Forgetting to Improve

Principled Data Removal in Active Learning


Manuel Wendl[1,2]


Erik Englesson[2]


Andreas Krause[1]


Carl Henrik Ek[2]

[1] ETH Zürich, ETH AI Center - [2] University of Cambridge

ETH zürich UNIVERSITY OF CAMBRIDGE

The Missing Delete Key

Do you have time for a meeting today?

```
I don't have time today
```

Which data is **detrimental** and how do we **forget** it to improve learning?

Problem Setting

🧩 The Model

Gaussian process prior on $\mathcal{X}$:

$$f \sim \mathcal{GP}(0, k)$$

Noisy observations $y_i = f(x_i) + \epsilon_i$, with a **Gamma prior** on the noise variance:

$$\epsilon_i \sim \mathcal{N}(0, \sigma_w^2), \qquad \sigma_w^2(\mathcal{S}) \sim \Gamma(a, b)$$

🎯 The Metric

Predictive uncertainty of a retained dataset $\mathcal{S}' \subseteq \mathcal{S}$ over a **target region** $\mathcal{A} \subseteq \mathcal{X}$:

$$\mathcal{J}(\mathcal{S}') = \int_{\mathcal{A}} \sigma^2_{\mathcal{S}'}(x)\, dx$$

Less data can mean less uncertainty.

🔑 **Objective:** Choose $\mathcal{S}' \subseteq \mathcal{S}$ to minimise $\mathcal{J}(\mathcal{S}')$.

Removing Data Points from GPs

How does the prediction change if we remove a data point?

Removal Direction:

Continuous removal direction of the data point under fixed noise level σ_w:

$$k(x, \mathcal{S}, h_i) := k(x, \mathcal{S}) - h_i k(x, x_i) e_i, \qquad y(h_i) = y - h_i y_i e_i,$$

Representer theorem [1]: $f = \sum_{i=1}^{n} \alpha_i k(\cdot, x_i)$

LOO [2] ($h_i \to 1$) - Unchanged α^* for the retained data points.

---

1 Bernhard Schölkopf, Ralf Herbrich, and Alex J Smola. A generalized representer theorem. In International conference on computational learning theory, Springer, 2001.

2 Carl Edward Rasmussen and Christopher K. I. Williams. Gaussian Processes for Machine Learning. The MIT Press, 2005.

Data Removal Revisited

Aleatoric Noise is typically considered as **irreducible**.
Model-Data mismatch is absorbed by this aleatoric noise as well.

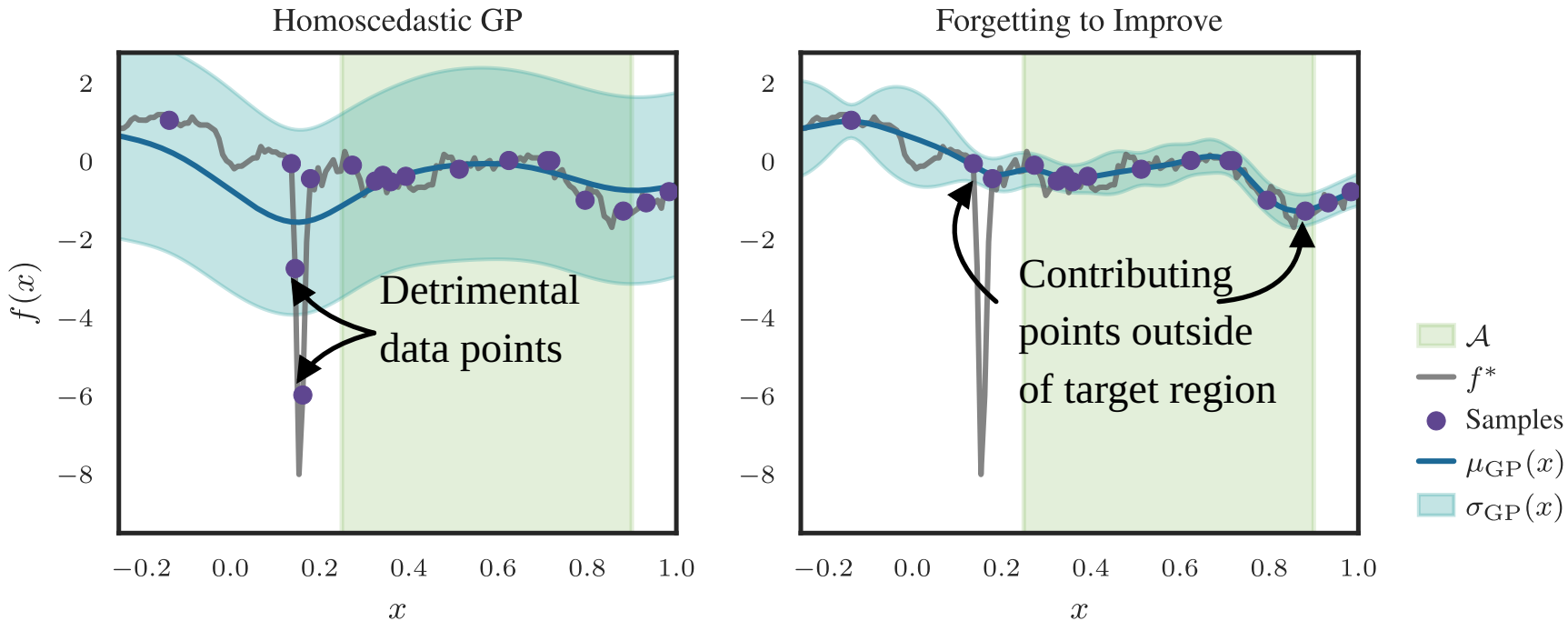
⚠️ We **break** with the classical **irreducibility** assumption of the **aleatoric noise**.
=> Trade off global aleatoric noise with local epistemic uncertainty.

💡 **Main idea:** combine how

1. Epistemic noise changes with the removal (fixed σ_w),
2. Aleatoric noise changes with the removal (fixed K).

$$\sigma^2_{\mathcal{S}'}(x) = k(x,x) - k_{x,\mathcal{S}'}(K + \sigma^2_w(\mathcal{S}')I)^{-1}k_{\mathcal{S}',x} + \sigma^2_w(\mathcal{S}'),$$

Trading Aleatoric Noise for Epistemic Uncertainty



Large global aleatoric noise

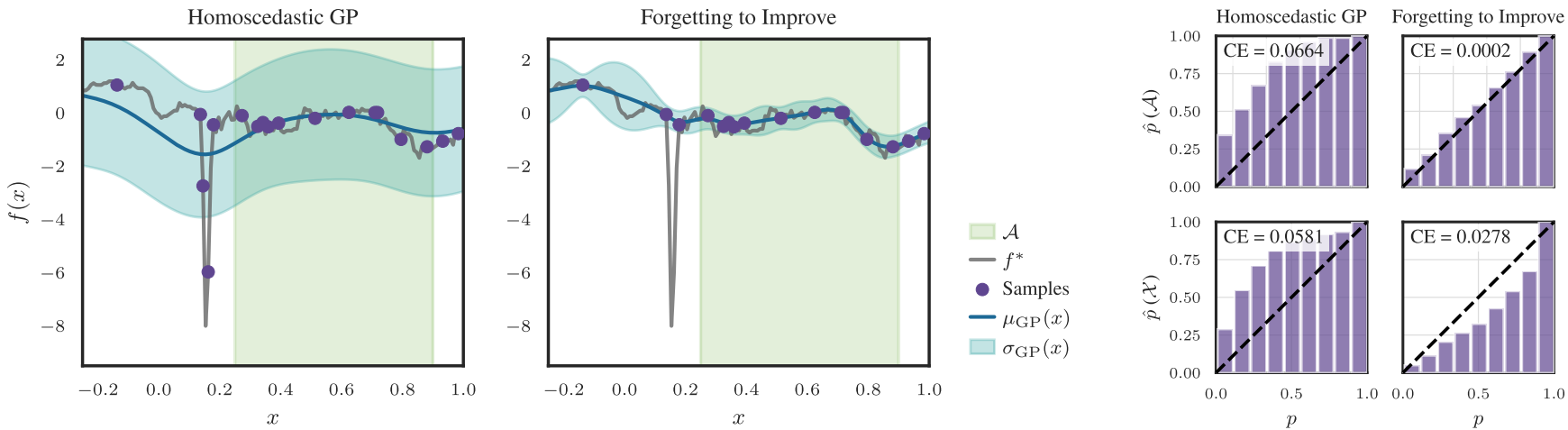
lowest possible epistemic uncertainty

Small global aleatoric noise

large epistemic uncertainty in detrimental region

Improved Calibration on Target Region $\mathcal{A}$

Improved calibration in the target region $\mathcal{A}$ (top row).



But **globally**, we trade this off with an **overconfident** global prediction (bottom row).

What is the Influence of a Sample?

Reconsider our **objective**:

$$\min_{\mathcal{S}' \subseteq \mathcal{S}} \mathcal{J}(\mathcal{S}') := \int_{\mathcal{A}} k(x,x) - k_{x,\mathcal{S}'}(K + \sigma_w^2(\mathcal{S}')I)^{-1} k_{\mathcal{S}',x} + \sigma_w^2(\mathcal{S}') dx.$$

How does a removal influence the predictive uncertainty in the target region $\mathcal{A}$?

Gradient of the Objective in Removal Direction: $\frac{\partial}{\partial h_i} \mathcal{J}\left(\mathcal{S}^{(k)}\right)\Big|_{h_i=0} = u_i(\mathcal{A}) + v_i(\mathcal{A}).$ Composed out of two terms:

- $u_i(\mathcal{A})$: Epistemic Influence of a Sample
- $v_i(\mathcal{A})$: Aleatoric Influence of a Sample

Epistemic Influence of a Sample

How does the epistemic uncertainty change if we remove a data point? (fixed σ_w)

Epistemic Influence Measure:

For a test point x, we define with $k_{x,\mathcal{S}} = k(x, \mathcal{S}),$ and $A = K + \sigma_w^2(\mathcal{S})I$

$$u_i(x) = 2k_{x,x_i}(A^{-1}k_{\mathcal{S},x})_i - 2(A^{-1}k_{\mathcal{S},x})_i(k_{x_i,\mathcal{S}}A^{-1}k_{\mathcal{S},x})$$

u_i is the directional derivative of the LOO epistemic uncertainty in the direction k_{x,x_i}.

$$u_i(x) = k_{x,x_i}\frac{\partial \Delta_i(x)}{\partial k_{x,x_i}}, \quad \Delta_i(x) = \sigma^2_{\mathcal{S}\setminus x_i}(x) - \sigma^2_{\mathcal{S}}(x) = \frac{(A^{-1}k_{\mathcal{S},x})_i^2}{A^{-1}_{i,i}}.$$

Aleatoric Uncertainty under Data Removal

How does the aleatoric noise change if we remove a data point?

MAP Type II: the noise level is fit by maximizing the regularized data likelihood

$$\sigma_w^2(\mathcal{S}) = \arg\max_{\sigma_w^2} \mathcal{L}(\sigma_w^2, \mathcal{S}) \coloneqq \log \prod_{i=1}^{n} p(y_i \mid x_i, \sigma_w^2) + \log p(\sigma_w^2).$$

Aleatoric Uncertainty Derivative:

Implicitly differentiating the first-order optimality condition of $\mathcal{L}(\sigma_w^2, \mathcal{S})$, the **optimal noise level** changes

$$\frac{\partial \sigma_w^2(\mathcal{S})}{\partial h_i} = \frac{-(A^{-1}y)_i (k_{x_i,\mathcal{S}}(A^{-2}y)) - (A^{-2}y)_i (k_{x_i,\mathcal{S}}(A^{-1}y)) + k_{x_i,\mathcal{S}}(A^{-2})_{\cdot,i} + y_i (A^{-2}y)_i}{-y^\top A^{-3} y + \frac{1}{2}\mathrm{tr}(A^{-2}) - \frac{a-1}{(\sigma_w^2)^2}}.$$

Aleatoric Influence of a Sample

How do **all aleatoric noise terms** change if we remove a data point? (fixed K)

Aleatoric Influence Measure:

We obtain for all test points x **globally** the **same influence** of removing a sample and keeping K fixed:

$$v_i(x) = (1 + k_{x,\mathcal{S}} A^{-2} k_{\mathcal{S},x}) \frac{\partial \sigma_w^2(\mathcal{S})}{\partial h_i},$$

We observe that the **aleatoric uncertainty** appears in two places in the predictive uncertainty $\sigma^2_{\mathcal{S}'}(x) = k(x,x) - k_{x,\mathcal{S}'}(K + \sigma_w^2(\mathcal{S}')I)^{-1} k_{\mathcal{S}',x} + \sigma_w^2(\mathcal{S}')$, leading to the two terms.

Forgetting to Improve: Combined Influence

💡 Both Influence Measures live in the same predictive uncertainty space.

From previous definitions, we obtain that $\frac{\partial}{\partial h_i}\mathcal{J}\left(\mathcal{S}^{(k)}\right)\Big|_{h_i=0} = u_i(\mathcal{A}) + v_i(\mathcal{A}).$

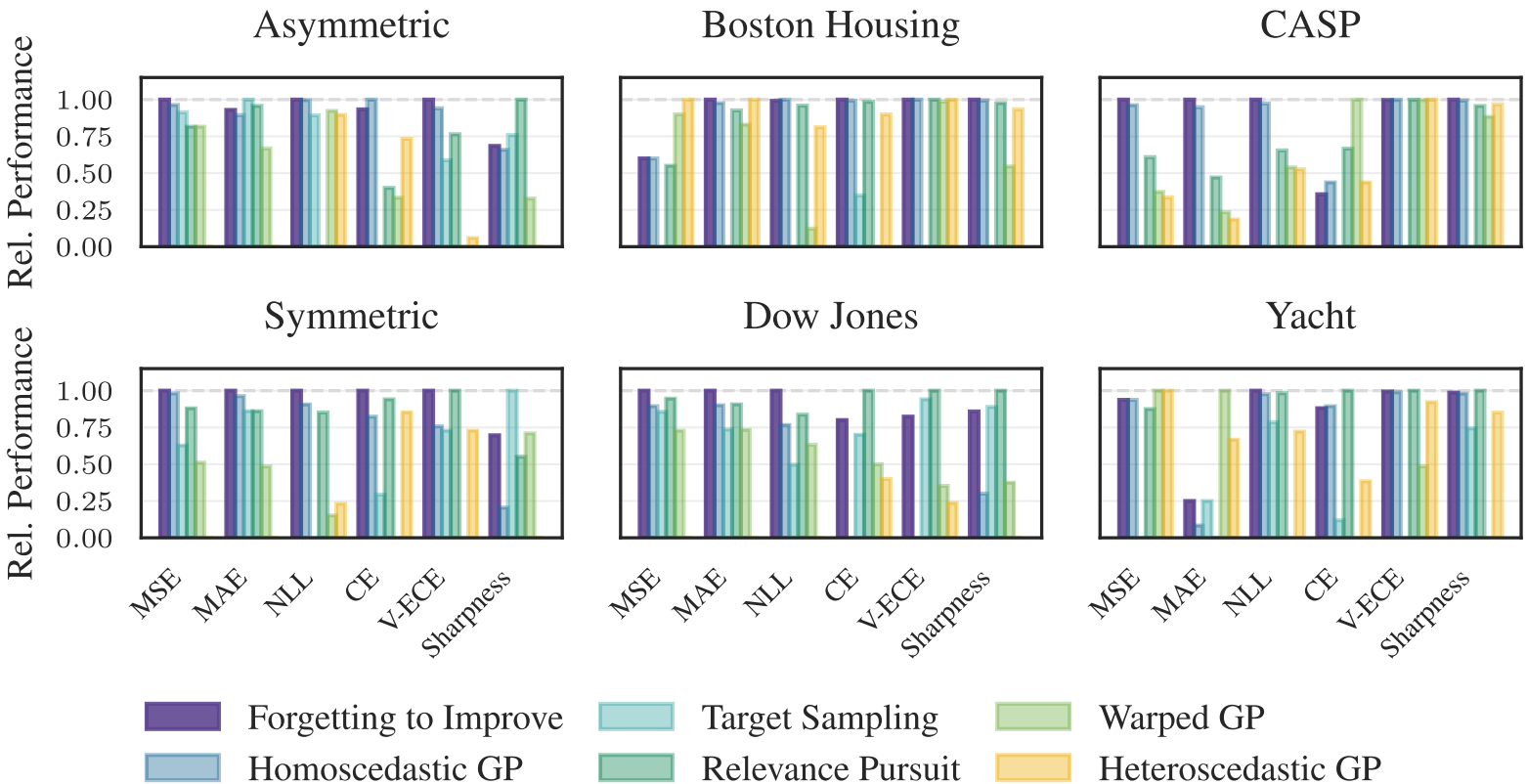
Algorithm 1 Forgetting to Improve

Require: Target region $\mathcal{A}$, all samples $\mathcal{S}$, and the GP model with kernel k.
Initialize $h = 0$, and $\mathcal{S}^{(0)} \leftarrow \mathcal{S}$.
Compute influence measures $u^{(0)}(\mathcal{A})$, and $v^{(0)}(\mathcal{A})$
while $\min(u^{(h)}(\mathcal{A}) + v^{(h)}(\mathcal{A})) \leq 0$ **do**
  Determine worst sample: $i = \arg\min_j(u_j^{(h)}(\mathcal{A}) + v_j^{(h)}(\mathcal{A}))$
  Update set of samples $\mathcal{S}^{(h+1)} \leftarrow \mathcal{S}^{(h)} \setminus x_i$ and GP noise MAP
  Increase $h \leftarrow h + 1$ and compute influences: $u^{(h)}(\mathcal{A}), v^{(h)}(\mathcal{A})$
end while
return $\mathcal{S}' = \mathcal{S}^{(h+1)}$

Termination: $\forall i : u_i + v_i > 0$

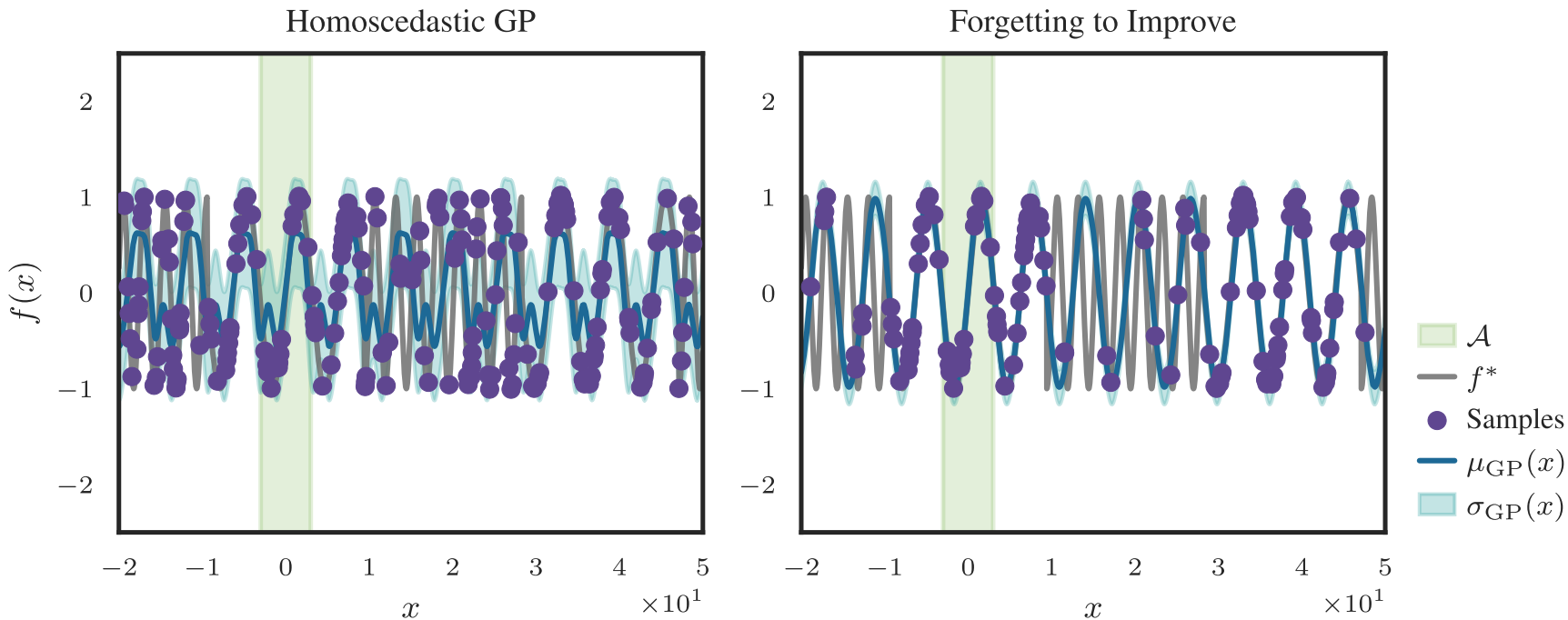
Greedy $i = \arg\min_j(u_j + v_j)$

It Works – Transductive Experiments



How do "global kernels" behave?

Peridic kernel (1D): $k(x, x') = \exp(-\sin^2(\frac{\pi(x-x')}{p})/l^2)$ with lengthscale l, and periodicity p.



Bayesian Optimization

Bayesian Optimization aims to solve

$$x^* = \arg\max_{x \in \mathcal{X}} f(x),$$

for an **unknown** objective function f, observed only through evaluations.

Classical BO pipeline:

1. Fit a surrogate posterior (μ_t, σ_t),
2. Choose the next query with an acquisition function $a_t(x; \mu_t, \sigma_t)$,
3. Evaluate and update.

⚠️ Most acquisition-function theory assumes the surrogate posterior is informative. If the surrogate is poor, acquisition decisions can systematically fail.

💡 Instead of only designing better acquisition rules, we improve **decision quality** by improving posterior predictions in the **regions of interest** (informative for optimum).

Forgetful Bayesian Optimization

Bayesian Optimization as a sequential transductive learning problem.

We define in each iteration a target region of potential improvement:

$$\mathcal{A}_t^{(k)} \leftarrow \{x_i \in \mathcal{X} : \underbrace{\mu_{\mathcal{S}_t^{(k)}}(x_i) + \beta_t \sigma_{\mathcal{S}_t^{(k)}}(x_i)}_{\mathrm{UCB}(x_i)} > \max_{x_m \in \mathcal{X}} \underbrace{\mu_{\mathcal{S}_t^{(k)}}(x_m) - \beta_t \sigma_{\mathcal{S}^{(k)}}(x_m)}_{\mathrm{LCB}(x_m)}\} \cap \mathcal{A}_t^{(k-1)}$$

💡 Note that we update the target region $\mathcal{A}_t^{(k)}$ during the removal, with the improved uncertainty estimates. We update by intersecting with the previous target region $\mathcal{A}_t^{(k-1)}$.

Very Important: Removing points inflates confidence bounds outside of $\mathcal{A}_t^{(k)}$, so that in the next iteration, we do not select points outside of the target region, due to the "traded" epistemic uncertainty.

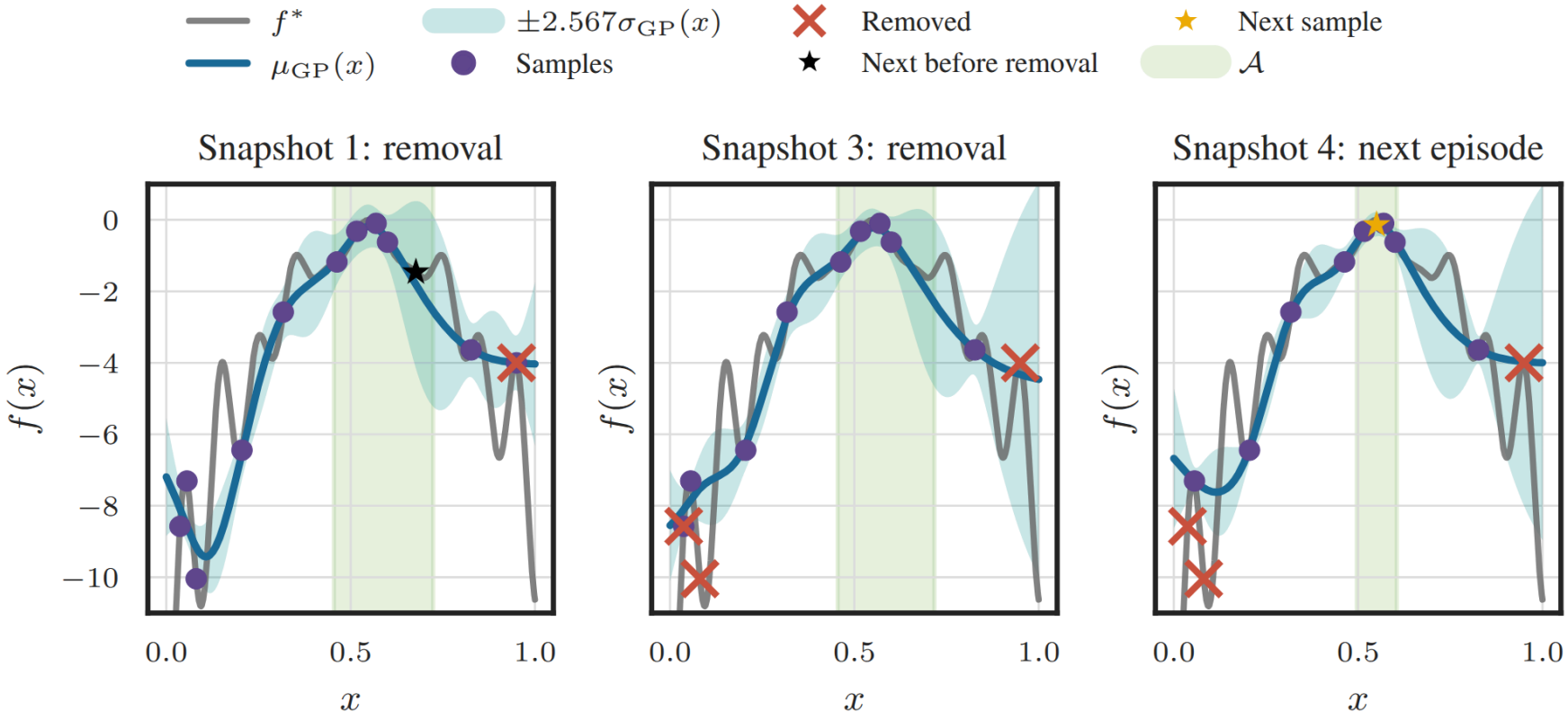
Forgetful BO – Algorithm

Algorithm 2 Forgetful BO

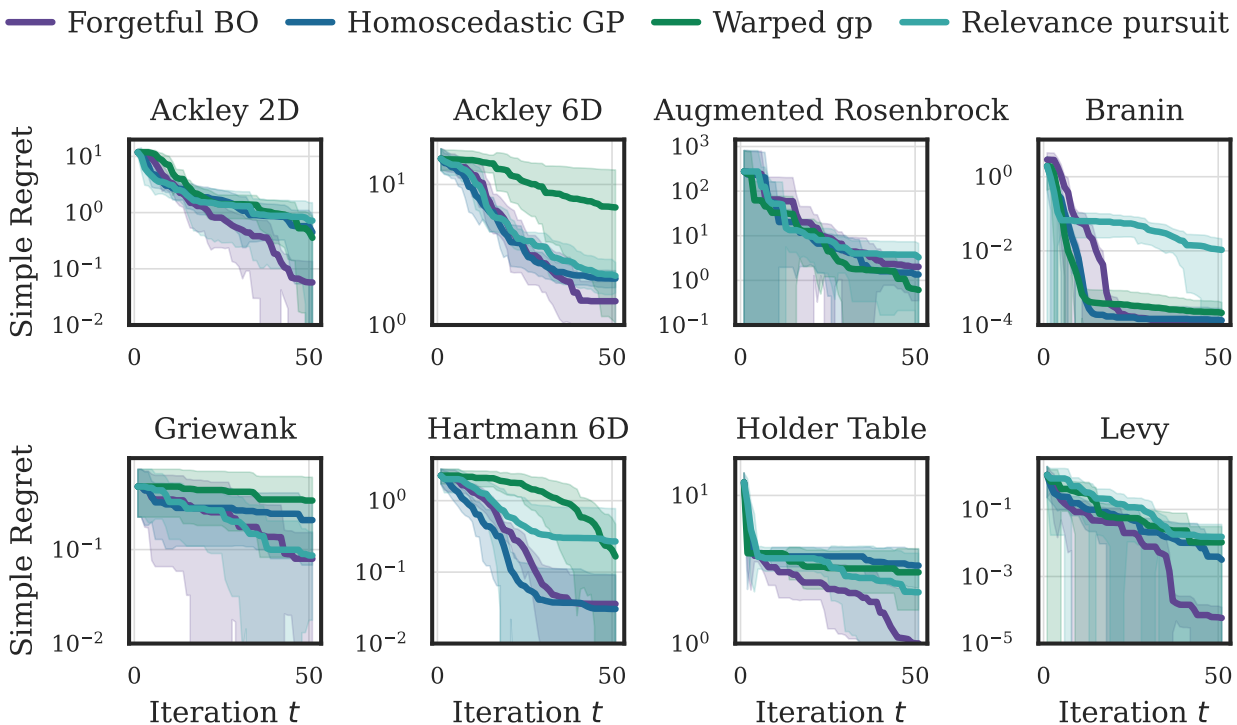
Require: Objective f, search space $\mathcal{X}$, acquisition function α, initial dataset size n_0, budget T
  Initialize dataset $\mathcal{S}_0 = \{(x_i, y_i)\}_{i=1}^{n_0}$ with $y_i = f(x_i)$
  Fit initial GP model $(\mu_{\mathcal{S}_0}(\cdot), \sigma_{\mathcal{S}_0}(\cdot))$ on $\mathcal{S}_0$
  Set $x^* = \arg\max_{x_i \in \mathcal{S}_0} y_i$, $y^* = \max_{x_i \in \mathcal{S}_0} y_i$
  for $t = 1, \ldots, T$ **do**
    Optimize acquisition function: $x_t = \arg\max_{x \in \mathcal{X}} \alpha(x | (\mu_{\mathcal{S}'_{t-1}}(\cdot), \sigma_{\mathcal{S}'_{t-1}}(\cdot)))$
    Evaluate objective: $y_t = f(x_t)$ and update $\mathcal{S}_t = \mathcal{S}_{t-1} \cup \{(x_t, y_t)\}$
    $(x^*, y^*) \leftarrow \max((x_t, y_t), (x^*, y^*))$
    Initialize $k = 0$, $\mathcal{S}_t^{(0)} = \mathcal{S}_t$ and fit GP model
    Determine $\mathcal{A}_t^{(0)}$ and compute $u^{(0)}(\mathcal{A}_t^{(0)})$, $v^{(0)}(\mathcal{A}_t^{(0)})$ ▷ Equation (3) and propositions 1 and 4
    while $\min(u^{(k)}(\mathcal{A}_t^{(k)}) + v^{(k)}(\mathcal{A}_t^{(k)})) \leq 0$ **do**
      Determine worst sample: $i = \arg\min_j (u_j^{(k)}(\mathcal{A}_t^{(k)}) + v_j^{(k)}(\mathcal{A}_t^{(k)}))$
      Update $\mathcal{S}_t^{(k+1)} = \mathcal{S}_t^{(k)} \setminus x_i$ and refit GP noise MAP
      Update $k \leftarrow k + 1$, $\mathcal{A}_t^{(k)}$, $u^{(k)}(\mathcal{A}_t^{(k)})$, $v^{(k)}(\mathcal{A}_t^{(k)})$ ▷ Equation (3) and propositions 1 and 4
    end while
    Set $\mathcal{S}'_t = \mathcal{S}_t^{(k)}$
  end for
  return x^*, y^*

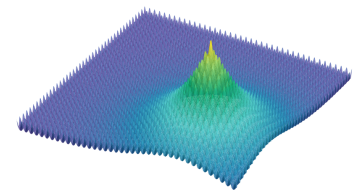
Simple Integration:
Standard BO pipeline + Removal and adapting target region

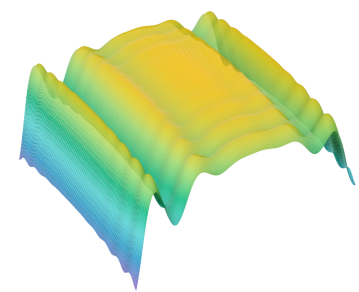
Forgetful BO – Intuition



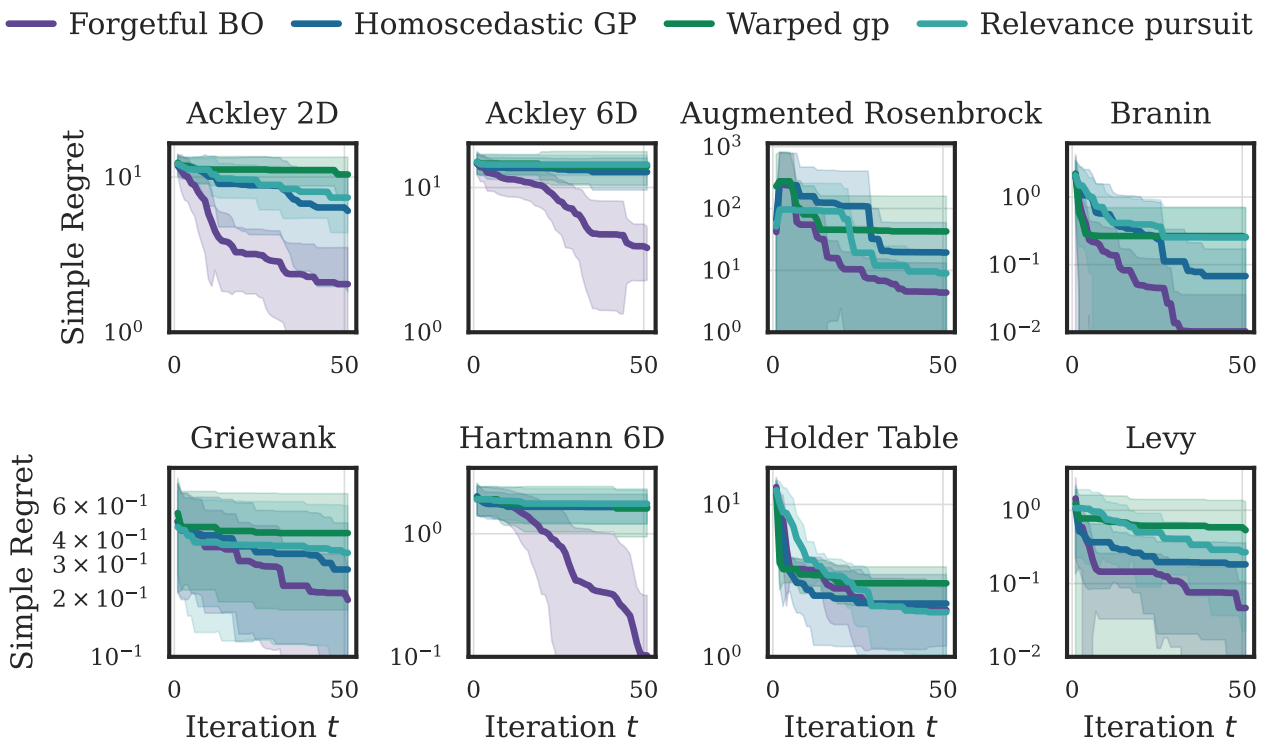
Forgetful BO – Experiments





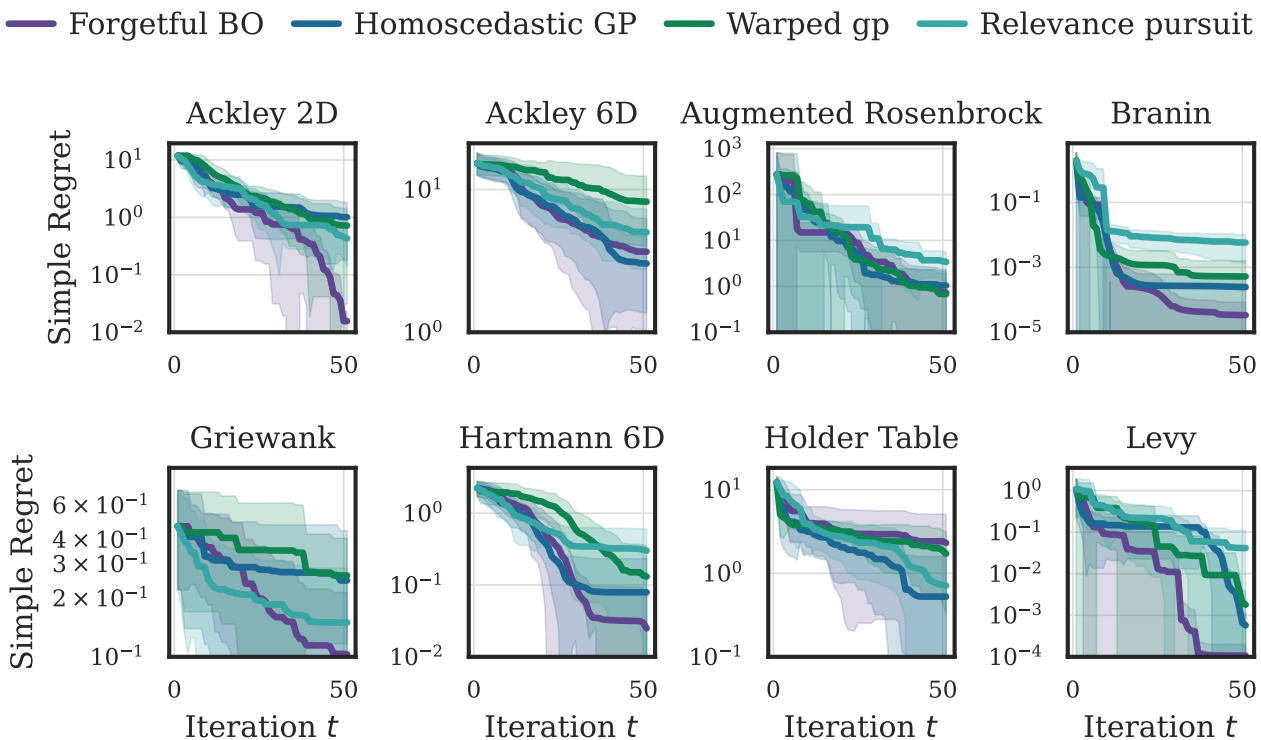


Forgetful BO – Noisy Experiments



Robust against outliers and noise. Successful removal of detrimental noise.

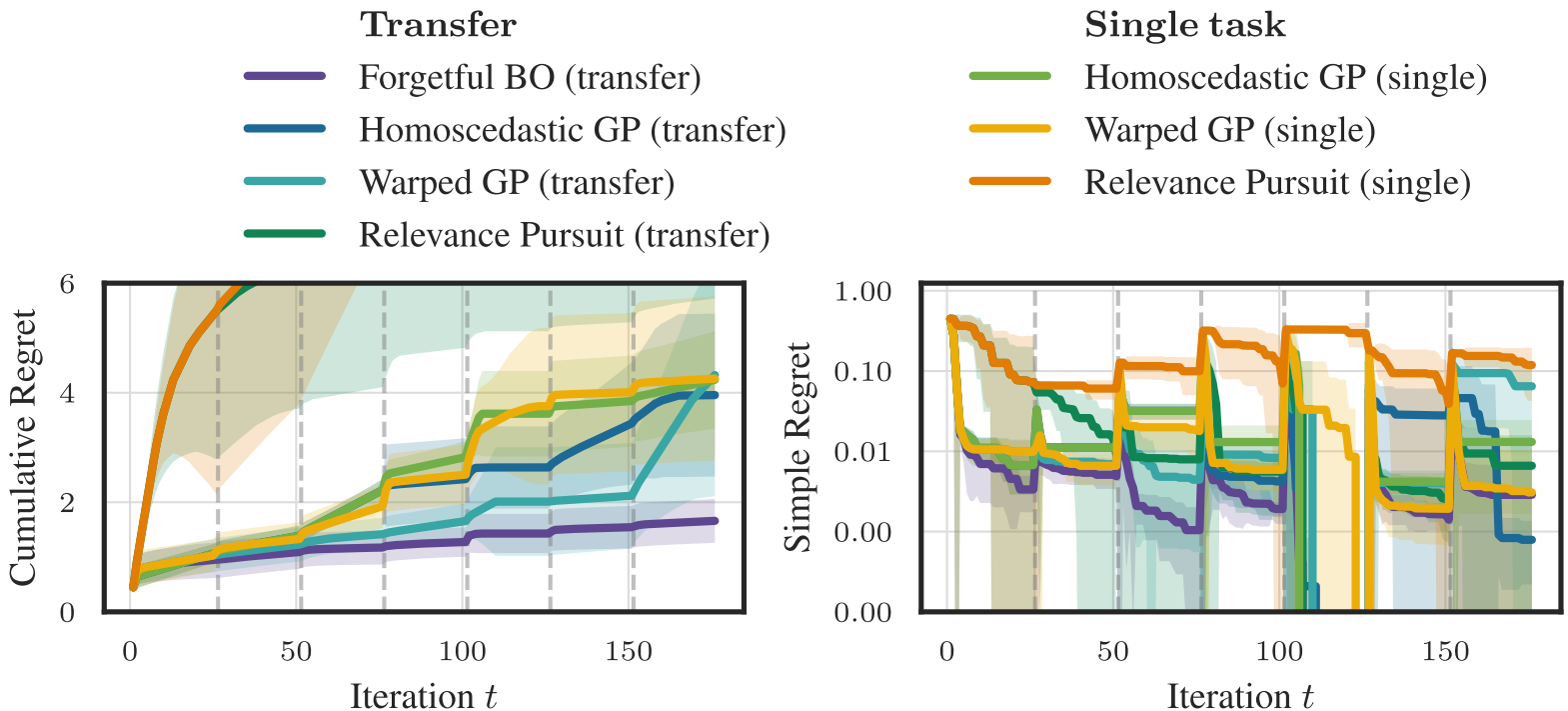
Forgetful BO – Acquisition Independent



Forgetful BO is acquisition independent and can be combined with different acquisition functions (e.g. Log Expected Improvement)

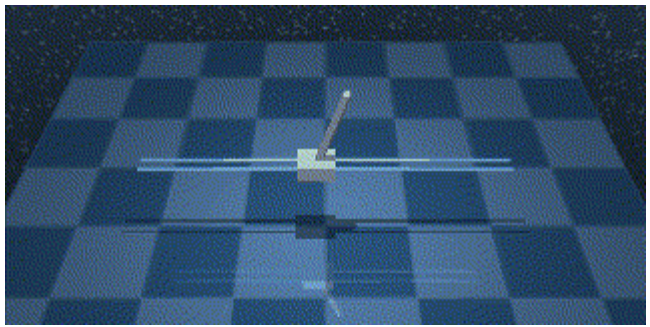
Forgetful BO – Transfer Learning

Continual transfer learning across different tasks. Leverage previous experience but forget detrimental points.



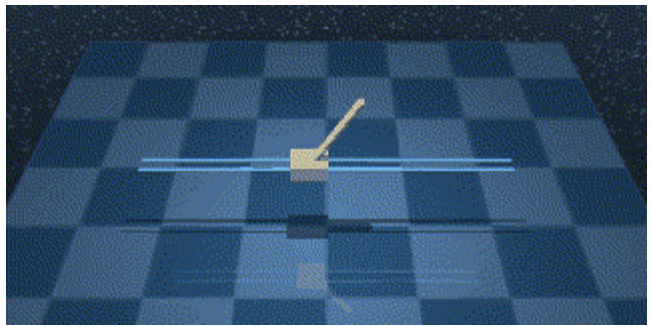
Pushing the Limits with Forgetting

LQR - Optimal Control



Same Task, but
Different Model?

Model-Based Reinforcement Learning

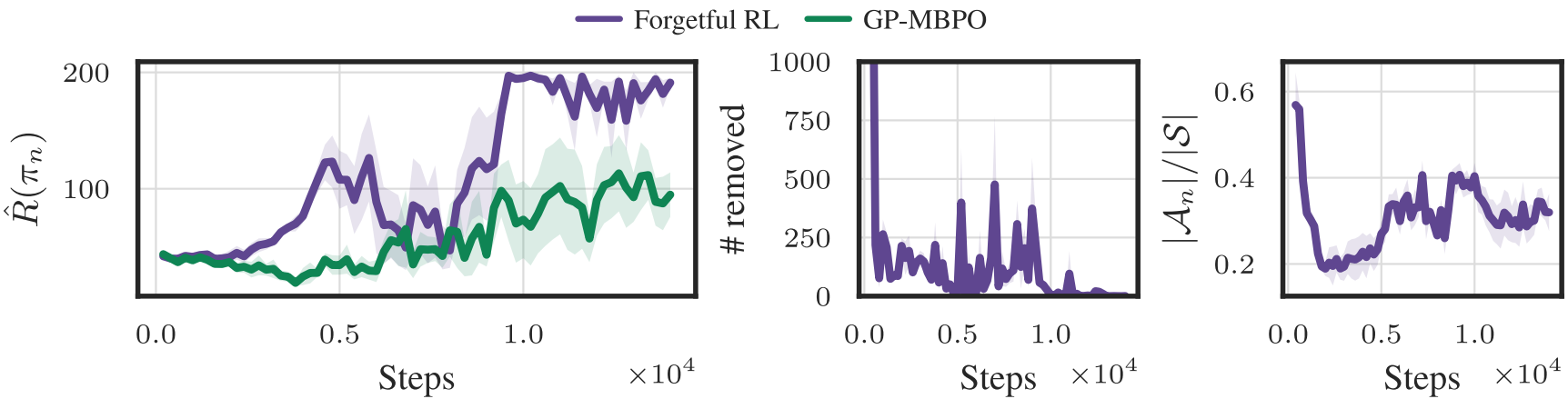


Linear Model solves task
$sin(\theta) \approx \theta$ for small θ.

MBRL: Learn from **trial and error**
Observe nonlinear experiences
⚠️ RL **requires** a **nonlinear** model.

💡 Can we remove **detrimental (nonlinear)** data and learn with **linear** model in **RL**?

Linear Model for MBRL



We converge despite initial failures, collecting detrimental (nonlinear) data.

💡 Delete detrimental data
Where do we linearize?

🔑 Model Target Region of potential improvement

Reinforcement Learning Objective

⚠️ What is the Target Region $\mathcal{A}$ for MBRL?

Reinforcement Learning Objective:

- MDP $\mathcal{M} = (\mathcal{X}, \mathcal{A}, P, r, \gamma)$: states $\mathcal{X}$, actions $\mathcal{A}$, transition dynamics P, rewards r, discounting γ.
- Goal: find policy π_n to maximize rewards

$$R(\pi_n) = \mathbb{E}_{\pi_n}\left[\sum_{t=0}^{\infty} \gamma^t r(x_t, a_t)\right] \tag{1}$$

- $Q^{\pi_n}(x, a) = \mathbb{E}_{\pi_n}\left[\sum_{t=0}^{\infty} \gamma^t r(x_t, a_t) | x_0 = x, a_0 = a\right]$ and $V^{\pi_n}(x) = Q^{\pi_n}(x, \pi_n(x))$.

Target Region for MBRL

What part of the state-action space do you want to model well?

💡 State-action with potential improvement:

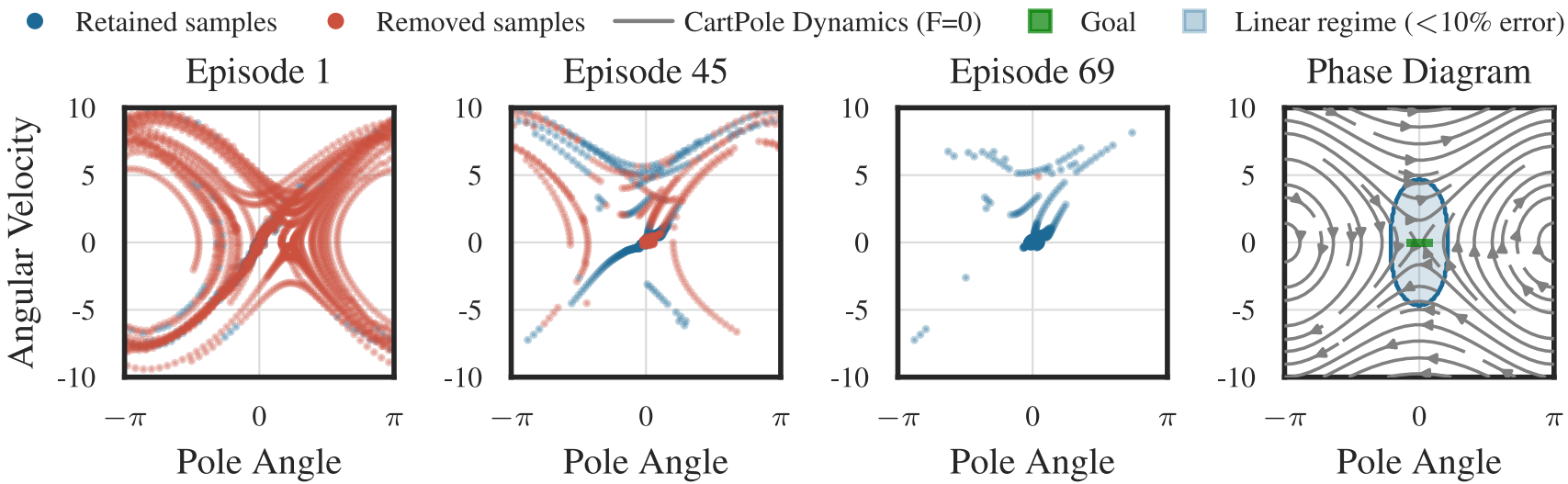
$$\mathcal{A}_n = \{(x_t, a_t) : \underbrace{\bar{Q}^{\pi_n}(x,a) \geq \underline{V}^{\pi_n}(x)}_{\substack{\text{actions with} \\ \text{positive advantage}}} \wedge \underbrace{\bar{Q}^{\pi_n}(x,a) \geq \min_{x_m \sim \rho_0} \underline{V}^{\pi_n}(x_m)}_{\text{states outside of } \rho^{\pi^*}}\},$$

Action Filtering: All potentially improving actions over the entire state space.

State Filtering: All states that are not visited by the current or better policies.

What Samples do we Remove in MBRL?

Removing **nonlinear** data, which is detrimental to our **linear** model.




Thank you!
Questions & Answers