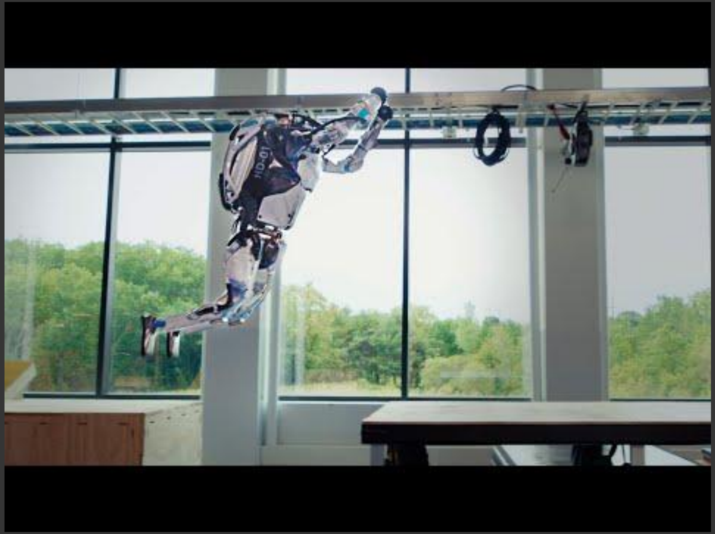
Model-based reinforcement learning under uncertainty: the importance of knowing what you don't know

by Aidan Scannell
15th November 2022

Machine learning for robotics

DARPA Robotics Challenge 2015



Atlas | Partners in Parkour | Boston Dynamics

Outline

1. What’s model-based RL?
2. Why model-based RL?
3. Uncertainty quantification in model-based RL
    a. Why uncertainty quantification in model-based RL?
    b. Sources of uncertainty
    c. How to quantify uncertainty?
    d. How to propagate uncertainty?
    e. Uncertainty-guided exploration
4. Examples
5. Issues in model-based RL?

What’s model-based RL?

Preliminaries

Goal: $$\operatorname*{argmax}_{\pi} \underset{\underbrace{s_{t+1} \sim p(\cdot \mid s_t, a_t)}_{\text{environment}}}{\underset{a_t \sim \pi(\cdot|s_t)}{\mathbb{E}}} \left[\sum_{t=0}^{\infty} \gamma^t r\left(s_t, a_t\right)\right]$$

Collect data

$$\mathcal{D} = \{s_t, a_t, r_{t+1}, s_{t+1}\}_{t=0}^{T}$$

Model-free: learn policy directly from data

$$\mathcal{D} \rightarrow \pi$$

Model-based: learn a model, then use it to improve policy

$$\mathcal{D} \rightarrow f \rightarrow \pi$$

What's a model?

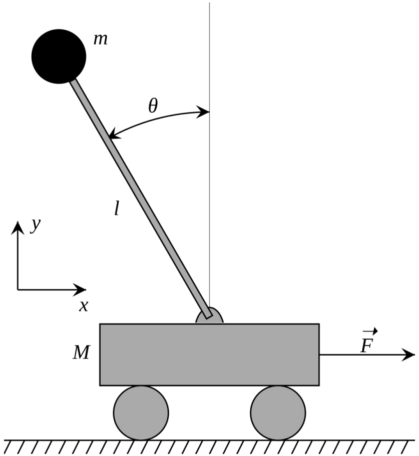
*Definition: a model is a representation that **explicitly** encodes knowledge about the structure of the environment and task.*

Dynamics/transition model	$s_{t+1} = f(s_t, a_t)$
Reward model	$r_{t+1} = f(s_t, a_t)$
Inverse dynamics/transition model	$a_t = f^{-1}(s_t, s_{t+1})$
Model of distance	$d_{ij} = f_d(s_i, s_j)$
Model of future returns	$G_t = Q(s_t, a_t)$

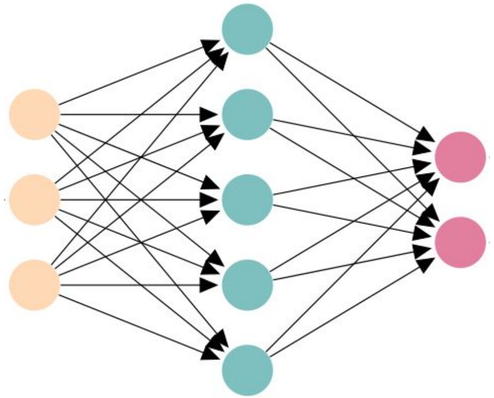
Typically this is what's meant in model-based RL

What’s a model?

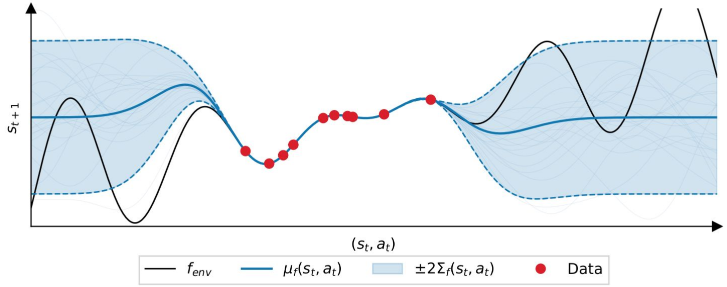
Physics based



Neural network

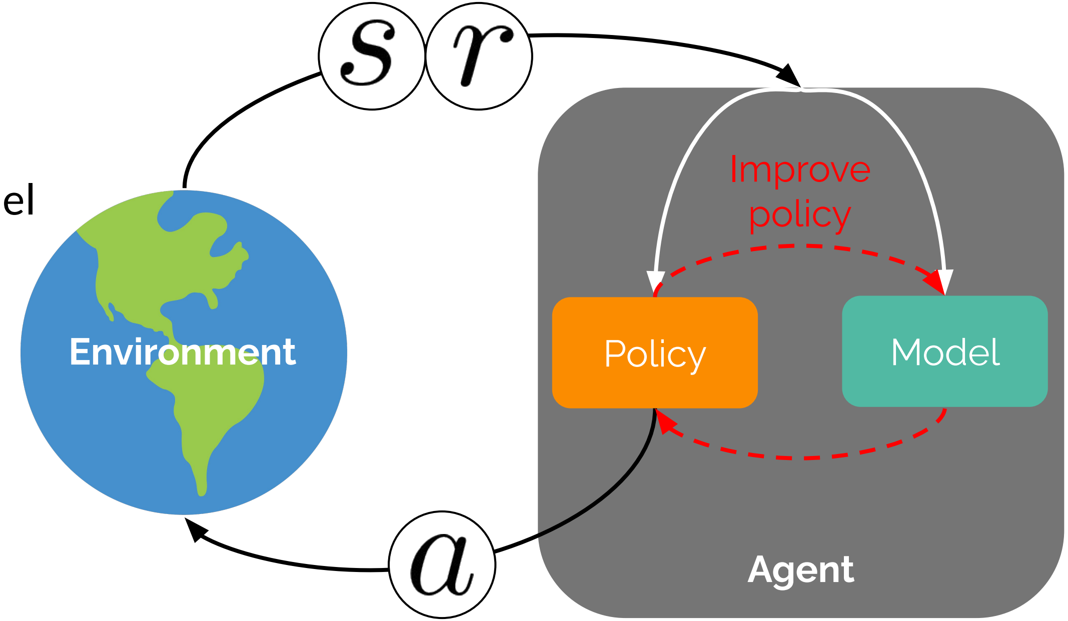


Gaussian process

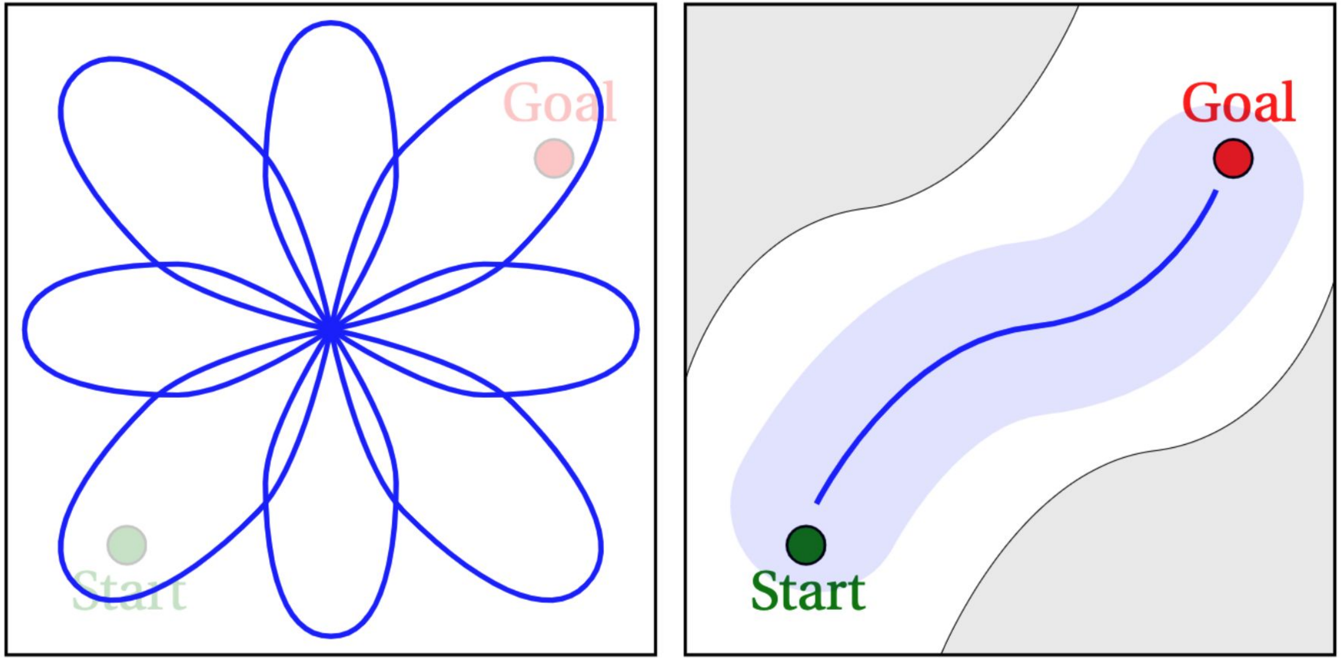


Model-based RL algorithm

1. Collect data using policy π
2. Learn model using data set
3. Improve policy using learned model



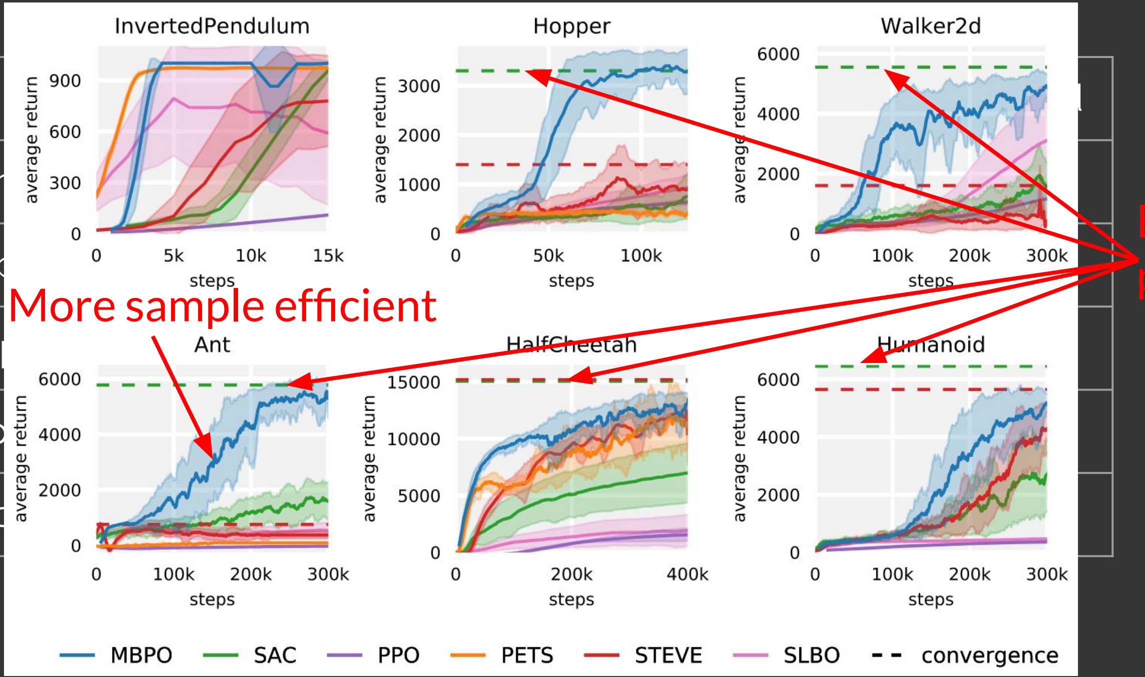
System identification vs model-based RL



Lambert, Nathan, et al. "Objective mismatch in model-based reinforcement learning." *arXiv preprint arXiv:2002.04523* (2020).

Why model-based RL?

Asym
Samp
Comp
Adap
Explo



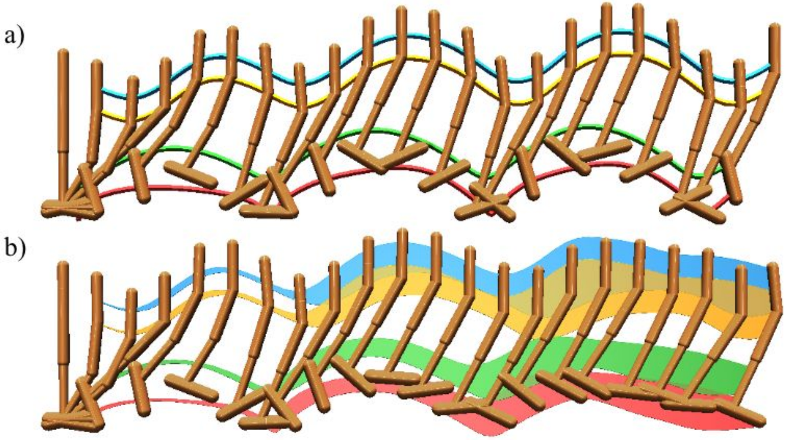
More sample efficient

Best asymptotic performance

Janner, Michael, et al. "When to trust your model: Model-based policy optimization." *Advances in Neural Information Processing Systems* 32 (2019).

Issues in model-based RL

Issues in MBRL



1. **Model bias**
    - **Overfitting in supervised learning**
        - model performs well on training
            - i.e. model overfits to training data
    - **Overfitting in model-based RL** - known as "model bias"
        - policy learning exploits model inaccuracies due to lack of training data
            - i.e. policy overfits to inaccurate dynamics model
2. **Compound error**
    - errors compound when making multi-step predictions
3. **Objective mismatch**
    - model training is a simple optimization problem disconnected from reward

Janner, Michael, et al. "When to trust your model: Model-based policy optimization." *Advances in Neural Information Processing Systems* 32 (2019).

Why uncertainty quantification in model-based RL?

- **Reduce model bias**
- **Exploration:** search where you haven't already observed
- **Risk-sensitive behaviour:** avoid places you haven't already observed

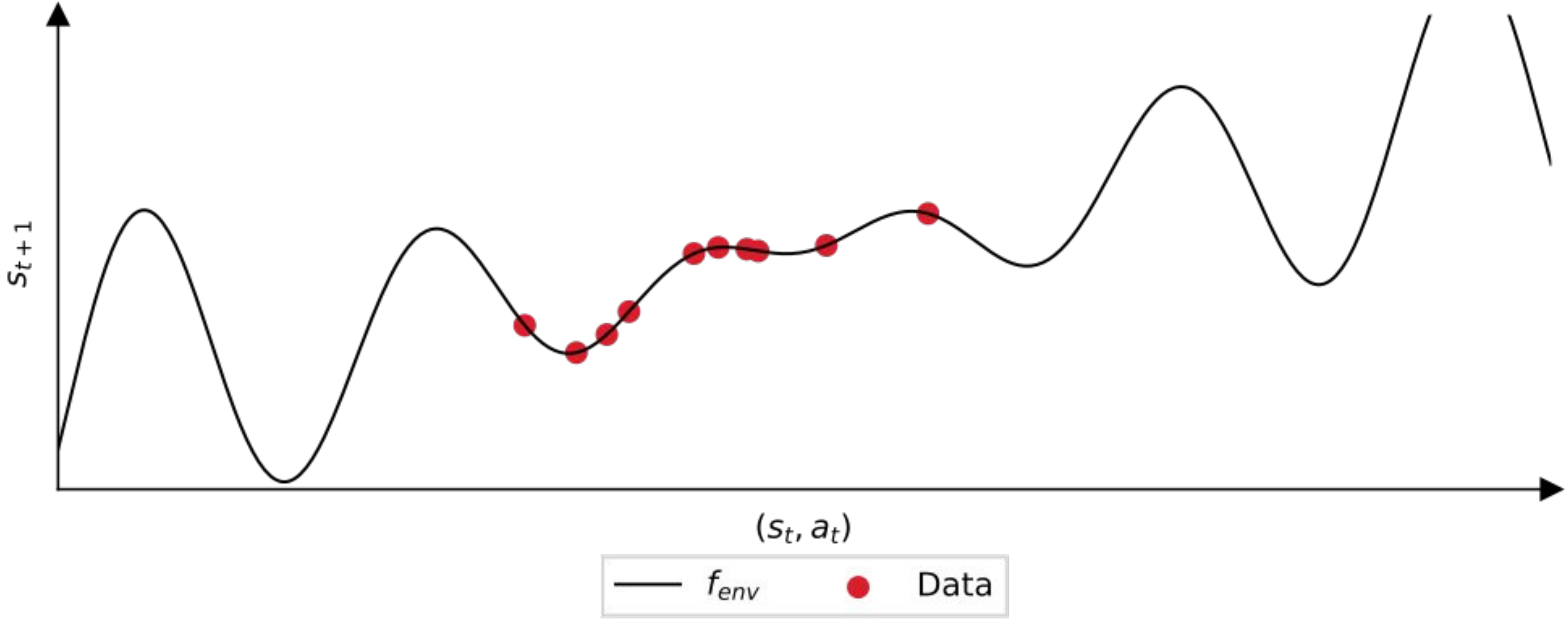
Uncertainty quantification

- **Aleatoric uncertainty**
  - **Transition noise** performing the same action in a given state does not always give same next state
  - **Measurement noise** imperfections in the measurement process
  - *cannot* be reduced

- **Epistemic uncertainty** - our model is not perfect
  - represents knowledge that we could know but do not know
  - *can* be reduced
    - collect more data and train on it

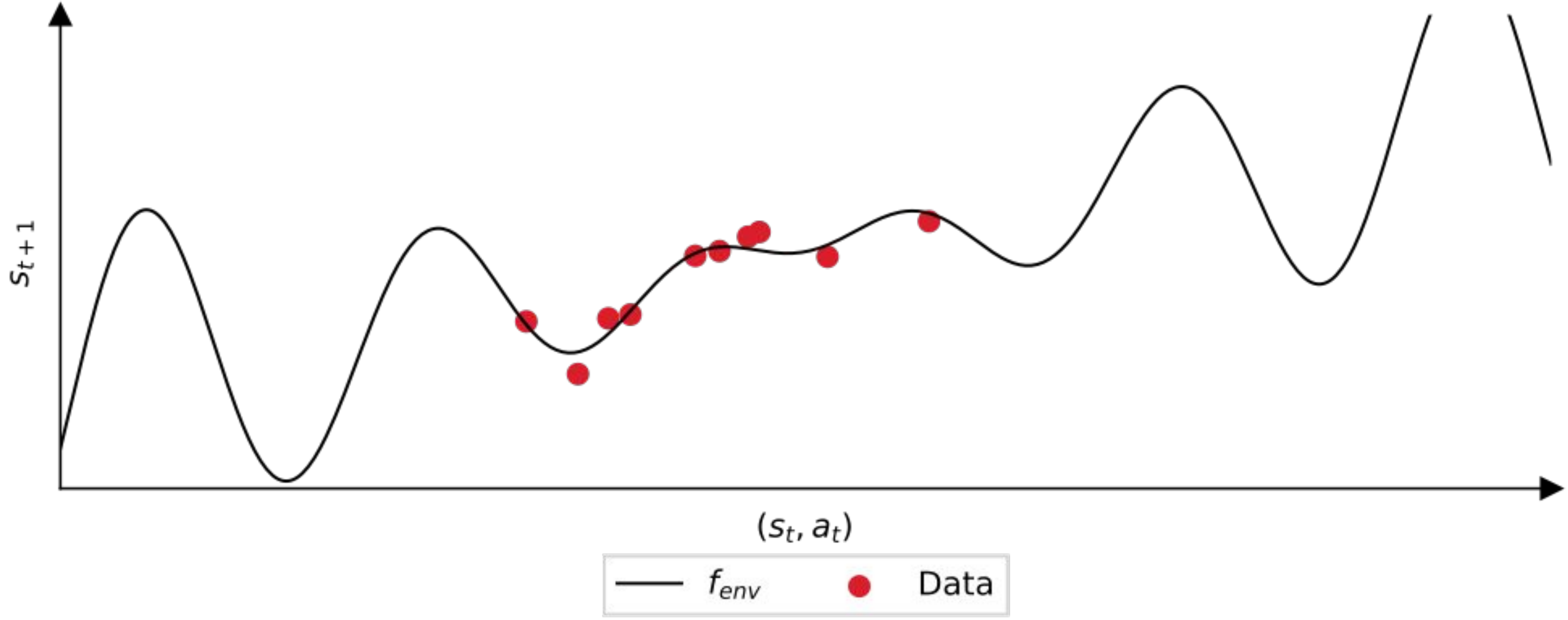
Deterministic environment

$$s_{t+1} = f_{\text{env}}(s_t, a_t)$$

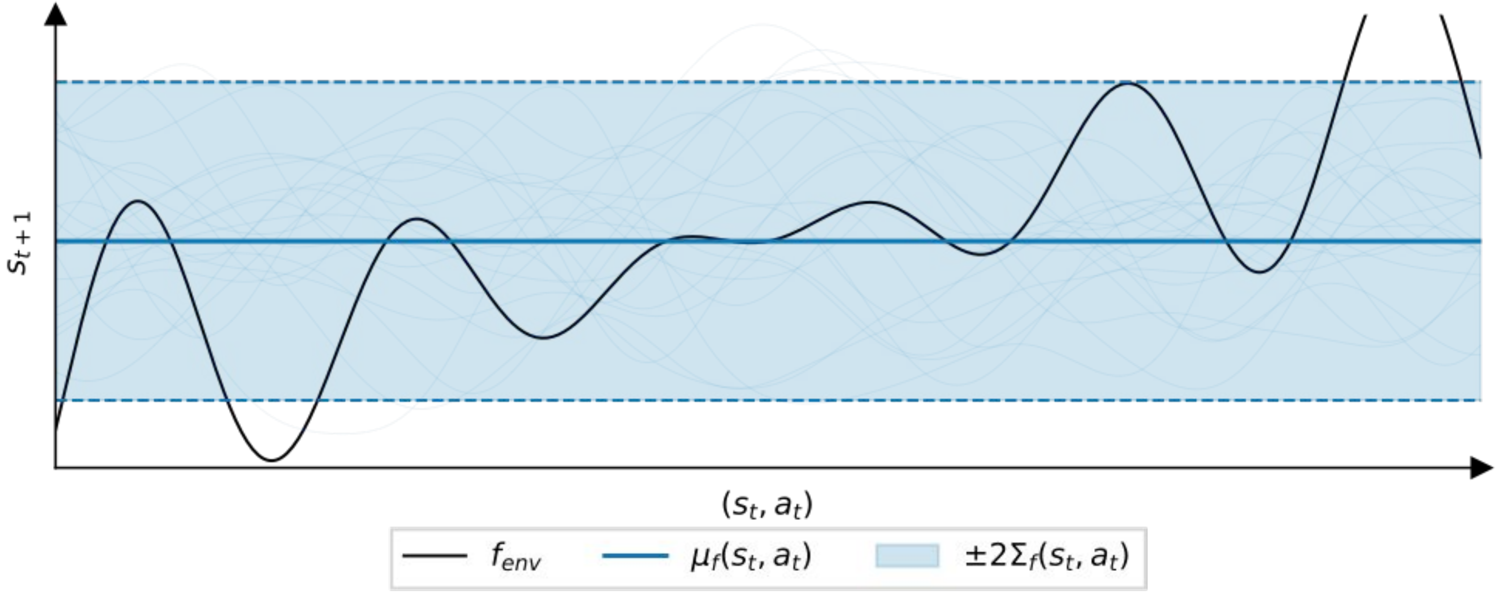


Stochastic environment

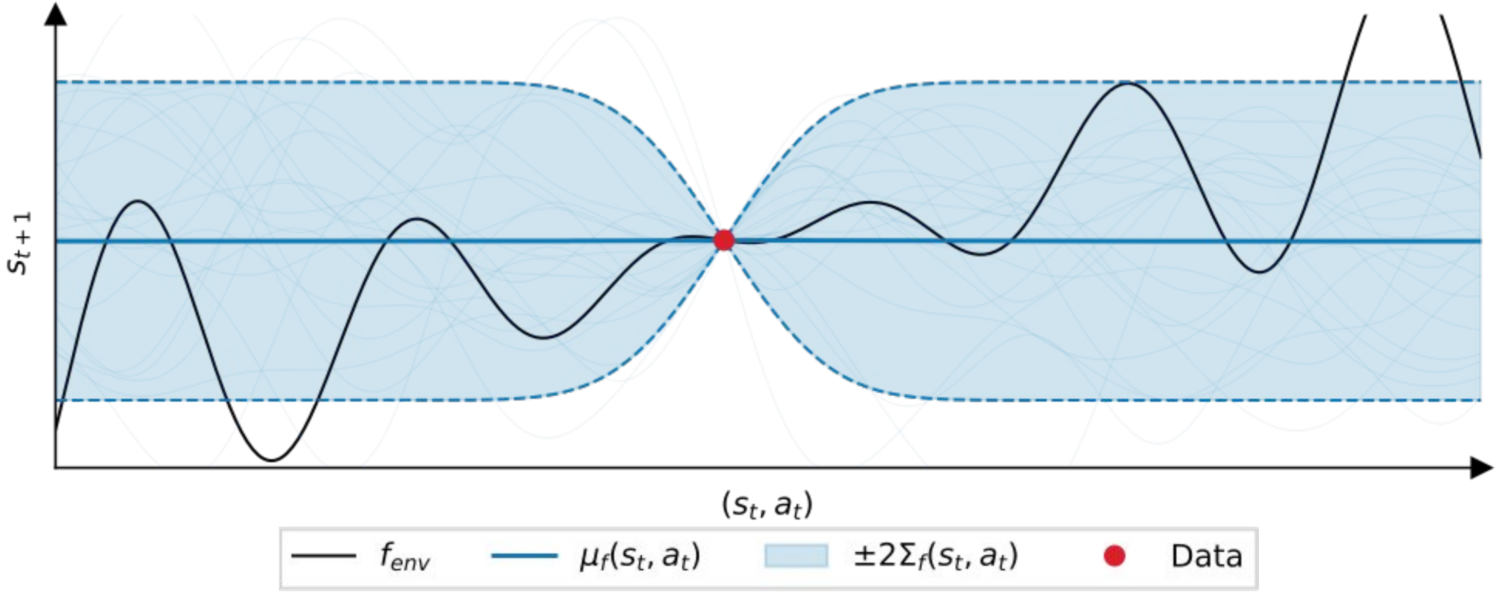
$$s_{t+1} = f_{\text{env}}(s_t, a_t) + \epsilon_t \qquad \epsilon_t \sim \mathcal{N}(0, \sigma_{\text{noise}})$$



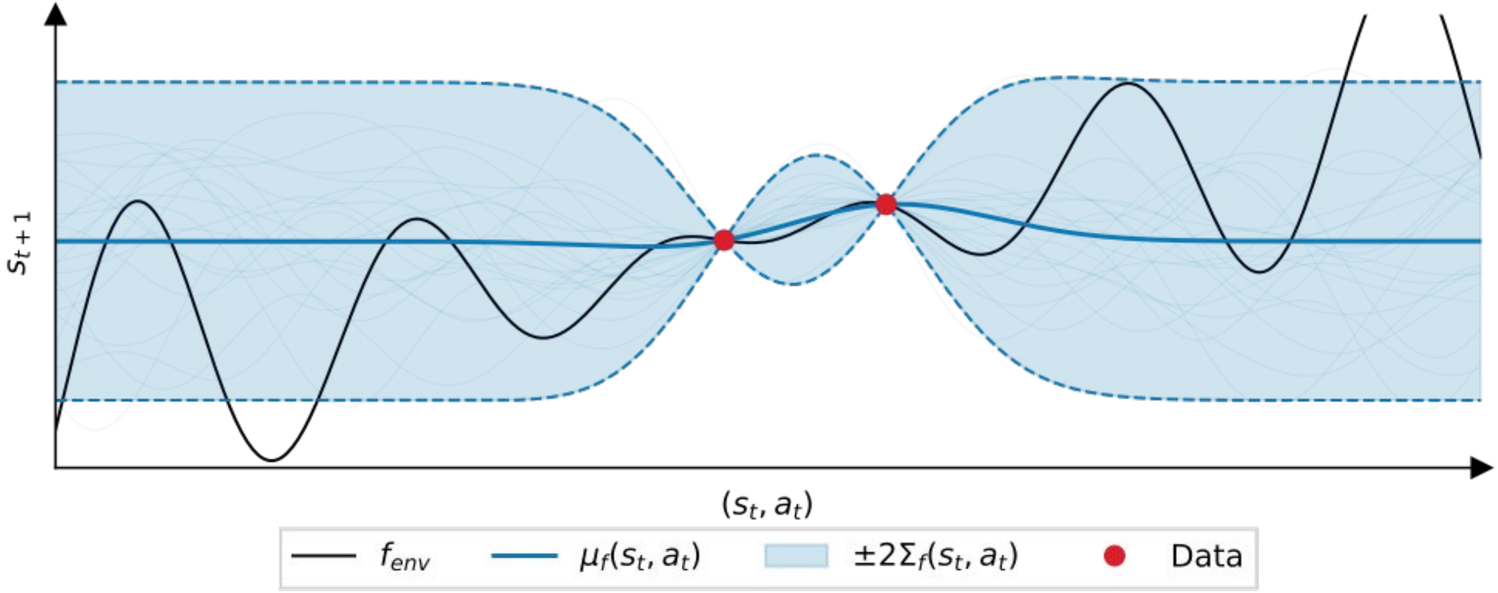
Epistemic uncertainty in model-based RL



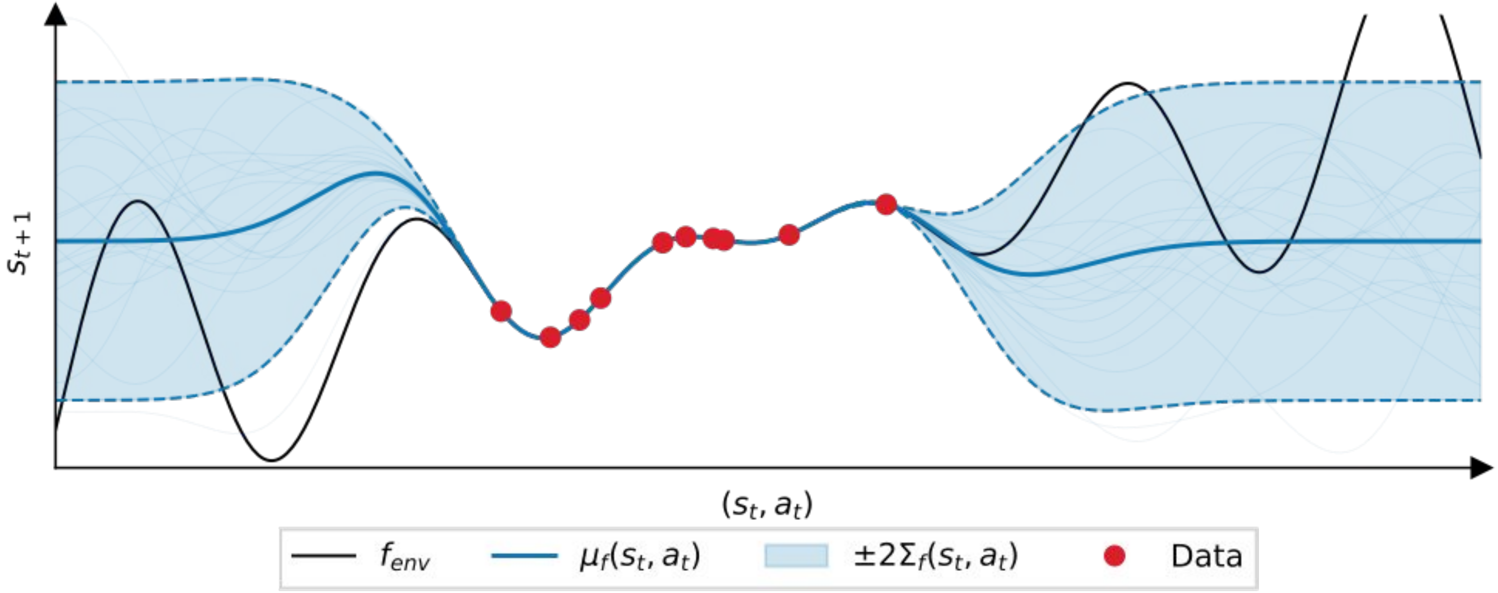
Epistemic uncertainty
in model-based RL



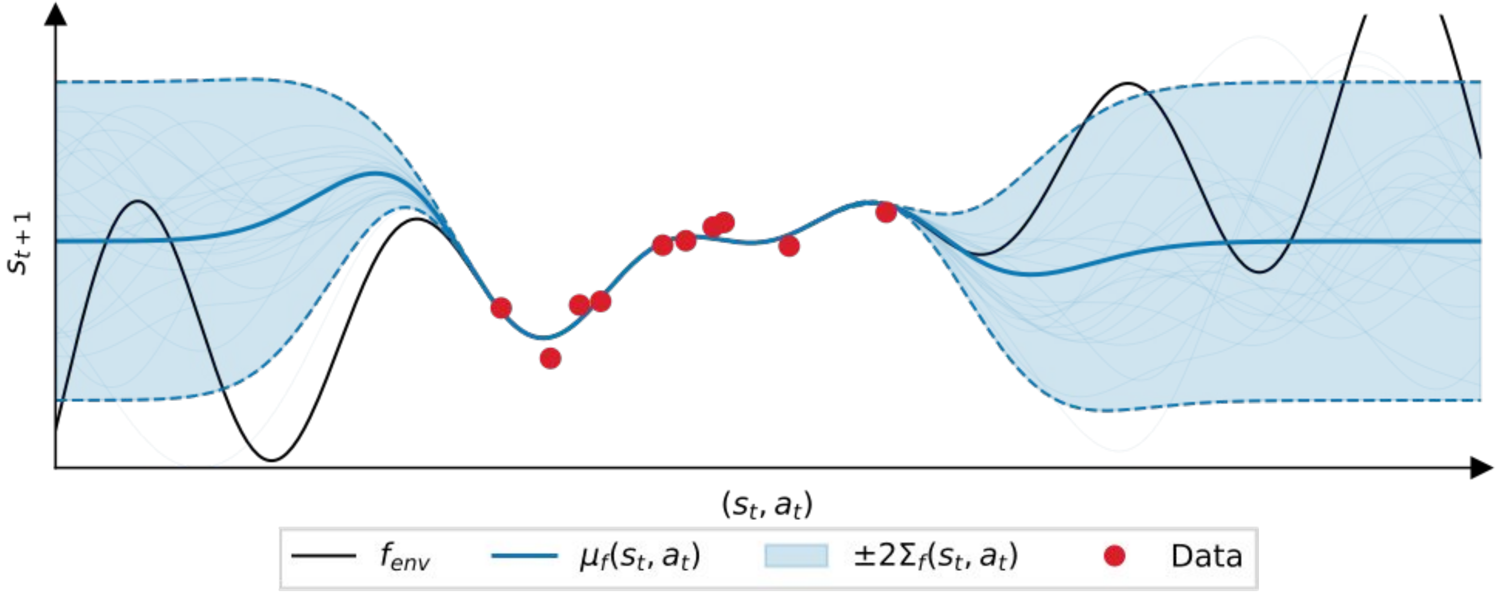
Epistemic uncertainty in model-based RL



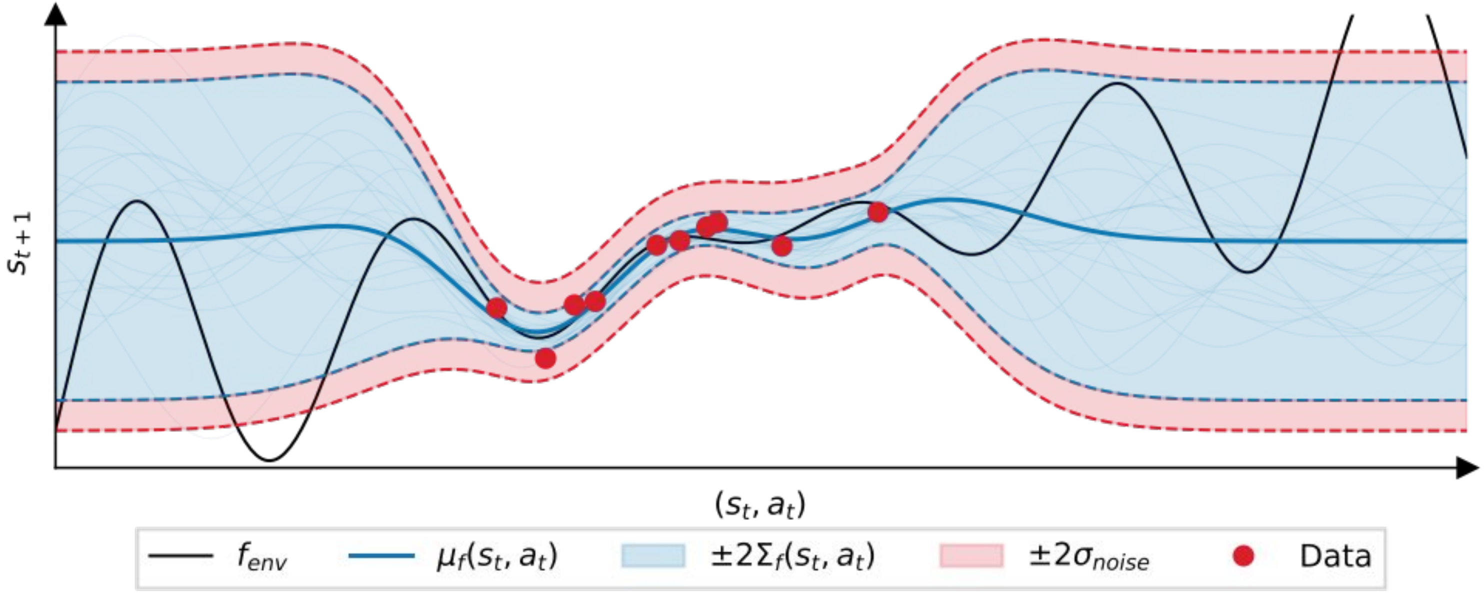
Epistemic uncertainty
in model-based RL



Aleatoric uncertainty
in model-based RL



Aleatoric uncertainty
in model-based RL



Uncertainty quantification in RL

Goal:

- Find policy π that maximises sum of rewards in expectation over?

$$J(f, \pi) = \mathbb{E}_{\epsilon_{0:\infty}}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$$

- Expectation is over transition noise, i.e. **aleatoric uncertainty**

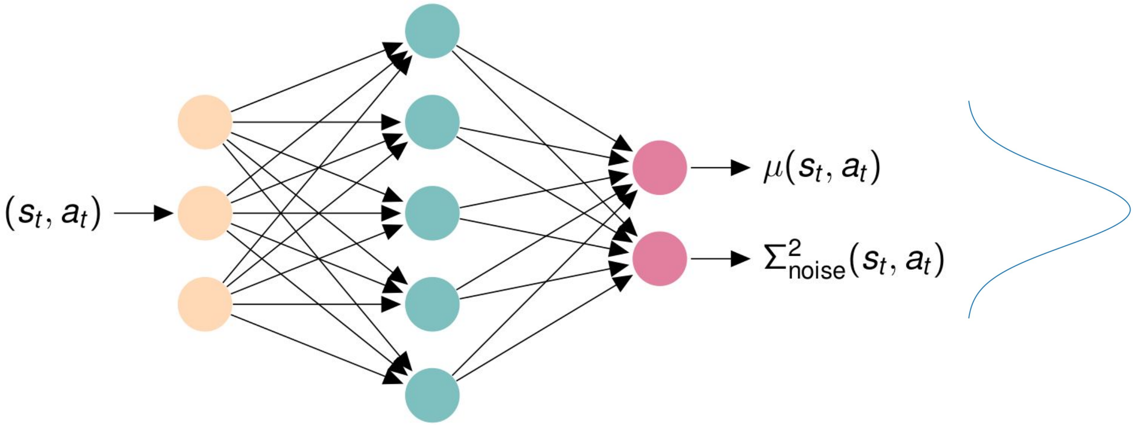
Uncertainty quantification in model-based RL

1. How to quantify uncertainty?
2. How to propagate uncertainty?
3. How to use uncertainty in decision-making (planning/policy learning)?

How to quantify uncertainty?

Probabilistic neural networks

- Capture **aleatoric uncertainty** (e.g. transition noise) with



- Train using negative log probability, i.e. maximum likelihood

$$p(s_{t+1} \mid s_t, a_t; \theta) = \mathcal{N}\left(s_{t+1} \mid \mu(s_t, a_t), \Sigma^2_{\text{noise}}(s_t, a_t)\right)$$

Ensemble of probabilistic neural networks

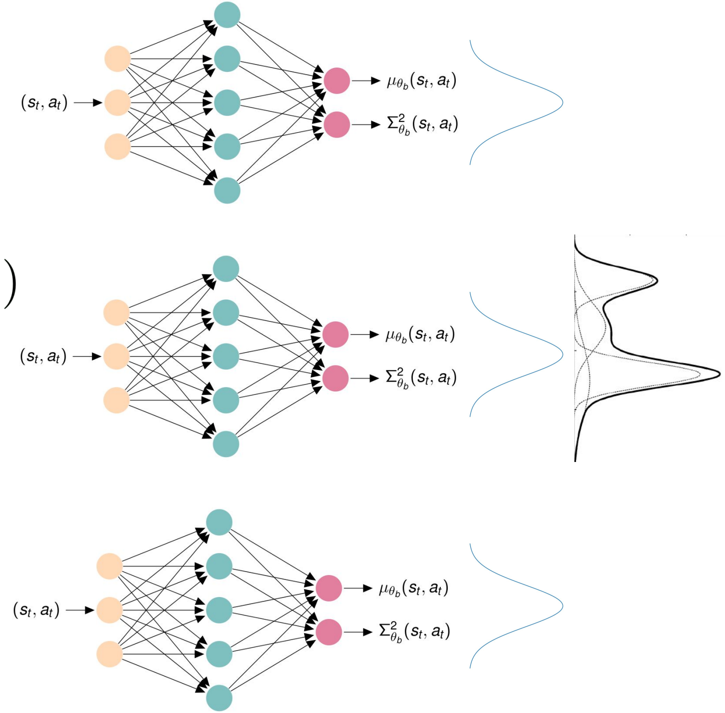
- Capture **epistemic uncertainty** with bootstrapped ensemble

$$f_\theta = \{f_{\theta_1}, \ldots, f_{\theta_B}\}$$

$$p(s_{t+1} \mid s_t, a_t, \theta_b) = \mathcal{N}\left(s_{t+1} \mid \mu_{\theta_b}(s_t, a_t), \Sigma_{\theta_b}(s_t, a_t)\right)$$

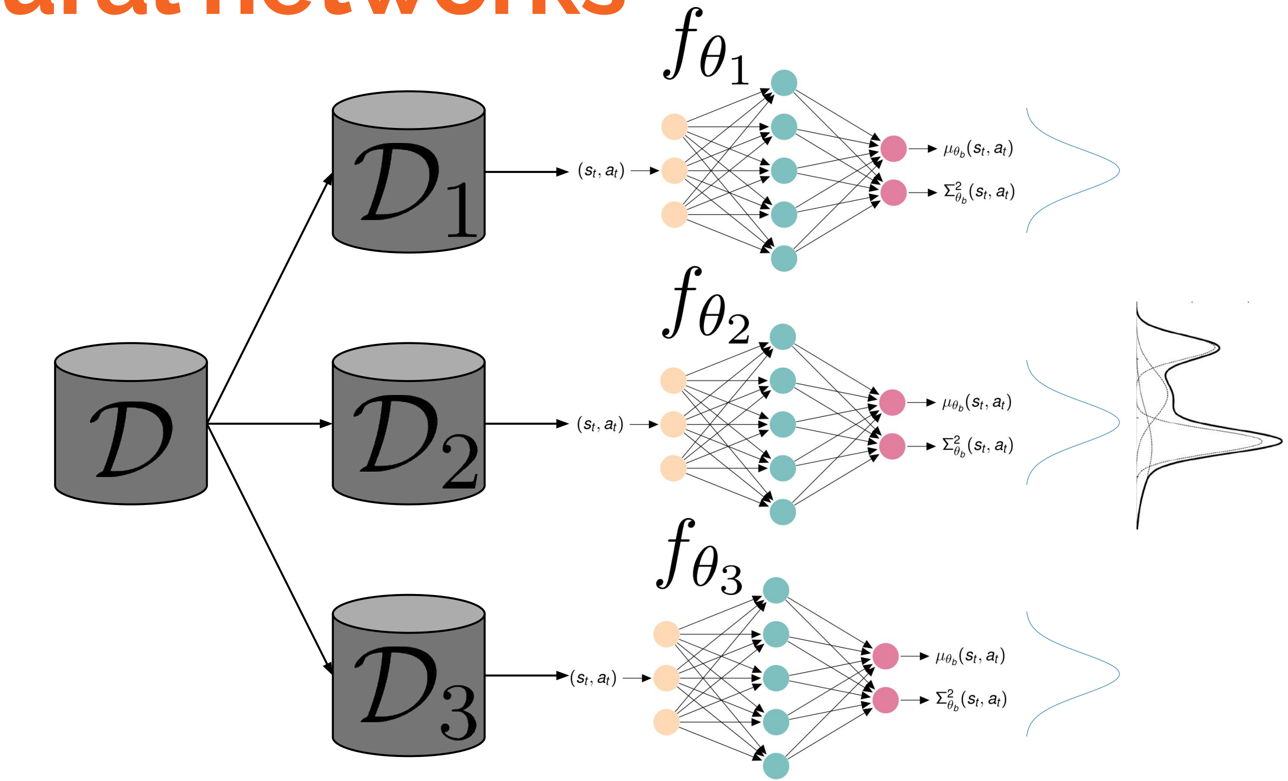
- Predictions are uniformly-weighted mixture

$$p(s_{t+1} \mid s_t, a_t) = \tfrac{1}{B} \textstyle\sum_{b=1}^{B} p(s_{t+1} \mid s_t, a_t, \theta_b)$$



(s_t, a_t)
$\mu_{\theta_b}(s_t, a_t)$
$\Sigma^2_{\theta_b}(s_t, a_t)$
(s_t, a_t)
$\mu_{\theta_b}(s_t, a_t)$
$\Sigma^2_{\theta_b}(s_t, a_t)$
(s_t, a_t)
$\mu_{\theta_b}(s_t, a_t)$
$\Sigma^2_{\theta_b}(s_t, a_t)$

Ensemble of probabilistic **neural networks**



$\mathcal{D}$
$\mathcal{D}_1$
$\mathcal{D}_2$
$\mathcal{D}_3$
f_{θ_1}
f_{θ_2}
f_{θ_3}
(s_t, a_t)
(s_t, a_t)
(s_t, a_t)
$\mu_{\theta_b}(s_t, a_t)$
$\Sigma^2_{\theta_b}(s_t, a_t)$
$\mu_{\theta_b}(s_t, a_t)$
$\Sigma^2_{\theta_b}(s_t, a_t)$
$\mu_{\theta_b}(s_t, a_t)$
$\Sigma^2_{\theta_b}(s_t, a_t)$

Bayesian uncertainty quantification

- Predictions at a new state-action input given by

$$p(s_{t+1} \mid s_t, a_t) = \int \underbrace{p(s_{t+1} \mid s_t, a_t, \theta)}_{\text{aleatoric unc.}} \; \underbrace{p(\theta \mid \mathcal{D})}_{\text{epistemic unc.}} \; \mathrm{d}\theta$$

- Capture **aleatoric uncertainty** with dist. over outputs (likelihood)

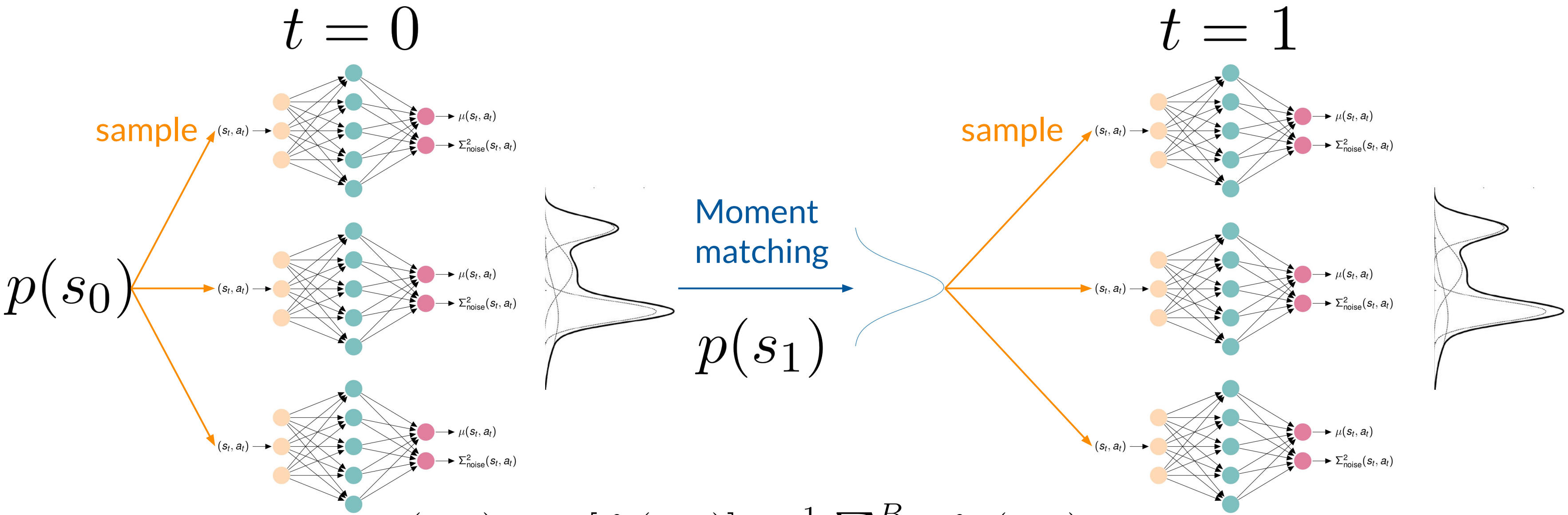
$$p(s_{t+1} \mid s_t, a_t, \theta) = \mathcal{N}(s_{t+1} \mid \mu(s_t, a_t), \Sigma_{\text{noise}}(s_t, a_t))$$

- Capture **epistemic uncertainty** with posterior dist. over model parameters

$$p(\theta \mid \mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{\int p(\mathcal{D}|\theta)p(\theta)\mathrm{d}\theta}$$

How to propagate uncertainty?

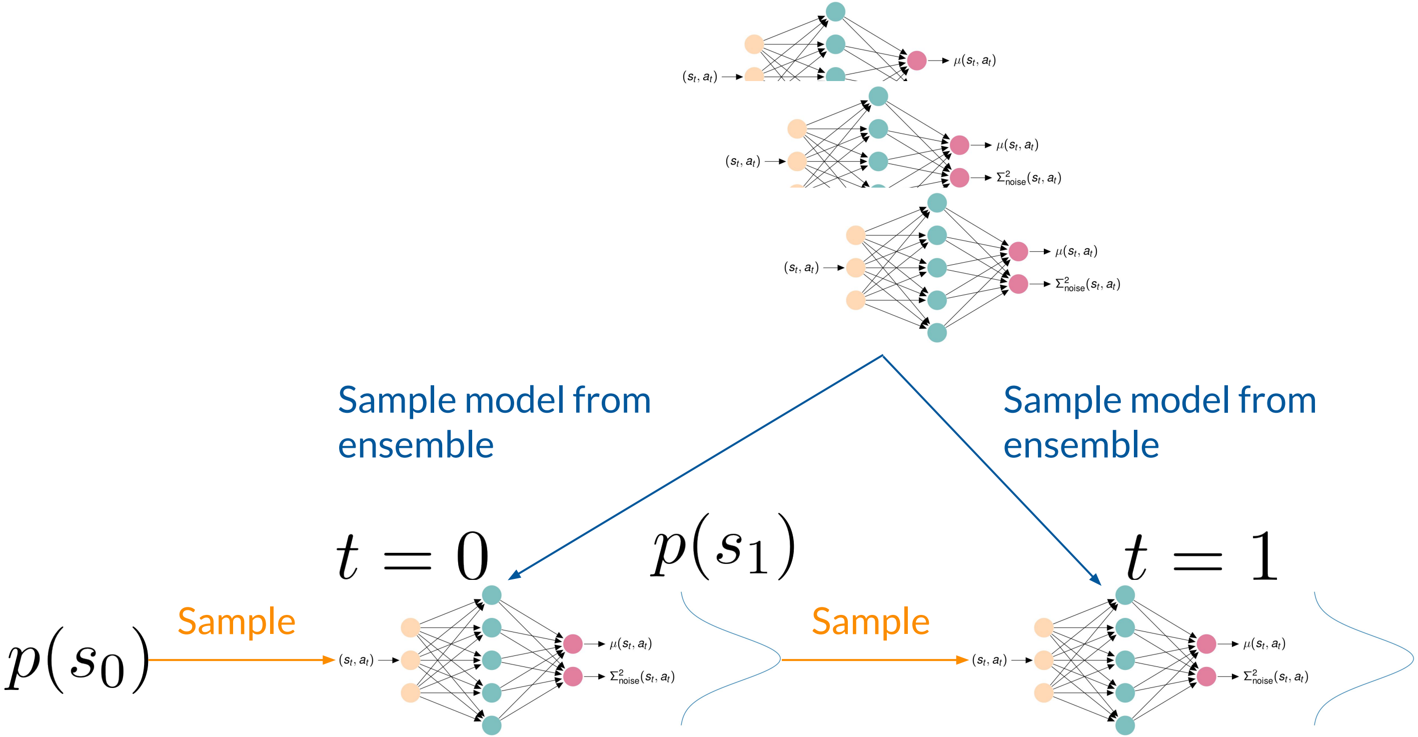
Uncertainty propagation via moment matching



$$\mu_f(s,a) = \mathbb{E}\left[f_\theta(s,a)\right] = \tfrac{1}{B}\sum_{b=1}^{B} f_{\theta_b}(s,a)$$

$$\Sigma_f^2(s,a) = \mathbb{V}\left[f_\theta(s,a)\right] = \tfrac{1}{B}\sum_{b=1}^{B}\left(\Sigma^2_{\theta_b}(s,a) + \mu^2_{\theta_b}(s,a)\right) - \mu^2_\theta(s,a)$$

Uncertainty propagation via trajectory sampling TS-1



Sample model from ensemble

Sample model from ensemble

$t = 0$

$p(s_1)$

$t = 1$

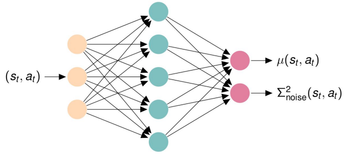
$p(s_0)$

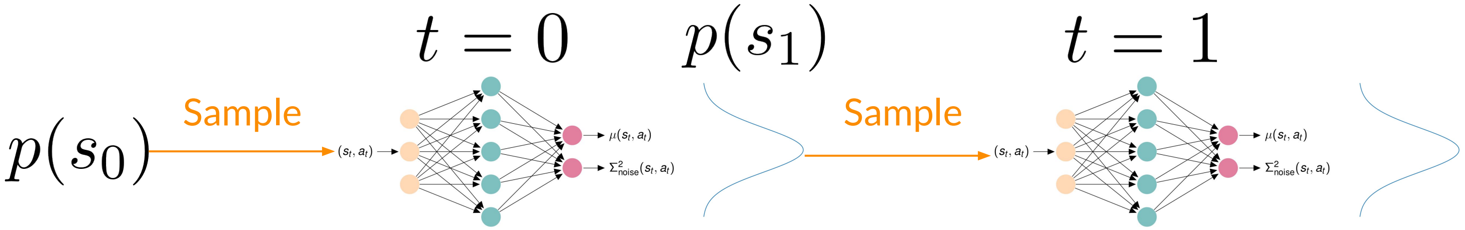
Sample

Sample

Uncertainty propagation via trajectory sampling TS-∞

Sample one dynamics model from ensemble





$p(s_0)$ Sample $t = 0$ $p(s_1)$ Sample $t = 1$

(s_t, a_t)
$\mu(s_t, a_t)$
$\Sigma^2_{\text{noise}}(s_t, a_t)$
(s_t, a_t)
$\mu(s_t, a_t)$
$\Sigma^2_{\text{noise}}(s_t, a_t)$

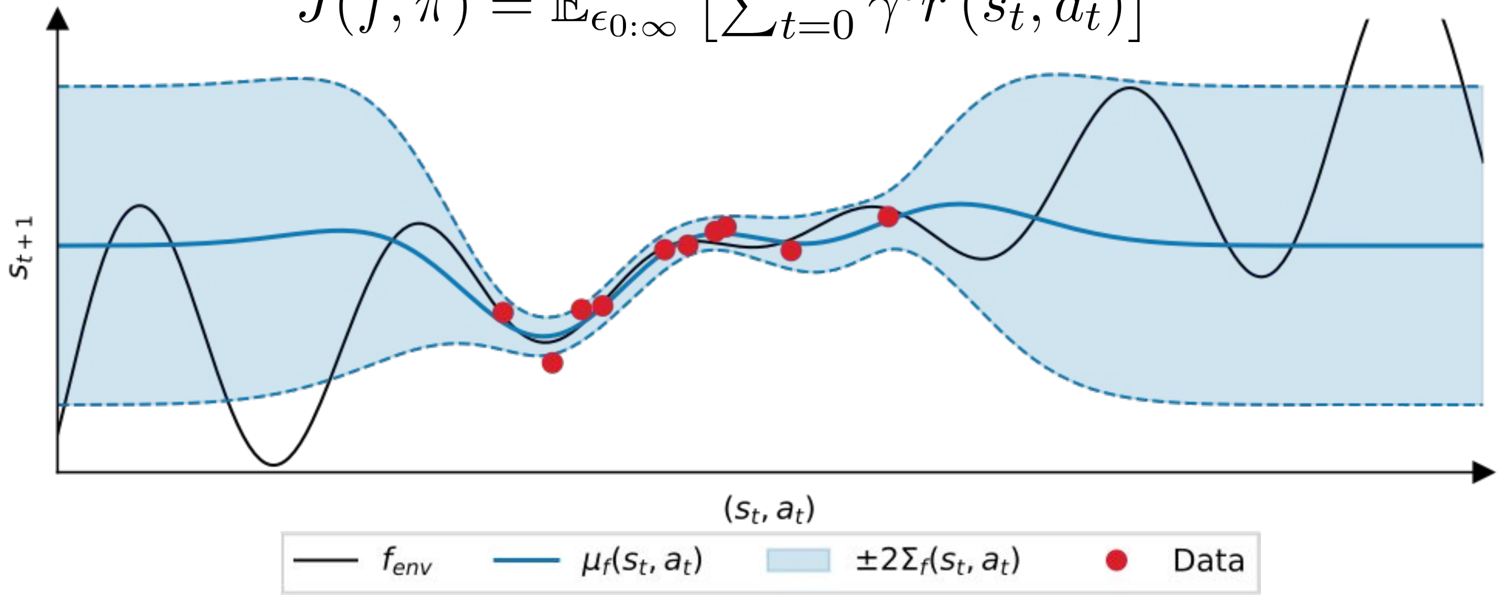
- TS-∞ captures time invariance of dynamics

Uncertainty-guided exploration

Uncertainty-guided exploration

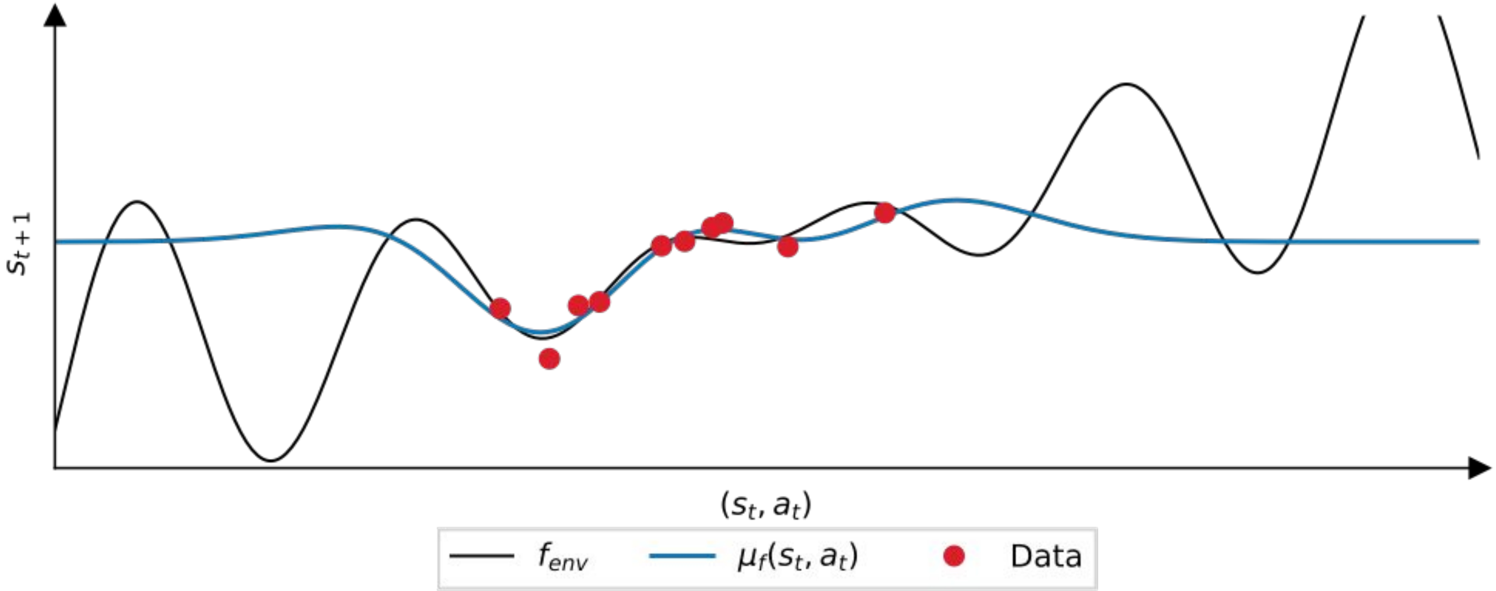
$$p(f \mid \mathcal{D} \cup (s_t, a_t)) = \mathcal{N}(f(s_t, a_t) \mid \mu_f(s, a), \Sigma_f(s_t, a_t))$$

$$J(f, \pi) = \mathbb{E}_{\epsilon_{0:\infty}} \left[\sum_{t=0}^{\infty} \gamma^t r\left(s_t, a_t\right) \right]$$



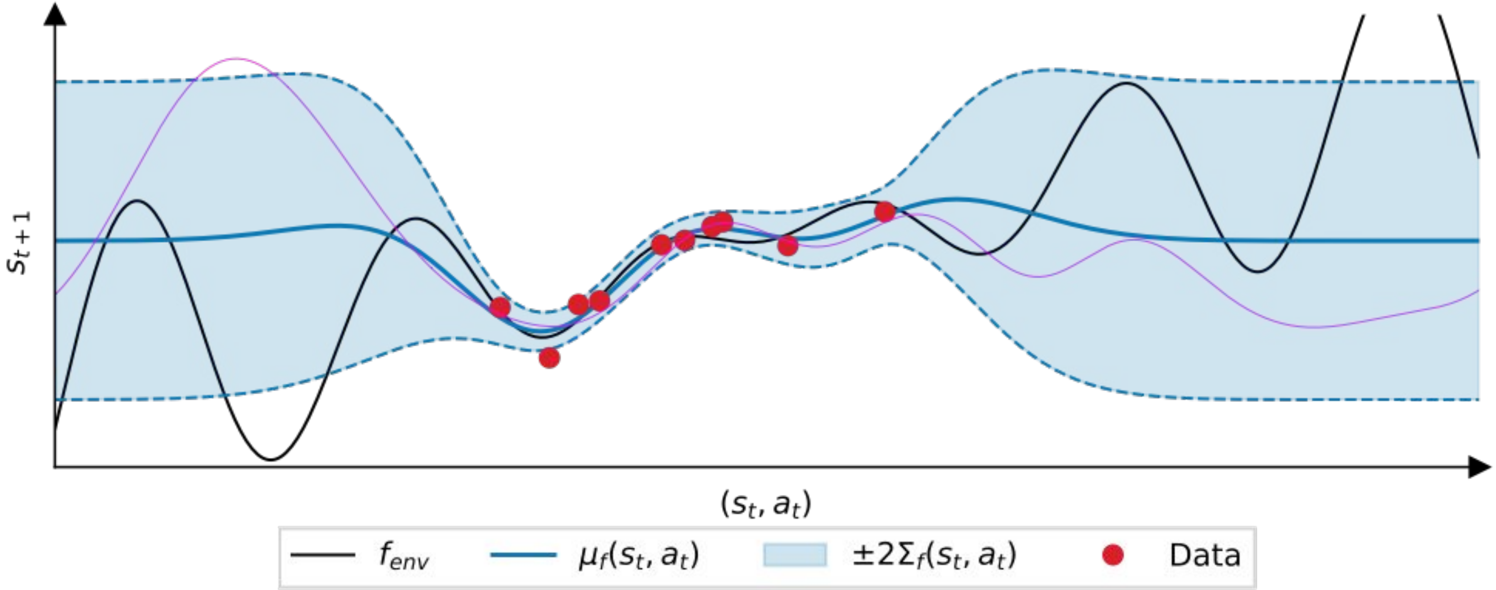
Exploration via greedy exploitation

$$\pi_{\text{greedy}} = \underset{\pi}{\operatorname{argmax}}\ \mathbb{E}_{f \sim p(f|\mathcal{D})}\left[J(f,\pi)\right]$$



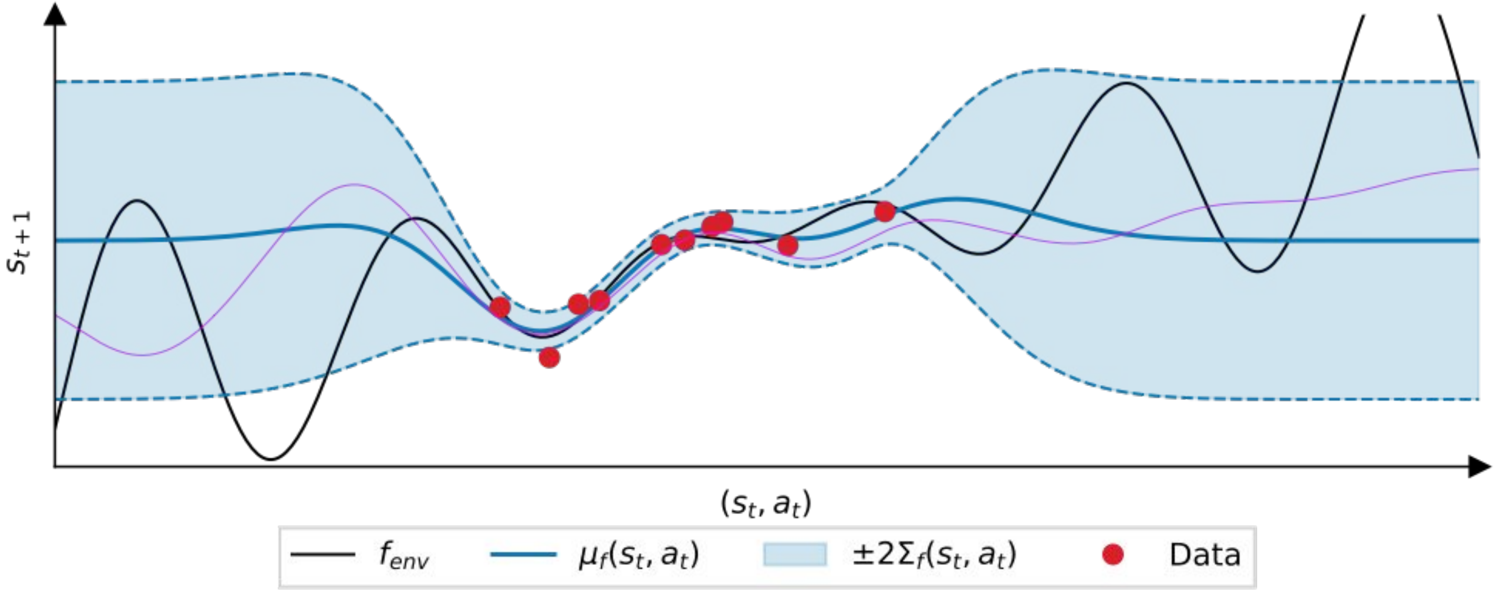
Exploration via Thompson sampling

$$\pi_{\mathrm{TS}} = \underset{\pi}{\operatorname{argmax}} \left[J(f, \pi) \right], \quad f \sim p(f \mid \mathcal{D})$$



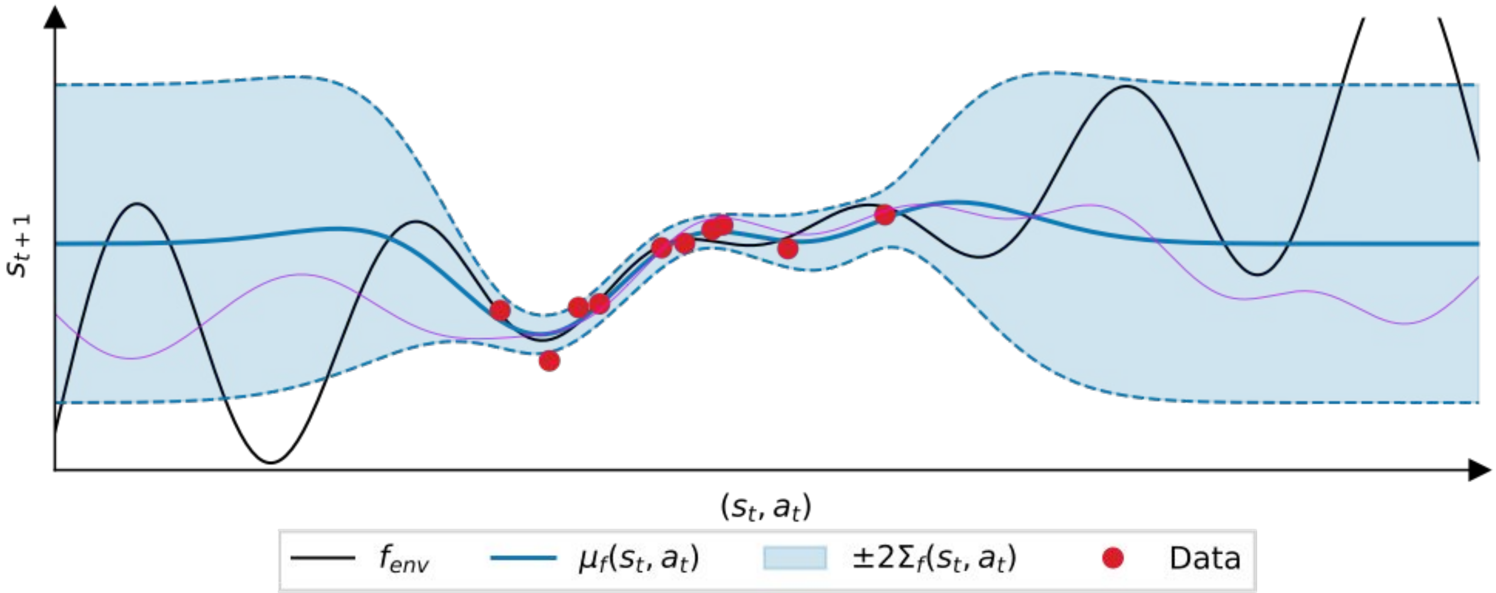
Exploration via Thompson sampling

$$\pi_{\text{TS}} = \underset{\pi}{\operatorname{argmax}} \left[J(f, \pi) \right], \quad f \sim p(f \mid \mathcal{D})$$



Exploration via Thompson sampling

$$\pi_{\mathrm{TS}} = \underset{\pi}{\operatorname{argmax}} \left[J(f, \pi) \right], \quad f \sim p(f \mid \mathcal{D})$$

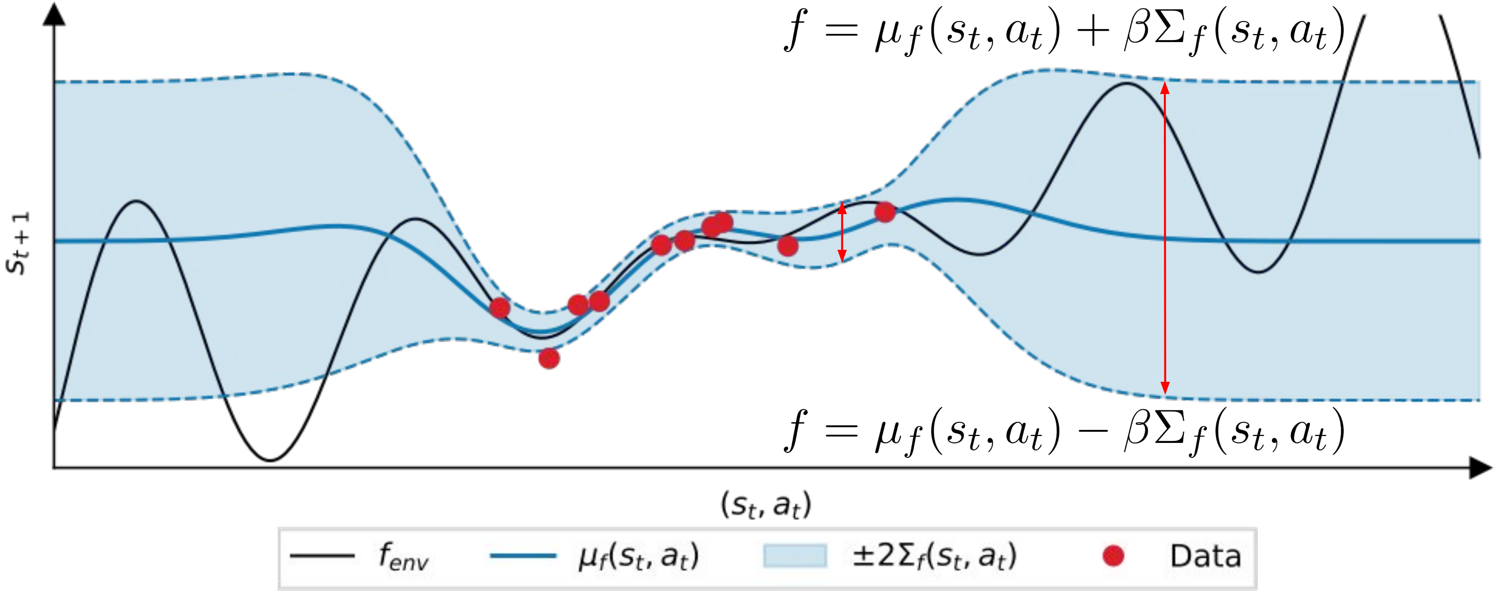


Exploration via
Upper Confidence Bound (UCB)

Optimism in the face of uncertainty

Exploration via Upper Confidence Bound (UCB)

$$\pi_{\mathrm{UCB}} = \underset{\pi}{\operatorname{argmax}} \max_{f \in \mathcal{M}} \left[J(f, \pi) \right] \qquad \mathcal{M} = \{ f \mid |f(s,a) - \mu_f(s,a)| \leq \beta \Sigma_f(s,a) \}$$



$$f = \mu_f(s_t, a_t) + \beta \Sigma_f(s_t, a_t)$$

$$f = \mu_f(s_t, a_t) - \beta \Sigma_f(s_t, a_t)$$

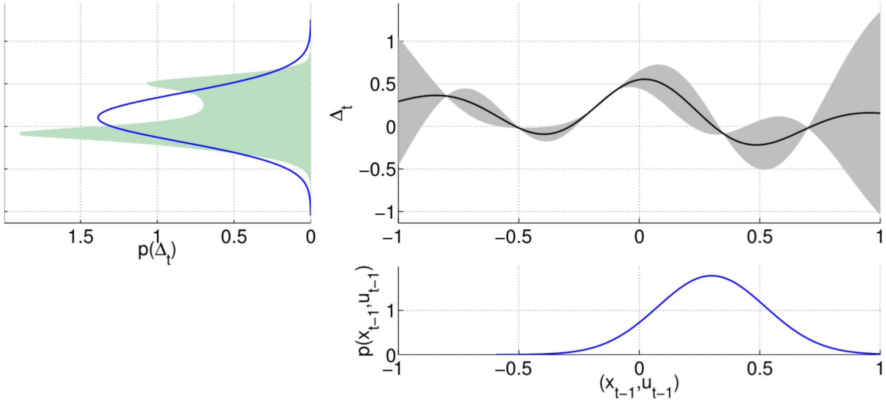
Other things to consider

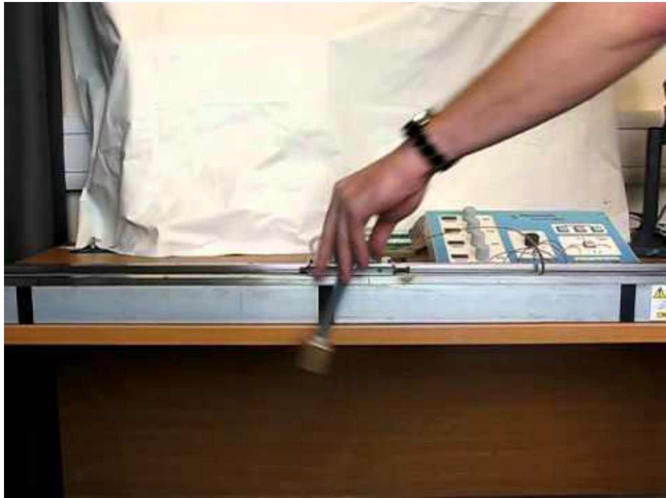
- **Try not to visit same state multiple times in episode**
- **Try not to revisit states seen in previous episodes**
  - unless needed for further exploration

Examples

PILCO: Probabilistic Inference for Learning cOntrol

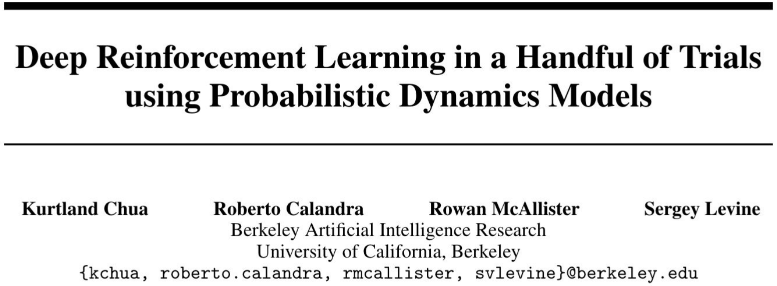
- **Dynamics model:** Gaussian processes
- **Uncertainty propagation:** moment matching
- **Decision making:**
  - greedy exploitation
  - learn RBF policy with closed-form objective





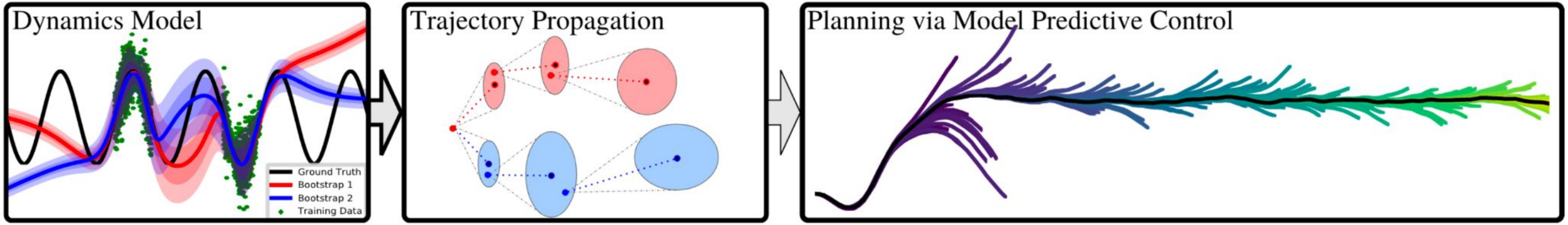
PETS: Probabilistic Ensembles with Trajectory Sampling

- **Dynamics model:** ensemble of probabilistic neural networks
- **Uncertainty propagation:** trajectory sampling
- **Decision making:**
  - planning via MPC (CEM)
  - greedy exploitation



Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models

Kurtland Chua **Roberto Calandra** **Rowan McAllister** **Sergey Levine**
Berkeley Artificial Intelligence Research
University of California, Berkeley
{kchua, roberto.calandra, rmcallister, svlevine}@berkeley.edu

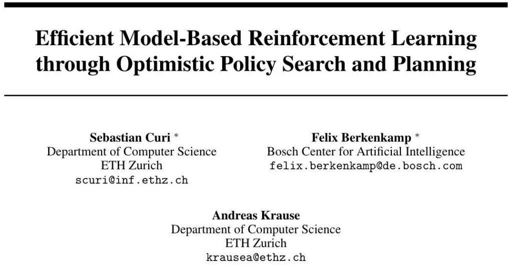


H-UCRL: Hallucinated Upper Confidence RL

- **Dynamics model:** ensemble of probabilistic neural networks
- **Uncertainty propagation:** N/A
- **Decision making:**
  - upper confidence bound (UCB)

$$\pi_{\text{H-UCRL}} = \operatorname*{argmax}_{\pi\in\Pi} \max_{\eta(\cdot)\in[-1,1]} [J(f,\pi)] \quad \text{s.t. } f = \mu_f(s_t,a_t) + \beta\Sigma_f(s_t,a_t)\eta(s_t,a_t)$$

  - combined offline policy-search with online planning



Efficient Model-Based Reinforcement Learning through Optimistic Policy Search and Planning

Sebastian Curi *
Department of Computer Science
ETH Zurich
scuri@inf.ethz.ch

Felix Berkenkamp *
Bosch Center for Artificial Intelligence
felix.berkenkamp@de.bosch.com

Andreas Krause
Department of Computer Science
ETH Zurich
krausea@ethz.ch

Key takeaways + food for thought

- **Methods are only as good as their uncertainty estimates!**
- **Decision-making under uncertainty can help relieve model bias**
- **Epistemic uncertainty can be used for exploration**
  - but important to disentangle epistemic and aleatoric uncertainties!
- **Epistemic uncertainty can be used for risk-sensitive behaviour**
  - i.e. keep policy in regions of dynamics with sufficient data

These slides were inspired by…

- Tutorial on Model-Based Methods in Reinforcement Learning @ ICML 2020 by Igor Mordatch and Jessica Hamrick
- Introduction to model-based RL by Chris Mutschler
- Deep RL Bootcamp Lecture 9 Model-based Reinforcement Learning by Chelsea Finn
- L6 Model-based RL (Foundations of Deep RL Series) by Pieter Abbeel
- Dissertation Talk: Synergy of Prediction and Control in Model-based Reinforcement Learning by Nathan Lambert

Thanks! Any questions?

Email:	aidan.scannell@aalto.fi
Website:	www.aidanscannell.com