Pre-Training for Robots: How Offline RL Enables Learning New Tasks from a Handful of Trials

Aviral Kumar*, Anikait Singh*, Frederik Ebert*, Yanlai Yang, Chelsea Finn, Sergey Levine
Our Vision: Incorporate Large Robotic Datasets
How To Learn From Large Robot Datasets

Pre-training on broad data (e.g., representation learning)

Adaptation on limited, task-specific data

Imitation learning

Ebert et al. 2021
Young et al. 2021
and many more….

Pre-trained representation

Nair et al. 2022
Shafiullah et al. 2022
and many more….

Can we instead use offline RL for both pre-training & fine-tuning?

Why?
Pre-Training for Robots Using Offline RL

1. Pre-train via offline RL (CQL)

Batch-mixing pre-training and target data

2. Continue fine-tuning with offline RL
Main Innovation: Architecture

Modified ResNet with group normalization

Output ResNet feature maps

Learned position embeddings

Task ID

Action vector duplicated

Fully-connected Layers

Task ID

Action vector duplicated

$Q(\phi(s,a))$
Task: Solving A Task in A New Domain

1. Pre-Train on Bridge Data, 12 doors 800 demonstrations

2. Fine-Tune on Target Domain Data:
   1 door, 10 demonstrations
Results: Solving A Task in A New Domain

Method: Imitation (Best prior method)

Task: Open Door

Method: PTR (Ours)

Task: Open Door
Task: Solving New Tasks in New Domains

10 target demonstrations
Results: Solving New Tasks in New Domains

Best Prior method

- Task: Put sweet potato on plate [X]
- Task: Put knife in pot [X]
- Task: Put sushi in pot [X]

PTR (Ours)

- Task: Put sweet potato on plate [✓]
- Task: Put knife in pot [✓]
- Task: Put sushi in pot [✓]
Some Quantitative Results

<table>
<thead>
<tr>
<th>Task</th>
<th>PTR (Ours)</th>
<th>BC (fine.)</th>
<th>Autoreg. BC</th>
<th>BeT</th>
<th>COG</th>
<th>BC</th>
<th>CQL</th>
<th>BC</th>
<th>R3M</th>
<th>MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Take croissant from metal bowl</td>
<td>7/10</td>
<td>3/10</td>
<td>5/10</td>
<td>1/10</td>
<td>4/10</td>
<td>4/10</td>
<td>0/10</td>
<td>1/10</td>
<td>1/10</td>
<td>3/10</td>
</tr>
<tr>
<td>Put sweet potato on plate</td>
<td>7/20</td>
<td>1/20</td>
<td>1/20</td>
<td>0/20</td>
<td>0/20</td>
<td>0/20</td>
<td>0/20</td>
<td>0/20</td>
<td>0/20</td>
<td>1/20</td>
</tr>
<tr>
<td>Place knife in pot</td>
<td>4/10</td>
<td>2/10</td>
<td>2/10</td>
<td>0/10</td>
<td>1/10</td>
<td>3/10</td>
<td>3/10</td>
<td>0/10</td>
<td>0/20</td>
<td>0/10</td>
</tr>
<tr>
<td>Put cucumber in pot</td>
<td>5/10</td>
<td>0/10</td>
<td>1/10</td>
<td>0/10</td>
<td>2/10</td>
<td>1/10</td>
<td>0/10</td>
<td>0/10</td>
<td>0/10</td>
<td>0/10</td>
</tr>
</tbody>
</table>

Imitation (using transformers, auto-regressive)

Self-supervised pre-training from internet data / bridge data

Better fine-tuning!

Representation learning

**Takeaway:** Offline RL learn useful representations + better fine-tuning
Scaling And Analysis

Why would RL enable better performance...
...when the data is collected via human teleoperation?

**Spoiler:** Value-functions can learn which decisions are more critical than others!

The larger the network, the better!
Takeaways and Future Work

- Offline RL can be good for both representation learning and control, even with human demonstration data

Future Directions:
- Goal specification: language? goals? reward learning?
- Multi-modal data: videos?