Data-Efficient Reinforcement Learning for Autonomous Robots

Marc Deisenroth
Centre for Artificial Intelligence
Department of Computer Science
University College London

m.deisenroth@ucl.ac.uk

IBIS 2019, Nagoya
November 20, 2019
Vision: Autonomous robots support humans in everyday activities ➤ Fast learning and automatic adaptation
Vision: Autonomous robots support humans in everyday activities ➤ Fast learning and automatic adaptation

Currently: Data-hungry learning or human guidance
Autonomous Robots

**Vision:** Autonomous robots support humans in everyday activities

- Fast learning and automatic adaptation

**Currently:** Data-hungry learning or human guidance

Fully autonomous learning and decision making with little data in real-life situations
Central Problem

Data-Efficient Reinforcement Learning

Ability to learn and make decisions in complex domains without requiring large quantities of data
1. **Model-based RL**
   - Data-efficient decision making

2. **Model predictive RL**
   - Speed up learning further by online planning

3. **Meta learning**
   - Generalization of knowledge to new situations
Data-efficient Reinforcement Learning

Model-based RL
Model Predictive Control
Meta Learning
Reinforcement Learning

\[ x_{t+1} = f(x_t, u_t) + w, \quad u_t = \pi(x_t, \theta) \]

State \quad Control \quad Policy \quad Policy parameters

Transition function

Objective (Controller Learning)

Find policy parameters \( \theta \) that minimize the expected long-term cost

\[ J_p(\theta) = \sum_{t=0}^{T} E_{\pi,x_0} \left[ c_p x_t | x_0, \pi, \theta \right] \]

Instantaneous cost \( c_p x_t \), e.g., \( x_t \to x_{\text{target}} \)

Typical objective in optimal control and reinforcement learning (Bertsekas, 2005; Sutton & Barto, 1998)
**Objective (Controller Learning)**

Find policy parameters $\theta^*$ that minimize the expected long-term cost

$$J(\theta) = \sum_{t=1}^{T} \mathbb{E}[c(x_t)|\theta], \quad p(x_0) = \mathcal{N}(\mu_0, \Sigma_0).$$

Instantaneous cost $c(x_t)$, e.g., $\|x_t - x_{\text{target}}\|^2$

- Typical objective in **optimal control** and **reinforcement learning**
  (Bertsekas, 2005; Sutton & Barto, 1998)
Fast Reinforcement Learning

Objective

Minimize expected long-term cost \( J(\theta) = \sum_t \mathbb{E}[c(x_t)|\theta] \)

PILCO Framework: High-Level Steps

1. Probabilistic model for transition function \( f \)
   - System identification

Deisenroth et al. (IEEE-TPAMI, 2015): Gaussian Processes for Data-Efficient Learning in Robotics and Control
Objective

Minimize expected long-term cost $J(\theta) = \sum_t \mathbb{E}[c(x_t)|\theta]$.

PILCO Framework: High-Level Steps

1. Probabilistic model for transition function $f$
   - System identification
2. Compute long-term predictions $p(x_1|\theta), \ldots, p(x_T|\theta)$

Deisenroth et al. (IEEE-TPAMI, 2015): Gaussian Processes for Data-Efficient Learning in Robotics and Control
Objective

Minimize expected long-term cost $J(\theta) = \sum_t \mathbb{E}[c(x_t)|\theta]$ 

PILCO Framework: High-Level Steps

1. Probabilistic model for transition function $f$
   - System identification
2. Compute long-term predictions $p(x_1|\theta), \ldots, p(x_T|\theta)$
3. Policy improvement

Deisenroth et al. (IEEE-TPAMI, 2015): Gaussian Processes for Data-Efficient Learning in Robotics and Control
Objective

Minimize expected long-term cost $J(\theta) = \sum_t \mathbb{E}[c(x_t)|\theta]$

PILCO Framework: High-Level Steps

1. Probabilistic model for transition function $f$
   - System identification
2. Compute long-term predictions $p(x_1|\theta), \ldots, p(x_T|\theta)$
3. Policy improvement
4. Apply controller

Deisenroth et al. (IEEE-TPAMI, 2015): *Gaussian Processes for Data-Efficient Learning in Robotics and Control*
Objective

Minimize expected long-term cost $J(\theta) = \sum_t \mathbb{E}[c(x_t)|\theta]$.

PILCO Framework: High-Level Steps

1. Probabilistic model for transition function $f$
   - System identification

2. Compute long-term predictions $p(x_1|\theta), \ldots, p(x_T|\theta)$

3. Policy improvement

4. Apply controller

Deisenroth et al. (IEEE-TPAMI, 2015): Gaussian Processes for Data-Efficient Learning in Robotics and Control
Model learning problem: Find a function $f : x \mapsto f(x) = y$
Model learning problem: Find a function $f : x \mapsto f(x) = y$
Model learning problem: Find a function $f : x \mapsto f(x) = y$

Plausible model

Predictions? Decision Making?
Model learning problem: Find a function $f : x \mapsto f(x) = y$
Model learning problem: Find a function $f : x \mapsto f(x) = y$

Distribution over plausible functions
Model learning problem: Find a function $f : x \mapsto f(x) = y$

Distribution over plausible functions

- Express **uncertainty** about the underlying function to be robust to model errors
- **Gaussian process** for model learning
  (Rasmussen & Williams, 2006)
Objective

Minimize expected long-term cost \( J(\theta) = \sum_t \mathbb{E}[c(x_t) | \theta] \)

PILCO Framework: High-Level Steps

1. Probabilistic model for transition function \( f \)
   - System identification
2. **Compute long-term predictions** \( p(x_1 | \theta), \ldots, p(x_T | \theta) \)
3. Policy improvement
4. Apply controller

Deisenroth et al. (IEEE-TPAMI, 2015): *Gaussian Processes for Data-Efficient Learning in Robotics and Control*
Iteratively compute $p(x_1|\theta), \ldots, p(x_T|\theta)$
Long-Term Predictions

- Iteratively compute $p(x_1|\theta), \ldots, p(x_T|\theta)$

\[
p(x_{t+1}|x_t, u_t) \quad p(x_t, u_t|\theta)
\]

GP prediction $\mathcal{N}(\mu, \Sigma)$

Deisenroth et al. (IEEE-TPAMI, 2015): Gaussian Processes for Data-Efficient Learning in Robotics and Control
Long-Term Predictions

Iteratively compute $p(x_1 | \theta), \ldots, p(x_T | \theta)$

$$p(x_{t+1} | \theta) = \int \int \int p(x_{t+1} | x_t, u_t) p(x_t, u_t | \theta) \, df \, dx_t \, du_t$$

GP prediction

$\mathcal{N}(\mu, \Sigma)$

Deisenroth et al. (IEEE-TPAMI, 2015): Gaussian Processes for Data-Efficient Learning in Robotics and Control
Iteratively compute $p(x_1|\theta), \ldots, p(x_T|\theta)$

$$p(x_{t+1}|\theta) = \int \int \left[ p(x_{t+1}|x_t, u_t) \cdot p(x_t, u_t|\theta) \right] df \, dx_t \, du_t$$

GP moment matching

(Girard et al., 2002; Quiñonero-Candela et al., 2003)

Deisenroth et al. (IEEE-TPAMI, 2015): Gaussian Processes for Data-Efficient Learning in Robotics and Control
Fast Reinforcement Learning

Objective

Minimize expected long-term cost $J(\theta) = \sum_t \mathbb{E}[c(x_t)|\theta]$

PILCO Framework: High-Level Steps

1. Probabilistic model for transition function $f$
   - System identification
2. Compute long-term predictions $p(x_1|\theta), \ldots, p(x_T|\theta)$
3. Policy improvement
   - Compute expected long-term cost $J(\theta)$
   - Find parameters $\theta$ that minimize $J(\theta)$
4. Apply controller

Deisenroth et al. (IEEE-TPAMI, 2015): Gaussian Processes for Data-Efficient Learning in Robotics and Control
Policy Improvement

Objective

Minimize expected long-term cost

\[ J(\theta) = \sum_t E[c(x_t)|\theta] \]

- (Gaussian) state distributions

\[ p(x_1|\theta), \ldots, p(x_T|\theta) \]

Deisenroth et al. (IEEE-TPAMI, 2015): *Gaussian Processes for Data-Efficient Learning in Robotics and Control*
Policy Improvement

Objective

Minimize expected long-term cost $J(\theta) = \sum_t \mathbb{E}[c(x_t) | \theta]$

- (Gaussian) state distributions $p(x_1 | \theta), \ldots, p(x_T | \theta)$
- Compute

$$\mathbb{E}[c(x_t) | \theta] = \int c(x_t) \mathcal{N}(x_t | \mu_t, \Sigma_t) \, dx_t, \quad t = 1, \ldots, T,$$

and sum them up to obtain $J(\theta)$

Deisenroth et al. (IEEE-TPAMI, 2015): Gaussian Processes for Data-Efficient Learning in Robotics and Control
Objective

Minimize expected long-term cost \( J(\theta) = \sum_t \mathbb{E}[c(x_t) | \theta] \)

- (Gaussian) state distributions \( p(x_1 | \theta), \ldots, p(x_T | \theta) \)
- Compute

\[
\mathbb{E}[c(x_t) | \theta] = \int c(x_t) \mathcal{N}(x_t | \mu_t, \Sigma_t) dx_t, \quad t = 1, \ldots, T,
\]

and sum them up to obtain \( J(\theta) \)

- Analytically compute gradient \( dJ(\theta)/d\theta \)
- Standard gradient-based optimizer (e.g., BFGS) to find \( \theta^* \)

Deisenroth et al. (IEEE-TPAMI, 2015): Gaussian Processes for Data-Efficient Learning in Robotics and Control
Fast Reinforcement Learning

Objective

Minimize expected long-term cost $J(\theta) = \sum_t \mathbb{E}[c(x_t)|\theta]$

PILCO Framework: High-Level Steps

1. Probabilistic model for transition function $f$
   - System identification
2. Compute long-term predictions $p(x_1|\theta), \ldots, p(x_T|\theta)$
3. Policy improvement
4. Apply controller
Standard Benchmark: Cart-Pole Swing-up

- Swing up and balance a freely swinging pendulum on a cart
- No knowledge about nonlinear dynamics ➤ Learn from scratch
- Cost function $c(\mathbf{x}) = 1 - \exp(-\|\mathbf{x} - \mathbf{x}_{\text{target}}\|^2)$

- Code: [https://github.com/ICL-SML/pilco-matlab](https://github.com/ICL-SML/pilco-matlab)

Deisenroth & Rasmussen (ICML, 2011): *PILCO: A Model-based and Data-efficient Approach to Policy Search*
Standard Benchmark: Cart-Pole Swing-up

- Swing up and balance a freely swinging pendulum on a cart
- No knowledge about nonlinear dynamics ➤ Learn from scratch
- Cost function $c(x) = 1 - \exp(-\|x - x_{\text{target}}\|^2)$

**Code:** [https://github.com/ICL-SML/pilco-matlab](https://github.com/ICL-SML/pilco-matlab)

Deisenroth & Rasmussen (ICML, 2011): PILCO: A Model-based and Data-efficient Approach to Policy Search
Swing up and balance a freely swinging pendulum on a cart

No knowledge about nonlinear dynamics – Learn from scratch

Cost function $c(x) = 1 - \exp(-\|x - x_{\text{target}}\|^2)$

Unprecedented learning speed compared to state-of-the-art

Code: [https://github.com/ICL-SML/pilco-matlab](https://github.com/ICL-SML/pilco-matlab)

Deisenroth & Rasmussen (ICML, 2011): PILCO: A Model-based and Data-efficient Approach to Policy Search
Wide Applicability

Application to a wide range of robotic systems

Deisenroth et al. (RSS, 2011): *Learning to Control a Low-Cost Manipulator using Data-efficient Reinforcement Learning*
Englert et al. (ICRA, 2013): *Model-based Imitation Learning by Probabilistic Trajectory Matching*
Deisenroth et al. (ICRA, 2014): *Multi-Task Policy Search for Robotics*
Kupcsik et al. (AIJ, 2017): *Model-based Contextual Policy Search for Data-Efficient Generalization of Robot Skills*
In robotics, **data-efficient** learning is critical

- Probabilistic, model-based RL approach
  - Reduce model bias
  - Unprecedented learning speed
  - Wide applicability
Data-efficient Reinforcement Learning

- Model-based RL
- Model Predictive Control
- Meta Learning
Safe Exploration

- Deal with real-world safety constraints (states/controls)
- Use probabilistic model to predict whether state constraints are violated (e.g., Sui et al., 2015; Berkenkamp et al., 2017)
- Adjust policy if necessary (during policy learning)
Safe Exploration

- Deal with real-world safety constraints (states/controls)
- Use probabilistic model to predict whether state constraints are violated (e.g., Sui et al., 2015; Berkenkamp et al., 2017)
- Adjust policy if necessary (during policy learning)
  - Safe exploration within an MPC-based RL setting
  - Optimize control signals $u_t$ directly (no policy parameters)
Approach

- Idea: Optimize control signals directly (instead of policy parameters)
- Few parameters to optimize ⇒ Low-dimensional search space
- Open-loop control
  ⇒ No chance of success (with minor model inaccuracies)
Approach

- Idea: Optimize control signals directly (instead of policy parameters)
- Few parameters to optimize ➤ Low-dimensional search space
- Open-loop control ➤ No chance of success (with minor model inaccuracies)
- Model Predictive Control (MPC) turns this into a closed-loop control approach
Approach

- Idea: Optimize control signals directly (instead of policy parameters)
- Few parameters to optimize ➤ Low-dimensional search space
- Open-loop control
  ➤ No chance of success (with minor model inaccuracies)
- Model Predictive Control (MPC) turns this into a closed-loop control approach
- Positive side-effect: Increase robustness to model errors (online approach) ➤ Increase data efficiency
Given a state $x_t$, plan (open loop) over a short horizon of length $H$ to get an open-loop control sequence $u^*_{t+0}, \ldots, u^*_{t+H-1}$.

After transitioning into a new state $x_{t+1}$, re-plan (as previously): Get $u^*_{t+1+0}, \ldots, u^*_{t+1+H-1}$ *closed-loop/feedback control*
- Given a state $x_t$, plan (open loop) over a short horizon of length $H$ to get an open-loop control sequence $u^*_{t+0}, \ldots, u^*_{t+H-1}$.
- After transitioning into a new state $x_{t+1}$, re-plan (as previously): Get $u^*_{t+1+0}, \ldots, u^*_{t+1+H-1}$ closed-loop/feedback control.
- Given a state $x_t$, plan (open loop) over a short horizon of length $H$ to get an open-loop control sequence $u^*_t+0, \ldots, u^*_{t+H-1}$
- After transitioning into a new state $x_{t+1}$, re-plan (as previously): Get $u^*_{t+1+0}, \ldots, u^*_{t+1+H-1}$  ▶ closed-loop/feedback control
Given a state $x_t$, plan (open loop) over a short horizon of length $H$ to get an open-loop control sequence $u^*_t, \ldots, u^*_{t+H-1}$.

After transitioning into a new state $x_{t+1}$, re-plan (as previously): Get $u^*_{t+1}, \ldots, u^*_{t+1+H-1}$  ➤ closed-loop/feedback control.
Given a state $x_t$, plan (open loop) over a short horizon of length $H$ to get an open-loop control sequence $u^*_{t+0}, \ldots, u^*_{t+H-1}$.

After transitioning into a new state $x_{t+1}$, re-plan (as previously): Get $u^*_{t+1+0}, \ldots, u^*_{t+1+H-1}$ — closed-loop/feedback control.
Given a state $x_t$, plan (open loop) over a short horizon of length $H$ to get an open-loop control sequence $u^*_t + 0, \ldots, u^*_{t+H-1}$

After transitioning into a new state $x_{t+1}$, re-plan (as previously): Get $u^*_{t+1+0}, \ldots, u^*_{t+1+H-1}$  » closed-loop/feedback control

Use this within a trial-and-error model-based RL setting
Learn GP model for transition dynamics

Repeat (while executing the policy):

1. In current state $x_t$, determine optimal control sequence $u_0^*, \ldots, u_{H-1}^*$
2. Apply first control $u_0^*$ in state $x_t$
3. Transition to next state $x_{t+1}$
4. Update GP transition model with observed transition $(x_t, u_0^*, x_{t+1})$
Theoretical Results

- Uncertainty propagation is deterministic (GP moment matching)
  - Re-formulate system dynamics:

  \[
  z_{t+1} = f_{MM}(z_t, u_t)
  \]

  \[
  z_t = \{\mu_t, \Sigma_t\} \quad \text{Collects moments}
  \]
Theoretical Results

- **Uncertainty propagation is deterministic** (GP moment matching)
  - Re-formulate system dynamics:
    \[ z_{t+1} = f_{MM}(z_t, u_t) \]
    \[ z_t = \{\mu_t, \Sigma_t\} \]  
    **Collects moments**

- **Deterministic** system function that propagates moments

- Lipschitz continuity (under mild assumptions) implies that we can apply **Pontryagin’s minimum principle**
  - **Principled treatment of constraints** on controls
Theoretical Results

- **Uncertainty propagation is deterministic** (GP moment matching)
  - Re-formulate system dynamics:
    \[ z_{t+1} = f_{MM}(z_t, u_t) \]
    \[ z_t = \{\mu_t, \Sigma_t\} \]  
    - Collects moments

- **Deterministic** system function that propagates moments

- Lipschitz continuity (under mild assumptions) implies that we can apply **Pontryagin’s minimum principle**
  - Principled treatment of **constraints** on controls

- Use predictive uncertainty to check violation of **state constraints**
- **Zero-Var**: Only use the mean of the GP, discard variances for long-term predictions

- **MPC**: Increased data efficiency (40% less experience required than PILCO)
  - MPC more robust to model inaccuracies than a parametrized feedback controller

Kamthe & Deisenroth (AISTATS, 2018): *Data-Efficient Reinforcement Learning with Probabilistic Model Predictive Control*
GP-MPC maintains the same improvement in data efficiency

Zero-Var fails:
- Gets stuck in local optimum near start state
- Insufficient exploration due to lack of uncertainty propagation

Kamthe & Deisenroth (AISTATS, 2018): Data-Efficient Reinforcement Learning with Probabilistic Model Predictive Control
GP-MPC maintains the same improvement in data efficiency

Zero-Var fails:
- Gets stuck in local optimum near start state
- Insufficient exploration due to lack of uncertainty propagation

Although MPC is fairly robust to model inaccuracies we cannot get away without uncertainty propagation

Kamthe & Deisenroth (AISTATS, 2018): Data-Efficient Reinforcement Learning with Probabilistic Model Predictive Control
Safety Constraints (Cart Pole)

Propagating model uncertainty important for safety

Kamthe & Deisenroth (AISTATS, 2018): *Data-Efficient Reinforcement Learning with Probabilistic Model Predictive Control*
- Probabilistic prediction models for safe exploration
- Uncertainty propagation can be used to reduce violation of safety constraints
- MPC framework increases robustness to model errors
  - Increased data efficiency
Data-efficient Reinforcement Learning

- Model-based RL
- Model Predictive Control
- Meta Learning
Meta Learning

Generalize knowledge from known tasks to new (related) tasks
Meta Learning

Generalize knowledge from known tasks to new (related) tasks

- Different robot configurations (link lengths, weights, ...)
- Re-use experience gathered so far generalize learning to new dynamics that are similar
  ▶️ Accelerated learning
Approach

- **Separate** global and task-specific properties
- **Shared** global parameters describe general dynamics
- **Describe** task-specific (local) configurations with latent variable
Approach

- **Separate** global and task-specific properties
- **Shared** global parameters describe general dynamics
- Describe **task-specific** (local) configurations with **latent variable**
- **Online variational inference** of (unseen) configurations
- **Few-shot model-based RL**
Meta Model Learning with Latent Variables

\[ y_t = f(x_t, u_t, h_p) \]

Sæmundsson et al. (UAI, 2018): *Meta Reinforcement Learning with Latent Variable Gaussian Processes*
Meta Model Learning with Latent Variables

\[ y_t = f(x_t, u_t, h_p) \]

- GP captures global properties of the dynamics

Sæmundsson et al. (UAI, 2018): Meta Reinforcement Learning with Latent Variable Gaussian Processes
\[ y_t = f(x_t, u_t, h_p) \]

- GP captures **global properties** of the dynamics
- Latent variable \( h_p \) describes **local configuration**
  - Variational inference to find a posterior on latent configuration

Sæmundsson et al. (UAI, 2018): *Meta Reinforcement Learning with Latent Variable Gaussian Processes*
Meta Model Learning with Latent Variables

\[ y_t = f(x_t, u_t, h_p) \]

- GP captures global properties of the dynamics
- Latent variable \( h_p \) describes local configuration
  - Variational inference to find a posterior on latent configuration
- Fast online inference of new configurations (no model re-training required)

Sæmundsson et al. (UAI, 2018): *Meta Reinforcement Learning with Latent Variable Gaussian Processes*
- Latent variable $h$ encodes length $l$ and mass $m$ of the cart pole
- 6 training tasks, 14 held-out test tasks

Sæmundsson et al. (UAI, 2018): *Meta Reinforcement Learning with Latent Variable Gaussian Processes*
Meta-RL (Cart Pole): Training

- Pre-trained on 6 training configurations until solved

<table>
<thead>
<tr>
<th>Model</th>
<th>Training (s)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Independent</td>
<td>16.1 ± 0.4</td>
<td>Independent GP-MPC</td>
</tr>
<tr>
<td>Aggregated</td>
<td>23.7 ± 1.4</td>
<td>Aggregated experience (no latents)</td>
</tr>
<tr>
<td>Meta learning</td>
<td><strong>15.1 ± 0.5</strong></td>
<td>Aggregated experience (with latents)</td>
</tr>
</tbody>
</table>

Meta learning can help speeding up RL

Sæmundsson et al. (UAI, 2018): *Meta Reinforcement Learning with Latent Variable Gaussian Processes*
- Few-shot generalization on 4 unseen configurations
- Success: solve all 10 (6 training + 4 test) tasks
- **Meta learning**: blue
- **Independent (GP-MPC)**: orange
- **Aggregated experience model (no latents)**: green

➤ **Meta RL generalizes well to unseen tasks**

Sæmundsson et al. (UAI, 2018): *Meta Reinforcement Learning with Latent Variable Gaussian Processes*
Generalize knowledge from known situations to unseen ones

- **Few-shot learning**

- Latent variable can be used to infer task similarities

- Significant speed-up in model learning and model-based RL
Team and Collaborators

Marc Deisenroth (UCL)

Data-Efficient Reinforcement Learning for Autonomous Robots

November 20, 2019
Data efficiency is a practical challenge for autonomous robots.

Three pillars of data-efficient reinforcement learning for autonomous robots:

1. **Model-based reinforcement learning** with learned probabilistic models for fast learning from scratch.
2. **Model predictive control** with learned dynamics models accelerate learning and allow for safe exploration.
3. **Meta learning** using latent variables to generalize knowledge to new situations.

Key to success: Probabilistic modeling and Bayesian inference.
Data efficiency is a practical challenge for autonomous robots.

Three pillars of data-efficient reinforcement learning for autonomous robots:

1. Model-based reinforcement learning with learned probabilistic models for fast learning from scratch.
2. Model predictive control with learned dynamics models accelerate learning and allow for safe exploration.
3. Meta learning using latent variables to generalize knowledge to new situations.

Key to success: Probabilistic modeling and Bayesian inference.

Thank you for your attention.

ありがとうございます。


Controller Parametrization

Controller:

\[
\tilde{\pi}(x, \theta) = \sum_{k=1}^{K} w_k \exp \left( -\frac{1}{2} (x - \mu_k)^\top \Lambda (x - \mu_k) \right)
\]

\[
u = \pi(x, \theta) = u_{\text{max}} \sigma(\tilde{\pi}(x, \theta)) \in [-u_{\text{max}}, u_{\text{max}}],
\]

where \(\sigma\) is a squashing function.
Controller Parametrization

- Controller:

\[
\tilde{\pi}(\mathbf{x}, \theta) = \sum_{k=1}^{K} w_k \exp \left( - \frac{1}{2} (\mathbf{x} - \mu_k)\Lambda (\mathbf{x} - \mu_k) \right)
\]

\[u = \pi(\mathbf{x}, \theta) = u_{\text{max}} \sigma(\tilde{\pi}(\mathbf{x}, \theta)) \in [-u_{\text{max}}, u_{\text{max}}],\]

where \(\sigma\) is a squashing function.

- Parameters:

\[\theta := \{w_k, \mu_k, \Lambda\}\]
Controller Parametrization

- Controller:

\[
\tilde{\pi}(x, \theta) = \sum_{k=1}^{K} w_k \exp \left( -\frac{1}{2}(x - \mu_k)^\top \Lambda (x - \mu_k) \right)
\]

\[
u = \pi(x, \theta) = u_{\max} \sigma(\tilde{\pi}(x, \theta)) \in [-u_{\max}, u_{\max}]
\]

where \(\sigma\) is a squashing function.

- Parameters:

\[
\theta := \{w_k, \mu_k, \Lambda\}
\]

- Squashing function:

\[
\sigma(z) = \frac{9}{8} \sin(z) + \frac{1}{8} \sin(3z)
\]
Cost Functions

- **Quadratic cost** \( c(x) = (x - x_{\text{target}})^\top W (x - x_{\text{target}}) \)
- **Saturating cost** \( c(x) = 1 - \exp \left( - (x - x_{\text{target}})^\top W (x - x_{\text{target}}) \right) \)

- Quadratic cost pays a lot of attention to states “far away”
  - **Bad idea for exploration**
Task: Minimize $\mathbb{E}[c(x_t)]$

- In the early stages of learning, state predictions are expected to be far away from the target.

Exploration favored

In the final stages of learning, state predictions are expected to be close to the target

Exploitation favored

Bayesian treatment: Natural exploration/exploitation trade-off
Natural Exploration with the Saturating Cost

Task: Minimize $\mathbb{E}[c(x_t)]$

In the early stages of learning, state predictions are expected to be far away from the target → Exploration favored
Natural Exploration with the Saturating Cost

Task: Minimize $\mathbb{E}[c(x_t)]$

In the early stages of learning, state predictions are expected to be far away from the target. Exploration favored.

In the final stages of learning, state predictions are expected to be close to the target.
Task: Minimize $\mathbb{E}[c(x_t)]$

- In the **early stages of learning**, state predictions are expected to be far away from the target ▶ Exploration favored
- In the **final stages of learning**, state predictions are expected to be close to the target ▶ Exploitation favored
Task: Minimize $\mathbb{E}[c(x_t)]$

In the early stages of learning, state predictions are expected to be far away from the target » Exploration favored

In the final stages of learning, state predictions are expected to be close to the target » Exploitation favored

» Bayesian treatment: Natural exploration/exploitation trade-off
GP Moment Matching: Some Details

\[ f \sim GP(0, k), \quad \text{Training data: } X, y \]
\[ x_* \sim \mathcal{N}(\mu, \Sigma) \]

- Compute \( \mathbb{E}[f(x_*)] \)
\[ f \sim GP(0, k), \quad \text{Training data: } X, y \]
\[ x_* \sim \mathcal{N}(\mu, \Sigma) \]

- Compute \( \mathbb{E}[f(x_*)] \)

\[
\mathbb{E}_{f, x_*}[f(x_*)] = \mathbb{E}_x\left[ \mathbb{E}_f[f(x_*)|x_*] \right] = \mathbb{E}_{x_*} \left[ m_f(x_*) \right]
\]
$f \sim \text{GP}(0, k)$, \quad \text{Training data: } X, y$

$x_* \sim \mathcal{N}(\mu, \Sigma)$

- Compute $\mathbb{E}[f(x_*)]$

$$\mathbb{E}_{f, x_*}[f(x_*)] = \mathbb{E}_x[\mathbb{E}_f[f(x_*)|x_*]] = \mathbb{E}_{x_*}[m_f(x_*)]$$

$$= \mathbb{E}_{x_*}[k(x_*, X)(K + \sigma^2_n I)^{-1}y]$$
GP Moment Matching: Some Details

\[ f \sim GP(0, k), \quad \text{Training data: } X, y \]
\[ x_\ast \sim N(\mu, \Sigma) \]

- Compute \( \mathbb{E}[f(x_\ast)] \)

\[
\mathbb{E}_{f, x_\ast}[f(x_\ast)] = \mathbb{E}_x[\mathbb{E}_f[f(x_\ast)|x_\ast]] = \mathbb{E}_{x_\ast}[m_f(x_\ast)]
\]
\[
= \mathbb{E}_{x_\ast}[k(x_\ast, X)(K + \sigma_n^2 I)^{-1} y]
\]
\[
= \beta^\top \int k(X, x_\ast)N(x_\ast | \mu, \Sigma) dx_\ast
\]
\[
\beta := (K + \sigma_n^2 I)^{-1} y \quad \text{\( \gg \) independent of } x_\ast
\]
GP Moment Matching: Some Details

\begin{align*}
f & \sim GP(0, k) , \quad \text{Training data: } X, y \\
x_* & \sim \mathcal{N}(\mu, \Sigma)
\end{align*}

- Compute \( \mathbb{E}[f(x_*)] \)

\[
\mathbb{E}_{f,x_*}[f(x_*)] = \mathbb{E}_x[\mathbb{E}_f[f(x_*)|x_*]] = \mathbb{E}_{x_*}[m_f(x_*)] \\
= \mathbb{E}_{x_*}[k(x_*, X)(K + \sigma_n^2 I)^{-1}y] \\
= \beta^\top \int k(X, x_*) \mathcal{N}(x_* | \mu, \Sigma) \, dx_*
\]

\[
\beta := (K + \sigma_n^2 I)^{-1}y \quad \blacktriangleright \text{independent of } x_*
\]

- If \( k \) is a Gaussian (squared exponential) kernel, this integral can be solved analytically

- Variance of \( f(x_*) \) can be computed similarly
Meta Learning Model

\[ f(\cdot) \sim GP \]

\[ p(H) = \prod_{p} p(h_p), \quad p(h_p) = \mathcal{N}(0, I) \]
Meta Learning Model

\[ f(\cdot) \sim GP \]

\[ p(H) = \prod_p p(h_p), \quad p(h_p) = \mathcal{N}(0, I) \]

\[ p(Y, H, f(\cdot)|X, U) = \prod_{p=1}^P p(h_p) \prod_{t=1}^{T_p} p(y_t|x_t, u_t, h_p, f(\cdot))p(f(\cdot)) \]

\[ y_t = x_{t+1} - x_t \]
Mean-field variational family:

\[ q(f(\cdot), H) = q(f(\cdot))q(H) \]

\[ q(H) = \prod_{p=1}^{P} \mathcal{N}(h_p|n_p, T_p), \]

\[ q(f(\cdot)) = \int p(f(\cdot)|f_Z)q(f_Z)df_Z \quad \text{SV-GP (Titsias, 2009)} \]
Evidence Lower Bound

$$ELBO = \mathbb{E}_{q(f(\cdot), H)} \left[ \log \frac{p(Y, H, f(\cdot)|X, U)}{q(f(\cdot), H)} \right]$$
Evidence Lower Bound

$$ELBO = \mathbb{E}_{q(f(\cdot), H)} \left[ \log \frac{p(Y, H, f(\cdot)|X, U)}{q(f(\cdot), H)} \right]$$

$$= \sum_{p=1}^{P} \sum_{t=1}^{T_p} \mathbb{E}_{q(f_t|x_t,u_t,h_p)q(h_p)} \left[ \log p(y_t|f_t) \right] - KL(q(H)||p(H)) - KL(q(f(\cdot)||p(f(\cdot)))$$
\[ ELBO = E_{q(f(\cdot), H)} \left[ \log \frac{p(Y, H, f(\cdot)|X, U)}{q(f(\cdot), H)} \right] \]

\[
= \sum_{p=1}^{P} \sum_{t=1}^{T_p} E_{q(f_t|x_t, u_t, h_p)} q(h_p) \left[ \log p(y_t|f_t) \right] - KL(q(H)||p(H)) - KL(q(f(\cdot)||p(f(\cdot)))
\]

Monte Carlo estimate

\[
= \sum_{p=1}^{P} \sum_{t=1}^{T_p} \left[ E_{q(f_t|x_t, u_t, h_p)} q(h_p) \left[ \log p(y_t|f_t) \right] \right]
- KL(q(H)||p(H)) - KL(q(F_Z)||p(F_Z))
\]

closed-form solution