

Bidirectional Model-based Policy Optimization

Hang Lai, Jian Shen, Weinan Zhang, Yong Yu
Shanghai Jiao Tong University

Content

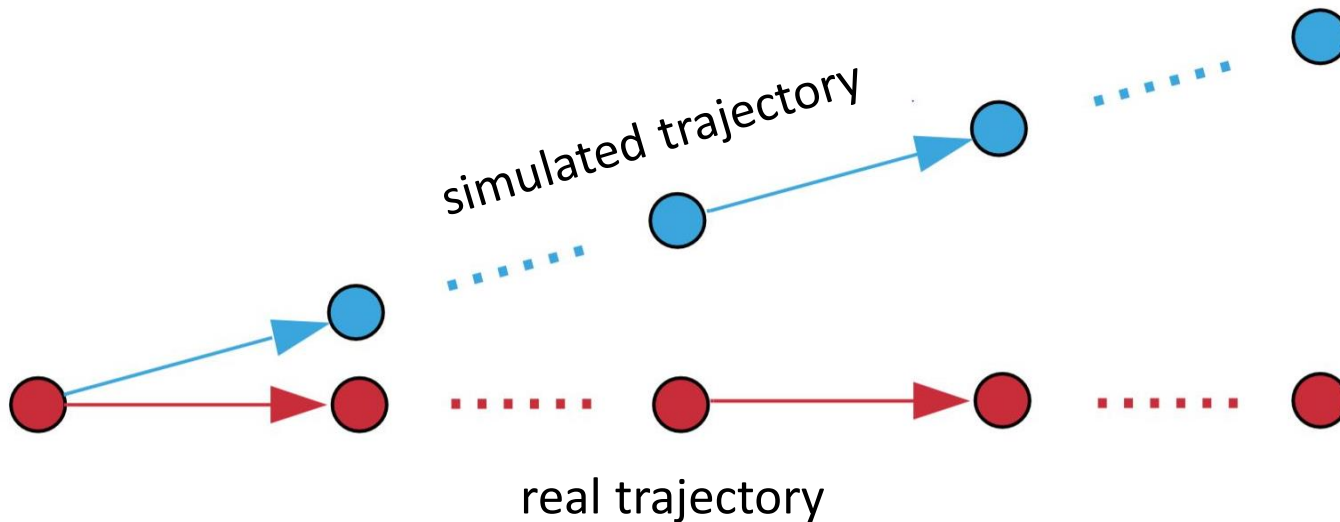
1. Background & Motivation
2. Method
3. Theoretical Analysis
4. Experiment Result

Background

Model-based reinforcement learning (MBRL):

- Build a model
- To help decision making

Challenge: compounding error



Motivation

Human beings in real world:

- Predict future consequences forward
- Imagine traces leading to a goal backward

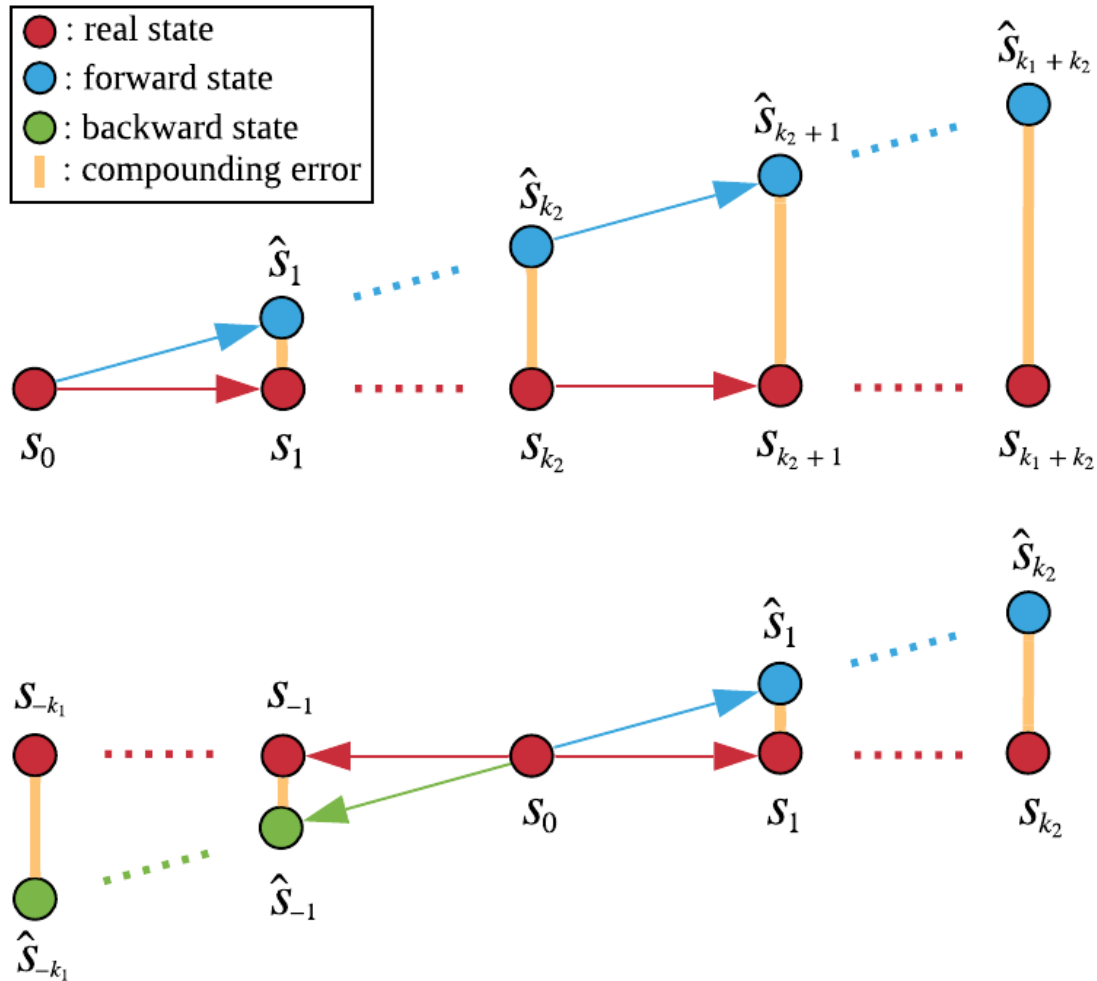
Existing methods:

- Learn a forward model $p(s'|s, a)$ to plan ahead.

This paper:

- Additionally learn a backward model $q(s|s', a)$ to reduce the reliance on accuracy in forward model.

Motivation



Content

1. Background & Motivation
2. Method
3. Theoretical Analysis
4. Experiment Result

Method

bidirectional models + MBPO[1] method




Bidirectional Model-based Policy Optimization (BMPO)

Other components:

- State sampling strategy
- Incorporating model predictive control

Preliminary: MBPO

- Interaction with environment with current policy.
 - Train forward model ensembles using real data.
 - Generate branched short rollouts with current policy.
 - Improve the policy with real & generated data.
- 

Model Learning

- Use an ensemble of probabilistic networks for both the forward model $p_{\theta}(s'|s, a)$ and the backward model $q_{\theta'}(s|s', a)$.
- The corresponding loss functions are:

$$\mathcal{L}_f(\theta) = \sum_{t=1}^N [\mu_{\theta}(s_t, a_t) - s_{t+1}]^{\top} \Sigma_{\theta}^{-1}(s_t, a_t) [\mu_{\theta}(s_t, a_t) - s_{t+1}] + \log \det \Sigma_{\theta}(s_t, a_t)$$

$$\mathcal{L}_b(\theta') = \sum_{t=1}^N [\mu_{\theta'}(s_{t+1}, a_t) - s_t]^{\top} \Sigma_{\theta'}^{-1}(s_{t+1}, a_t) [\mu_{\theta'}(s_{t+1}, a_t) - s_t] + \log \det \Sigma_{\theta'}(s_{t+1}, a_t)$$

μ and Σ : mean and covariance

N : number of real transitions

Backward Policy

Backward policy $\tilde{\pi}_{\phi'}(a|s')$: take actions given the next state. Used to generate backward policy.

- By maximum likelihood estimation:

$$\mathcal{L}_{MLE}(\phi') = - \sum_{t=0}^N \log \tilde{\pi}_{\phi'}(a_t | s_{t+1})$$

- By conditional GAN:

$$\min_{\tilde{\pi}} \max_D V(D, \tilde{\pi}) = \mathbb{E}_{(a, s') \sim \pi} [\log D(a, s')] + \mathbb{E}_{s' \sim \pi} [\log (1 - D(\tilde{\pi}(\cdot | s'), s'))]$$

State Sampling Strategy

MBPO: randomly select states from environment data buffer to begin model rollouts.

BMPO(ours): sample high value states according to the probability calculated by:

$$p(s) \propto e^{\beta V(s)}$$

Probability of s to selected Estimated value

Environment Interaction

MBPO: directly use the current policy

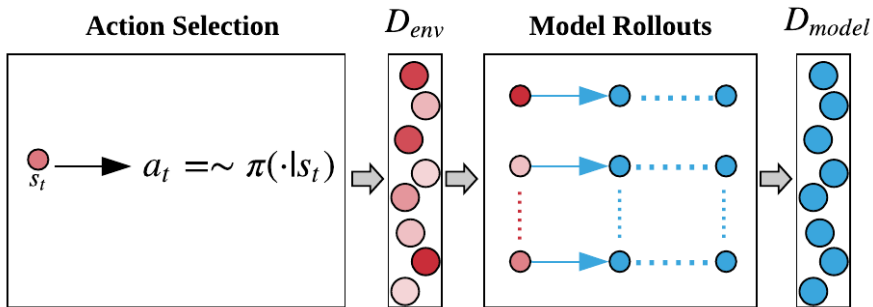
BMPO: use MPC

- Generate N candidate action sequences from current policy.
- Simulate the corresponding trajectories in the model.
- Select the first action of the sequence that yields the highest return.

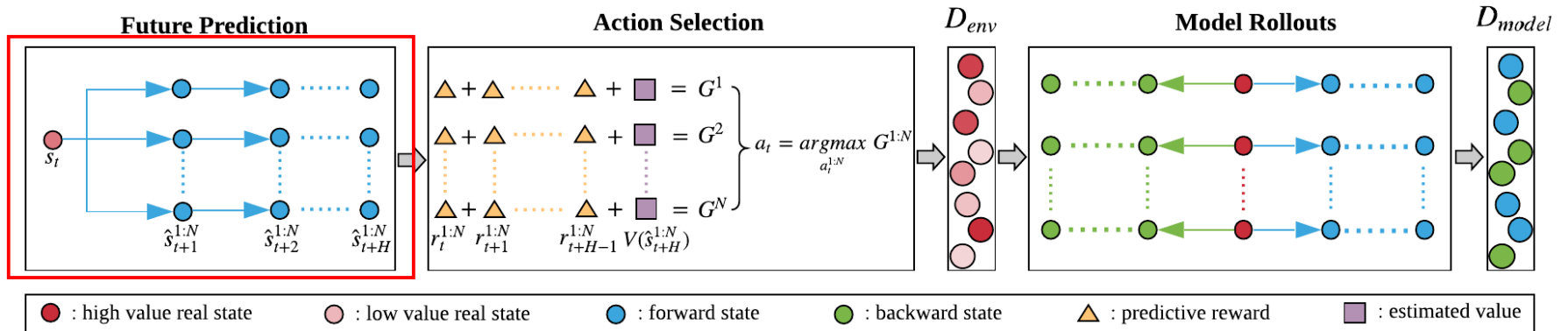
$$a_t = \operatorname{argmax}_{a_t^{1:N} \sim \pi} \sum_{t'=t}^{t+H-1} \gamma^{t'-t} r(s_{t'}, a_{t'}) + \gamma^H V(s_{t+H})$$

Overall algorithm

MBPO:

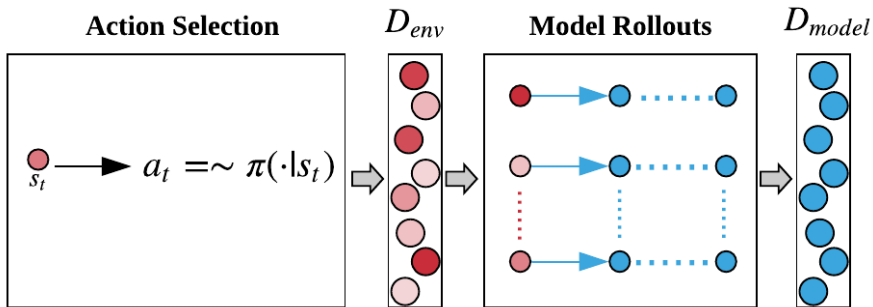


BMPO (ours):

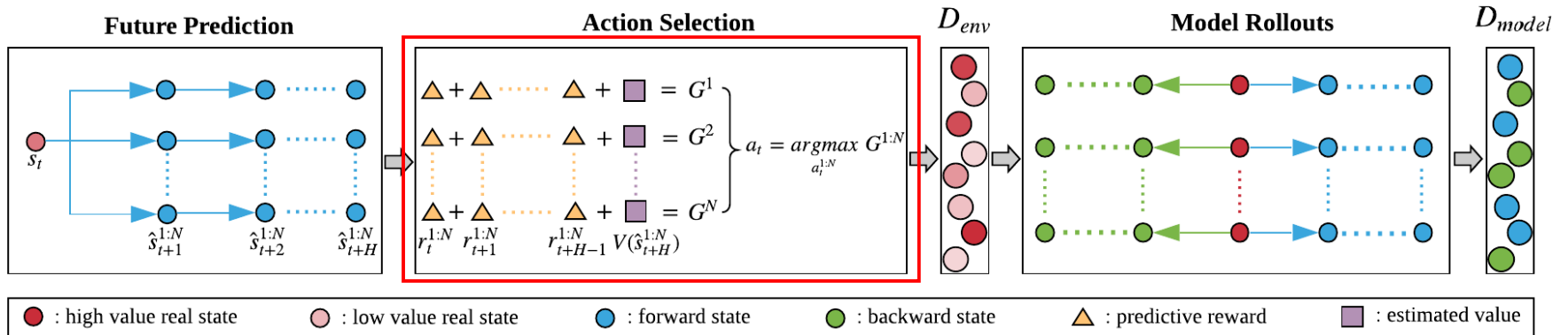


Overall algorithm

MBPO:

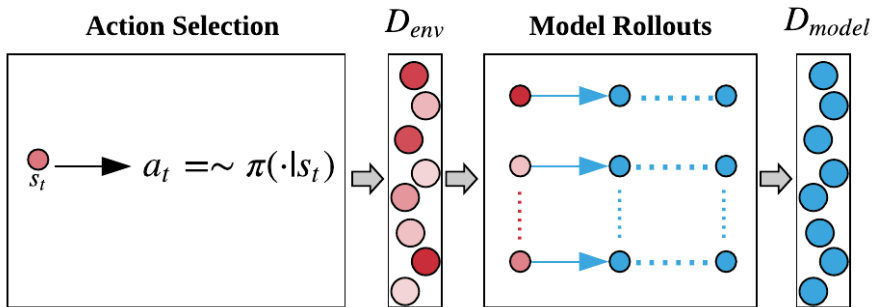


BMPO (ours):

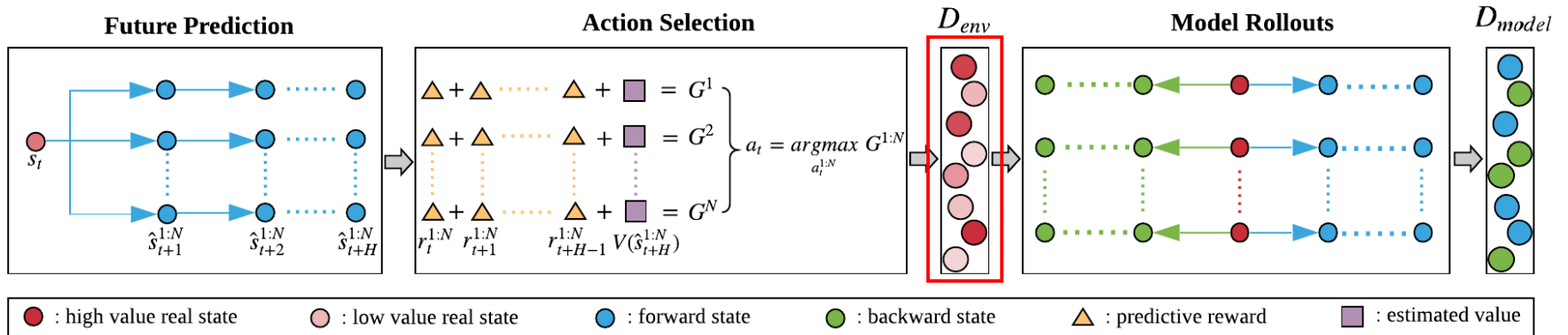


Overall algorithm

MBPO:

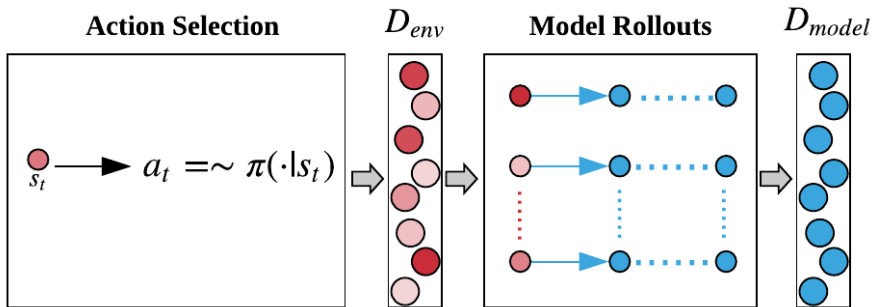


BMPO (ours):

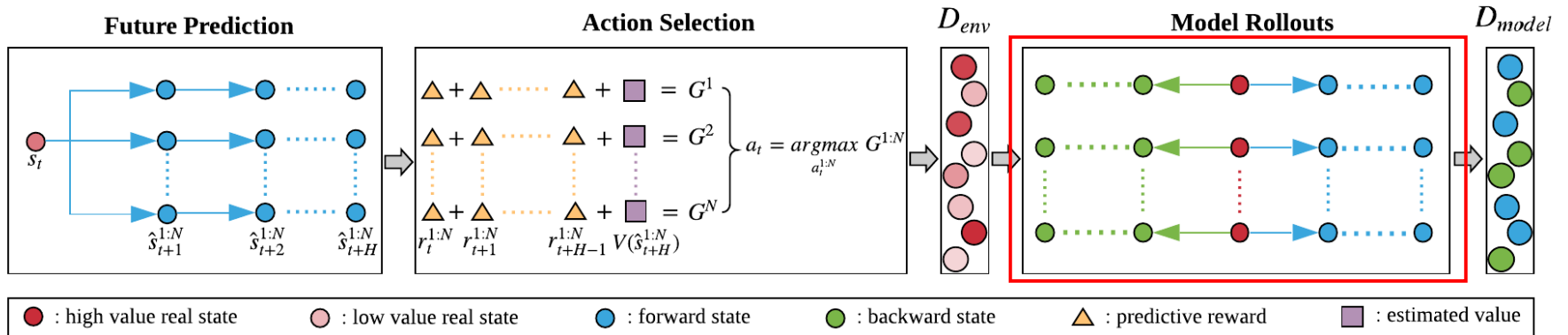


Overall algorithm

MBPO:

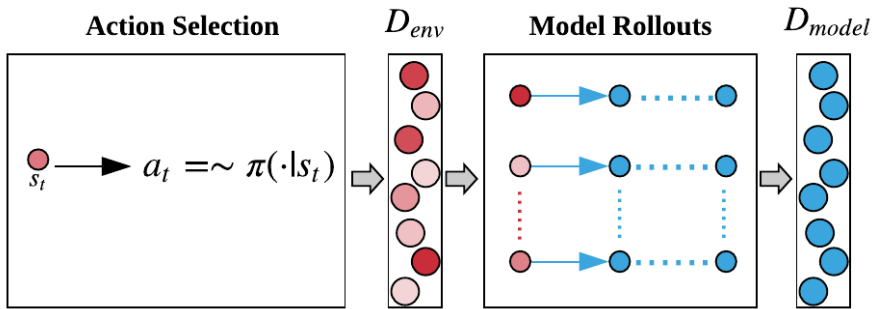


BMPO (ours):

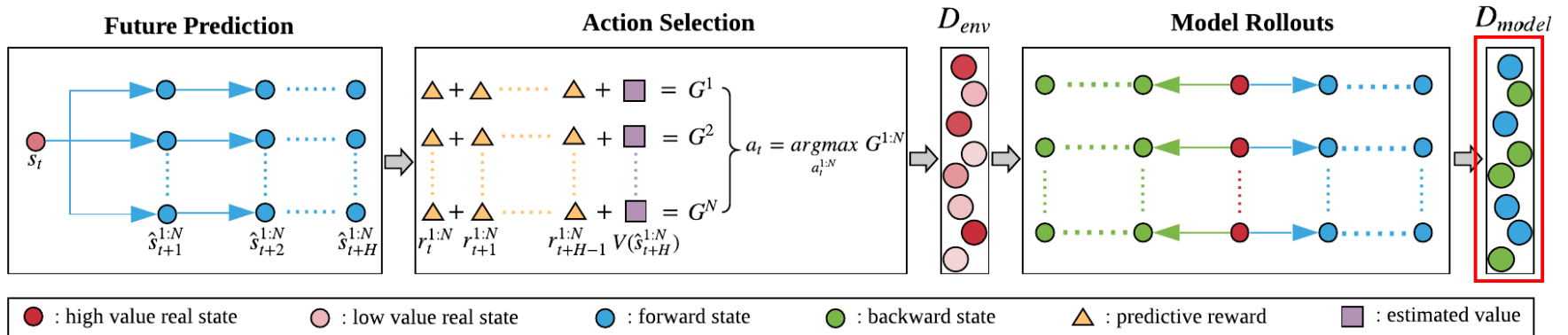


Overall algorithm

MBPO:



BMPO (ours):



Content

1. Motivation
2. Method
3. Theoretical Analysis
4. Experiment Result

Theoretical Analysis

For bidirectional model with backward rollout length k_1 and forward length k_2 :

Expected return in environment

Policy shift variation

Expected return in branched rollout

Model error

$$\left| \eta[\pi] - \eta^{\text{branch}}[\pi] \right| \leq 2r_{\max} \left[\frac{\gamma^{k_1+k_2+1} \epsilon_{\pi}}{(1-\gamma)^2} + \frac{\gamma^{k_1+k_2} \epsilon_{\pi}}{(1-\gamma)} + \frac{\max(k_1, k_2) \epsilon_m}{1-\gamma} \right]$$

For the forward only model with rollout length $k_1 + k_2$:

$$\left| \eta[\pi] - \eta^{\text{branch}}[\pi] \right| \leq 2r_{\max} \left[\frac{\gamma^{k_1+k_2+1} \epsilon_{\pi}}{(1-\gamma)^2} + \frac{\gamma^{k_1+k_2} \epsilon_{\pi}}{(1-\gamma)} + \frac{(k_1+k_2) \epsilon_m}{1-\gamma} \right]$$

Theoretical Analysis

For bidirectional model with backward rollout length k_1 and forward length k_2 :

Expected return in environment

Policy shift variation

Expected return in branched rollout

Model error

$$|\eta[\pi] - \eta^{\text{branch}}[\pi]| \leq 2r_{\max} \left[\frac{\gamma^{k_1+k_2+1} \epsilon_{\pi}}{(1-\gamma)^2} + \frac{\gamma^{k_1+k_2} \epsilon_{\pi}}{(1-\gamma)} + \frac{\max(k_1, k_2) \epsilon_m}{1-\gamma} \right]$$

For the forward only model with rollout length $k_1 + k_2$:

$$|\eta[\pi] - \eta^{\text{branch}}[\pi]| \leq 2r_{\max} \left[\frac{\gamma^{k_1+k_2+1} \epsilon_{\pi}}{(1-\gamma)^2} + \frac{\gamma^{k_1+k_2} \epsilon_{\pi}}{(1-\gamma)} + \frac{(k_1+k_2) \epsilon_m}{1-\gamma} \right]$$

Content

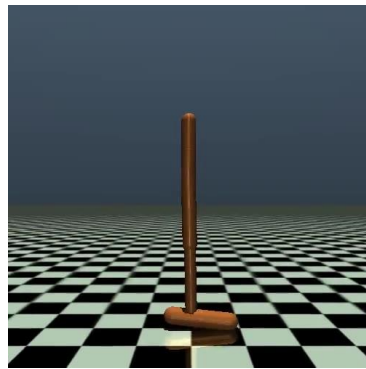
1. Motivation
2. Method
3. Theoretical Analysis
4. Experiment Result

Settings

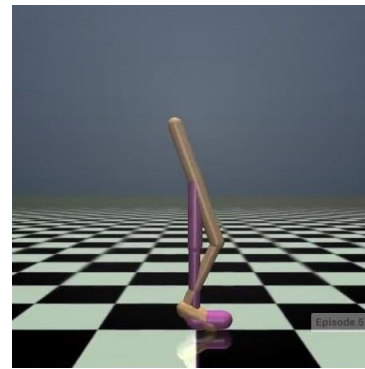
- Environments



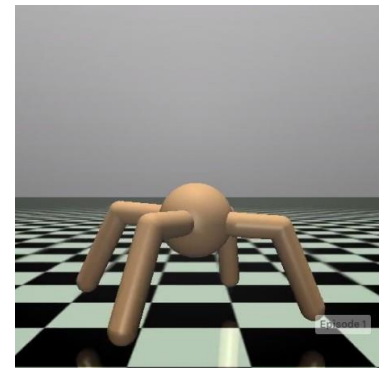
Pendulum



Hopper



Walker

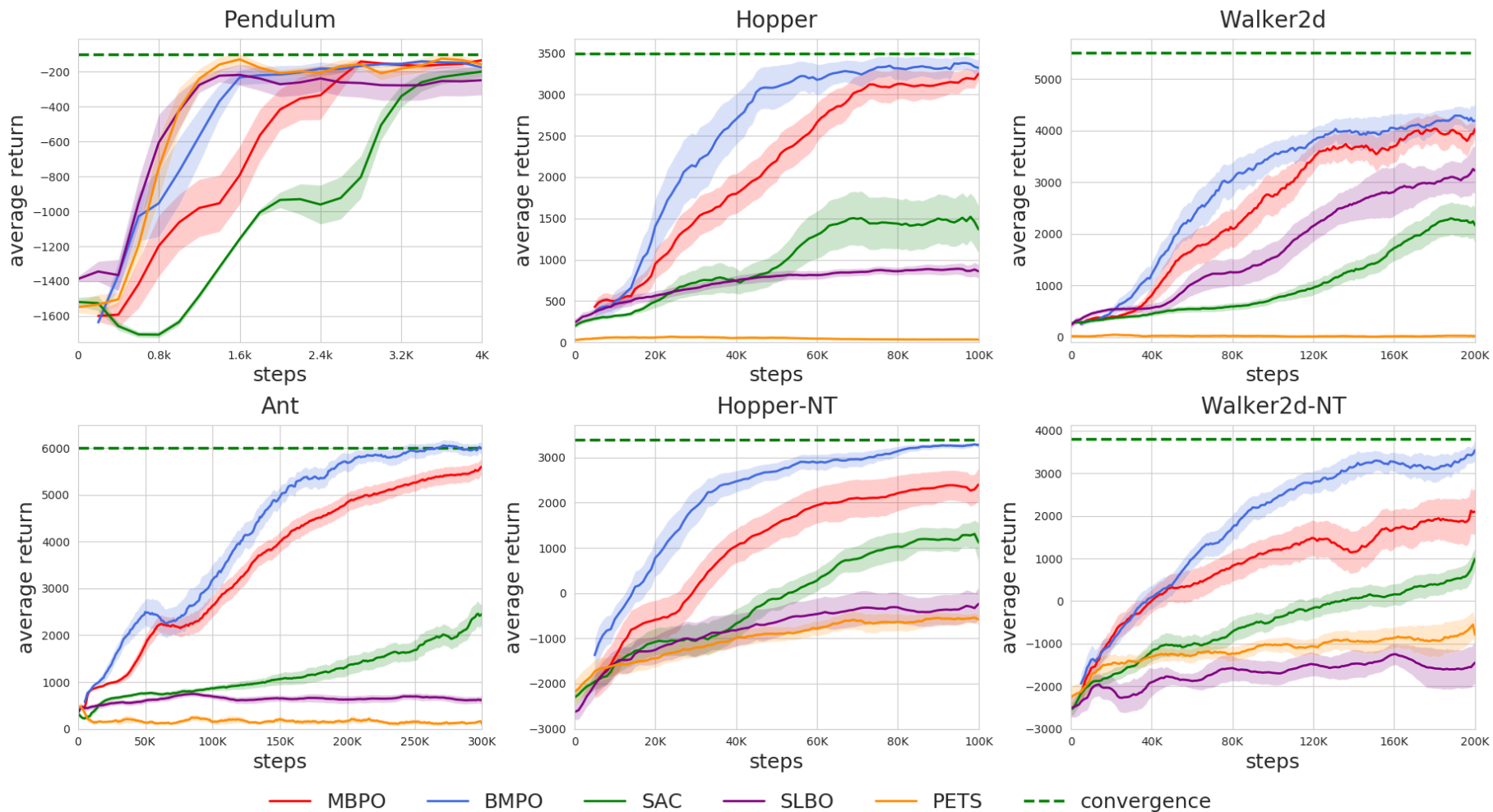


Ant

- Baselines

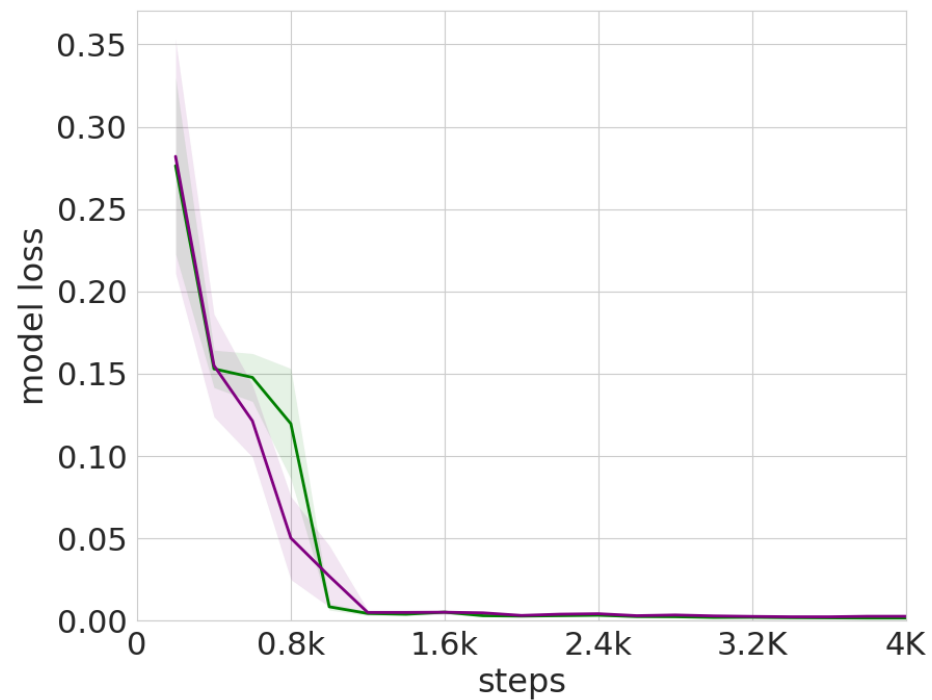
MBPO, SLBO[2], PETS[3], SAC[4]

Comparison Result

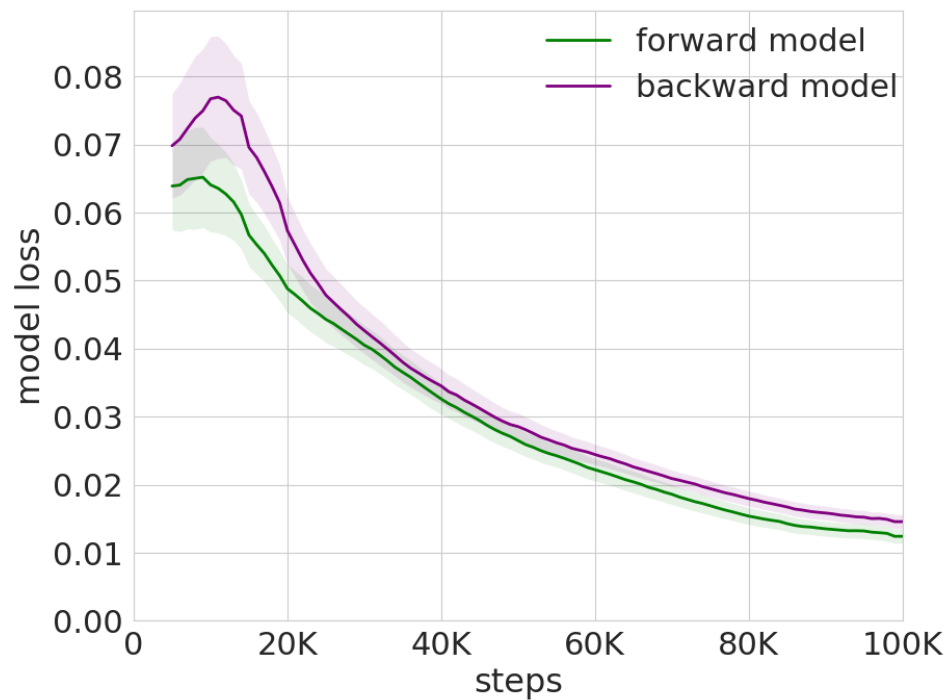


Model Error

Pendulum



Hopper-NT



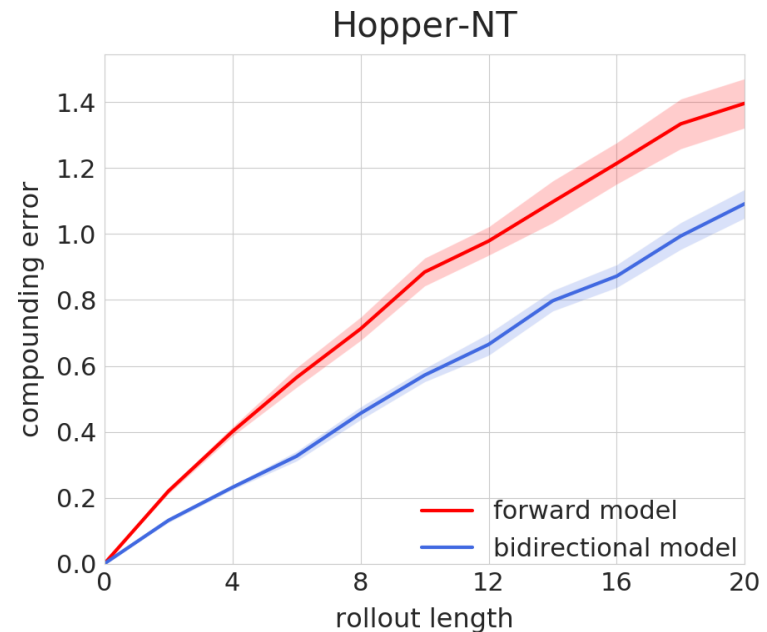
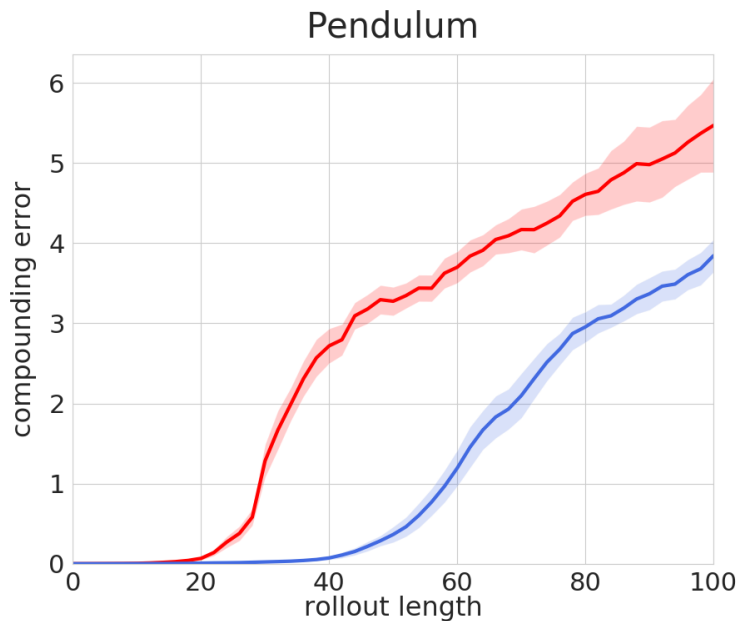
Model validation loss(single step error)

Compounding Model Error

Assume a real trajectory of length $2h$ is $(s_0, a_0, s_1, \dots, s_{2h})$.

$$\text{Error}_{for} = \frac{1}{2h} \sum_{i=1}^{2h} \|\hat{s}_i - s_i\|_2^2$$

$$\text{Error}_{bi} = \frac{1}{2h} \sum_{i=1}^h \left(\|\hat{s}_{h+i} - s_{h+i}\|_2^2 + \|\hat{s}_{h-i} - s_{h-i}\|_2^2 \right)$$



Backward Policy Choice

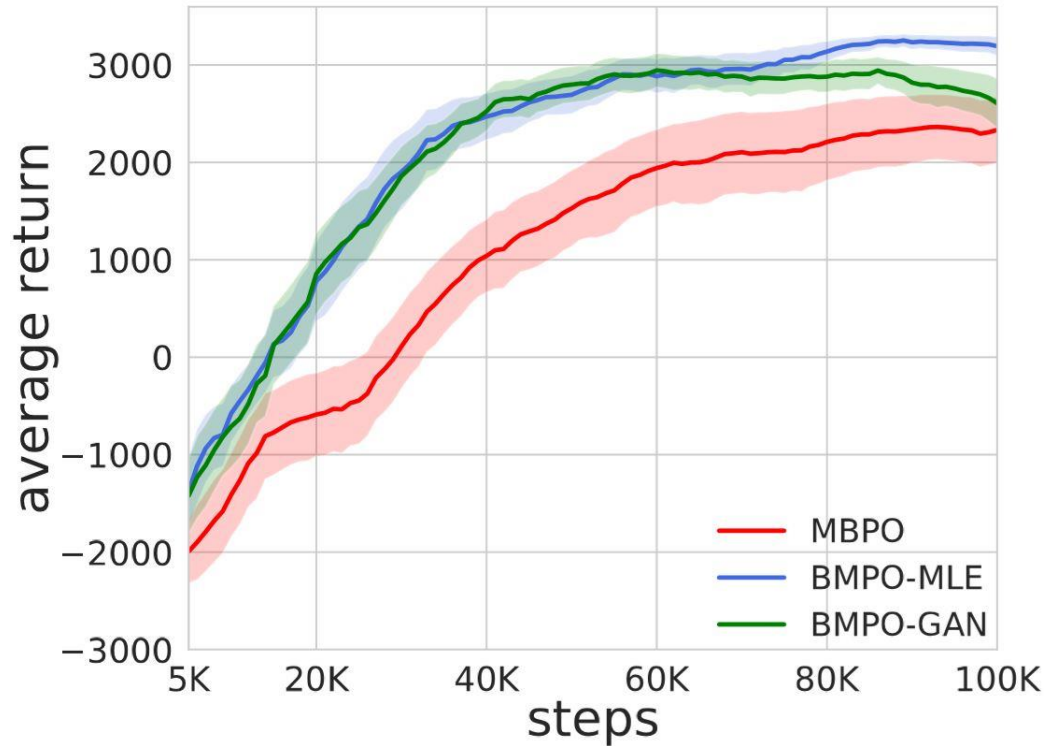


Figure 1): Comparison of two heuristic design choices for the backward policy loss: MLE loss and GAN loss.

Ablation study

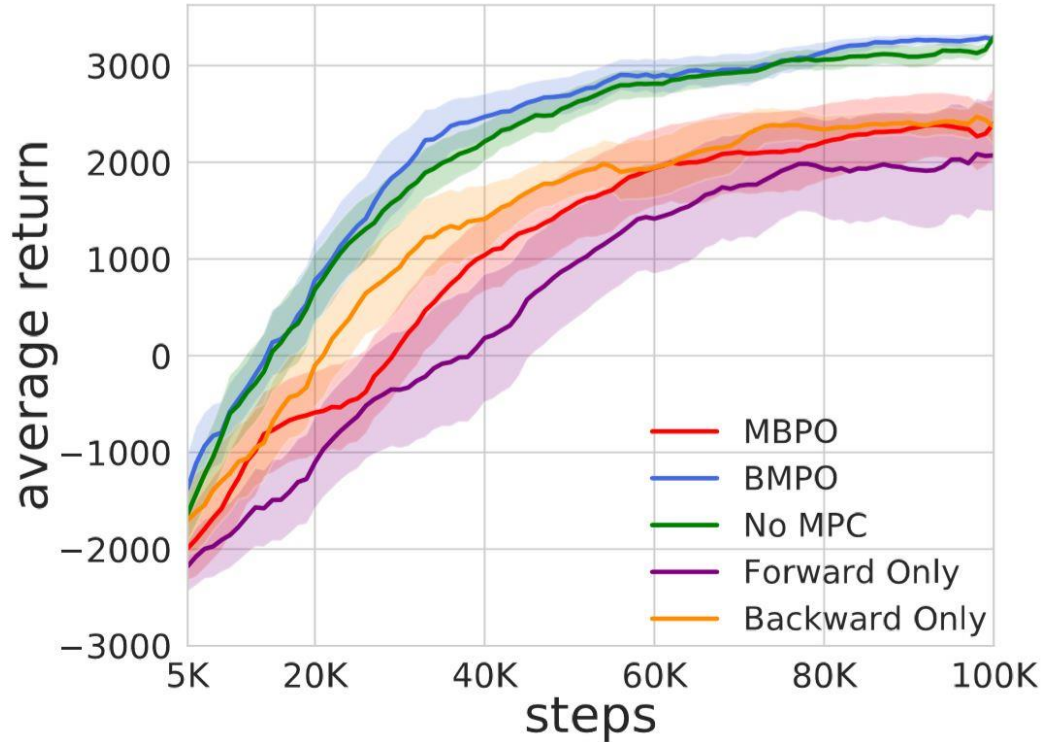


Figure 2): Ablation study of three crucial components: forward model, backward model, and MPC.

Hyperparameter study: β

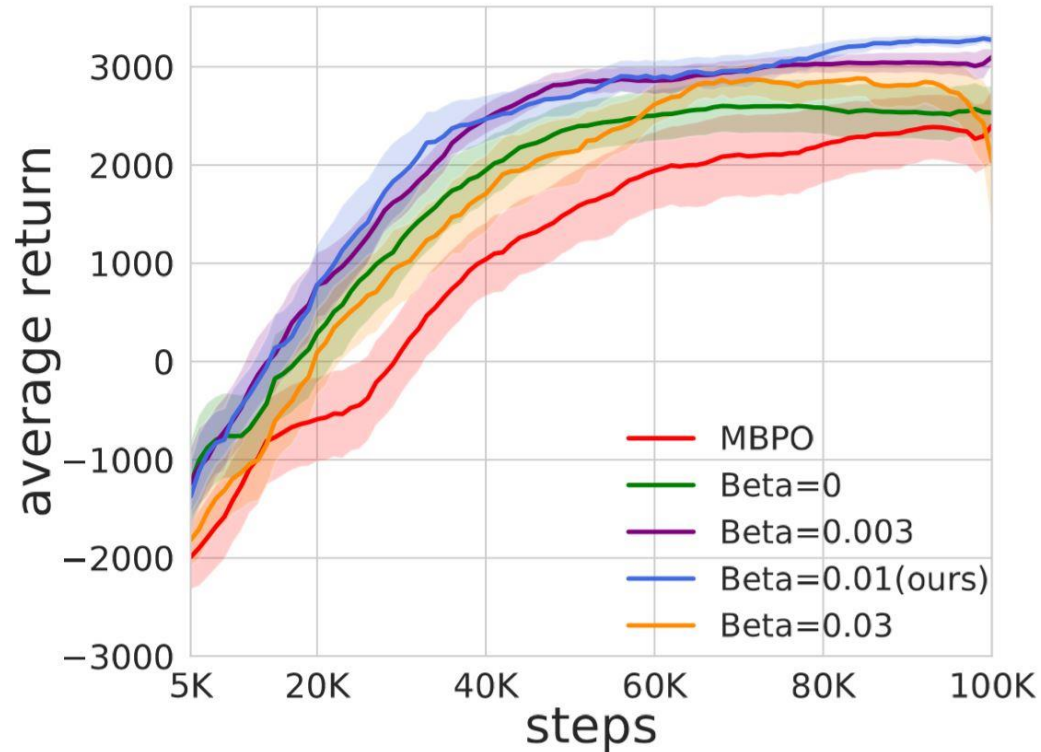


Figure 3): the sensitivity of our algorithm to the hyper-parameter β .

Hyperparameter study: k_1

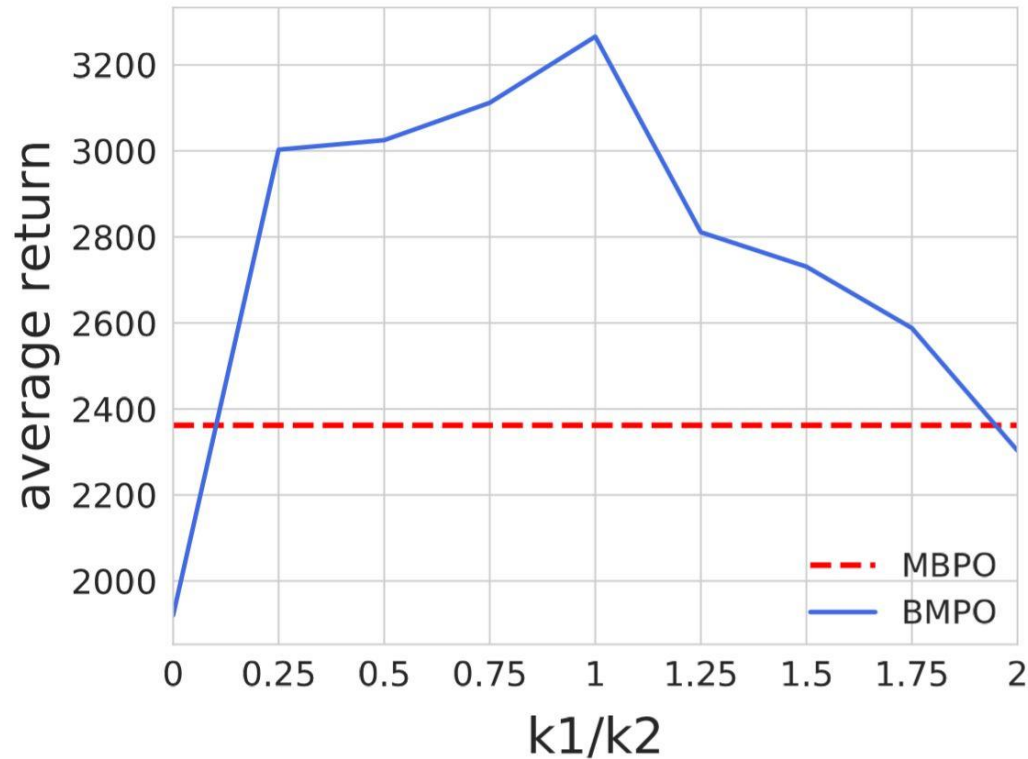


Figure 4): Average return with different backward rollout lengths k_1 and fixed forward length k_2 .

Reference

- [1] Janner, Michael, et al. "When to trust your model: Model-based policy optimization." *Advances in Neural Information Processing Systems*. 2019.
- [2] Luo, Yuping, et al. "Algorithmic framework for model-based deep reinforcement learning with theoretical guarantees." *arXiv preprint arXiv:1807.03858* (2018).
- [3] Chua, Kurtland, et al. "Deep reinforcement learning in a handful of trials using probabilistic dynamics models." *Advances in Neural Information Processing Systems*. 2018.
- [4] Haarnoja, Tuomas, et al. "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor." *arXiv preprint arXiv:1801.01290* (2018).

Thanks for your interest!

Please feel free to contact me at
laihang@apex.sjtu.edu.cn