Guiding Policies with Language via Meta-Learning

John D (JD) Co-Reyes, Abhishek Gupta, Suvansh Sanjeev, Nick Altieri, John DeNero, Pieter Abbeel, Sergey Levine
Ideal Robot
Learning new tasks quickly

- Want diverse range of skills
- Cost of supervision can be high
- Want to learn new things with as little supervision as possible
Meta-RL

Leverage prior experience to quickly learn new tasks

Meta-Training

Prior Experience

Fast Learning of New Tasks

Meta-Testing
Problem with reward design

Hard to design

\[ r = \text{game score} \]

Hard to provide

\[ r_{\text{shaped}}(s,a) = \|\text{hammer} - \text{palm}\|_2 \]
\[ + \|\text{hammer} - \text{nail}\|_2 \]
\[ + \|\text{nail} - \text{goal}\| \]

Hard to learn from

\[ r = 1 (\text{both blocks in box}) \]
More natural way to provide supervision

Human feedback
Human-in-the-loop supervision

Replace reward with human feedback

Deep TAMER

Preferences

Warnell et al

Christiano et al
Why current methods are insufficient?

Very few bits of information per intervention

Scalar Feedback

More bits of information per intervention

Language Feedback

RL Algorithm

Significant human effort

1 bit

Less human effort

230 bits
Language Corrections
Problem Setting

Agent provided with ambiguous/incomplete instruction

Quickly incorporate **language corrections** in the loop

Ambiguous Instruction (L): Move blue cylinder in between red circle and green triangle
Language Guided Policy Model

Model improves based on previous trajectories and corrections.

3 modules – corrections, policy and instruction modules
Algorithm Overview

Meta-Training

- Task 1 expert
- Task 2 expert
- ...
- Task N expert

Labeler

Ground corrections

Meta-Testing

Correction

Trajectory

i=1...n
Meta-Training

Data Collection (Task 1)

\[ C_0 \quad \pi \quad \tau_1 \quad \text{Expert } \pi^* \quad (s, a^*, \tau_0, C_0, L_1) \]

\[ C_1 \quad \pi \quad \tau_2 \quad \text{Expert } \pi^* \quad (s, a^*, \tau_1, C_1, L_1) \]

\[ C_2 \quad \pi \quad \tau_3 \quad \text{Expert } \pi^* \quad (s, a^*, \tau_2, C_2, L_1) \]

Data Buffer
Meta-Training

Data Collection (Task 1)

Data Collection (Task 2)

Data Collection (Task N)

Data Buffer

Supervised Learning

\[ \pi \]

\[ \text{Data Collection (Task i)} \]

\[ \text{Expert}^{\pi*}(s, a^*, \tau_0, C_0, L_1) \]

\[ \text{Expert}^{\pi*}(s, a^*, \tau_1, C_1, L_1) \]

\[ \text{Expert}^{\pi*}(s, a^*, \tau_2, C_2, L_1) \]
Experimental Setup

Multi-room domain

Instruction: Move green triangle to yellow goal.

Instruction: Move red square to yellow goal.
Experimental Setup

Block pushing domain

Instruction: Move red block above magenta block.

Instruction: Move cyan block left of blue block.
Quick Learning of New Tasks

Instruction: Move blue triangle to green goal.

Correction 1: Enter the blue room.

Correction 2: Enter the red room.

Correction 3: Exit the blue room.

Correction 4: Pick up the blue triangle.

Solved
Quick Learning of New Tasks

Instruction: Move cyan block below magenta block.

Correction 1: Touch cyan block.
Correction 2: Move closer to magenta block.
Correction 3: Move a lot up.
Correction 4: Move a little up.

Solved
### Quantitative Evaluation

**Success Rates on New Tasks**

<table>
<thead>
<tr>
<th>Env</th>
<th>Instruction</th>
<th>Full Info</th>
<th>MIVOA (Instr.)</th>
<th>MIVOA (Full Info)</th>
<th>$c_0$</th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
<th>$c_4$</th>
<th>$c_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-room</td>
<td>0.075</td>
<td>0.73</td>
<td>0.067</td>
<td>0.63</td>
<td>0.066</td>
<td>0.46</td>
<td>0.65</td>
<td>0.73</td>
<td>0.77</td>
<td>0.82</td>
</tr>
<tr>
<td>Obj Relocation</td>
<td>0.64</td>
<td>0.96</td>
<td>0.65</td>
<td>-</td>
<td>0.65</td>
<td>0.80</td>
<td>0.84</td>
<td>0.85</td>
<td>0.88</td>
<td>0.90</td>
</tr>
</tbody>
</table>

**Much quicker learning than using rewards**

- **Multi-Room Object Manipulation**
- **Robotic Object Relocation**

---

RL – PPO with reward used to train expert

GPL – Ours
Summary

• Avoid demos/reward functions using human-in-the-loop

• Language provides more information per intervention

• Ground language in multi-task setup; learn new tasks quickly with corrections
Thank you

Abhishek Gupta  Suvansh Sanjeev  Nick Altieri

John DeNero  Pieter Abbeel  Sergey Levine