Towards real-world RL

Felix Berkenkamp @ RL4AA 2024-02-05

BOSCH

Problem setup
Real-world conditions

Problem setup

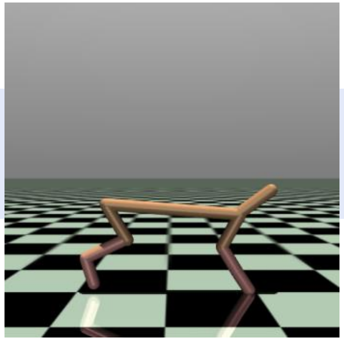
Markov Decision Process

Agent

Reward r_t

State $\boldsymbol{s}_t$

Action $\boldsymbol{a}_t$



Find a policy $\boldsymbol{a}_t = \pi(\boldsymbol{s}_t)$ that maximizes the sum of expected rewards

Reinforcement Learning: An Introduction
R. Sutton, A.G. Barto, 1998

Problem setup
Partial observability

Find a policy $\boldsymbol{a}_t = \pi(\boldsymbol{o}_t)$ that maximizes the sum of expected rewards

Reinforcement Learning: An Introduction
R. Sutton, A.G. Barto, 1998

Problem setup
Oh no

Agent

Reward r_t

Observation $\boldsymbol{o}_t$

Action $\boldsymbol{a}_t$

This is a bad idea because...

- Data-efficiency
- Random actuation not great for real systems
- Robustness to different situations
- Safety during and after learning

Problem setup

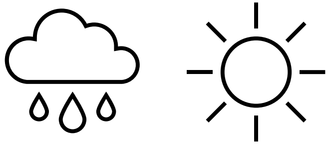
Simulated Off-policy meTA learning (SOTA)

Agent

Reward r_t

Observation $\boldsymbol{o}_t$

Action $\boldsymbol{a}_t$



Hope for the best

Build simulator

Randomize over environment conditions

(recurrent policy)

$\boldsymbol{a}_t = \pi(\boldsymbol{o}_t)$

BOSCH

Problem setup

Beyond SOTA

Agent

Reward r_t

Observation $\boldsymbol{o}_t$

Action $\boldsymbol{a}_t$

This is a bad idea *at the moment* because...

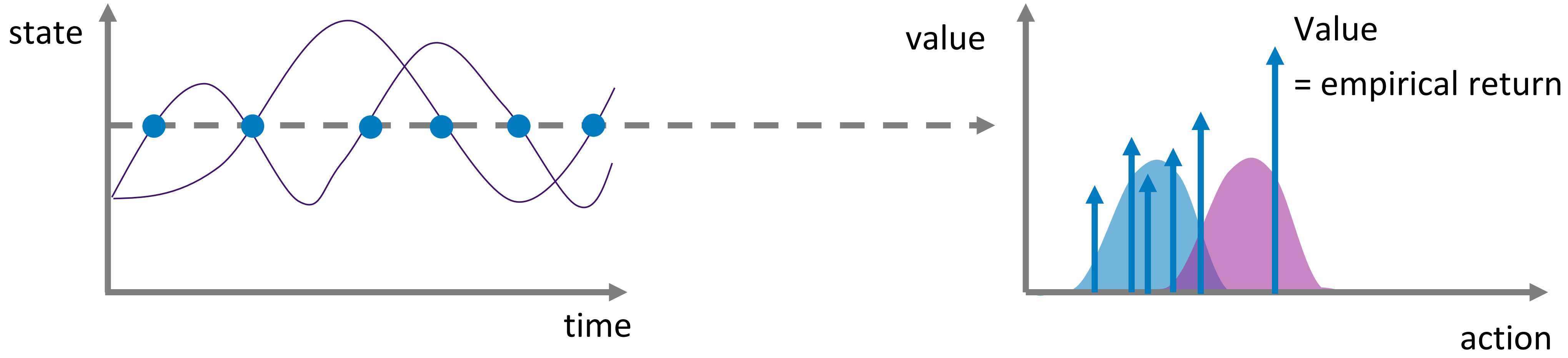
- Data-efficiency
- Random actuation not great for real systems
- Robustness to different situations
- Safety during and after learning

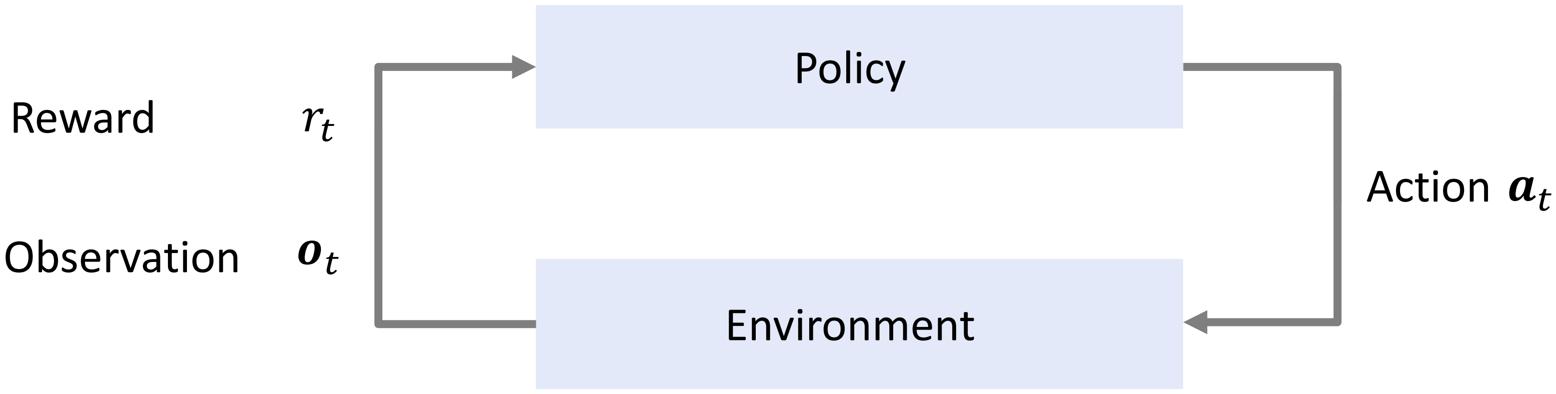
Exploration / model-based RL

Safe / Meta RL

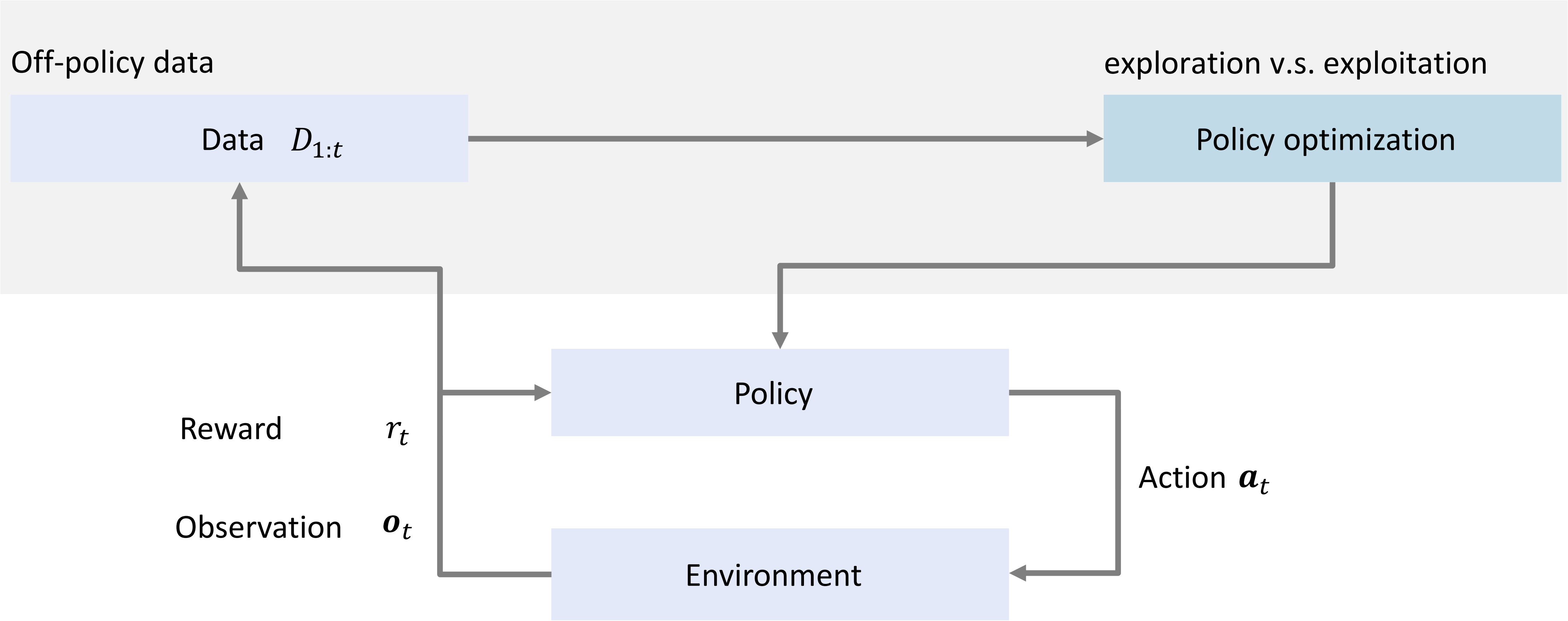
Reinforcement Learning

on-policy





Reinforcement Learning
Off-policy („offline“)



Reinforcement Learning
Partial observability

Off-policy data: $(o_0, a_0), (o_1, a_1), (o_2, a_2), (o_3, a_3), \cancel{(o_4, a_4), (o_5, a_5)}$

$$a_3 = \arg\max_a Q(s_3, a)$$

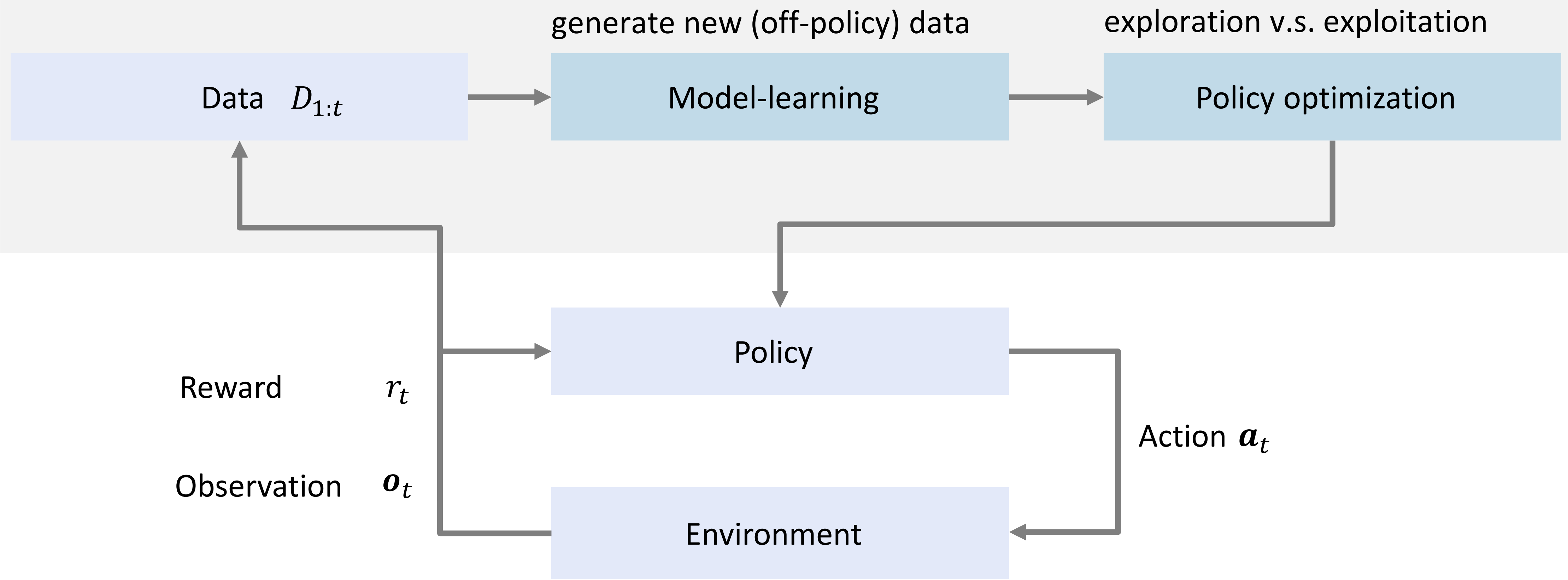
Can use standard algorithms again!

Learned inference network

Common mistake: MDP: $Q(s_t, a_t)$ POMDP: $\cancel{Q(o_{1:t}, a_t)}$ $Q(o_{1:t}, \boldsymbol{a_{1:t-1}}, a_t)$

Actor & Critic must depend on past observations & actions (also on-policy)

Reinforcement Learning
Model-based



Reinforcement Learning
Reinforced flavours

On-policy

- Reliable algorithms
- Easy to use with recurrent policies

- Extremely data in-efficient

Off-policy

- More data-efficient

- Quality of the learned critic extremely important
- Learning recurrent policies requires rollin

  → computationally expensive

Model-based

- Use cheap model data with inefficient RL algorithms!
- Model usually not good enough for full episode rollouts → off policy RL
- Hybrid models seem great, but can also make things worse and difficult to learn
- Usually actuators are the most critical part to model

Reinforcement Learning
Iterated Offline

Data-collection

Data from π_n

π_n

Policy optimization

Find the best policy given the data

Add trust regions: $d(\pi, \pi_n) \leq \epsilon$

Reinforcement Learning
Reward

Environment

Simple, smoothed dynamics

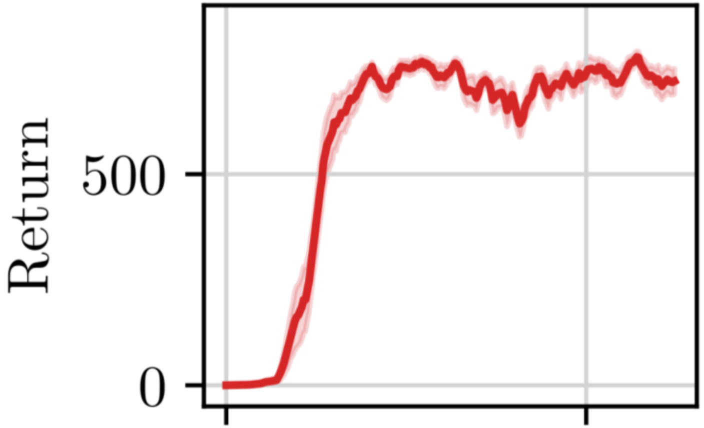
Reward function (no action cost)

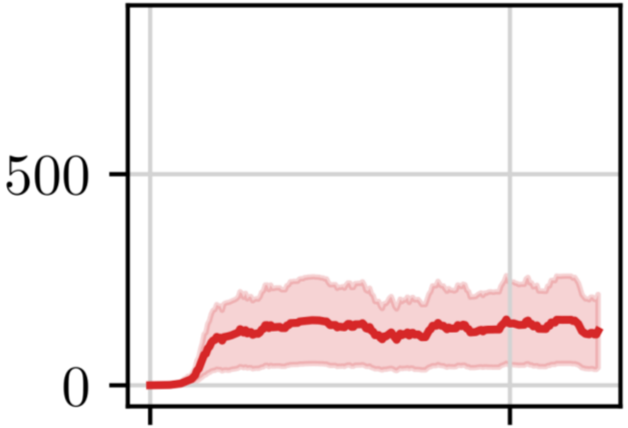
action cost: $0.01\|a_t\|_2^2$

Cartpole

Swingup

SAC





BOSCH

Reinforcement Learning
Reward

Environment

Partially-observable dynamics

observations

actions

Reward function

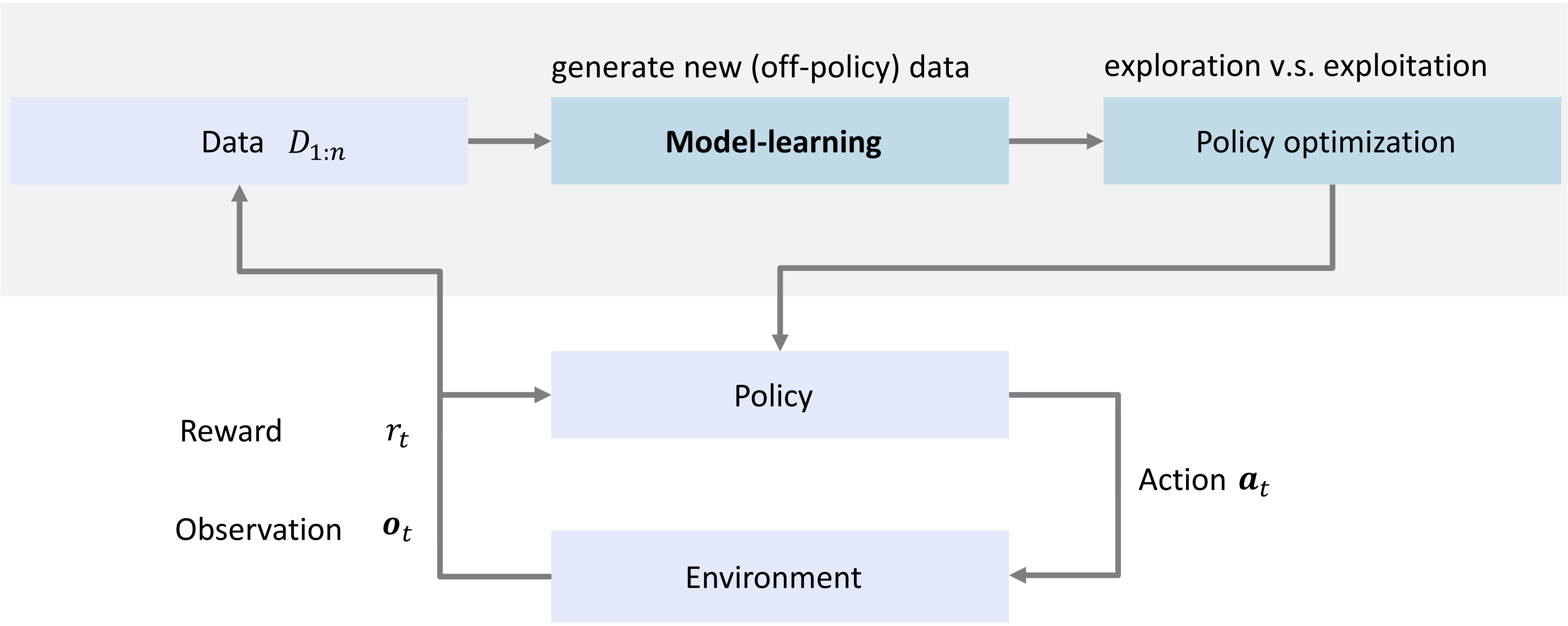
Tuning factor

Potentially varies over time

Some guidelines:

- Start with dense, bounded rewards
- Start with observation-dependent reward (action costs fights with entropy-based exploration)
- It’s okay to change rewards → Re-label past data!
- Be careful about termination signals & rewards (easy to encode unexpected behavior)
  - Desired termination: reward > max return
  - Undesired termination: reward < min return
- If this is a key issue, might be easier to look into constrained reinforcement learning

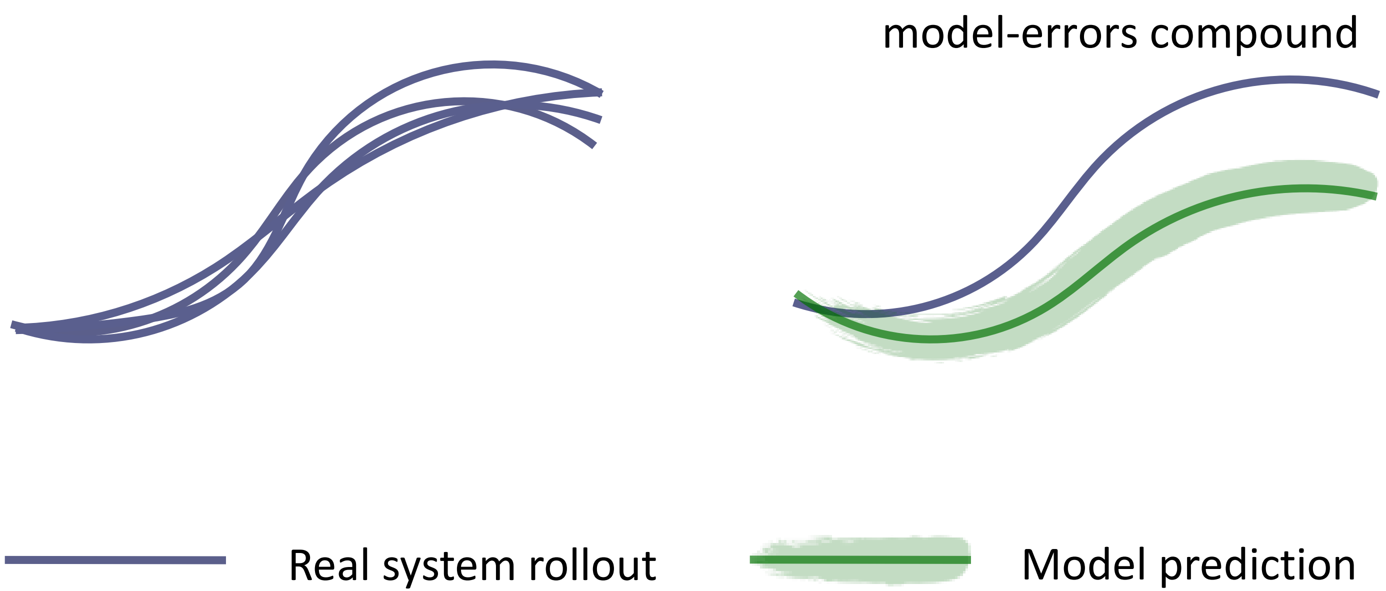
Model-based Reinforcement Learning



Model-based Reinforcement Learning
Model errors

Environment data /
Experience replay

Learned model



Model-based Reinforcement Learning

Policy Improvement

$$\underbrace{\eta_{n+1} - \eta_n}_{\text{True policy improvement}} \qquad \underbrace{\tilde{\eta}_{n+1} - \tilde{\eta}_n}_{\text{Model policy improvement}} - \underbrace{\left|\eta_{n+1} - \tilde{\eta}_{n+1}\right|}_{\text{Off-policy model error}} - \underbrace{\left|\eta_n - \tilde{\eta}_n\right|}_{\text{On-policy model error}}$$

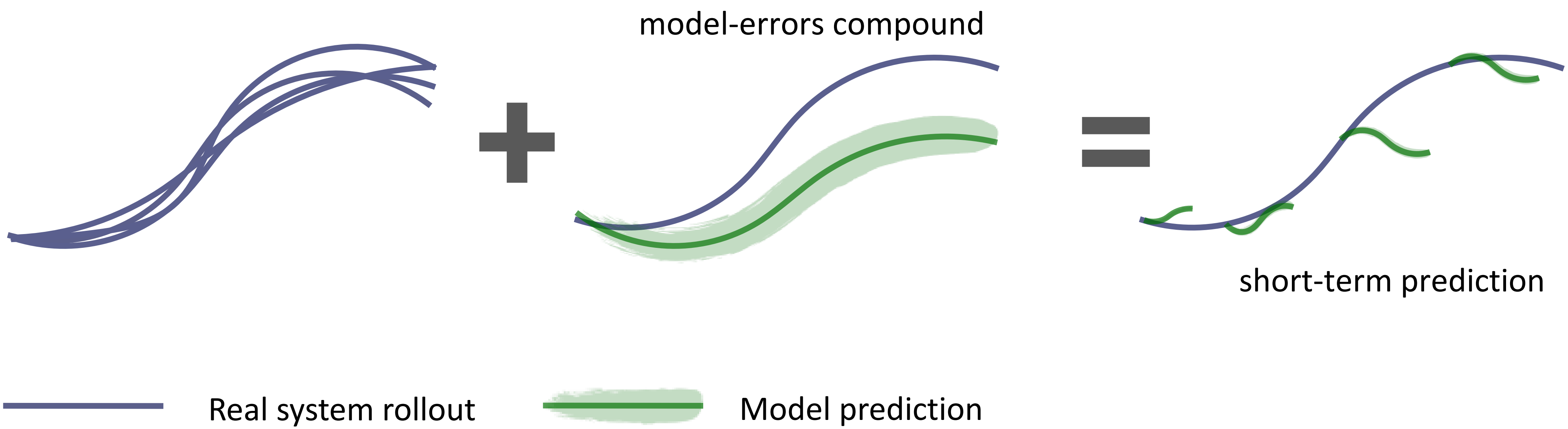
	Off-policy model error	On-policy model error
Experience Replay	↗	↘
Learned Dynamics	↘	↗

Model-based Reinforcement Learning
Model errors

Environment data / Experience replay

Learned model

DYNA / MBPO
Partial rollouts



Model-based Reinforcement Learning

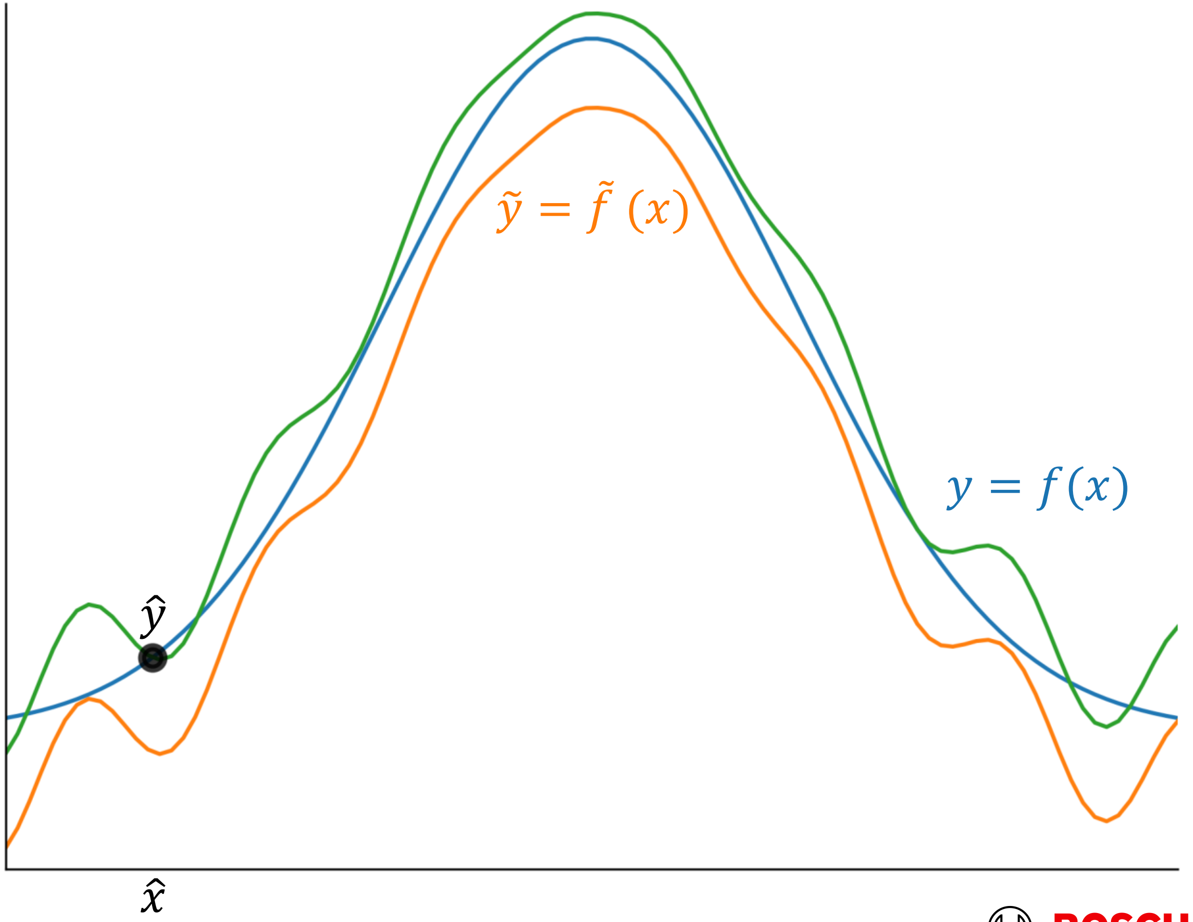
$$f^{opc}(x) = \hat{y} + [\tilde{f}(x) - \tilde{f}(\hat{x})]$$

Asymptotically correct:

$$\tilde{f}(x) = f(x)\ \forall\, x \in X \quad \Longrightarrow \quad f^{opc}(x) = f(x)$$

Locally no errors:

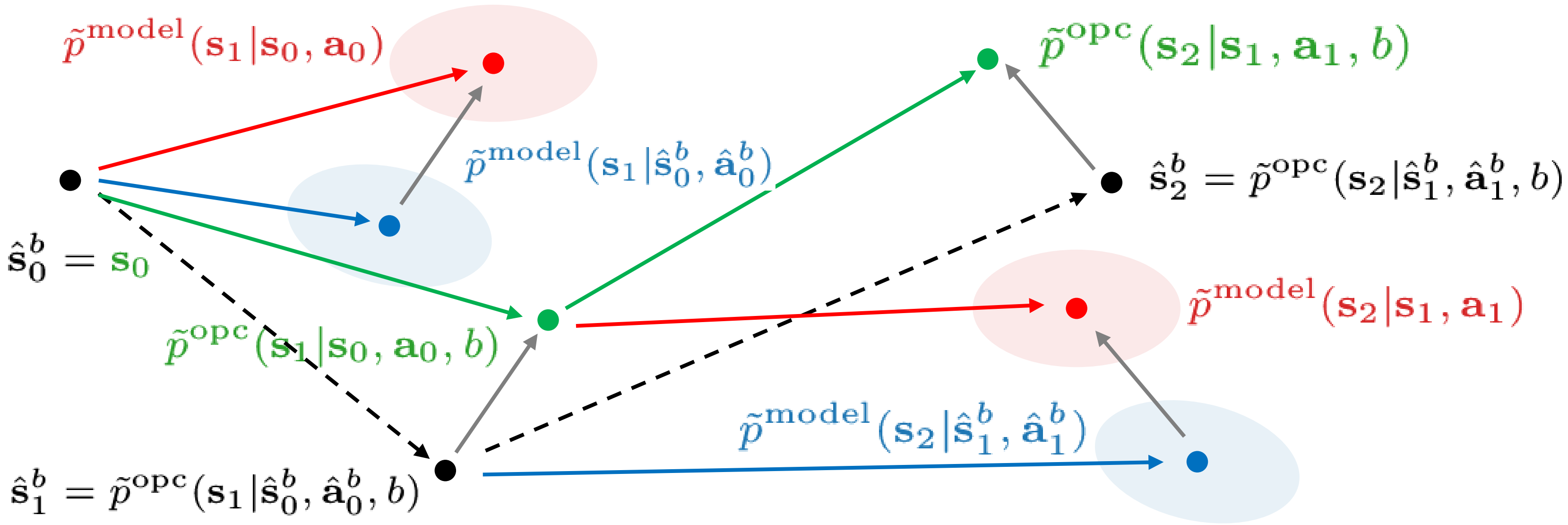
$$f^{opc}(\hat{x}) = f(\hat{x})$$



Model-based Reinforcement Learning
On-Policy Corrections (OPC)

Idea: Generalize replay buffer $(\hat{s}_t, \hat{a}_t, \hat{s}_{t+1})$ with model:

$$\mathbf{s}_{t+1}^{\mathrm{opc}} = \hat{\mathbf{s}}_{t+1} + \mathbb{E}\left[\tilde{p}^{\mathrm{model}}(\mathbf{s}_t, \mathbf{a}_t) - \tilde{p}^{\mathrm{model}}(\hat{\mathbf{s}}_t, \hat{\mathbf{a}}_t)\right]$$



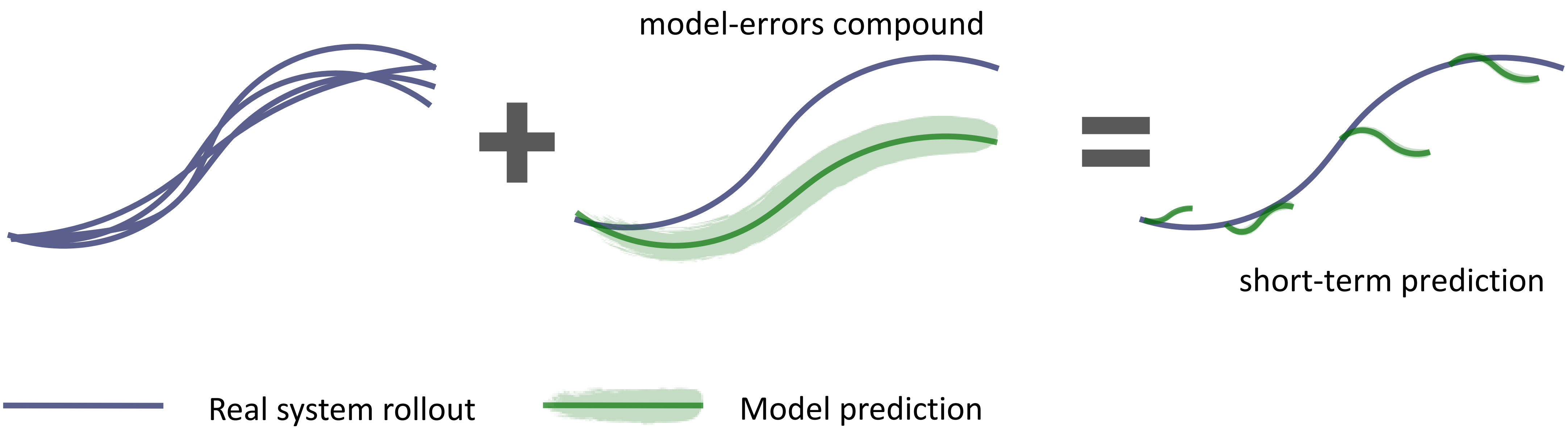
On-policy Model Errors in Reinforcement Learning
L.P. Fröhlich, M. Lefarov, M.N. Zeilinger, F. Berkenkamp, ICLR 2022

Model-based Reinforcement Learning
Model errors

Environment data /
Experience replay

Learned model

DYNA / MBPO
Partial rollouts



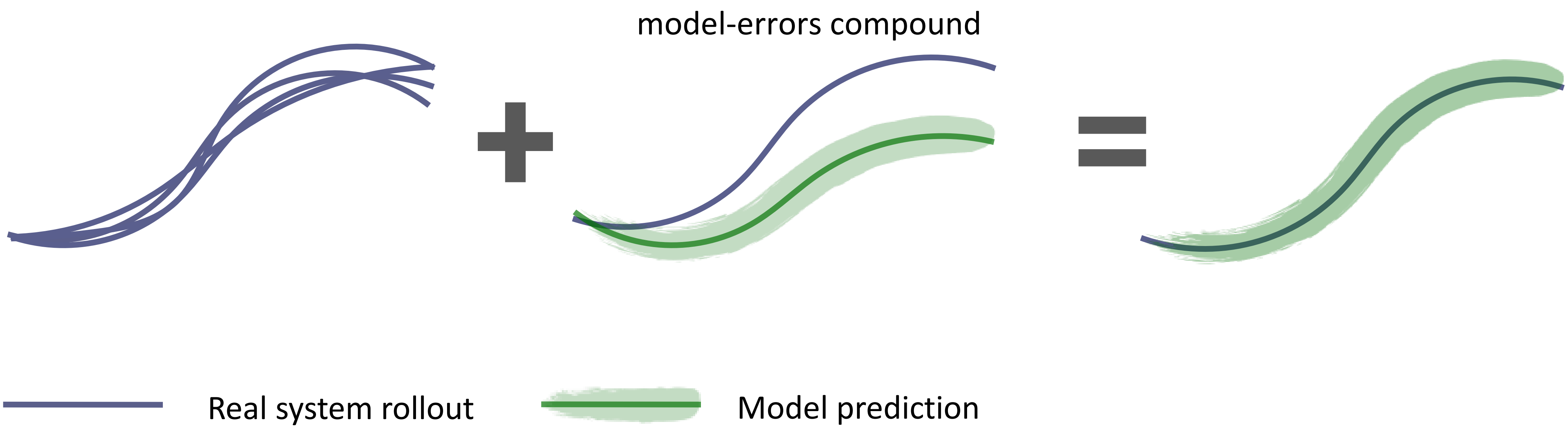
Model-based Reinforcement Learning

Model errors

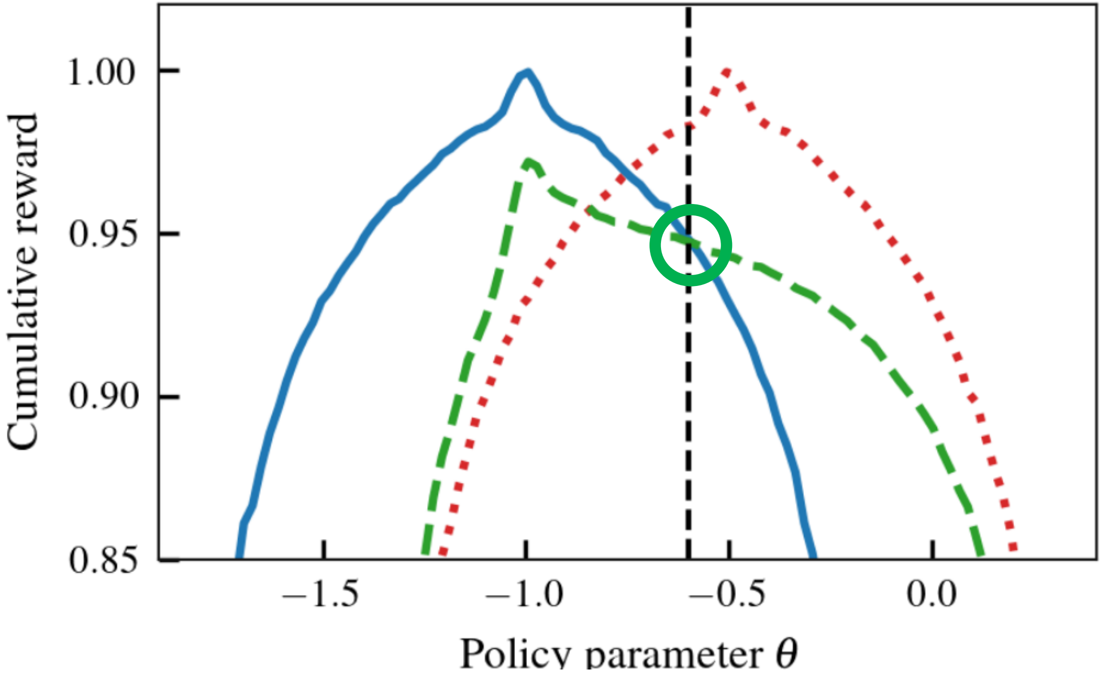
Environment data / Experience replay

Learned model

On-policy corrected rollouts



Model-based Reinforcement Learning
Scalar toy system



True system · Model · Model + OPC · Reference policy π^n

Model-based Reinforcement Learning
Theoretical guarantees

Idealized model: $$\tilde{p}_{\star}^{\text{opc}}(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t, b) = \underbrace{p(\hat{\mathbf{s}}_{t+1} \mid \hat{\mathbf{s}}_t^b, \hat{\mathbf{a}}_t^b)}_{\text{On-policy transition}} * \underbrace{\delta\Big(\mathbf{s}_{t+1} - \Big[\tilde{f}(\mathbf{s}_t, \mathbf{a}_t) - \tilde{f}(\hat{\mathbf{s}}_t^b, \hat{\mathbf{a}}_t^b)\Big]\Big)}_{\text{Mean correction term}}$$

Theorem (informal): Under continuity assumptions, for a deterministic policy we have

on-policy error: $$\left|\eta_t - \tilde{\eta}_t^{\text{opc}}\right| = 0$$

On-policy Model Errors in Reinforcement Learning
L.P. Fröhlich, M. Lefarov, M.N. Zeilinger, F. Berkenkamp, ICLR 2022

Towards real-world RL
Summary

- One of the key questions is how to bring RL to real systems:

  partial observability, data-efficiency, robustness
- Need to take care when choosing algorithms and rewards

- Model-based methods might be promising, but there's a sim2real gap
- One solution: Use model only to predict changes to real-environment data

https://berkenkamp.me