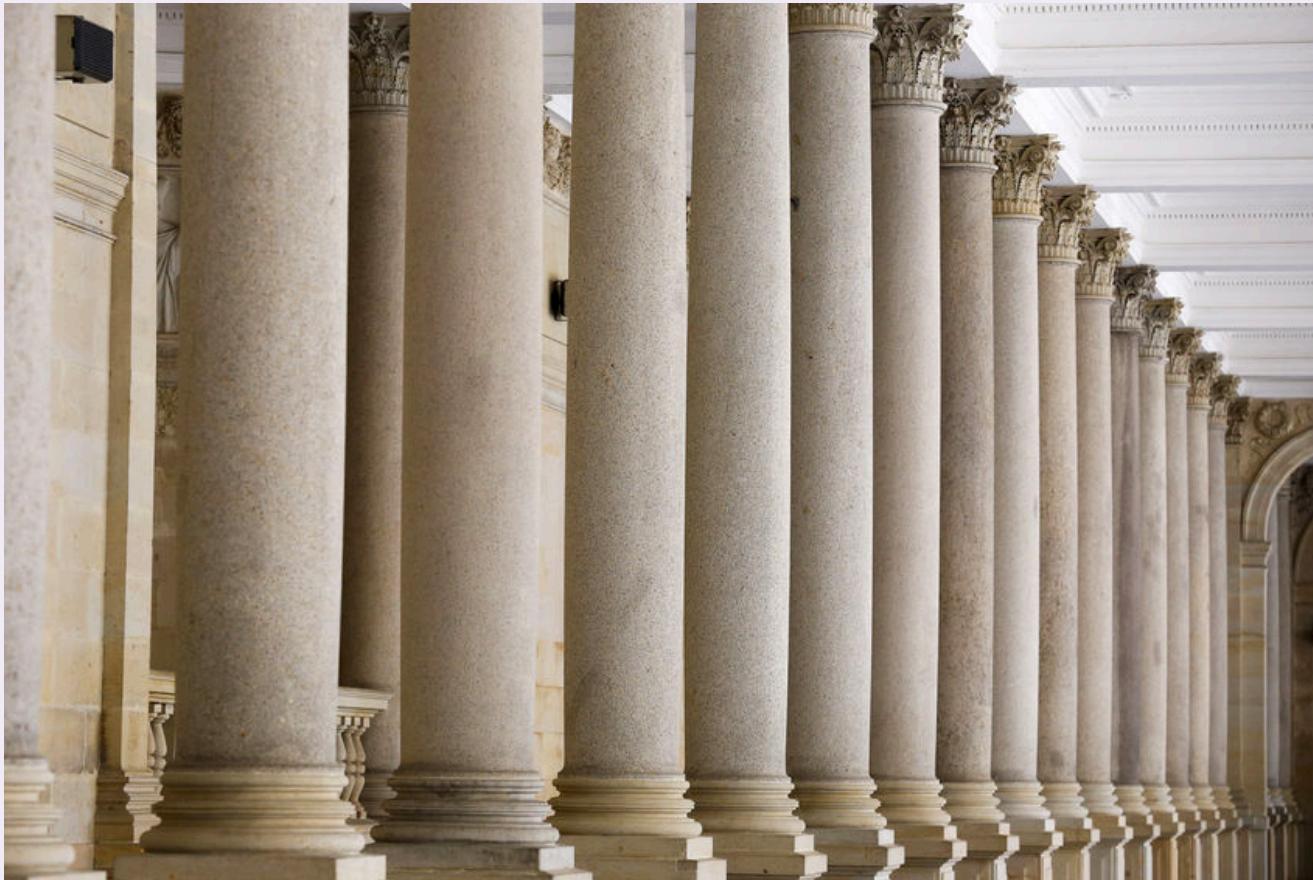


MBRL-Lib: a PyTorch toolbox for model-based reinforcement learning research

Luis Pineda

Research Engineer, FAIR Montreal

Our vision is to foster the model-based RL community with a high-quality library dedicated to MBRL



Summary

1. Model-based Reinforcement Learning - quick recap
2. Why MBRL-Lib?
3. Overview of toolbox
4. Examples
5. Baseline implementations

What is MBRL?



Train a MODEL to predict the future



Train a MODEL to predict the future



Take actions according to the model



Train a MODEL to predict the future



Take actions according to the model

Collect rewards and observations





Train a MODEL to predict the future



Take actions according to the model

Update your model

Collect rewards and observations



Some ways a model can be used:

- Derive expressions for locally-optimal control
- Perform receding horizon control over the model
- Generate simulated data to train a model-free RL agent

Why MBRL?

Some of the benefits suggested in the literature are...

Some of the benefits suggested in the literature are...

Easier control problem

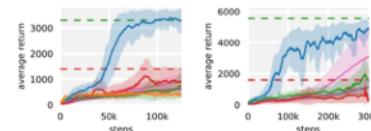


Some of the benefits suggested in the literature are...

Easier control problem



Faster convergence (using less data)

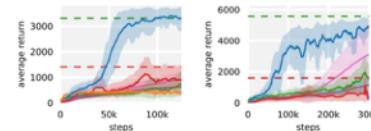


Some of the benefits suggested in the literature are...

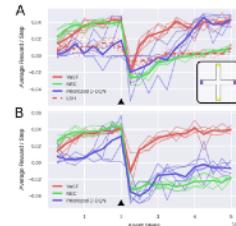
Easier control problem



Faster convergence (using less data)



Better transfer and generalization



Cornell, Dane et al. "Efficient model-based deep reinforcement learning with variational state tabulation." International Conference on Machine Learning, PMLR, 2018.

MBRL-Lib

Why this library?



MBRL is an active area of research...



MBRL is an active area of research...



...but not existing libraries to facilitate
research and experimentation



MBRL is an active area of research...



...but not existing libraries to facilitate
research and experimentation



and reproducibility of results is also an issue

MBRL-Lib

How does it look?

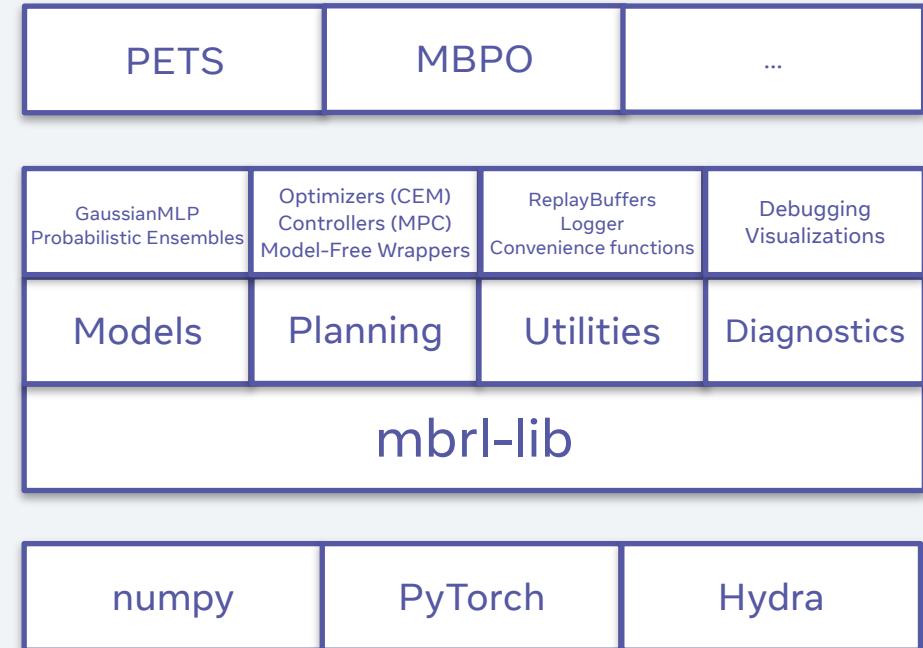
MBRL-Lib high-level overview



Algorithms

Core

Base



mbrl.models module

`mbrl.models` module

- Tools for creating and training dynamics models

mbrl.models module

- Tools for creating and training dynamics models
 - Basic model architectures

mbrl.models module

- Tools for creating and training dynamics models
 - Basic model architectures
 - Probabilistic ensembles

mbrl.models module

- Tools for creating and training dynamics models
 - Basic model architectures
 - Probabilistic ensembles
 - Pre- and post-processing of observations and rewards

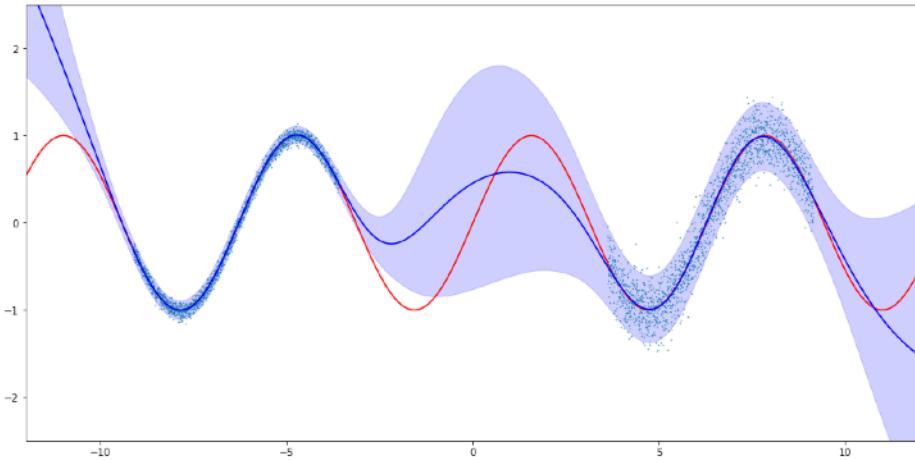
mbrl.models module

- Tools for creating and training dynamics models
 - Basic model architectures
 - Probabilistic ensembles
 - Pre- and post-processing of observations and rewards
 - Trainers customized for dynamics models

mbrl.models module

- Tools for creating and training dynamics models
 - Basic model architectures
 - Probabilistic ensembles
 - Pre- and post-processing of observations and rewards
 - Trainers customized for dynamics models
 - Wrappers to use dynamics models as gym-like environments

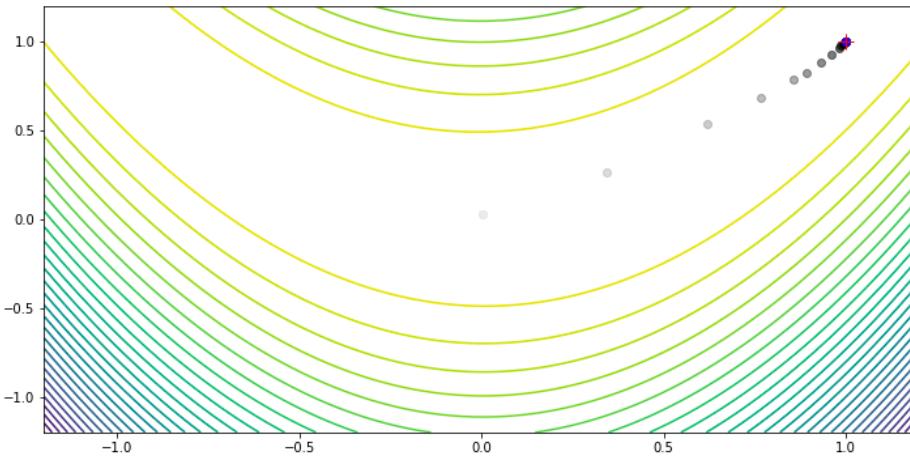
Jupyter example: training a probabilistic ensemble



mbrl.planning module

- Tools for control
 - A minimal Agent abstraction
 - Wrappers for popular model-free agents (e.g., DQN, SAC)
 - Model-based planners (CEM, Random shooting)

Jupyter example: Cross-Entropy Method optimizer



mbrl.util package

mbrl.util package

- Utility and convenience functions for:

mbrl.util package

- Utility and convenience functions for:
 - Creating dynamics model and replay buffers from a Hydra config

mbrl.util package

- Utility and convenience functions for:
 - Creating dynamics model and replay buffers from a Hydra config
 - Run dynamics model training and save data to disk

mbrl.util package

- Utility and convenience functions for:
 - Creating dynamics model and replay buffers from a Hydra config
 - Run dynamics model training and save data to disk
 - Rollout models using a given Agent

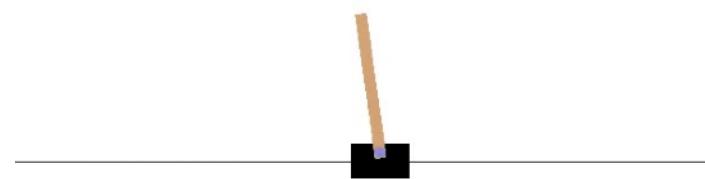
mbrl.util package

- Utility and convenience functions for:
 - Creating dynamics model and replay buffers from a Hydra config
 - Run dynamics model training and save data to disk
 - Rollout models using a given Agent
 - Freeze a gym environment, do rollouts, restore to original state

mbrl.util package

- Utility and convenience functions for:
 - Creating dynamics model and replay buffers from a Hydra config
 - Run dynamics model training and save data to disk
 - Rollout models using a given Agent
 - Freeze a gym environment, do rollouts, restore to original state
 - Populate replay buffers with a given Agent

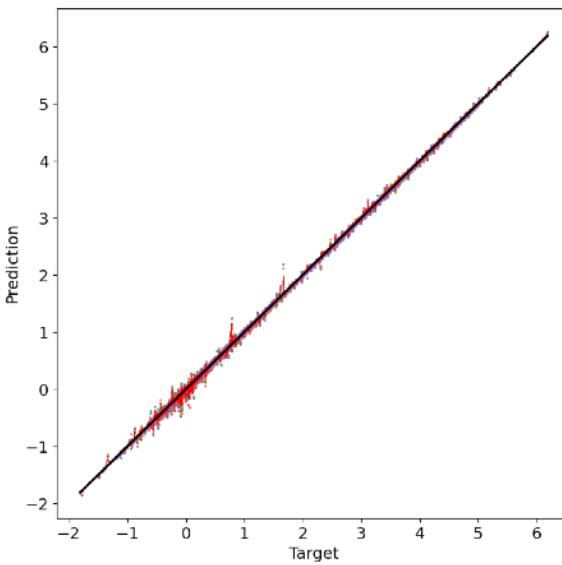
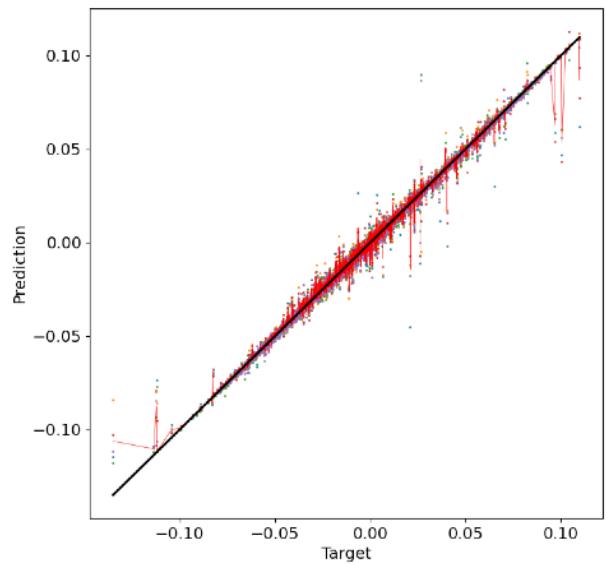
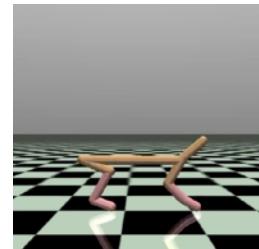
Jupyter example: Implementing PETSc on cartpole



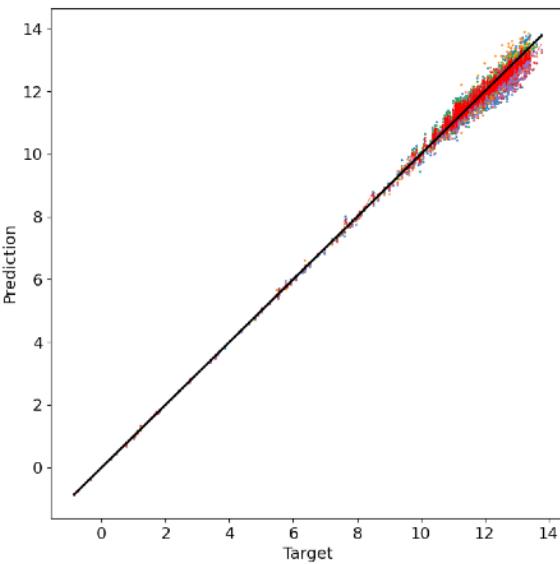
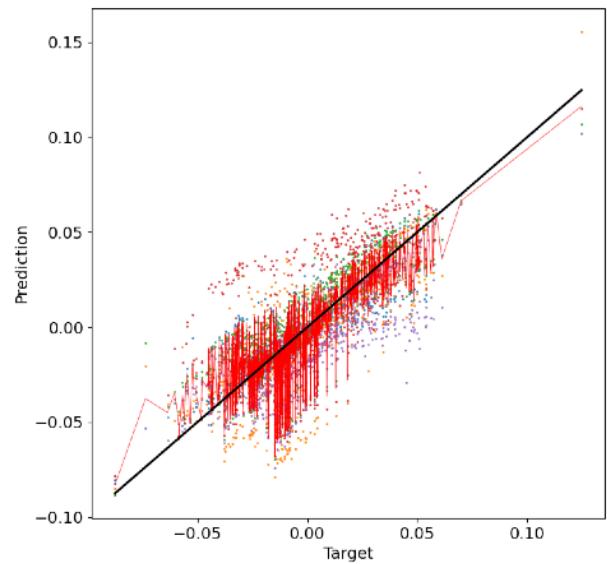
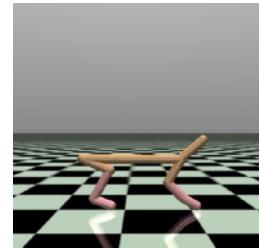
mbrl.diagnostics package

- Scripts to diagnose/visualize individual components, for example:
 - Evaluate dynamics models on dataset of stored transitions
 - Evaluate dynamics models over a receding horizon
 - Evaluate an agent on the true dynamics

Evaluating model on a dataset of transitions

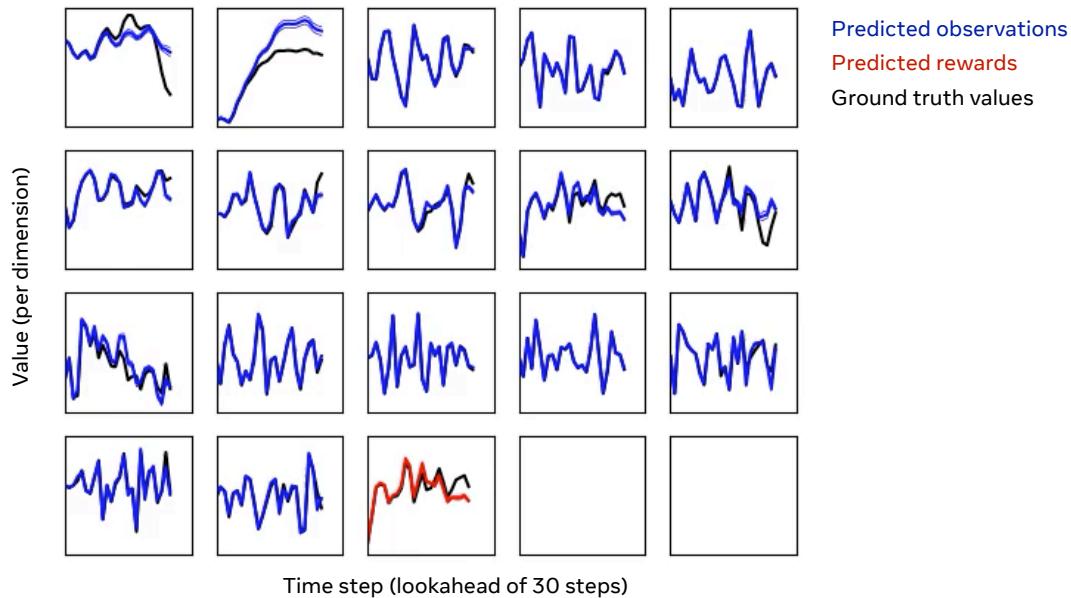
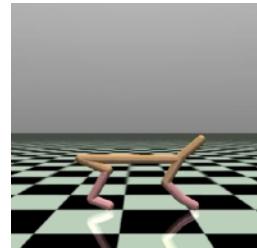


Evaluating model on a dataset of transitions



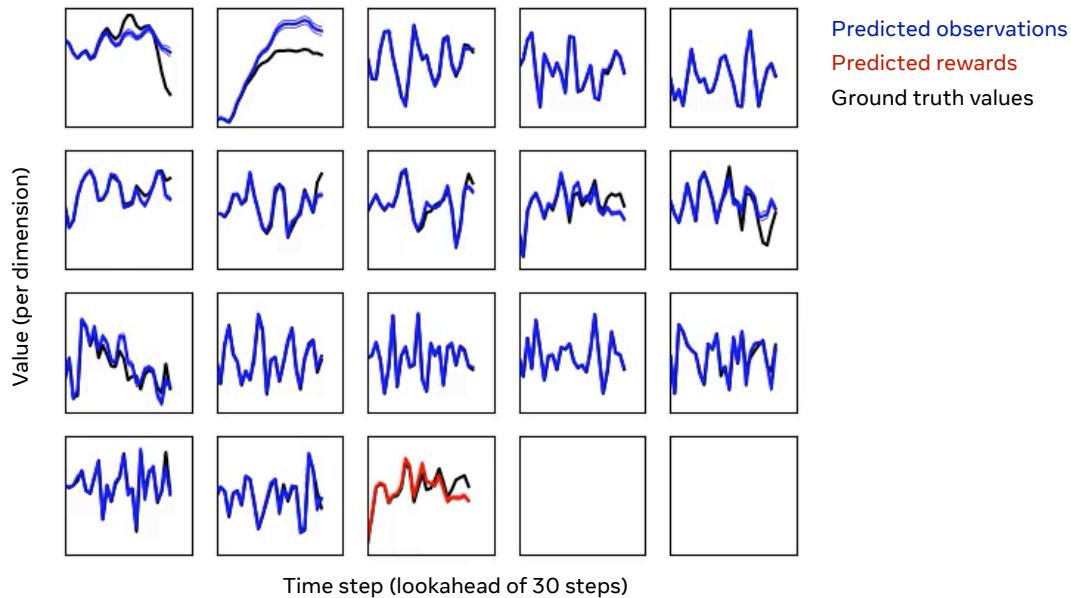
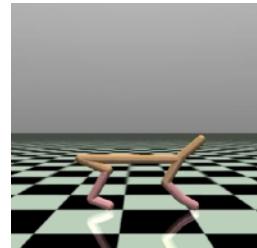
Evaluate models on a receding horizon

HalfCheetah environment. Rolling out the MPC controller



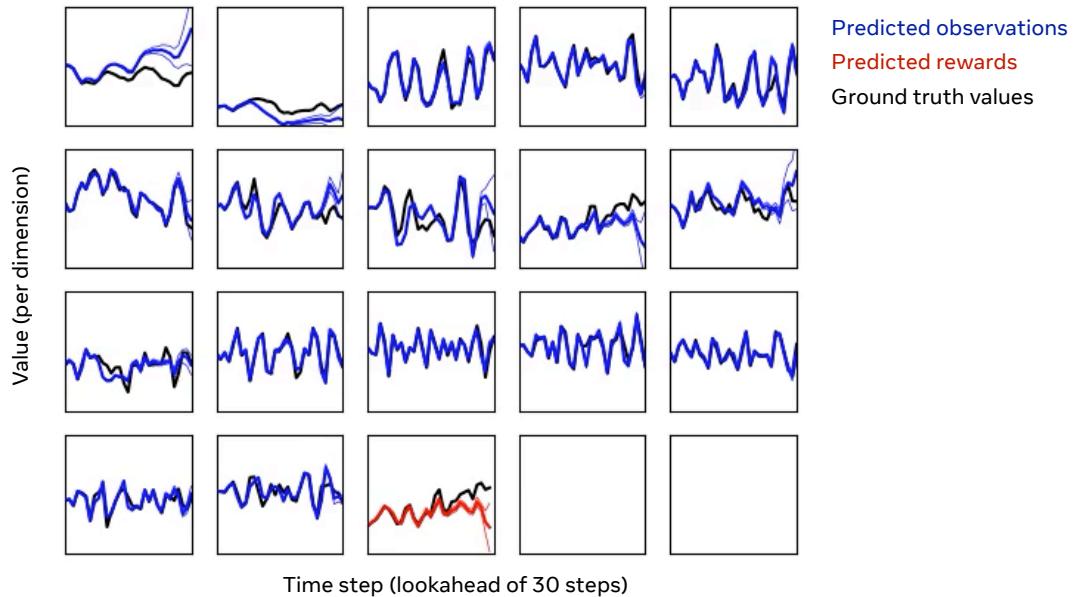
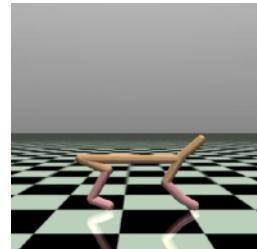
Evaluate models on a receding horizon

HalfCheetah environment. Rolling out the MPC controller



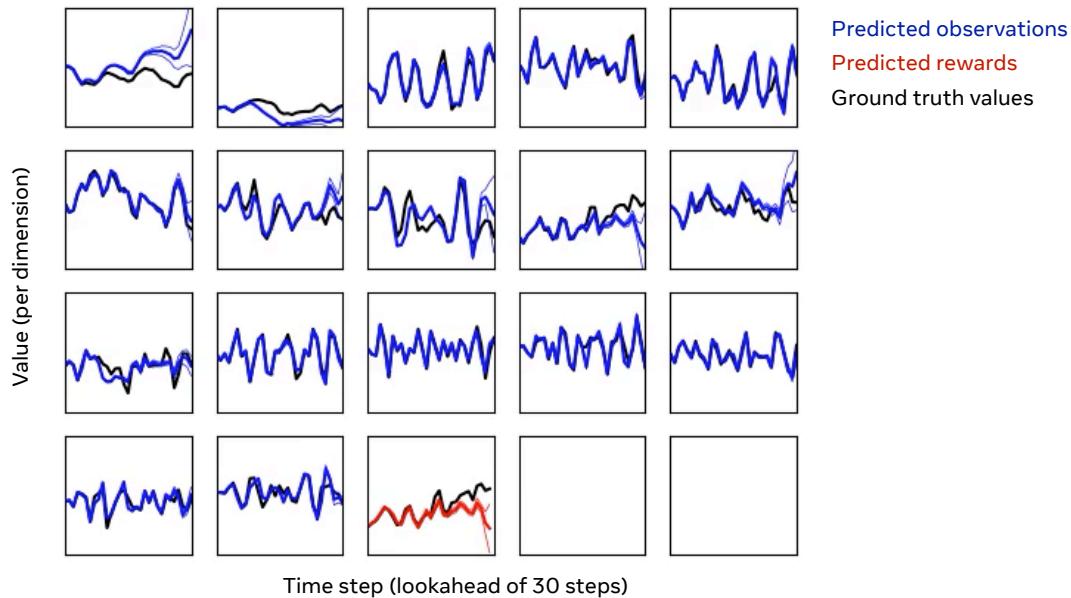
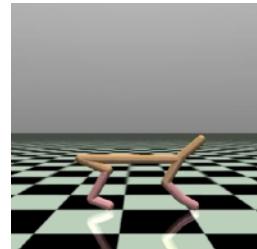
Evaluate models on a receding horizon

HalfCheetah environment. Rolling out a SAC controller



Evaluate models on a receding horizon

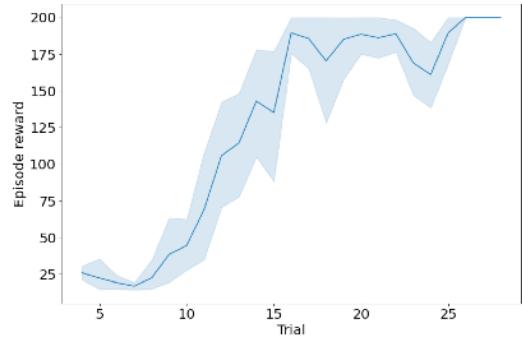
HalfCheetah environment. Rolling out a SAC controller



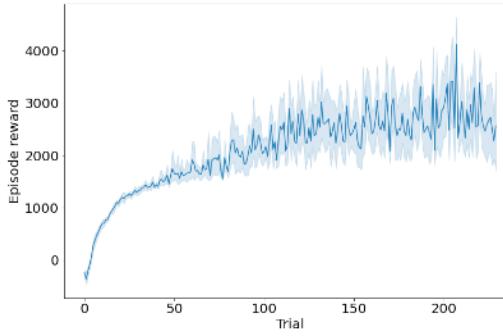
Example baselines

PETS

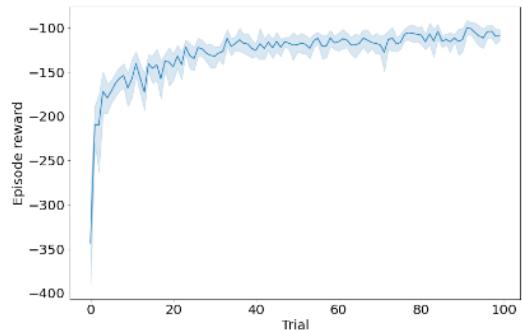
K. Chua et al. "Deep reinforcement learning in a handful of trials using probabilistic dynamics models." NeurIPS, 2018



Cartpole-v2 (continuous mod.)



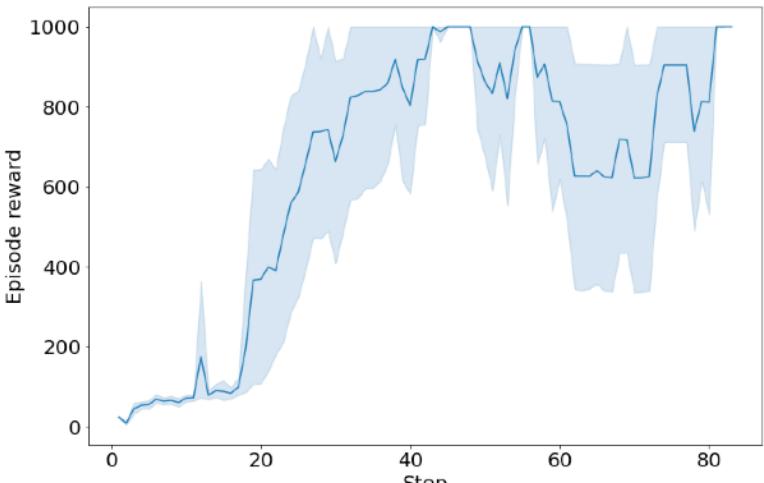
HalfCheetah (PETS version)



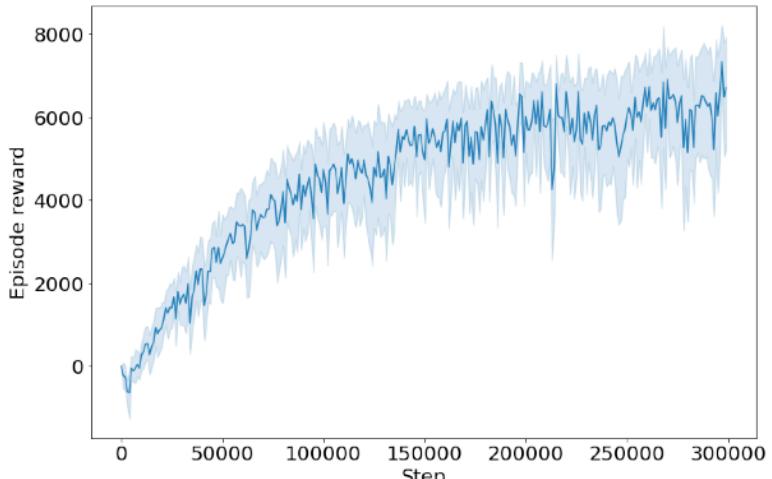
7 DOF Pusher (PETS version)

MBPO

Janner, Michael, et al. "When to trust your model: Model-based policy optimization." NeurIPS. 2019.



InvertedPendulum-v2



HalfCheetah-v-2

Next steps

Next steps

- More features (other control methods, architectures, MBRL algorithms)

Next steps

- More features (other control methods, architectures, MBRL algorithms)
- More importantly, **supporting you!**



Collaborators



Roberto Calandra
FAIR



Amy Zhang
FAIR/MILA



Brandon Amos
FAIR



Nathan Lambert
UC Berkeley

MBRL-Lib high-level overview

For early access, contact
Luis Pineda lep@fb.com



Algorithms

Core

Base

PETS

MBPO

...

GaussianMLP
Probabilistic Ensembles

Optimizers (CEM)
Controllers (MPC)
Model-Free Wrappers

ReplayBuffers
Logger
Convenience functions

Debugging
Visualizations

Models

Planning

Utilities

Diagnostics

mbrl-lib

numpy

PyTorch

Hydra