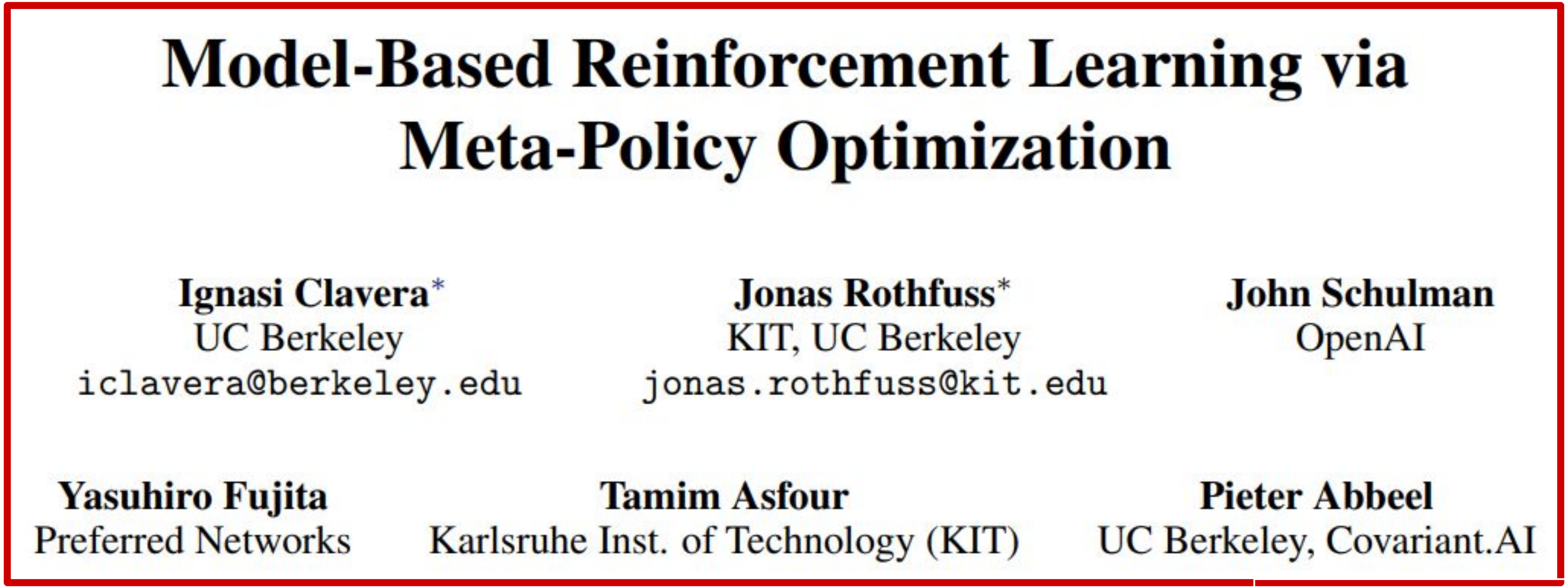


Model-Based Reinforcement Learning via Meta-Policy Optimization

Ignasi Clavera[*]
UC Berkeley
iclavera@berkeley.edu

Jonas Rothfuss[*]
KIT, UC Berkeley
jonas.rothfuss@kit.edu

John Schulman
OpenAI

Yasuhiro Fujita
Preferred Networks

Tamim Asfour
Karlsruhe Inst. of Technology (KIT)

Pieter Abbeel
UC Berkeley, Covariant.AI

arxiv:1809.05214

Presented by Elliot Creager for CSC2621
February 18, 2020

Overview

- Problem statement: *model bias* in MB-RL
- Contributions of the paper
- Background
  - Model bias
  - Meta Learning
- Proposed solution: MB-MPO
- Experiments & results
- Discussion: limitations and open issues

Motivation

Model-based RL is sample-efficient *assuming a good model of the environment*

- A dynamics model can (a) provide training trajectories for policy learning

  (b) provide gradient information in control

But *“accurate dynamics models can often be far more complex than good policies”*

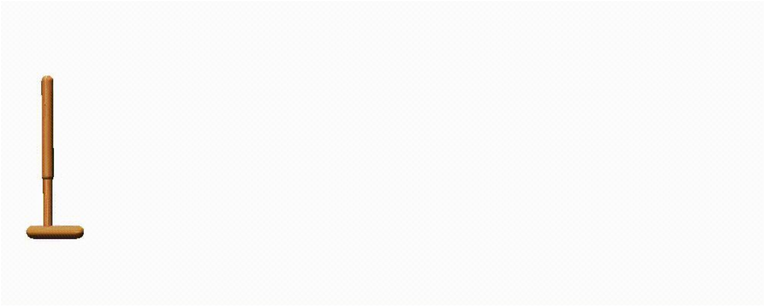
- E.g. Pouring water into cup.
- Policy (state → action mapping) description is simpler than physics required for next-state prediction (state, action → next state mapping)

Motivation

What if the model is wrong?

Any errors in the dynamics model propagate to policy learning (*model bias*)

Model-based RL must account for *uncertainty* of model fit



[https://bair.berkeley.edu/blog/2019/12/12/mbpo/]

Mod *nt*

-

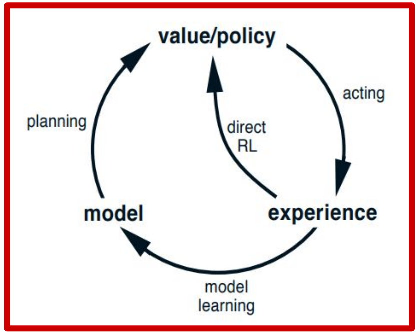
But “ *icies”*

-
- ate

Contributions

- Cast (single-environment) MB-RL as meta learning
    - Each model (“learner”) in the ensemble adapts to its “task”, and policy (“meta learner”) seeks best average performance across ensemble after adaptation
- Propose MB-MPO algorithm that uses MAML-style meta learner
    - Meta learner chooses *initial* parameters that yield best single-gradient-step adaptation
    - Performance gains over model-based ensembling w/o meta learning (ME-TRPO)
    - Sample efficiency gains over model-free methods

Background: MB-RL



[Sutton & Barto 2018]

RL is about learning a policy that does well in an environment

Model-based RL uses *models* of environment dynamics towards this goal

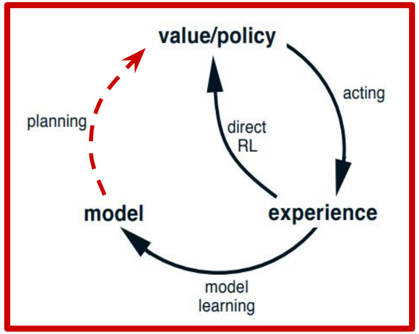
Models fit via *supervised learning* using relatively few *off-policy* trajectories

Models can be leveraged by policy learners in a variety of ways

- Random shooting: choose best next action over random trajectory rollouts
- Propagate gradients of policy parameters through trajectory rollouts
- *Sample (“imagine”) many trajectories to train policy via model-free method*
    - Focus of this paper; policy updates using 100k trajectories while models fit using 4k

Note: see Wang et al 2019 *Benchmarking MB-RL* for helpful taxonomy of model-based methods

Background: Mitigating model bias



[Sutton & Barto 2018]

“Policy optimization is prone to overfit to deficiencies of the model”

Possible approaches to mitigating model bias:

- Probabilistic model to *explicitly* capture environment variance (PILCO, PETS)
  - Limitations: density modeling is difficult.
    - GP gives good non-parametric uncertainty estimates but doesn’t scale.
    - Neural nets scale but make simplifying distributional assumptions
- Learn policy that does well on average over *ensemble* of models (ME-TRPO)
  - Limitations: each ensemble member is still free to overfit
  - In principle, environments with multimodal transitions necessitate large ensembles

Note: see Wang et al 2019 *Benchmarking MB-RL* for helpful taxonomy of model-based methods

Background: Meta learning

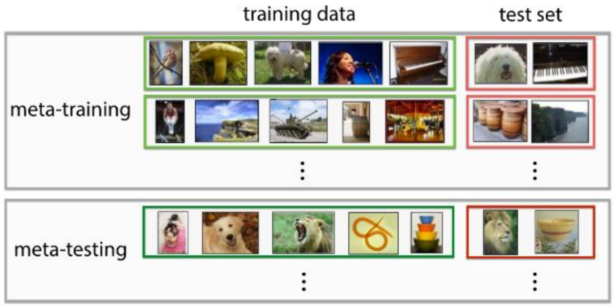
Learning seeks to *generalize* to new examples

Meta learning seeks to generalize to *new experiences/tasks*

Model-agnostic meta learner (MAML) is a popular approach for learning good *initial parameters* given a sequence of tasks

$$\max_{\boldsymbol{\theta}} \ \mathbb{E}_{\substack{\mathcal{M}_k\sim\rho(\mathcal{M})\\ \boldsymbol{s}_{t+1}\sim p_k\\ \boldsymbol{a}_t\sim\pi_{\boldsymbol{\theta}'}(\boldsymbol{a}_t|\boldsymbol{s}_t)}} \left[\sum_{t=0}^{H-1} r_k(\boldsymbol{s}_t,\boldsymbol{a}_t)\right] \quad \text{s.t.: } \boldsymbol{\theta}' = \boldsymbol{\theta} + \alpha \, \nabla_{\boldsymbol{\theta}} \mathbb{E}_{\substack{\boldsymbol{s}_{t+1}\sim p_k\\ \boldsymbol{a}_t\sim\pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{s}_t)}} \left[\sum_{t=0}^{H-1} r_k(\boldsymbol{s}_t,\boldsymbol{a}_t)\right] \quad (1)$$

[Clavera et al 2018]



Example meta-learning set-up for few-shot image classification, visual adapted from Ravi & Larochelle '17.

[https://bair.berkeley.edu/blog/2017/07/18/learning-to-learn/]

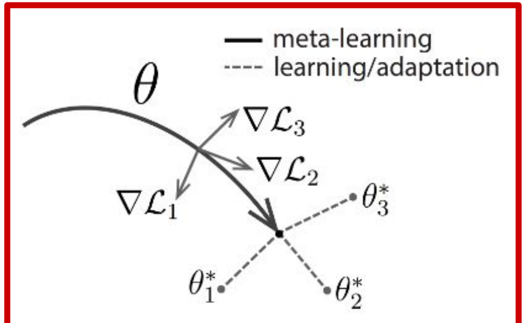


Diagram of the MAML approach.

[https://bair.berkeley.edu/blog/2017/07/18/learning-to-learn/]

Proposed method: MB-MPO

Supervised learning of model ensemble from shared off-policy trajectory buffer

$$\min_{\phi_k} \frac{1}{|\mathcal{D}_k|} \sum_{(\boldsymbol{s}_t,\boldsymbol{a}_t,\boldsymbol{s}_{t+1})\in\mathcal{D}_k} \|\boldsymbol{s}_{t+1} - \hat{f}_{\phi_k}(\boldsymbol{s}_t,\boldsymbol{a}_t)\|_2^2$$

[Clavera et al 2018]

Generate set of imagined trajectories using the model ensemble

For each model, compute average returns under current policy, and grads w.r.t. parameters

$$J_k(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{a}_t\sim\pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{s}_t)}\left[\sum_{t=0}^{H-1} r(\boldsymbol{s}_t,\boldsymbol{a}_t)\middle|\boldsymbol{s}_{t+1} = \hat{f}_{\phi_k}(\boldsymbol{s}_t,\boldsymbol{a}_t)\right]$$

[Clavera et al 2018]

For each model, take an adaptation step on own trajectories

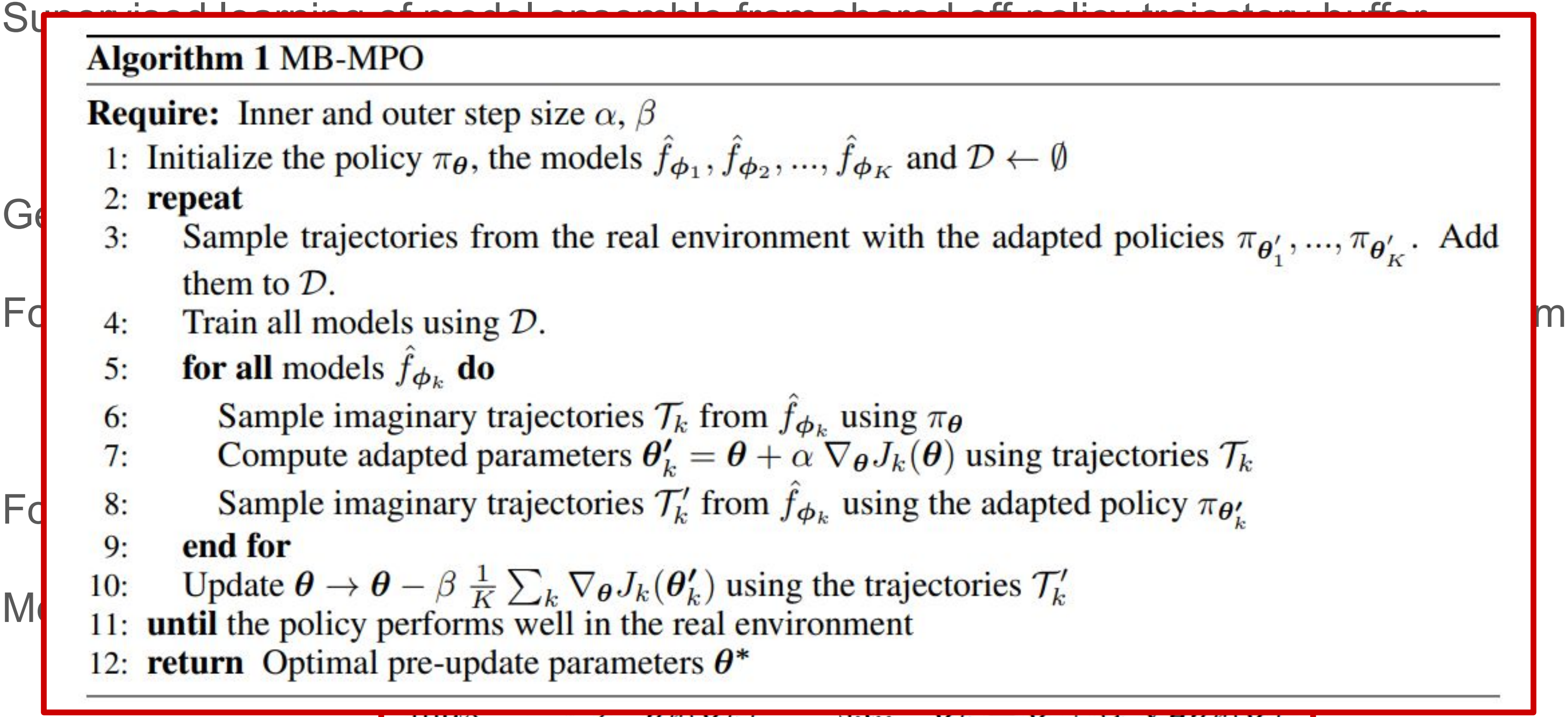
Meta-update initial policy parameters to improve average adapted returns

$$\max_{\boldsymbol{\theta}} \quad \frac{1}{K}\sum_{k=0}^{K} J_k(\boldsymbol{\theta}'_k) \qquad \text{s.t.:} \quad \boldsymbol{\theta}'_k = \boldsymbol{\theta} + \alpha \nabla_{\boldsymbol{\theta}} J_k(\boldsymbol{\theta})$$

[Clavera et al 2018]

Proposed method: MB-MPO

Supervised learning of model ensemble from shared off-policy trajectory buffer



Algorithm 1 MB-MPO

```
Require: Inner and outer step size $\alpha$, $\beta$
Initialize the policy $\pi_{\boldsymbol{\theta}}$, the models $\hat{f}_{\boldsymbol{\phi}_1}, \hat{f}_{\boldsymbol{\phi}_2}, ..., \hat{f}_{\boldsymbol{\phi}_K}$ and $\mathcal{D} \leftarrow \emptyset$
repeat
    Sample trajectories from the real environment with the adapted policies $\pi_{\boldsymbol{\theta}'_1}, ..., \pi_{\boldsymbol{\theta}'_K}$. Add them to $\mathcal{D}$.
    Train all models using $\mathcal{D}$.
    for all models $\hat{f}_{\boldsymbol{\phi}_k}$ do
        Sample imaginary trajectories $\mathcal{T}_k$ from $\hat{f}_{\boldsymbol{\phi}_k}$ using $\pi_{\boldsymbol{\theta}}$
        Compute adapted parameters $\boldsymbol{\theta}'_k = \boldsymbol{\theta} + \alpha \nabla_{\boldsymbol{\theta}} J_k(\boldsymbol{\theta})$ using trajectories $\mathcal{T}_k$
        Sample imaginary trajectories $\mathcal{T}'_k$ from $\hat{f}_{\boldsymbol{\phi}_k}$ using the adapted policy $\pi_{\boldsymbol{\theta}'_k}$
    end for
    Update $\boldsymbol{\theta} \rightarrow \boldsymbol{\theta} - \beta \frac{1}{K} \sum_k \nabla_{\boldsymbol{\theta}} J_k(\boldsymbol{\theta}'_k)$ using the trajectories $\mathcal{T}'_k$
until the policy performs well in the real environment
return Optimal pre-update parameters $\boldsymbol{\theta}^*$
```

Ge

Fo meters

Fo

Mo

$$\max_{\boldsymbol{\theta}} \quad \frac{1}{K} \sum_{k=0} J_k(\boldsymbol{\theta}'_k) \quad \text{s.t.:} \quad \boldsymbol{\theta}'_k = \boldsymbol{\theta} + \alpha \nabla_{\boldsymbol{\theta}} J_k(\boldsymbol{\theta})$$

[Clavera et al 2018]

Implementation details

Evaluated on continuous control tasks with *deterministic dynamics*

- Variance across model ensembles due to data shuffling and env. init.
- Shuffling dominates randomness since *all* models share experience.

Outer loop policy updated with TRPO, while inner loop uses vanilla policy gradient

Second-order gradients are numerically approximated, rather than using costly but exact automatic differentiation.

$$\boxed{\frac{d}{d\theta}\mathbb{E}_x[f(x,\theta)] = \mathbb{E}_x\left[\left(\frac{\partial}{\partial\theta}\log p_x(x|\theta)\right) f(x,\theta) + \frac{\partial f}{\partial\theta}\right]}$$

[REINFORCE under chain rule, Weber 2019]

Some standard tricks (reward baselines, weight norm) used to stabilize training

Experiments & Results

Six continuous control benchmark tasks from Mujoco with deterministic dynamics

Meta-learned init. params. used to report rewards (adaptation for inner loop only)

Key experiments:

- Show the inner loop adaptation meaningfully changes the policy distribution
- Show sample efficiency win over model-free algos
- Show performance win over model-based algos that account for model bias
- Show robustness to observation noise in experience buffer used to fit models

Experiments & Results

Simple synthetic setting

- Agent must move to goal in 2-D space

“Plasticity” computed as KL-divergence of pre- and post-update policy within the inner loop

- This quantity depends on the agents *current state*
- Agent shows greater plasticity when far away from goal; these states are *underrepresented* in the experience buffer

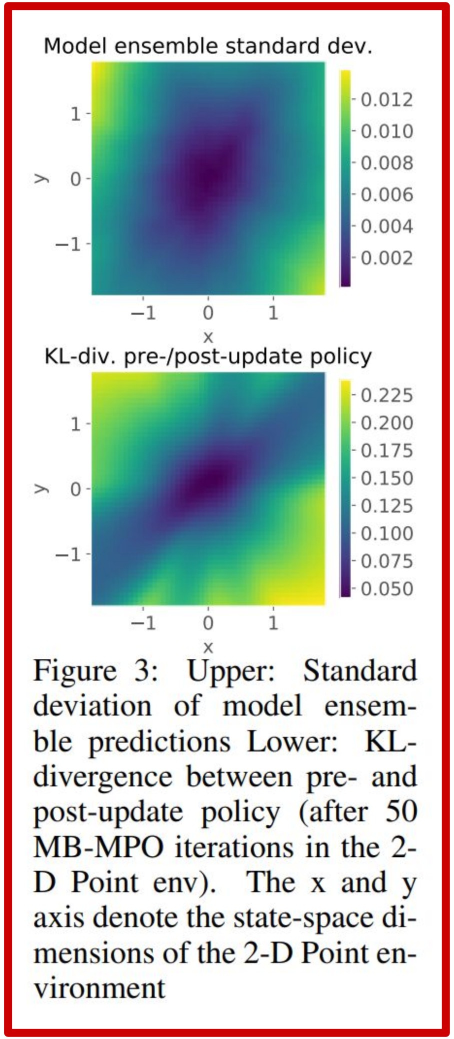


Figure 3: Upper: Standard deviation of model ensemble predictions Lower: KL-divergence between pre- and post-update policy (after 50 MB-MPO iterations in the 2-D Point env). The x and y axis denote the state-space dimensions of the 2-D Point environment

[Clavera et al 2018]

Experiments & Results

MB-MPO achieves sample efficiency wins over model-free methods

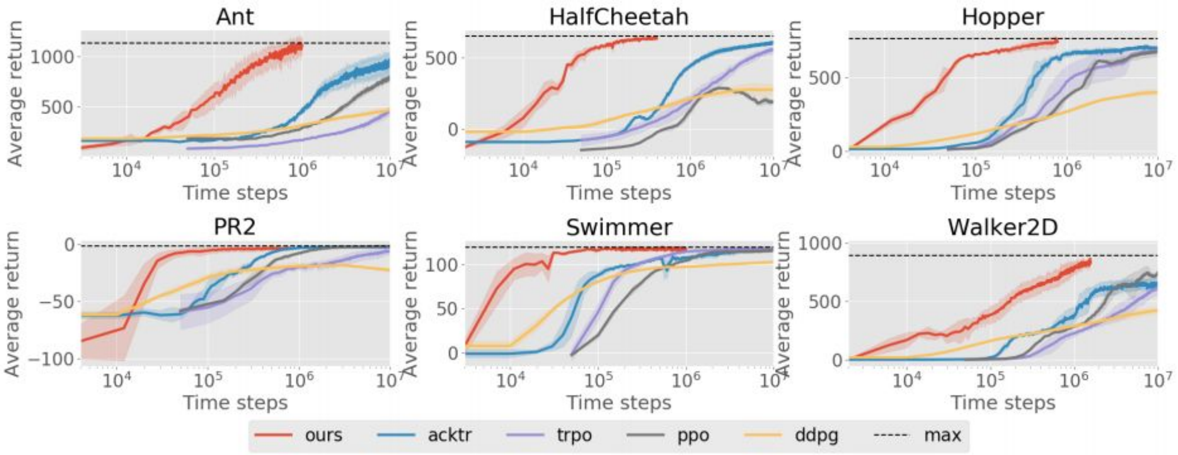


Figure 1: Learning curves of MB-MPO ("ours") and four state-of-the-art model-free methods in six different Mujoco environments with a horizon of 200. MB-MPO is able to match the asymptotic performance of model-free methods with two orders of magnitude less samples.

[Clavera et al 2018]

Experiments & Results

MB-MPO outperforms model-biased-aware model-based methods

Baselines: standard ensembles (ME-TRPO) and model-free fine tuning (MB-MPC)

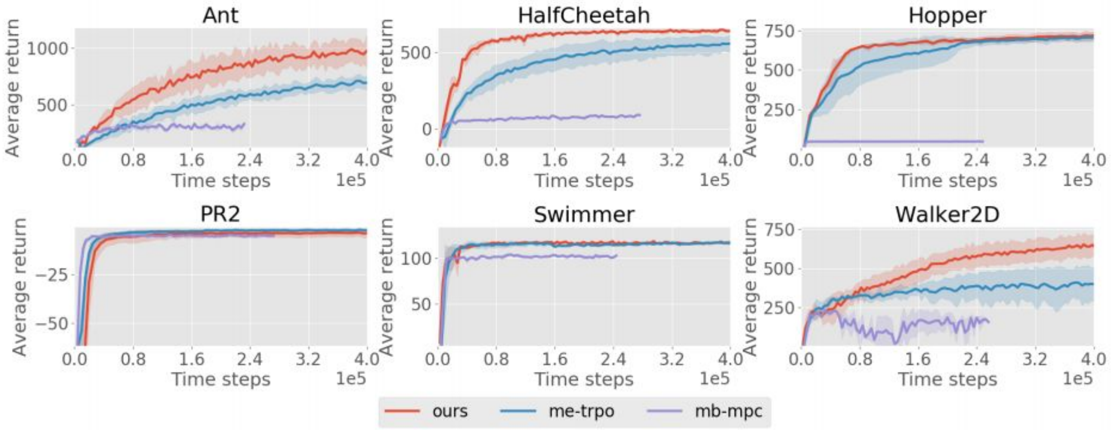


Figure 2: Learning curves of MB-MPO ("ours") and two MB methods in 6 different Mujoco environments with a horizon of 200. MB-MPO achieves better asymptotic performance and faster convergence rate than previous MB methods.

[Clavera et al 2018]

Experiments & Results

Measurement noise added to trajectories collected from environment

Therefore each ensemble is potentially unreliable

MB-MPO is more robust than standard ensembling (ME-TRPO)

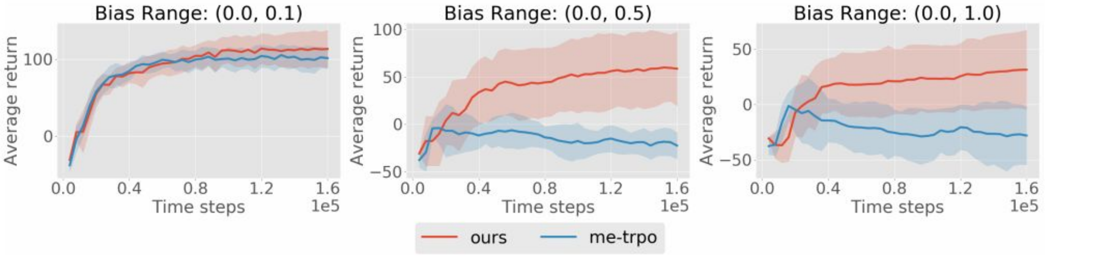


Figure 4: Comparison of MB-MPO ("ours") and ME-TRPO using 5 biased and noisy dynamic models in the half-cheetah environment with a horizon of 100 time steps. A bias term b is sampled uniformly from a denoted interval in every iteration. During the iterations we add to the predicted observation a Gaussian noise $\mathcal{N}(b, 0.1)$.

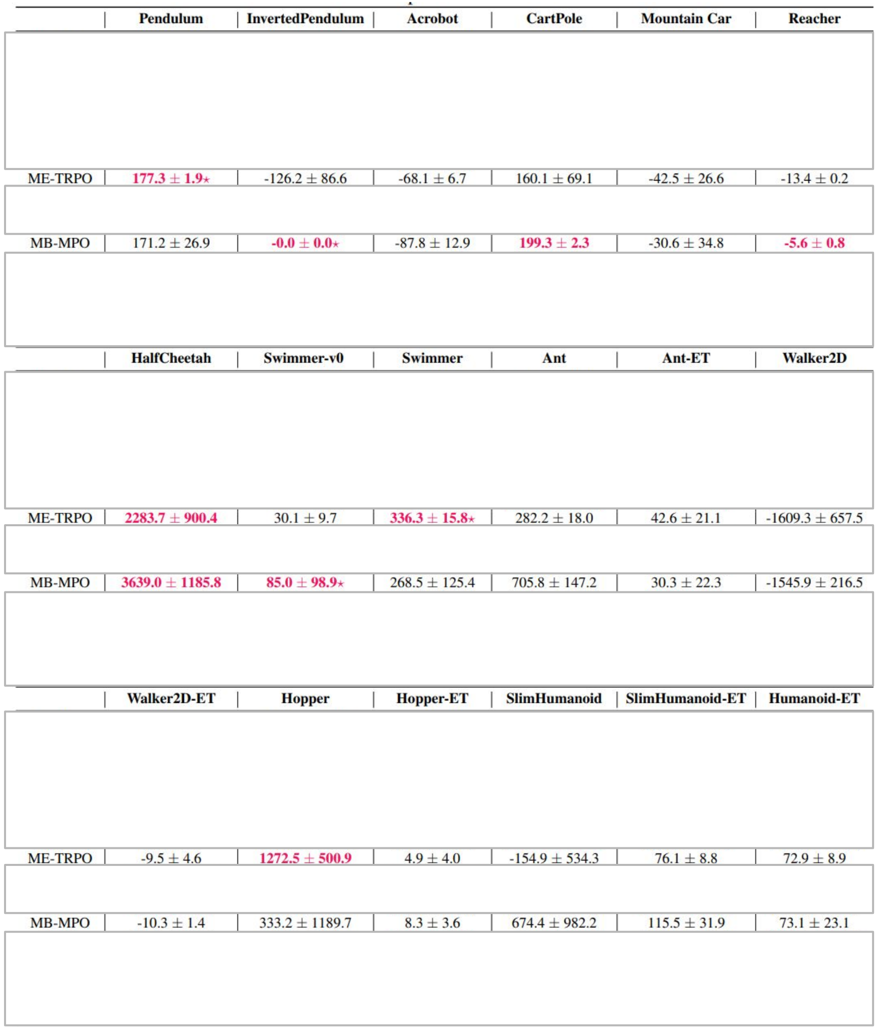
[Clavera et al 2018]

More results

MB-MPO wins: 8
ME-TRPO wins: 3
Statistical tie: 7

Total: 18

[Wang et al 2019]



	Pendulum	InvertedPendulum	Acrobot	CartPole	Mountain Car	Reacher
ME-TRPO	**177.3 ± 1.9⋆**	-126.2 ± 86.6	-68.1 ± 6.7	160.1 ± 69.1	-42.5 ± 26.6	-13.4 ± 0.2
MB-MPO	171.2 ± 26.9	**-0.0 ± 0.0⋆**	-87.8 ± 12.9	**199.3 ± 2.3**	-30.6 ± 34.8	**-5.6 ± 0.8**

	HalfCheetah	Swimmer-v0	Swimmer	Ant	Ant-ET	Walker2D
ME-TRPO	**2283.7 ± 900.4**	30.1 ± 9.7	**336.3 ± 15.8⋆**	282.2 ± 18.0	42.6 ± 21.1	-1609.3 ± 657.5
MB-MPO	**3639.0 ± 1185.8**	**85.0 ± 98.9⋆**	268.5 ± 125.4	705.8 ± 147.2	30.3 ± 22.3	-1545.9 ± 216.5

	Walker2D-ET	Hopper	Hopper-ET	SlimHumanoid	SlimHumanoid-ET	Humanoid-ET
ME-TRPO	-9.5 ± 4.6	**1272.5 ± 500.9**	4.9 ± 4.0	-154.9 ± 534.3	76.1 ± 8.8	72.9 ± 8.9
MB-MPO	-10.3 ± 1.4	333.2 ± 1189.7	8.3 ± 3.6	674.4 ± 982.2	115.5 ± 31.9	73.1 ± 23.1

Limitations & Open Issues

Meta learning is about *generalizing to new experiences*

- Despite empirical strength of MB-MPO, *is meta learning the right tool to tackle model bias?*
  - If yes, *is MAML the right meta learning approach* to tackle model bias?
- Does this approach work for environments with stochastic dynamics? What about discrete states?

Summary

Model bias is a key technical issue blocking the potential sample efficiency wins of model-based RL over model-free.

Explicitly modeling environment uncertainty (e.g., PILCO) does not scale.

In general, ensembling is an easy way to capture environment variance that works decently in practice.

Ensembles can be improved by the proposed MB-MPO, a MAML-style meta learning algorithm that seeks optimal *initial* policy parameters.

Model bias remains an open problem with many possible approaches!