Autoregressive Models for Offline Policy Evaluation + Optimization

Michael R. Zhang, Tom Le Paine, Ofir Nachum, Cosmin Paduraru, George Tucker, Ziyu Wang, Mohammad Norouzi

Published at ICLR 2021

Motivation

- Model-based RL has been shown to improve sample efficiency on various control tasks
- Goal is to improve dynamics and rewards models by making predictions in autoregressive way

Deep reinforcement learning in a handful of trials using probabilistic dynamics models (Chua et al. 2018)
When to Trust Your Model: Model-Based Policy Optimization (Janner et al. 2019)
Deep Dynamics Models for Learning Dexterous Manipulation (Nagabandi et al 2019)

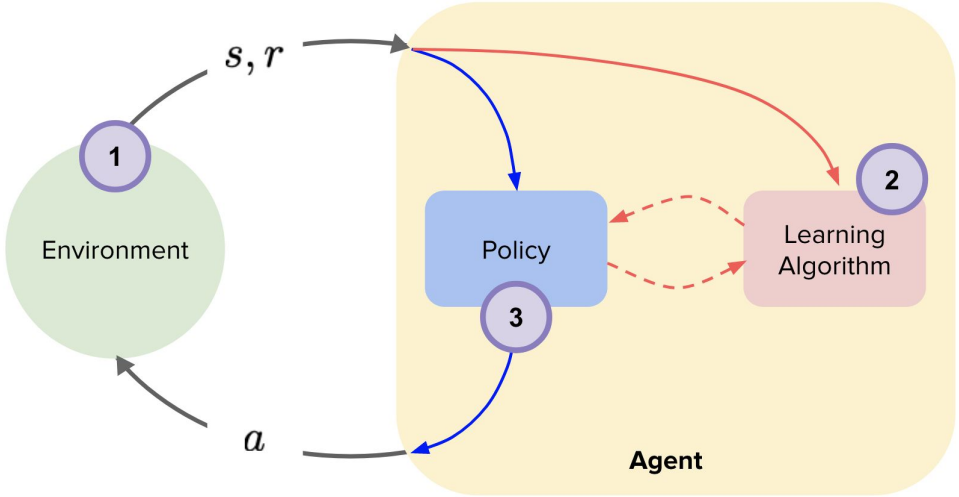
Motivation

Where does the model fit into the picture?

1. Simulating the environment

2. Assisting the learning algorithm

3. Strengthening the policy

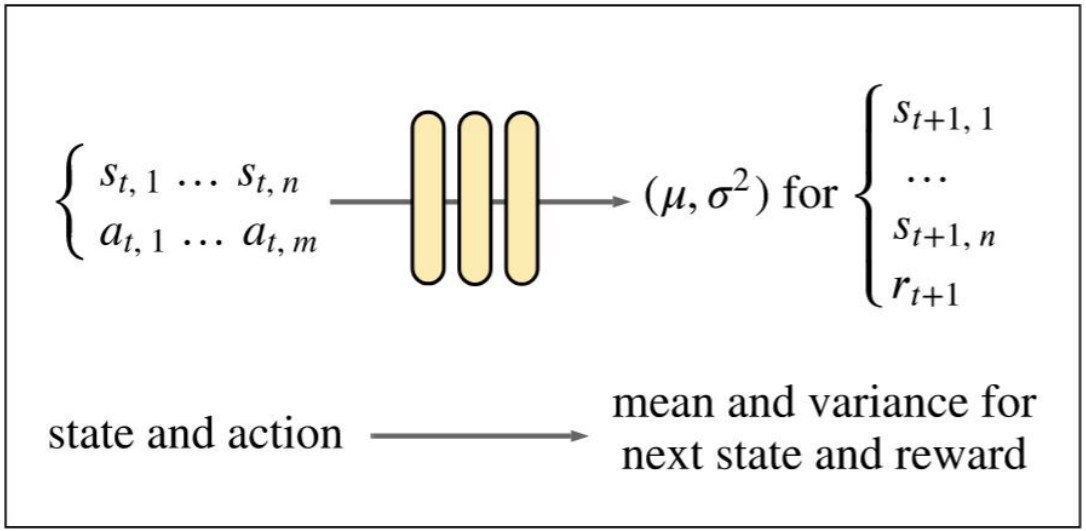


From “Tutorial on Model-Based Methods in Reinforcement Learning” (Mordatch and Hamrick)
see also: “Objective Mismatch in Model-based Reinforcement Learning” (Lambert et al.)

Tackle RL using generative models of the dynamics

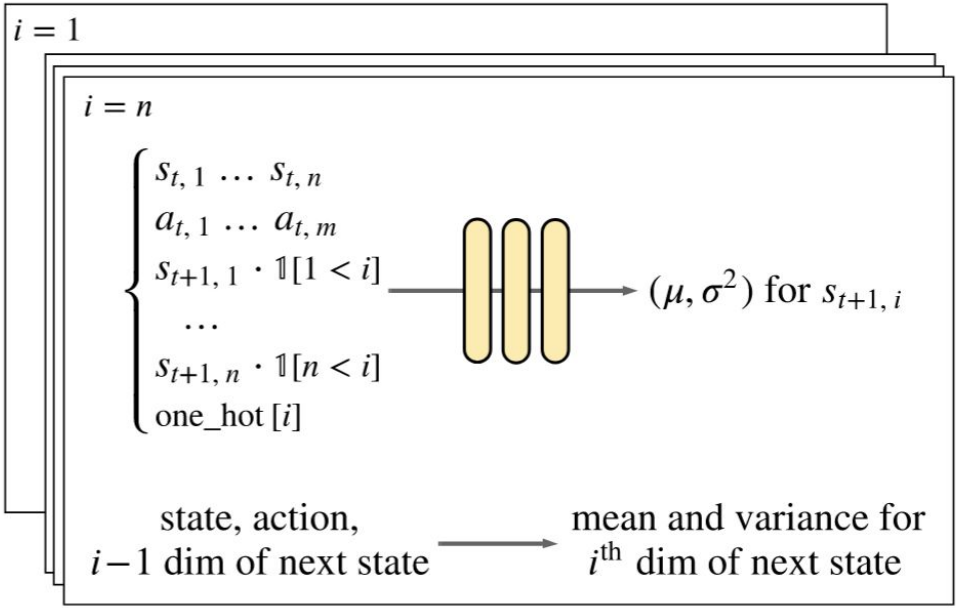
- Given a dataset of **(s, a, r', s')** tuples, *a.k.a. replay buffer*,
  train a conditional generative model of **(s, a) → (r', s')**,
  i.e., **p(r', s'| s, a)**,
  and use it for *policy evaluation* and *policy optimization*.
- Dynamics: (state, action) → (reward, next state)
  (s, a) → (r', s')

Feedforward dynamics model

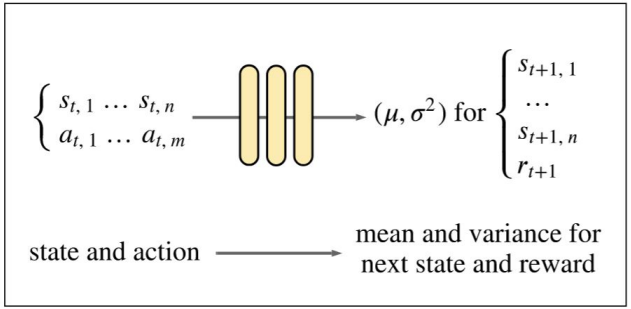


Standard Feedforward Dynamics Models

Autoregressive dynamics model



Proposed Autoregressive Dynamics Model

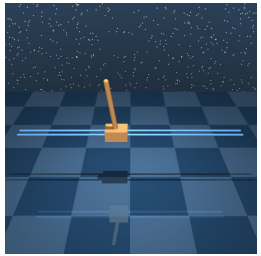


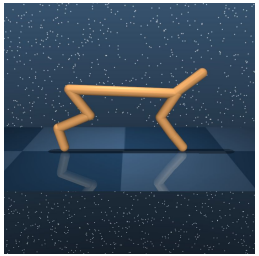
Standard Feedforward Dynamics Models

Continuous control tasks

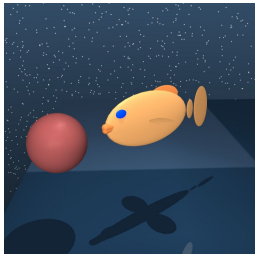
Summary of the offline datasets used. Dataset size indicates the number of (s, a, r', s') tuples.

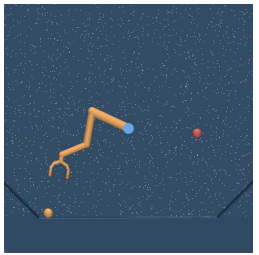
	cartpole swingup	cheetah run	finger turn hard	fish swim	humanoid run	walker stand	walker walk	manipulator insert ball	manipulator insert peg
State dim.	5	17	12	24	67	24	24	44	44
Action dim.	1	6	2	5	21	6	6	5	5
Dataset size	40K	300K	500K	200K	3M	200K	200K	1.5M	1.5M


Cartpole swingup


Cheetah run


Walker walk/ stand


Fish swim


Manipulator insert ball


Humanoid run

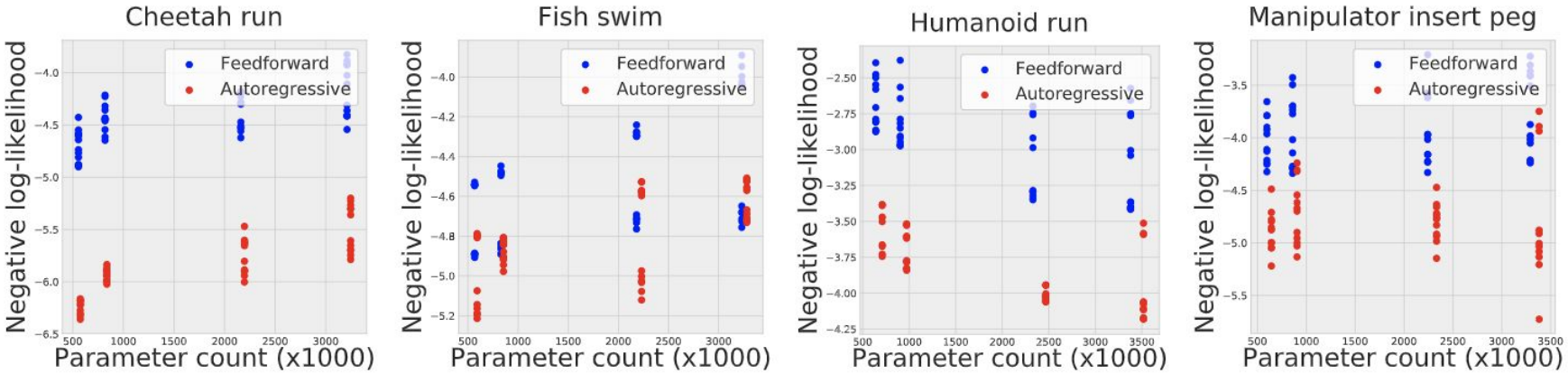
RL Unplugged: A Suite of Benchmarks for Offline Reinforcement Learning (Gulchere et al.)
Benchmarks for Deep Off-Policy Evaluation (Fu et al.)

Autoregressive models have better inductive biases

Negative log-likelihood on heldout validation sets for different tasks (lower is better).

	Dynamics model architecture	cartpole swingup	cheetah run	finger turn hard	fish swim	humanoid run	walker stand	walker walk	manipulator insert ball	manipulator insert peg
Top-1	Feedforward	-6.81	-4.90	-5.58	-4.91	-3.42	-4.52	-3.84	-4.74	-4.34
	Autoregressive	-7.21	-6.36	-6.14	-5.21	-4.18	-4.73	-4.17	-5.62	-5.73
Top-5	Feedforward	-6.75	-4.85	-5.50	-4.90	-3.40	-4.49	-3.81	-4.64	-4.31
	Autoregressive	-7.14	-6.32	-5.94	-5.18	-4.15	-4.71	-4.15	-5.58	-5.29

Autoregressive models have better inductive biases



Validation negative log-likelihood *vs.* parameter count for autogressive and feedforward models. Autoregressive models often have a lower validation NLL irrespective of parameter count.

Model-based Off-policy Evaluation

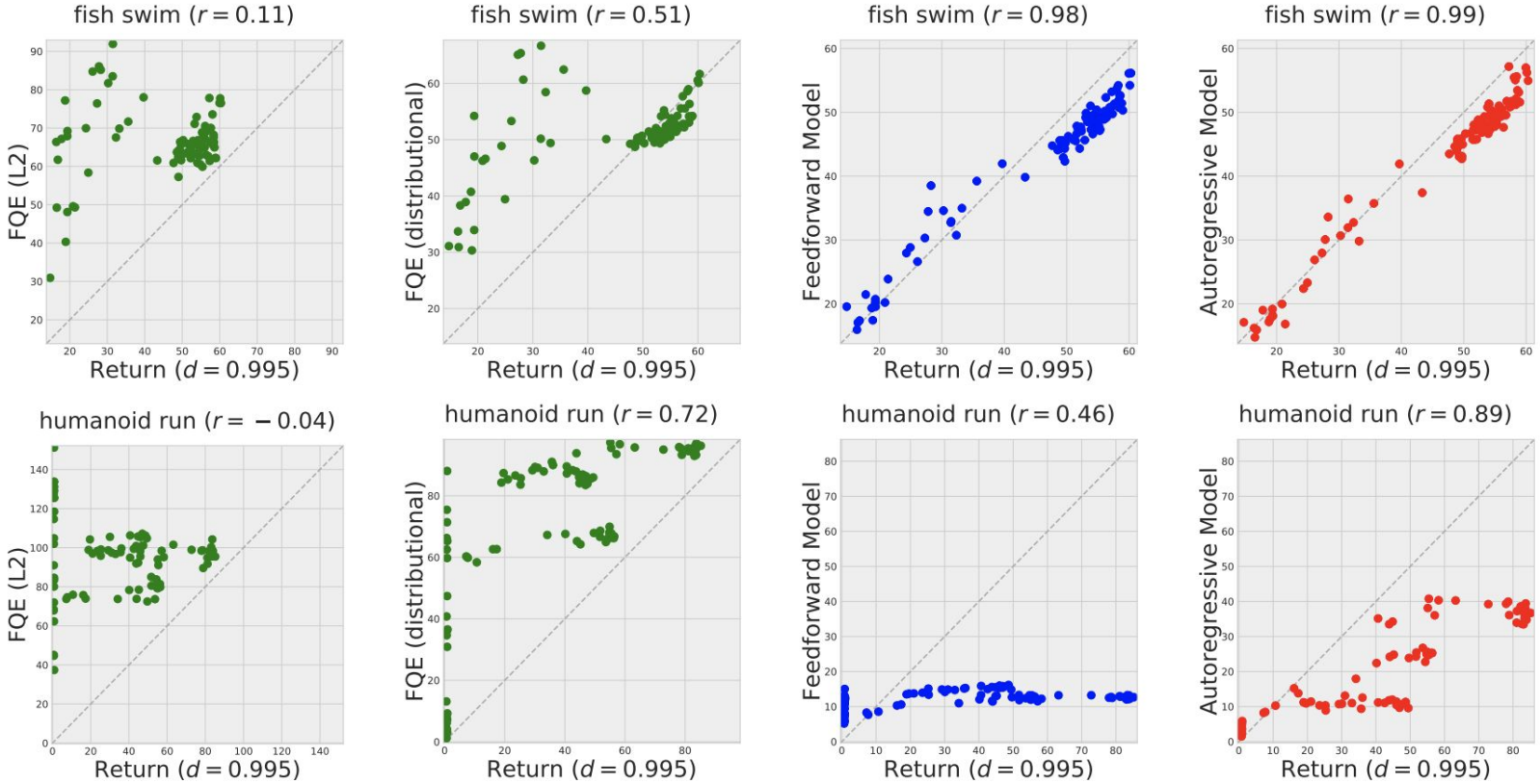
Train a generative model of dynamics, `p(r', s'| s, a)`, once, and use it for evaluating as many policies as you want.

$$\texttt{s} \xrightarrow[\pi]{} \texttt{a} \xrightarrow[p]{} \texttt{r', s'} \xrightarrow[\pi]{} \ldots$$

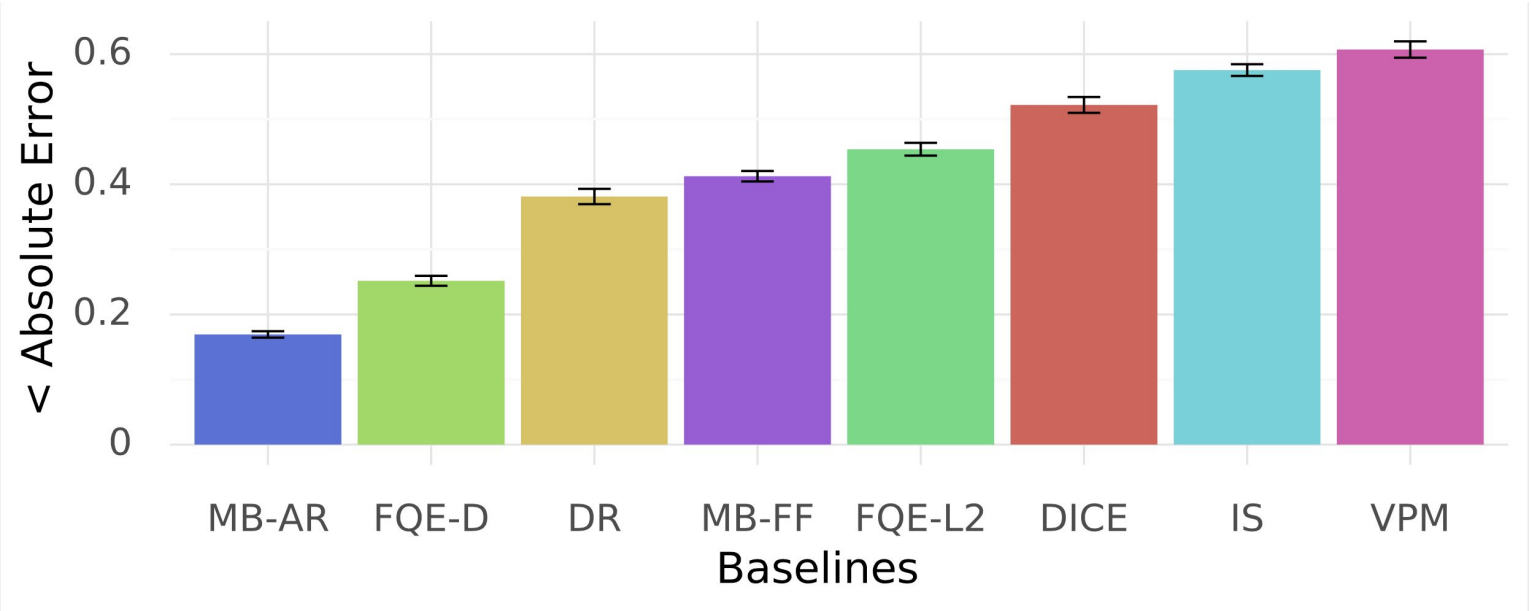
Algorithm 1 Model-based OPE

Require: Number of rollouts n, discount factor γ, horizon length H, policy π, dynamics model p, set of initial states S_0
for $i = 1, 2, \ldots n$ **do**
  $R_i \leftarrow 0$
  sample initial state $s_0 \sim \mathcal{S}_0$
  for $t = 0, 1, 2, \ldots, H-1$ **do**
    sample from policy: $a_t \sim \pi(\cdot \mid s_t)$
    sample from the dynamics model:
    $$s_{t+1}, r_{t+1} \sim p(\cdot, \cdot \mid s_t, a_t)$$
    $R_i \leftarrow R_i + \gamma^t r_{t+1}$
  end for
end for
return $\frac{1}{n} \sum_{i=1}^{n} R_i$

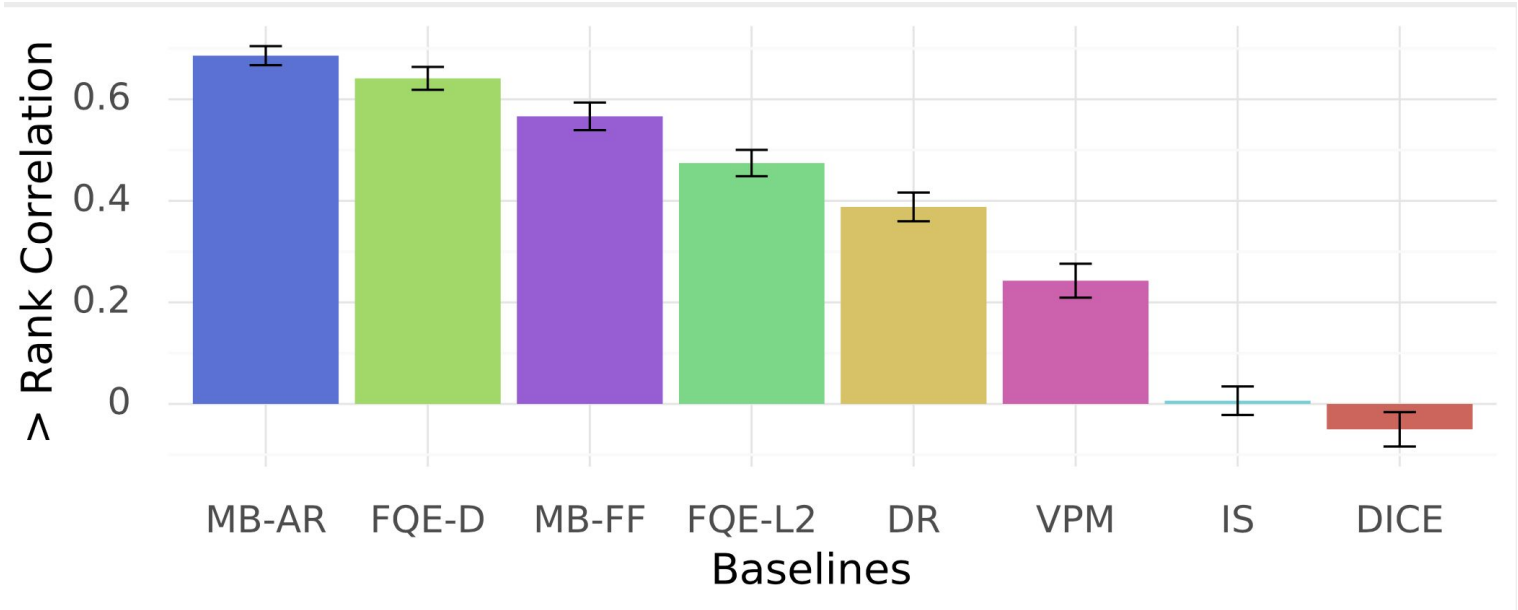
OPE results


fish swim ($r = 0.11$)
FQE (L2)
Return ($d = 0.995$)
fish swim ($r = 0.51$)
FQE (distributional)
Return ($d = 0.995$)
fish swim ($r = 0.98$)
Feedforward Model
Return ($d = 0.995$)
fish swim ($r = 0.99$)
Autoregressive Model
Return ($d = 0.995$)
humanoid run ($r = -0.04$)
FQE (L2)
Return ($d = 0.995$)
humanoid run ($r = 0.72$)
FQE (distributional)
Return ($d = 0.995$)
humanoid run ($r = 0.46$)
Feedforward Model
Return ($d = 0.995$)
humanoid run ($r = 0.89$)
Autoregressive Model
Return ($d = 0.995$)

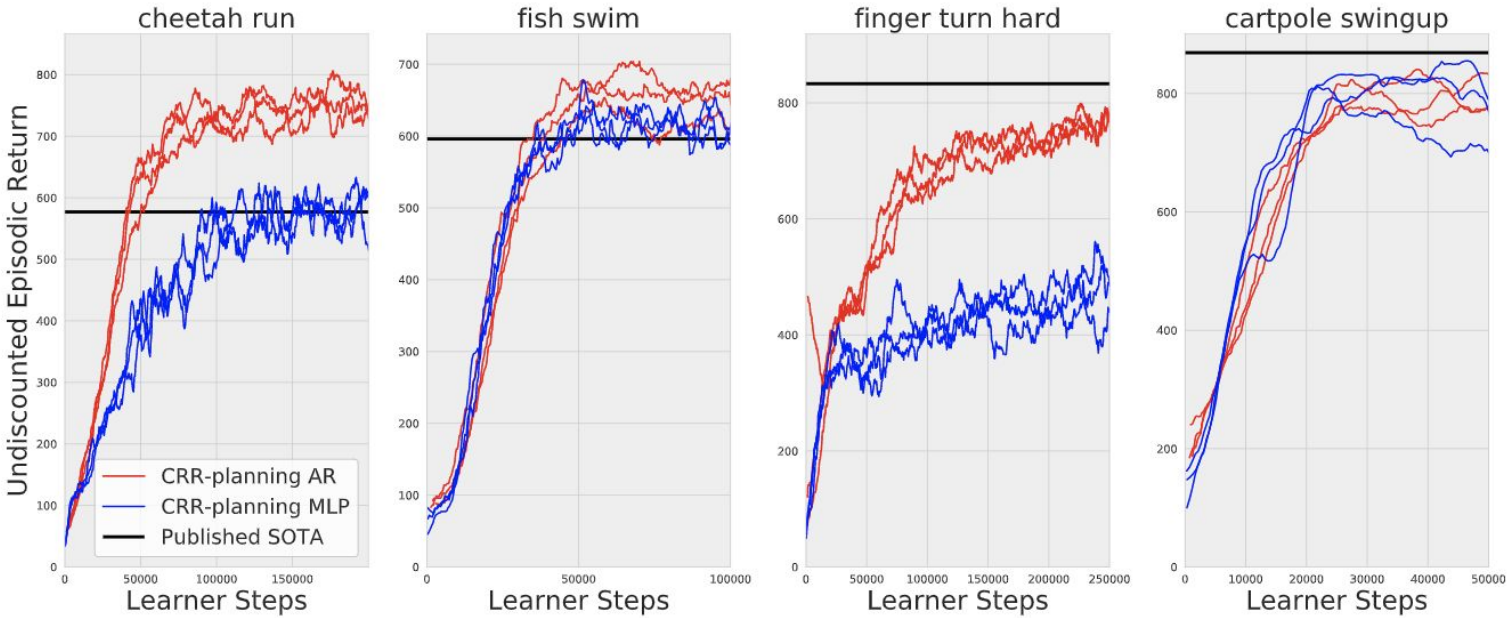
OPE results


< Absolute Error
0.6
0.4
0.2
0
MB-AR
FQE-D
DR
MB-FF
FQE-L2
DICE
IS
VPM
Baselines

OPE results


> Rank Correlation
0.6
0.4
0.2
0
MB-AR
FQE-D
MB-FF
FQE-L2
DR
VPM
IS
DICE
Baselines

Can we use dynamics models for policy optimization?



Model-based offline Policy optimization results. With planning and data augmentation, we manage to improve the performance over CRR exp (our baseline algorithm). When using autoregressive dynamics models (CRR-planning AR), we outperform state-of-the-art on Cheetah run and Fish swim.