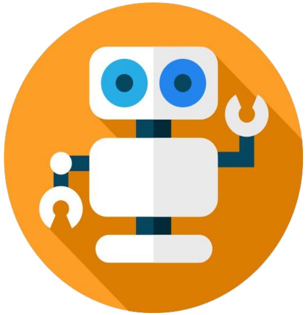
Decision Transformer for Robot Imitation Learning

Presenters: Alex Chandler, Jake Grigsby, Omeed Tehrani

11-29-2022

Motivation

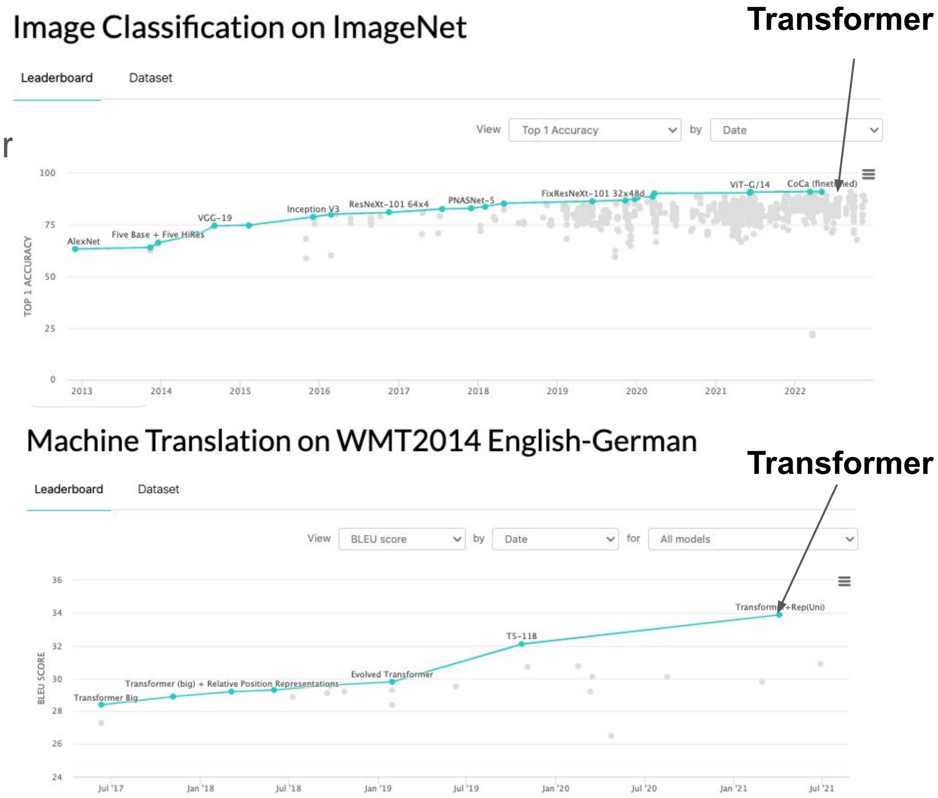


Real world imitation learning has questionable results on datasets with varying quality.

We seek to quantify the benefit of return conditioned imitation learning on mixed quality data by leveraging robomimic.

Additional Motivation

- Recent gains in performance by the Transformer architecture in NLP and Computer Vision
  - Larger models require fewer samples to reach comparable performance
  - Stable training in large language models
- Simplicity of converting RL to a sequence modeling problem
  - No need to estimate a good value function or rely on policy gradient methods



Prior Work: Sequence Modeling in RL

Attention Is All You Need

- Original Transformer architecture

Decision Transformer: Reinforcement Learning via Sequence Modeling

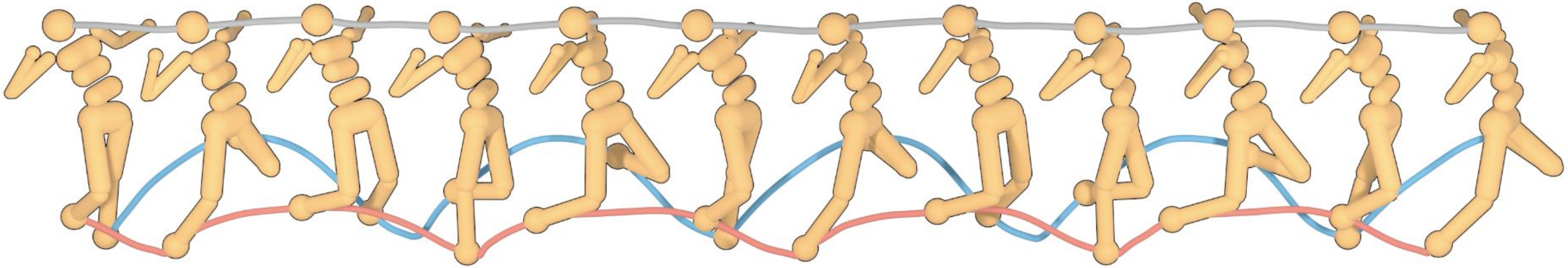
- Basis of our project

Offline Reinforcement Learning as One Big Sequence Modeling Problem

- Concurrent work from Berkeley

What Matters in Learning from Offline Human Demonstrations for Robot Manipulation

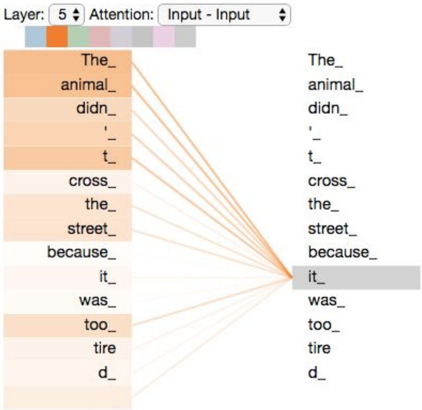
- robomimic dataset

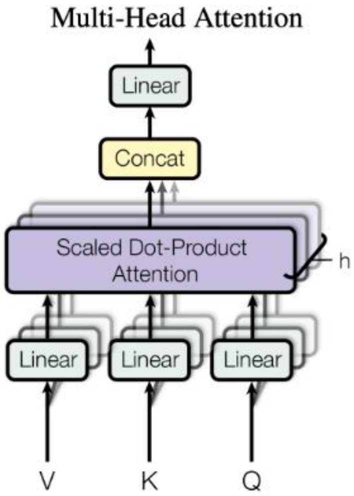


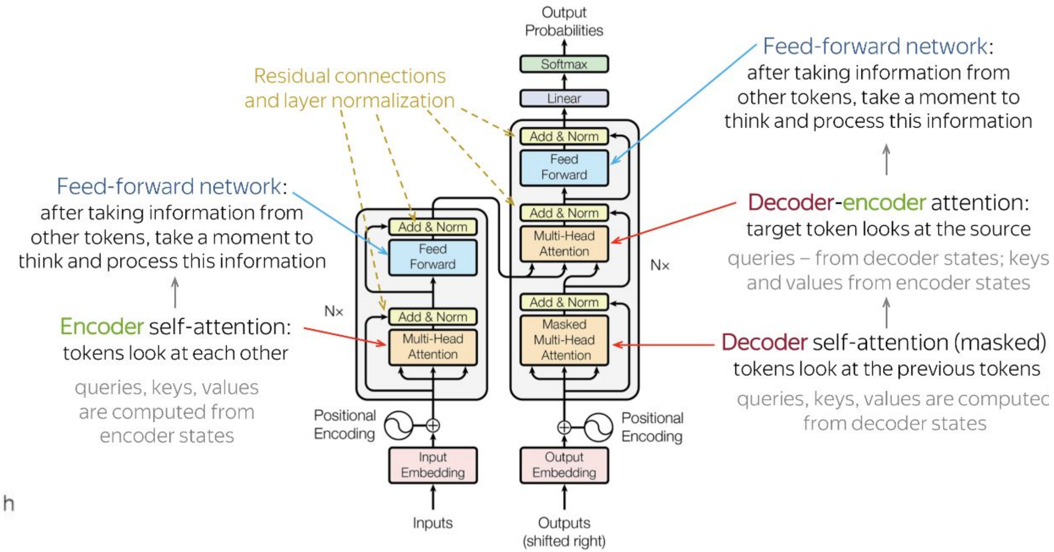
Transformers

Let's recap, what exactly is a transformer?

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$







Original Decision Transformer

- Decision Transformers abstract reinforcement learning as a sequence modeling problem.

- Offline reinforcement learning.

- We input state, actions, and returns-to-go into a causal transformer to get our desired actions.

What is robomimic?

robomimic
a framework for robot learning from demonstration

Born from a paper known as *What Matters in Learning from Offline Human Demonstrations for Robot Manipulation….*

- This paper studies challenges in offline reinforcement learning from human datasets → lessons to guide future work → and release of all datasets and code to facilitate future work.

Study design is a large evaluation of offline learning from human datasets:

- 8 tasks: lift, can, square, transport (for coordination), tool hang.
- 3 types of data sets: machine generate data, proficient human data, multi-human datasets.
- 6 offline learning algorithms: BC, BC-RNN, HBC, BCQ, CQL, IRIS
- 2 observation spaces: low dimensional agents with ground truth, image agents that receive camera observations

Dataset Types

robomimic
a framework for robot learning from demonstration

- **Machine-Generated (MG)**
  - Mixture of suboptimal data from state-of-the-art RL agents

- **Proficient-Human (PH) and Multi-Human (MH)**
  - 500 total, with 200 proficient human and 300 multi-human.
  - Demonstrations from teleoperators of varying proficiency

- **Our setting: ALL data**
  - More challenging combination of MG, MH, and PH
  - Weighted towards lower-quality MG data

Dataset Tasks

robomimic
a framework for robot learning from demonstration

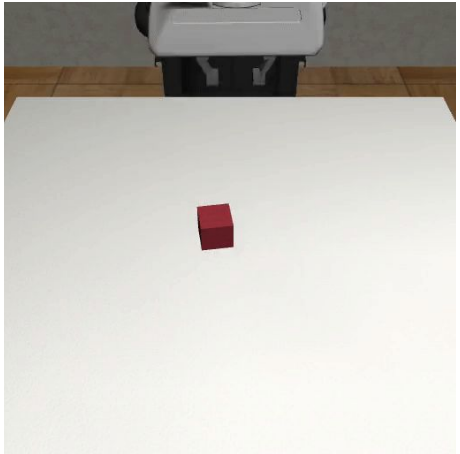
We mainly focus on two tasks:

1. **Lift**: lift the cube

2. **Can**: pick up the can and place it in proper spot

Why? These tasks have large machine-generated (lower-quality) datasets

Lift



lift the cube

Can

pickup and place the can

Semi-Sparse Reward Function

1

Found that datasets in robomimic use sparse rewards

2

Attempt at dense rewards

3

Semi-Sparse Reward Function

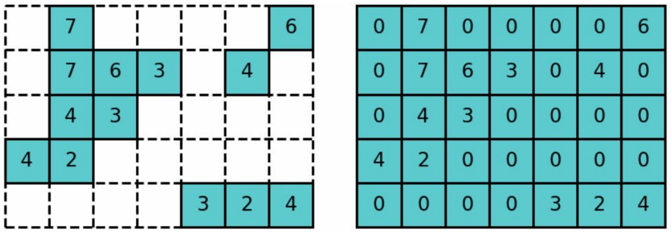
The Semi Sparse Reward Function:

max(500 - success time, 0)

4

Training data altered

Sparse vs. Dense Rewards



Sparse reward:

1. In reinforcement learning, sparse is typically correspondent to a binary success. It is given to an agent when the task is successfully complete, which can be a rare occurrence.
2. It is typically given for long-term goals and complex tasks.

Dense Rewards:

1. Type of reward that has a lot of specificity and precision and provides feedback to the agent
2. Difficult to tune and implement in the real world
3. In our case, we use the default dense reward from robomimic, which includes metrics like **object distance from the gripper**.

Semi-Sparse Reward Function (additional info)

Found that datasets in robomimic use sparse rewards.

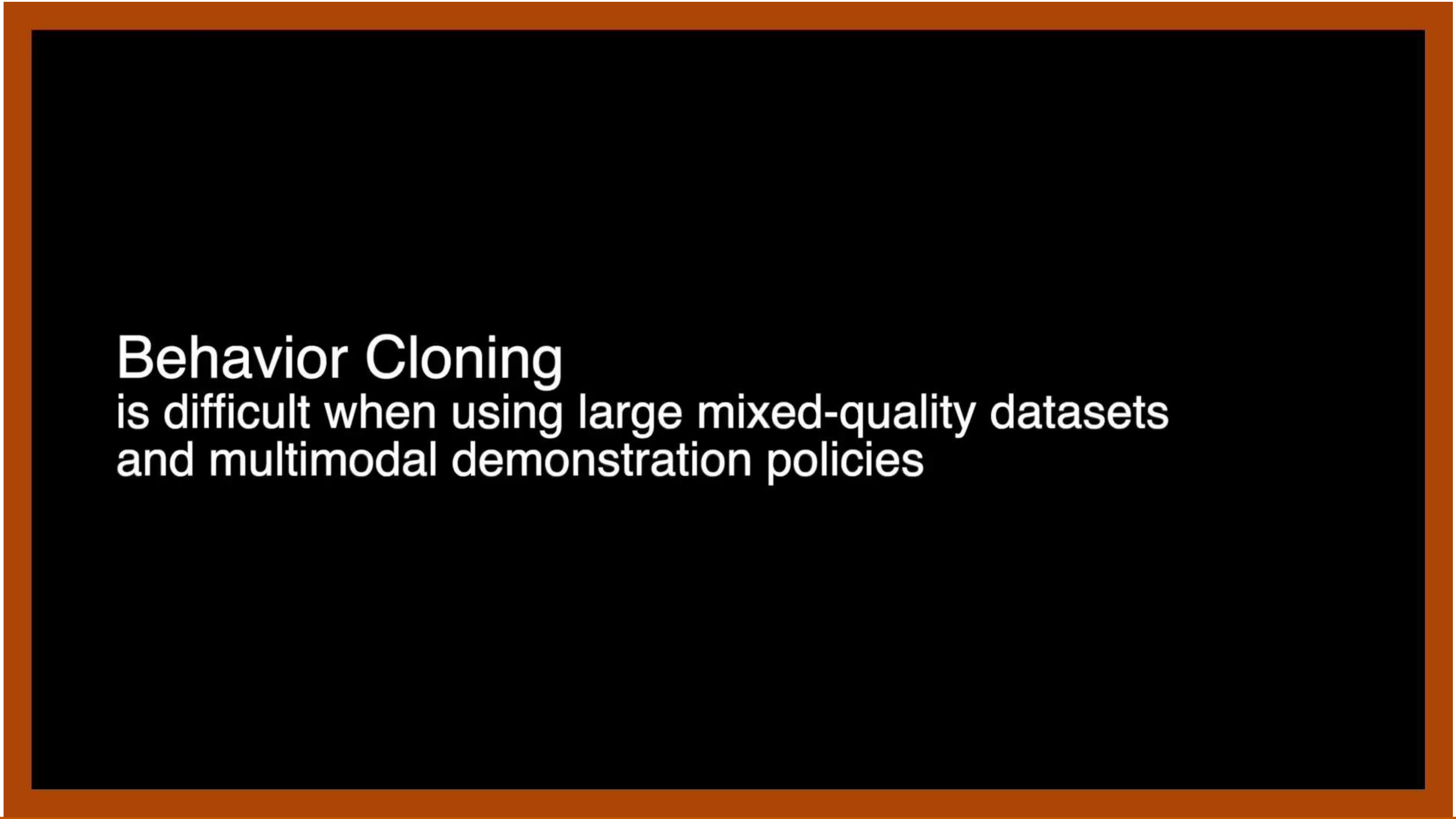
1. This initially would require us to download every single robomimic dataset and remake it with dense rewards.

Our Architecture

Experiments

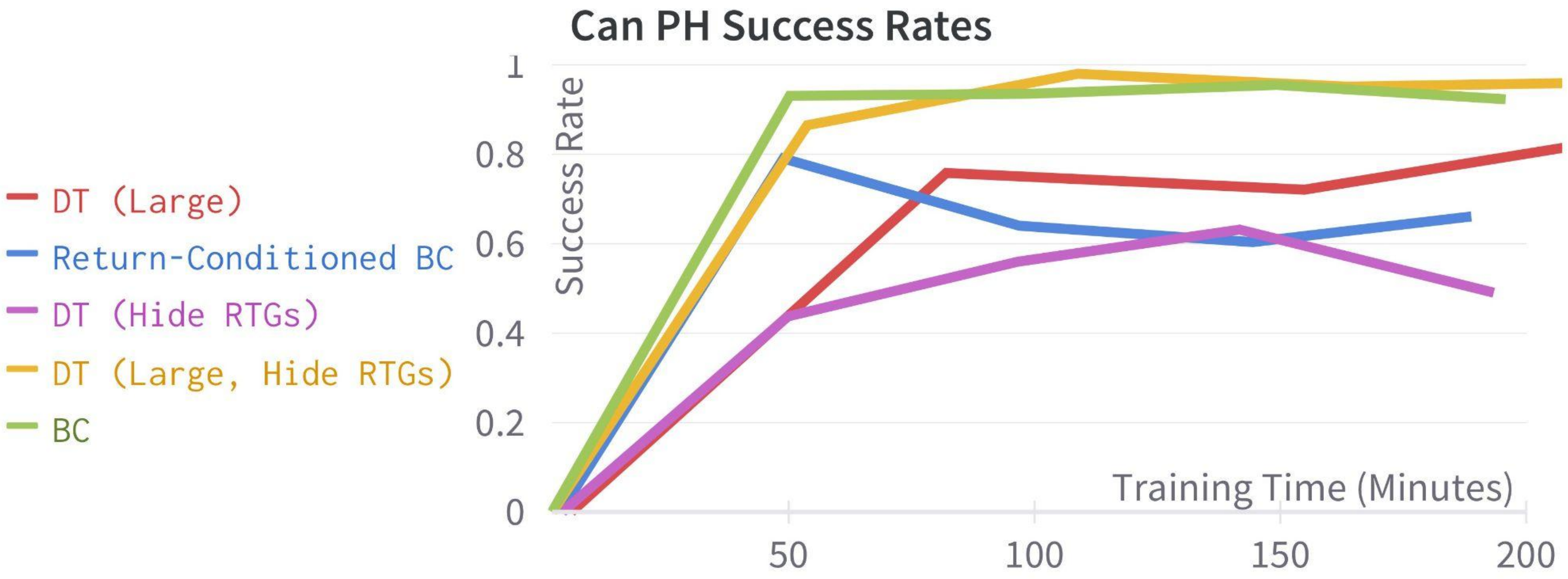
- Naive BC
- BC + Action Inp
- DT-1 (PH Only)
- DT-3 (Context length of 3)
- DT-10 (Context length of 10)
- DT-20 (Context length of 20)

Demo


Behavior Cloning
is difficult when using large mixed-quality datasets
and multimodal demonstration policies

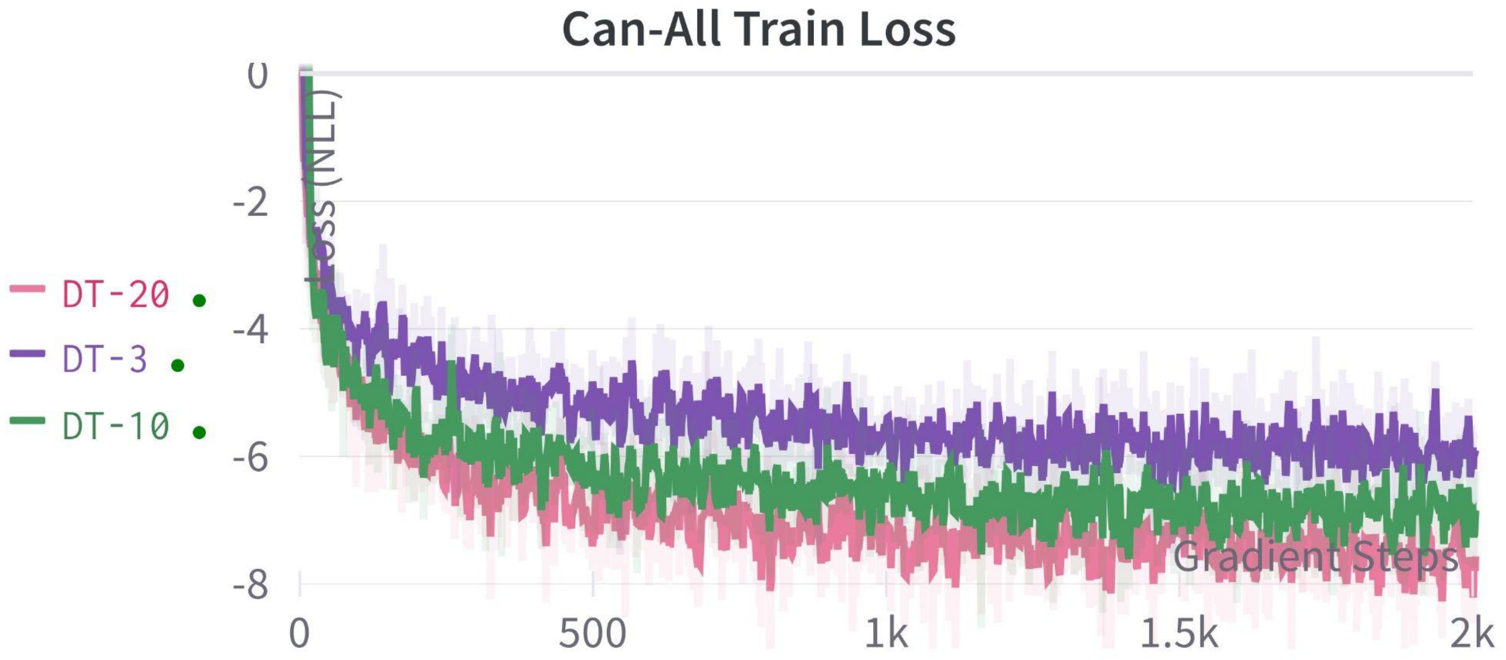
Results

Return and Past-Action conditioning can make robomimic tasks more difficult



Results

Longer sequence modeling improves action prediction and eases problems caused by multi-modal demonstrations

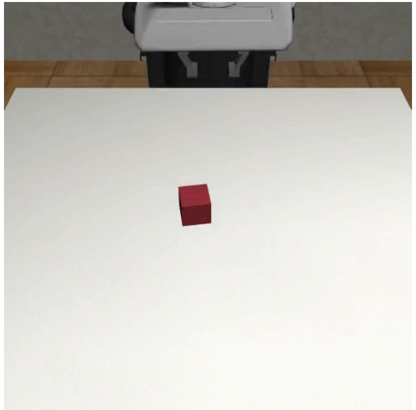


Results

Task: Lift (All Data)

	Naive BC	BC + Action Inp.	DT-1 (PH Only)	DT-1	DT-3	DT-10	DT-20
Success Rate (%)	35	20	*100*	85	92	92	**94**
Return	189	113	*463*	397	428	**433**	421

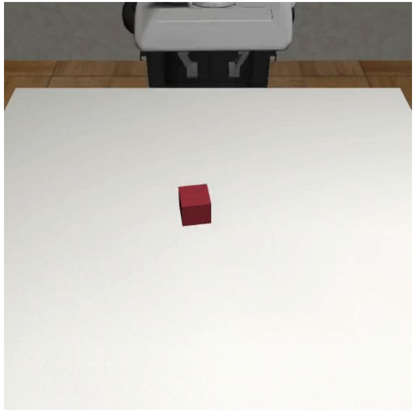
Behavior Cloning large, mixed-quality data leads to surprisingly poor performance



Results

Task: Lift (All Data)

	Naive BC	BC + Action Inp.	DT-1 (PH Only)	DT-1	DT-3	DT-10	DT-20
Success Rate (%)	35	20	*100*	85	92	92	**94**
Return	189	113	*463*	397	428	**433**	421

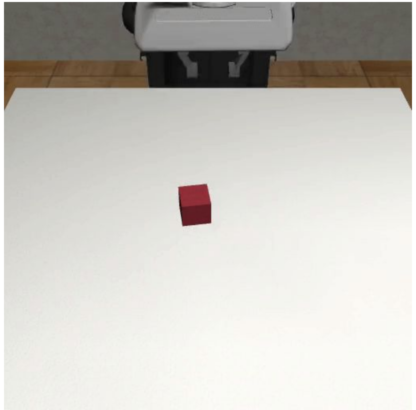


Removing the low-quality data allows for expert performance, as in original robomimic

Results

Task: Lift (All Data)

	Naive BC	BC + Action Inp.	DT-1 (PH Only)	DT-1	DT-3	DT-10	DT-20
Success Rate (%)	35	20	*100*	85	92	92	**94**
Return	189	113	*463*	397	428	**433**	421

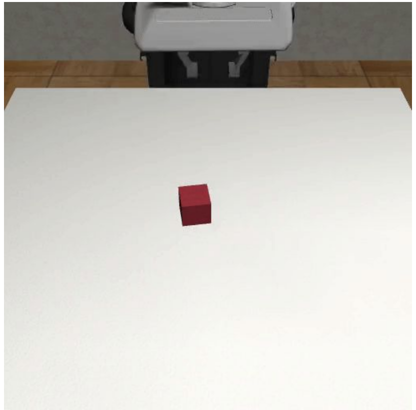


Decision Transformer can (mostly) filter the good demonstrations from the machine-generated noise

Results

Task: Lift (All Data)

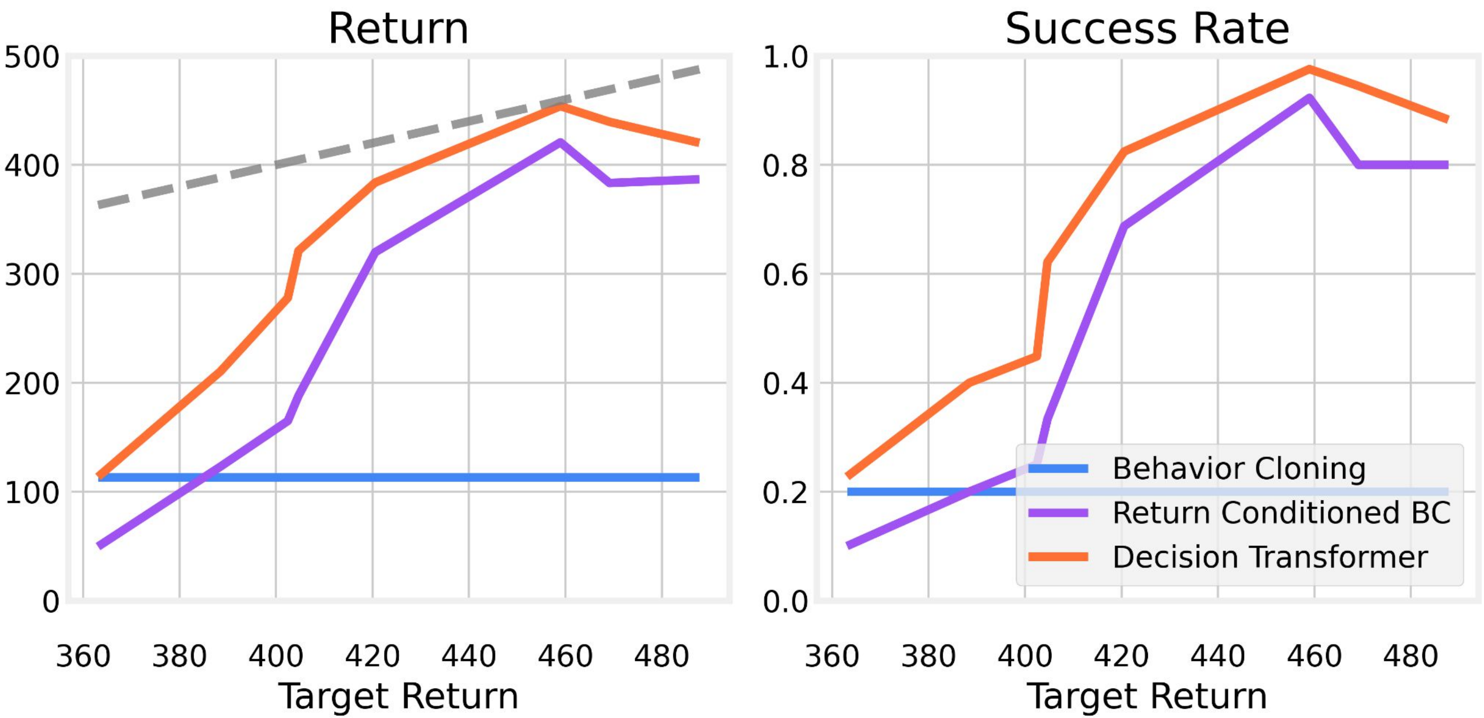
	DT-3	DT-10	DT-20	DT-3 (Gaussian)	DT-10 (Gaussian)	DT-20 (Gaussian)
Success Rate (%)	92	92	**94**	85	86	81
Return	428	**433**	421	403	406	386



The GMM policy is much better at modeling multi-modal policies than the standard Gaussian policy used by most RL agents

Results

Decision Transformer lets us model the whole range of returns, not just the expert



Results

Task: Can (All Data)

	BC (PH Only)	Naive BC	BC + Action Inp.	DT-3	DT-10	DT-20
Success Rate (%)	*99*	14	15	**81**	76	63
Return	*396*	72	74	**362**	337	286

Action and RTG input sequence makes this task significantly more difficult. But DT is much better than naive BC

See our poster for more

Results

Task: Can (All Data)

	DT-3 (Large)	DT-10 (Large)	DT-20 (Large)	DT-3 (Small)	DT-10 (Small)	DT-20 (Small)	DT-3 (Large, Gaussian)
Success Rate (%)	**81**	76	63	65	57	61	66
Return	**362**	337	286	292	262	278	204

Smaller Transformer sizes decrease performance in the Can task

Results

Task: Can (All Data)

	DT-3 (Large)	DT-10 (Large)	DT-20 (Large)	DT-3 (Small)	DT-10 (Small)	DT-20 (Small)	DT-3 (Large, Gaussian)
Success Rate (%)	**81**	76	63	65	57	61	66
Return	**362**	337	286	292	262	278	204

Standard Gaussian policies are less capable of modeling multi-modal action distributions than our Gaussian Mixture Model default

Critiques and Limitations

1. **Extremely long training time** → the datasets become extremely large when we combine all 3 data types, so more gradient updates are required to get good performance. Approximate 24 hours of training time.
2. **Dense Rewards in Robomimic** → Robomimic was not designed for dense rewards. We believe that altering the reward that is returned to match our goal would likely lead to very good results.
3. **Reward Function** → Creating a better reward function would likely yield some great results.
4. **Data Quality** → Lack of mixed-quality data for other tasks

Code Structure

Main files:

- `agent.py`
  - Main Class: Agent
- `transformer.py`
  - Contains actual transformer implementation
  - Key Classes
    - TransformerEncoder
    - TransformerLayer
- `learn.py`
  - Includes command line arguments for experimentation with ArgumentParser
  - Main Class: Experiment

Citations

- Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).
- Chen, Lili, et al. "Decision transformer: Reinforcement learning via sequence modeling." Advances in neural information processing systems 34 (2021): 15084-15097.
- Janner, Michael, Qiyang Li, and Sergey Levine. "Offline reinforcement learning as one big sequence modeling problem." Advances in neural information processing systems 34 (2021): 1273-1286.
- Mandlekar, Ajay, et al. "What matters in learning from offline human demonstrations for robot manipulation." arXiv preprint arXiv:2108.03298 (2021).