

Wait for force field to go down and then walk under the key

New video with language description

Climb down the ladder while avoiding the skulls

Video Captioner

Caption-guided Visual Saliency

Frames with supervised attention map generated from language

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