Online Decision Transformer

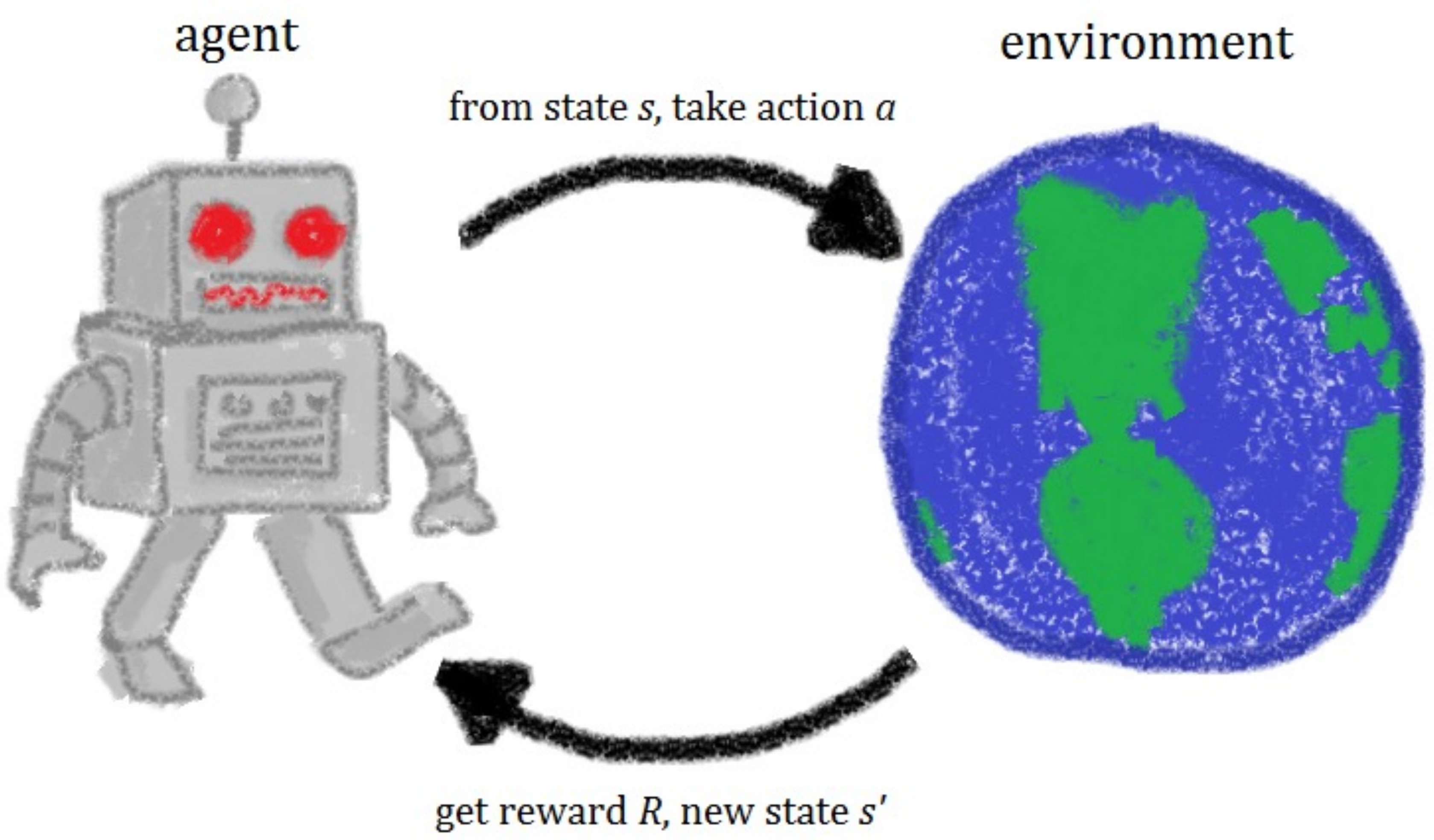
QINQING ZHENG, AMY ZHANG, ADITYA GROVER

ICML 2022

Meta

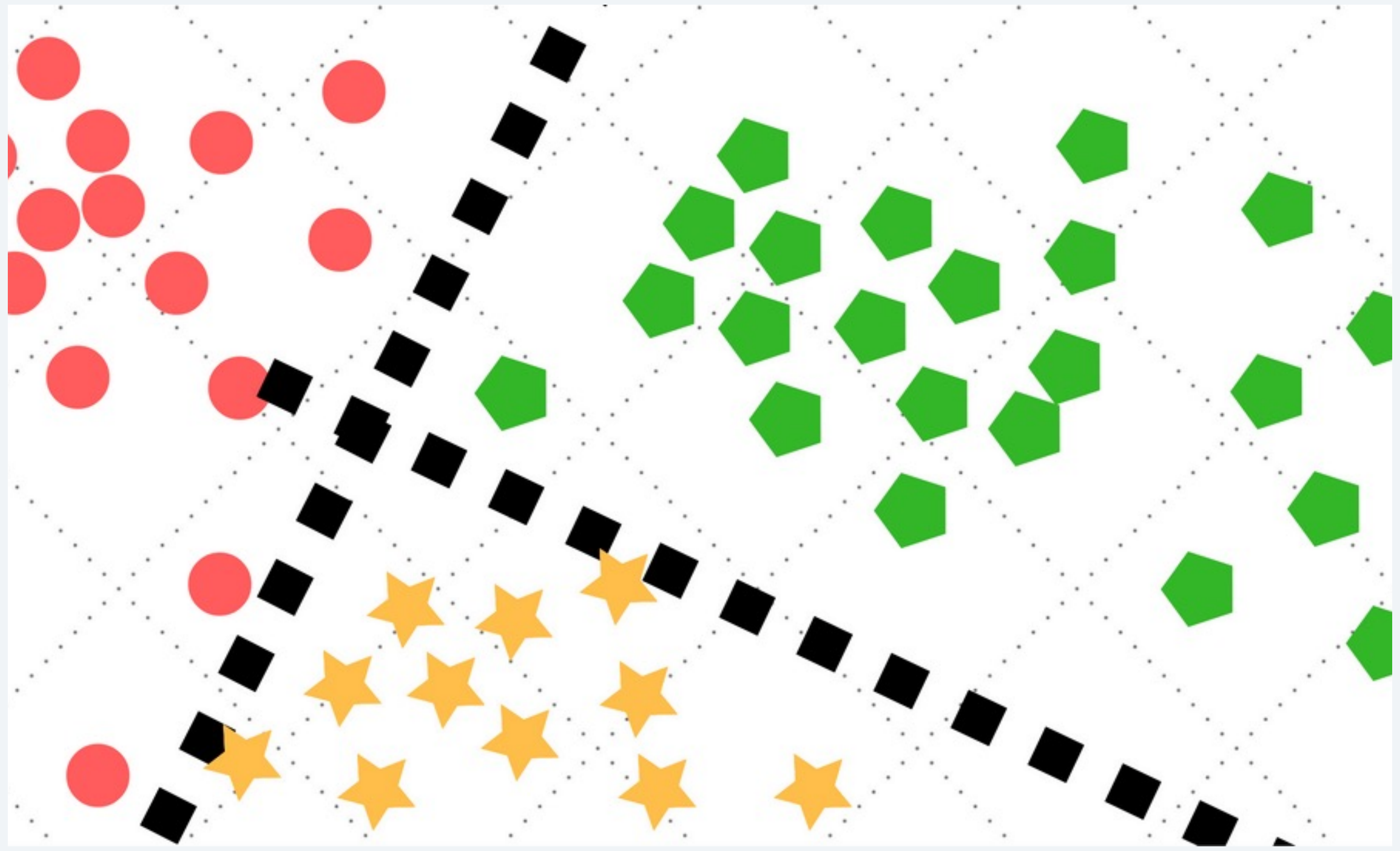
01 Problem & Motivation

Reinforcement Learning



Nonstatic dataset via feedback loop

Supervised Learning



Static labeled dataset

Data Obstacle of Reinforcement Learning

- Online data collection can be expensive, dangerous, and even infeasible (e.g., healthcare)
- Online data is limited in size, whereas utilizing extra, previously collected data is preferred for complex tasks

Offline Reinforcement Learning

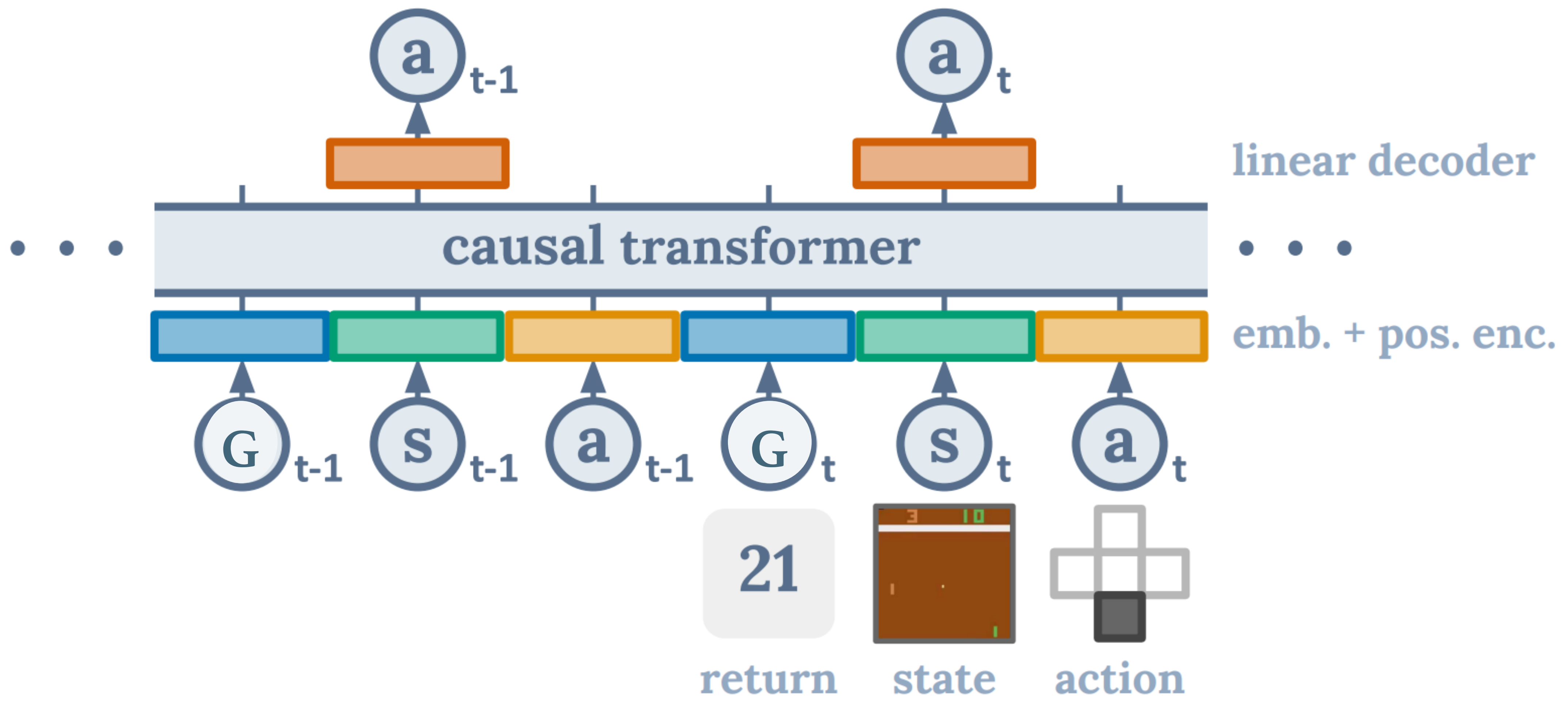
- Static dataset collected by certain (unknown) policies
- No online interactions
- Goal is still the same: obtain high return (total reward)

(figures taken from DeepMind blog)

Offline RL as Sequence Modeling

Decision Transformer (Chen et al. 2021), Trajectory Transformer (Janner et al. 2021):

- trajectory = sequence of (state, action, reward) tuples
- Transformer for autoregressive sequence modeling
- Conditional behavior cloning (BC)



DT architecture (Chen et al. 2021)

From Offline to Online Again

- Offline RL: performance is greatly influenced by the data quality
  - Data collected by expert/sub-optimal policies -> good/poor performance

From Offline to Online Again

- Offline RL: performance is greatly influenced by the data quality
  - Data collected by expert/sub-optimal policies -> good/poor performance
- Online RL: data collection is infeasible or expensive

From Offline to Online Again

- Offline RL: performance is greatly influenced by the data quality
  - Data collected by expert/sub-optimal policies -> good/poor performance
- Online RL: data collection is infeasible or expensive
- Hybrid: leverage both the stability of offline training and fresh data from online exploration

  Often needed in production systems! e.g. Ads Recommendation

From Offline to Online Again

- Hybrid: leverage both the stability of offline training and fresh data from online exploration

Can the pretraining (offline) + finetuning (online) paradigm, remarkably successful in language and vision, also be successful in RL? Improve upon the offline performance using very few online data.

From Offline to Online Again

- Hybrid: leverage both the stability of offline training and fresh data from online exploration

Can the pretraining (offline) + finetuning (online) paradigm, remarkably successful in language and vision, also be successful in RL? Improve upon the offline performance using very few online data.

At a high level, can purely supervised learning methods work well for RL in the online setting?

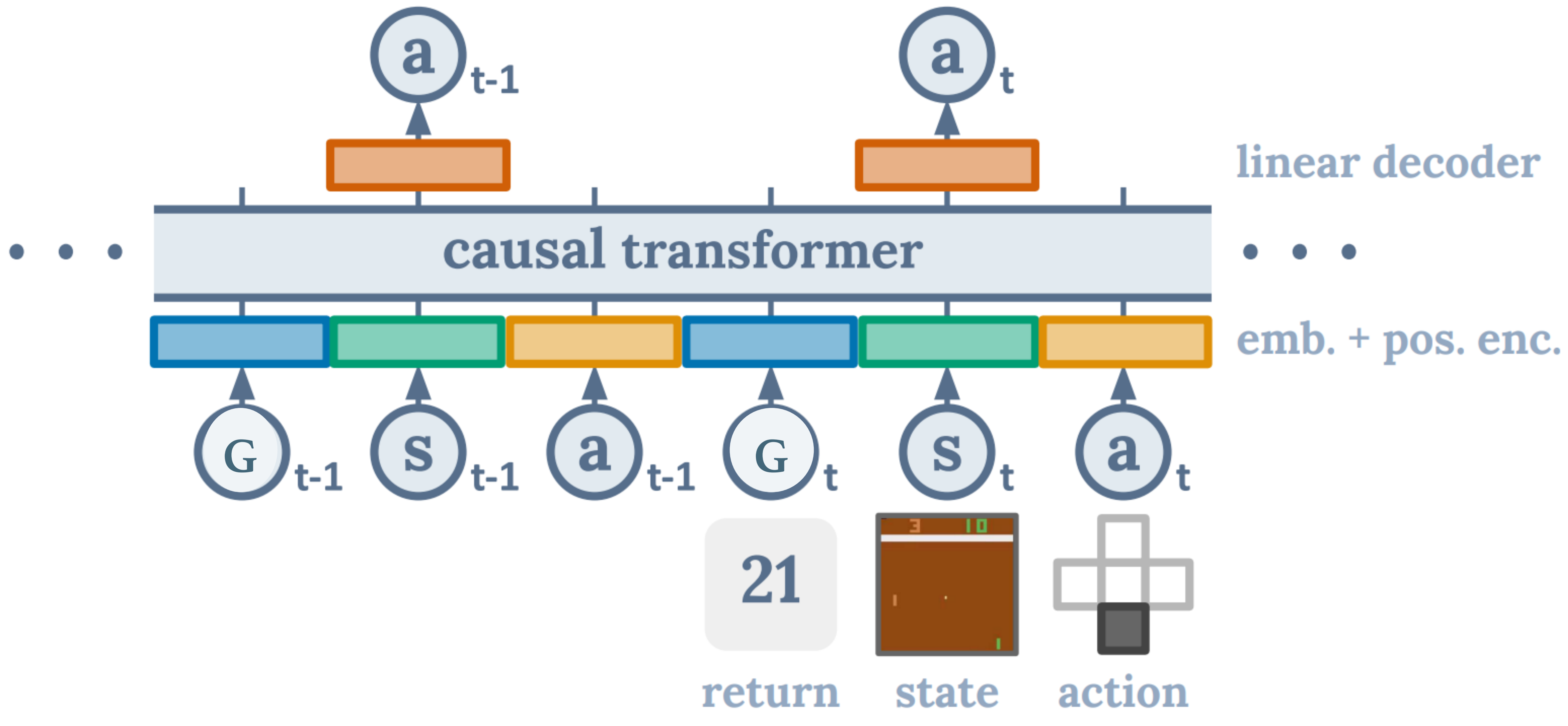
02 Online Decision Transformer

Basics

Decision Transformer (DT) models a trajectory τ as sequence of (RTG g, state s and action a) tuples

$$(g_1, s_1, a_1, g_2, s_2, a_2, \dots, g_{|\tau|}, s_{|\tau|}, a_{|\tau|})$$

$g_t = \sum_{t'=t}^{|\tau|} r_{t'}$ return-to-go (RTG) at timestep t



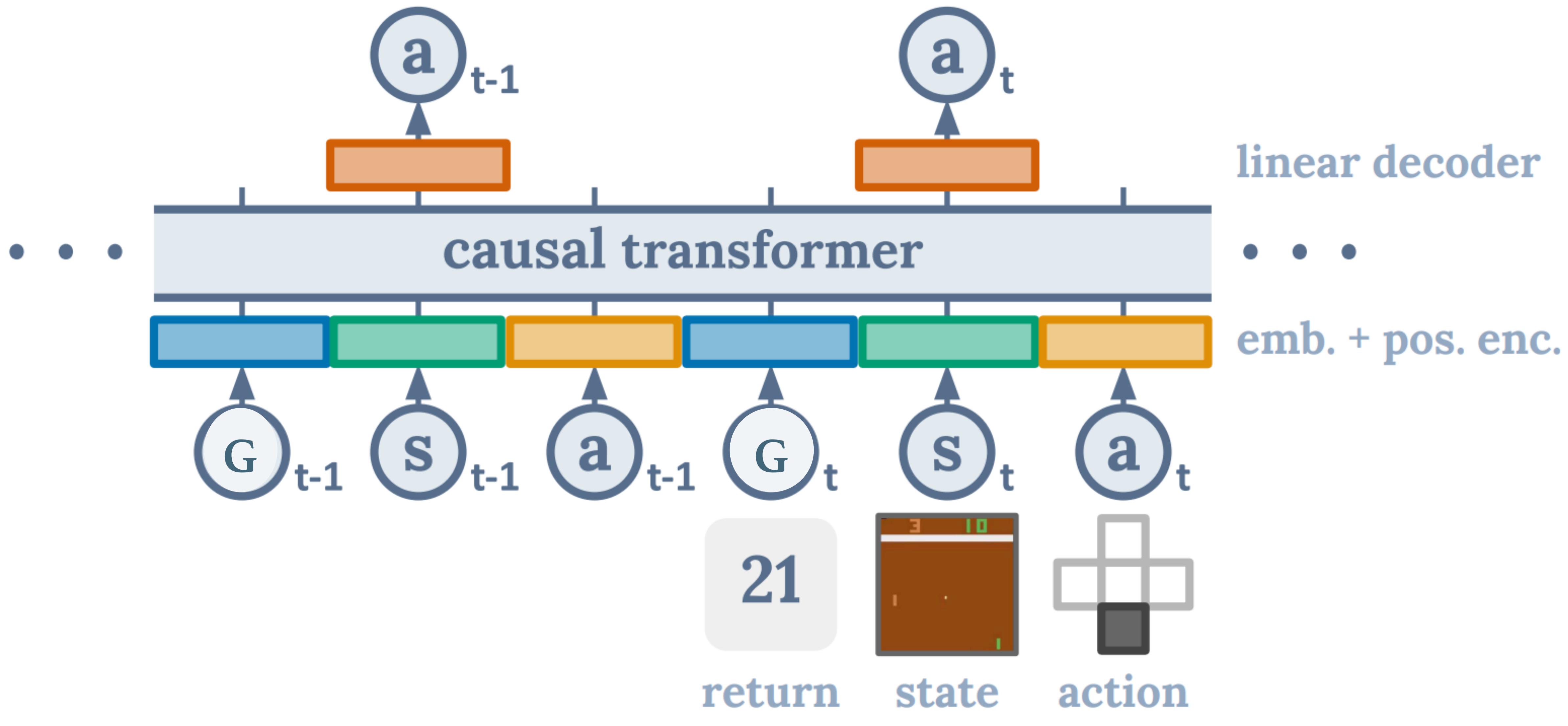
DT architecture (Chen et al. 2021).

Basics

DT generates return-conditioned policies.

Rollout:

1. Specify the desired return g_1 and an initial state s_1.
2. Generate a_1, execute it and then observe s_2 and r_1.
3. Compute $g_2 = g_1 - r_1$. Now we can generate a_2.
4. Repeat until the episode terminates.



DT architecture (Chen et al. 2021).

Online Decision Transformer

How to enable sample-efficient online exploration?

Online Decision Transformer

How to enable sample-efficient online exploration?

Max-entropy sequence modeling with carefully chosen design choices.

Max-Ent Sequence Modeling

Notation

$\mathcal{T}$ - training data distribution

K – context (input seq) length of Transformer

θ - parameter

$(\boldsymbol{a}, \boldsymbol{s}, \boldsymbol{g})$ - subtrajectory of length K

Stochastic Policy

$$\pi_\theta(a_t|\mathbf{s}_{-K,t}, \mathbf{g}_{-K,t}) = \mathcal{N}(\mu_\theta(\mathbf{s}_{-K,t}, \mathbf{g}_{-K,t}), \Sigma_\theta(\mathbf{s}_{-K,t}, \mathbf{g}_{-K,t})), \ \forall t.$$

generate action based on recent K states and RTGs

Formulation

$$\min_\theta J(\theta) \text{ subject to } H_\theta^{\mathcal{T}}[\mathbf{a}|\mathbf{s}, \mathbf{g}] \geqslant \beta$$

$J(\theta)$ – Negative log-likelihood

$H_\theta^{\mathcal{T}}[\mathbf{a}|\mathbf{s}, \mathbf{g}]$ – Policy Entropy

β – hyperparameter

we use –(act dim) as in SAC (Haarnoja et al. 2018)

Max-Ent Sequence Modeling

Key differences to SAC (Haarnoja et al. 2018) and other classic max-ent RL methods:

- Purely supervised learning of action sequences as opposed to maximizing returns

Objective of ODT

$$J(\theta) = \frac{1}{K}\mathbb{E}_{(\mathbf{a},\mathbf{s},\mathbf{g})\sim\mathcal{T}}[-\log \pi_\theta(\mathbf{a}|\mathbf{s},\mathbf{g})]$$
$$= \frac{1}{K}\mathbb{E}_{(\mathbf{a},\mathbf{s},\mathbf{g})\sim\mathcal{T}}[-\textstyle\sum_{k=1}^{K}\log \pi_\theta(a_k|\mathbf{s}_{-K,k},\mathbf{g}_{-K,k})]$$

minimize the loglikelihood of observed actions

Objective of Classic Max-Ent RL Methods

$$\mathrm{E}_{s_t\sim P(\cdot|s_{t-1}),a_t\sim\pi(\cdot|s_t)}[\textstyle\sum_t \gamma^t r(s_t,a_t)]$$

maximize the expected return

Max-Ent Sequence Modeling

Key differences to SAC (Haarnoja et al. 2018) and other classic max-ent RL methods:

- Purely supervised learning of action sequences as opposed to maximizing returns
- Entropy defined on sequence level as opposed to transition-level. For the same β, ODT has larger feasible set than SAC.

Policy Entropy of ODT

$$H_\theta^{\mathcal{T}}[\mathbf{a}|\mathbf{s},\mathbf{g}] = \frac{1}{K}\mathbb{E}_{(\mathbf{s},\mathbf{g})\sim\mathcal{T}}\left[H[\pi_\theta(\mathbf{a}|\mathbf{s},\mathbf{g})]\right]$$
$$=\frac{1}{K}\mathbb{E}_{(\mathbf{s},\mathbf{g})\sim\mathcal{T}}\left[\sum_{k=1}^{K} H[\pi_\theta(a_k|\mathbf{s}_{-K,k},\mathbf{g}_{-K,k})]\right]$$

expected average entropy of consecutive K actions

Policy Entropy of SAC

$$\mathbb{E}_{(\mathbf{s}_t,\mathbf{a}_t)\sim\rho_\pi}\left[-\log(\pi_t(\mathbf{a}_t|\mathbf{s}_t))\right]$$

expected per-action entropy

Training Pipeline

1. Offline Pretraining: train a policy on logged dataset
2. Online Finetuning

   Initialize the replay buffer by top logged trajectories

   Repeat

   1. Rollout a trajectory $\tau = \{(g_t, s_t, a_t)\}_{t=1}^{|\tau|}$ with chosen exploration RTG $g_1 = g_{online}$
   2. Hindsight return relabeling: edit RTG tokens in τ with observed reward $g_t = \sum_{t'=t}^{|\tau|} r_{t'}$
   3. Append τ to the replay buffer and remove the oldest trajectory
   4. Update the policy using data sampled from the replay buffer

03 Experiments

Benchmark Comparison

dataset	ODT (offline)	ODT (0.2m)	δ_{ODT}	IQL (offline)	IQL (0.2m)	δ_{IQL}
hopper-medium	66.95 ± 3.26	$\mathbf{97.54 \pm 2.10}$	**30.59**	63.81 ± 9.15	66.79 ± 4.07	2.98
hopper-medium-replay	86.64 ± 5.41	88.89 ± 6.33	2.25	92.13 ± 10.43	$\mathbf{96.23 \pm 4.35}$	**4.10**
walker2d-medium	72.19 ± 6.49	76.79 ± 2.30	**4.60**	79.89 ± 3.06	$\mathbf{80.33 \pm 2.33}$	0.44
walker2d-medium-replay	68.92 ± 4.79	$\mathbf{76.86 \pm 4.04}$	**7.94**	73.67 ± 6.37	70.55 ± 5.81	-3.12
halfcheetah-medium	42.72 ± 0.46	42.16 ± 1.48	-0.56	47.37 ± 0.29	47.41 ± 0.15	**0.04**
halfcheetah-medium-replay	39.99 ± 0.68	40.42 ± 1.61	**0.43**	44.10 ± 1.14	44.14 ± 0.3	0.04
ant-medium	91.33 ± 4.13	90.79 ± 5.80	-0.54	99.92 ± 5.86	$\mathbf{100.85 \pm 2.02}$	**0.93**
ant-medium-replay	86.56 ± 3.26	$\mathbf{91.57 \pm 2.73}$	**5.01**	91.21 ± 7.27	91.36 ± 1.47	0.15
sum		**605.02**	**49.72**		597.66	5.56
antmaze-umaze	53.10 ± 4.21	$\mathbf{88.5 \pm 5.88}$	**35.4**	87.1 ± 2.81	$\mathbf{89.5 \pm 5.43}$	2.4
antmaze-umaze-diverse	50.20 ± 6.69	$\mathbf{56.00 \pm 5.69}$	**7.99**	64.4 ± 8.95	$\mathbf{56.8 \pm 6.42}$	-7.6
sum		**144.5**	**43.39**		**146.3**	-5.2

Dataset: D4RL

Baseline: Implicit Q Learning (IQL, Kostrikov et al. 2021)

Benchmark Comparison

dataset	ODT (offline)	ODT (0.2m)	δ_{ODT}	IQL (offline)	IQL (0.2m)	δ_{IQL}
hopper-medium	66.95 ± 3.26	$\mathbf{97.54 \pm 2.10}$	**30.59**	63.81 ± 9.15	66.79 ± 4.07	2.98
hopper-medium-replay	86.64 ± 5.41	88.89 ± 6.33	2.25	92.13 ± 10.43	$\mathbf{96.23 \pm 4.35}$	**4.10**
walker2d-medium	72.19 ± 6.49	76.79 ± 2.30	**4.60**	79.89 ± 3.06	$\mathbf{80.33 \pm 2.33}$	0.44
walker2d-medium-replay	68.92 ± 4.79	$\mathbf{76.86 \pm 4.04}$	**7.94**	73.67 ± 6.37	70.55 ± 5.81	-3.12
halfcheetah-medium	42.72 ± 0.46	42.16 ± 1.48	-0.56	47.37 ± 0.29	47.41 ± 0.15	**0.04**
halfcheetah-medium-replay	39.99 ± 0.68	40.42 ± 1.61	**0.43**	44.10 ± 1.14	44.14 ± 0.3	0.04
ant-medium	91.33 ± 4.13	90.79 ± 5.80	-0.54	99.92 ± 5.86	$\mathbf{100.85 \pm 2.02}$	**0.93**
ant-medium-replay	86.56 ± 3.26	$\mathbf{91.57 \pm 2.73}$	**5.01**	91.21 ± 7.27	91.36 ± 1.47	0.15
sum		**605.02**	**49.72**		597.66	5.56
antmaze-umaze	53.10 ± 4.21	$\mathbf{88.5 \pm 5.88}$	**35.4**	87.1 ± 2.81	$\mathbf{89.5 \pm 5.43}$	2.4
antmaze-umaze-diverse	50.20 ± 6.69	$\mathbf{56.00 \pm 5.69}$	**7.99**	64.4 ± 8.95	$\mathbf{56.8 \pm 6.42}$	-7.6
sum		**144.5**	**43.39**		**146.3**	-5.2

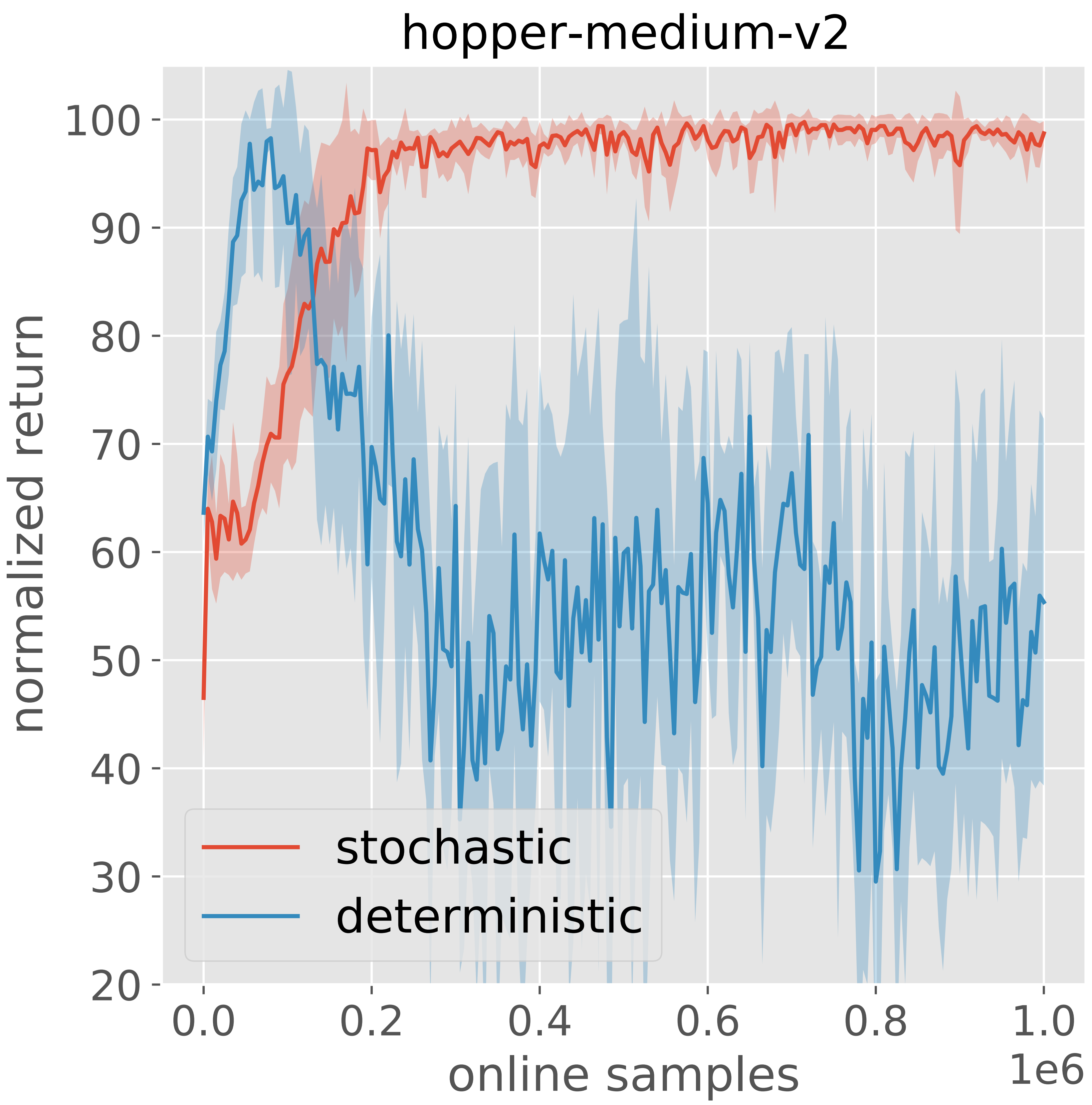
Absolute Performance: ODT is better or comparable

Benchmark Comparison

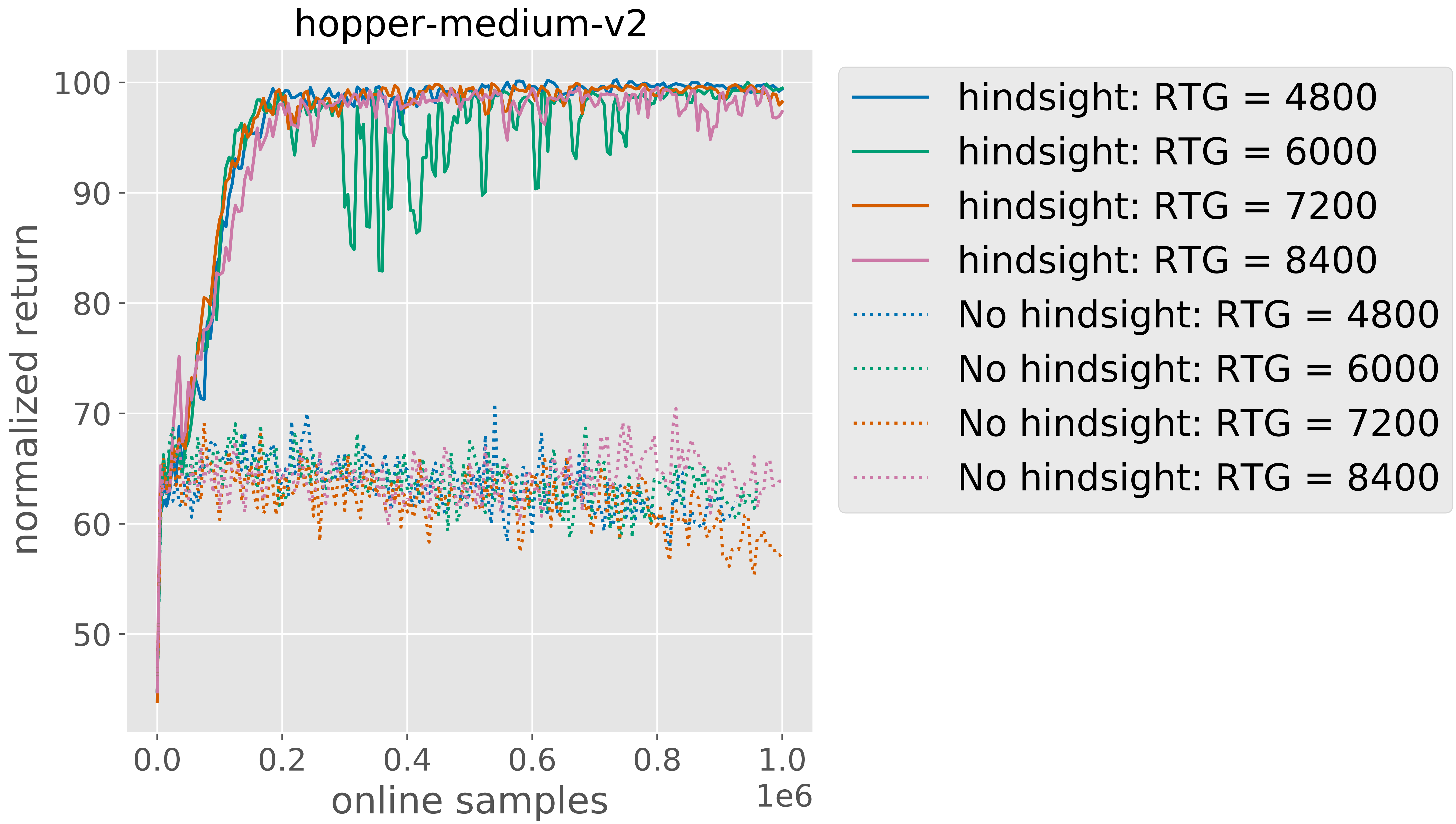
dataset	ODT (offline)	ODT (0.2m)	δ_{ODT}	IQL (offline)	IQL (0.2m)	δ_{IQL}
hopper-medium	66.95 ± 3.26	**97.54 ± 2.10**	**30.59**	63.81 ± 9.15	66.79 ± 4.07	2.98
hopper-medium-replay	86.64 ± 5.41	88.89 ± 6.33	2.25	92.13 ± 10.43	**96.23 ± 4.35**	**4.10**
walker2d-medium	72.19 ± 6.49	76.79 ± 2.30	**4.60**	79.89 ± 3.06	**80.33 ± 2.33**	0.44
walker2d-medium-replay	68.92 ± 4.79	**76.86 ± 4.04**	**7.94**	73.67 ± 6.37	70.55 ± 5.81	−3.12
halfcheetah-medium	42.72 ± 0.46	42.16 ± 1.48	−0.56	47.37 ± 0.29	47.41 ± 0.15	**0.04**
halfcheetah-medium-replay	39.99 ± 0.68	40.42 ± 1.61	**0.43**	44.10 ± 1.14	44.14 ± 0.3	0.04
ant-medium	91.33 ± 4.13	90.79 ± 5.80	−0.54	99.92 ± 5.86	**100.85 ± 2.02**	**0.93**
ant-medium-replay	86.56 ± 3.26	**91.57 ± 2.73**	**5.01**	91.21 ± 7.27	91.36 ± 1.47	0.15
sum		**605.02**	**49.72**		597.66	5.56
antmaze-umaze	53.10 ± 4.21	**88.5 ± 5.88**	**35.4**	87.1 ± 2.81	**89.5 ± 5.43**	2.4
antmaze-umaze-diverse	50.20 ± 6.69	**56.00 ± 5.69**	**7.99**	64.4 ± 8.95	**56.8 ± 6.42**	−7.6
sum		**144.5**	**43.39**		**146.3**	−5.2

Finetuning Gain: ODT is much better!

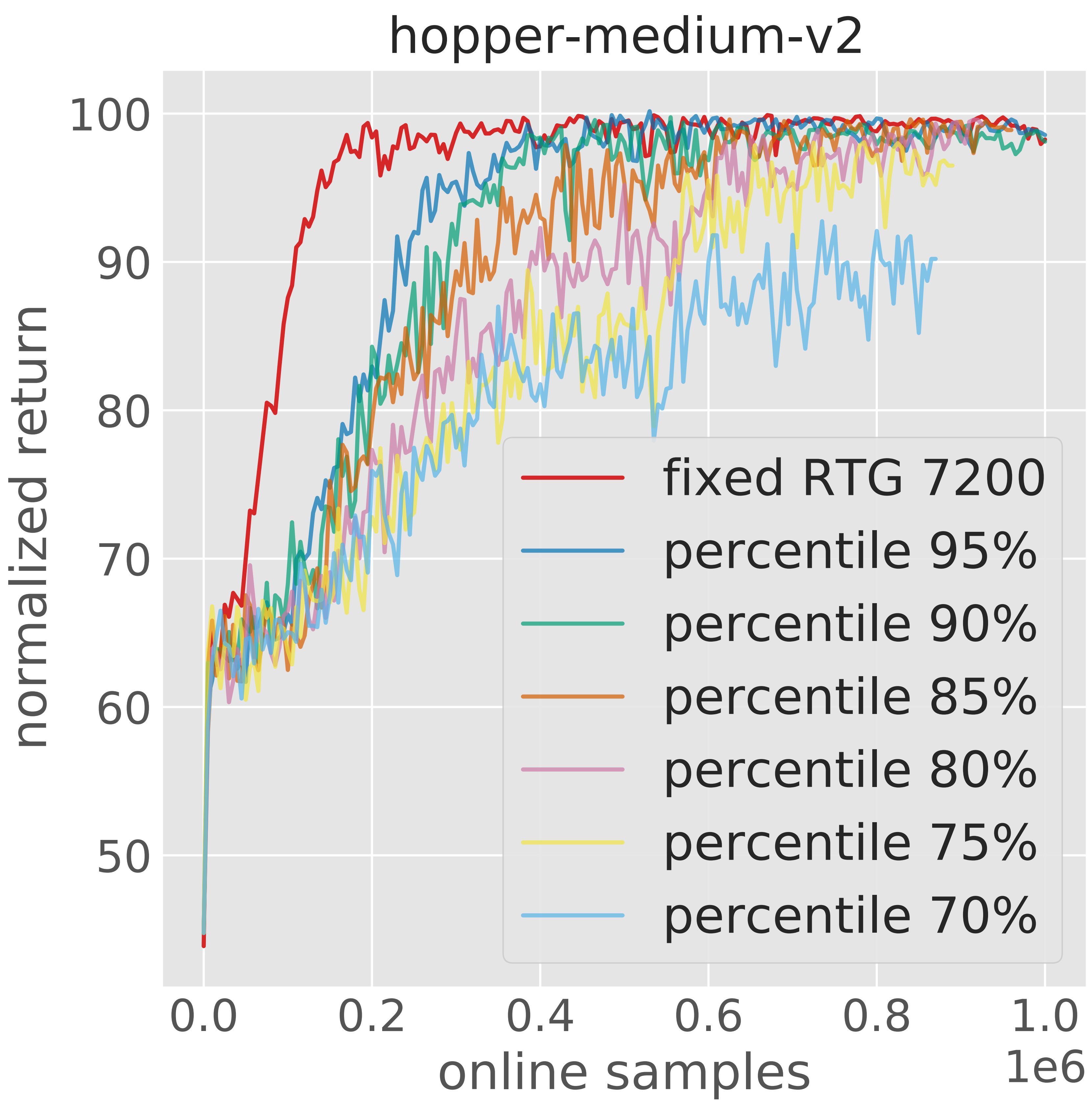
Ablation Study



Stochasticity is important to enable stable performance improvement in online training



Hindsight return relabeling is critical for correcting bias in the collected data



Fixed, large, (potentially) out-of-distribution return is good for g_{online}

We use 2x expert performance

04 Summary and Open Problems

Summary

- Blend offline pretraining with efficient online finetuning of sequence models for RL in a unified framework
- Supervised learning paradigm is of great potential in online settings

Open Problems

Optimization

Could we establish the convergence guarantee of ODT?

Optimization

Could we establish the convergence guarantee of ODT?

Generalization

When will ODT perform well or poorly?

Could ODT account for purely online settings?

Optimization

Could we establish the convergence guarantee of ODT?

Generalization

When will ODT perform well or poorly?

Could ODT account for purely online settings?

BC vs Value

How does ODT, or, in general, online conditional BC algorithms, compare to value-based RL methods?

Thanks!