Using Natural Language for Task Specification in Sequential Decision Making Problems

Prasoon Goyal

Dissertation Defense
A lot of progress in AI in the last several decades...
Sequential Decision Making

Can be used for a broad class of real-world tasks:
- Turn-based games like chess, Go, etc.
- Video games
- Cooking tasks
- ...
Sequential Decision Making

Can be used for a broad class of real-world tasks:
- Turn-based games like chess, Go, etc.
- Video games
- Cooking tasks
- ...

Rewards

Describing the desired task to the agent

Demonstrations

Reinforcement Learning (RL)

Imitation Learning (IL)
How can natural language be used as an auxiliary signal to reduce the burden of task specification on the end user?
Sequential Decision Making

Challenge          Solution

RL

IL
Talk Outline

Background

Core Contributions:

Challenge  Solution

Sequential Decision Making

RL

IL

Future Directions
Talk Outline

Background

Core Contributions:

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>RL</td>
<td></td>
</tr>
<tr>
<td>IL</td>
<td></td>
</tr>
</tbody>
</table>

Sequential Decision Making

Future Directions
Background: Language Encoders

- One-hot embeddings of words + RNN to encode the sequence
- Word embeddings + RNN to encode the sequence
- Transformer-based pretrained sentence encoders (e.g. BERT, CLIP)
- Transformer-based sentence encoder trained from scratch

Turn the red handle down.
Talk Outline

Background

Core Contributions:

- Sequential Decision Making

Challenge | Solution

- RL

Future Directions
Reinforcement Learning

Markov Decision Process (MDP), \( M = \langle S, A, T, R, \gamma \rangle \)

- \( S \): State space
- \( A \): Action space
- \( T \): Transition function
- \( R \): Reward function
- \( \gamma \): Discount factor

Policy: Maps states to action probabilities

Objective: Learn a policy that maximizes the discounted sum of future rewards.
Reinforcement Learning: Proximal Policy Optimization (PPO)
Challenge in RL: Designing rewards is Hard

- Dense: Better
- Sparse

Ease of Designing vs. Ease of Learning
Challenge in RL: Designing rewards is Hard

Ease of Designing

Ease of Learning

Dense

Sparse

Better

Use natural language to generate auxiliary rewards
Talk Outline

Background

Core Contributions:

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>RL</td>
<td>Language-based Rewards</td>
</tr>
</tbody>
</table>

Sequential Decision Making

IL

Future Directions
Related Work: Instruction-following

- **Chen and Mooney, 2011**
- **Blukis et al., 2018**
- **Anderson et al., 2018**
- **Tellex et al., 2011**
- **Bisk et al., 2018**
- **Shridhar et al., 2020**

**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.

“On the (new) fourth tower, mirror Nvidia with UPS.”
Talk Outline

Background

Core Contributions:

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>RL</td>
<td>Reward design</td>
</tr>
<tr>
<td>IL</td>
<td>Language-based Rewards</td>
</tr>
</tbody>
</table>

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

Jump over the skull while going to the left.
LanguagE Action Reward Network (LEARN): Approach

- 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.
LanguagE Action Reward Network (LEARN): Approach

Language Action Reward Network (LEARN)

Related / Unrelated

Action-frequency vector

Trajectory

Language

Clip 1:

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. climbing down the ladder
7. ladder down 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.

Please enter the description below:
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.
Talk Outline

Background

Core Contributions:

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reward design</td>
<td>Language-based Rewards</td>
</tr>
</tbody>
</table>

RL  +  LEARN  =  PixL2R

Future Directions

Core Contributions:

Sequential Decision Making

IL
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): Data Augmentation

- Frame dropping

- Partial trajectories
Pixels and Language to Rewards (PixL2R): Training Objective

**Classification:**
- Blue circle: Starting position
- Green circle: Correct object
- Red circle: Incorrect object

Diagram:
- Top left: +1
- Top right: +1
- Bottom left: -1
- Bottom right: -1
Pixels and Language to Rewards (PixL2R): Training Objective

Classification: Regression:

- Starting position
- Correct object
- Incorrect object

+1

-1

+0.3

-0.3
Pixels and Language to Rewards (PixL2R): Results
Pixels and Language to Rewards (PixL2R): Results

The graphs show the number of successful episodes over time for different conditions. The x-axis represents the number of timesteps (in thousands), and the y-axis represents the number of successful episodes.
Pixels and Language to Rewards (PixL2R): Results

New RL training regime: Coarse hand-designed dense rewards + Language-based rewards
Talk Outline

Background

Core Contributions:

- **Challenge**
  - RL: Reward design
  - Language-based Rewards

- **Solution**
  - LEARN
  - PixL2R

Future Directions
Imitation Learning

3 broad classes of approaches:

1. Behavior cloning:
   - Supervised Learning approach
   - Use state-action pairs in the demonstrations

2. Inverse Reinforcement Learning (IRL):
   - Infer a reward function from demonstrations
   - Use RL to learn a policy

3. Adversarial Imitation Learning (AIL):
   - Generator-discriminator-based approach
Challenge in IL: A lot of demonstrations are needed

- Multiple demonstrations are often needed to specify a task.
Challenge in IL: A lot of demonstrations are needed

- Multiple demonstrations are often needed to specify a task.

- A new set of demonstrations is needed for each new task.
Challenge in IL: A lot of demonstrations are needed

- Multiple demonstrations are often needed to specify a task.

- A new set of demonstrations is needed for each new task.

A lot of demonstrations may need to be given to teach multiple tasks.
Challenge in IL: A lot of demonstrations are needed

- Multiple demonstrations are often needed to specify a task.

A lot of demonstrations may need to be given to teach multiple tasks.

- A new set of demonstrations is needed for each new task.

Use natural language to reuse demos from related tasks.
Talk Outline

Background

Core Contributions:

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>RL Reward design</td>
<td>Language-based Rewards</td>
</tr>
<tr>
<td>IL Many demos needed</td>
<td>Language-guided Task Adaptation</td>
</tr>
</tbody>
</table>

Future Directions

Core Contributions:

- Sequential Decision Making
- IL Many demos needed
- RL Reward design
- Language-based Rewards

Future Directions

Core Contributions:

- LEARN
- PixL2R

Core Contributions:

- Sequential Decision Making
- IL Many demos needed
- RL Reward design
- Language-based Rewards

Future Directions
Related Work: Transfer Learning

Source Task

![Diagram showing the flow of data, model, and predictions from the source task.]

Target Task

![Diagram showing the flow of data, model, and predictions from the target task.]

Transfer
Talk Outline

Background

Core Contributions:

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>RL</td>
<td>Language-based Rewards</td>
</tr>
<tr>
<td>IL</td>
<td>Language-guided Task Adaptation</td>
</tr>
<tr>
<td>Many demos needed</td>
<td></td>
</tr>
<tr>
<td>Reward design</td>
<td></td>
</tr>
</tbody>
</table>

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

GOAL: Learn the target task \textit{without any demonstrations}. 

Language-Aided Reward and Value Adaptation (LARVA): Motivation
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

6 types of adaptations: e.g. add a step, delete a step, etc.

Delete a step

Example Source Task

Example Target Task

Insert a step

Example Source Task

Example Target Task
Language-Aided Reward and Value Adaptation (LARVA): Experiments

6 types of adaptations: e.g. add a step, delete a step, etc.

Language Data:
- Template-based
- Paraphrases from Amazon Mechanical Turk
Language-Aided Reward and Value Adaptation (LARVA): Experiments

6 types of adaptations: e.g. add a step, delete a step, etc.

**Language Data:**
- Template-based
- Paraphrases from Amazon Mechanical Turk

**Evaluation:**

If $s_i$ is the true goal state for the target task, then $success=1$ else $0$. 

\[
\begin{align*}
S_1 & \rightarrow R_1 \\
S_2 & \rightarrow R_2 \\
\vdots & \\
S_i & \rightarrow R_i \\
\vdots & \\
S_N & \rightarrow R_N \\
\end{align*}
\]
Language-Aided Reward and Value Adaptation (LARVA): Experiments

Reward Prediction

- Synthetic Language: 97.8
- Natural Language: 75.7

Value Prediction

- Synthetic Language: 97.7
- Natural Language: 73.3
Talk Outline

Background

Core Contributions:

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>RL</td>
<td>Reward design</td>
</tr>
<tr>
<td>IL</td>
<td>Language-based Rewards</td>
</tr>
<tr>
<td>Many demos needed</td>
<td>Language-guided</td>
</tr>
<tr>
<td></td>
<td>Task Adaptation</td>
</tr>
</tbody>
</table>

Future Directions

Core Contributions:

- LEARN
- PixL2R
- LARVA
- RETAIL
Background: Transformer Model

Scaled Dot-Product Attention

Multi-Head Attention
Background: Relational Reasoning

- Input: Set of entities
- Model reasons about the relationships between these entities
Background: Relational Reasoning

- Input: Set of entities
- Model reasons about the relationships between these entities
- Shown to be effective in
  - RL
  - Visual question answering
  - Learning passive dynamics

[Box-World diagram with entities: Agent, Key, Lock, Loose key, Gem]

What shape is the small object that is in front of the yellow matte thing and behind the gray sphere? [Santoro et al., 2017]

[Image of three boxes with arrows]

[Didolkar et al., 2021]
RElational Task Adaptation for Imitation with Language (RETAIL)

Motivation

Limitations of LARVA:

- Does not reason about the structure of the tasks or the environment.
- Does not have an explicit policy learning phase for evaluation.
- Assumes data of the form (src demo, lang, tgt goal, tgt reward/value function).
RElational Task Adaptation for Imitation with Language (RETAIL)

Approach Overview

Source demo

Language

Target reward

Target policy

Relational Reward Adaptation
RElational Task Adaptation for Imitation with Language (RETAIL)

Approach Overview

Source demo -> Language

Source policy

Target policy

Relational Policy Adaptation
RElational Task Adaptation for Imitation with Language (RETAIL)

Approach Overview

Source demo → Language → Target reward → Target policy

Source policy → Language → Target reward → Target policy

Relational Reward Adaptation

Relational Policy Adaptation
RElational Task Adaptation for Imitation with Language (RETAIL)
Domains: Room Rearrangement

- Agent + 2 objects

- Actions:
  - Up, Down, Left, Right
  - Grasp, Release
  - Stop

- Tasks: Moving each object to desired goal locations.
RElational Task Adaptation for Imitation with Language (RETAIL)

Domains: Room Navigation

- Agent + 4 objects

- Actions:
  - $(\Delta x, \Delta y)$

- Tasks: Navigating to a desired goal location.
RElational Task Adaptation for Imitation with Language (RETAIL)

Domains

Room Rearrangement
- Discrete states and actions
- Short horizon tasks (~30 steps)
- Multiple optimal trajectories

Room Navigation
- Continuous states and actions
- Long horizon tasks (~150 steps)
- Unique optimal trajectory
RElational Task Adaptation for Imitation with Language (RETAIL)
Domains: Objects

<table>
<thead>
<tr>
<th>Table</th>
<th>Chair</th>
<th>Sofa</th>
<th>Light</th>
<th>Shelf</th>
<th>Wardrobe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Large</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wide</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wooden</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Metallic</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Corner</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Foldable</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

36 total (attribute, object) combinations
24 for training
6 for validation
6 for testing
RElational Task Adaptation for Imitation with Language (RETAIL)
Domains: Data Collection

- Planner to generate source and target demos

- For each type of adaptation:
  - Training set: 5000
  - Validation set for supervised learning: 100
  - Validation set for RL: 5
  - Test set: 10

- Language:
  - Synthetic: Using templates
  - Natural: Paraphrases using Amazon Mechanical Turk
RElational Task Adaptation for Imitation with Language (RETAIL)
Domains: Data Collection

<table>
<thead>
<tr>
<th>Template</th>
<th>Natural language paraphrase</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. go further away from the metallic table</td>
<td>Increase your distance from the metallic table</td>
</tr>
<tr>
<td>2. go closer to the foldable light</td>
<td>Move in the direction of the light that is foldable</td>
</tr>
<tr>
<td>3. go to the opposite side of the corner light</td>
<td>Move across from the corner light</td>
</tr>
<tr>
<td>4. move the large chair one unit farther from</td>
<td>Increment the distance of the big chair from the wide couch by one.</td>
</tr>
<tr>
<td>the wide couch</td>
<td></td>
</tr>
<tr>
<td>5. move corner table two units further left and metallic shelf one unit further backward</td>
<td>slide the corner table two units left and move the metal shelf a single unit back</td>
</tr>
<tr>
<td>6. move the large table to where the large sofa was moved, and vice versa</td>
<td>swap the place of the table with the sofa</td>
</tr>
</tbody>
</table>
RElational Task Adaptation for Imitation with Language (RETAIL)
Relational Reward Adaptation

Source demo —> Target reward

Language

Source policy —> Target policy

Relational Reward Adaptation

Relational Policy Adaptation
RElational Task Adaptation for Imitation with Language (RETAIL)

Relational Reward Adaptation: Approach

Reward:

\[ R(s, s') = \phi(s') - \phi(s) \]

\( g_{src} \) : Goal state for the source task

\( g_{tgt} \) : Goal state for the target task

\( l \) : Language describing the difference
RElational Task Adaptation for Imitation with Language (RETAIL)

Relational Reward Adaptation: Approach

Reward:

\[ R(s, s') = \phi(s') - \phi(s) \]

\( g_{src} \) : Goal state for the source task

\( g_{tgt} \) : Goal state for the target task

\( l \) : Language describing the difference

Goal prediction:

\[ g_{tgt} = \text{Adapt}(g_{src}, l) \]

Distance function:

\[ d(s, s') \]
RElational Task Adaptation for Imitation with Language (RETAIL)

Relational Reward Adaptation: Approach

Reward:
\[ R(s, s') = \phi(s') - \phi(s) \]

\( g_{src} \): Goal state for the source task
\( g_{tgt} \): Goal state for the target task
\( l \) : Language describing the difference

Goal prediction: \( g_{tgt} = \text{Adapt}(g_{src}, l) \)

Distance function: \( d(s, s') \)

\[ \implies \phi_{tgt}(s | g_{src}, l) = -d(s, \text{Adapt}(g_{src}, l)) \]
RElational Task Adaptation for Imitation with Language (RETAIL)
Relational Reward Adaptation: Goal Prediction

**Goal state encoder:**

- Entity i goal position in the source task: (x, y)
- Entity i attribute (e.g. “large”)
- Entity i noun (e.g. “table”)

Linear Layer → Concatenate Layer
RElational Task Adaptation for Imitation with Language (RETAIL)
Relational Reward Adaptation: Goal Prediction

Goal state encoder:

Entity i goal position in the source task: \((x, y)\)

Entity i attribute (e.g. “large”)

Entity i noun (e.g. “table”)

Linear Layer

Embedding Layer

Concat

Language encoder:

Language Description (e.g. “Move the large table one unit closer to the metallic shelf.”)

Language Encoder

Encoded Tokens
RElational Task Adaptation for Imitation with Language (RETAIL)
Relational Reward Adaptation: Goal Prediction

Goal state encoder:
- Entity i goal position in the source task: (x, y)
- Entity i attribute (e.g. “large”)
- Entity i noun (e.g. “table”)

Language encoder:
- Language Description (e.g. “Move the large table one unit closer to the metallic shelf.”)

Transformer Module:
- Predicted (x, y) coordinates of entities in the target task
- Encoded goal states of entities in the target task
- Encoded goal states of entities in the source task
- Encoded language tokens
RElational Task Adaptation for Imitation with Language (RETAIL)
Relational Reward Adaptation: Distance Function
RElational Task Adaptation for Imitation with Language (RETAIL)
Relational Reward Adaptation: Experiments
RElational Task Adaptation for Imitation with Language (RETAIL)
Relational Reward Adaptation: Experiments

Rearrangement

Navigation

No. of successful episodes

Ours           Oracle           Zero reward
RElational Task Adaptation for Imitation with Language (RETAIL)
Relational Reward Adaptation: Experiments

Rearrangement

Navigation
RElational Task Adaptation for Imitation with Language (RETAIL)
Relational Reward Adaptation: Experiments

Rearrangement

Navigation
RElational Task Adaptation for Imitation with Language (RETAIL)
Relational Reward Adaptation: Experiments

Rearrangement

<table>
<thead>
<tr>
<th>Method</th>
<th>No. of Successful Episodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>3000</td>
</tr>
<tr>
<td>Cricket</td>
<td>4000</td>
</tr>
<tr>
<td>Zero reward</td>
<td>4000</td>
</tr>
<tr>
<td>Truc goal, zero dist</td>
<td>4000</td>
</tr>
<tr>
<td>Part goal, true dist</td>
<td>4000</td>
</tr>
<tr>
<td>Synthetic language</td>
<td>4000</td>
</tr>
<tr>
<td>Non relational</td>
<td>4000</td>
</tr>
</tbody>
</table>

Navigation

<table>
<thead>
<tr>
<th>Method</th>
<th>No. of Successful Episodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>200</td>
</tr>
<tr>
<td>Cricket</td>
<td>300</td>
</tr>
<tr>
<td>Zero reward</td>
<td>300</td>
</tr>
<tr>
<td>Truc goal, zero dist</td>
<td>300</td>
</tr>
<tr>
<td>Part goal, true dist</td>
<td>300</td>
</tr>
<tr>
<td>Synthetic language</td>
<td>300</td>
</tr>
<tr>
<td>Non relational</td>
<td>300</td>
</tr>
</tbody>
</table>
RElational Task Adaptation for Imitation with Language (RETAIL)
Relational Policy Adaptation

Source demo

Source policy

Target policy

Language

Target reward

Relational Reward Adaptation

Relational Policy Adaptation
RElational Task Adaptation for Imitation with Language (RETAIL)

Relational Policy Adaptation: Approach

Step 1: Train a goal-conditioned policy

- Goal $g$
- State $s$
- MLP
- Action
RElational Task Adaptation for Imitation with Language (RETAIL)
Relational Policy Adaptation: Approach

**Step 1: Train a goal-conditioned policy**

- Goal $g$
- State $s$
- MLP
- Action

**Step 2: Generate data**

- Source goal $g_{src}$
- State $s$
- MLP
- Source action $a_{src}$

- Target goal $g_{tgt}$
- State $s$
- MLP
- Target action $a_{tgt}$
RElational Task Adaptation for Imitation with Language (RETAIL)
Relational Policy Adaptation: Approach

**Step 1: Train a goal-conditioned policy**

**Step 2: Generate data**

**Step 3: Train policy adaptation model using the generated data**

**State encoder:**
- Entity i position: \((x, y)\)
- Entity i attribute (e.g., "large")
- Entity i noun (e.g., "table")

**Transformer Module:**

**Language encoder:**
- Language Description (e.g., "Move the large table one unit closer to the metallic shelf.")

**Transformer**

**Encoded Tokens**

**Encoded state**

**Encoded language**
RElational Task Adaptation for Imitation with Language (RETAIL)
Relational Policy Adaptation: Experiments

Rearrangement

Success Rate (%)

- Without goal prediction
- With goal prediction

Natural Language
Synthetic Language
RElational Task Adaptation for Imitation with Language (RETAIL)
Relational Policy Adaptation: Experiments

Rearrangement

- Without goal prediction
- With goal prediction

Success Rate (%)
RElational Task Adaptation for Imitation with Language (RETAIL)
Relational Policy Adaptation: Experiments

Rearrangement

- Without goal prediction
- With goal prediction

Navigation

- Without goal prediction
- With goal prediction

Success Rate (%)
RElational Task Adaptation for Imitation with Language (RETAIL)
Combining Reward and Policy Adaptation

Source demo \rightarrow Language \rightarrow Target reward

Source policy \rightarrow Language \rightarrow Target policy

Relational Reward Adaptation

Relational Policy Adaptation
RElational Task Adaptation for Imitation with Language (RETAIL)
Combining Reward and Policy Adaptation
RElational Task Adaptation for Imitation with Language (RETAIL)
Combining Reward and Policy Adaptation
RElational Task Adaptation for Imitation with Language (RETAIL)
Combining Reward and Policy Adaptation
RElational Task Adaptation for Imitation with Language (RETAIL)
Combining Reward and Policy Adaptation

Knowledge Distillation:
- Use states from the demonstration data
- Losses:
  - Value network: Mean squared error
  - Policy network:
    - Mean squared error for continuous actions
    - Cross-entropy for discrete actions
RElational Task Adaptation for Imitation with Language (RETAIL)
Combining Reward and Policy Adaptation

Knowledge Distillation:
- Use states from the demonstration data
- Losses:
  - Value network: Mean squared error
  - Policy network:
    - Mean squared error for continuous actions
    - Cross-entropy for discrete actions

Finetuning using PPO:
- Networks initialized using knowledge distillation have low entropy
=> Continuous actions: Tune the standard deviation of the policy network
=> Discrete actions: Add an entropy loss during knowledge distillation
RElational Task Adaptation for Imitation with Language (RETAIL)
Combining Reward and Policy Adaptation

Rearrangement

Navigation
Talk Outline

Background

Core Contributions:

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>RL</td>
<td>Reward design</td>
</tr>
<tr>
<td>IL</td>
<td>Many demos needed</td>
</tr>
</tbody>
</table>

Future Directions
Future Work
Richer Domains

[Chen et al., 2019]

[Kolve et al., 2017]

[Lee et al., 2021]

[Mandlekar et al., 2018]
Future Work
Hierarchical Tasks

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

[Shridhar et al., 2020]
Future Work

Task Adaptation with Multiple Source Tasks
Future Work
Language-aided Imitation Learning

Humans use linguistic cues when giving demonstrations to other humans, e.g., “Turn off the heat when the water starts boiling”

Gridworld Cooking / Repairing:
- Language: What to do, Why
- Demonstrations: How

[Carroll et al., 2019]
Language can be used in a lot of ways:

- Communicating the task
- Providing feedback
- Guiding the agent to focus on the important aspects of the task
- Enabling the agent to ask clarification questions
Language can be used in a lot of ways:

- Communicating the task  
  Focus of this dissertation
- Providing feedback
- Guiding the agent to focus on the important aspects of the task
- Enabling the agent to ask clarification questions

Also related to these
## Conclusion

<table>
<thead>
<tr>
<th>Problem Settings</th>
<th>Rewards + Task Description</th>
<th>Source demo + Language describing the difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithms</td>
<td>LEARN PixL2R</td>
<td>LARVA RETAIL</td>
</tr>
<tr>
<td>Benchmarks</td>
<td>Montezuma’s Revenge</td>
<td>Organizer Relational Rearrangement Relational Navigation</td>
</tr>
</tbody>
</table>
## Conclusion

<table>
<thead>
<tr>
<th>Problem Settings</th>
<th>Rewards + Task Description</th>
<th>Source demo + Language describing the difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithms</td>
<td>LEARN, PixL2R</td>
<td>LARVA, RETAIL</td>
</tr>
<tr>
<td>Benchmarks</td>
<td>Montezuma’s Revenge, Robot Manipulation</td>
<td>Organizer, Relational Rearrangement, Relational Navigation</td>
</tr>
</tbody>
</table>
Acknowledgements

Raymond Mooney

Scott Niekum