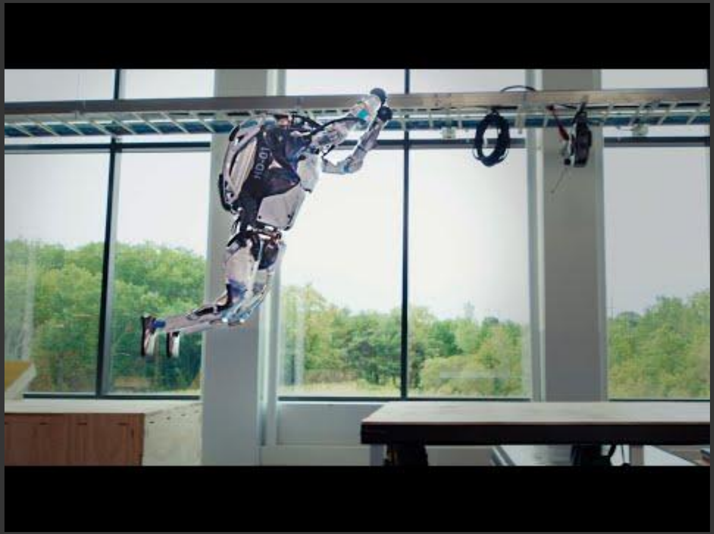
Model-based reinforcement learning under uncertainty: the importance of knowing what you don't know

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:

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

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 $r_{t+1} = f(s_t, a_t)$

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

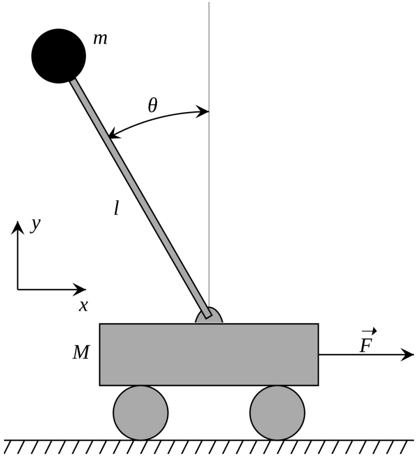
Inverse dynamics/transition model $a_t = f^{-1}(s_t, s_{t+1})$

Model of distance $d_{ij} = f_d(s_i, s_j)$

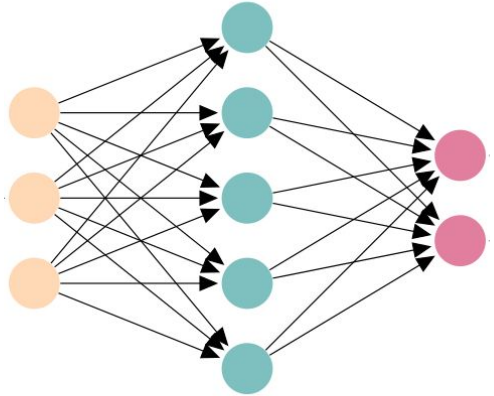
Model of future returns $G_t = Q(s_t, a_t)$

What's a model?

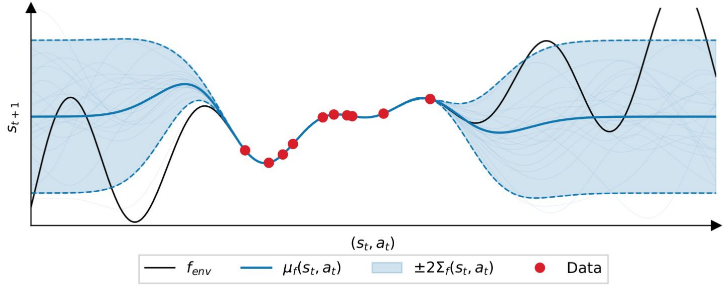
Physics based



Neural network

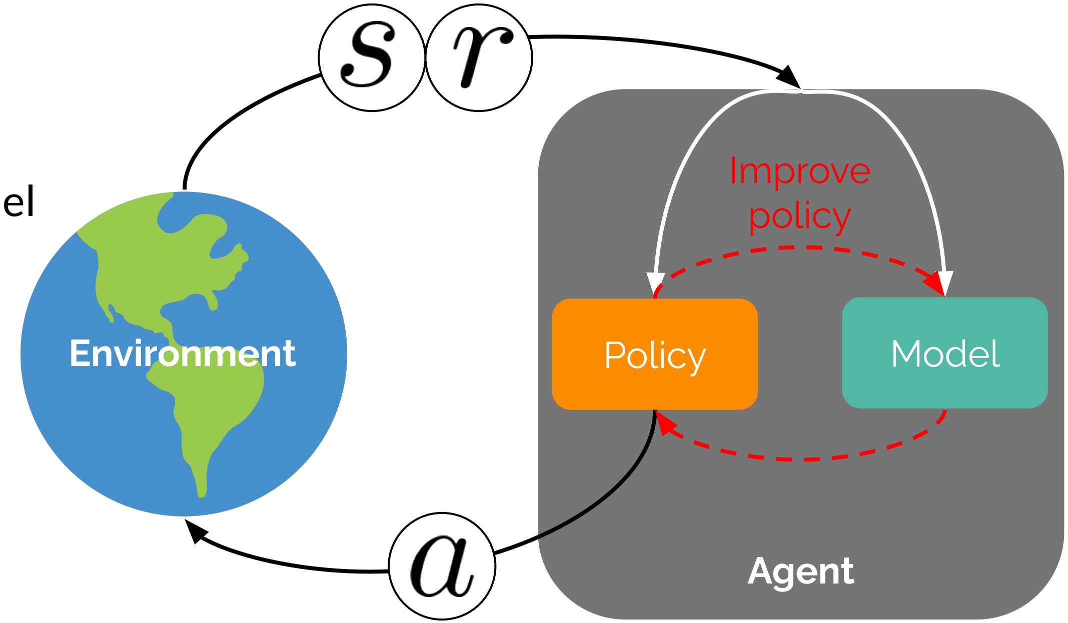


Gaussian process

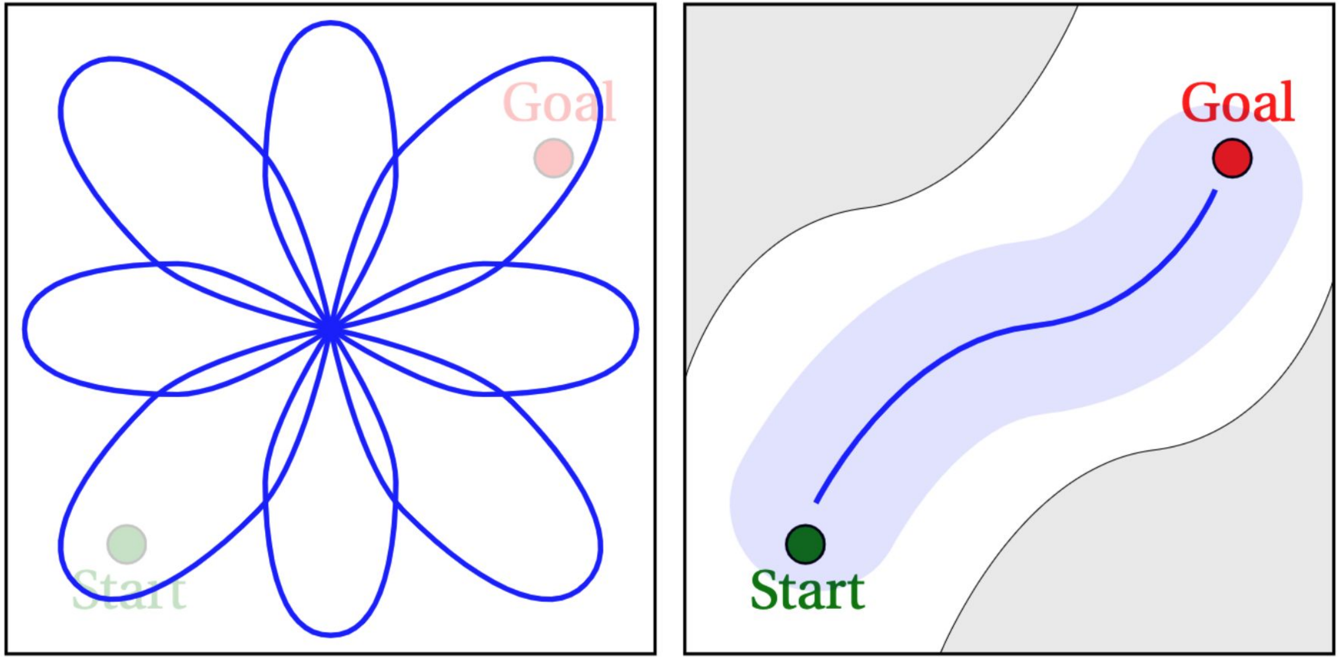


Model-based RL algorithm

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



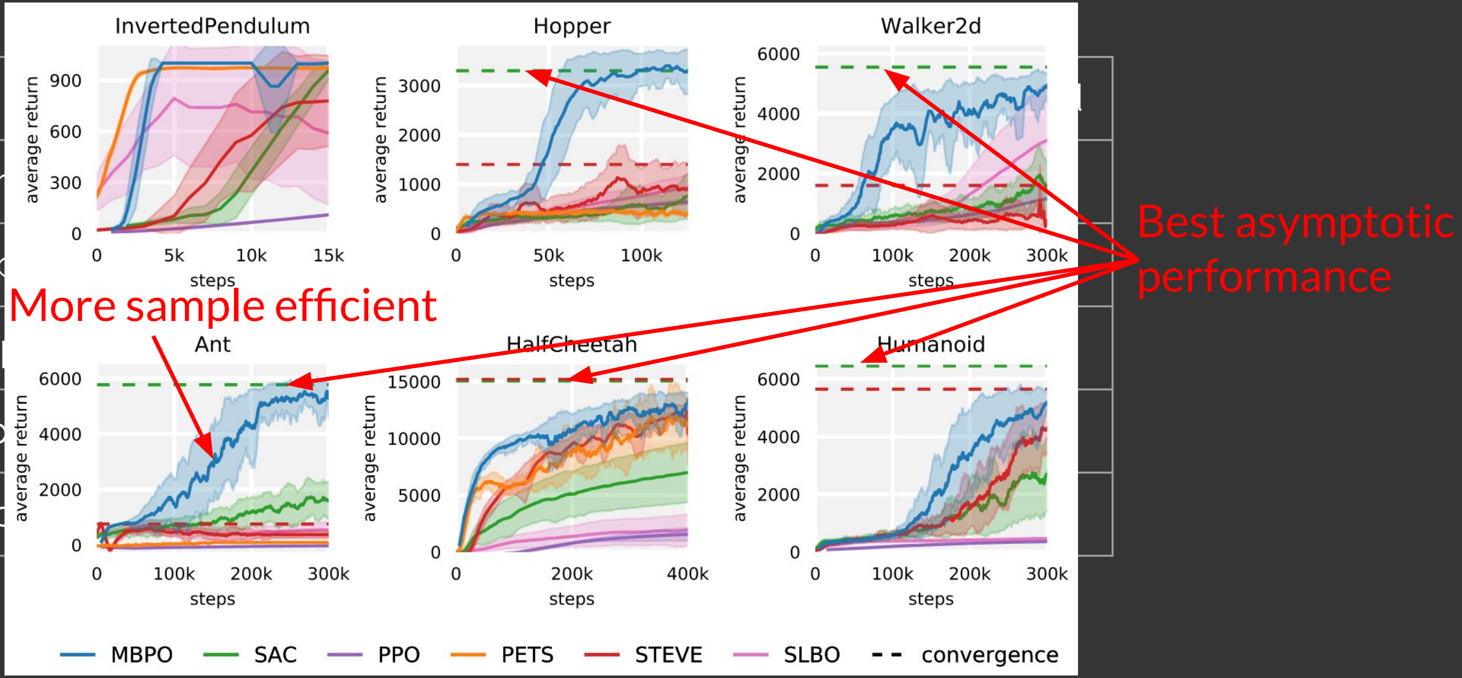
System identification vs model-based RL



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



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?

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

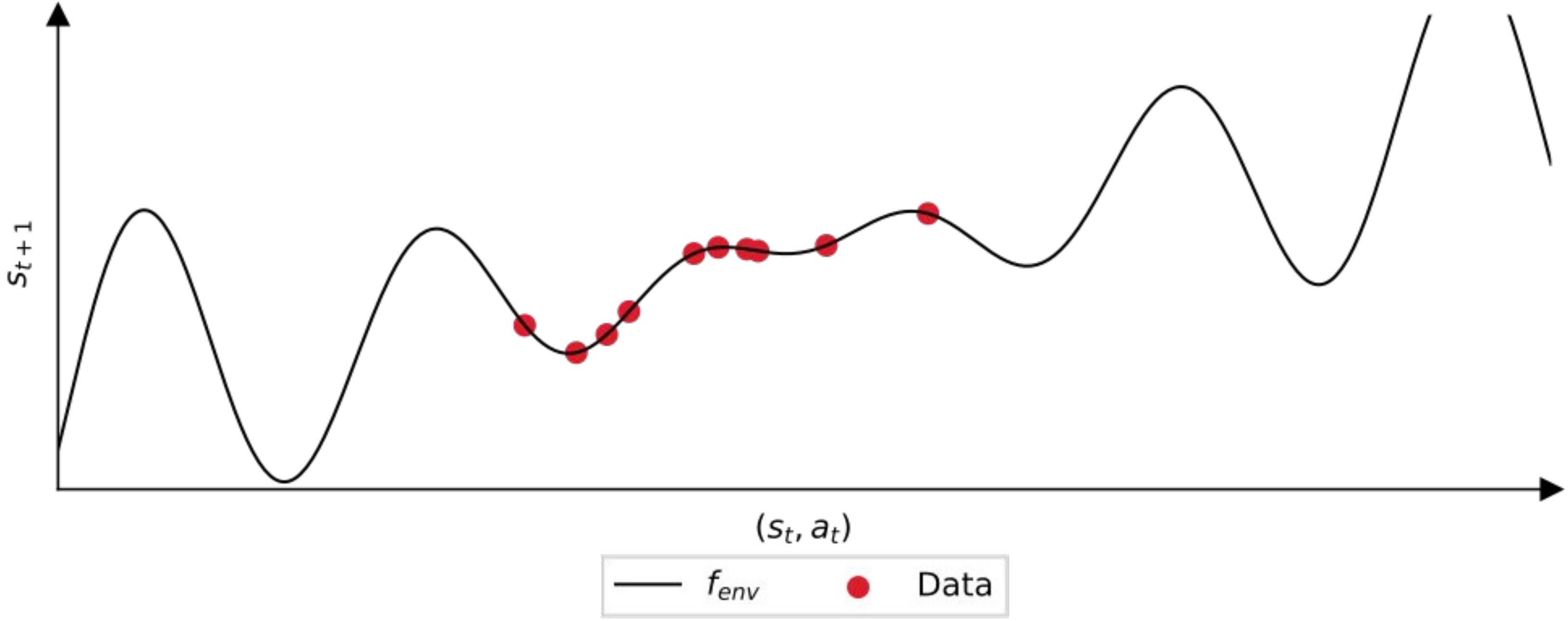
Uncertainty quantification

- **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

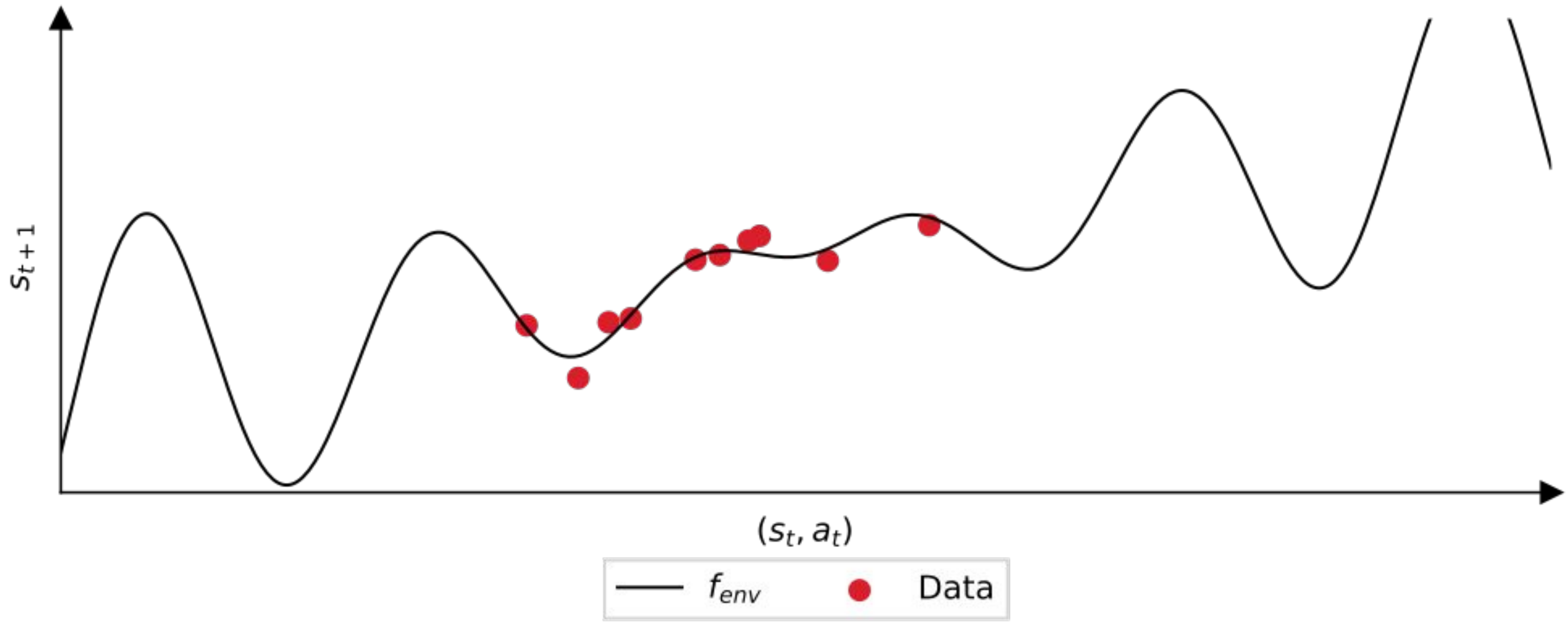
Deterministic environment

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

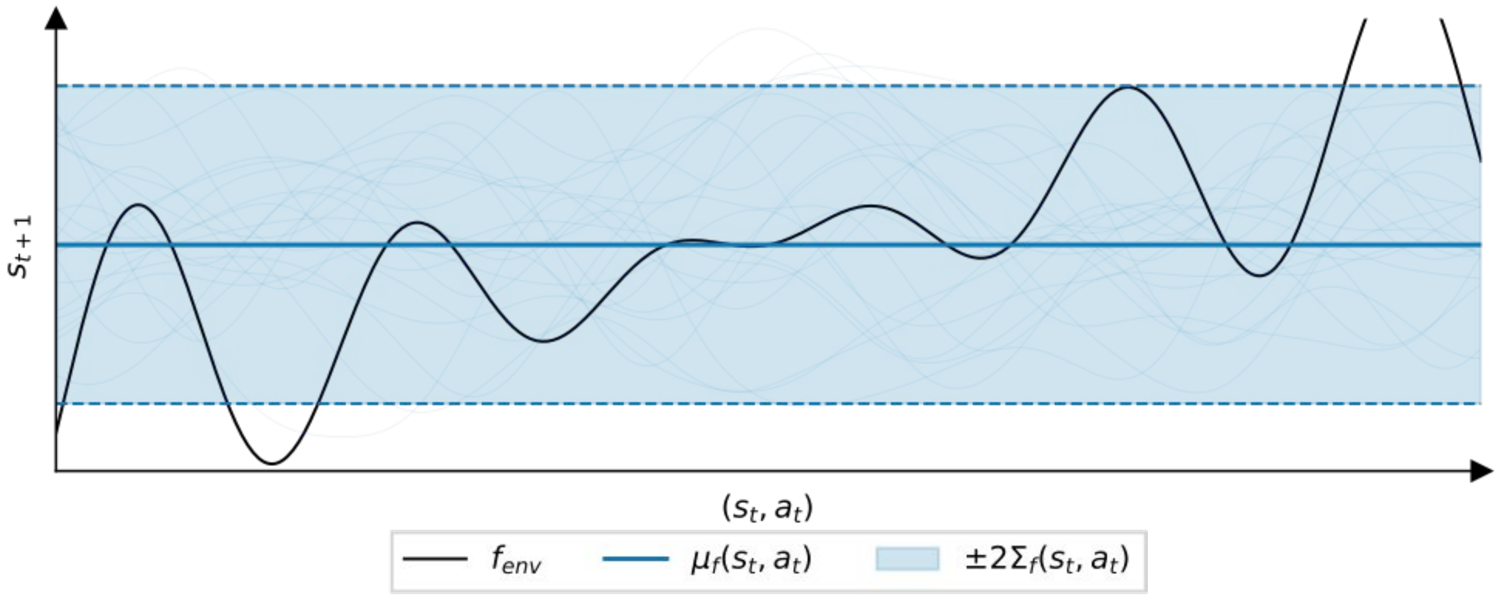


Stochastic environment

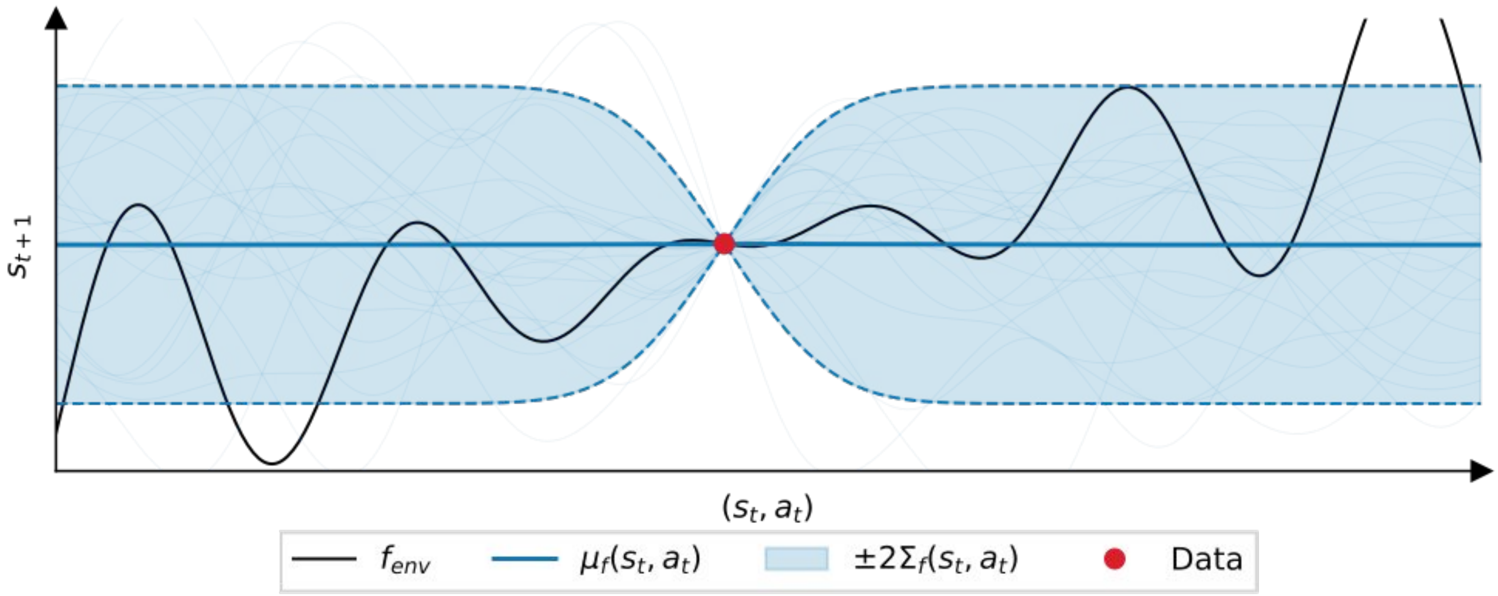
$$s_{t+1} = f_{\text{env}}(s_t, a_t) + \epsilon_t \qquad \epsilon_t \sim \mathcal{N}(0, \sigma_{\text{noise}})$$



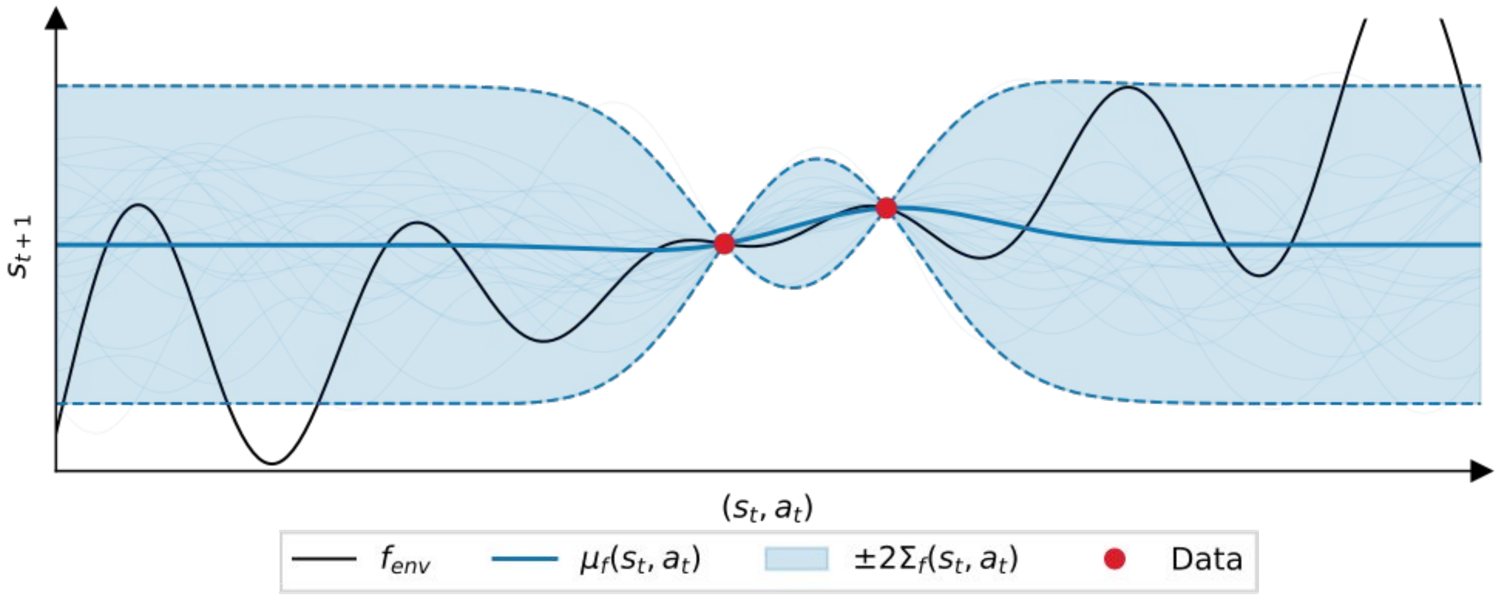
Epistemic uncertainty
in model-based RL



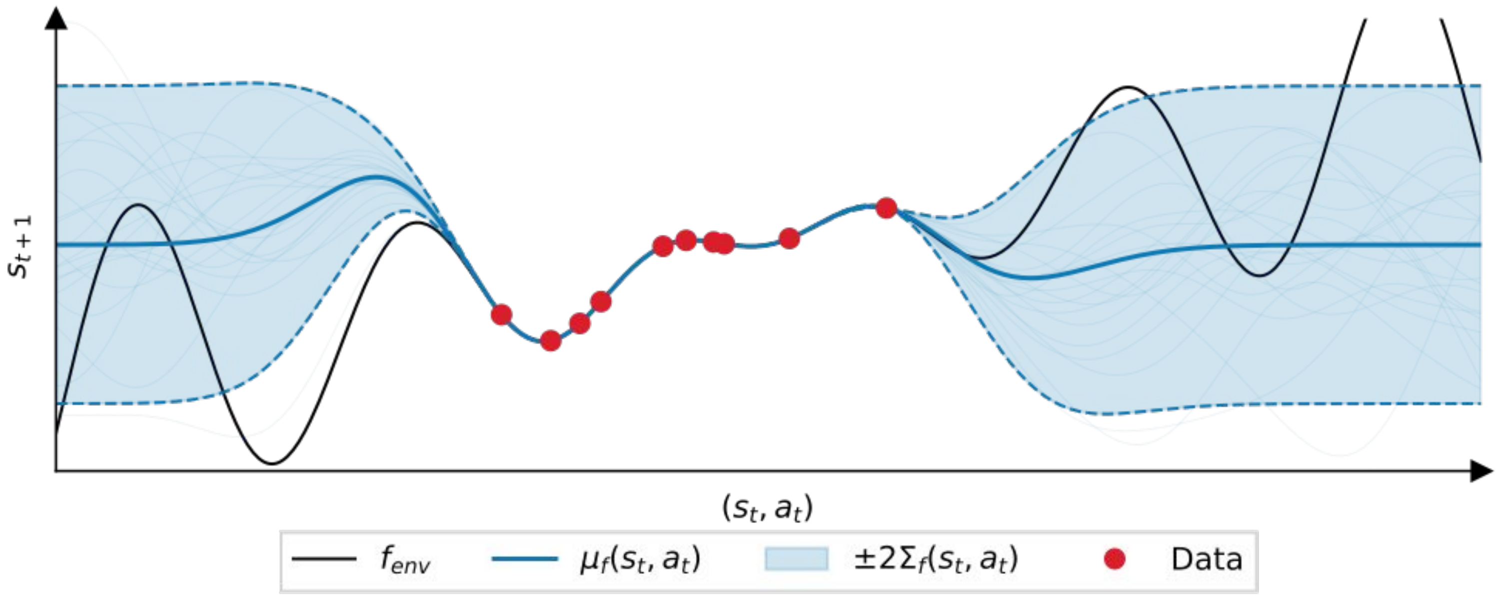
Epistemic uncertainty
in model-based RL



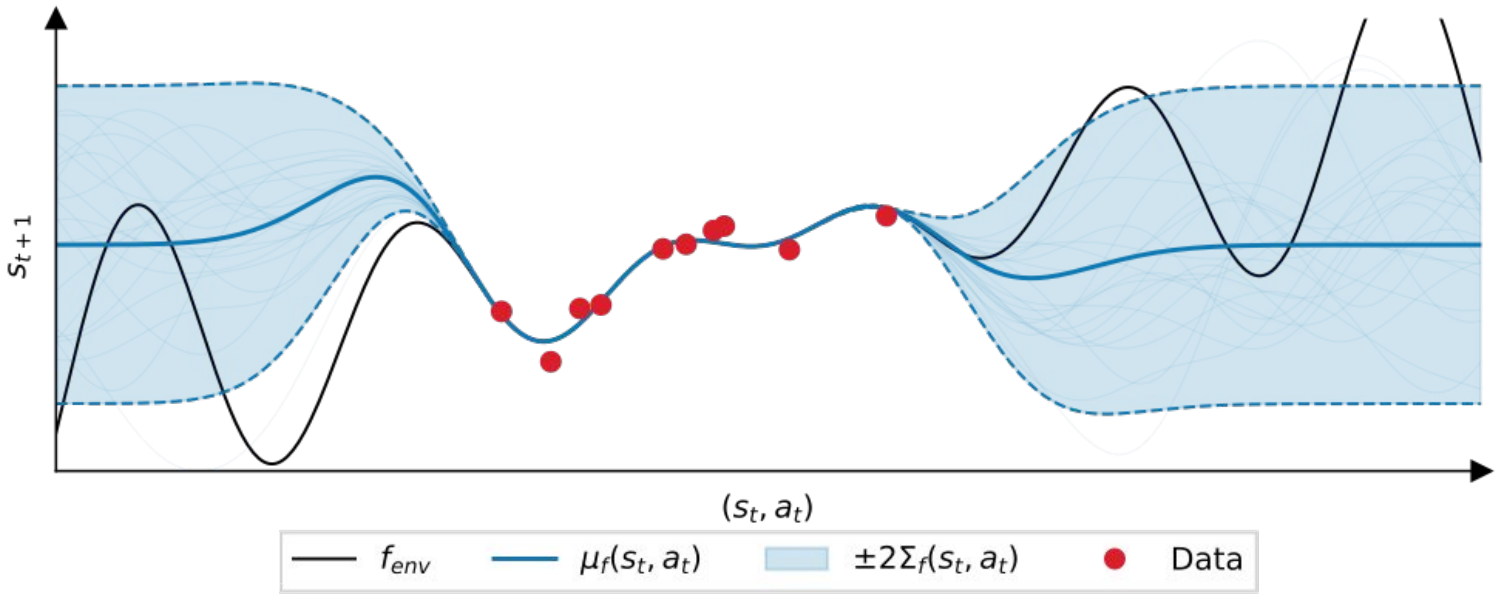
Epistemic uncertainty in model-based RL



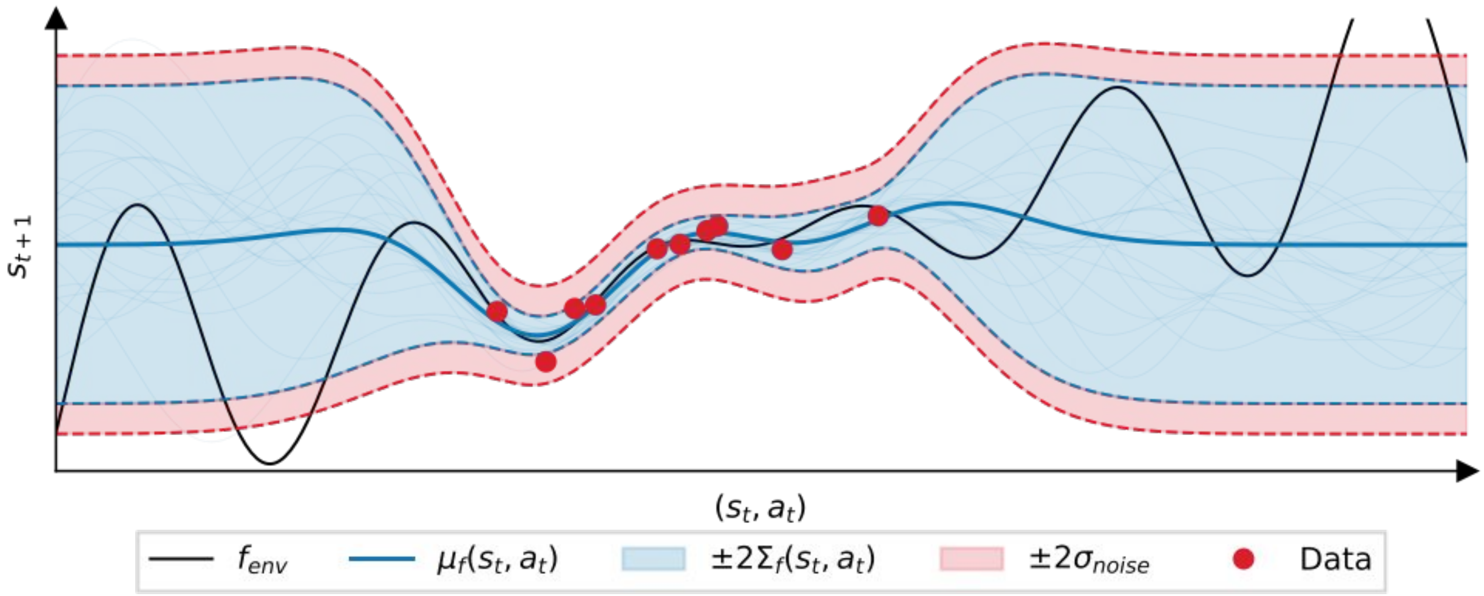
Epistemic uncertainty
in model-based RL



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**

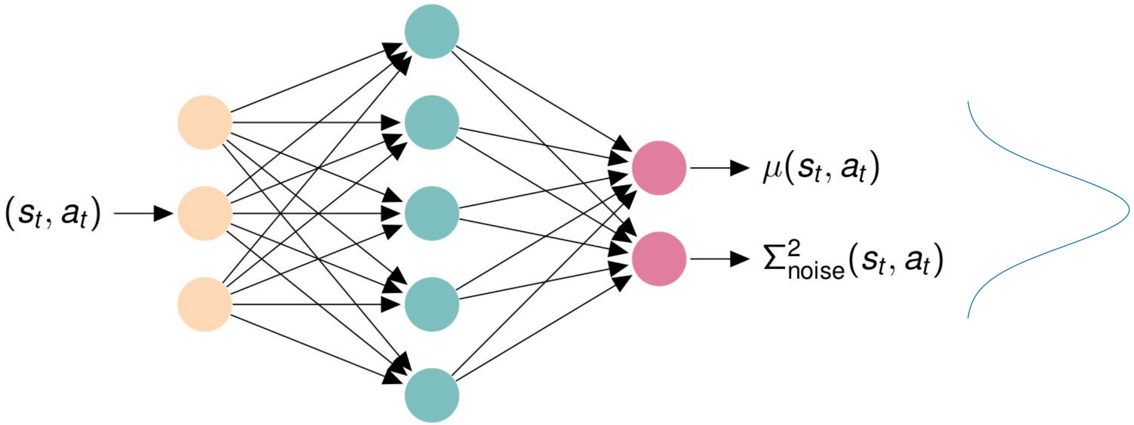
Uncertainty quantification in model-based RL

1. **How to quantify uncertainty?**
   - Gaussian processes (GPs)
   - Bayesian neural networks (BNNs)
2. **How to propagate uncertainty?**
   - sampling
   - moment-matching
3. **How to use uncertainty in decision-making (planning/policy learning)?**
   - exploration-exploitation trade-off
   - risk-sensitive behaviour

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**

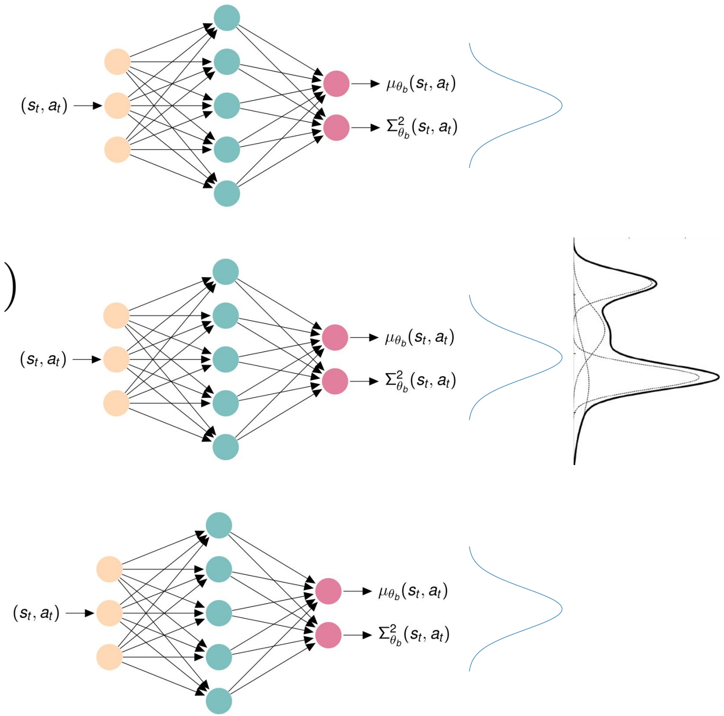
- Capture **epistemic uncertainty** with bootstrapped ensemble

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

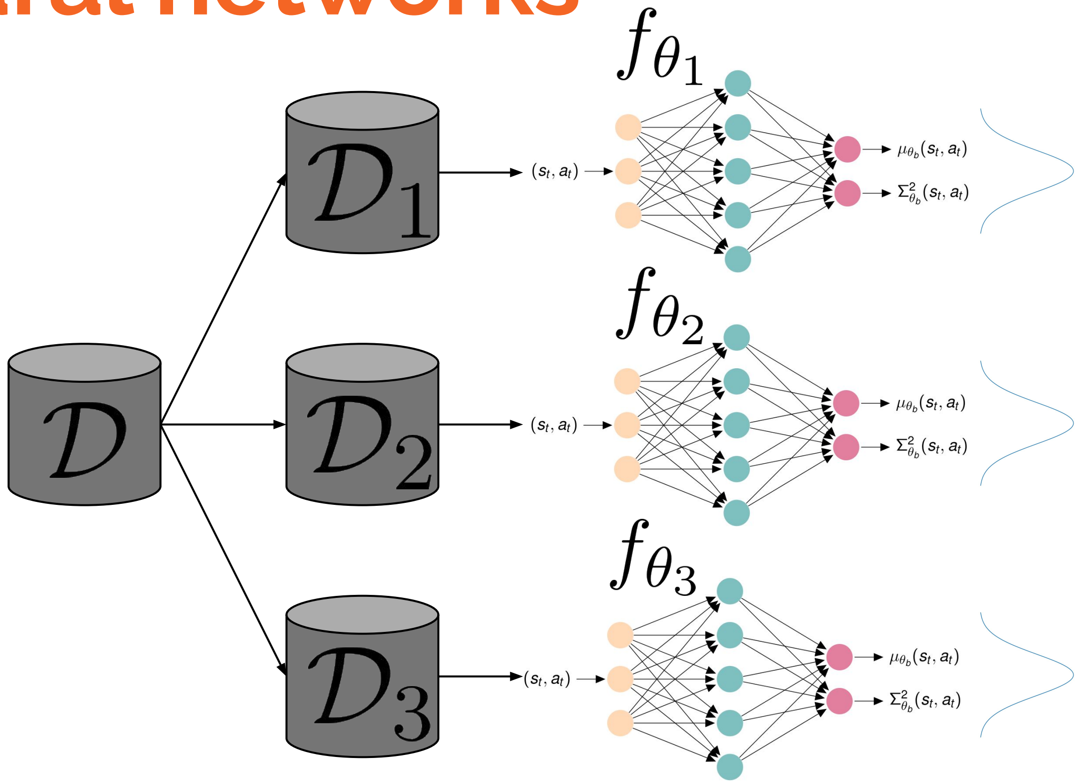
$$p(s_{t+1} \mid s_t, a_t, \theta_b) = \mathcal{N}\left(s_{t+1} \mid \mu_{\theta_b}(s_t, a_t), \Sigma_{\theta_b}(s_t, a_t)\right)$$

- Predictions are uniformly-weighted mixture

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



Ensemble of probabilistic neural networks



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

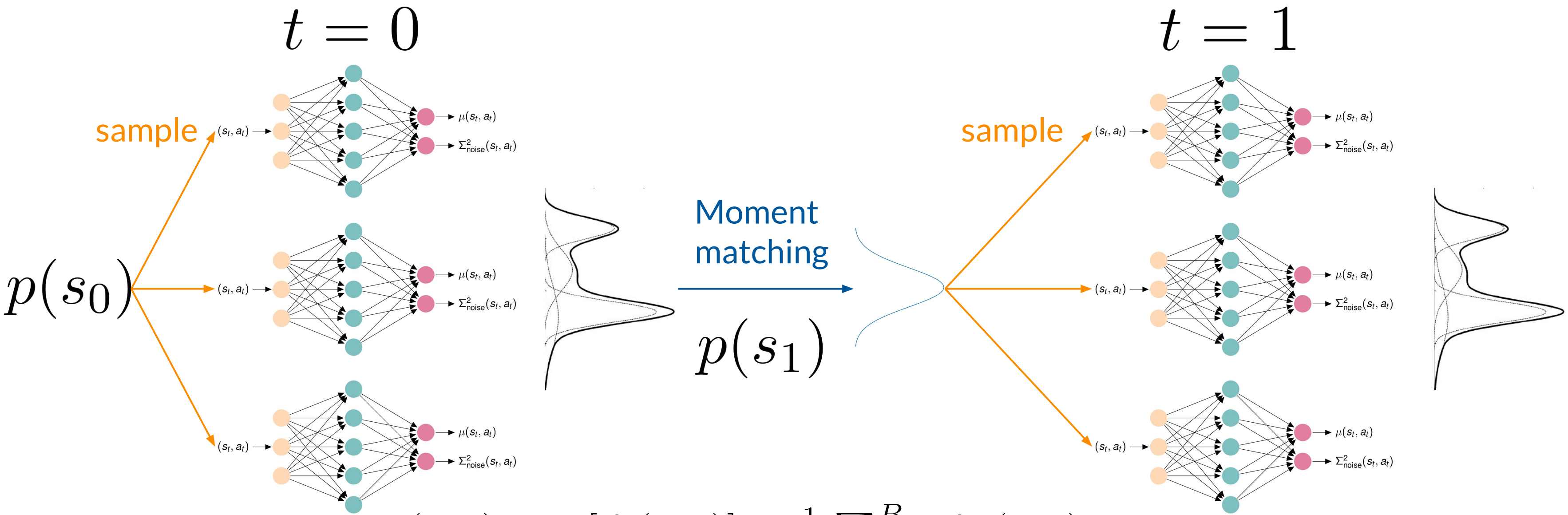
$$p(\theta \mid \mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{\int p(\mathcal{D}|\theta)p(\theta)\mathrm{d}\theta}$$

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

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

How to propagate uncertainty?

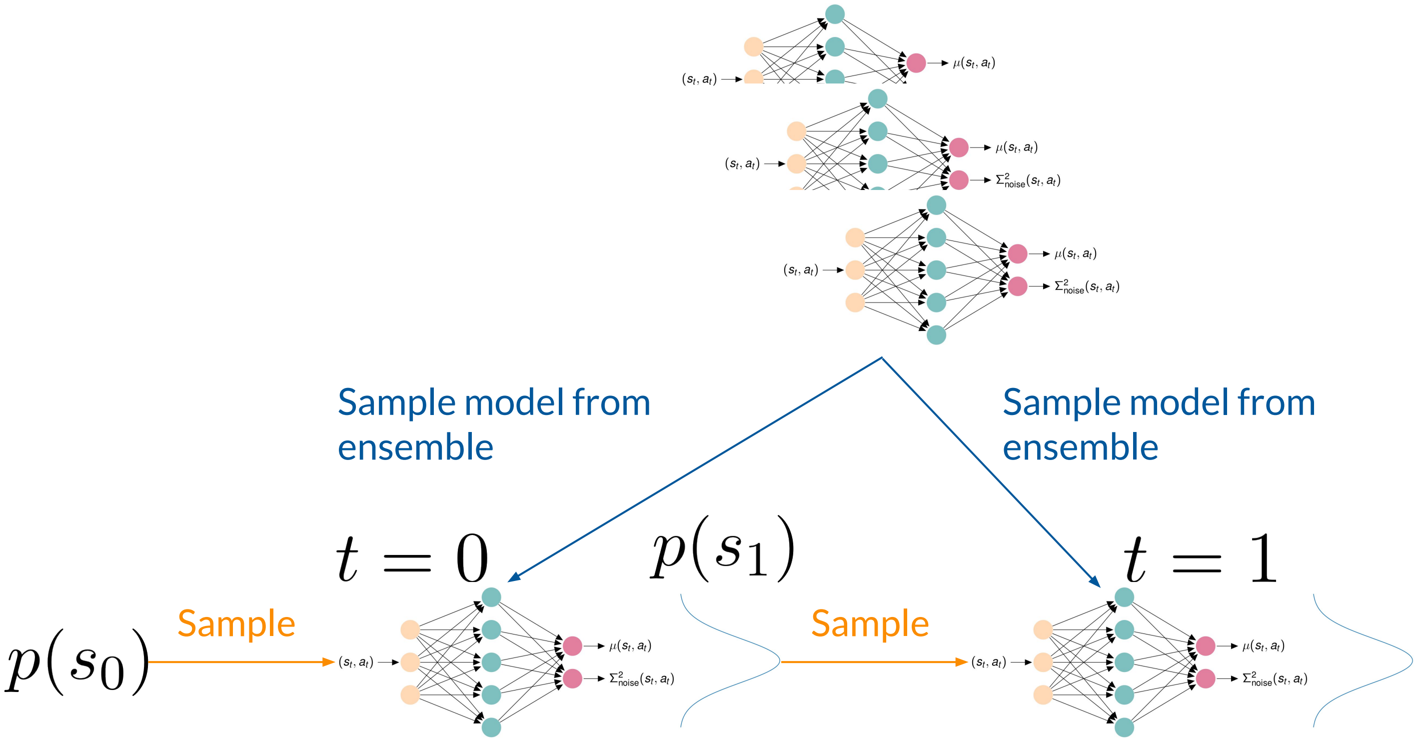
Uncertainty propagation via moment matching



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

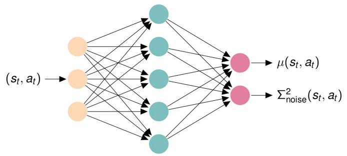
$$\Sigma_f^2(s,a) = \mathbb{V}\left[f_\theta(s,a)\right] = \frac{1}{B}\sum_{b=1}^{B}\left(\Sigma^2_{\theta_b}(s,a) + \mu^2_{\theta_b}(s,a)\right) - \mu^2_\theta(s,a)$$

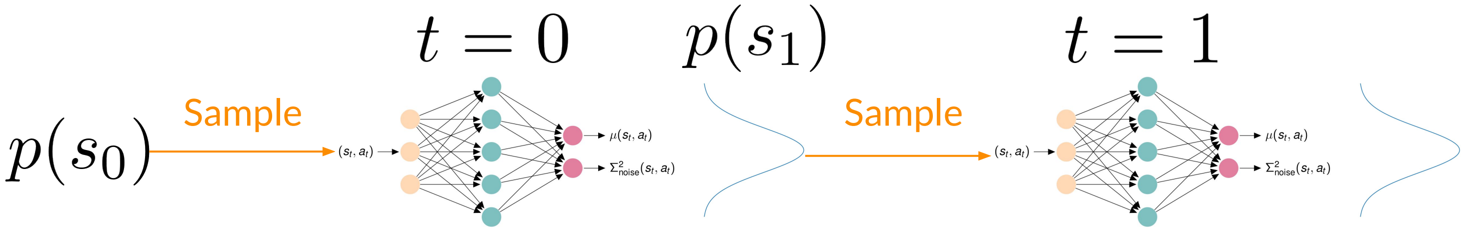
Uncertainty propagation via trajectory sampling TS-1



Uncertainty propagation via trajectory sampling TS-∞

Sample one dynamics model from ensemble





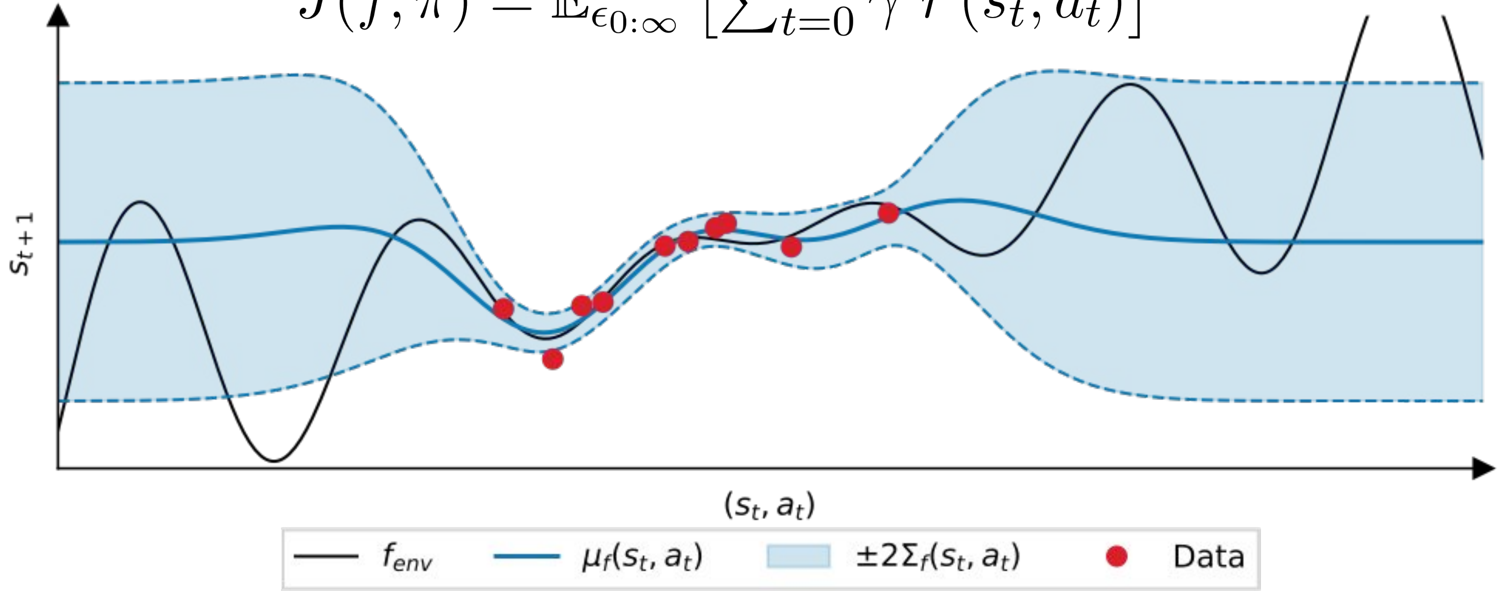
- TS-∞ captures time invariance of dynamics

Uncertainty-guided exploration

Uncertainty-guided exploration

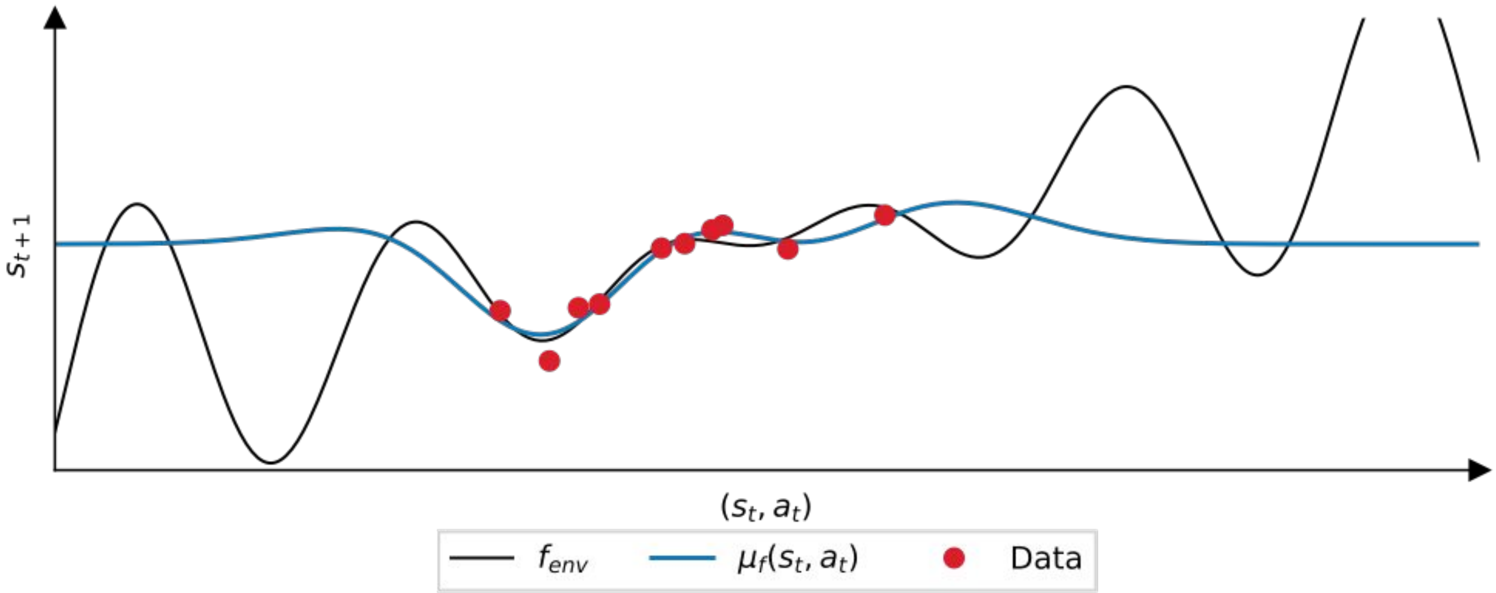
$$p(f \mid \mathcal{D} \cup (s_t, a_t)) = \mathcal{N}(f(s_t, a_t) \mid \mu_f(s, a), \Sigma_f(s_t, a_t))$$

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



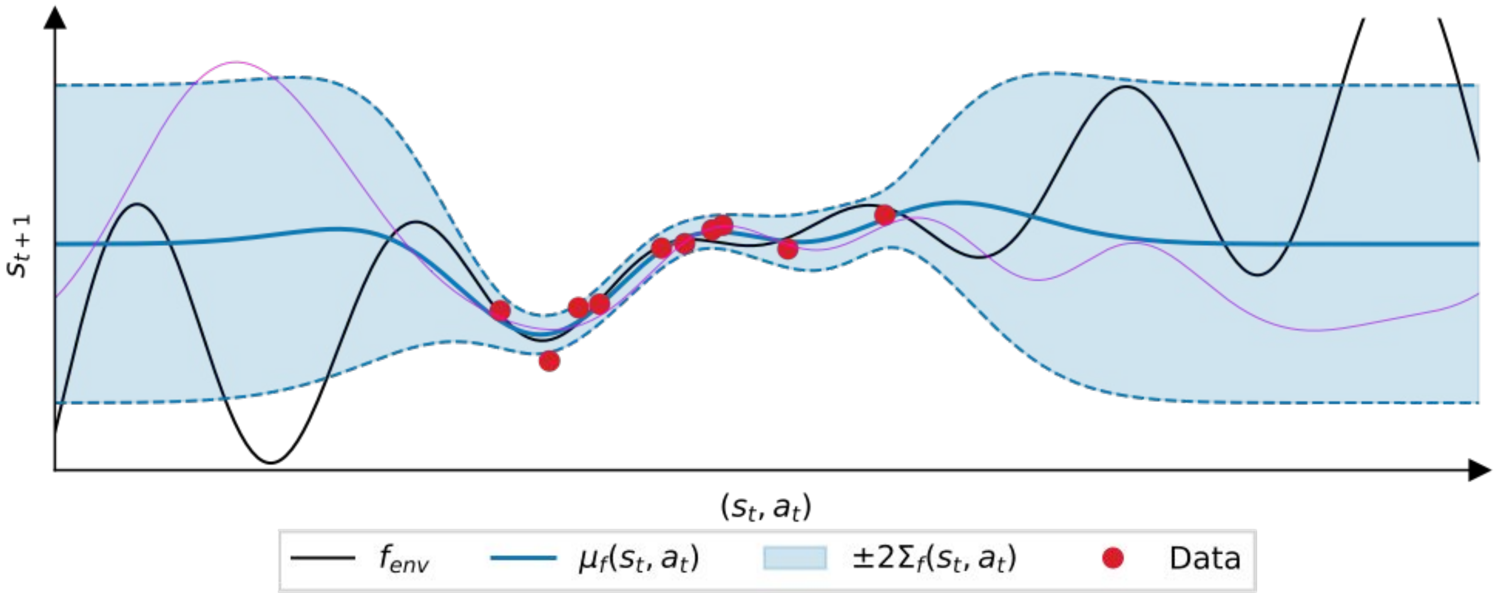
Exploration via greedy exploitation

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



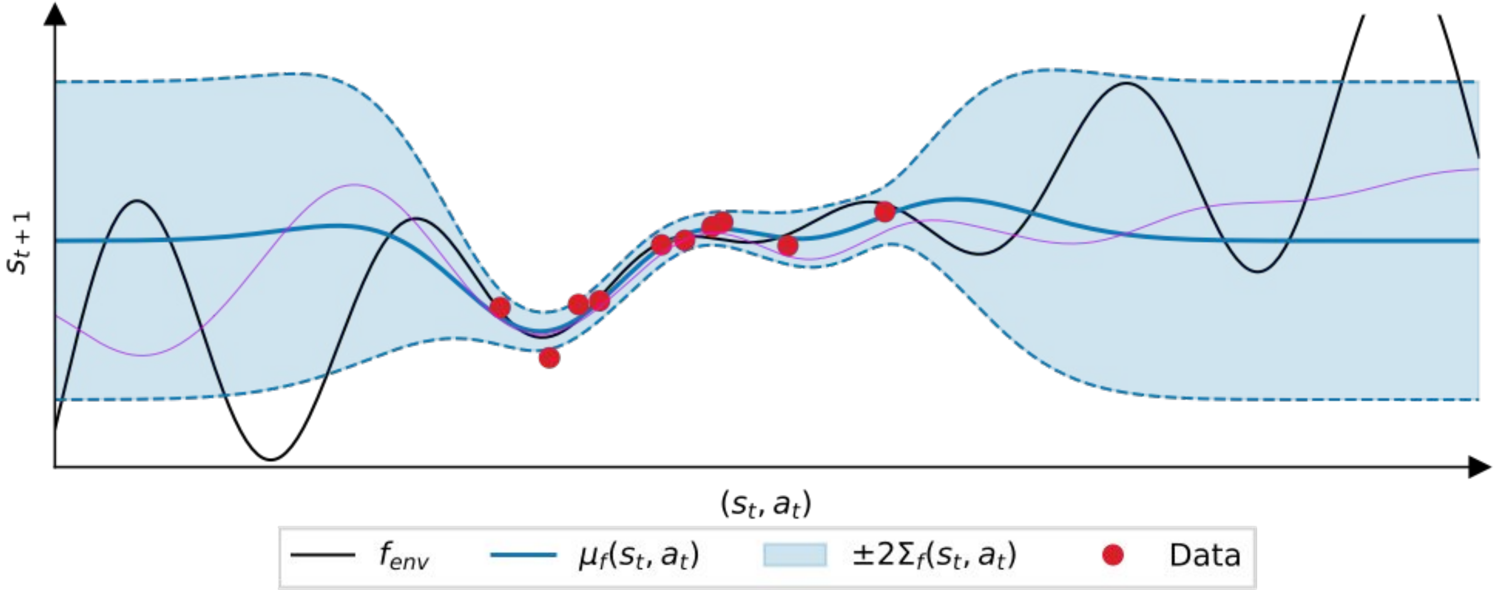
Exploration via Thompson sampling

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



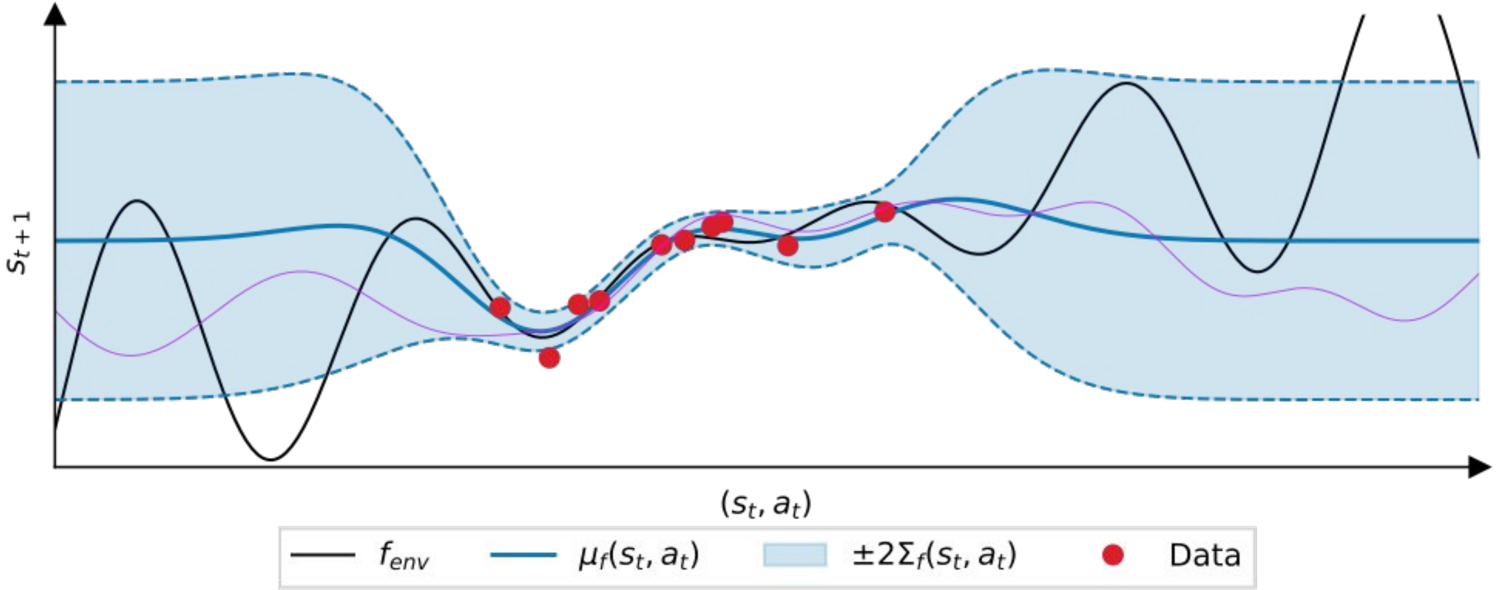
Exploration via Thompson sampling

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



Exploration via Thompson sampling

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

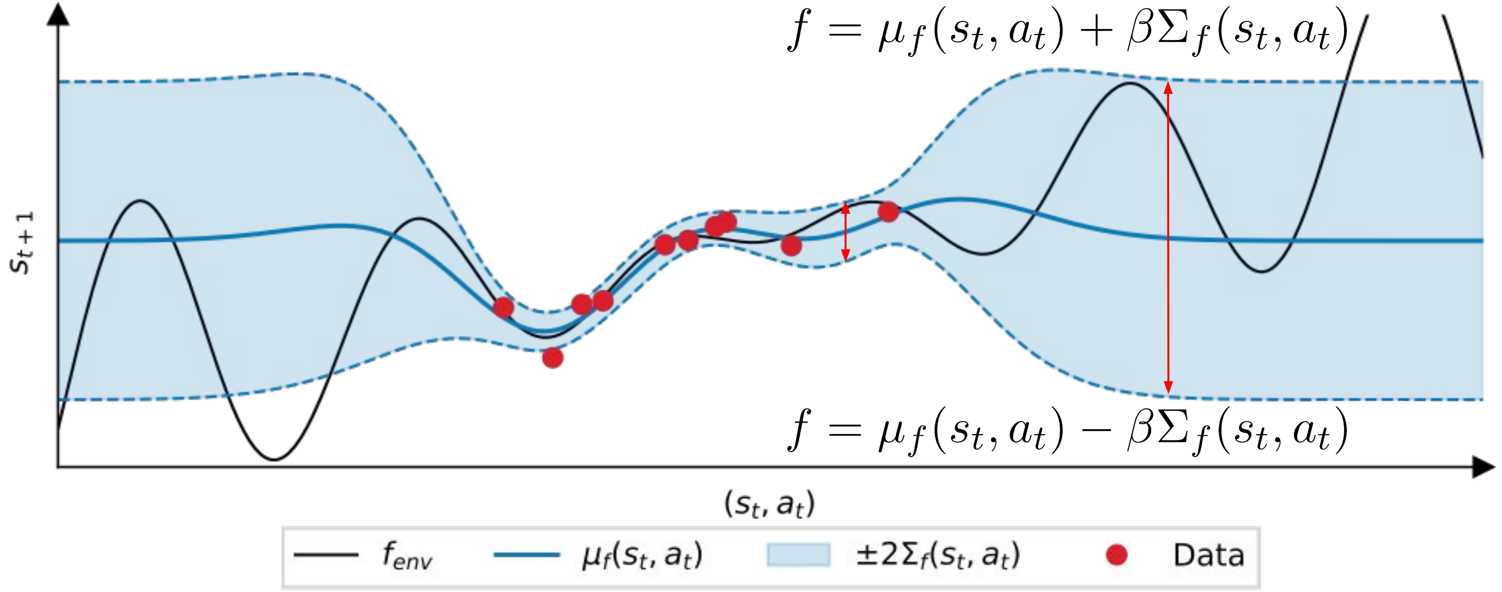


Exploration via
Upper Confidence Bound (UCB)

Optimism in the face of uncertainty

Exploration via
Upper Confidence Bound (UCB)

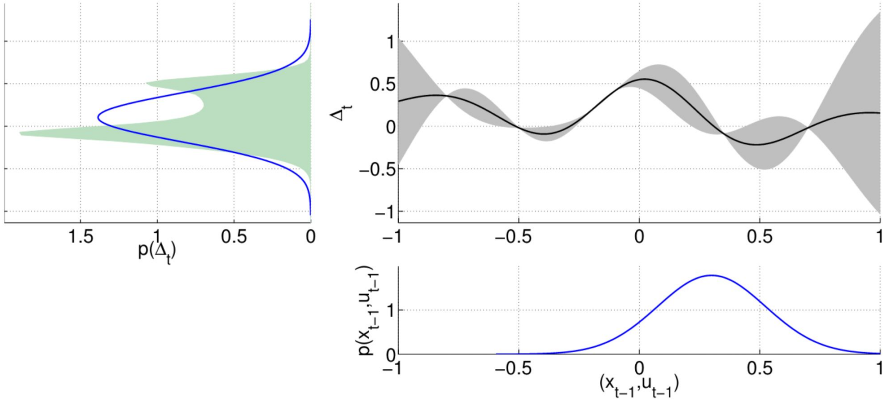
$$\pi_{\text{UCB}} = \underset{\pi}{\text{argmax}} \max_{f \in \mathcal{M}} [J(f, \pi)] \qquad \mathcal{M} = \{f \mid |f(s,a) - \mu_f(s,a)| \leq \beta \Sigma_f(s,a)\}$$

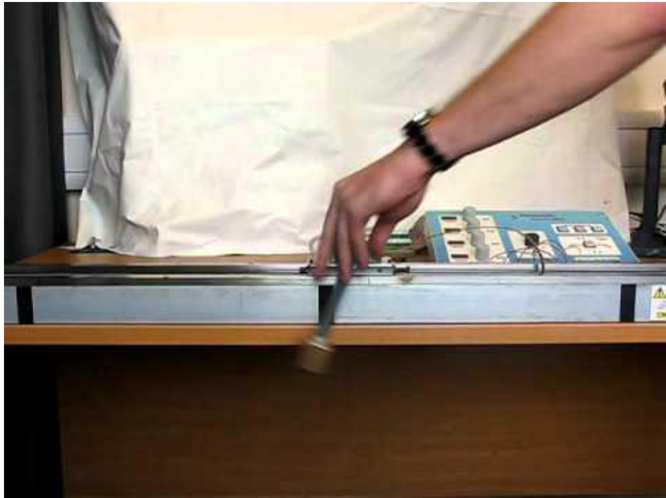


Examples

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



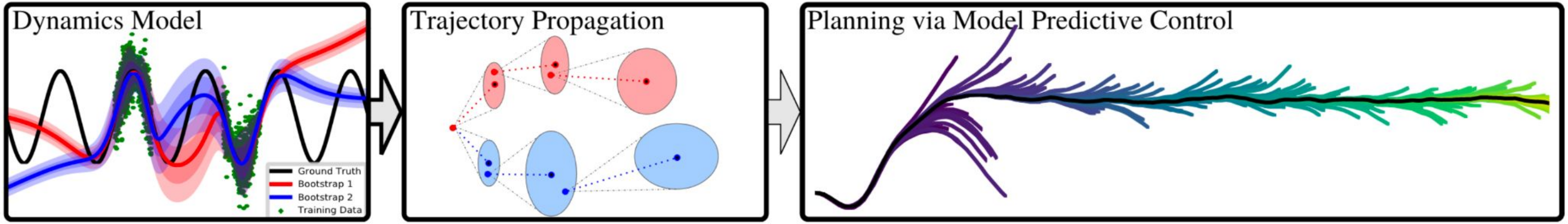


PETS: Probabilistic Ensembles with Trajectory Sampling

- **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)

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

    - 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

Why not model-based RL

- **Poor asymptotic performance (sometimes)**
  - not matching asymptotic performance of model-free methods
  - lacks improvement guarantees that underpin model-free methods

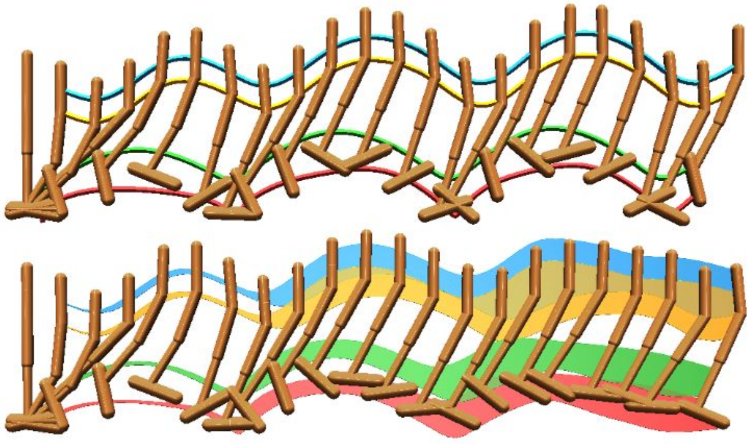
Model-bias in model-based RL

- **Overfitting in supervised learning**
  - model performs well on training data but poorly on test data
    - 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

Issues in MBRL

1. **Model bias**
    - policy learning exploits model inaccuracies
2. **Compound error**
    - errors compound when making multi-step predictions
3. **Objective mismatch**
    - model training is a simple optimizati

a)
b)


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

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?

aidan.scannell@aalto.fi

www.aidanscannell.com

MOPO: Model-based offline policy optimization

- **Offline RL**
  - no need to explore
- **Keep policy in regions of dynamics with low aleatoric uncertainty**
  - Penalize reward with uncertainty of the dynamics

$$\widehat{T}_{\theta,\phi}(s_{t+1}, r | s_t, a_t) = \mathcal{N}(\mu_\theta(s_t, a_t), \Sigma_\phi(s_t, a_t))$$

$$\{\widehat{T}^i_{\theta,\phi} = \mathcal{N}(\mu^i_\theta, \Sigma^i_\phi)\}_{i=1}^N$$

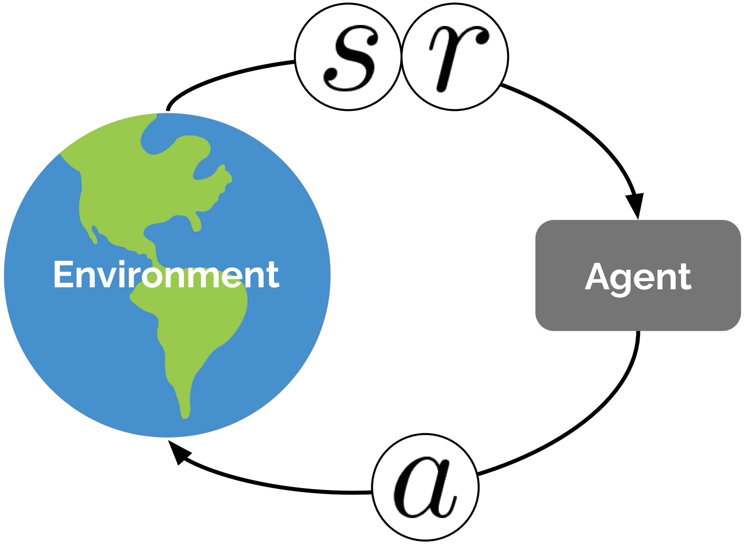
$$\tilde{r}(s,a) = \hat{r}(s,a) - \lambda \max_{i=1,\dots,N} \|\Sigma^i_\phi(s,a)\|_\mathrm{F}$$

MOPO: Model-based Offline Policy Optimization

Tianhe Yu*[1], Garrett Thomas*[1], Lantao Yu[1], Stefano Ermon[1], James Zou[1], Sergey Levine[2], Chelsea Finn†[1], Tengyu Ma†[1]
Stanford University[1], UC Berkeley[2]
{tianheyu,gwthomas}@cs.stanford.edu

RL algorithm

1. Collect data using policy π
2. Improve policy using data



Lecture outcomes

1. Understand some of the difficulties in MBRL
2. Know the different sources of uncertainty in MBRL
3. Understand the role of uncertainty in MBRL

Potential fixes

1. **Keep policy in region of dynamics with sufficient training data** e.g. ME-TRPO
    a. but how to explore?
        i. insufficient exploration could trap model and policy optimization
        ii. excessive exploration confuses model and causes policy chattering
2. **Limit model use** e.g. MBPO
    a. model error compounds when making multi-step predictions
        i. so restrict how many steps model is used for...
3. **And more, open research question...**

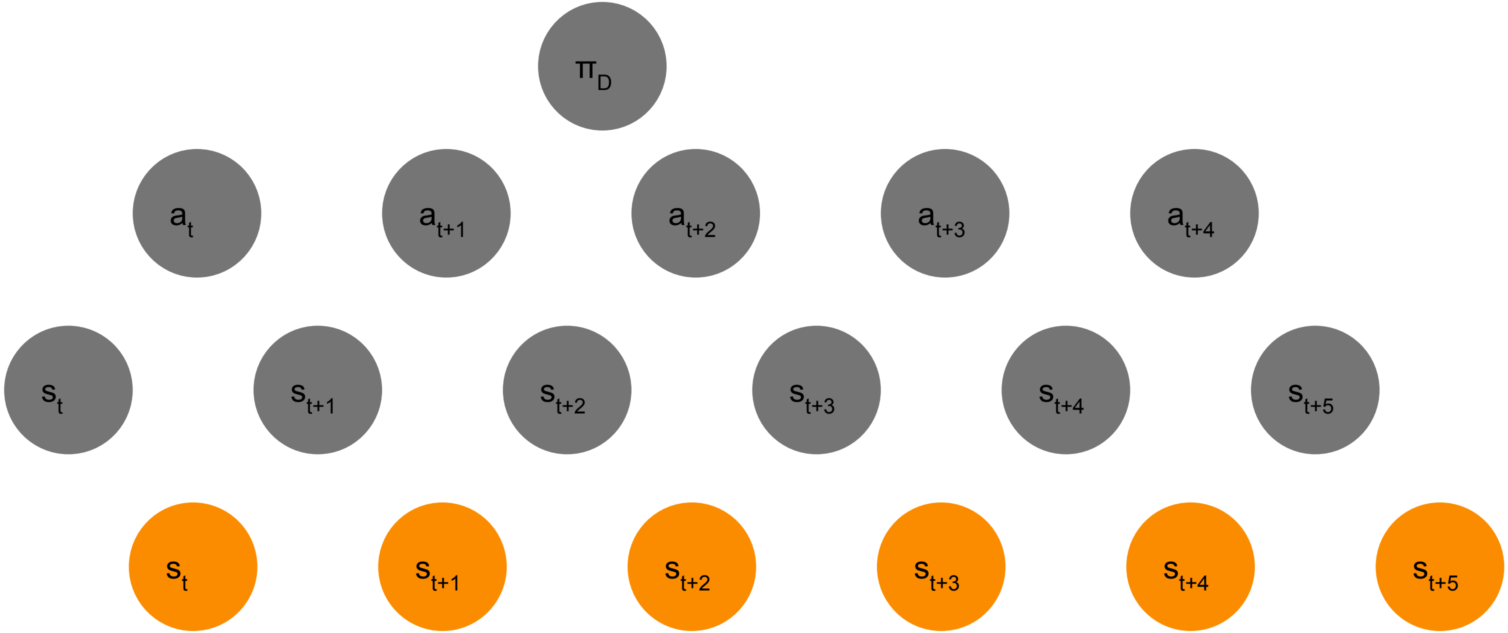
Summary

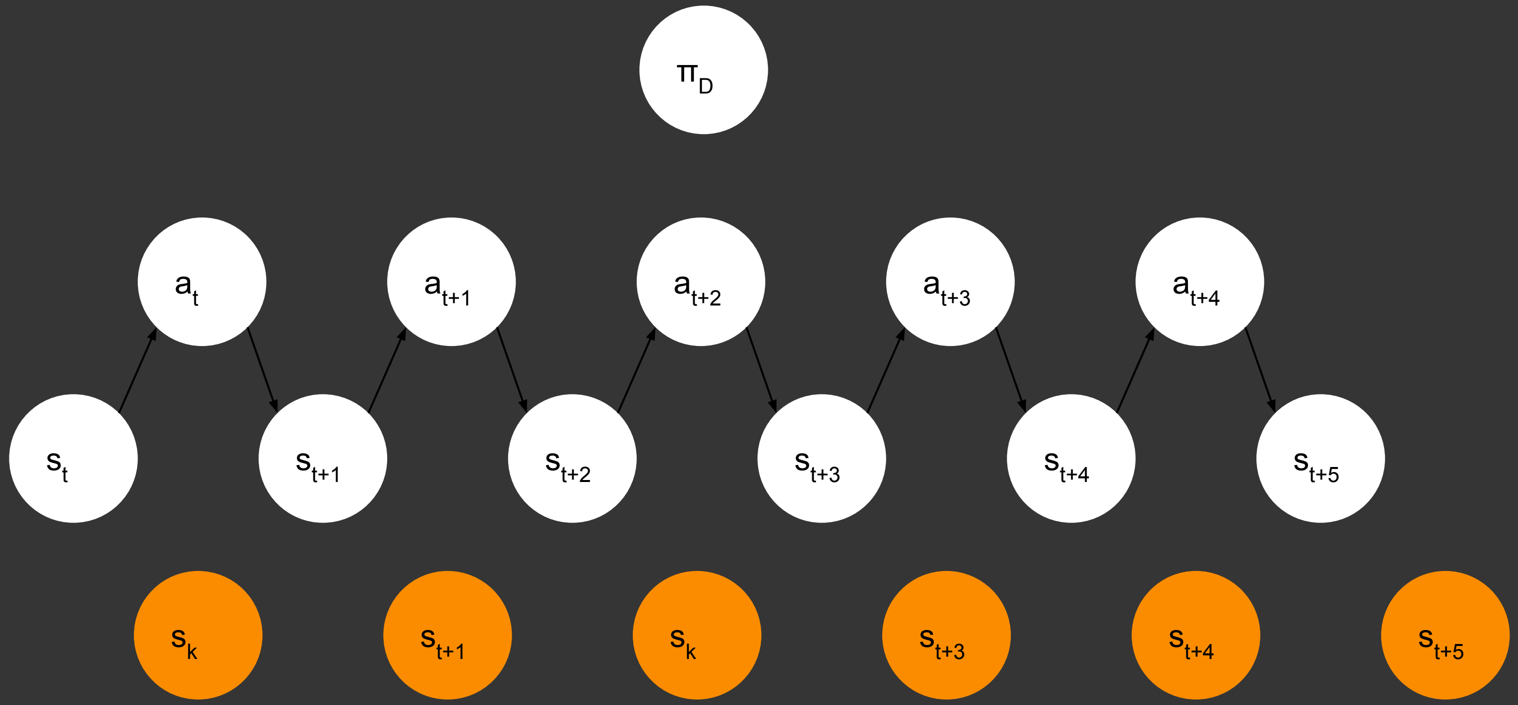
- Important to disentangle epistemic/aleatoric uncertainties
- Epistemic uncertainty can be used for exploration

Points to think about

- Epistemic uncertainty can be used
  - to keep the policy in regions of dynamics with sufficient data
    -
  - For exploration
    - but too much exploration can lead to training instability...
- Methods are only as good as their uncertainty estimates
  - How to get better uncertainty estimates?
  - Ensembles vs MC dropout vs variational inference vs Laplace
- Predict entire trajectories instead of recursively using single-step model

MBPO: Model Based Policy Optimisation





π_D
a_t
a_{t+1}
a_{t+2}
a_{t+3}
a_{t+4}
s_t
s_{t+1}
s_{t+2}
s_{t+3}
s_{t+4}
s_{t+5}
s_k
s_{t+1}
s_k
s_{t+3}
s_{t+4}
s_{t+5}

Issues in model-based RL

- **Model bias** - analogous to off-policy error
  - models enable us to generate samples from **current policy** at **any state**
    - circumvents **off-policy error**
    - but introduces **model bias** to reduce **off-policy error**

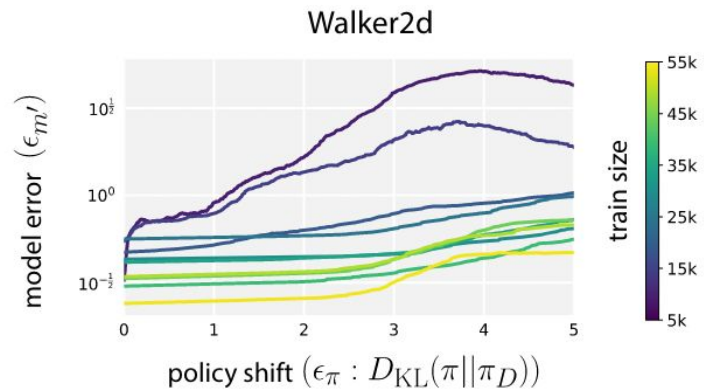
- **Exploration**
  - try not to visit same state multiple times in episode
  - try not to revisit states seen in previous episodes
    - unless needed for further exploration

Issues in model-based RL

- **Training instability**
  - Drop in performance during training
  - Model bias
  - Constrain policy update to stabilise learning (PPO/TRPO)
- **Asymptotic performance not matching model-free**
- **Model bias** - analogous to off-policy error
  - models enable us to generate samples from **current policy** at **any state**
    - circumvents **off-policy error**
    - but introduces **model bias** to reduce **off-policy error**

Issues in model-based RL

- Training instability
  - Model bias
  - Constrain policy update to stabilise learning (PPO/TRPO)

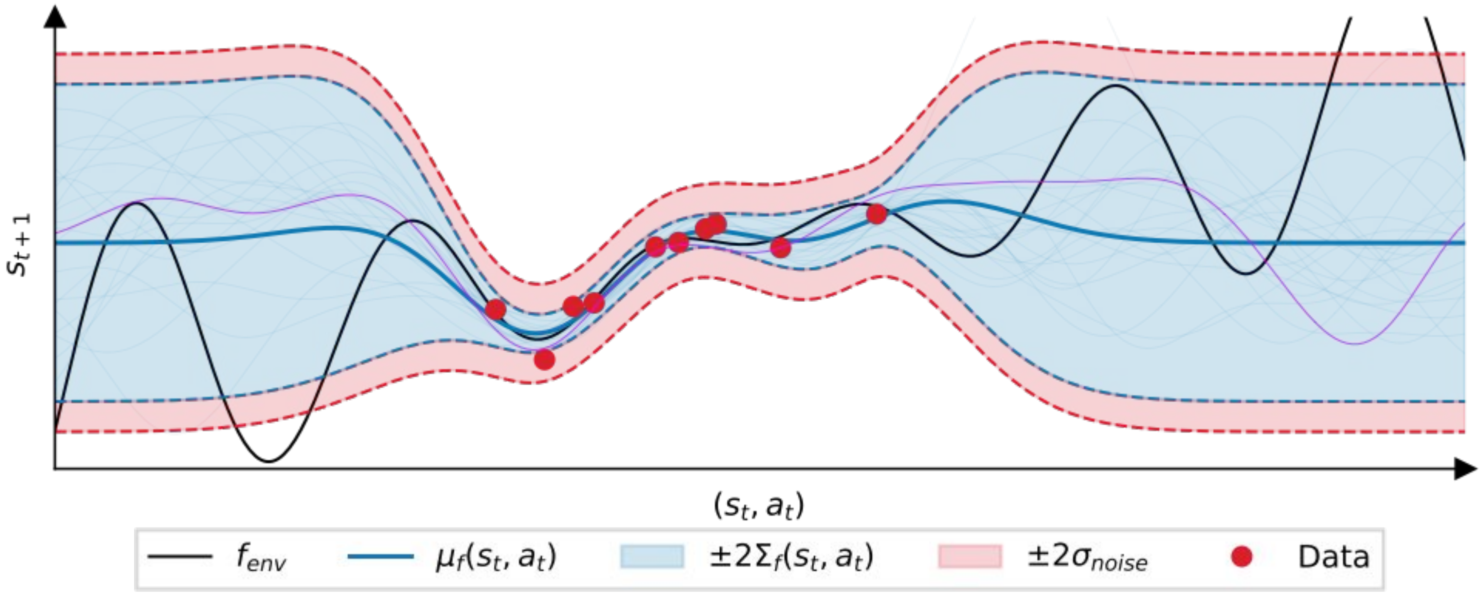


Janner, Michael, et al. "When to trust your model: Model-based policy optimization." *Advances in Neural Information Processing Systems* 32 (2019).

Exploration in RL

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

Exploration via Thompson sampling



PILCO: Probabilistic Inference for Learning cOntrol

- **Pros**
  - Sample efficient
  - Robust to model errors
- Cons
  - Poor scaling (GPs scale cubically with number of training points)
  - Can only be used with restricted class of policies/reward functions
  - Assumes smooth dynamics

Exploration in model-based RL

- Greedy exploitation

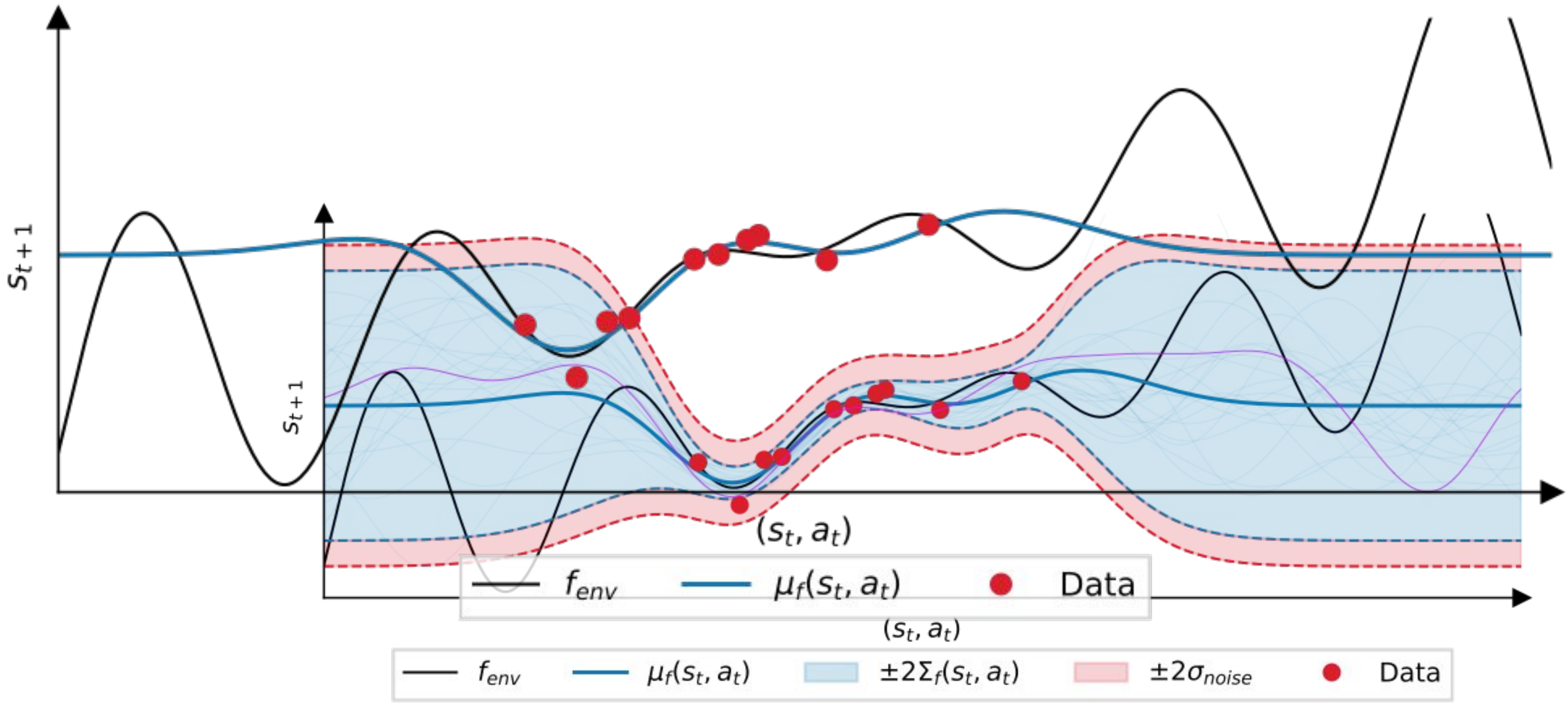
$$\pi_{\text{greedy}} = \operatorname*{argmax}_{\pi} \mathbb{E}_{f \sim p(f|\mathcal{D})} \left[J(f, \pi) \right]$$

- Thompson sampling

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

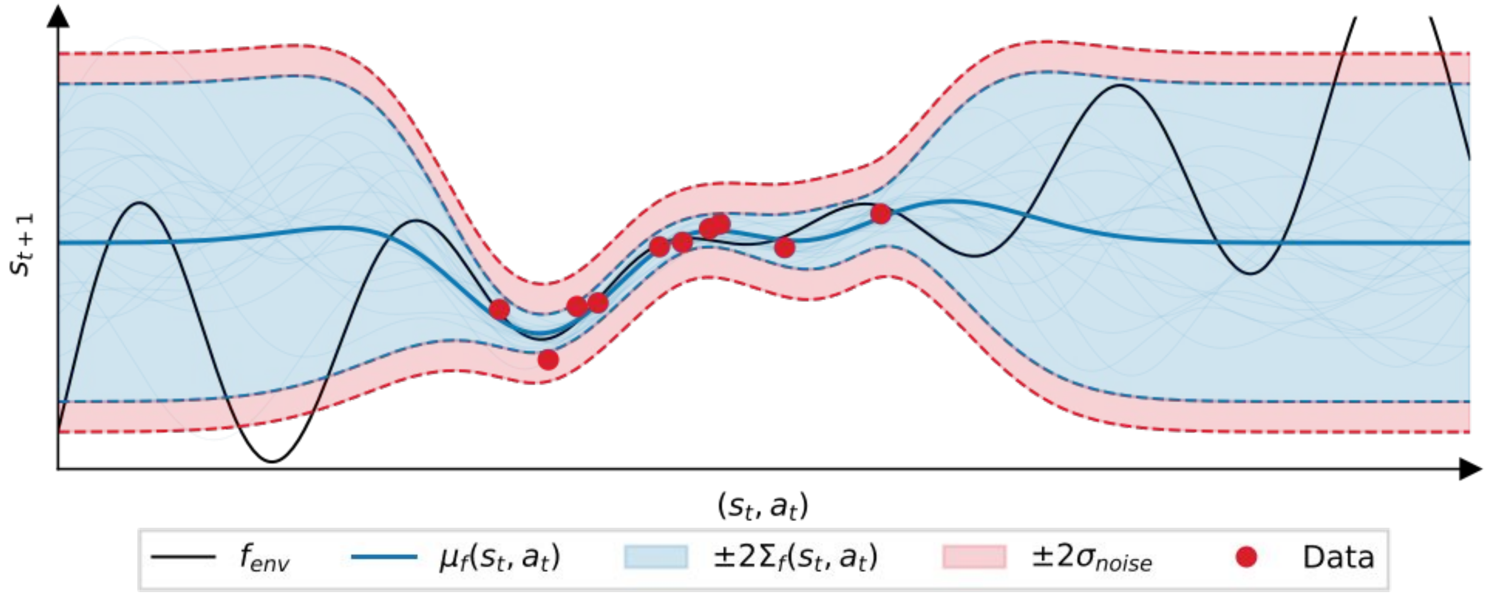
- Upper confidence bound

$$\pi_{\text{UCB}} = \operatorname*{argmax}_{\pi} \max_{f \in \mathcal{M}} \left[J(f, \pi) \right]$$

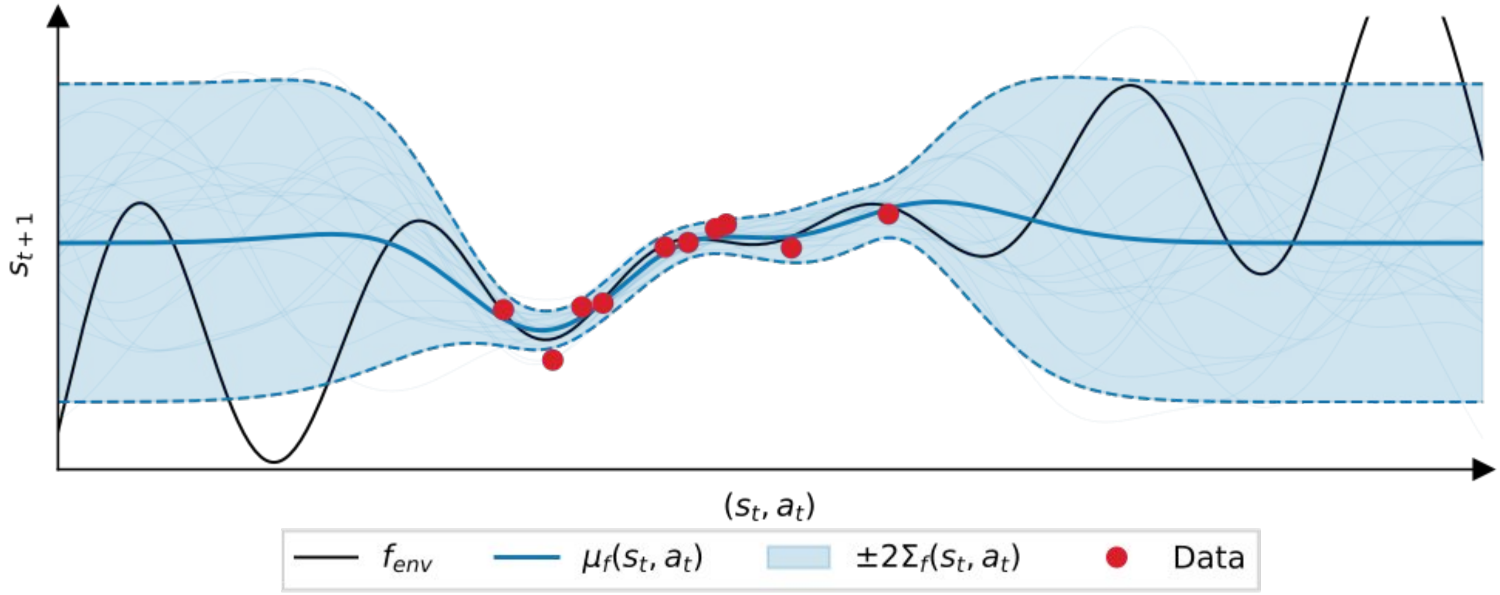


s_{t+1}
s_{t+1}
(s_t, a_t)
(s_t, a_t)
f_{env}
$\mu_f(s_t, a_t)$
Data
f_{env}
$\mu_f(s_t, a_t)$
$\pm 2\Sigma_f(s_t, a_t)$
$\pm 2\sigma_{noise}$
Data

Exploration in model-based RL



Exploration in model-based RL



PETS: Probabilistic Ensembles with Trajectory Sampling

- **Pros**
  - Sample efficient
  - Robust to model errors
- Cons
  - Poor scaling (GPs scale cubically with number of training points)
  - Can only be used with restricted class of policies/reward functions
  - Assumes smooth dynamics