Efficient Offline Policy Optimization with a Learned Model

Zichen Liu[1,2], Siyi Li[1], Wee Sun Lee[2], Shuicheng Yan[1], Zhongwen Xu[1]

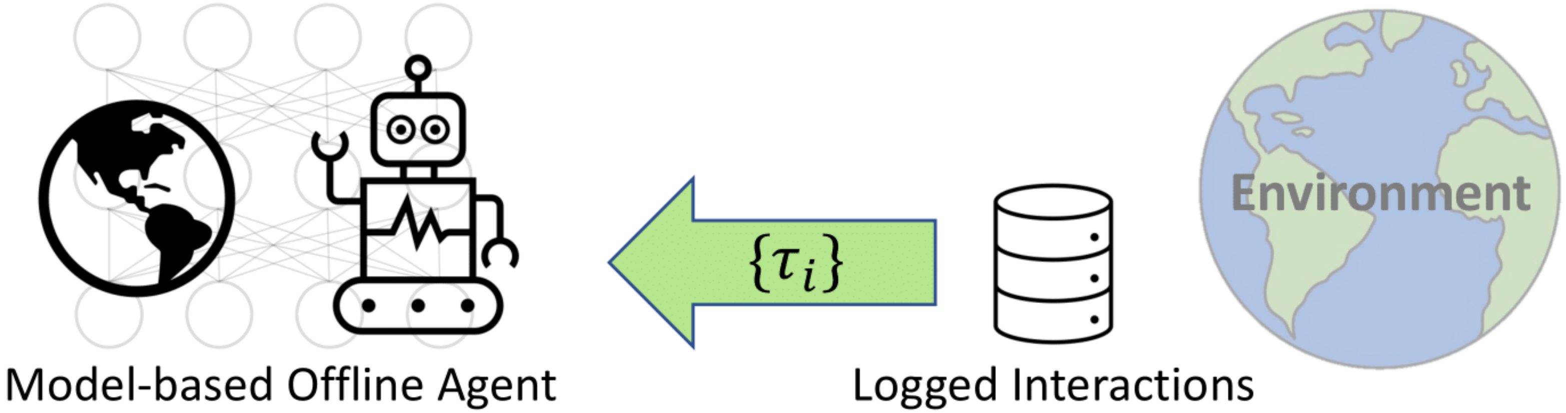



1 sea connecting the dots | AI Lab

2 NUS National University of Singapore | Computing

Background

❑ We work on the intersection of **model-based RL** and **offline RL**



❑ offline RL

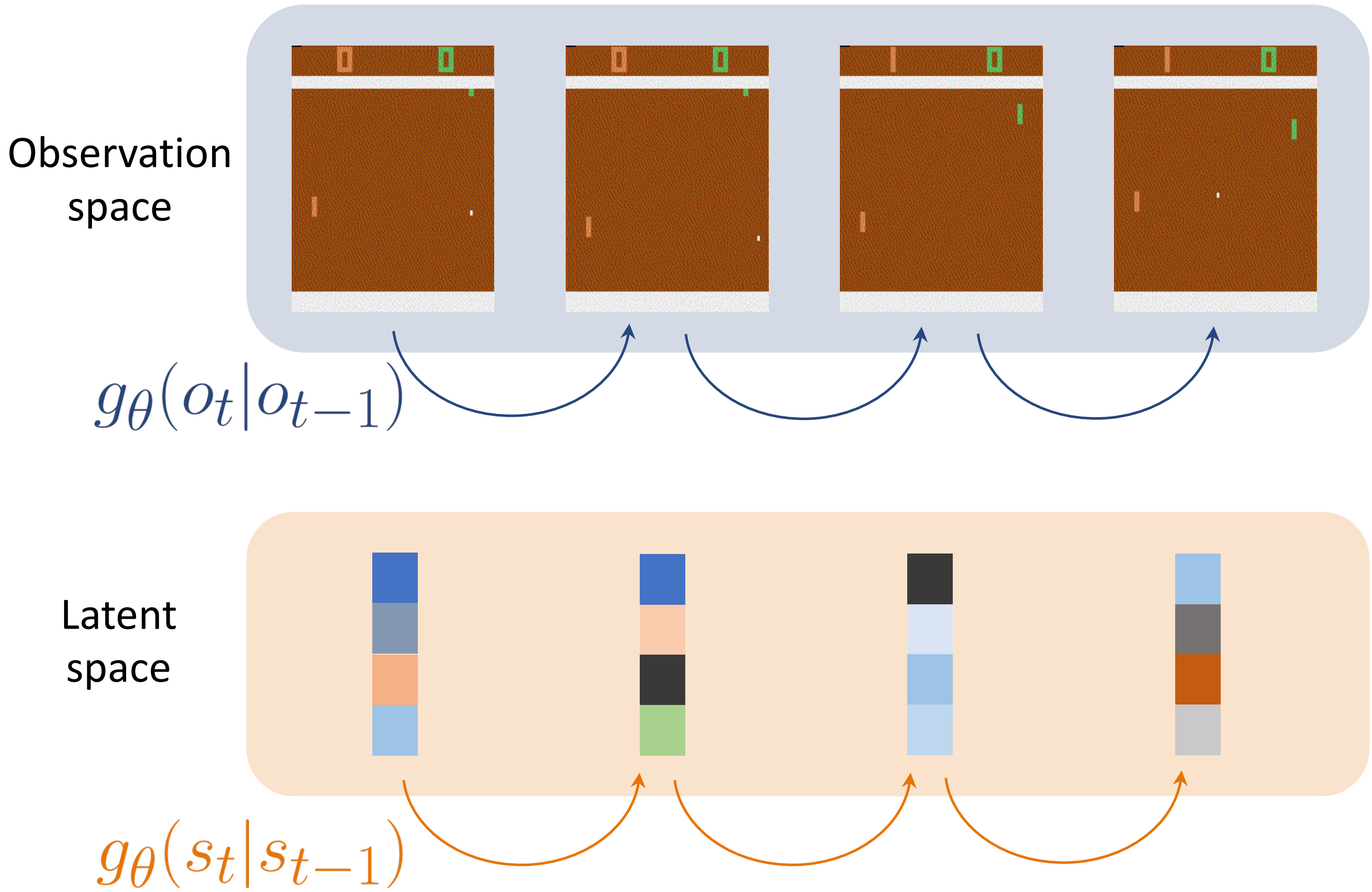
- ❑ avoid environment interaction, which can be costly or unsafe

❑ model-based RL

- ❑ data-efficiency
- ❑ richer supervision signals
- ❑ helps policy and value learning

Environment Model

❑ Learning an <u>explicit</u> or <u>latent</u> model about the environment dynamics?



Model in latent space advantages

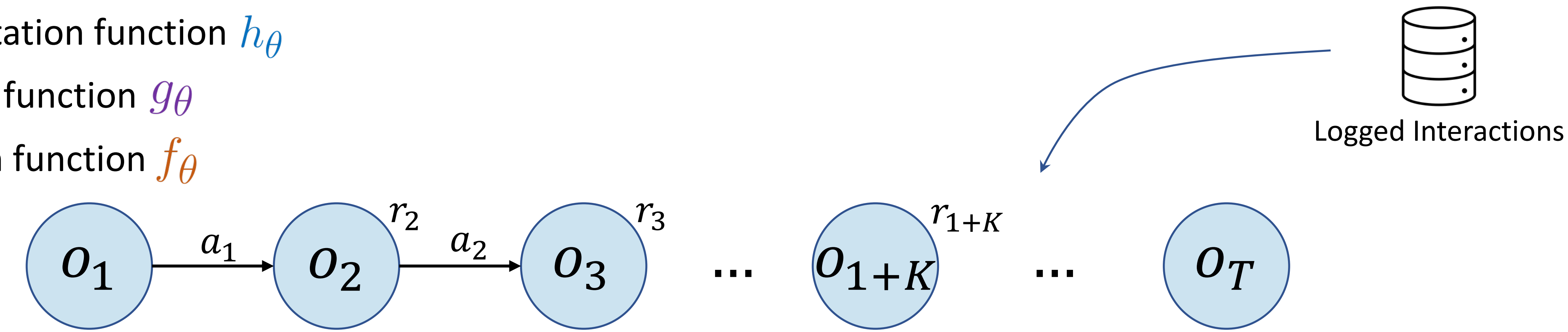
- Computational efficiency
- Helps representation learning (Schrittwieser, 2020; Hessel, 2021)

Latent Dynamics Model

- Representation function h_θ
- Dynamics function g_θ
- Prediction function f_θ

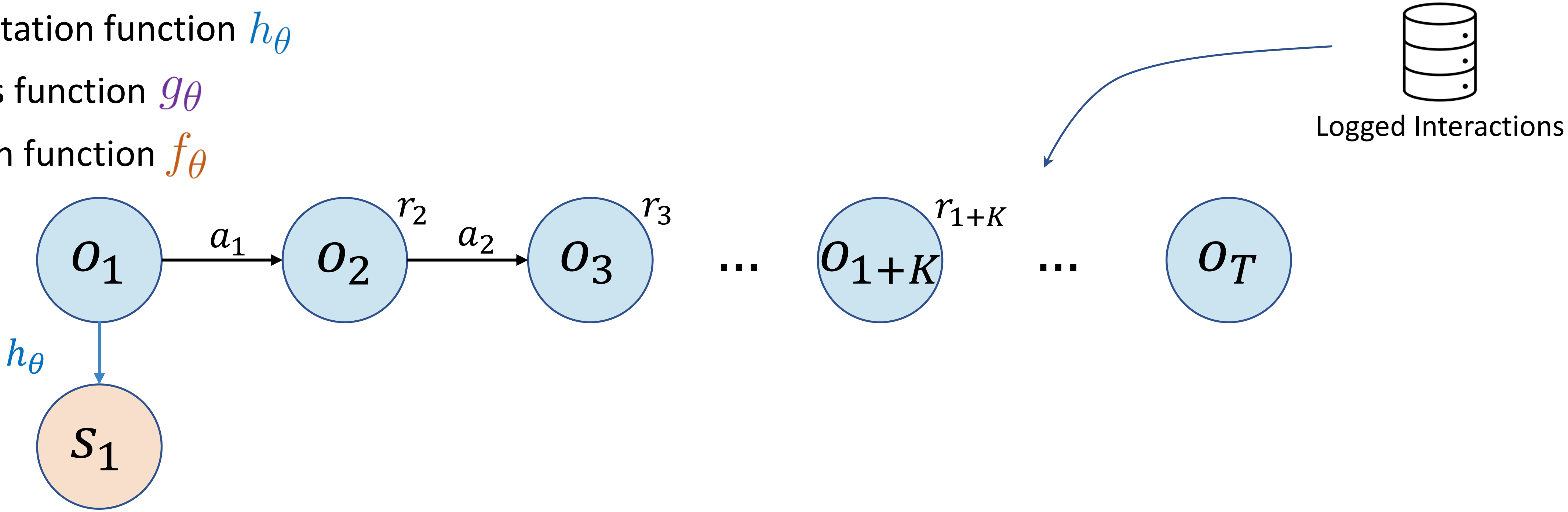
Latent Dynamics Model

- ❑ Representation function h_θ
- ❑ Dynamics function g_θ
- ❑ Prediction function f_θ



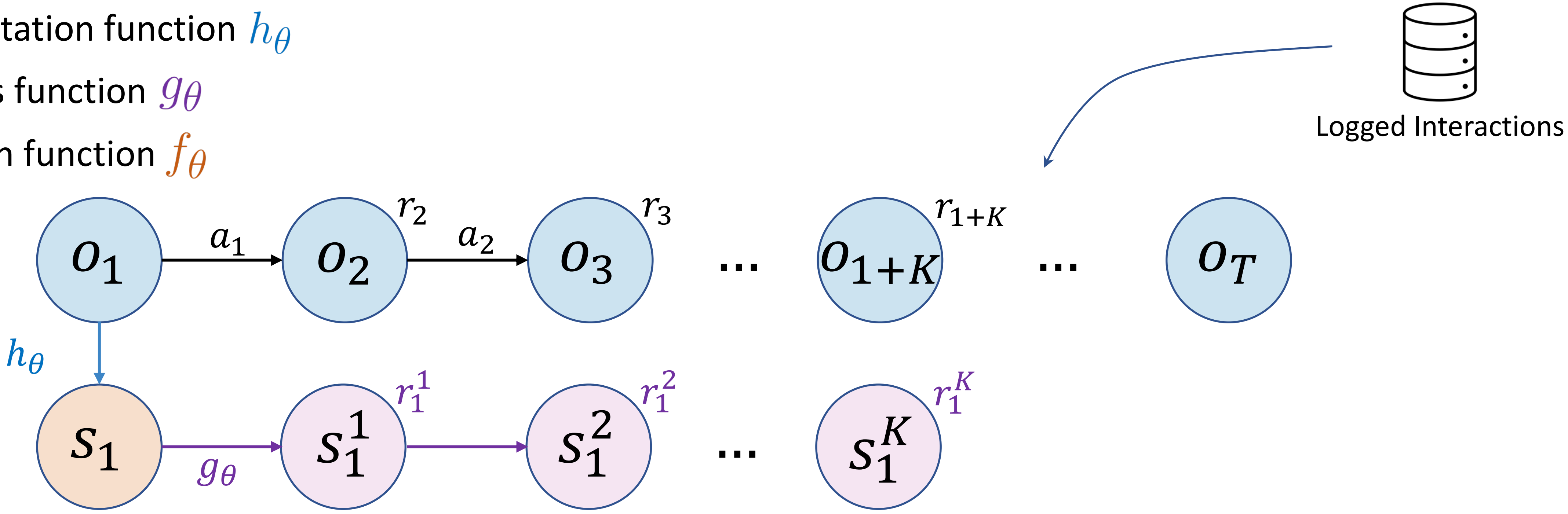
Latent Dynamics Model

- ❑ Representation function h_θ
- ❑ Dynamics function g_θ
- ❑ Prediction function f_θ



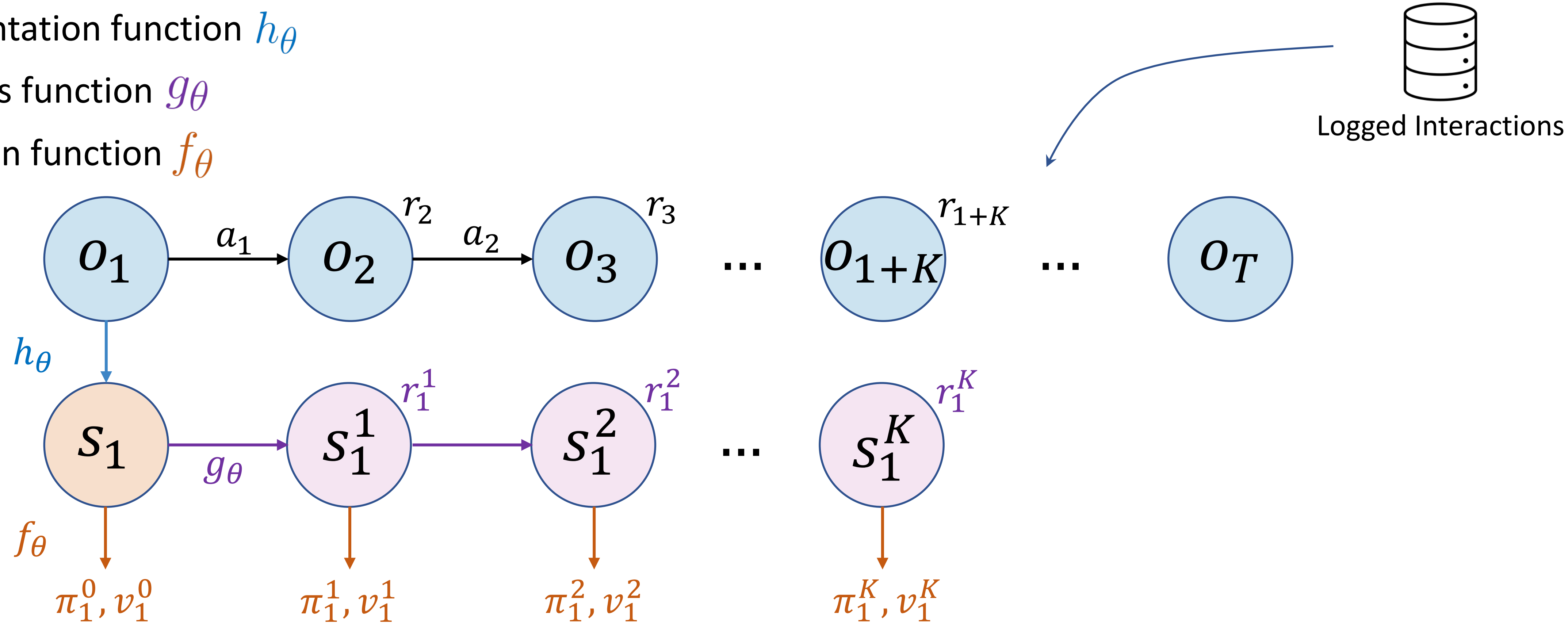
Latent Dynamics Model

- ❑ Representation function h_θ
- ❑ Dynamics function g_θ
- ❑ Prediction function f_θ



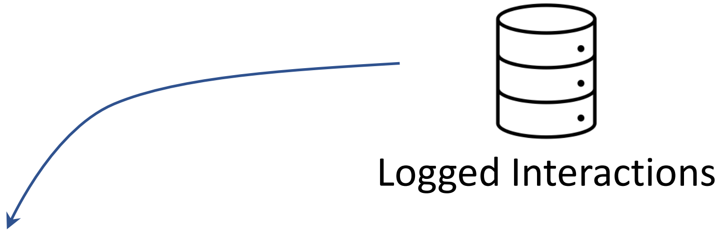
Latent Dynamics Model

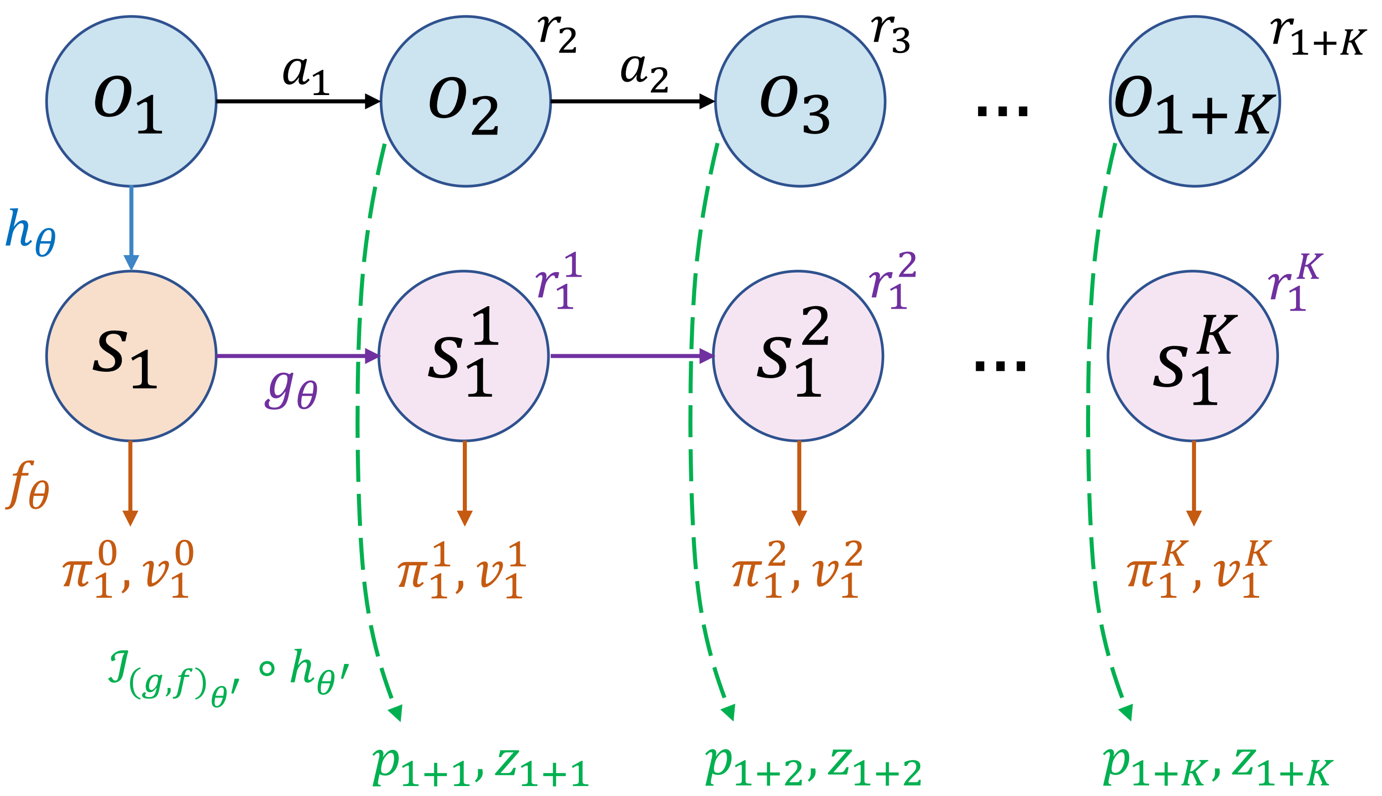
- ❑ Representation function h_θ
- ❑ Dynamics function g_θ
- ❑ Prediction function f_θ



Latent Dynamics Model

- ❑ Representation function h_θ
- ❑ Dynamics function g_θ
- ❑ Prediction function f_θ





- Improvement operator $\mathcal{I}$ with target network θ'

$$\ell_t(\theta) = \sum_{k=0}^{K} \ell^v\left(\boxed{v_t^k}, \boxed{z_{t+k}}\right) + \ell^\pi\left(\boxed{\pi_t^k}, \boxed{p_{t+k}}\right)$$

Monte-Carlo Tree Search as Improvement

- [1]Action selection using the pUCT rule

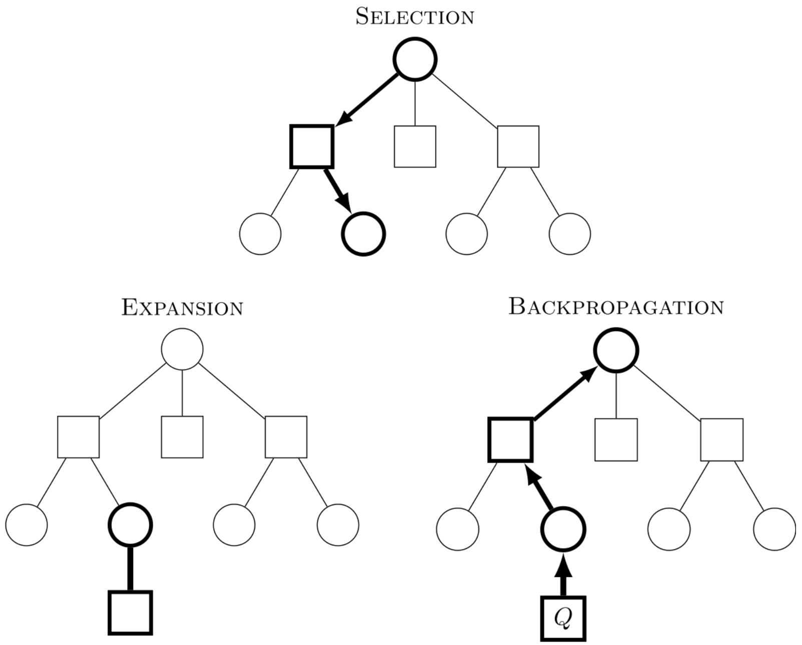
$$a^k = \arg\max_a \left[Q(s,a) + \pi_{\text{prior}}(s,a) \cdot \frac{\sqrt{\sum_b n(s,b)}}{1+n(s,a)} \cdot \left(c_1 + \log\left(\frac{\sum_b n(s,b) + c_2 + 1}{c_2} \right) \right) \right]$$

- [2]Node expansion and [3]backup to update Q and n statistics

- After exhausting simulation budget, output *normalized visit count* and *n-step return*

$$p_{\text{MCTS}}(a|s_t) = \frac{n(s_t^0, a)^{1/T}}{\sum_b n(s_t^0, b)^{1/T}},$$

$$z_{\text{MCTS}}(s_t) = \gamma^n \sum_a \left(\frac{n(s_{t+n}^0, a)}{\sum_b n(s_{t+n}^0, b)} \right) Q(s_{t+n}^0, a) + \sum_{t'=t}^{t+n-1} \gamma^{t'-t} r_{t'}^{\text{env}}.$$



$$\ell_t(\theta) = \sum_{k=0}^{K} \ell^v \left(v_t^k, \boxed{z_{t+k}} \right) + \ell^\pi \left(\pi_t^k, \boxed{p_{t+k}} \right)$$

Monte-Carlo Tree Search as Improvement

- (1)Action selection using the pUCT rule

$$a^k = \arg\max_a \left[Q(s,a) + \pi_{\text{prior}}(s,a) \cdot \frac{\sqrt{\sum_b n(s,b)}}{1+n(s,a)} \cdot \left(c_1 + \log\left(\frac{\sum_b n(s,b) + c_2 + 1}{c_2} \right) \right) \right]$$

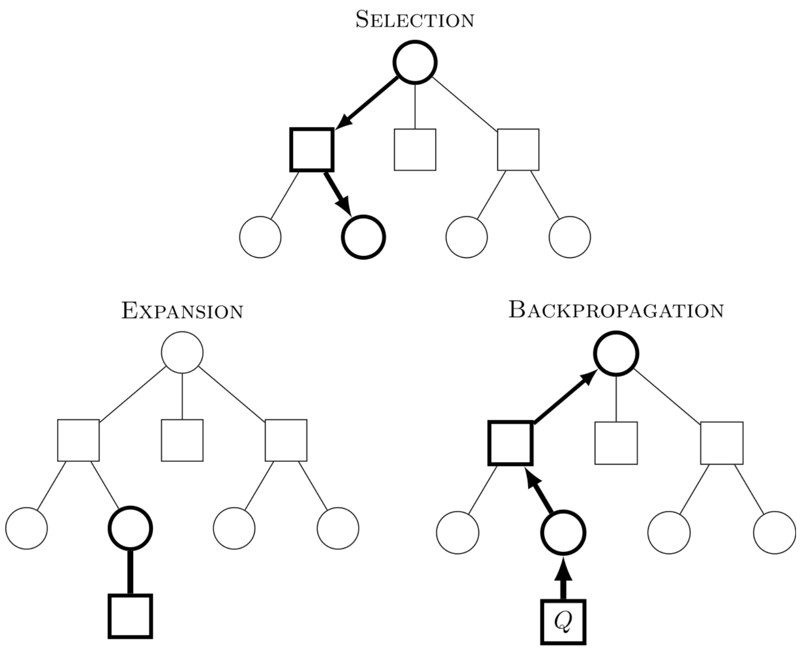
- (2)Node expansion and (3)backup to update Q and n statistics

- After exhausting simulation budget, output *normalized visit count* and *n-step return*

$$p_{\text{MCTS}}(a|s_t) = \frac{n(s_t^0, a)^{1/T}}{\sum_b n(s_t^0, b)^{1/T}},$$

$$z_{\text{MCTS}}(s_t) = \gamma^n \sum_a \left(\frac{n(s_{t+n}^0, a)}{\sum_b n(s_{t+n}^0, b)} \right) Q(s_{t+n}^0, a) + \sum_{t'=t}^{t+n-1} \gamma^{t'-t} r_{t'}^{\text{env}}.$$

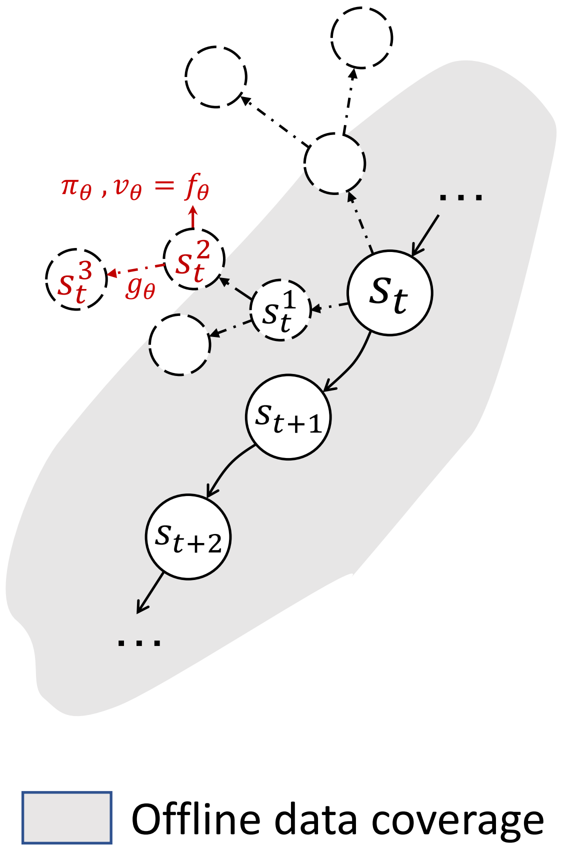
- Used as the improvement operator in MuZero (Schrittwieser, 2020) and achieves SoTA in offline RL (Schrittwieser, 2021)



$$\ell_t(\theta) = \sum_{k=0}^{K} \ell^v\left(v_t^k, \boxed{z_{t+k}}\right) + \ell^\pi\left(\pi_t^k, \boxed{p_{t+k}}\right)$$

MCTS Deficiencies

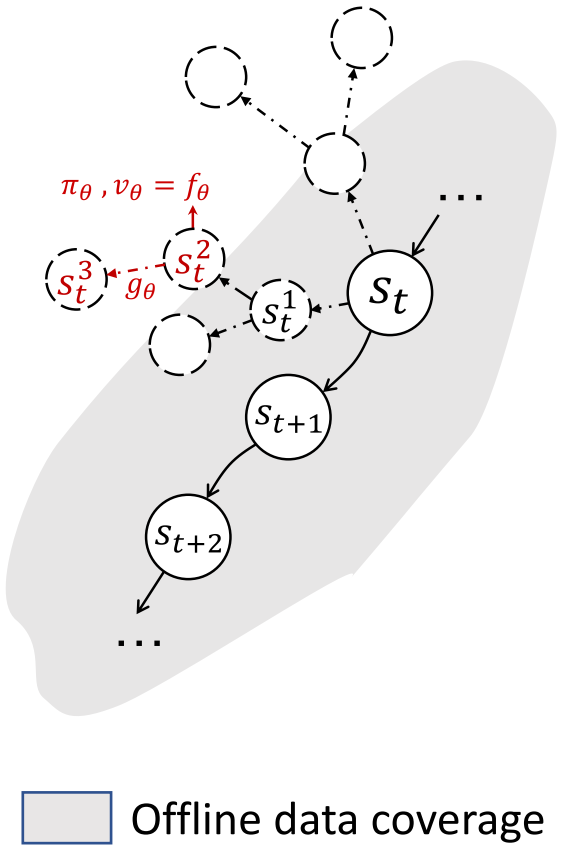
- **Compounding model errors** outside the coverage of the offline data



MCTS Deficiencies

- ❑ **Compounding model errors** outside the coverage of the offline data
- ❑ **Limited expressiveness** when simulation budget is low
  - ❑ Theoretically justified by Grill (2020)
- ❑ **Computational burden** while increasing simulation budget

$$p_{\mathrm{MCTS}}(a|s_t) = \frac{n(s_t^0, a)^{1/T}}{\sum_b n(s_t^0, b)^{1/T}}$$



Proposal 1/2: One-step Look-ahead

- In model-based RL, *short-horizon* rollout is shown to be better (Janner, 2019; Hessel, 2021)

- We propose to utilize the learned latent dynamics model for *one-step look-ahead*

Proposal 1/2: One-step Look-ahead

- In model-based RL, *short-horizon* rollout is shown to be better (Janner, 2019; Hessel, 2021)

- We propose to utilize the learned latent dynamics model for *one-step look-ahead*

$$\ell^p = -\mathbf{p}^\top \log \boldsymbol{\pi}$$

Proposal 1/2: One-step Look-ahead

- In model-based RL, *short-horizon* rollout is shown to be better (Janner, 2019; Hessel, 2021)

- We propose to utilize the learned latent dynamics model for *one-step look-ahead*

$$\ell^p = -\mathbf{p}^\top \log \boldsymbol{\pi}$$

$$p_{\mathrm{mcts}}(a|s) = \frac{n(s,a)}{\sum_b n(s,b)}$$

Proposal 1/2: One-step Look-ahead

- In model-based RL, *short-horizon* rollout is shown to be better (Janner, 2019; Hessel, 2021)

- We propose to utilize the learned latent dynamics model for *one-step look-ahead*

$$\ell^p = -\mathbf{p}^\top \log \boldsymbol{\pi}$$

$$p_{\mathrm{mcts}}(a|s) = \frac{n(s,a)}{\sum_b n(s,b)}$$

$$p_{\mathrm{os}}(a|s) = \frac{\pi_{\mathrm{prior}}\,(a|s) \exp\left(\mathrm{adv}_g(s,a)\right)}{Z(s)}$$

Proposal 1/2: One-step Look-ahead

- In model-based RL, *short-horizon* rollout is shown to be better (Janner, 2019; Hessel, 2021)

- We propose to utilize the learned latent dynamics model for *one-step look-ahead*

$$\ell^p = -\mathbf{p}^\top \log \boldsymbol{\pi}$$

$$p_{\mathrm{mcts}}(a|s) = \frac{n(s,a)}{\sum_b n(s,b)}$$

$$p_{\mathrm{os}}(a|s) = \frac{\pi_{\mathrm{prior}}\,(a|s)\exp\left(\mathrm{adv}_g(s,a)\right)}{Z(s)}$$

advantage function

$$\mathrm{adv}_g(s,a) = q_g(s,a) - v(s)$$

one-step model rollout

$$r_g, s_g' = g_\theta(s,a)$$

definition of action-value

$$q_g(s,a) = r_g + \gamma f_{\theta,v}\left(s_g'\right)$$

Proposal 2/2: Behavior Regularization

- Constrain the policy towards behavior policy (inferred from the dataset) is a key technique in offline RL (Levine, 2020)

- We propose to perform a filtered behavior cloning

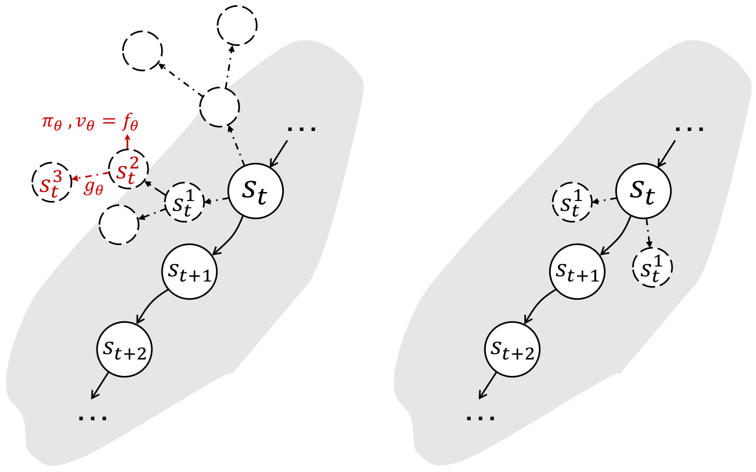
$$\ell_{\pi}^{\mathrm{reg}}(\theta) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[-\log \pi(a \mid s) \cdot H\left(\mathrm{adv}_g(s,a)\right)\right]$$

$$H(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

Desiderata Revisited

- ❑ Combat with model errors
  - ❑ One-step unrolling limits model expansion depth up to 1
  - ❑ Behavior regularization ensures policy not go far from data coverage

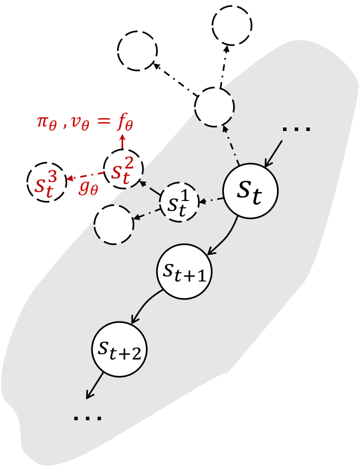
- ❑ Computational efficiency
  - ❑ No need for expensive MCTS



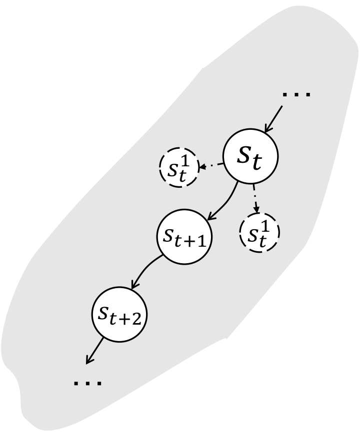
MCTS Ours

Desiderata Revisited

- ❑ Combat with model errors
  - ❑ One-step unrolling limits model expansion depth up to 1
  - ❑ Behavior regularization ensures policy not go far from data coverage



MCTS



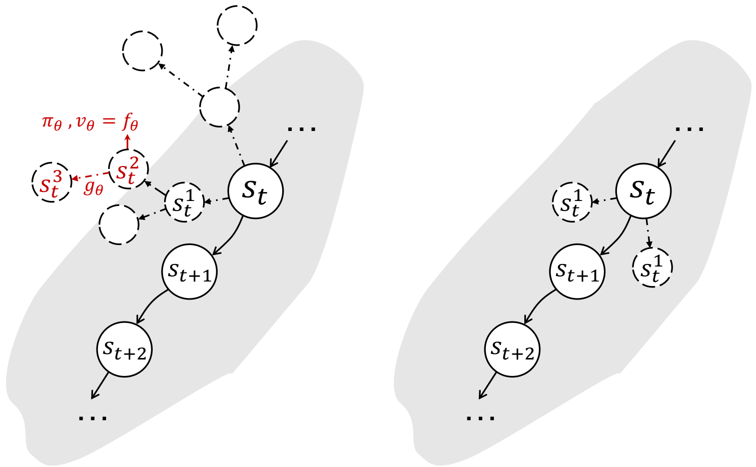
Ours

- ❑ Computational efficiency
  - ❑ No need for expensive MCTS
  - ❑ Sampling is available when action space is too large

$$\ell_{\mathrm{os}}^{\pi} \approx -\frac{1}{N}\sum_{i=1}^{N}\left[\frac{\exp\left(\mathrm{adv}_g\left(s,a^{(i)}\right)\right)}{Z(s)}\log\pi\left(a^{(i)}|s\right)\right],\ a^{(i)}\sim\pi_{\mathrm{prior}}(s)$$

Desiderata Revisited

- ❑ Combat with model errors
  - ❑ One-step unrolling limits model expansion depth up to 1
  - ❑ Behavior regularization ensures policy not go far from data coverage



MCTS Ours

- ❑ Computational efficiency
  - ❑ No need for expensive MCTS
  - ❑ Sampling is available when action space is too large

$$\ell_{\text{os}}^{\pi} \approx -\frac{1}{N}\sum_{i=1}^{N}\left[\frac{\exp\left(\text{adv}_g\left(s,a^{(i)}\right)\right)}{Z(s)}\log\pi\left(a^{(i)}|s\right)\right],\ \ a^{(i)} \sim \pi_{\text{prior}}\left(s\right)$$

- ❑ ROSMO: a Regularized One-Step Model-based algorithm for Offline reinforcement learning

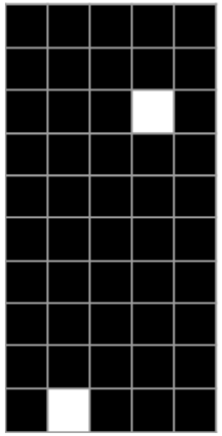
Experiment Settings

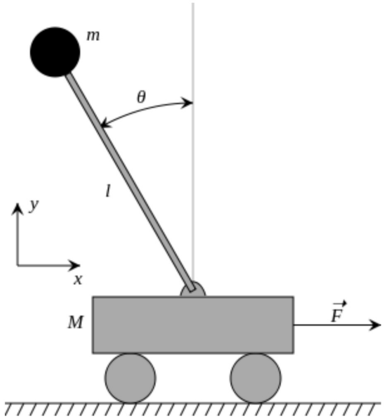
- Training is on the offline datasets
    - Usually, pre-collected and stored as static data
    - Small-scale **BSuite** dataset for detailed analysis and comparison
    - Large-scale **Atari** dataset for benchmarking

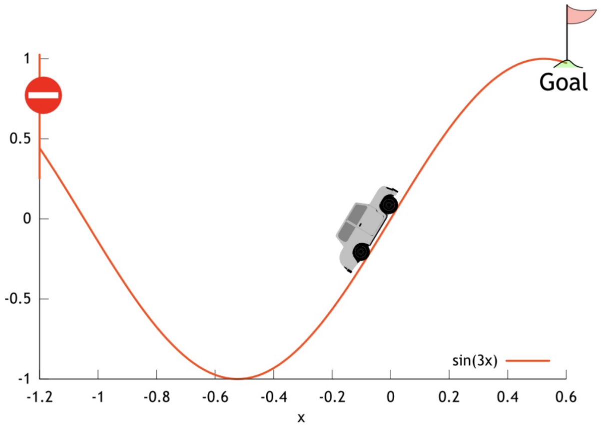
- Evaluation is done by testing the learned agent in corresponding online environments
    - IQM normalized score (Agarwal, 2021) is adopted as a robust performance measure

Small-scale BSuite Dataset

- We validate our method and compare it with MuZero (which bases on MCTS) on the BSuite dataset
  - Lightweight, low-dimension, less demanding for neural network representation learning
  - Containing environments: catch, cartpole, mountain car



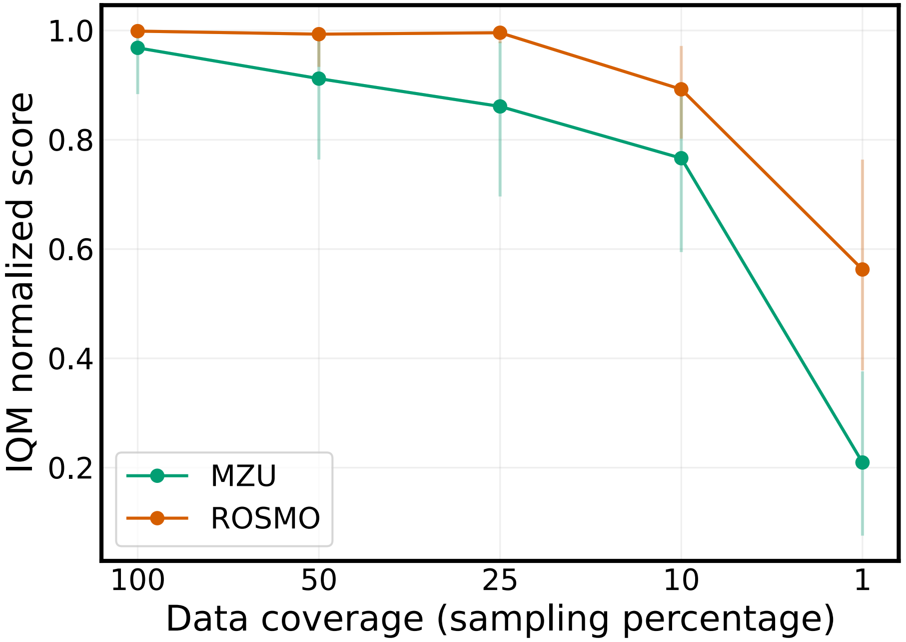


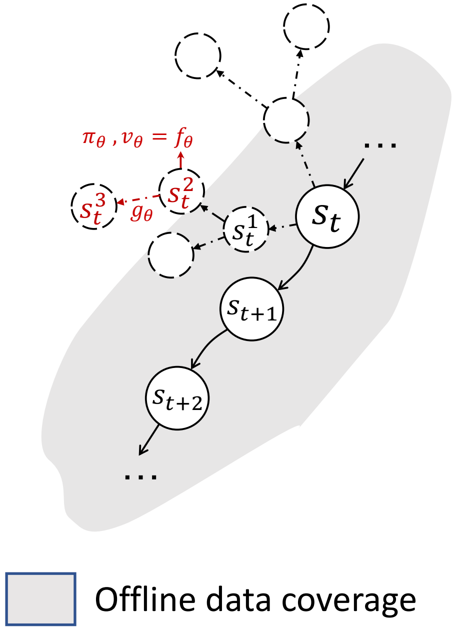


Small-scale BSuite Dataset

- We validate our method and compare it with MuZero (which bases on MCTS) on the BSuite dataset

- Observation 1: Limited data coverage shrinks the safe region, leading to erroneous estimations as searching outside the region

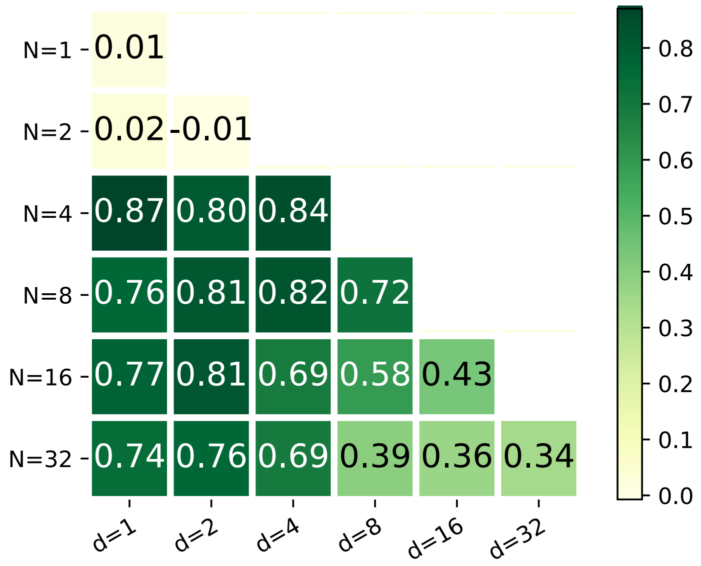




Small-scale BSuite Dataset

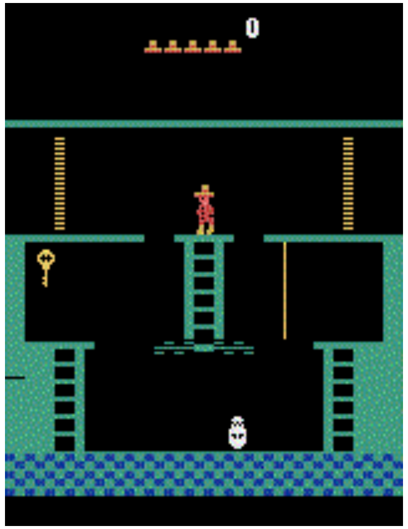
- ❑ We validate our method and compare it with MuZero (which bases on MCTS) on the BSuite dataset

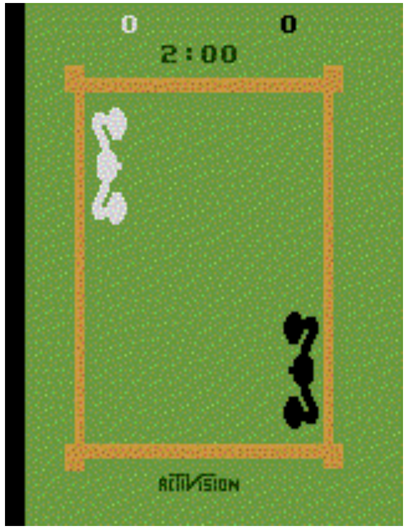
- ❑ Observation 2: MCTS is sensitive to simulation budget and search depth

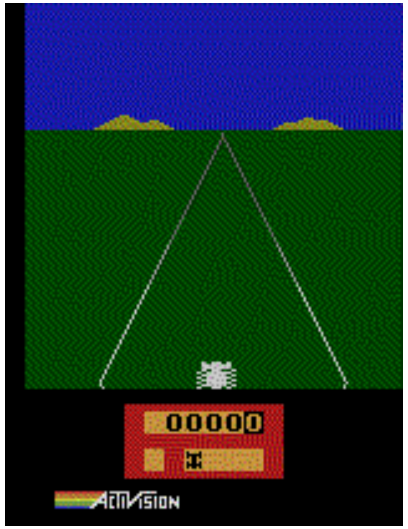


Large-scale Atari Benchmark

- We benchmark our algorithm (ROSMO) with prior offline RL methods on the offline Atari benchmark
  - Pixel-based, challenging for perception
  - A set of games where players use joystick to control the agent to solve for different tasks
  - Widely used by the community

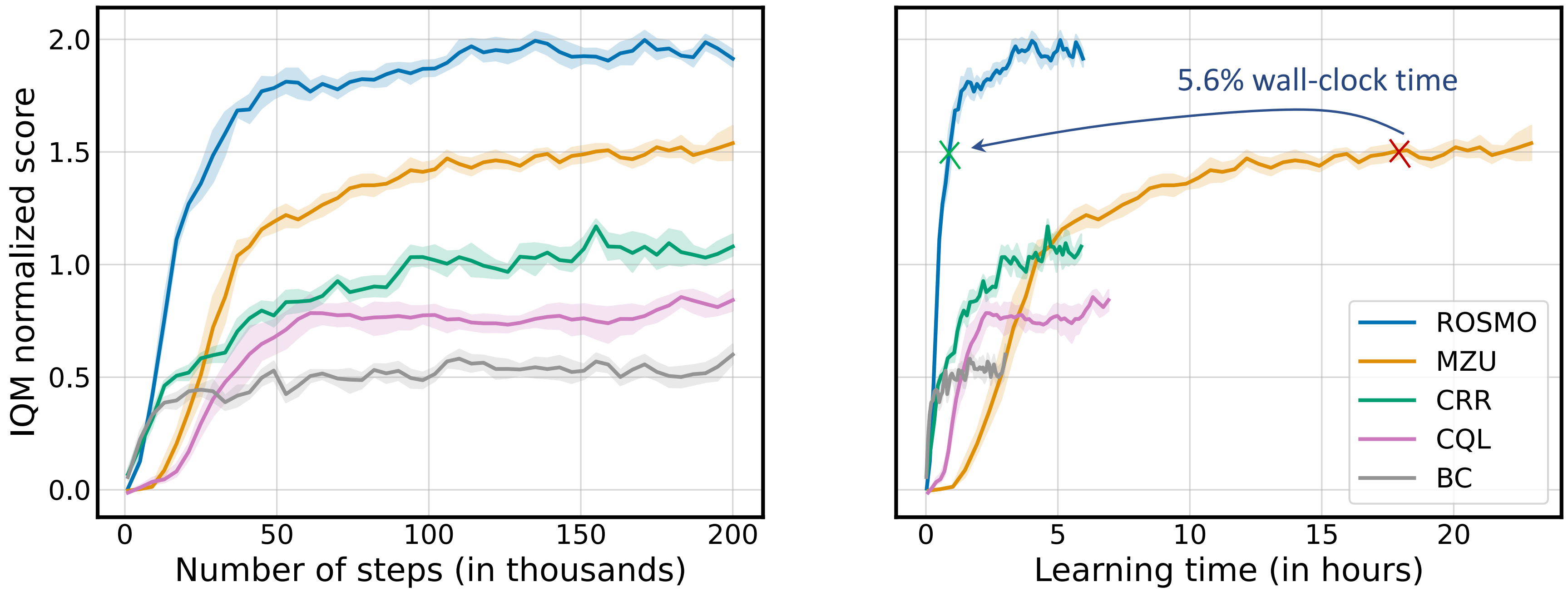






Large-scale Atari Benchmark

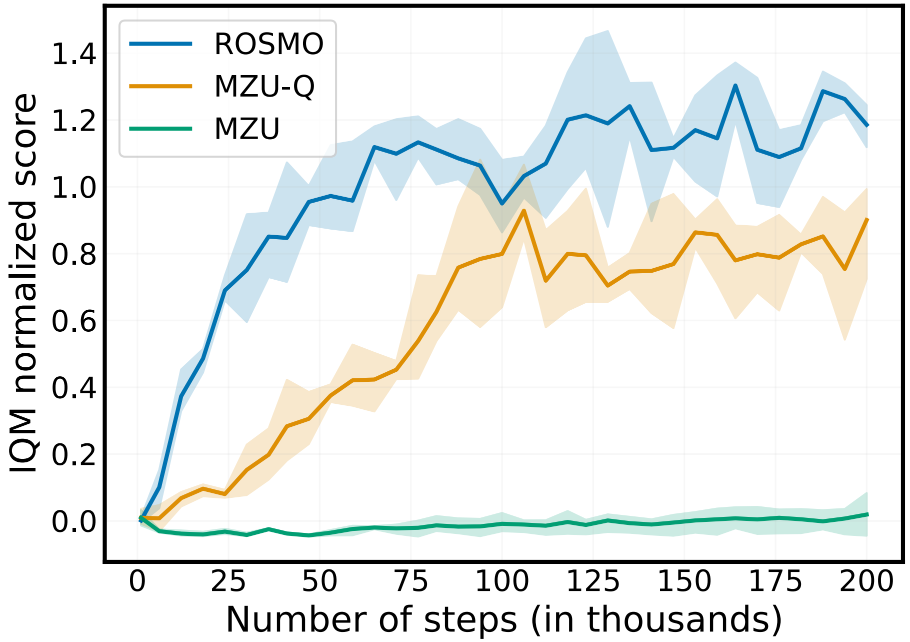
- We benchmark our algorithm (ROSMO) with prior offline RL methods on the offline Atari benchmark
- Main result: Ours achieves the best final performance as well as learning efficiency



Large-scale Atari Benchmark

- We benchmark our algorithm (ROSMO) with prior offline RL methods on the offline Atari benchmark

- Ablation 1: Performance on limited simulation budget (N=4)



$$\pi^{\text{ROSMO}} = \frac{\pi_{\text{prior}}(a|s)\exp\big(\text{adv}_g(s,a)\big)}{Z(s)}$$

$$\pi^{\text{MZU-Q}} \propto \pi_\theta \cdot \exp\left(Q^{\text{MCTS}}/\tau\right)$$ (Grill, 2020)

$$\pi^{\text{MZU}} = \frac{n(s,a)}{\sum_b n(s,b)}$$

Large-scale Atari Benchmark

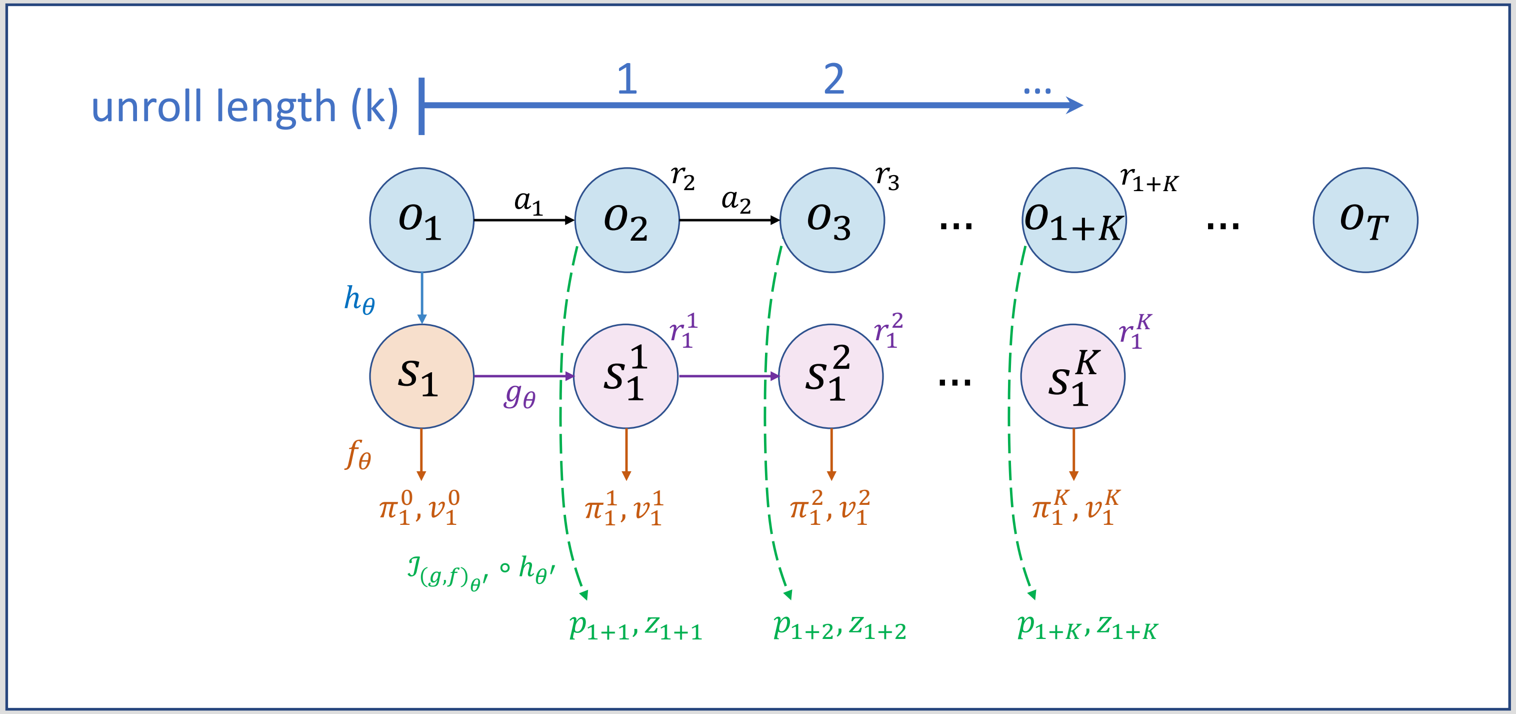
- We benchmark our algorithm (ROSMO) with prior offline RL methods on the offline Atari benchmark

- Ablation 2: Robustness on the training with different unroll lengths

Large-scale Atari Benchmark

- We benchmark our algorithm (ROSMO) with prior offline RL methods on the offline Atari benchmark

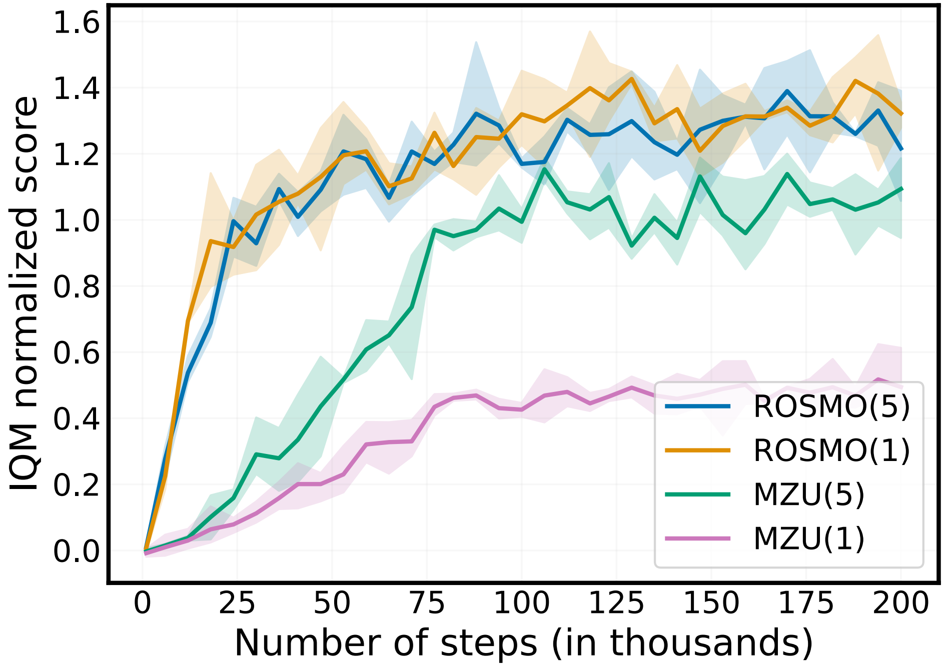
- Ablation 2: Robustness on the training with different unroll lengths



Large-scale Atari Benchmark

- We benchmark our algorithm (ROSMO) with prior offline RL methods on the offline Atari benchmark

- Ablation 2: Robustness on the training with different unroll lengths



Summary

- In this work, we
  - Scrutinize the MCTS, which is the core of SoTA MuZero (Schrittwieser, 2021) algorithm, in offline RL settings
  - Propose a simple, efficient yet strong agent that is more robust and achieves new SoTA
  - Open source the research codes: https://github.com/sail-sg/rosmo

References

- Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *nature*, *529*(7587), 484-489.
- Berner, C., Brockman, G., Chan, B., Cheung, V., Dębiak, P., Dennison, C., ... & Zhang, S. (2019). Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*.
- Liu, Z., Li, S., Lee, W. S., Yan, S., & Xu, Z. (2023). Efficient Offline Policy Optimization with a Learned Model. *ICLR*.
- Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., ... & Silver, D. (2020). Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, *588*(7839), 604-609.
- Hessel, M., Danihelka, I., Viola, F., Guez, A., Schmitt, S., Sifre, L., ... & Van Hasselt, H. (2021, July). Muesli: Combining improvements in policy optimization. In *International conference on machine learning* (pp. 4214-4226). PMLR.
- Schrittwieser, J., Hubert, T., Mandhane, A., Barekatain, M., Antonoglou, I., & Silver, D. (2021). Online and offline reinforcement learning by planning with a learned model. *Advances in Neural Information Processing Systems*, *34*, 27580-27591.
- Grill, J. B., Altché, F., Tang, Y., Hubert, T., Valko, M., Antonoglou, I., & Munos, R. (2020, November). Monte-Carlo tree search as regularized policy optimization. In International Conference on Machine Learning (pp. 3769-3778). PMLR.
- Janner, M., Fu, J., Zhang, M., & Levine, S. (2019). When to trust your model: Model-based policy optimization. Advances in neural information processing systems, 32.
- Levine, S., Kumar, A., Tucker, G., & Fu, J. (2020). Offline reinforcement learning: Tutorial, review, and perspectives on open problems. arXiv preprint arXiv:2005.01643.
- Agarwal, R., Schwarzer, M., Castro, P. S., Courville, A. C., & Bellemare, M. (2021). Deep reinforcement learning at the edge of the statistical precipice. Advances in neural information processing systems, 34, 29304-29320.