Online out-of-distribution detection via simulation-informed deep Gaussian process state space models

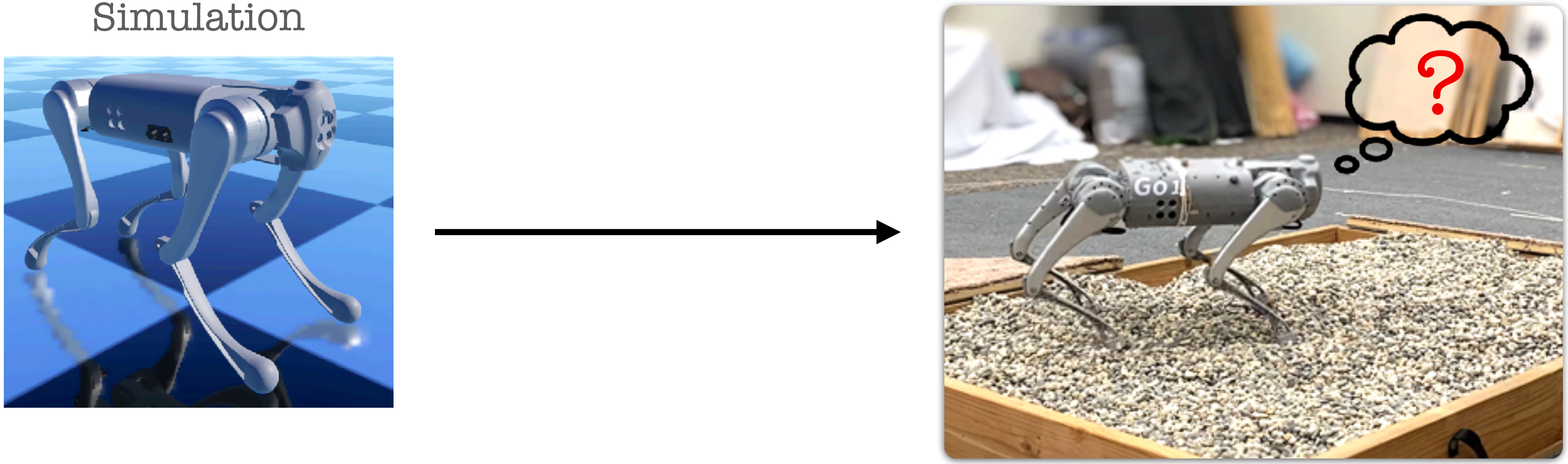
Alonso Marco and Claire Tomlin

June 20, 2023

Hybrid Systems Laboratory

Berkeley
UNIVERSITY OF CALIFORNIA

Motivation


Simulation

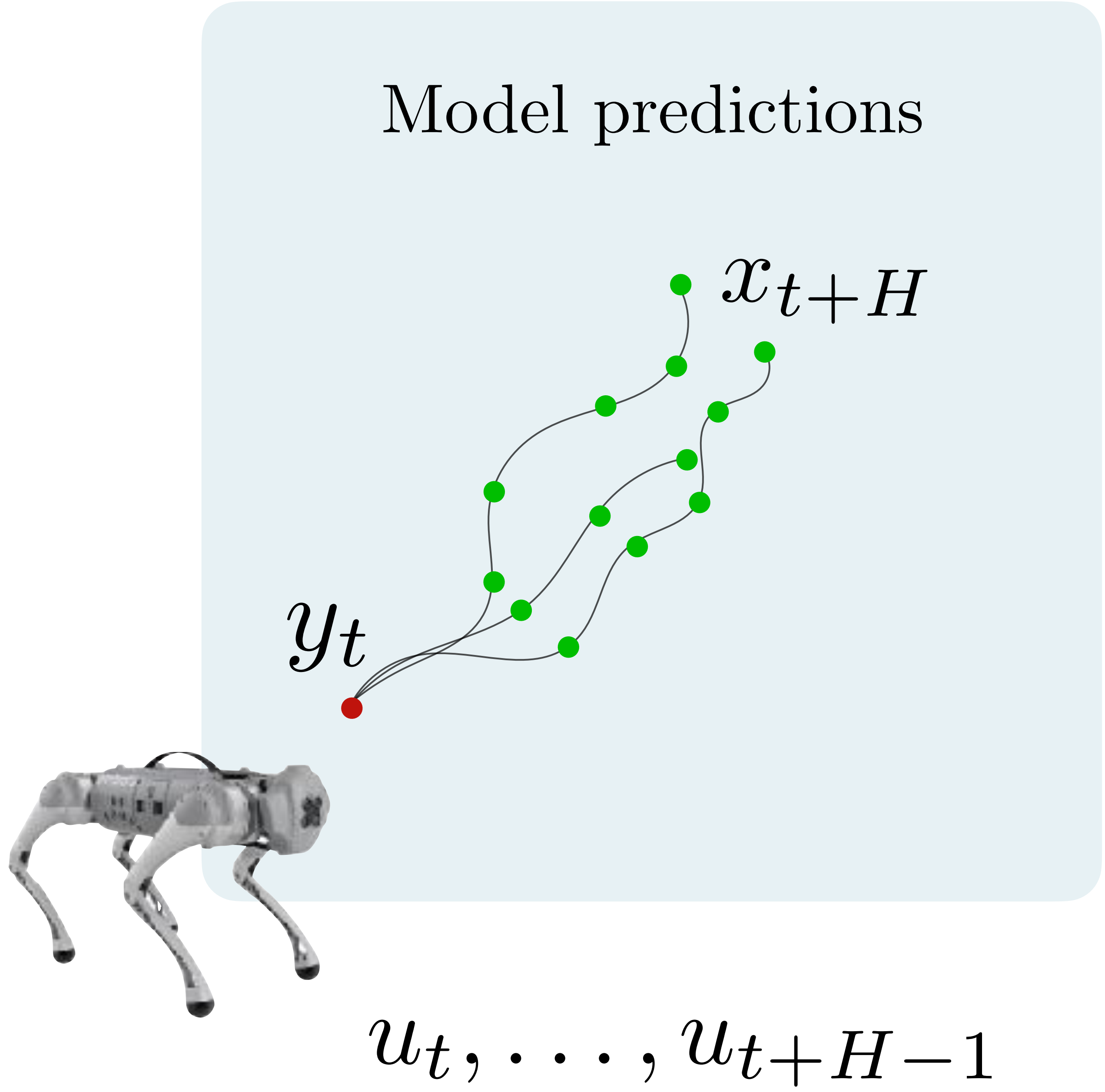
Real world

- Corner cases
- Unexpected events

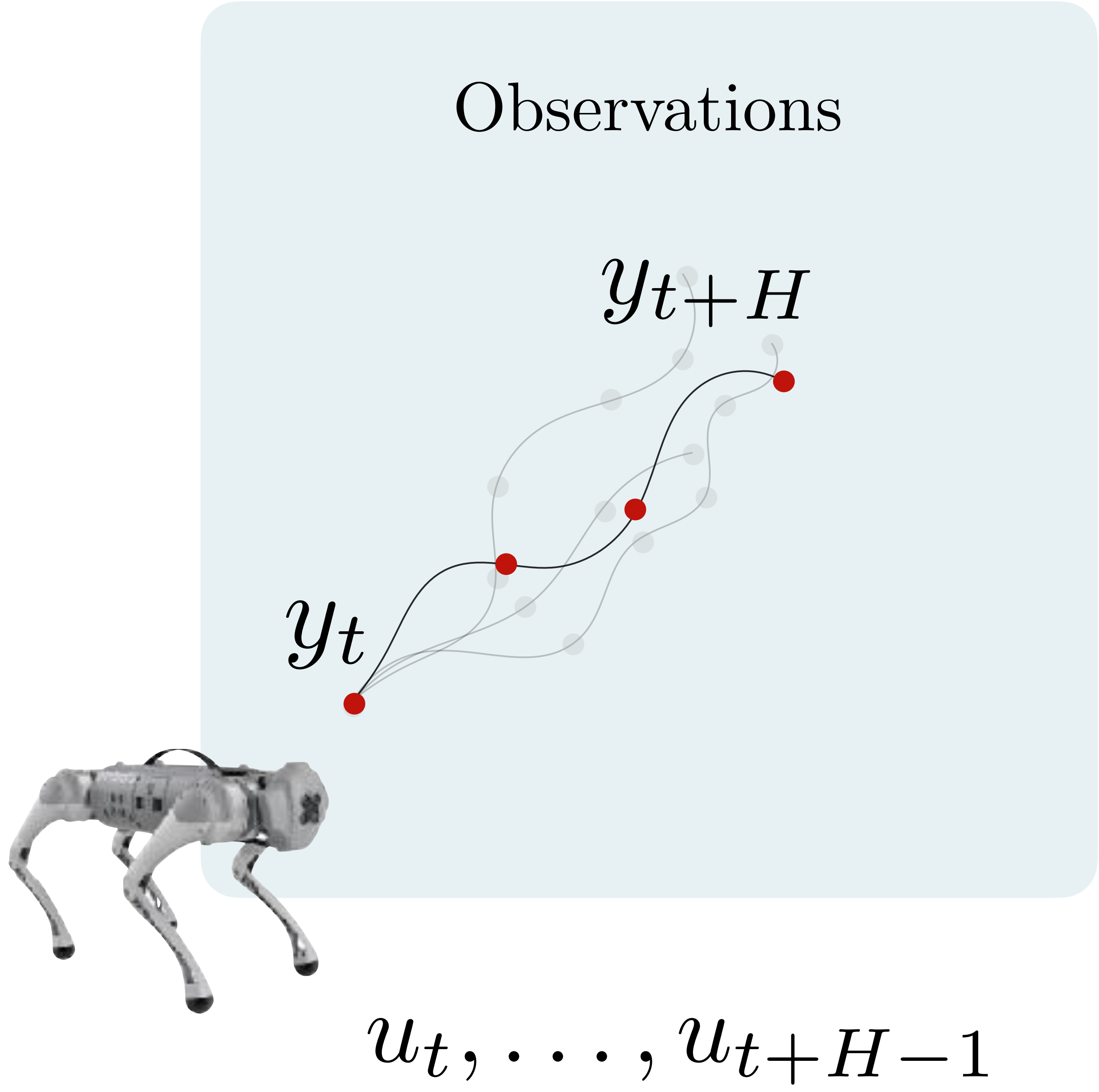
Goals

- Detect out-of-distribution (OoD) scenarios using probabilistic state predictions
- Well calibrated uncertainties via informed priors

Idea



Idea



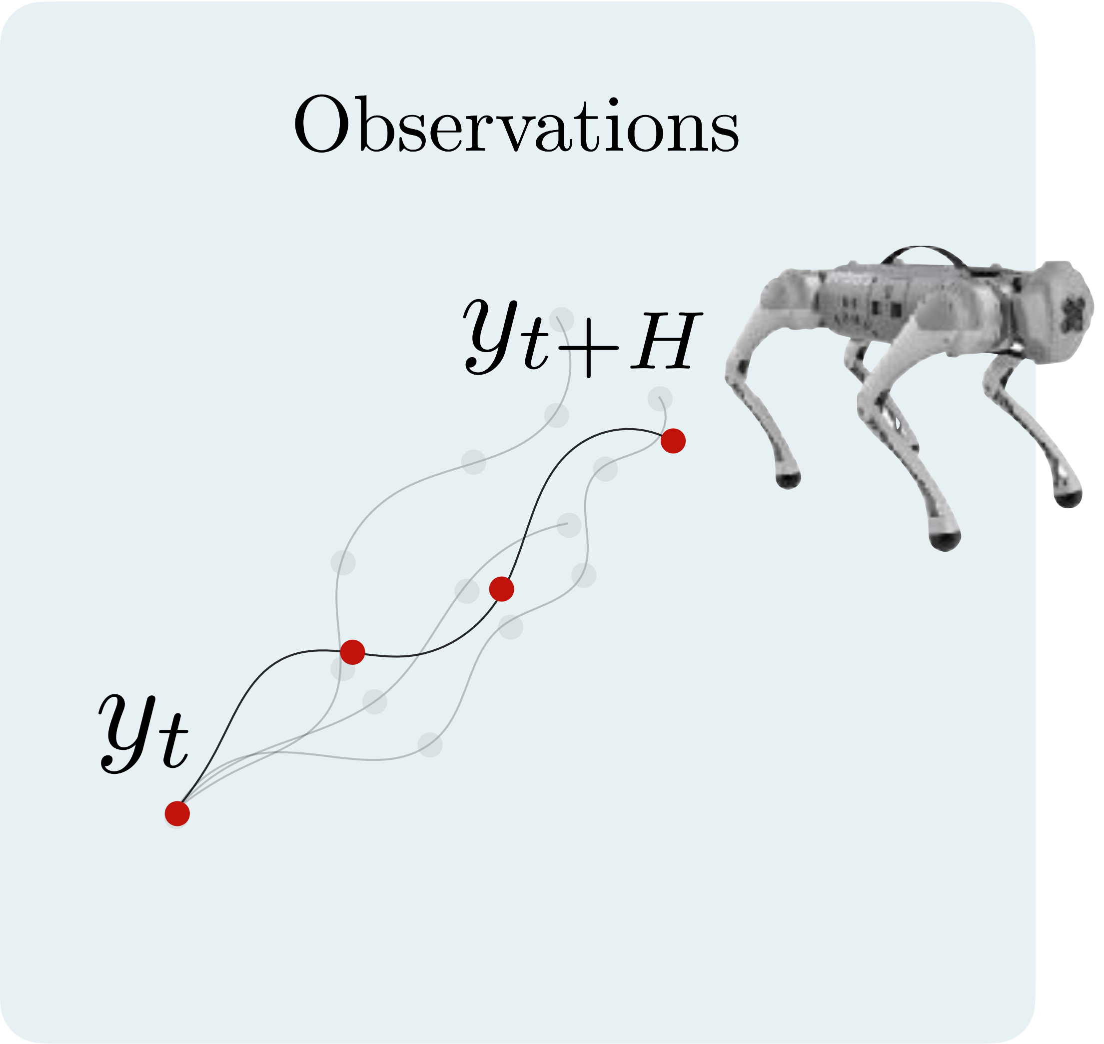
Observations

y_{t+H}

y_t

$u_t, \ldots, u_{t+H-1}$

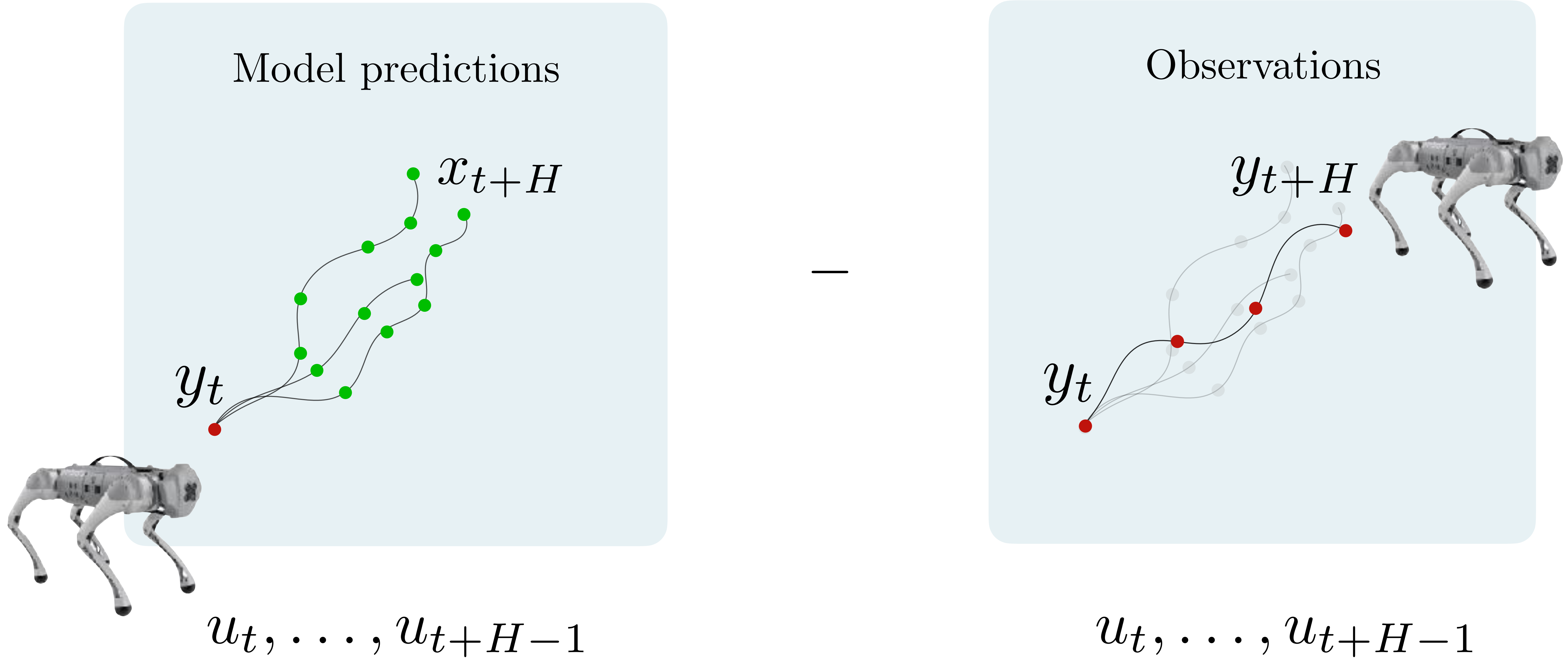
Idea



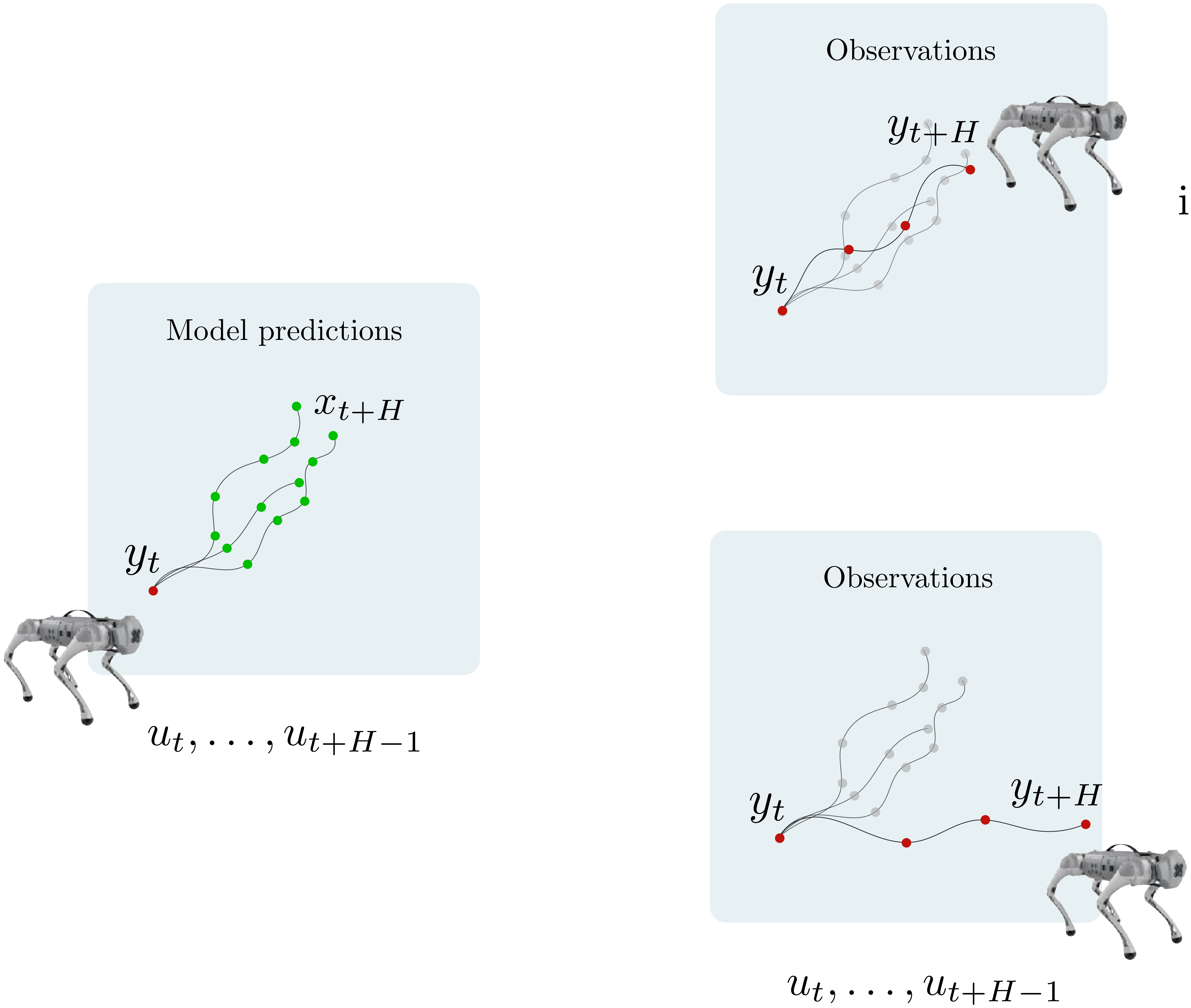
$$u_t, \ldots, u_{t+H-1}$$

Idea

$$\mathcal{L}_{\mathrm{OoD}}(\cdot) \quad =$$



Idea



Model predictions

x_{t+H}

y_t

$u_t, \ldots, u_{t+H-1}$

Observations

y_{t+H}

y_t

in distribution

$\downarrow \mathcal{L}_{\text{OoD}}(\cdot)$

Observations

y_t

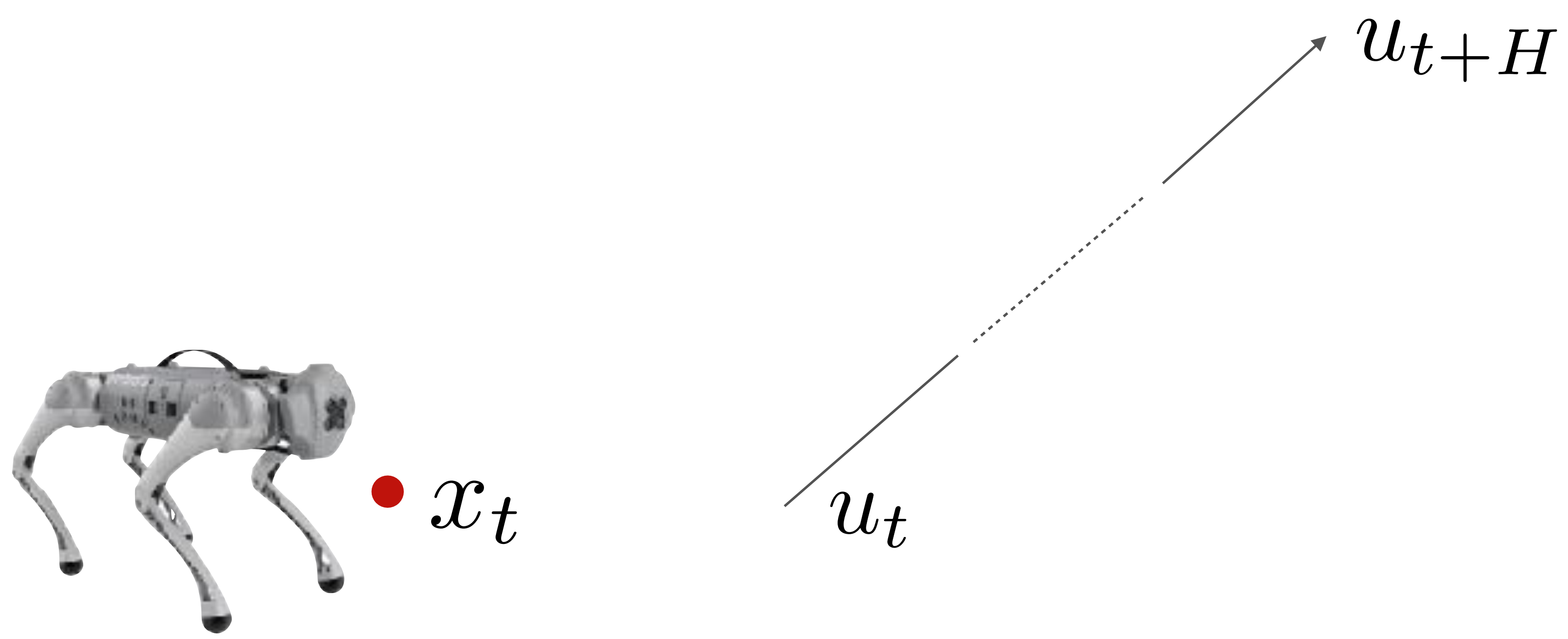
y_{t+H}

$u_t, \ldots, u_{t+H-1}$

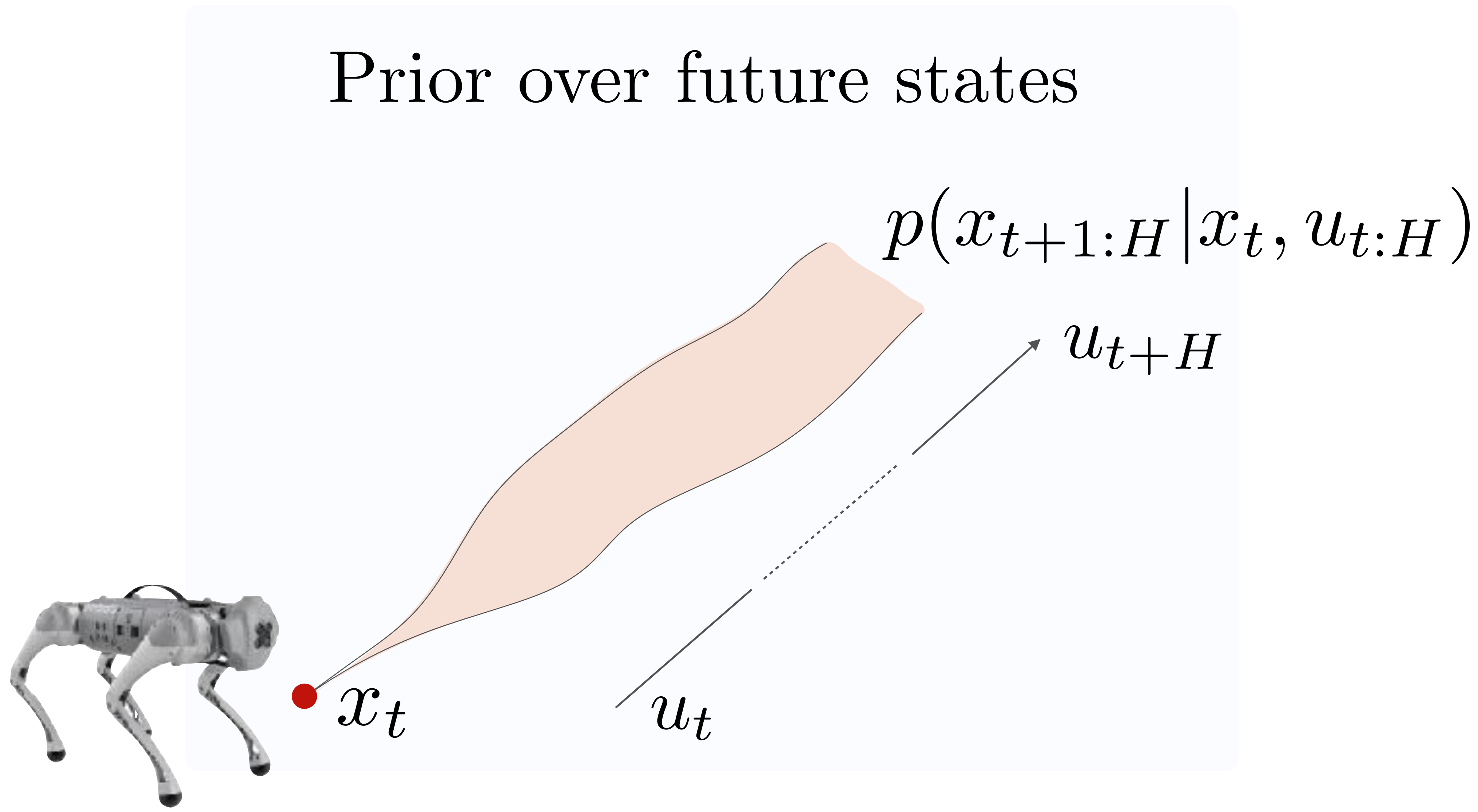
out of distribution

$\uparrow \mathcal{L}_{\text{OoD}}(\cdot)$

Probabilistic approach to OoD



Probabilistic approach to OoD

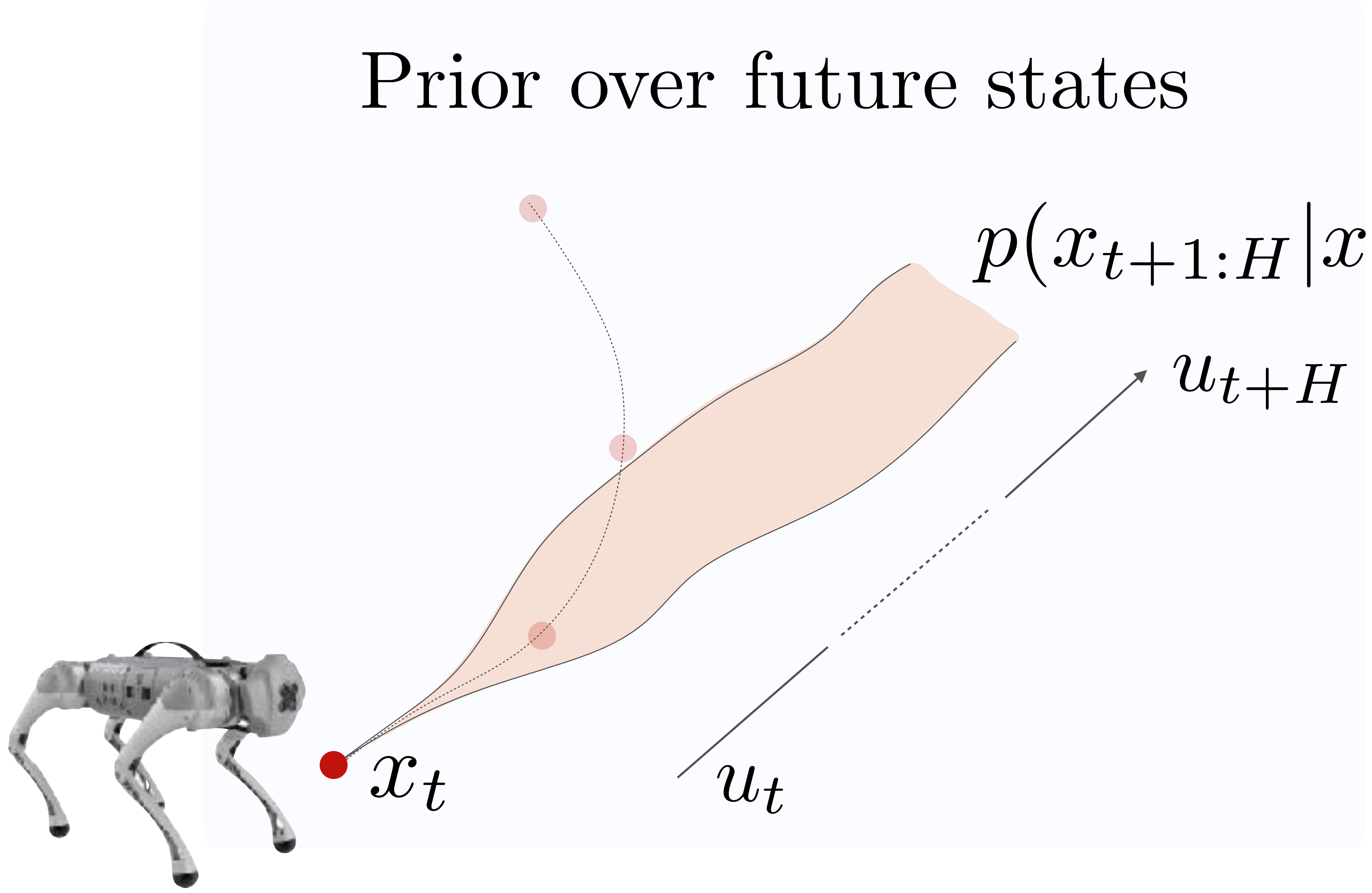


Prior over future states

$p(x_{t+1:H} | x_t, u_{t:H})$

- Aleatoric uncertainty
- Epistemic uncertainty

Probabilistic approach to OoD

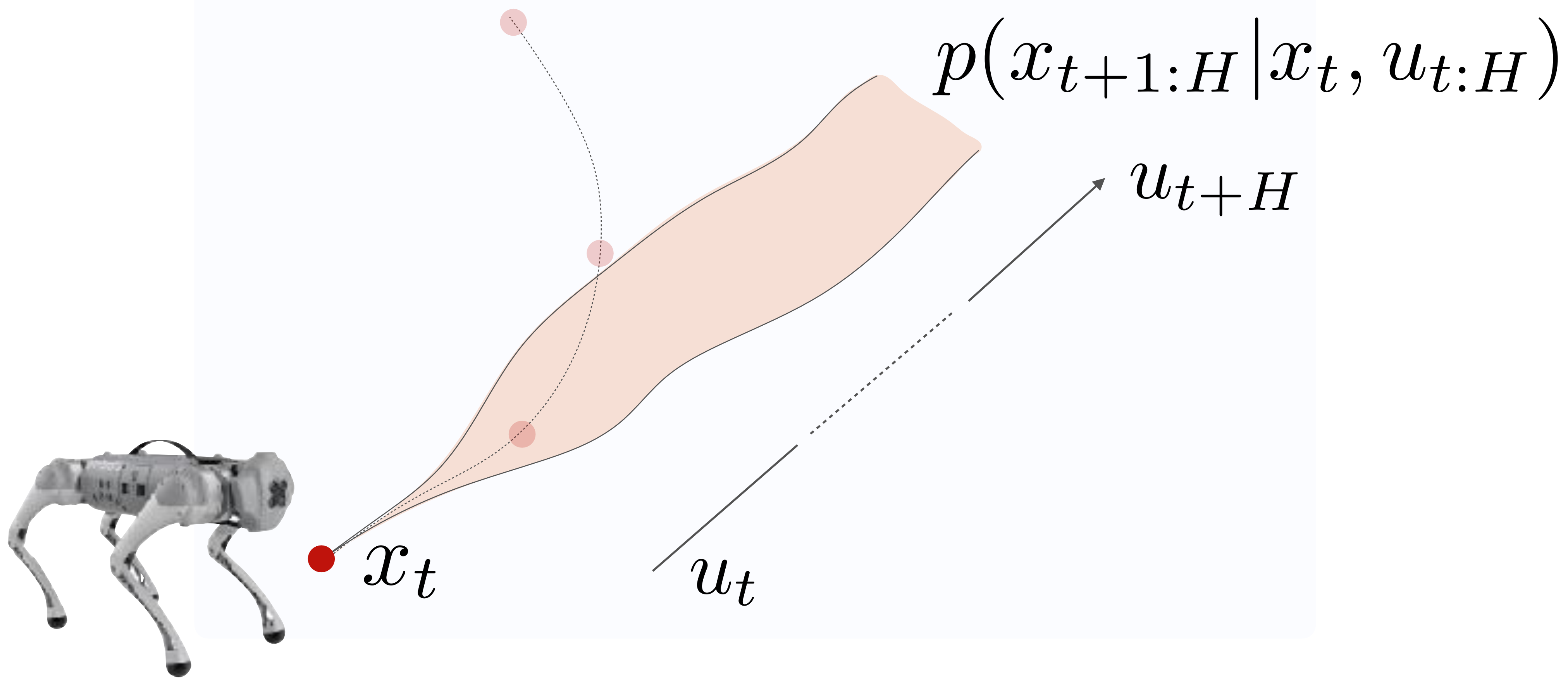


$$p(x_{t+1:H}|x_t, u_{t:H})$$

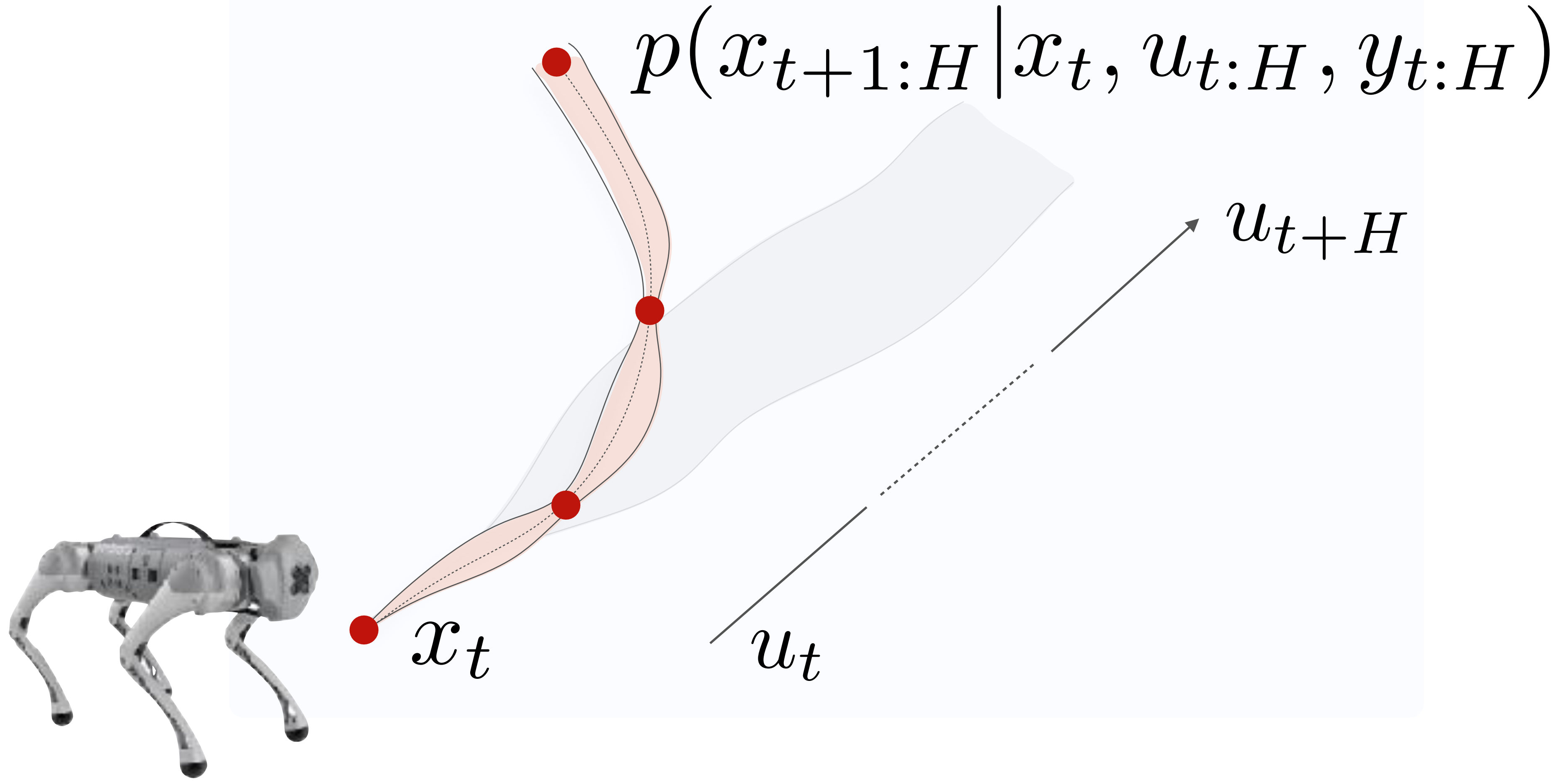
- Aleatoric uncertainty
- Epistemic uncertainty

Probabilistic approach to OoD

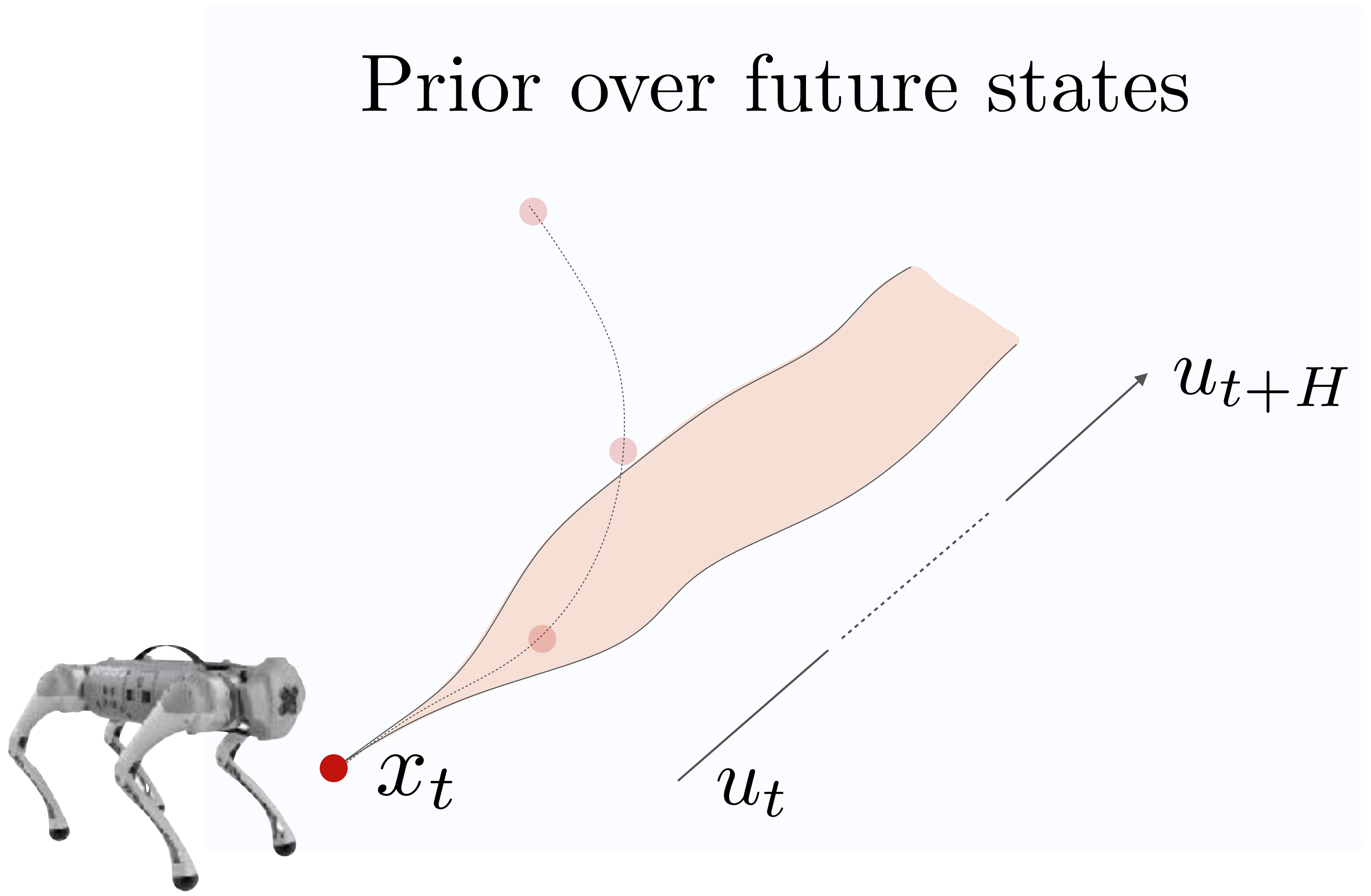
Prior over future states



Posterior over past states

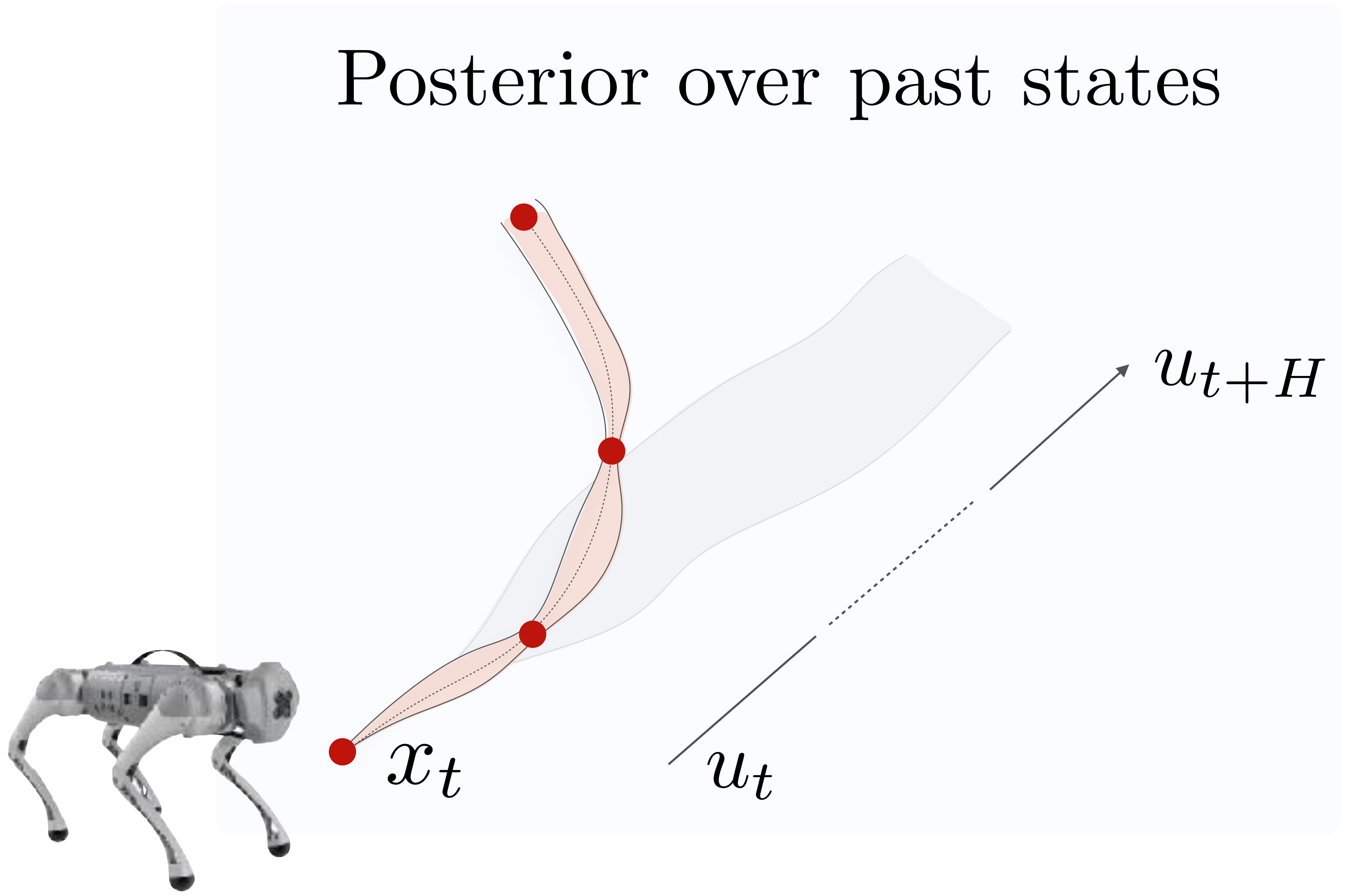


Probabilistic approach to OoD



Prior over future states

$$q(x_{t+1:H}|x_t, u_{t:H})$$

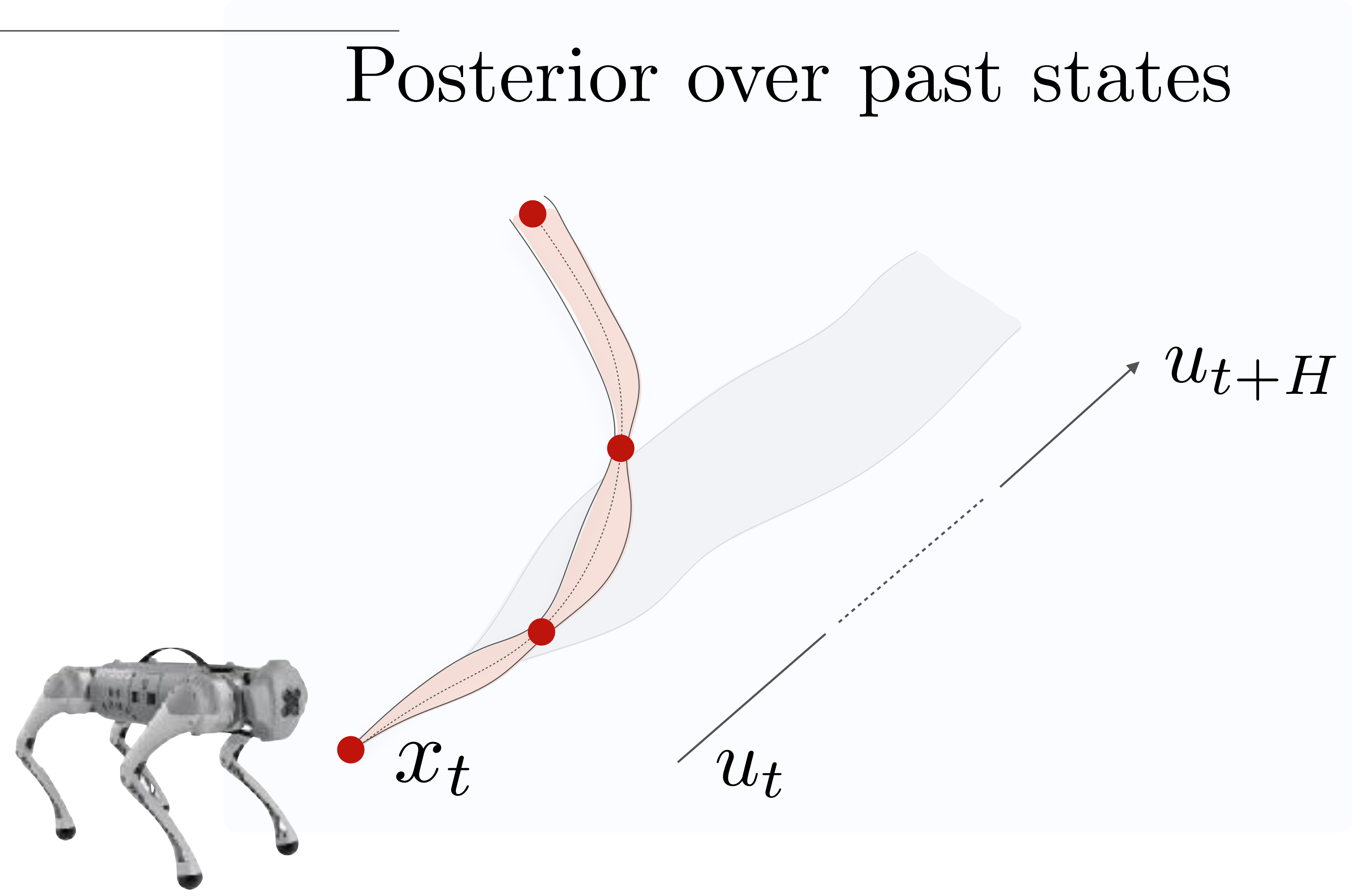


Posterior over past states

$$\underbrace{p(x_{t+1:H}|x_t, u_{t:H}, y_{t:H})}_{\text{intractable}}$$

$$\hat{\mathcal{L}}_{\text{OoD}} = D_{\text{KL}}(q||p)$$ ✗

Probabilistic approach to OoD



$$p(x_{t+1:H} | x_t, u_{t:H}, y_{t:H}) \propto \underline{p(y_{t:H} | x_{t:H}, u_{t:H})}\,\underline{q(x_{t+1:H} | x_t, u_{t:H})}$$

$$= \prod_{h=0}^{H} \underbrace{p(y_{t+h} | x_{t+h})}_{\text{likelihood}}$$

$$= \prod_{h=0}^{H-1} \underbrace{q(x_{t+h+1} | x_{t+h}, u_{t+h})}_{\text{dynamics model}}$$

Probabilistic approach to OoD

$$p(x_{t+1:H}|x_t, u_{t:H}, y_{t:H}) \propto \underline{p(y_{t:H}|x_{t:H}, u_{t:H})}\,\underline{q(x_{t+1:H}|x_t, u_{t:H})}$$

$$= \prod_{h=0}^{H} \underbrace{p(y_{t+h}|x_{t+h})}_{\text{likelihood}}$$

$$= \prod_{h=0}^{H-1} \underbrace{q(x_{t+h+1}|x_{t+h}, u_{t+h})}_{\text{dynamics model}}$$

$$\min_q \underbrace{\mathcal{L}_{\mathrm{OoD}}(q)} \iff \min_q D_{\mathrm{KL}}(q||p)$$

Negative evidence lower bound (ELBO)

$$\mathcal{L}_{\mathrm{OoD}} = -\mathbb{E}_q[\log p(y_{t:H}|x_{t:H})] - \mathbb{E}_q[\log p(x_{t:H})] - H(q)$$ ✓

Probabilistic approach to OoD

$$p(x_{t+1:H}|x_t, u_{t:H}, y_{t:H}) \propto \underline{p(y_{t:H}|x_{t:H}, u_{t:H})}\,\underline{q(x_{t+1:H}|x_t, u_{t:H})}$$

$$= \prod_{h=0}^{H} \underbrace{p(y_{t+h}|x_{t+h})}_{\text{likelihood}}$$

$$= \prod_{h=0}^{H-1} \underbrace{q(x_{t+h+1}|x_{t+h}, u_{t+h})}_{\text{dynamics model}}$$

- Model choices: $p(y_t|x_t) = \mathcal{N}(x_t, R)$

$$q(x_{t+1}|x_t, u_t) = \mathcal{N}(f(x_t, u_t), Q)$$

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

Fast Online OoD

- Probabilistic dynamics model

epistemic aleatoric

$$q(x_{t+1}|x_t, u_t) = \mathcal{N}(f(x_t, u_t), Q)$$

$$f(x_t, u_t) = \begin{bmatrix} \beta_1^\top \Phi_1(x_t, u_t) \\ \vdots \\ \beta_D^\top \Phi_D(x_t, u_t) \end{bmatrix}$$

Prior $\beta_d \sim p(\beta_d)$ → **Training** $\mathcal{D} = \{\tilde{x}_t, \tilde{u}_t\}_{t=0}^T$ → **Posterior** $\beta_d \sim p(\beta_d|\mathcal{D}) = \mathcal{N}(\mu_d, \Sigma_d)$

$$p(\beta_d|\mathcal{D}) \propto p(\mathcal{D}|\beta_d)p(\beta_d)$$

- Sampling rollouts is cheap

① Collect R samples from posterior $\hat{\beta}_d^{(r)} = \mu_d + \Sigma_d^{1/2}\xi^{(r)}$ $\begin{matrix} r = 1, \dots, R \\ d = 1, \dots, D \end{matrix}$

② Construct callable transition function

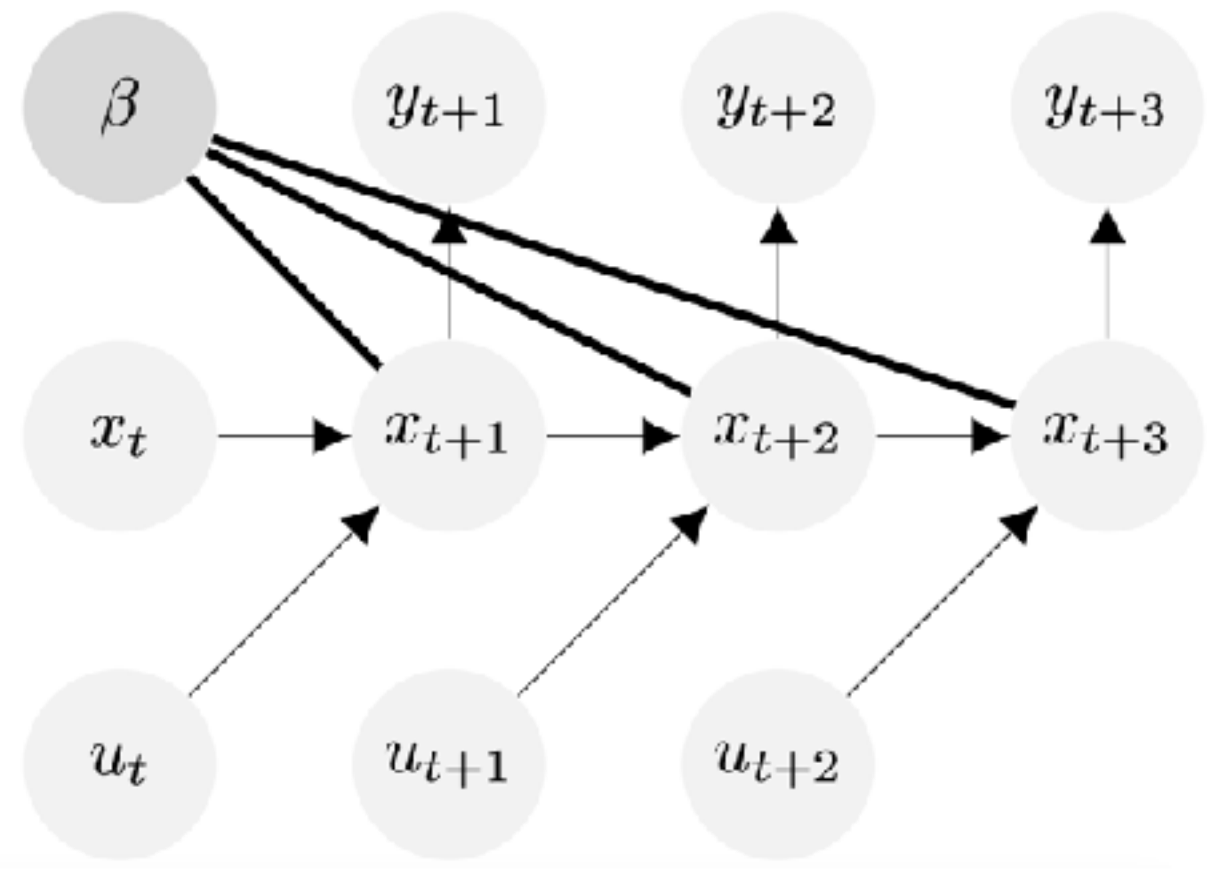
$$f^{(r)}(x_t, u_t) = \begin{bmatrix} (\hat{\beta}_1^{(r)})^\top \Phi_1(x_t, u_t) \\ \vdots \\ (\hat{\beta}_D^{(r)})^\top \Phi_D(x_t, u_t) \end{bmatrix}$$

③ Construct callable state transitions

$$x_{t+1}^{(r)}(x_t, u_t) = f^{(r)}(x_t, u_t) + L_Q\xi\,, \quad Q = L_Q L_Q^\top,\ \xi \sim \mathcal{N}(0, I)$$

④ Sample rollouts at cost $O(HD)$

$$\{\hat{x}_{t+1}^r, \dots, \hat{x}_{t+H}^r\} \sim q(x_{t+1:H}|x_t, u_{t:H-1})$$



Fast Online OoD

④ Sample rollouts at cost $O(HD)$

$$\{\hat{x}^r_{t+1}, \dots, \hat{x}^r_{t+H}\} \sim q(x_{t+1:H} | x_t, u_{t:H-1})$$

⑤ Monte Carlo approximation

$$\mathcal{L}_{\text{OoD}} = -\mathbb{E}_q[\log p(y_{t:H}|x_{t:H})] - \cancelto{\text{const}}{\mathbb{E}_q[\log p(x_{t:H})]} - H(q)$$

$$\mathbb{E}_q[\log p(y_{t:H}|x_{t:H})] \approx \frac{1}{R}\sum_{r=1}^{R}\sum_{h=1}^{H} \log \mathcal{N}(y_{t+h}|\hat{x}^r_{t+h}, R)$$

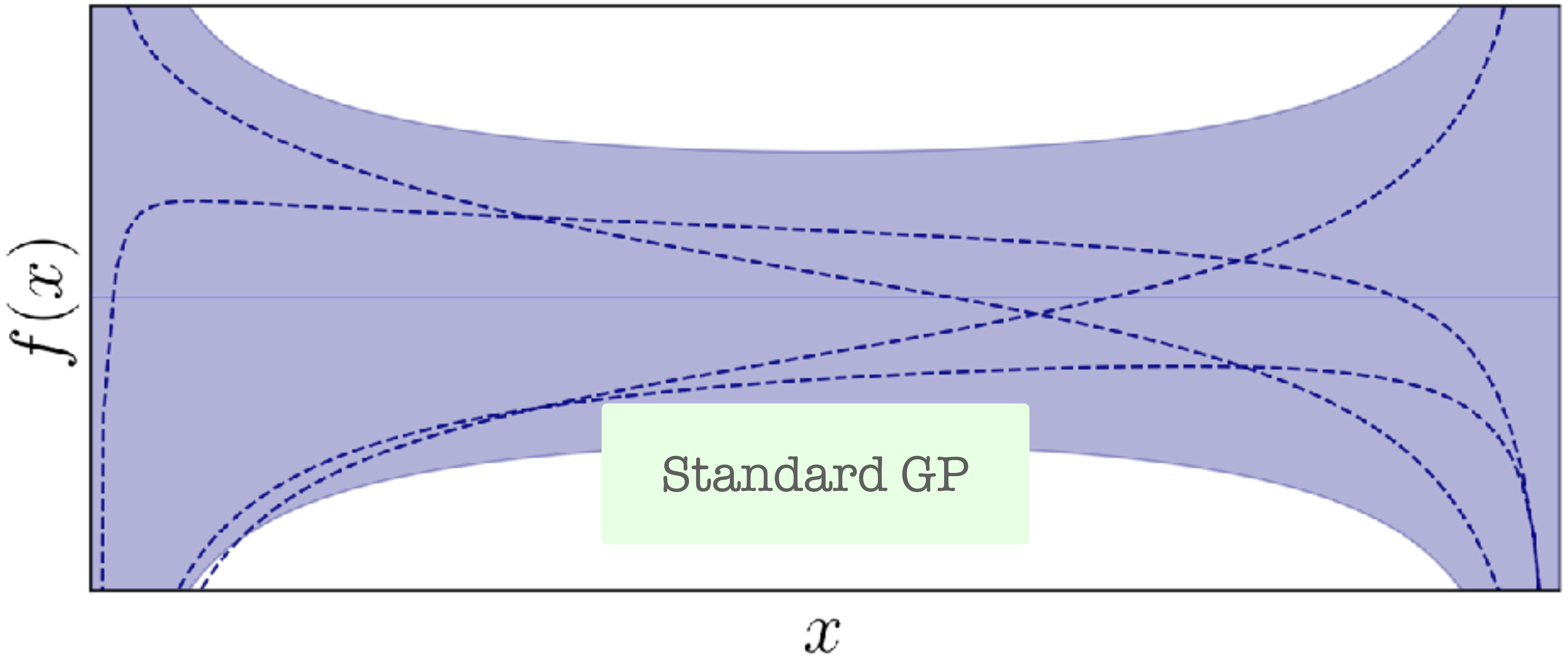
$$p(x_{t:H}) = \mathrm{U}(x_{t:H})$$

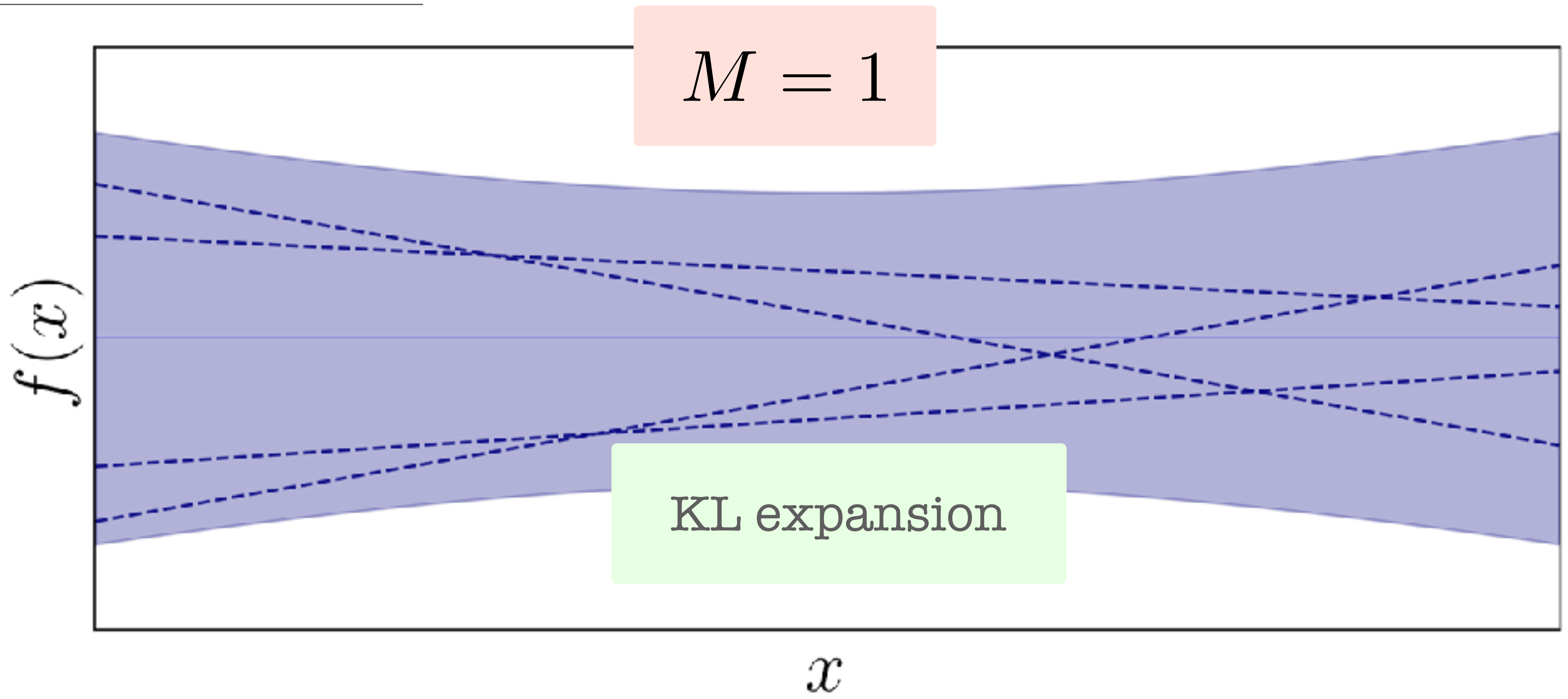
$$H(q) \propto \frac{1}{R}\sum_{d=1}^{D}\sum_{r=1}^{R}\sum_{h=1}^{H} \log\left(\Phi_d^\top(\hat{x}^{(r)}_{t+h}, \hat{u}_{t+h})\Sigma_d\Phi_d(\hat{x}^{(r)}_{t+h}, \hat{u}_{t+h})\right)$$

Overall cost $O(HRD)$

c++ implementation - Eigen (x10 faster)

Efficient long-term predictions: Karhunen-Loève expansion of GPs





A $$k(x,\bar{x}) = \frac{1}{1 - bx\bar{x}} \qquad \begin{array}{c} 0 < b < 1 \\ -1 < x, \bar{x} < 1 \end{array}$$

↓

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

B $$k(x,\bar{x}) = \sum_{j=1}^{M} \nu_j \phi_j(x) \phi_j(\bar{x})$$

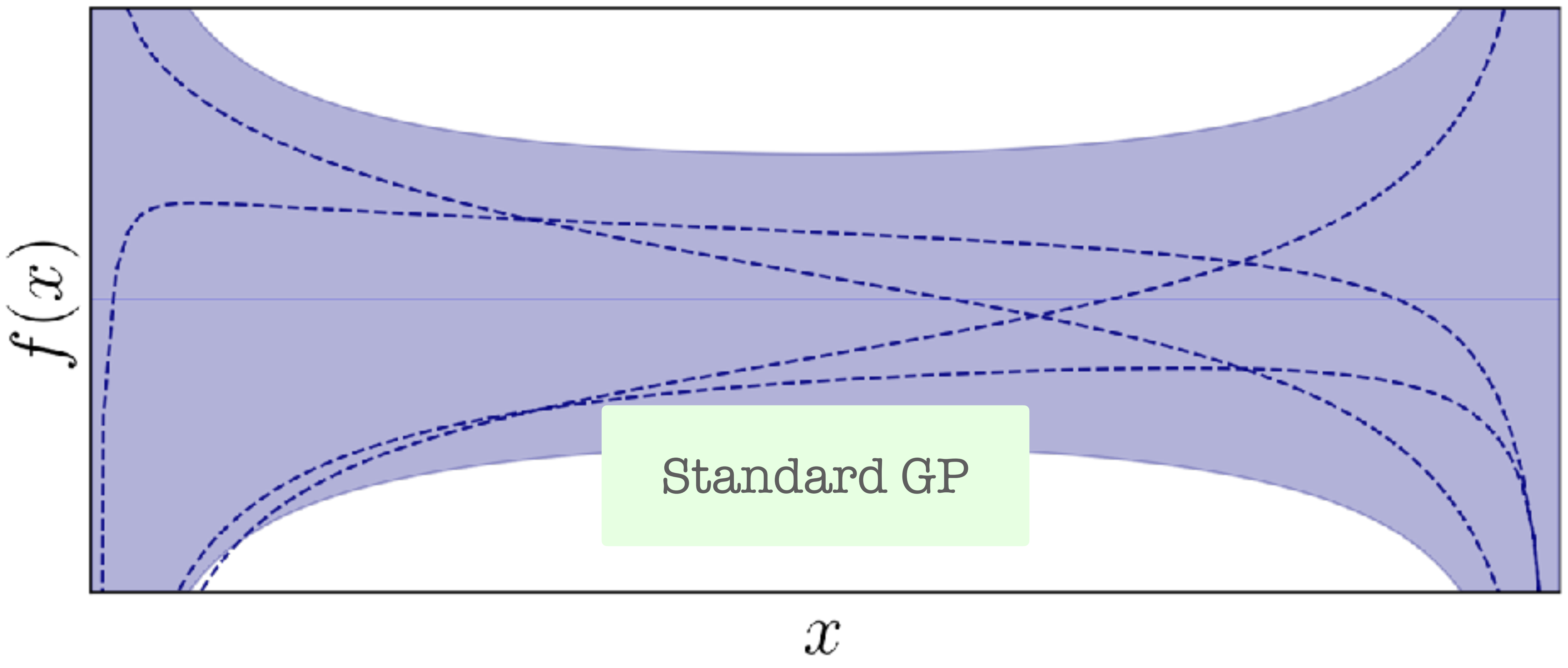
↑

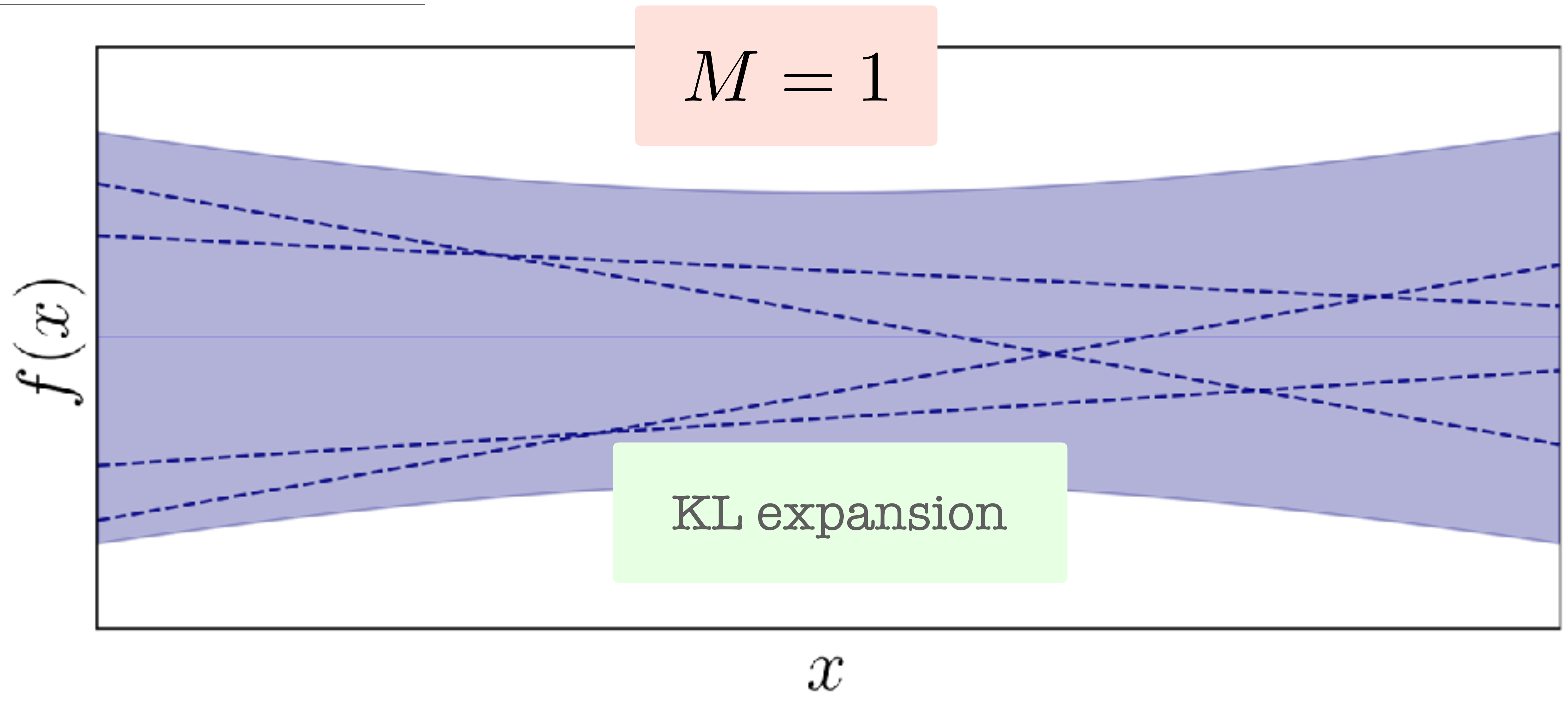
A $$f(x) \sim \sum_{j=1}^{M} \beta_j \phi_j(x), \quad \beta_j \sim \mathcal{N}(0, \nu_j)$$

$\phi_j(x) = x^j$

$\nu_j = b^j$

Efficient long-term predictions: Karhunen-Loève expansion of GPs





A $$k(x,\bar{x}) = \frac{1}{1 - bx\bar{x}} \qquad \begin{array}{c} 0 < b < 1 \\ -1 < x, \bar{x} < 1 \end{array}$$

↓

B $$f_\infty(x) \sim \mathcal{GP}(0, k(x,\bar{x}))$$

B $$k(x,\bar{x}) = \sum_{j=1}^{M} b^j x^j \bar{x}^j$$

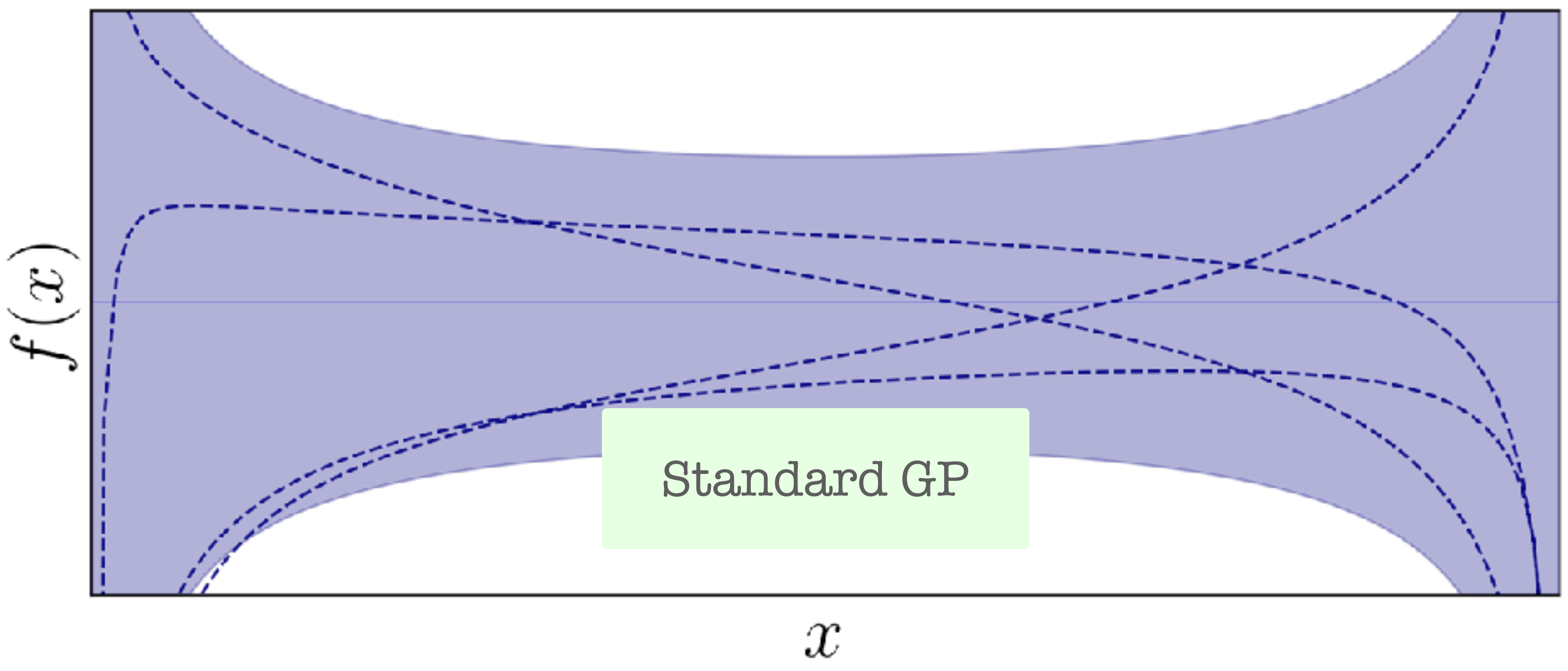
↑

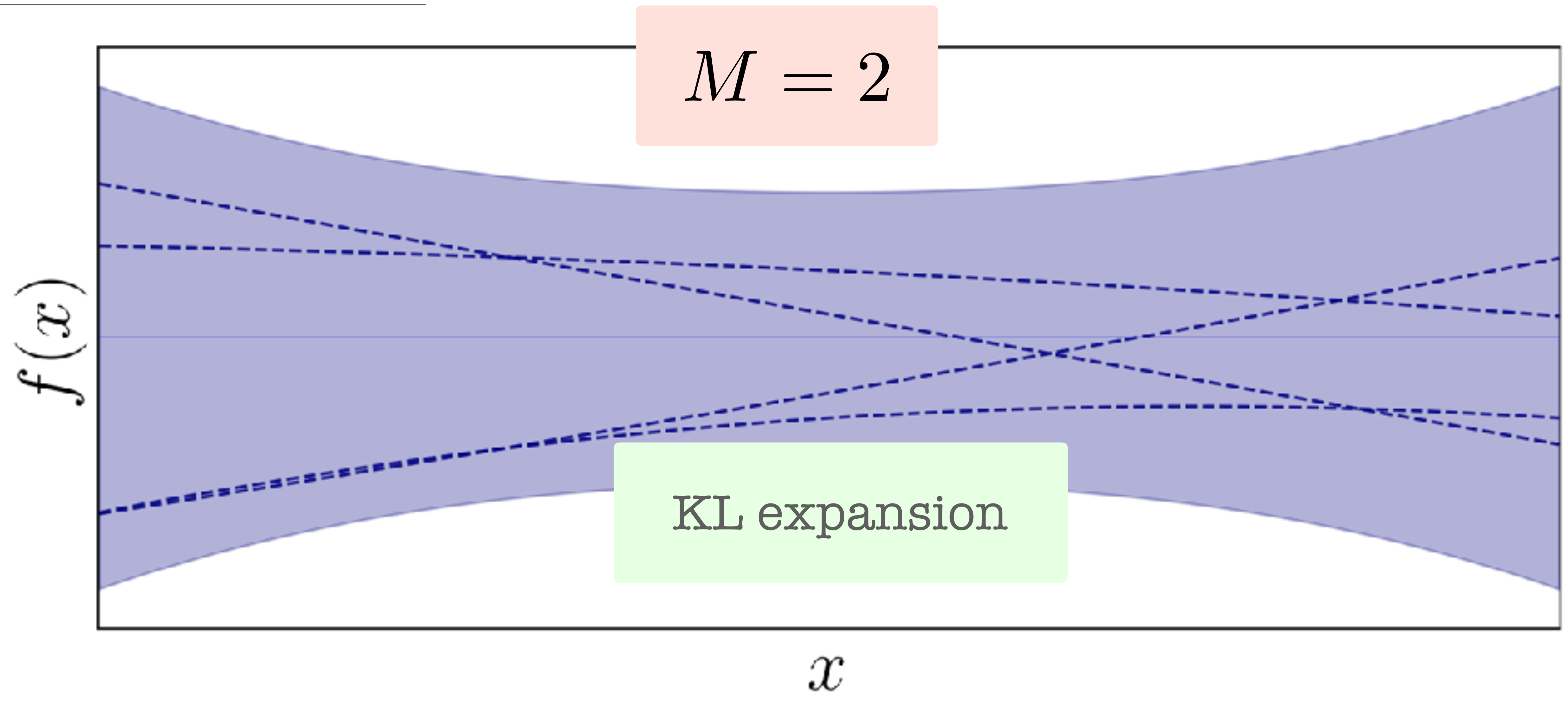
A $$f_M(x) \sim \sum_{j=1}^{M} \beta_j x^j, \quad \beta_j \sim \mathcal{N}(0, b^j)$$

Convergence in mean-square sense

$$\lim_{M\to\infty} \mathbb{E}\left[\left(f_\infty(x) - \sum_{j=1}^{M} \beta_j \phi_j(x)\right)^2\right] = 0$$

Efficient long-term predictions: Karhunen-Loève expansion of GPs





A $k(x, \bar{x}) = \dfrac{1}{1 - bx\bar{x}}$ $\quad 0 < b < 1 \quad -1 < x, \bar{x} < 1$

B $f_\infty(x) \sim \mathcal{GP}(0, k(x, \bar{x}))$

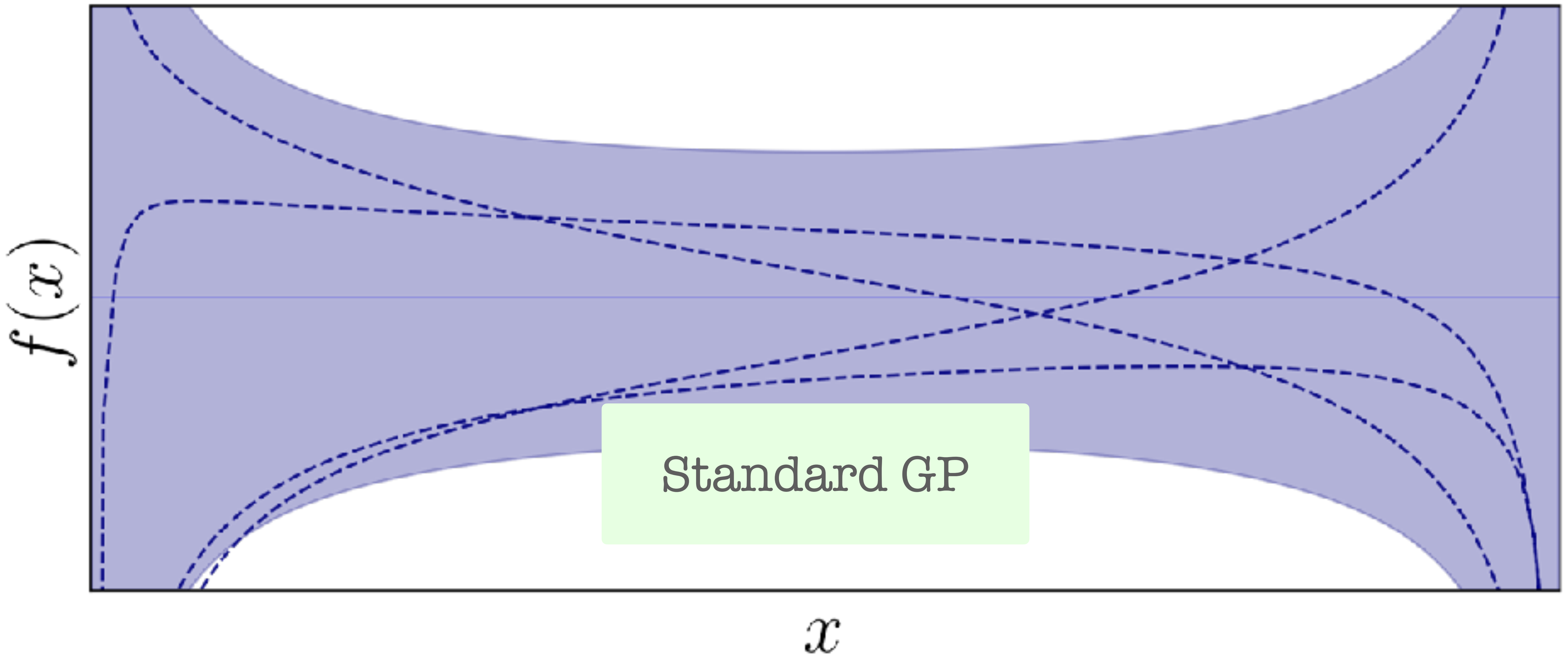
B $k(x, \bar{x}) = \sum_{j=1}^{M} b^j x^j \bar{x}^j$

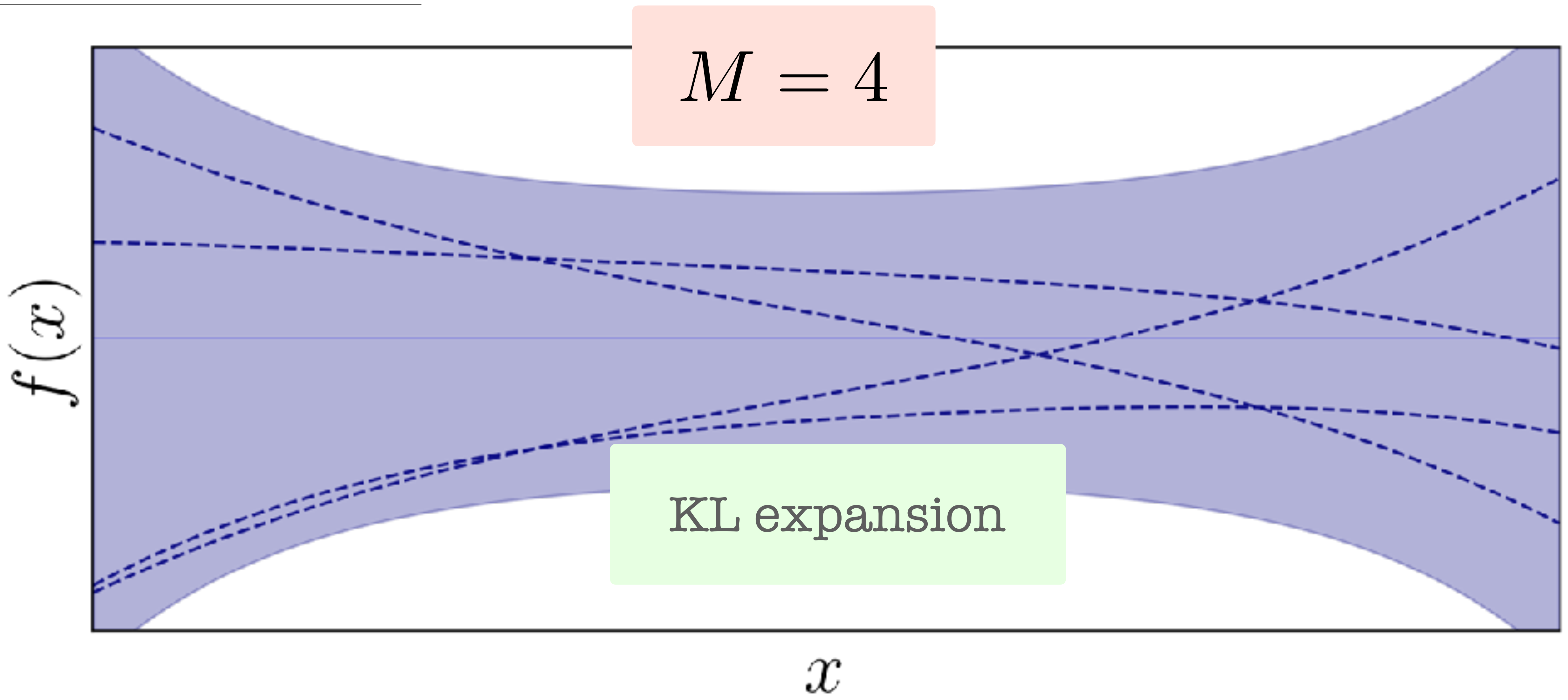
A $f_M(x) \sim \sum_{j=1}^{M} \beta_j x^j, \quad \beta_j \sim \mathcal{N}(0, b^j)$

Convergence in mean-square sense

$$\lim_{M \to \infty} \mathbb{E}\left[\left(f_\infty(x) - \sum_{j=1}^{M} \beta_j \phi_j(x)\right)^2\right] = 0$$

Efficient long-term predictions: Karhunen-Loève expansion of GPs





A $$k(x, \bar{x}) = \frac{1}{1 - bx\bar{x}} \qquad \begin{array}{c} 0 < b < 1 \\ -1 < x, \bar{x} < 1 \end{array}$$

B $$f_\infty(x) \sim \mathcal{GP}(0, k(x, \bar{x}))$$

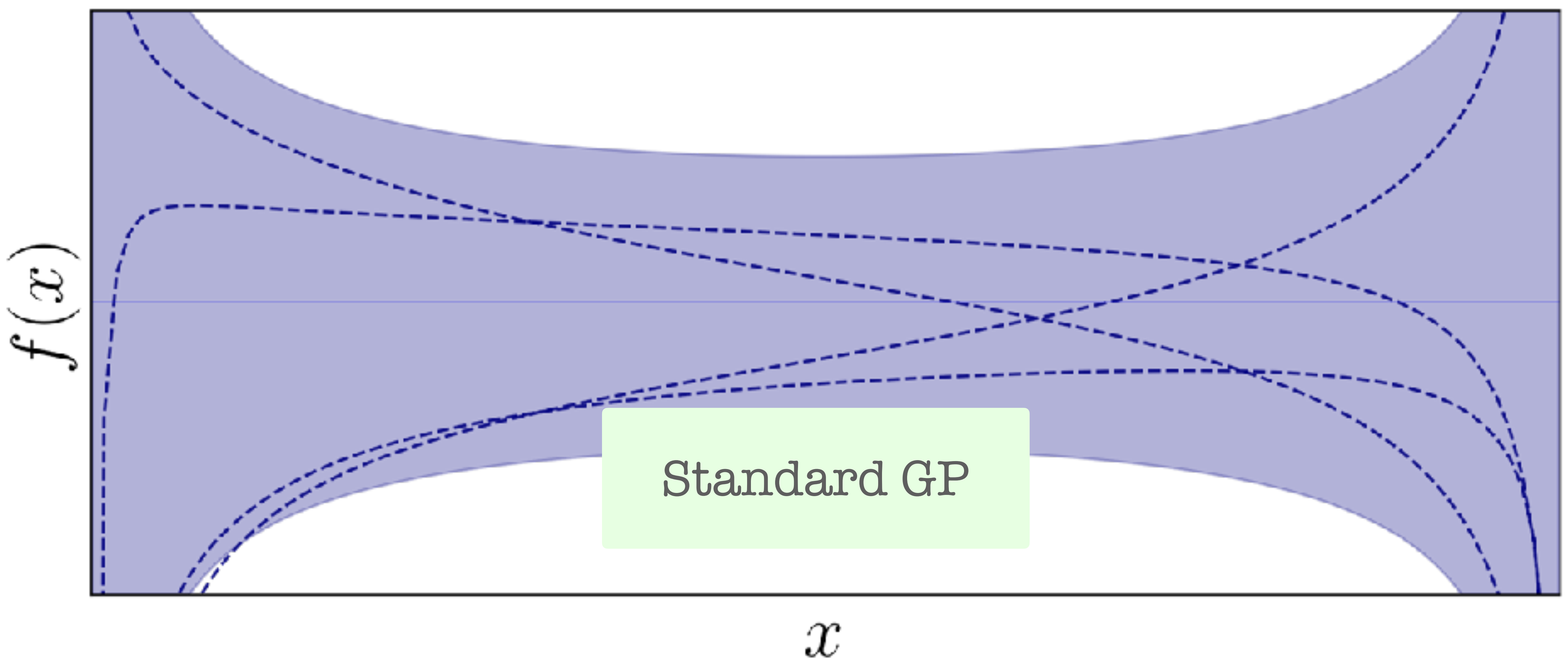
B $$k(x, \bar{x}) = \sum_{j=1}^{M} b^j x^j \bar{x}^j$$

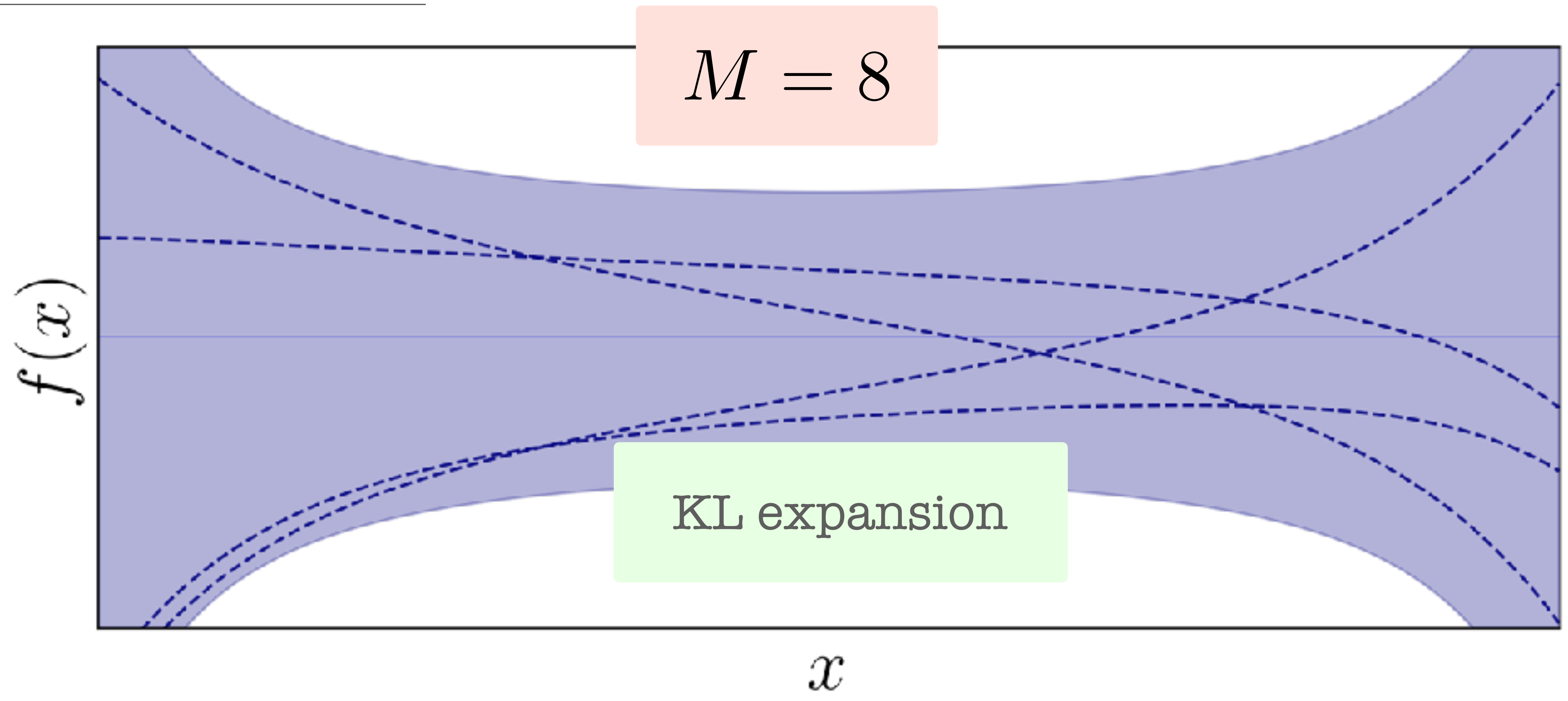
A $$f_M(x) \sim \sum_{j=1}^{M} \beta_j x^j, \quad \beta_j \sim \mathcal{N}(0, b^j)$$

Convergence in mean-square sense

$$\lim_{M \to \infty} \mathbb{E}\left[\left(f_\infty(x) - \sum_{j=1}^{M} \beta_j \phi_j(x)\right)^2\right] = 0$$

Efficient long-term predictions: Karhunen-Loève expansion of GPs





A $$k(x,\bar{x}) = \frac{1}{1 - bx\bar{x}} \qquad \begin{array}{c} 0 < b < 1 \\ -1 < x, \bar{x} < 1 \end{array}$$

B $$f_\infty(x) \sim \mathcal{GP}(0, k(x,\bar{x}))$$

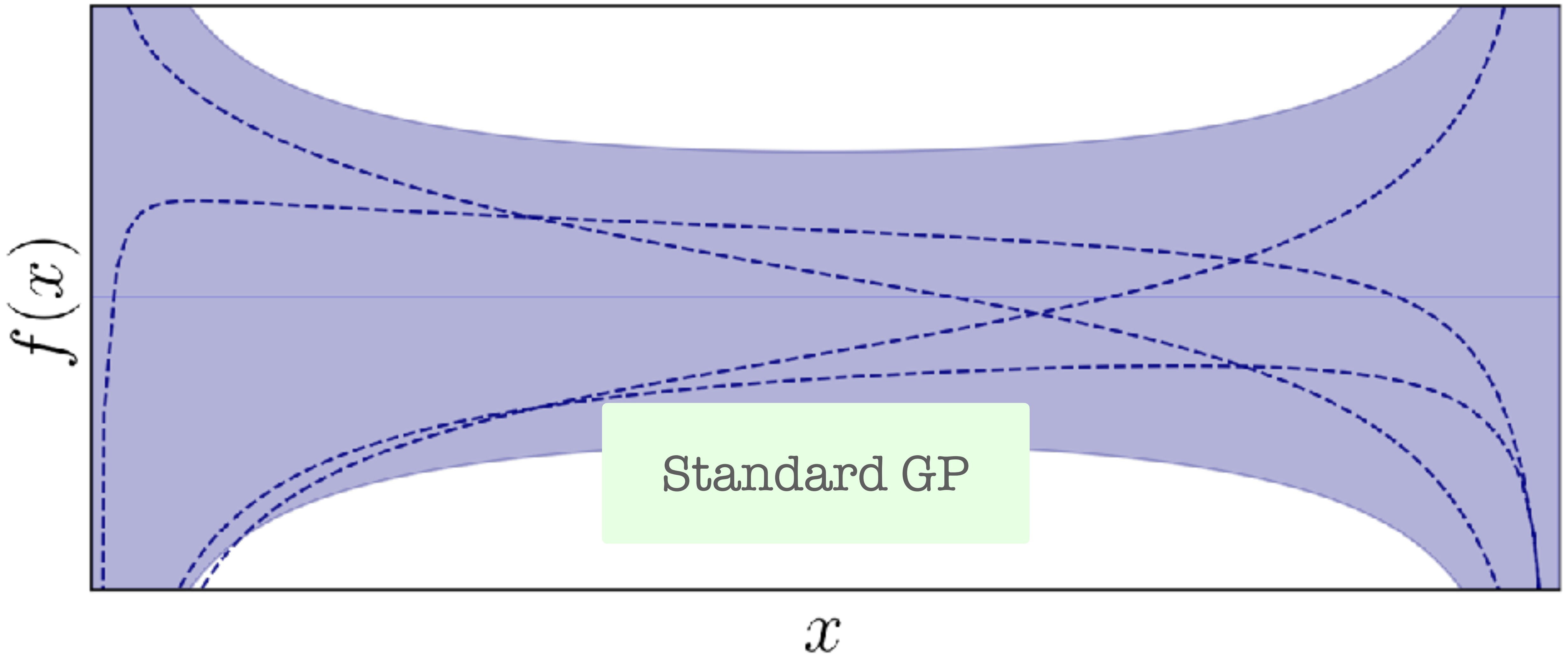
B $$k(x,\bar{x}) = \sum_{j=1}^{M} b^j x^j \bar{x}^j$$

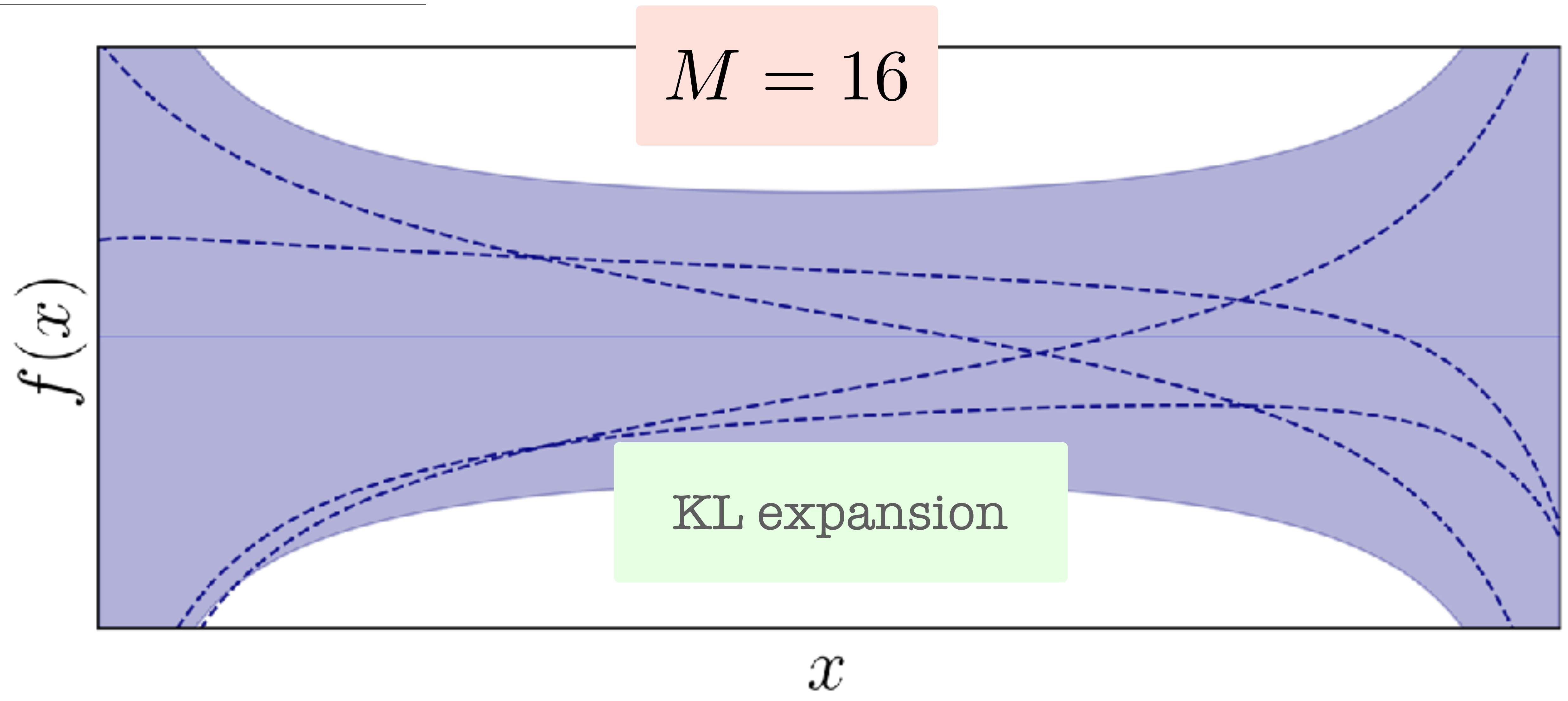
A $$f_M(x) \sim \sum_{j=1}^{M} \beta_j x^j, \quad \beta_j \sim \mathcal{N}(0, b^j)$$

Convergence in mean-square sense

$$\lim_{M \to \infty} \mathbb{E}\left[\left(f_\infty(x) - \sum_{j=1}^{M} \beta_j \phi_j(x)\right)^2\right] = 0$$

Efficient long-term predictions: Karhunen-Loève expansion of GPs





A $\quad k(x, \bar{x}) = \dfrac{1}{1 - bx\bar{x}} \qquad 0 < b < 1, \quad -1 < x, \bar{x} < 1$

B $\quad f_\infty(x) \sim \mathcal{GP}(0, k(x, \bar{x}))$

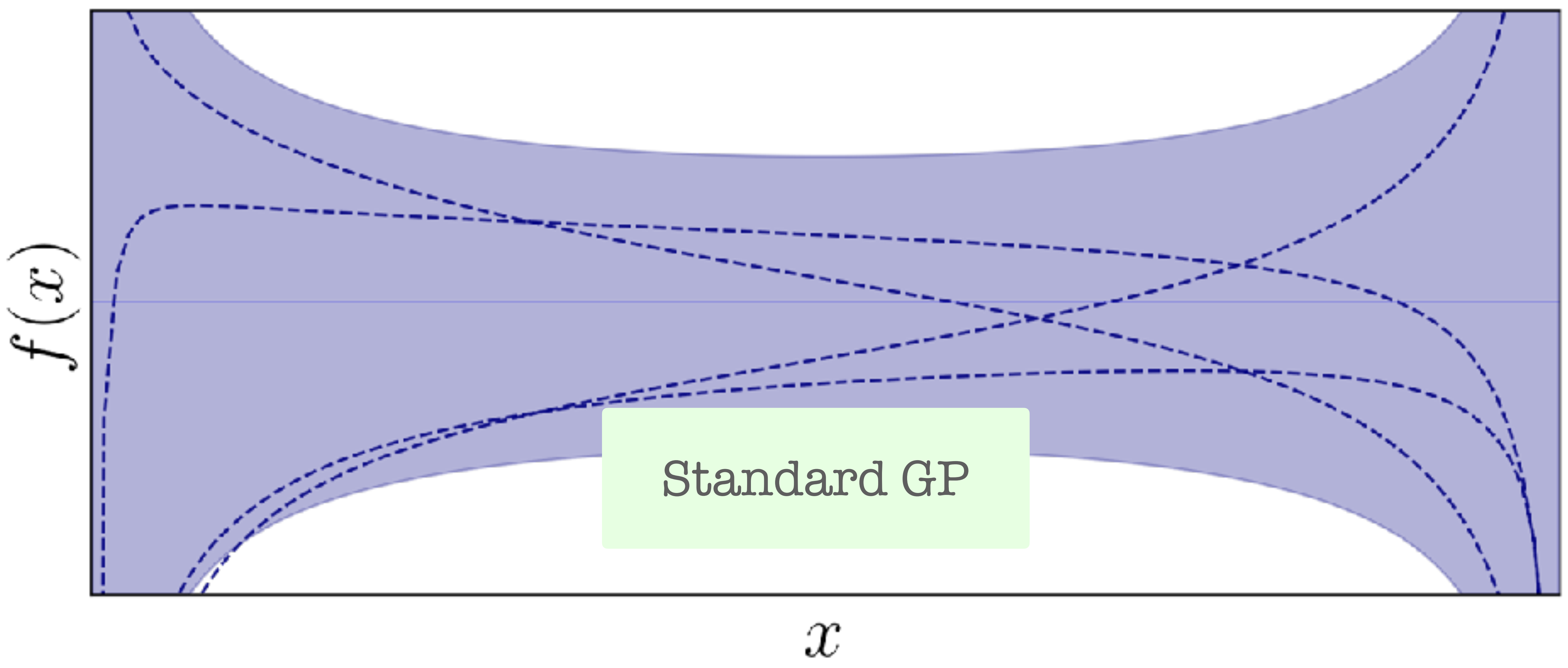
B $\quad k(x, \bar{x}) = \sum_{j=1}^{M} b^j x^j \bar{x}^j$

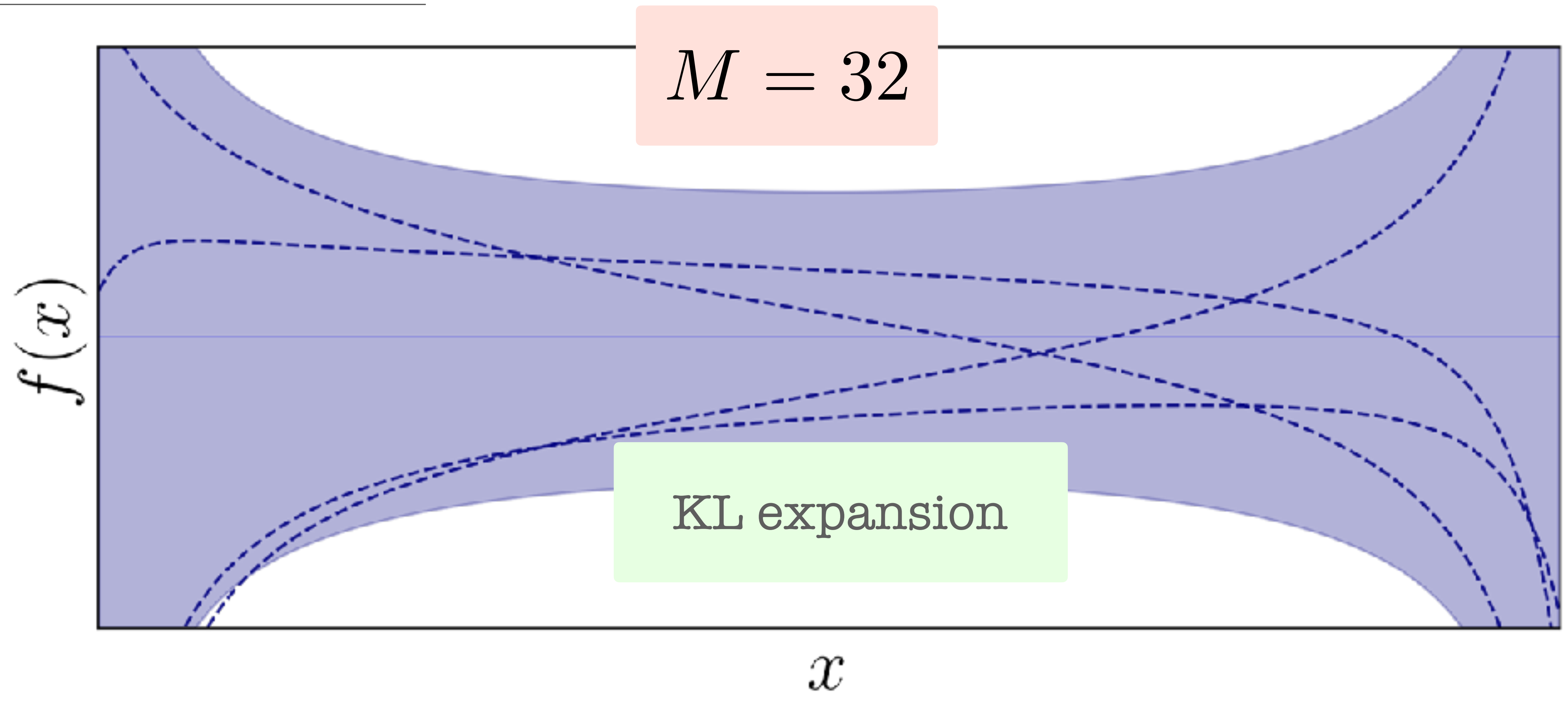
A $\quad f_M(x) \sim \sum_{j=1}^{M} \beta_j x^j, \quad \beta_j \sim \mathcal{N}(0, b^j)$

Convergence in mean-square sense

$$\lim_{M \to \infty} \mathbb{E}\left[\left(f_\infty(x) - \sum_{j=1}^{M} \beta_j \phi_j(x)\right)^2\right] = 0$$

Efficient long-term predictions: Karhunen-Loève expansion of GPs





A $k(x, \bar{x}) = \dfrac{1}{1 - bx\bar{x}}$ $\quad 0 < b < 1$, $\quad -1 < x, \bar{x} < 1$

↓

B $f_\infty(x) \sim \mathcal{GP}(0, k(x, \bar{x}))$

B $k(x, \bar{x}) = \sum_{j=1}^{M} b^j x^j \bar{x}^j$

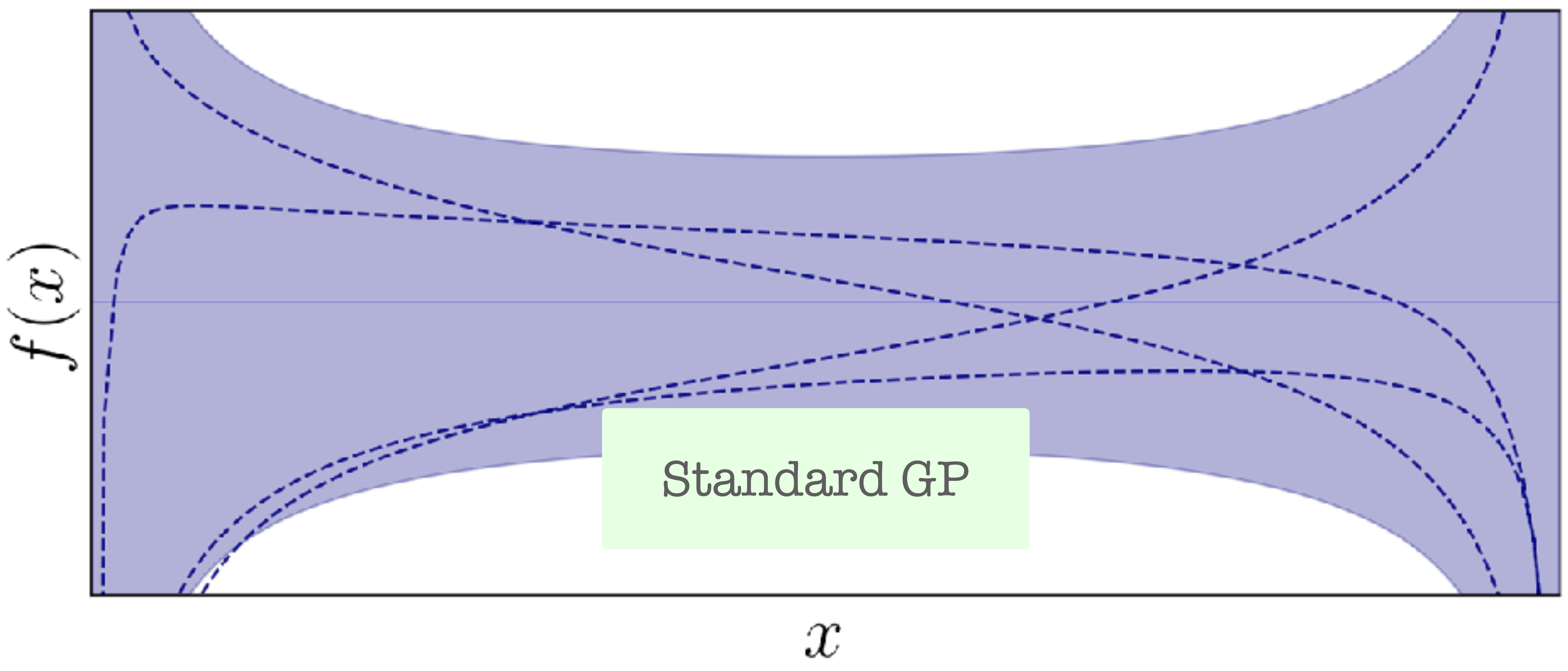
↑

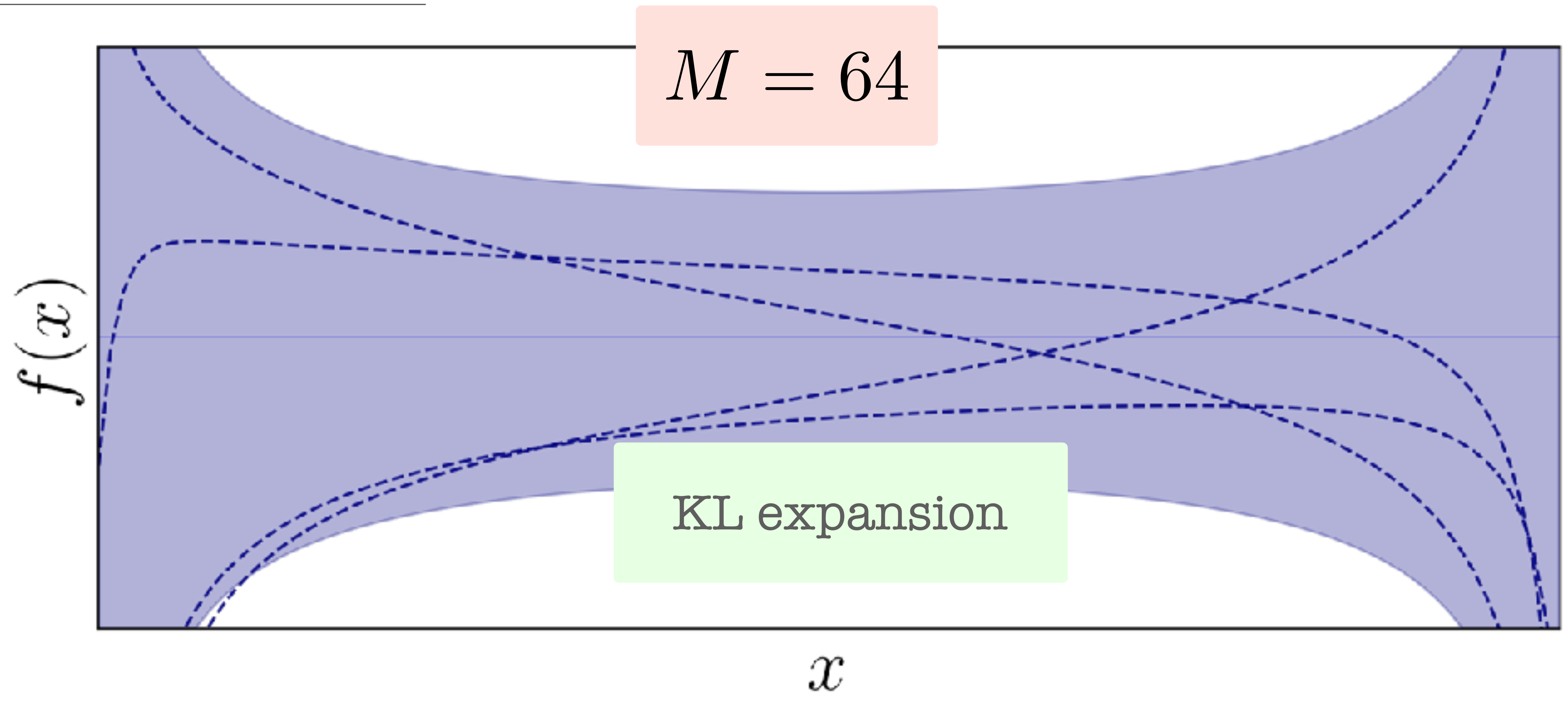
A $f_M(x) \sim \sum_{j=1}^{M} \beta_j x^j, \quad \beta_j \sim \mathcal{N}(0, b^j)$

Convergence in mean-square sense

$$\lim_{M \to \infty} \mathbb{E}\left[\left(f_\infty(x) - \sum_{j=1}^{M} \beta_j \phi_j(x)\right)^2\right] = 0$$

Efficient long-term predictions: Karhunen-Loève expansion of GPs





A $k(x, \bar{x}) = \frac{1}{1 - bx\bar{x}}$ $\quad 0 < b < 1 \quad -1 < x, \bar{x} < 1$

↓

B $f_\infty(x) \sim \mathcal{GP}(0, k(x, \bar{x}))$

B $k(x, \bar{x}) = \sum_{j=1}^{M} b^j x^j \bar{x}^j$

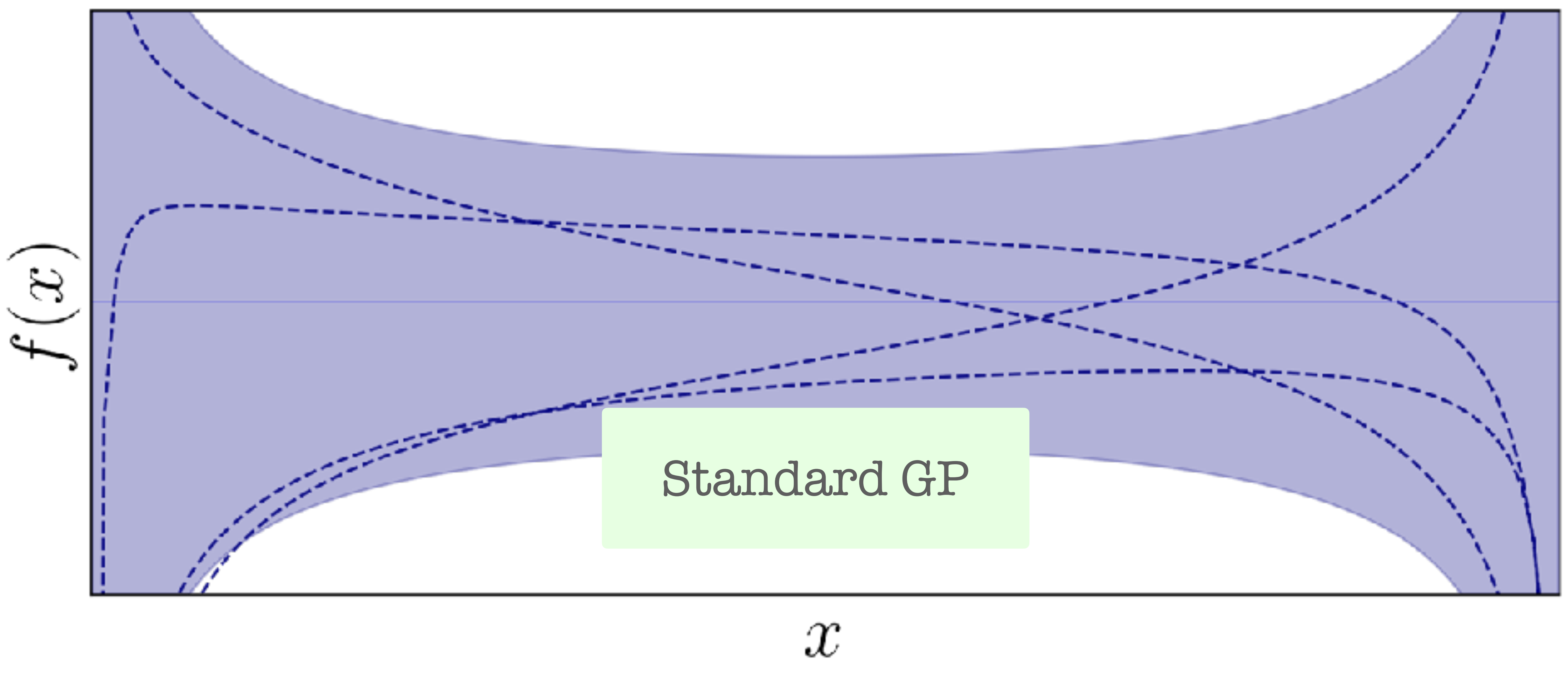
↑

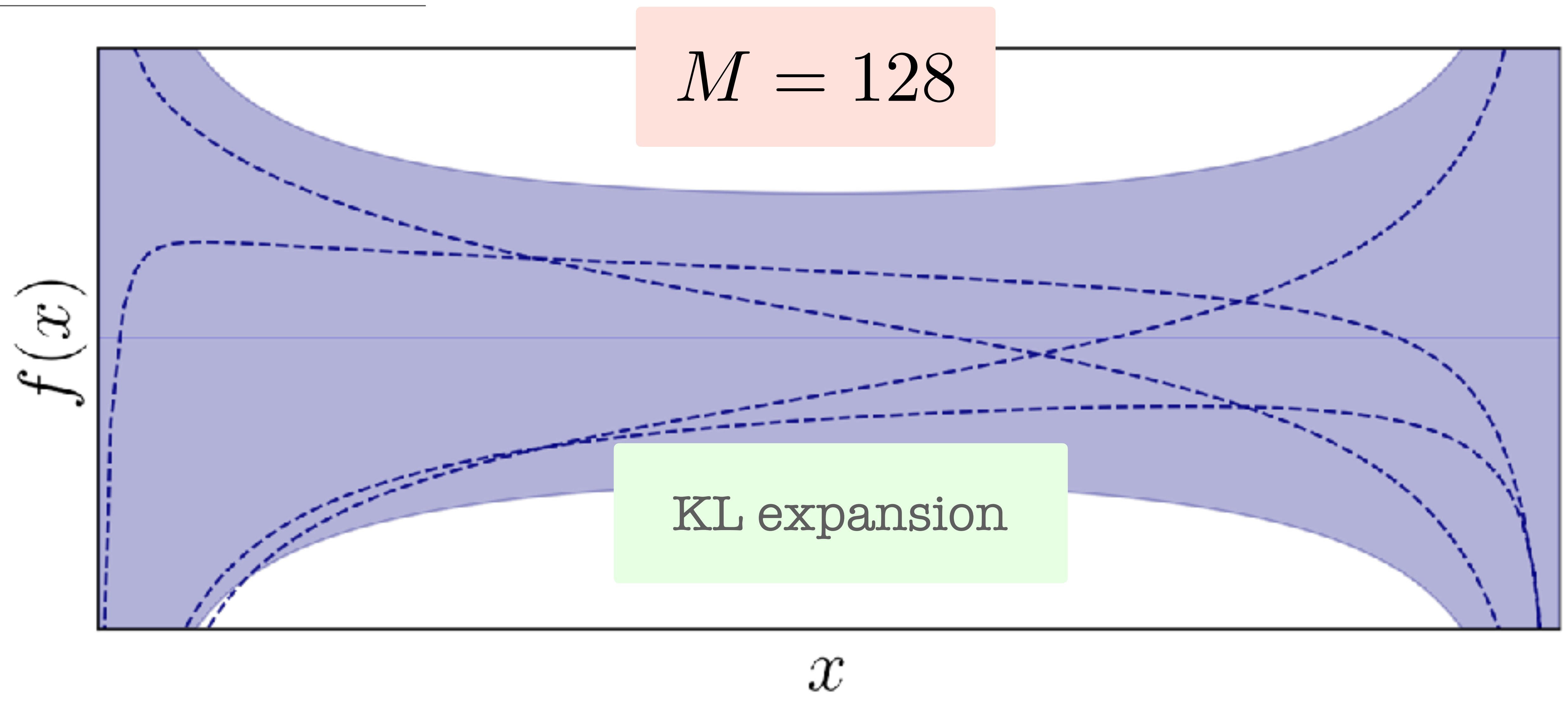
A $f_M(x) \sim \sum_{j=1}^{M} \beta_j x^j, \quad \beta_j \sim \mathcal{N}(0, b^j)$

Convergence in mean-square sense

$$\lim_{M \to \infty} \mathbb{E}\left[\left(f_\infty(x) - \sum_{j=1}^{M} \beta_j \phi_j(x)\right)^2\right] = 0$$

Efficient long-term predictions: Karhunen-Loève expansion of GPs





A $$k(x,\bar{x}) = \frac{1}{1 - bx\bar{x}} \qquad \begin{array}{c} 0 < b < 1 \\ -1 < x, \bar{x} < 1 \end{array}$$

$\downarrow$

B $$f_\infty(x) \sim \mathcal{GP}(0, k(x,\bar{x}))$$

B $$k(x,\bar{x}) = \sum_{j=1}^{M} b^j x^j \bar{x}^j$$

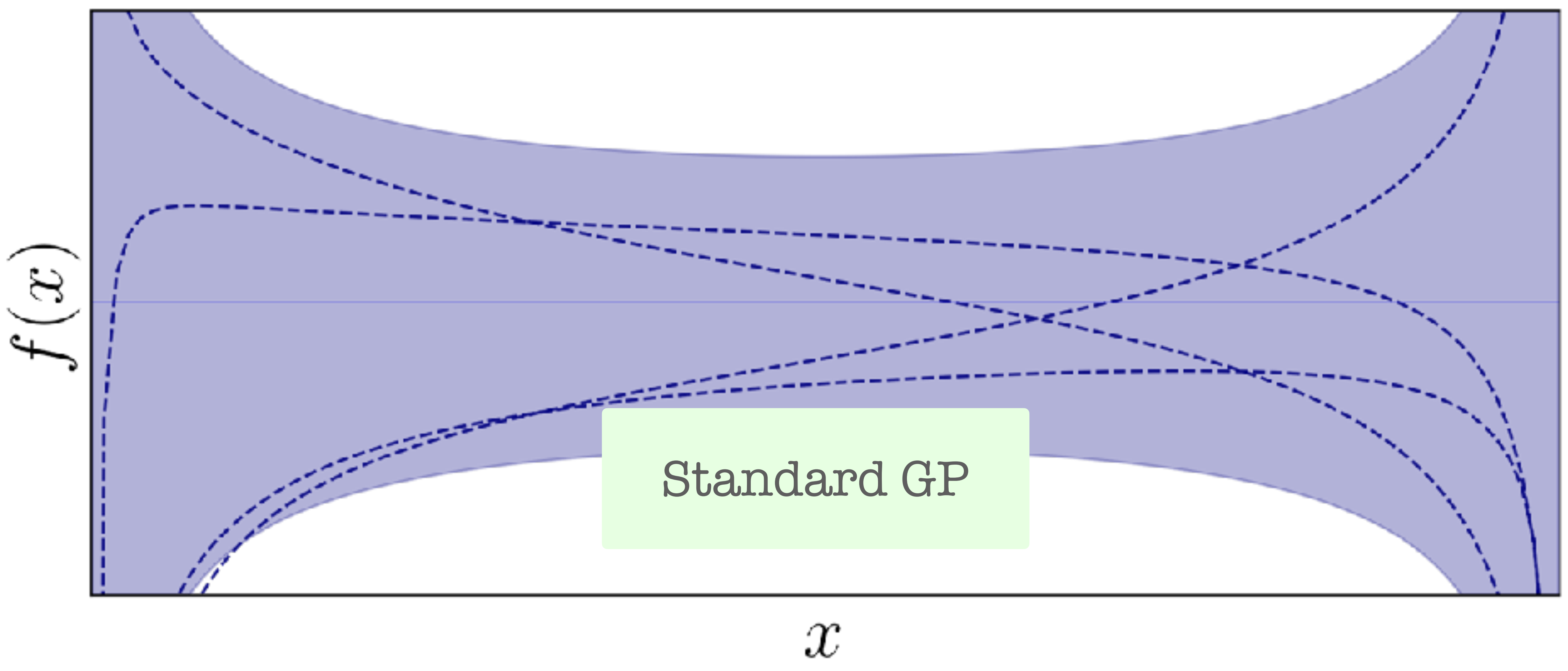
$\uparrow$

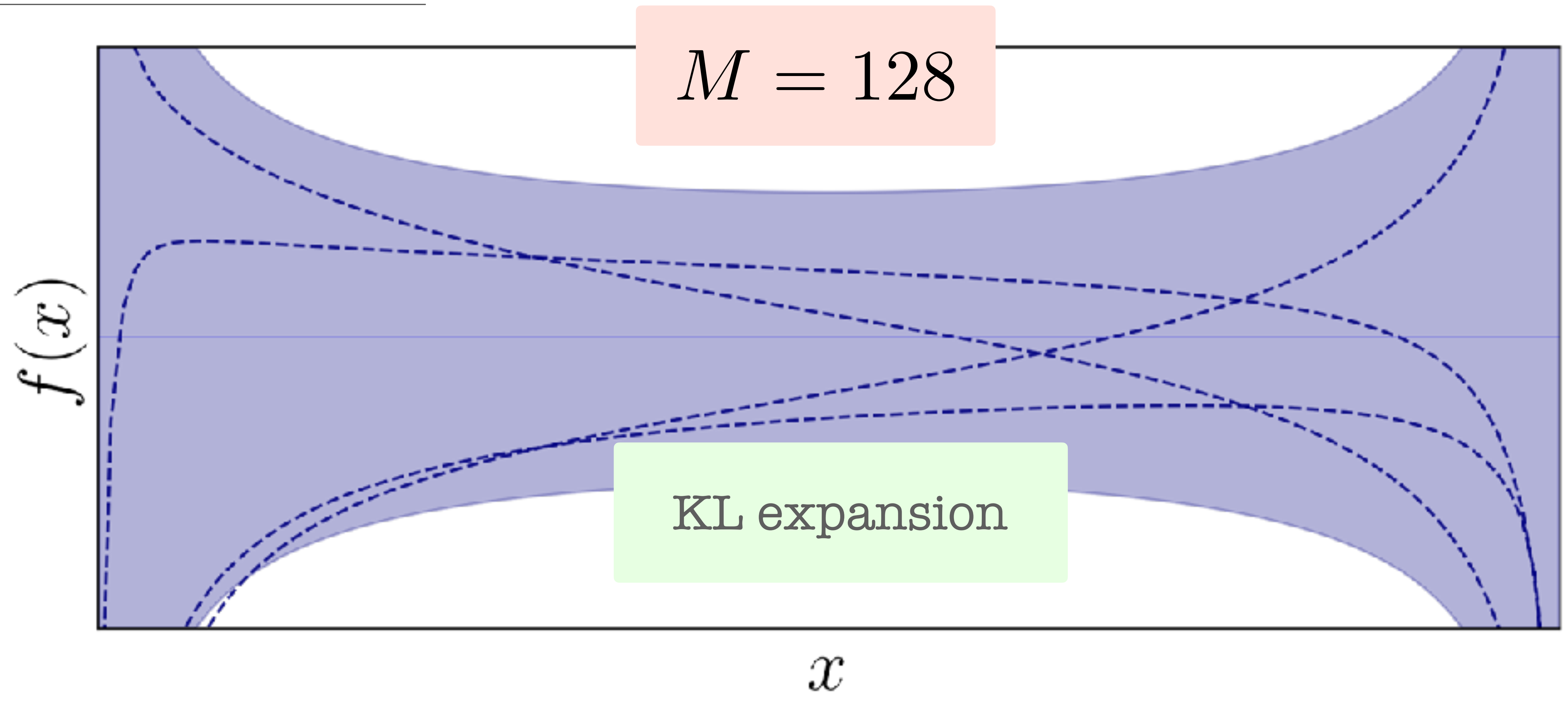
A $$f_M(x) \sim \sum_{j=1}^{M} \beta_j x^j, \quad \beta_j \sim \mathcal{N}(0, b^j)$$

Convergence in mean-square sense

$$\lim_{M \to \infty} \mathbb{E}\left[\left(f_\infty(x) - \sum_{j=1}^{M} \beta_j \phi_j(x)\right)^2\right] = 0$$

Efficient long-term predictions: Karhunen-Loève expansion of GPs





A $$k(x,\bar{x}) = \frac{1}{1 - bx\bar{x}} \qquad \begin{array}{c} 0 < b < 1 \\ -1 < x, \bar{x} < 1 \end{array}$$

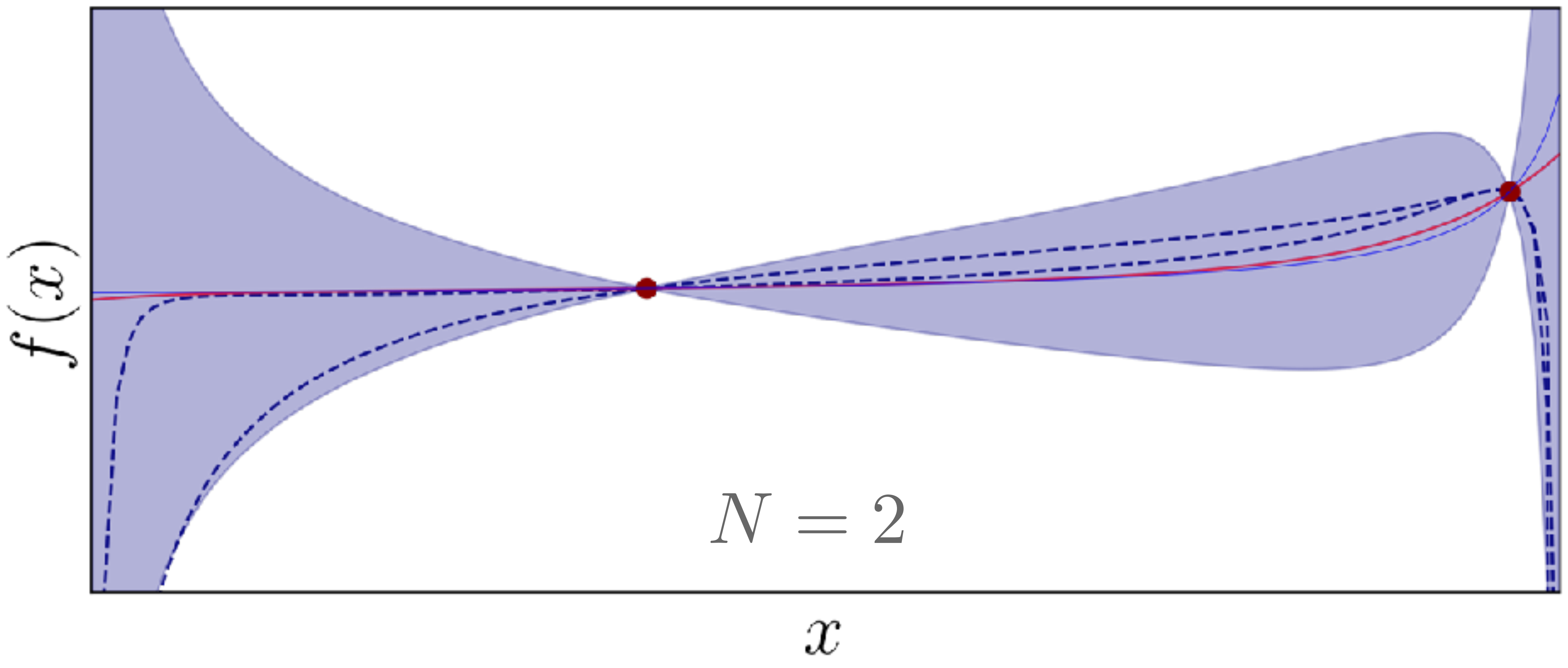
B $$f_\infty(x) \sim \mathcal{GP}(0, k(x,\bar{x}))$$

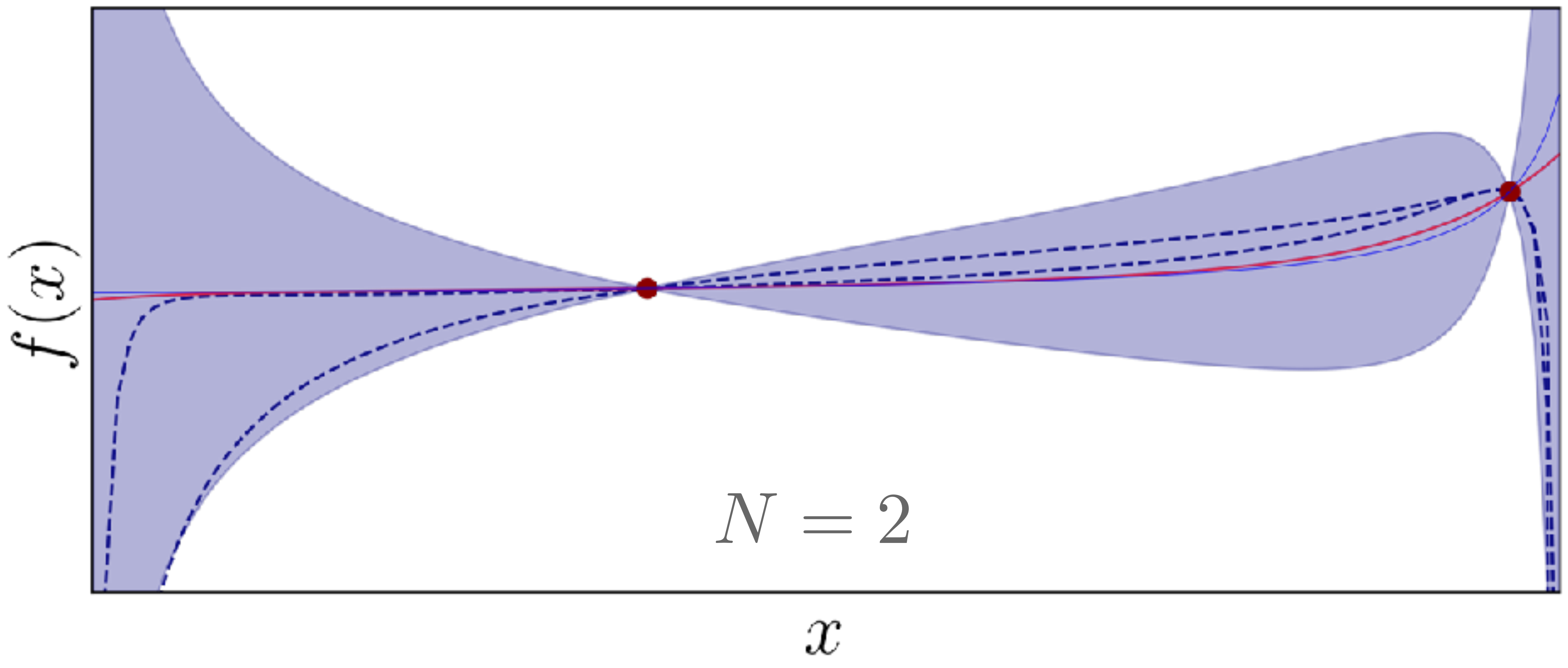
B $$k(x,\bar{x}) = \sum_{j=1}^{M} b^j x^j \bar{x}^j$$

A $$f_M(x) \sim \sum_{j=1}^{M} \beta_j x^j, \quad \beta_j \sim \mathcal{N}(0, b^j)$$

- Features encode prior information $\phi_j(\cdot) \rightarrow k(\cdot,\cdot)$
- Model expressivity limited by M

Efficient long-term predictions: Karhunen-Loève expansion of GPs





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

$$f(x) \sim \beta^\top \Phi(x), \quad \beta_j \sim \mathcal{N}(0, \nu_j)$$

Posterior

$$p(f|x, \mathcal{D}) = \mathcal{N}(\mu(x), \sigma^2(x))$$

$$\mu(x) = k(x, X)\underbrace{[k_{XX} + \sigma^2 I]^{-1}}_{O(N^3)\ ✗}Y$$

Posterior

$$p(f|x, \mathcal{D}) = \mathcal{N}(m_\beta^\top \Phi(x), \Phi^\top(x)\Sigma_\beta \Phi(x))$$

$$m_\beta = \overbrace{\underbrace{[\sigma^2 I + \Phi_X \Phi_X^\top]^{-1}}_{O(M^3)}\Phi_X Y}^{O(NM^3)\ ✓}$$

▶ Useful for large datasets $N \gg M$

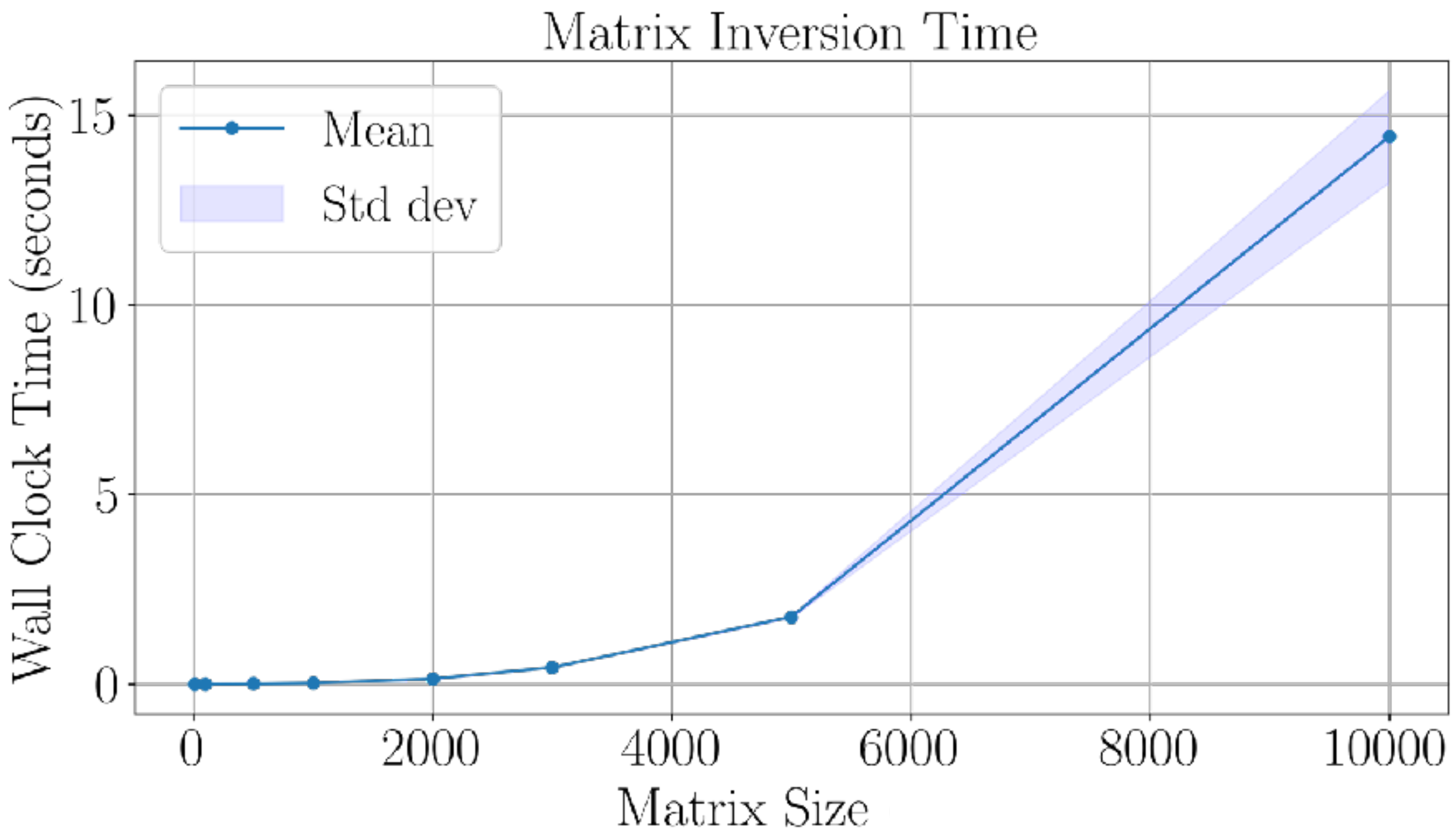
Efficient long-term predictions: Karhunen-Loève expansion of GPs

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

Posterior

$$p(f|x, \mathcal{D}) = \mathcal{N}(\mu(x), \sigma^2(x))$$

$$\mu(x) = k(x, X)\underbrace{[k_{XX} + \sigma^2 I]^{-1}}_{O(N^3)\ ✗}Y$$



$$f(x) \sim \beta^\top \Phi(x), \quad \beta_j \sim \mathcal{N}(0, \nu_j)$$

Posterior

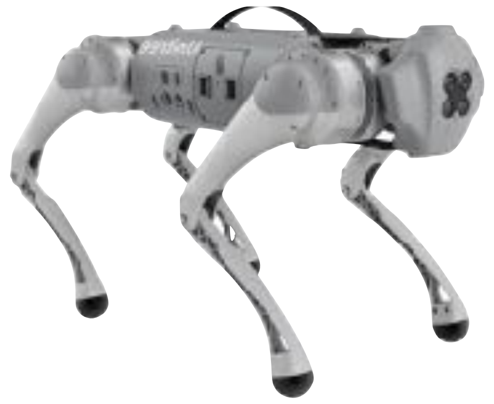
$$p(f|x, \mathcal{D}) = \mathcal{N}(m_\beta^\top \Phi(x), \Phi^\top(x)\Sigma_\beta \Phi(x))$$

$$m_\beta = \overbrace{\underbrace{[\sigma^2 I + \Phi_X \Phi_X^\top]^{-1}}_{O(M^3)}\Phi_X Y}^{O(NM^3)\ ✓}$$

▶ Useful for large datasets $N \gg M$

Increase data-efficiency for learning: embedding prior info via simulator

A $$f(x_t, u_t) \sim \sum_{j=1}^{M} \beta_j \phi_j(x_t, u_t), \quad \beta_j \sim \mathcal{N}(m_j, \nu_j)$$ → GPs: universal function approximations



$$= \sum_{j=1}^{M} \nu_j \phi_j(x_t, u_t) \phi_j(\bar{x}_t, \bar{u}_t)$$

B $$k(x_t, u_t, \bar{x}_t, \bar{u}_t) = \mathbb{C}\mathrm{ov}[f(x_t, u_t), f(\bar{x}_t, \bar{u}_t)]$$ → Encodes distribution over functions

Goal

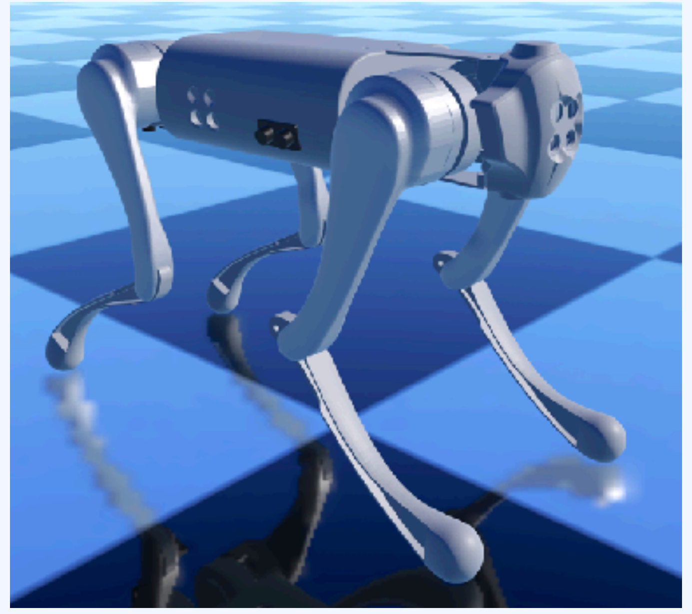
- Learn faster with informed features $\phi_j(\cdot)$
- Result: informed GP

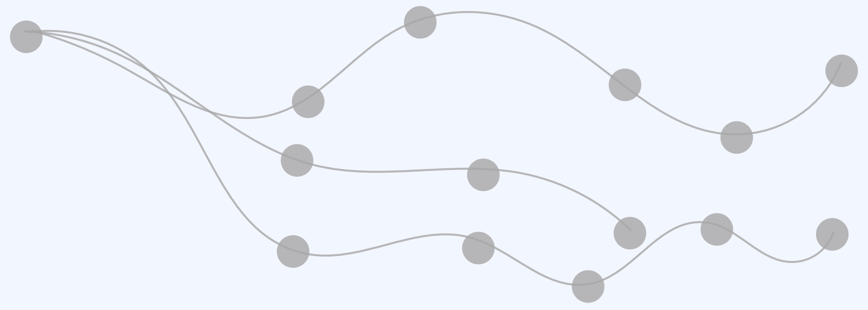
Increase data-efficiency for learning: embedding prior info via simulator

$$f_{\text{real}}(x_t, u_t) \sim \sum_{j=1}^{M} \beta_j \phi_j(x_t, u_t) \overset{\text{D}}{\approx} f_{\text{sim}}(x_t, u_t; \theta)$$

Simulate roll-outs

$$\beta_j \sim \mathcal{N}(m_j, \nu_j)$$





$$\theta \sim p(\theta)$$

▷ Moment matching

① Fourier series expansion

$$\sum_{j=1}^{M} \underbrace{\mathbb{E}[\beta_j]}\underbrace{\phi_j(x_t, u_t)} = \mathbb{E}_\theta[f_{\text{sim}}(x_t, u_t; \theta)]$$

▷ Square integrable

$$m_j \cos(\omega_j^\top [x_t; u_t] + \varphi_j)$$

$$\begin{cases} m_j = S(\omega_j) = |\mathcal{F}[\cdot]| \\ \varphi_j = \angle \mathcal{F}[\cdot] \\ \omega_j \sim S(\omega) \\ \nu_j = \sigma^2 S(\omega_j) \end{cases}$$

- Benefits: interpretability; kernel <-> Fourier
- Caveats: Low model capacity

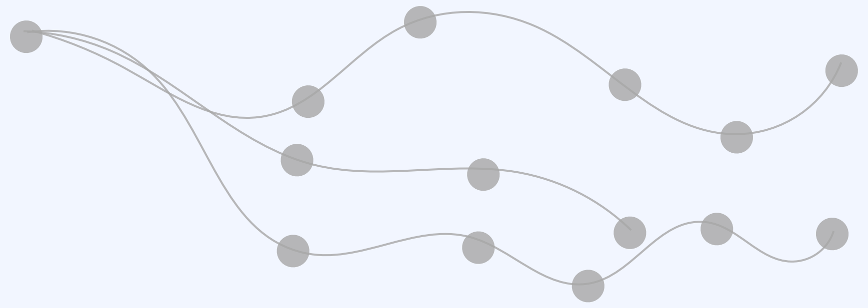
✘

Increase data-efficiency for learning: embedding prior info via simulator

$$f_{\text{real}}(x_t, u_t) \sim \sum_{j=1}^{M} \beta_j \phi_j(x_t, u_t) \overset{\text{D}}{\approx} f_{\text{sim}}(x_t, u_t; \theta)$$

$$\beta_j \sim \mathcal{N}(m_j, \nu_j)$$

Simulate roll-outs



$$\theta \sim p(\theta)$$

- Moment matching

① Fourier series expansion

$$\sum_{j=1}^{M} \underbrace{\mathbb{E}[\beta_j]}_{m_j} \underbrace{\phi_j(x_t, u_t)}_{\cos(\omega_j^\top \Psi(x_t, u_t) + \varphi_j)} = \mathbb{E}_\theta[f_{\text{sim}}(x_t, u_t; \theta)]$$

- Square integrable

$$m_j \cos(\omega_j^\top \Psi(x_t, u_t) + \varphi_j)$$

$$\begin{cases} m_j = 1/M \text{ [*]} \\ \omega_j \sim S(\omega) = \mathcal{N}(0, \Sigma) \\ \varphi_j \sim \text{U}(-\pi, \pi) \\ \nu_j = \sigma^2 S(\omega_j) \end{cases}$$

- Benefits: High model capacity
- Caveats: Low interpretability ✓

[*] Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J. and Ng, R., 2020. Fourier features let networks learn high frequency functions in low dimensional domains. NeurIPS

Increase data-efficiency for learning: embedding prior info via simulator

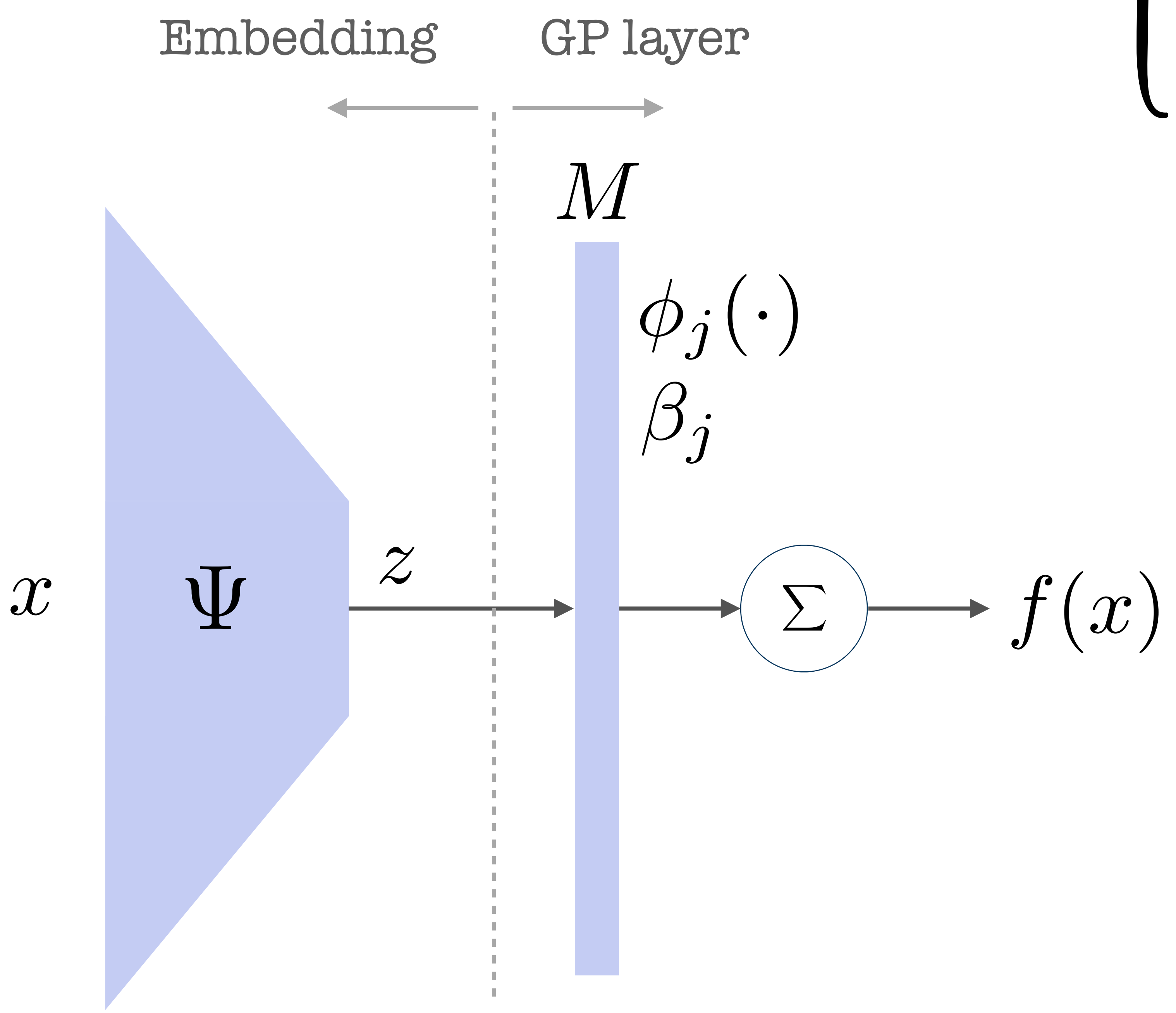
① Fourier series expansion

$$\sum_{j=1}^{M} \underbrace{\mathbb{E}[\beta_j]}_{m_j} \underbrace{\phi_j(x_t, u_t)}_{\cos(\omega_j^\top \Psi(x_t, u_t) + \varphi_j)} = \mathbb{E}_\theta[f_{\text{sim}}(x_t, u_t; \theta)]$$

▶ Square integrable

$$\begin{cases} m_j = 1/M \quad [*] \\ \omega_j \sim S(\omega) = \mathcal{N}(0, \Sigma) \end{cases}$$

$$\begin{cases} \varphi_j \sim \mathrm{U}(-\pi, \pi) \\ \nu_j = \sigma^2 S(\omega_j) \end{cases}$$



Increase data-efficiency for learning: embedding prior info via simulator

① Fourier series expansion $$\sum_{j=1}^{M} \underbrace{\mathbb{E}[\beta_j]}_{m_j} \underbrace{\phi_j(x_t, u_t)}_{\cos(\omega_j^\top \Psi(x_t, u_t) + \varphi_j)} = \mathbb{E}_\theta[f_{\text{sim}}(x_t, u_t; \theta)]$$

$$\begin{cases} m_j = 1/M \\ \omega_j \sim S(\omega) = \mathcal{N}(0, \Sigma) \\ \varphi_j \sim \mathrm{U}(-\pi, \pi) \\ \nu_j = \sigma^2 S(\omega_j) \end{cases}$$

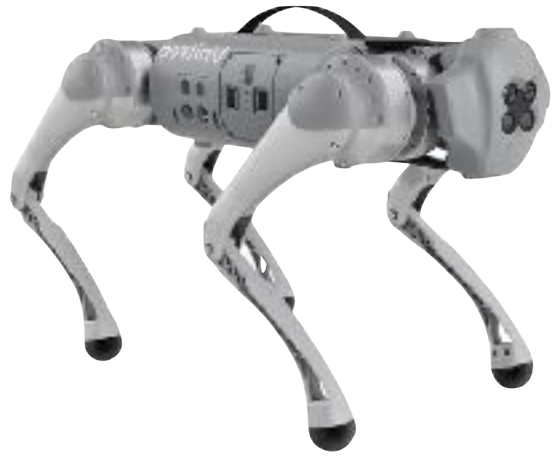
② Moment matching: $$\Psi_* = \arg\min_{\Psi} \int_{(x_t, u_t) \in \mathcal{X} \times \mathcal{U}} ||\mathbb{E}_\theta f_{\text{sim}}(x_t, u_t) - \sum_j m_j \phi_j(x_t, u_t)|| + \lambda ||\mathbb{V}\mathrm{ar}_\theta f_{\text{sim}}(x_t, u_t) - \sum_j \nu_j \phi_j^2(x_t, u_t)|| \mathrm{d}\rho(x_t, u_t)$$

Monte Carlo: $$\frac{1}{RT} \sum_{r=1}^{R} \sum_{t=1}^{T} \mathcal{L}(\hat{x}_t^{(r)}, \hat{u}_t^{(r)})$$

Increase data-efficiency for learning: embedding prior info via simulator

A

$$f_{\text{real}}(x_t, u_t) \sim \sum_{j=1}^{M} \beta_j \cos(\omega_j^\top \Psi_*(x_t, u_t) + \varphi_j) \quad \beta_j \sim \mathcal{N}(m_j, \nu_j)$$



B

$$k(x_t, u_t, \bar{x}_t, \bar{u}_t) = \sum_{j=1}^{M} \nu_j \cos(\omega_j^\top \Psi_*(x_t, u_t) + \varphi_j) \cos(\omega_j^\top \Psi_*(\bar{x}_t, \bar{u}_t) + \varphi_j)$$

Increase data-efficiency for learning: toy example

$$f_{\text{real}}(x_t) \sim \sum_{j=1}^{M} \beta_j \phi_j(x_t) \overset{\text{D}}{\approx} f_{\text{sim}}(x_t;\theta) = \theta x_t^2$$

$$\theta \sim \text{U}(-1,1)$$

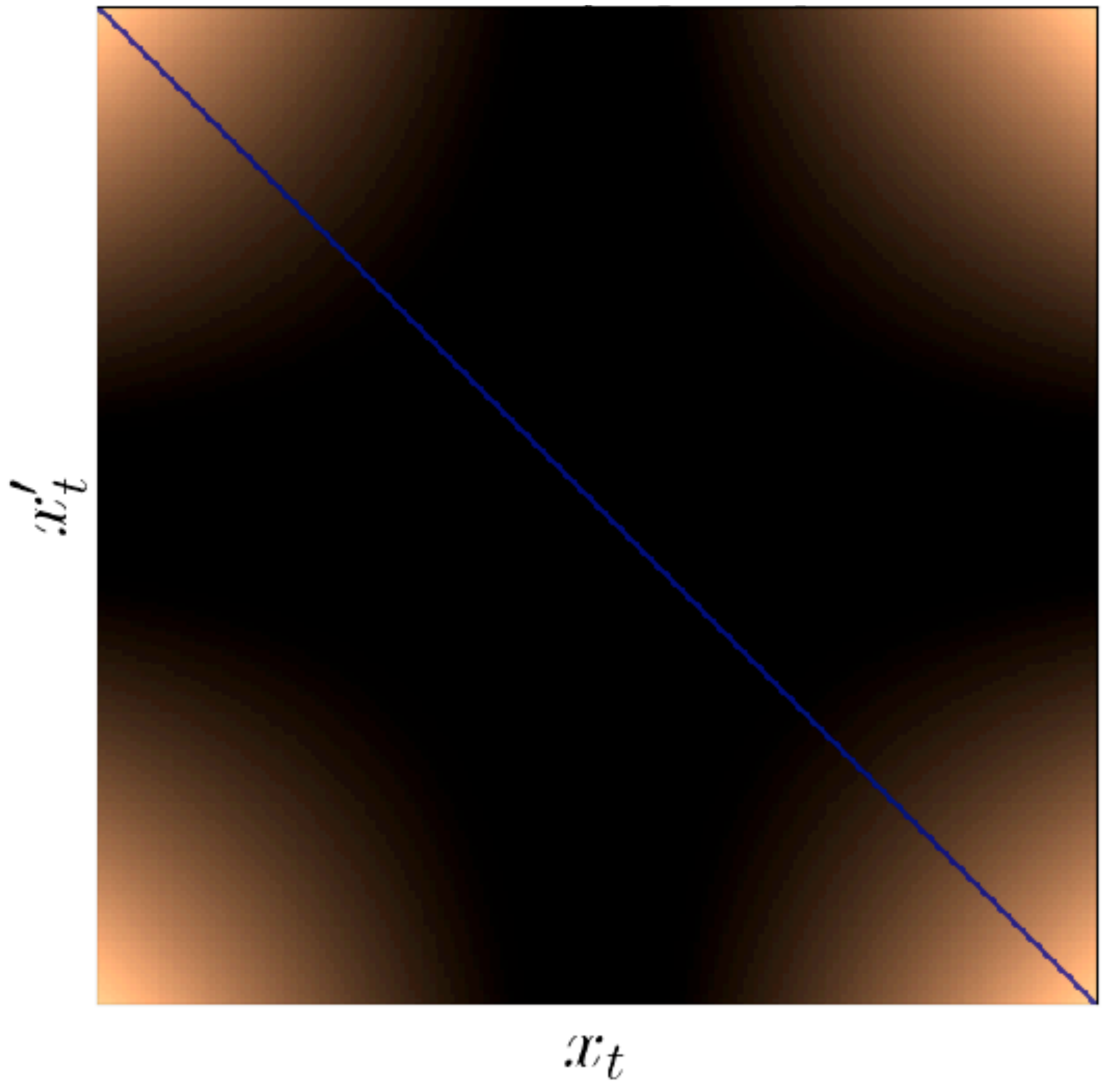
$$\beta_j \sim \mathcal{N}(m_j, \nu_j)$$

$$\phi_j(x_t) = \cos(\omega_j^\top \Psi(x_t) + \varphi_j)$$

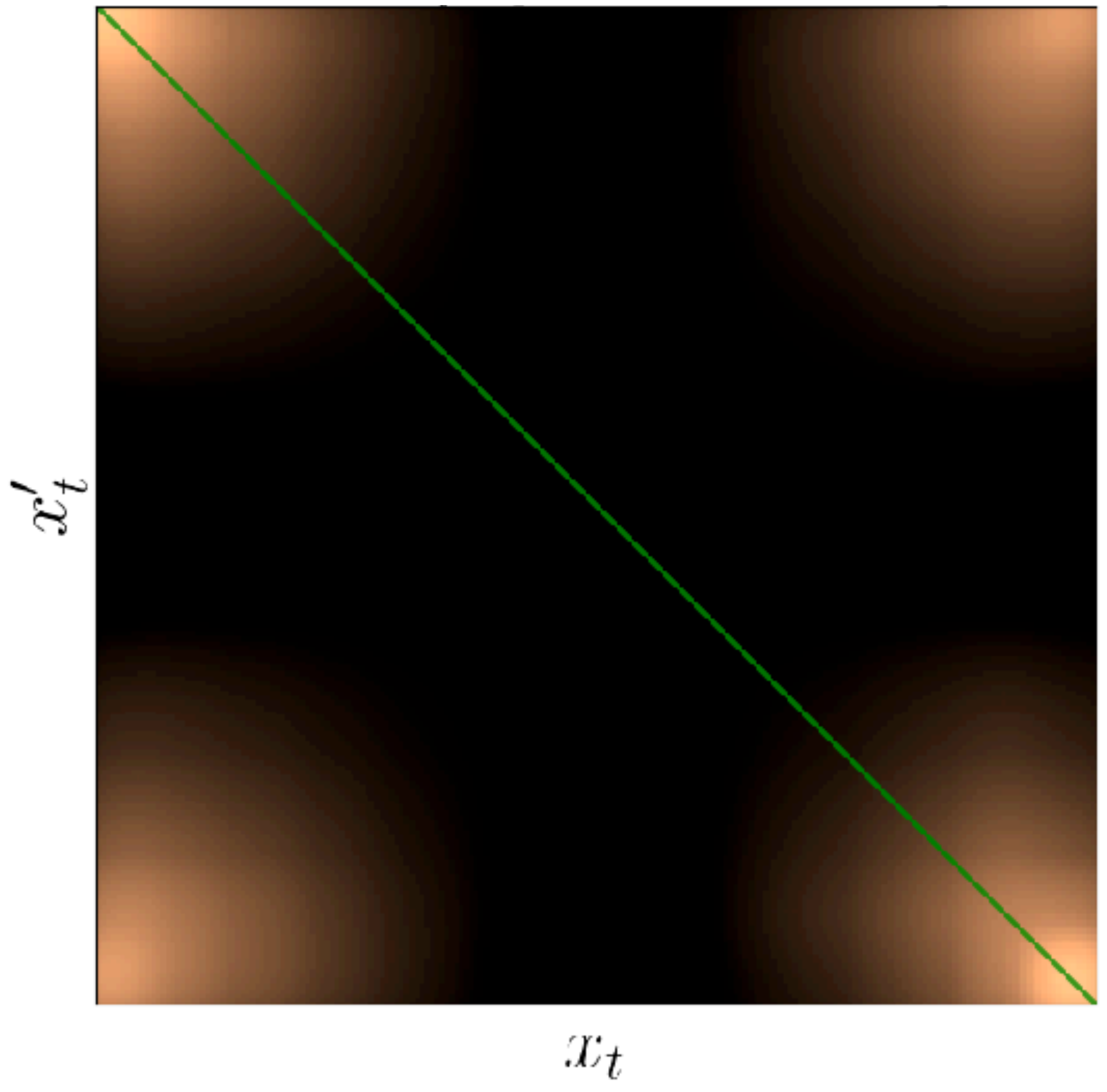
Moment matching ⟶ Ψ_*

$$\begin{cases} \mathbb{E}[f_{\text{sim}}(x_t;\theta)] = 0 & \approx \sum_j m_j \phi_j(x_t) \\ \mathbb{V}\text{ar}[f_{\text{sim}}(x_t;\theta)] = \frac{1}{3} x_t^4 & \approx \sum_j \nu_j \phi_j(x_t) \phi_j(\bar{x}_t) \end{cases}$$

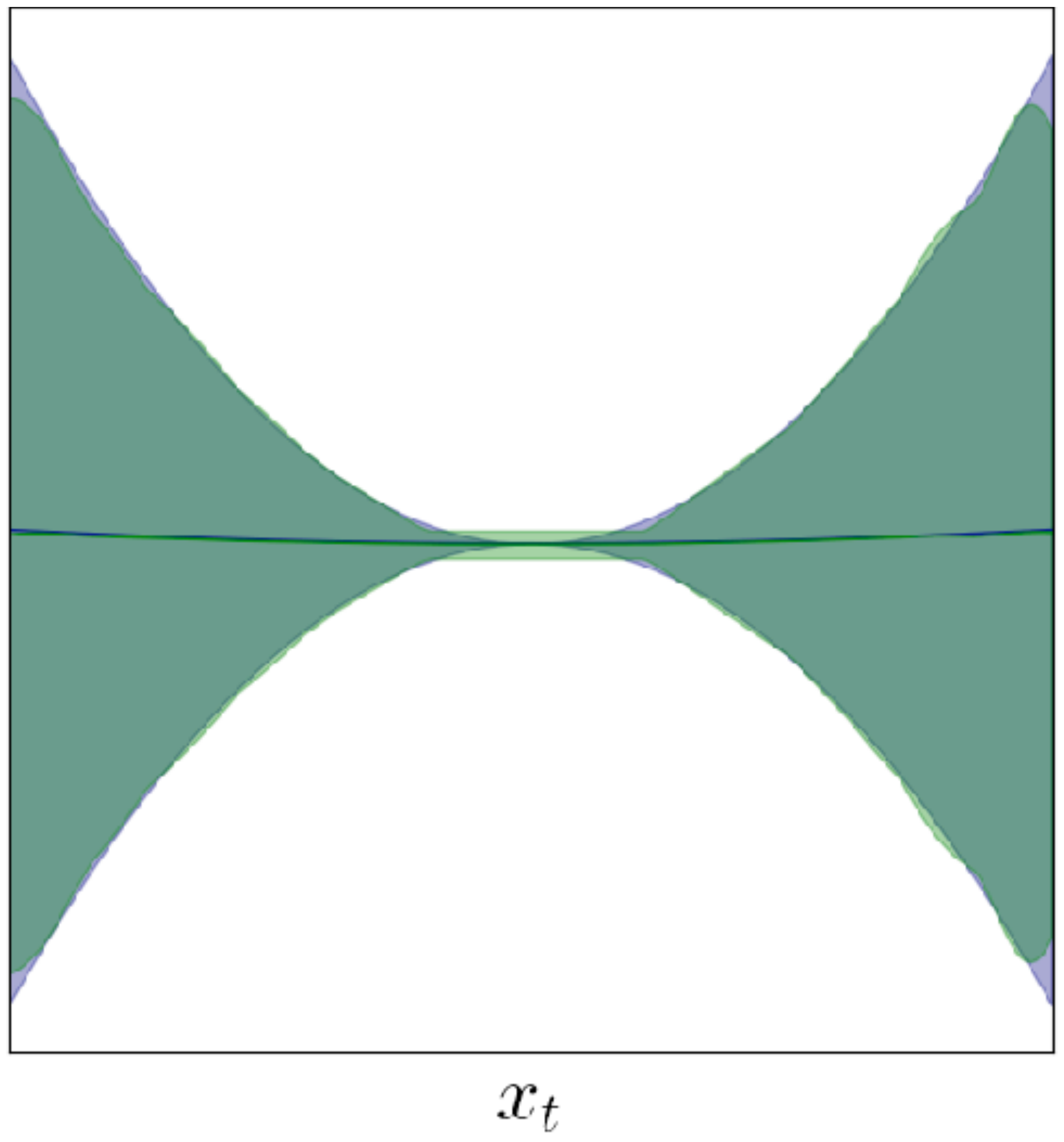
$$k_{\text{sim}}(x, \bar{x}_t) = \mathbb{C}\text{ov}(x_t, \bar{x}_t) = \frac{1}{3} x_t^2 \bar{x}_t^2$$



$$k_{\text{rec}}(x, \bar{x}_t)$$



$$M = 20$$

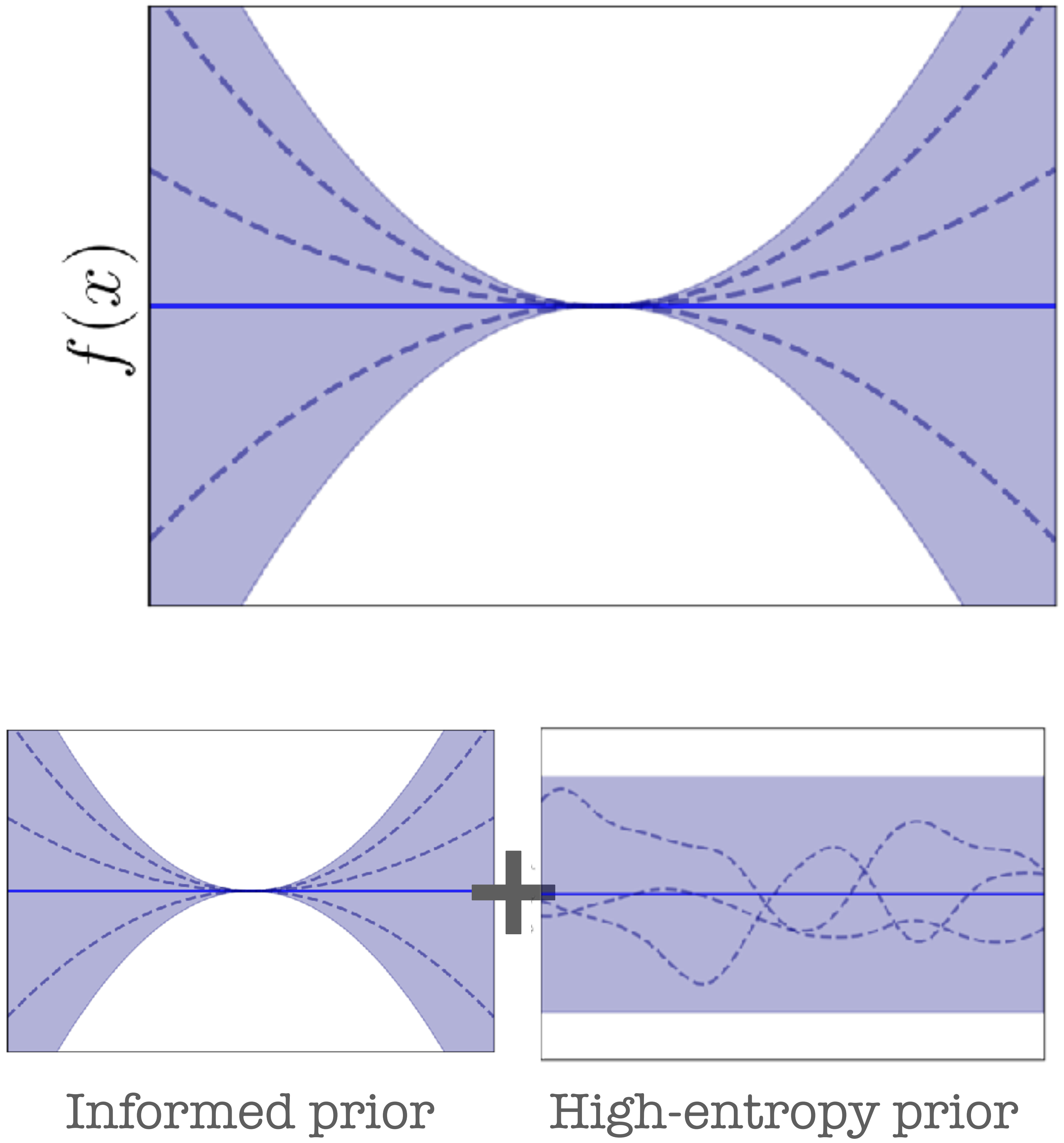


Increase data-efficiency for learning: toy example

$$f_{\text{real}}(x_t) \sim \sum_{j=1}^{M} \beta_j \phi_j(x_t) \overset{\text{D}}{\approx} f_{\text{sim}}(x_t;\theta) = \theta x_t^2$$

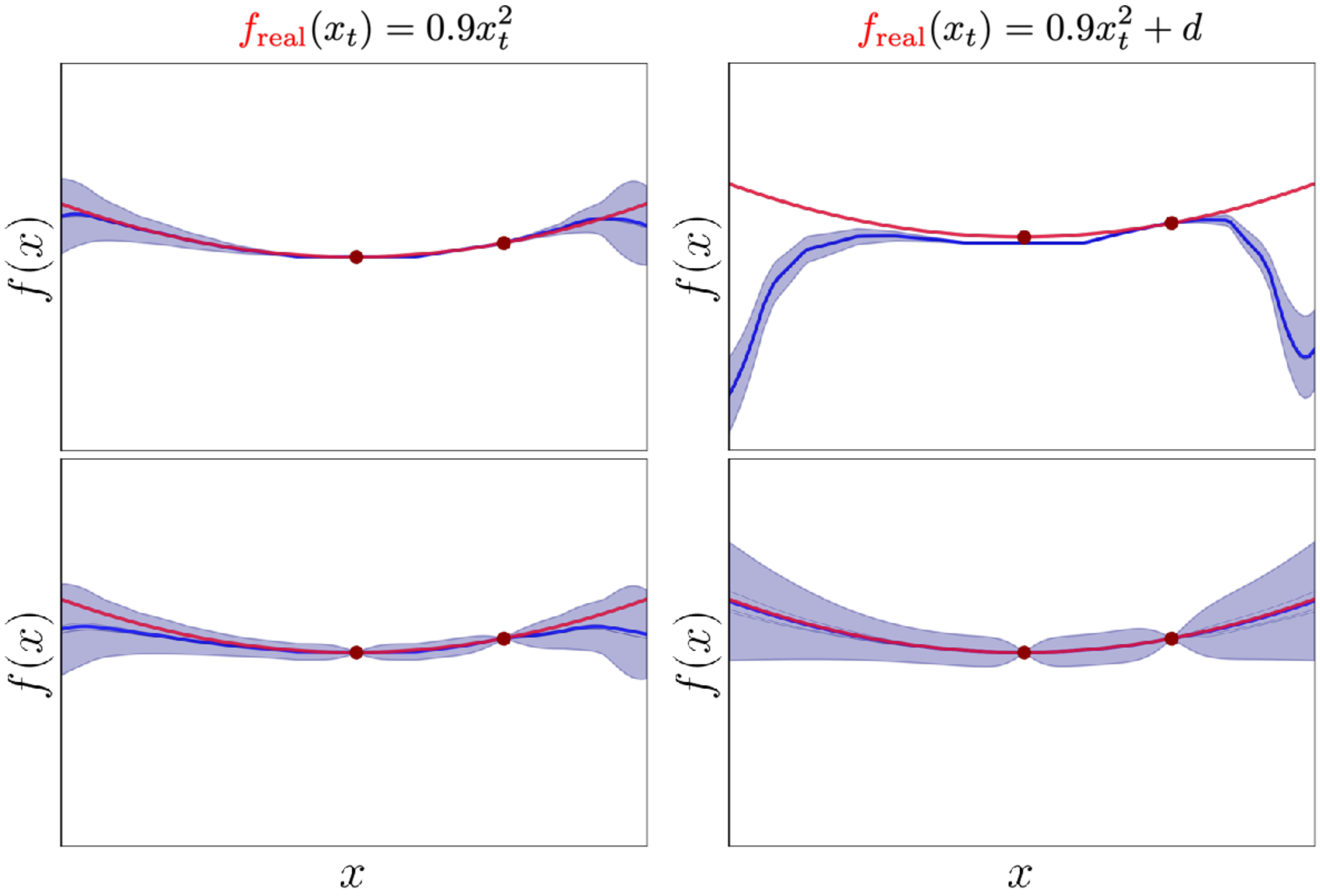
$$\theta \sim \mathrm{U}(-1,1)$$

- Stiff model
- Hypothesis space very reduced



Informed prior

High-entropy prior



$f_{\text{real}}(x_t) = 0.9x_t^2$

$f_{\text{real}}(x_t) = 0.9x_t^2 + d$

Model training: preliminary results

- Goals
  1) Validate predictive capabilities of the informed GPSSM
  2) Validate the OoD detection

- State

$x_t = [\mathrm{x}_t, \mathrm{y}_t, \theta_t]$ VICON

$\dot{x}_t = [\dot{\mathrm{x}}_t, \dot{\mathrm{y}}_t, \dot{\theta}_t]$ Estimated

- Commands

$u_t = [\mathrm{v}_t, \omega_t]$

world

c++

Python

Vicon

Robot

/vicon/go1/go1

100 Hz

ROS node

Robot Interface

- Connect via UDP
- Read state from sensors and publish them
- Send commands collected from subscribers

500 Hz

highState.msg

/high_state_from_robot

ROS node

Robot State Creation

500 Hz

Go1State.msg

/experiments_gpssm_ood/robot_state

100 Hz

ROS node

Control Task

- Follow velocity profile
- Walk to random goal

highCmd.msg

highCmd.msg

/high_cmd_to_robot

DataCollection.msg

- Trigger with Start/Stop

/experiments_gpssm_ood/data_collection_triggers

ROS node

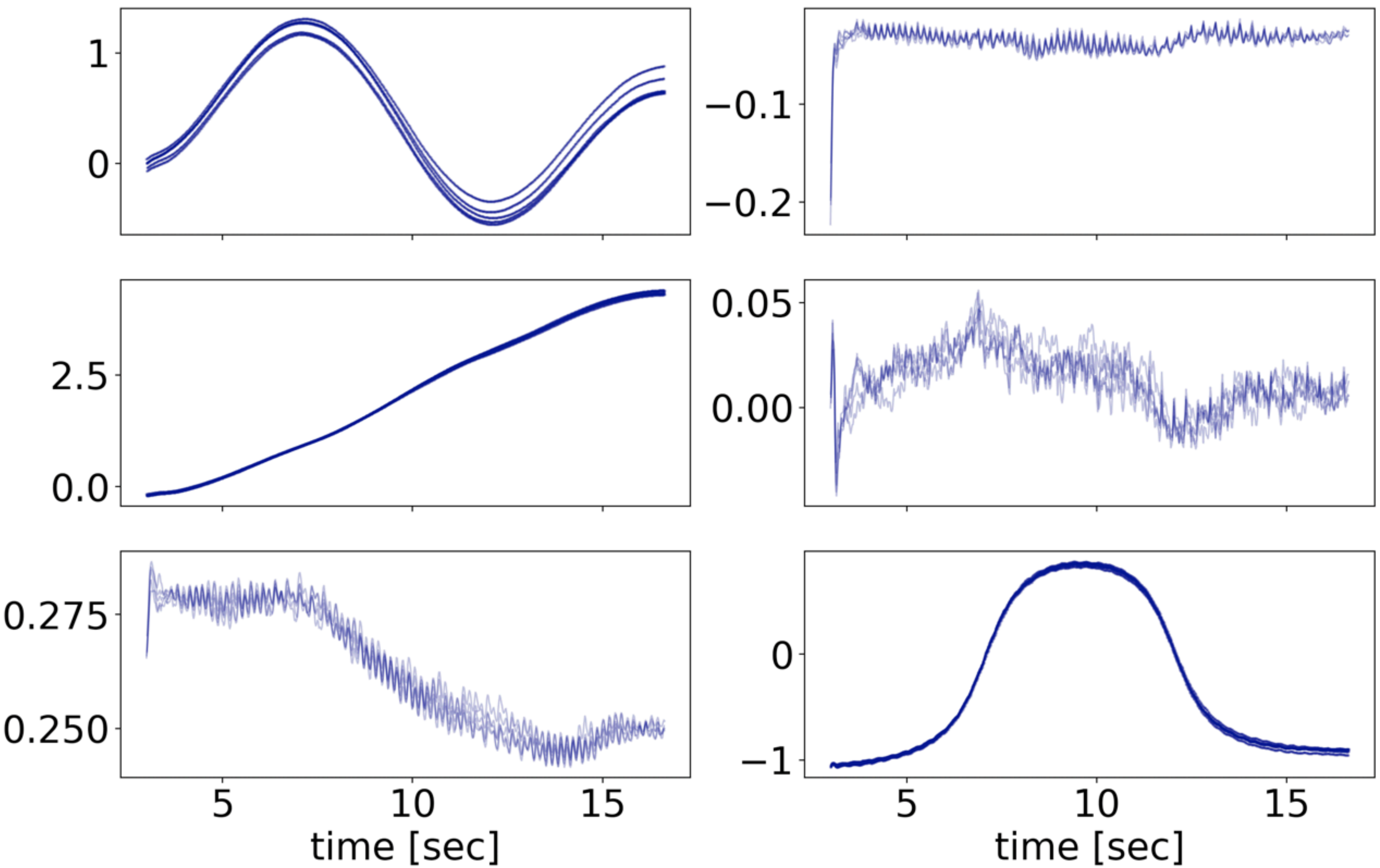
Data Collection

500 Hz

.pickle file

Model training: preliminary results

Robot Pose (from Vicon)



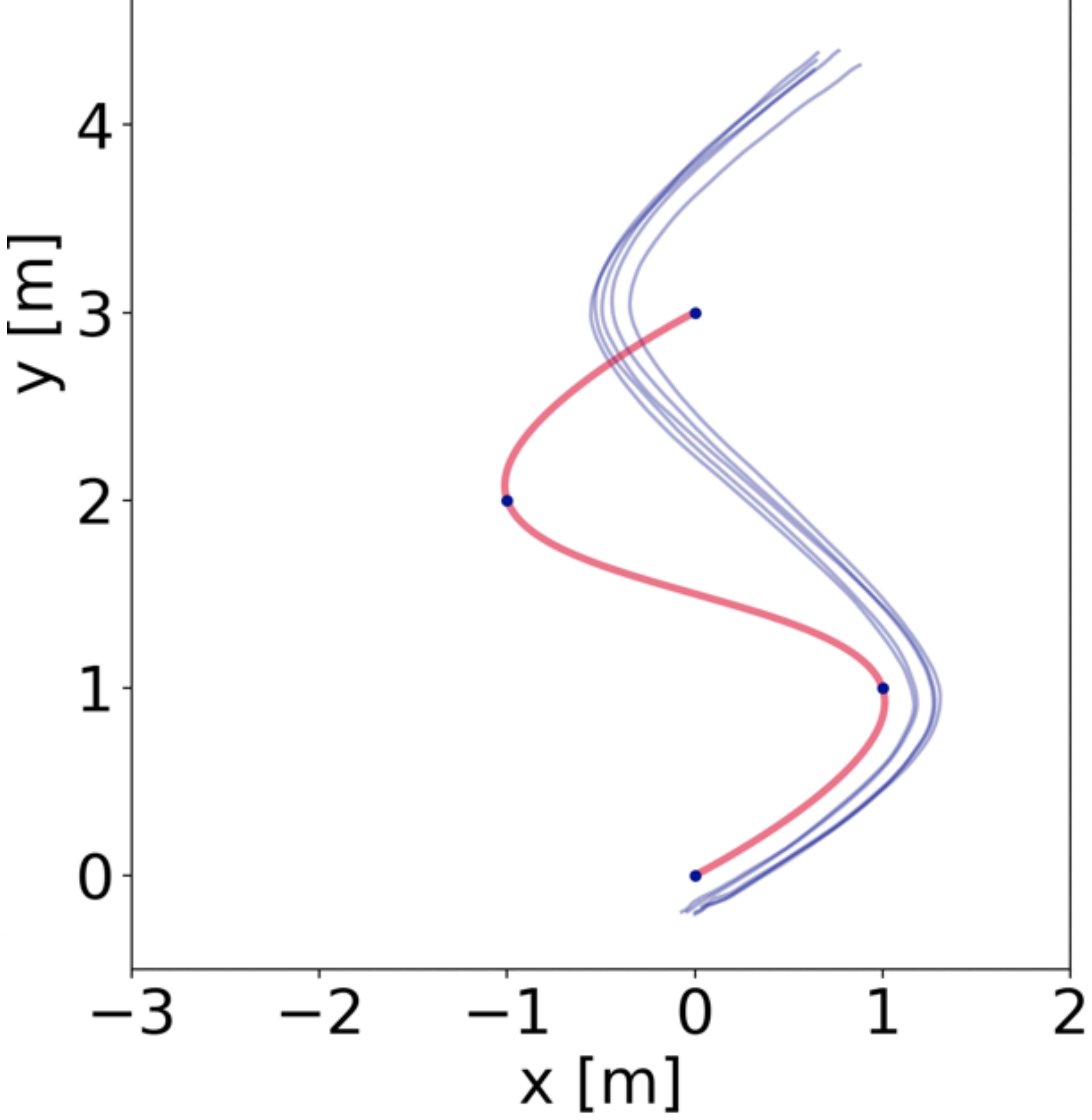
- State

$$x_t = [\mathrm{x}_t, \mathrm{y}_t, \theta_t]$$ VICON

$$\dot{x}_t = [\dot{\mathrm{x}}_t, \dot{\mathrm{y}}_t, \dot{\theta}_t]$$ Estimated

- Commands

$$u_t = [\mathrm{v}_t, \omega_t]$$



Model training: preliminary results

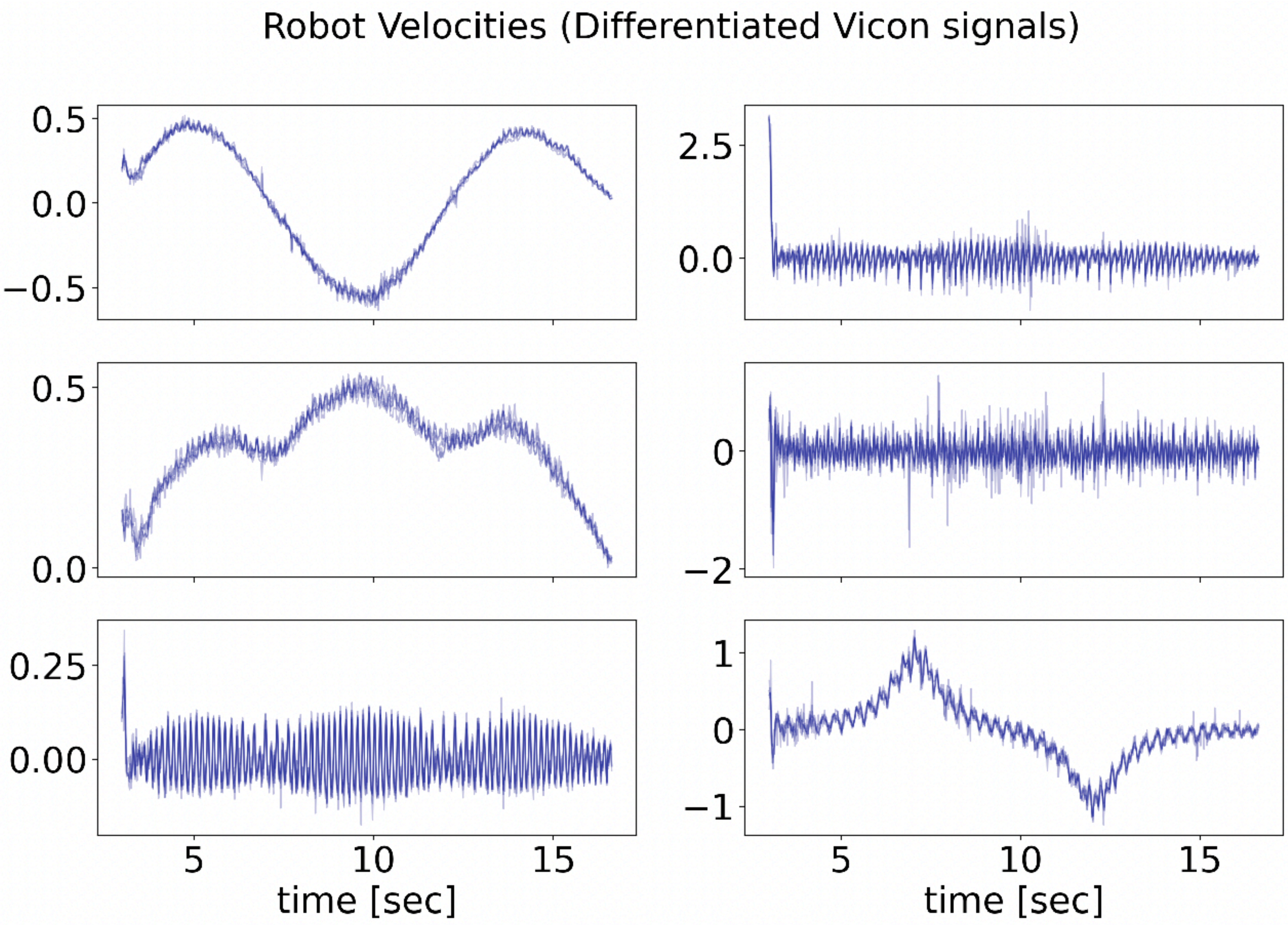
- State

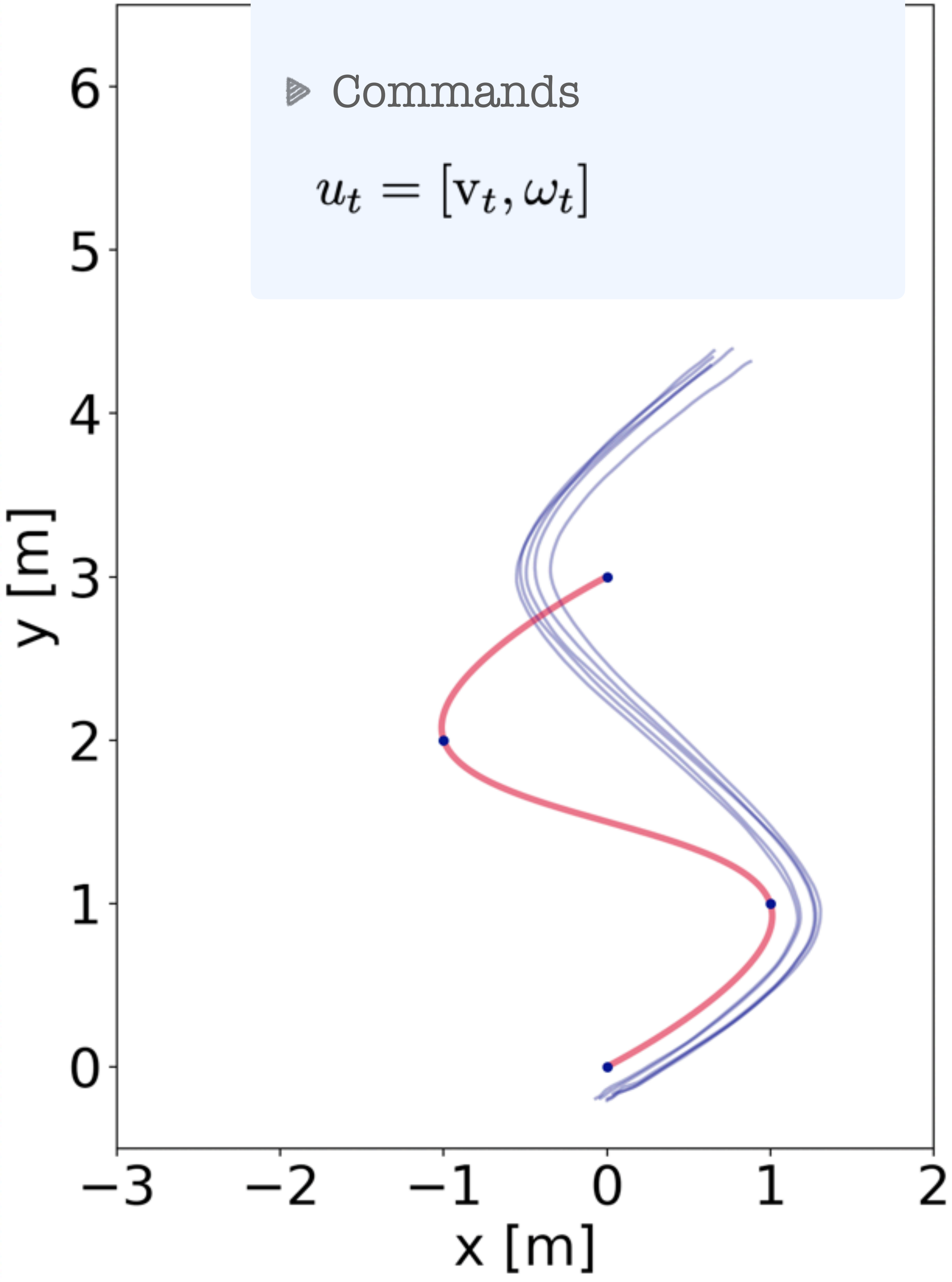
$x_t = [\mathrm{x}_t, \mathrm{y}_t, \theta_t]$ VICON

$\dot{x}_t = [\dot{\mathrm{x}}_t, \dot{\mathrm{y}}_t, \dot{\theta}_t]$ Estimated

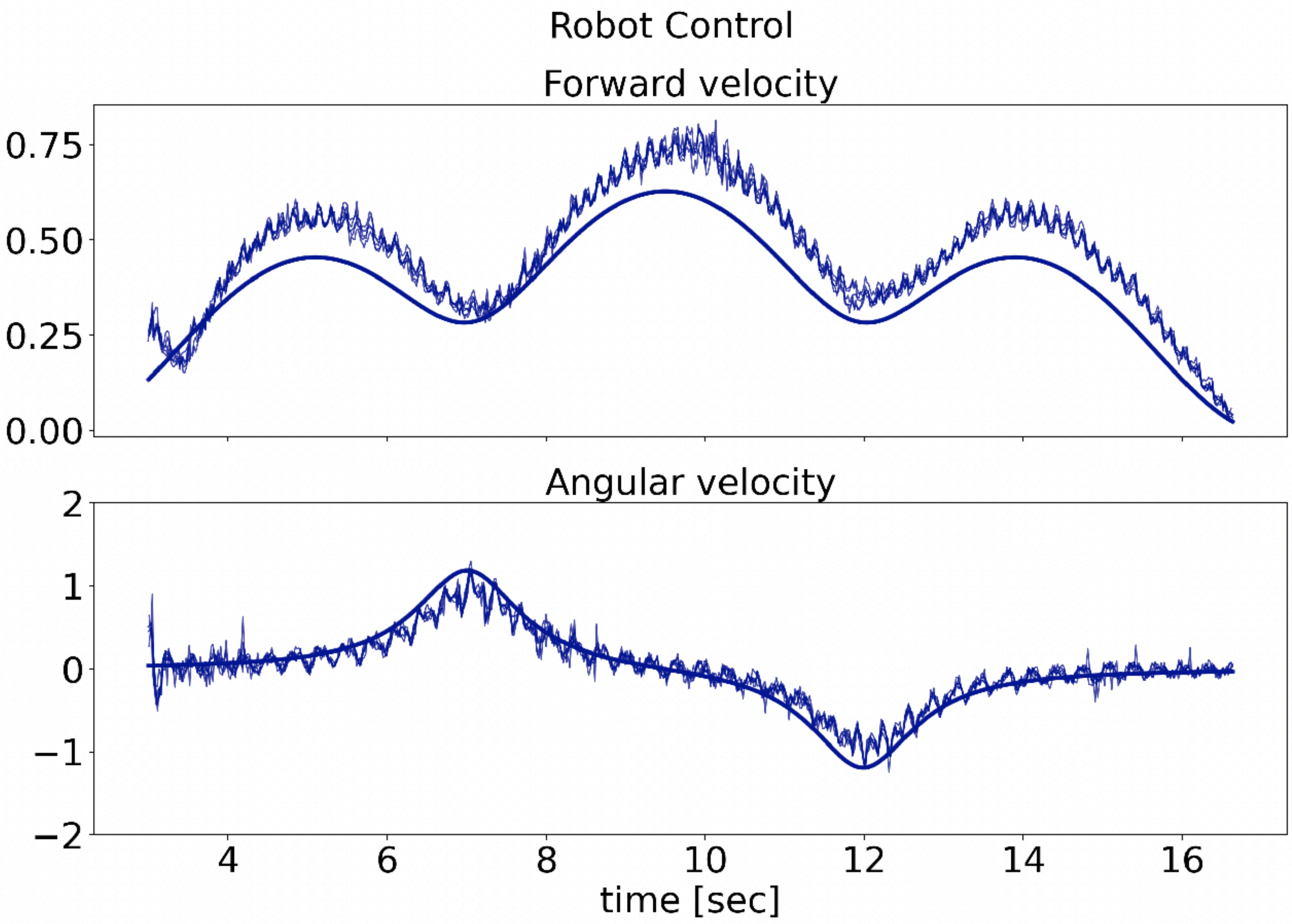
- Commands

$u_t = [\mathrm{v}_t, \omega_t]$


Robot Velocities (Differentiated Vicon signals)



Model training: preliminary results



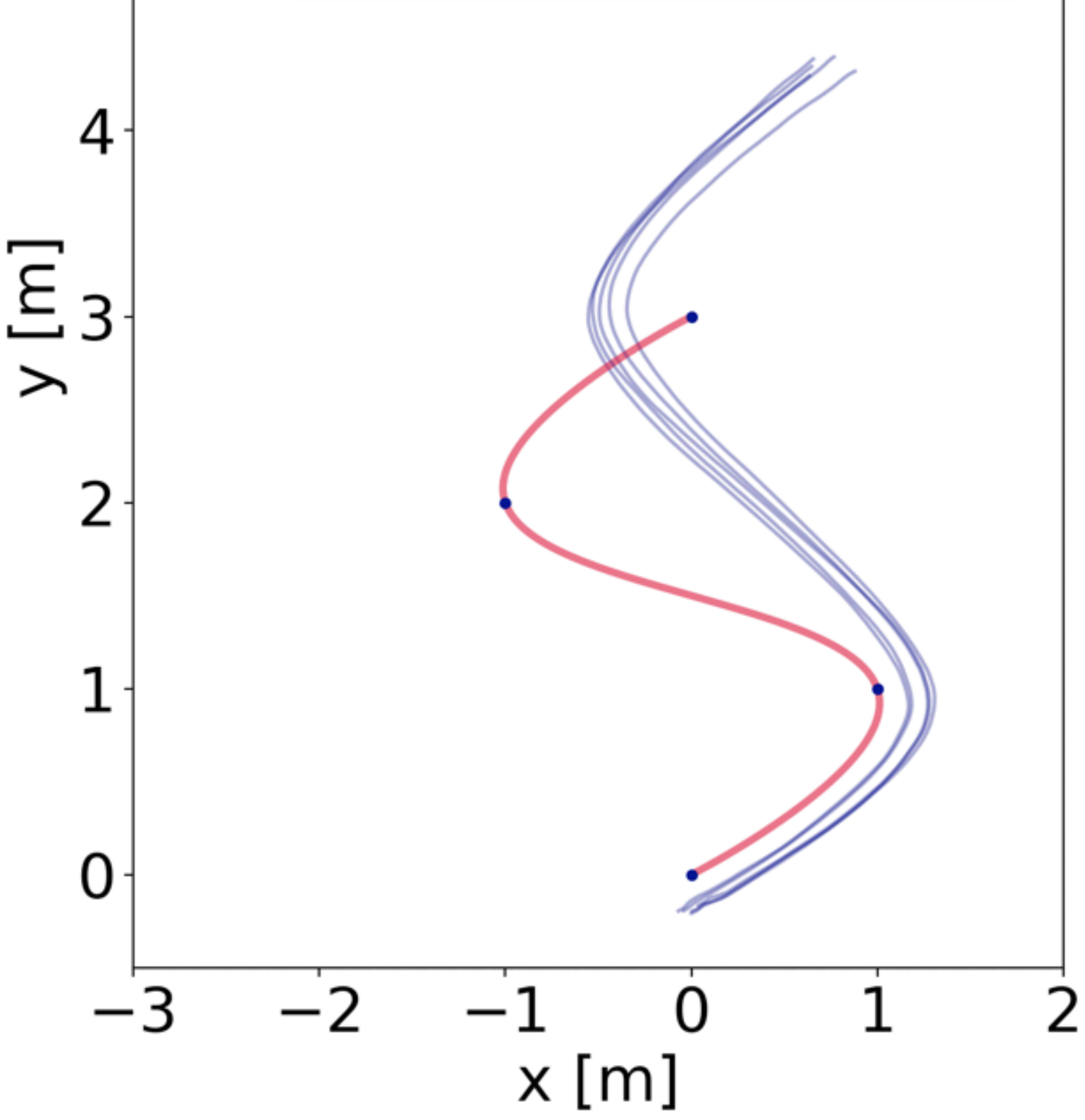
- State

$x_t = [\mathrm{x}_t, \mathrm{y}_t, \theta_t]$ VICON

$\dot{x}_t = [\dot{\mathrm{x}}_t, \dot{\mathrm{y}}_t, \dot{\theta}_t]$ Estimated

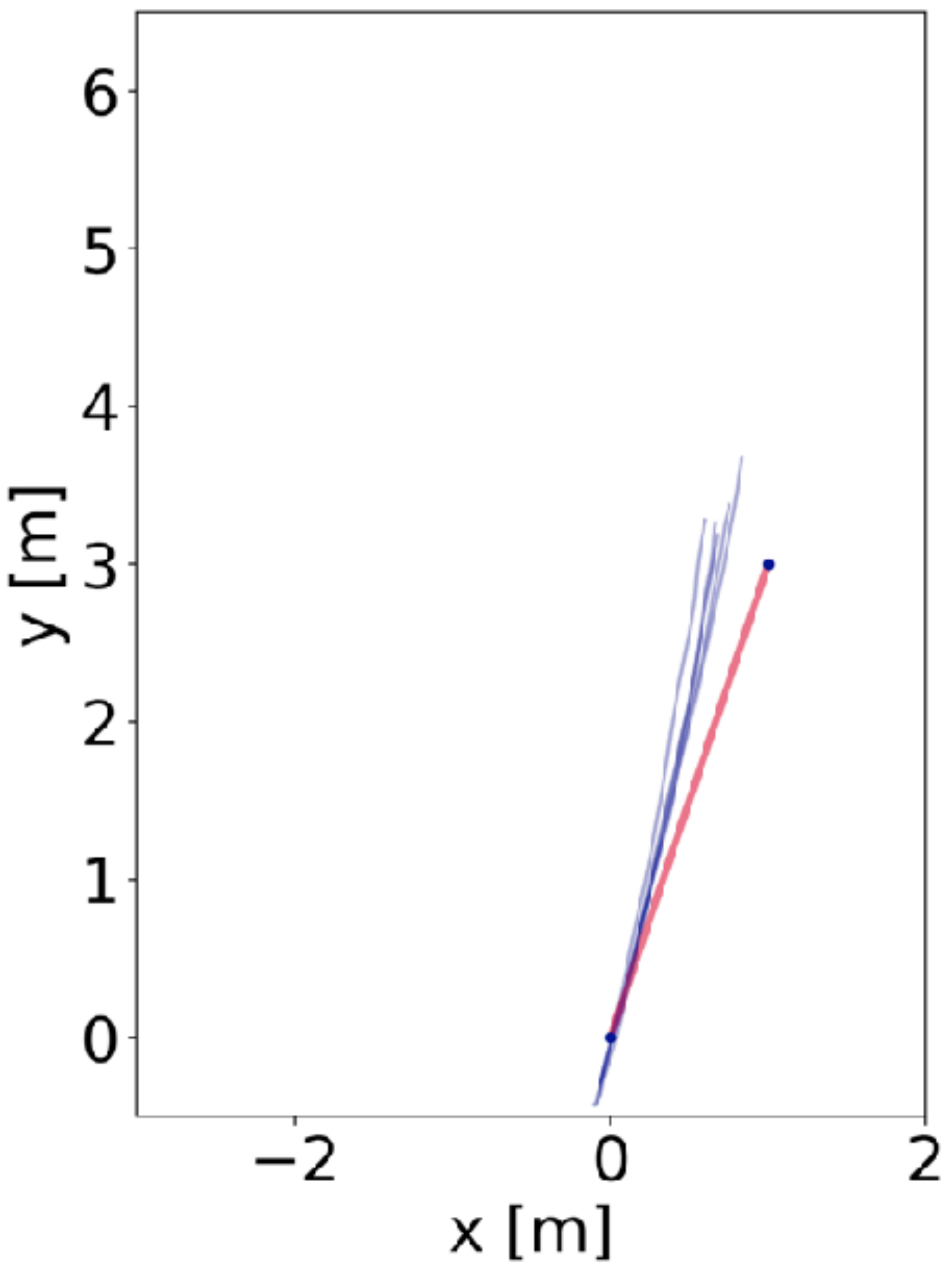
- Commands

$u_t = [\mathrm{v}_t, \omega_t]$

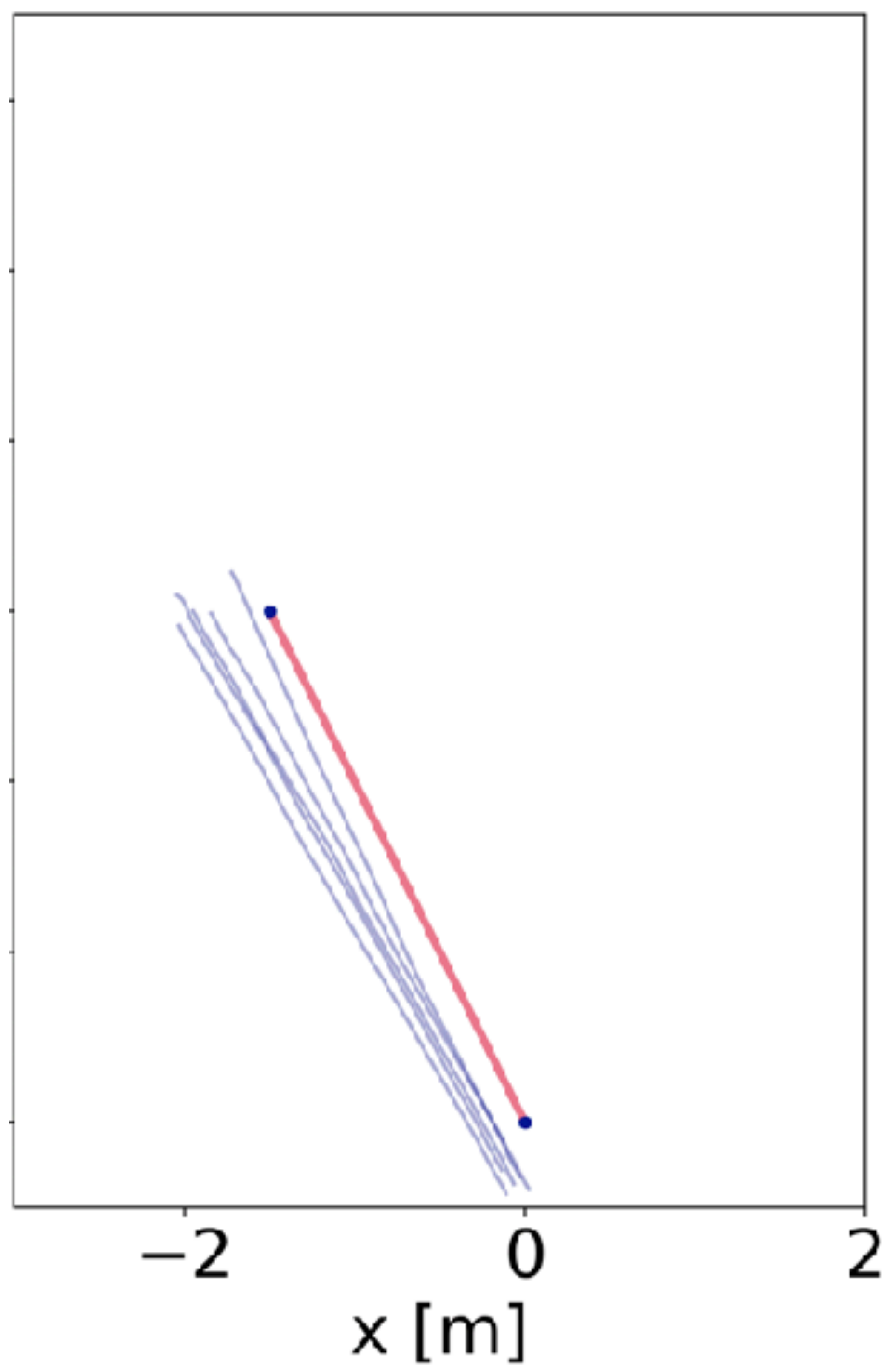


Model training: preliminary results

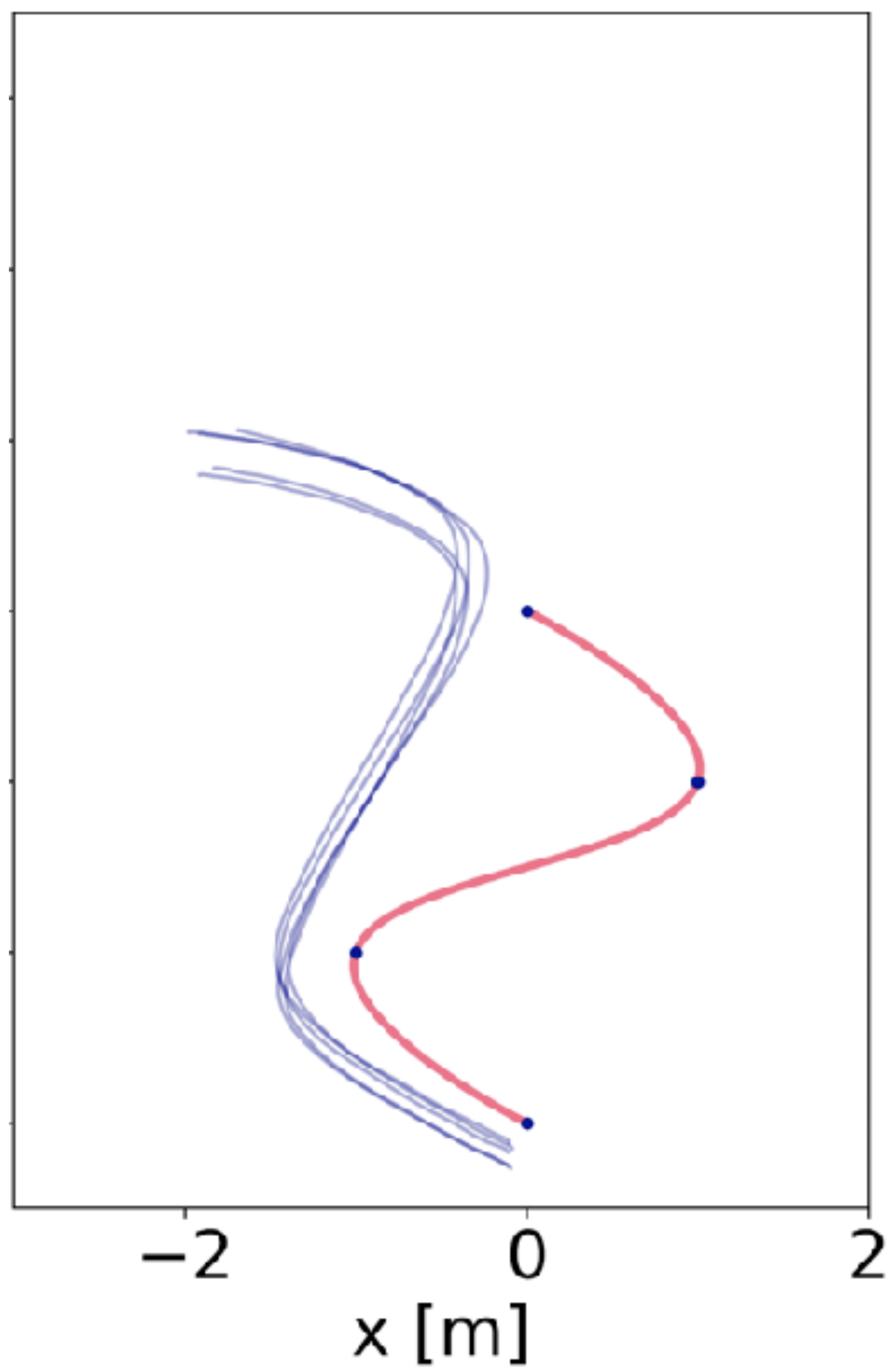
Training



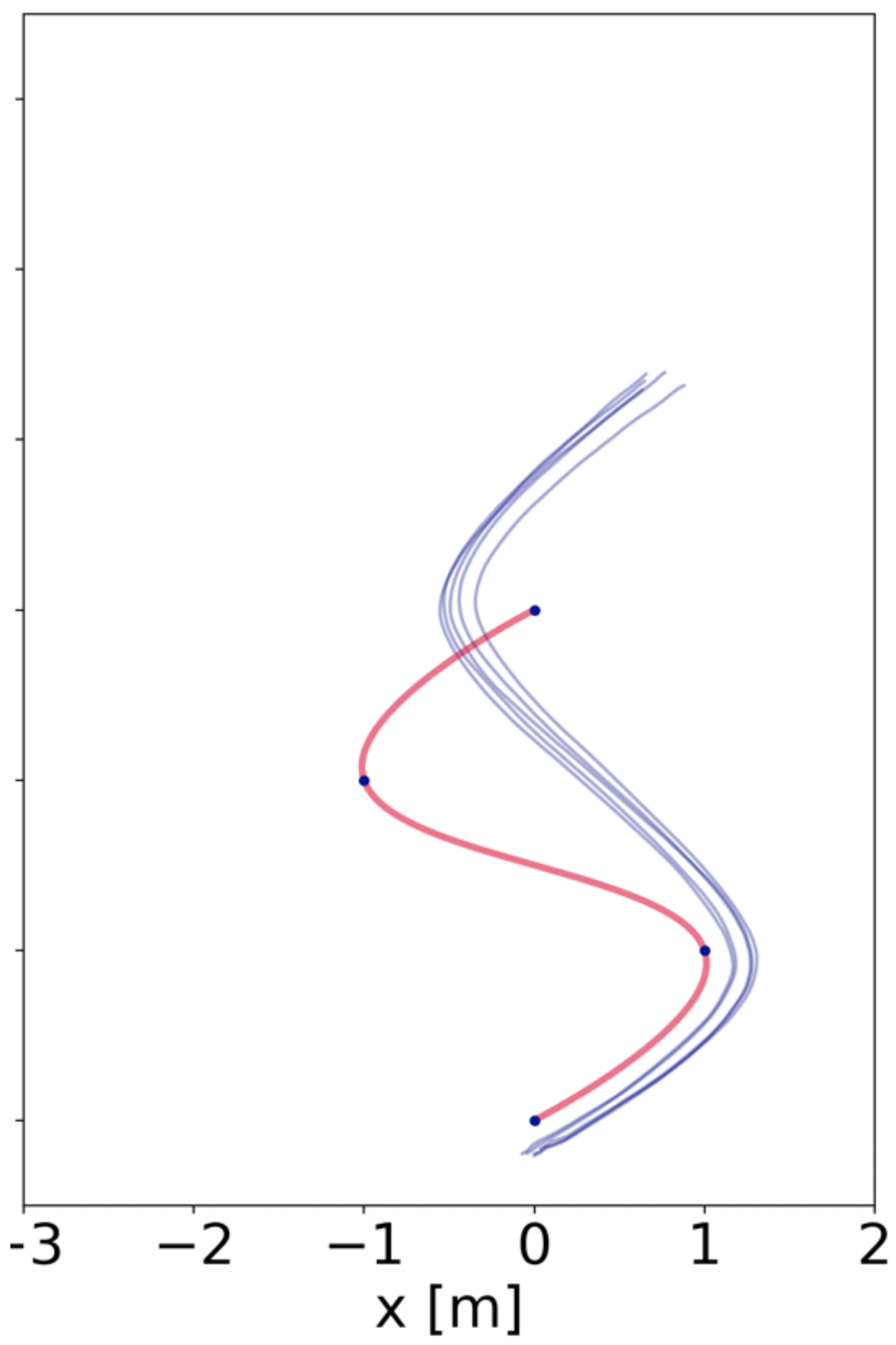
Training



Training



Training/Testing

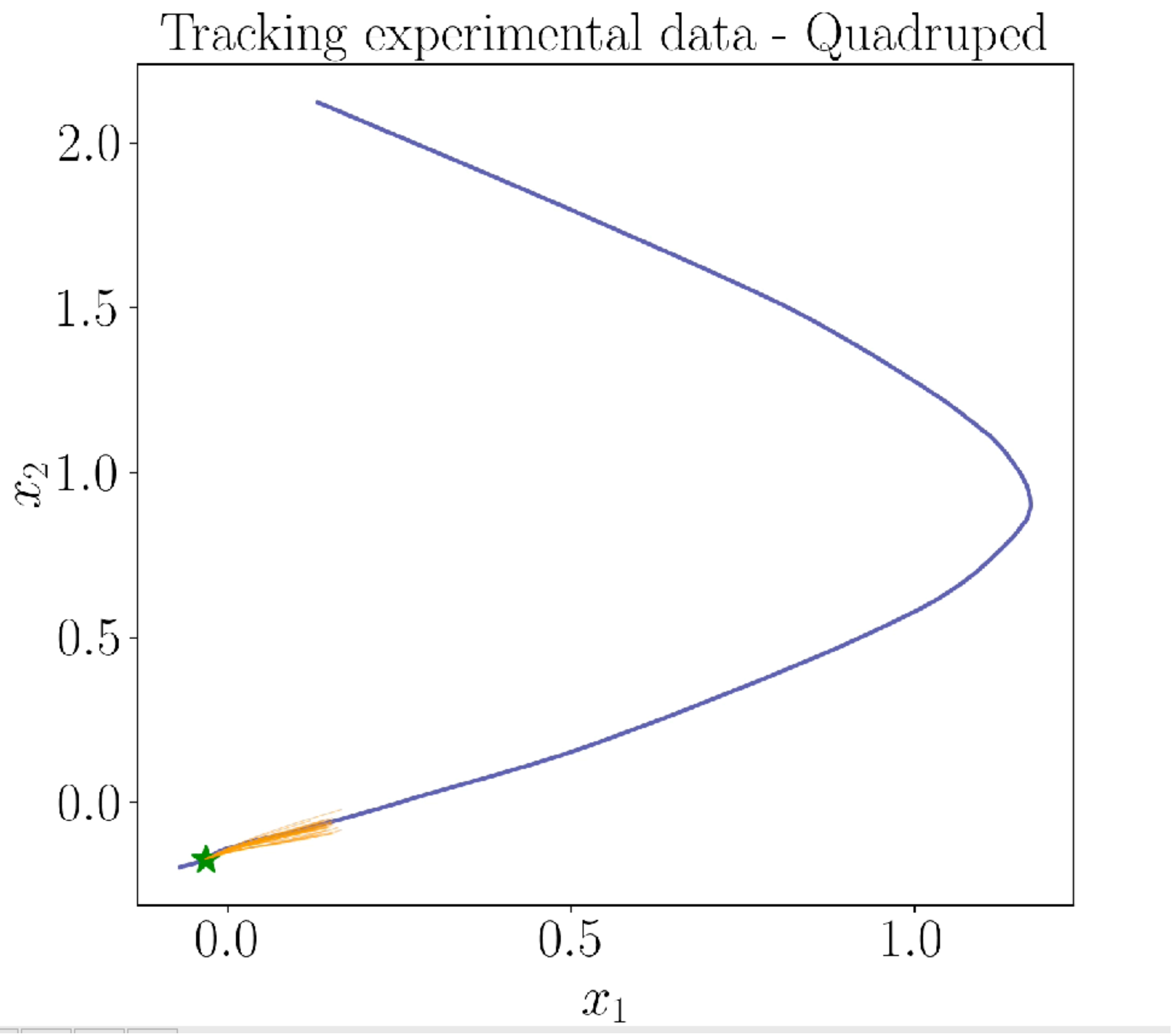


Data snapshots: 100 Hz

Subsampled at 10 Hz

$\Delta T = 100$ ms

Model training: preliminary results



- Prediction horizon $\begin{cases} H = 40 \\ \Delta t_H = 4 \text{ s} \end{cases}$

- Sampling using pre-constructed callable

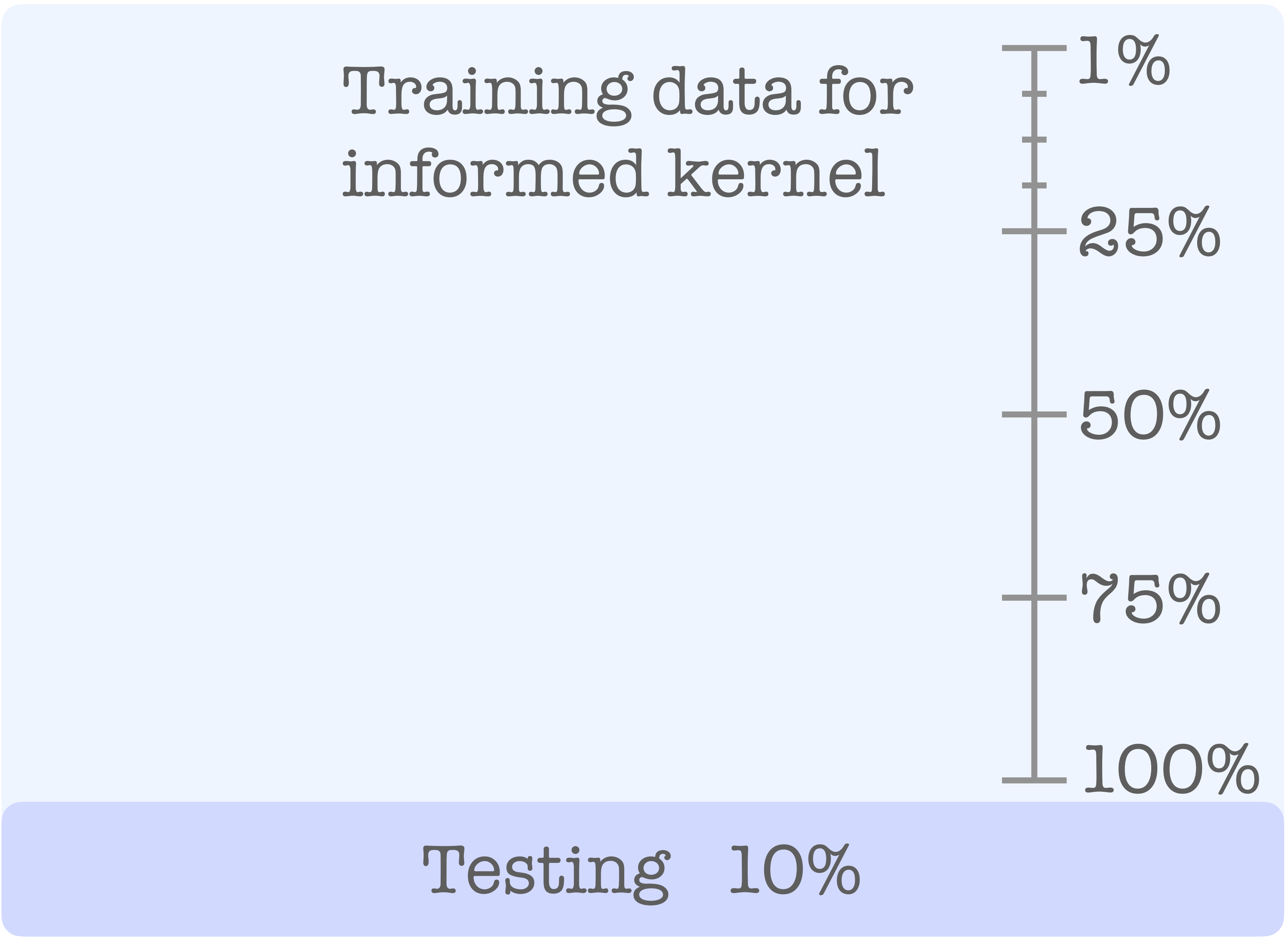
$$\{\hat{x}^r_{t+1}, \dots, \hat{x}^r_{t+H}\} \sim q(x_{t+1:H} | x_t, u_{t:H-1})$$

$$x^{(r)}_{t+1}(x_t, u_t) = f^{(r)}(x_t, u_t) + L_Q \xi$$

Model training: preliminary results

- Do simulation-informed kernels really help to learn faster?

Dataset: real trajectories



Training data for informed kernel

1%

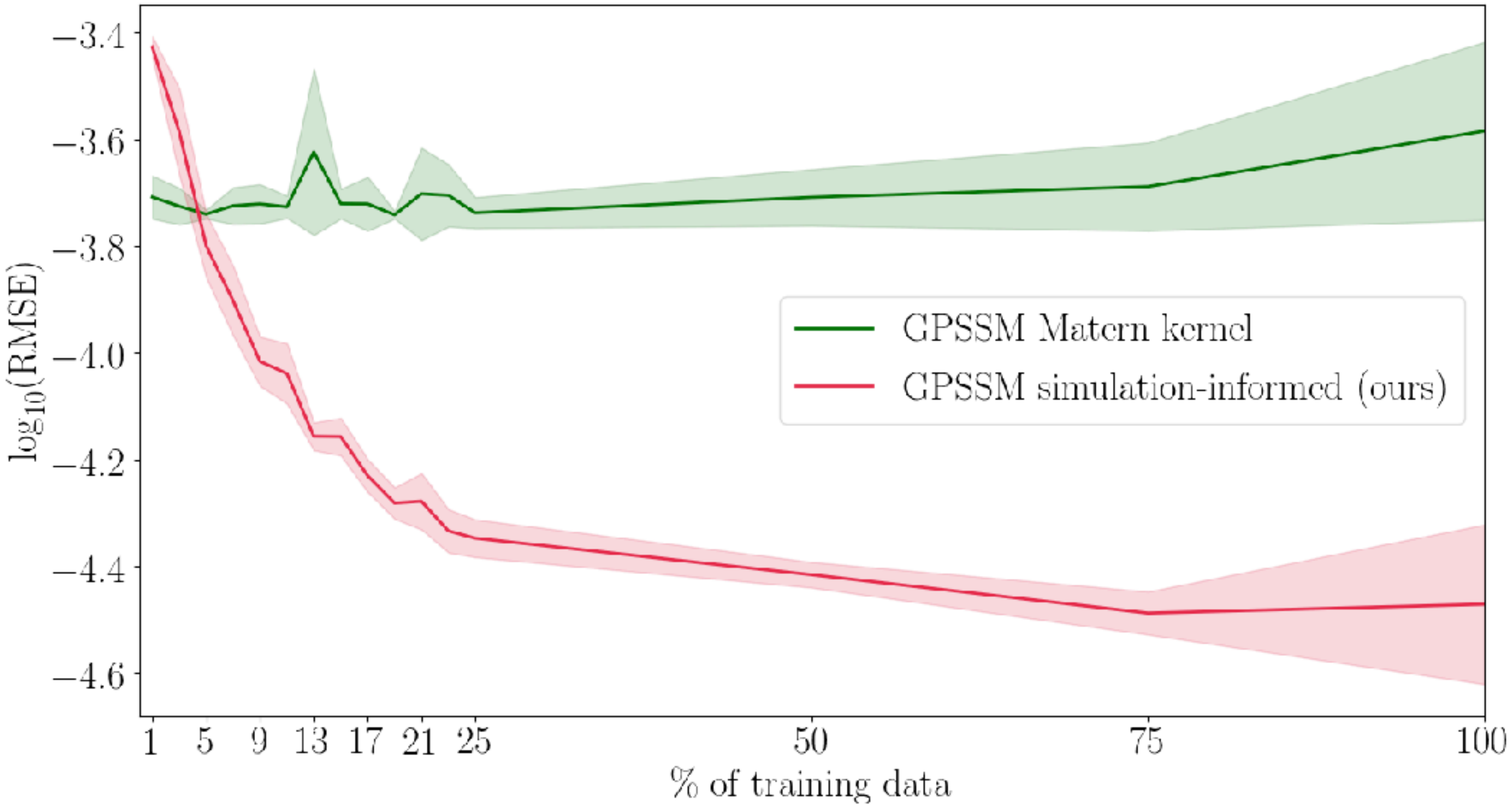
25%

50%

75%

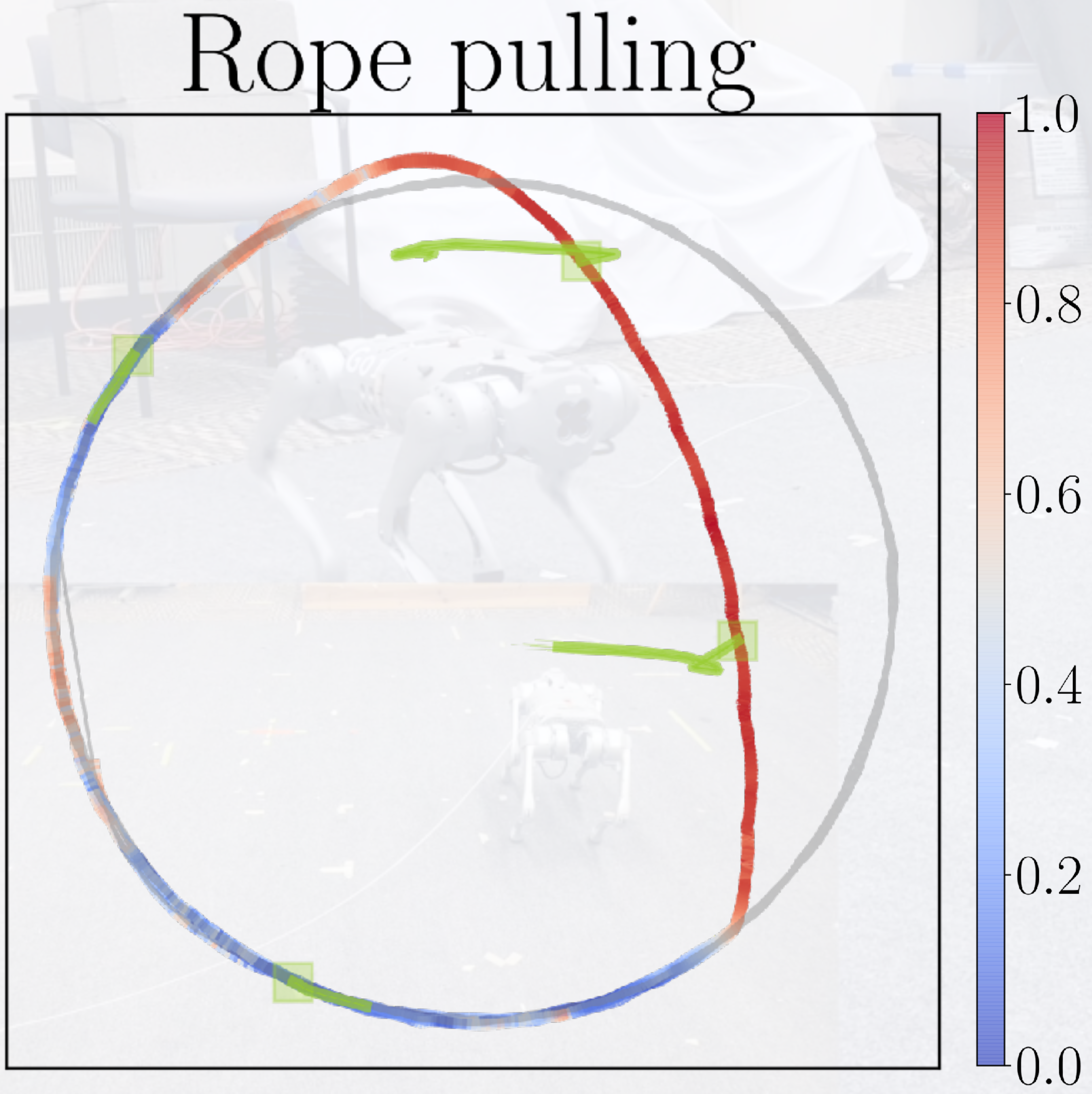
100%

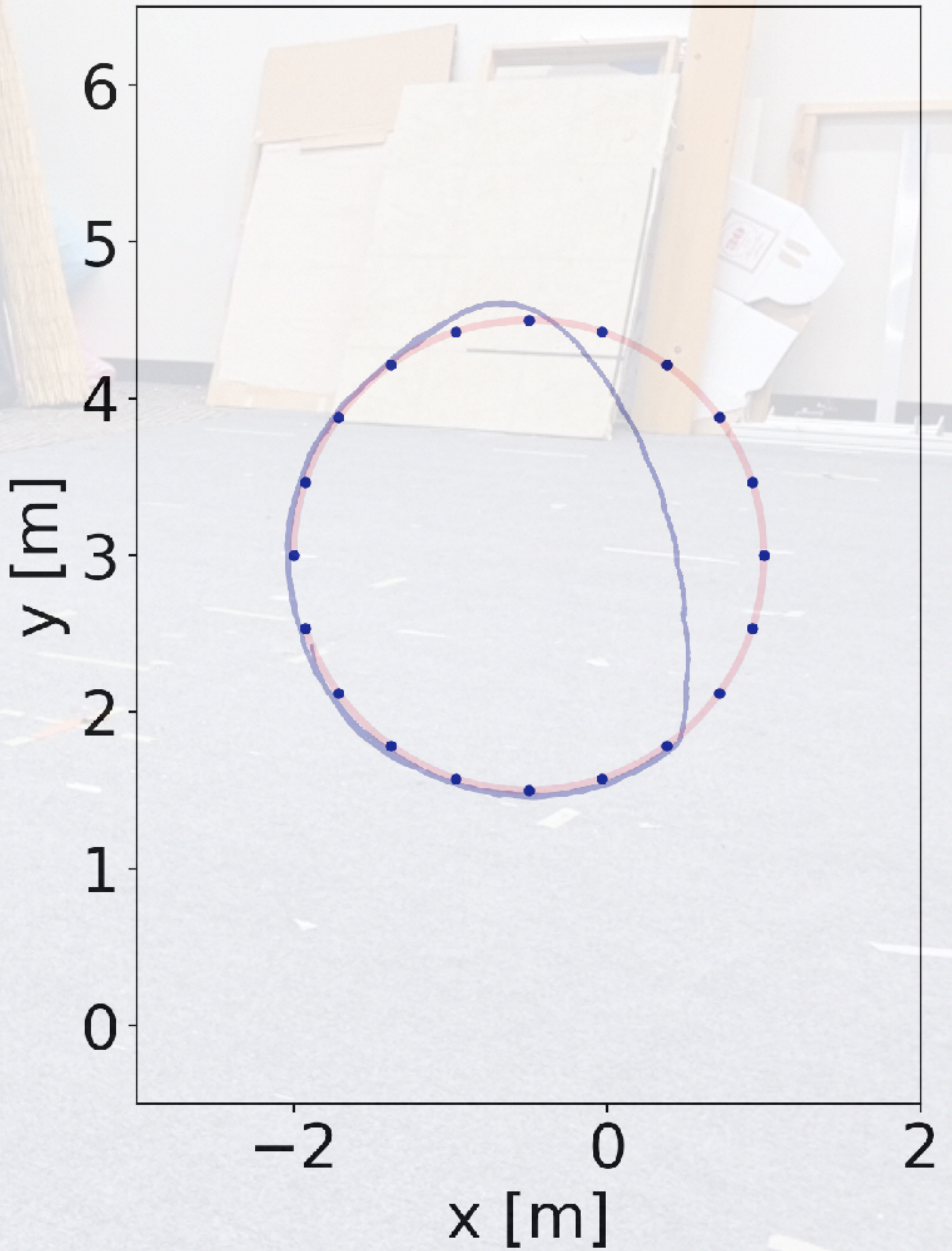
Testing 10%



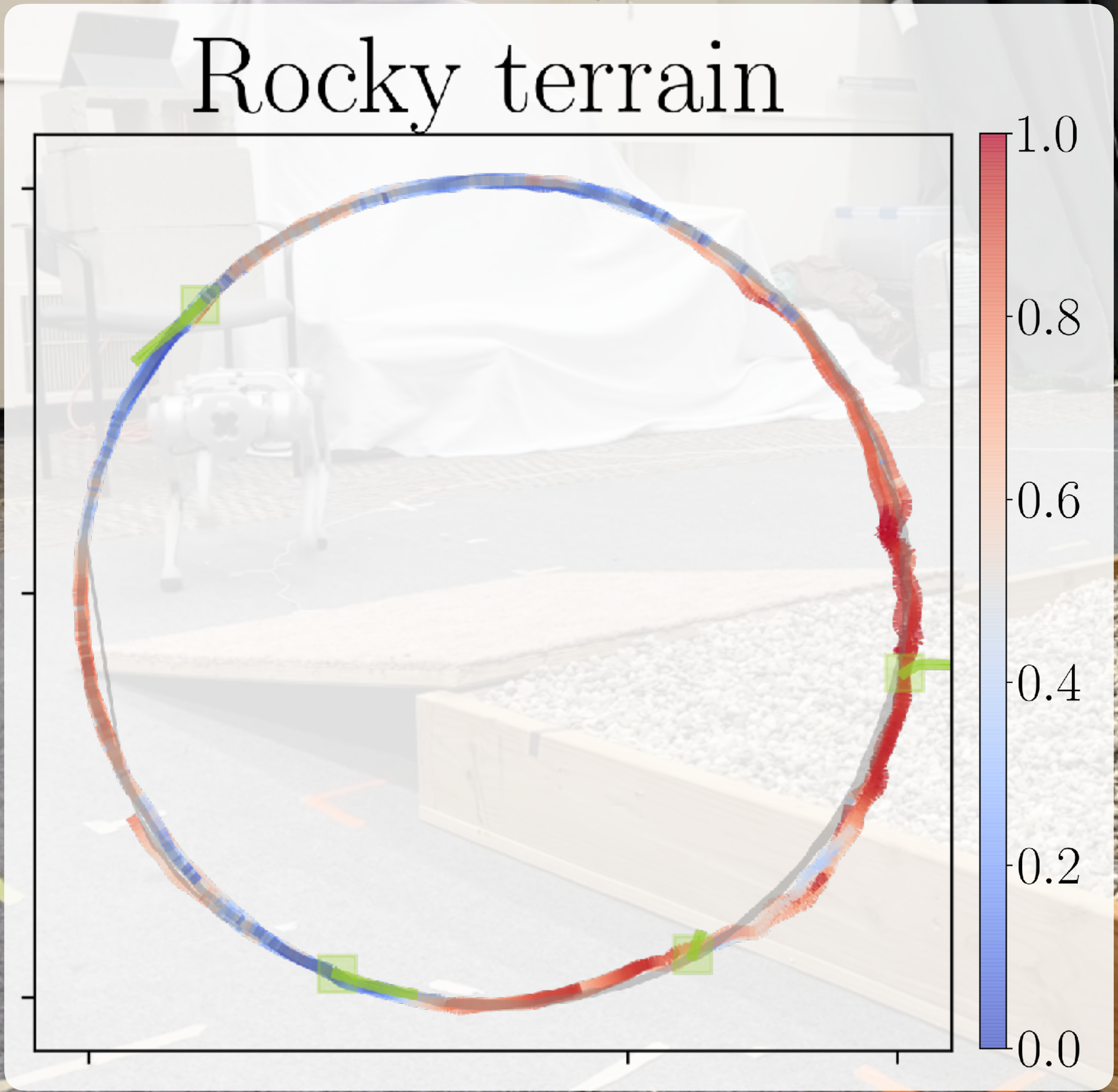
▶ Collect data on flat terrain

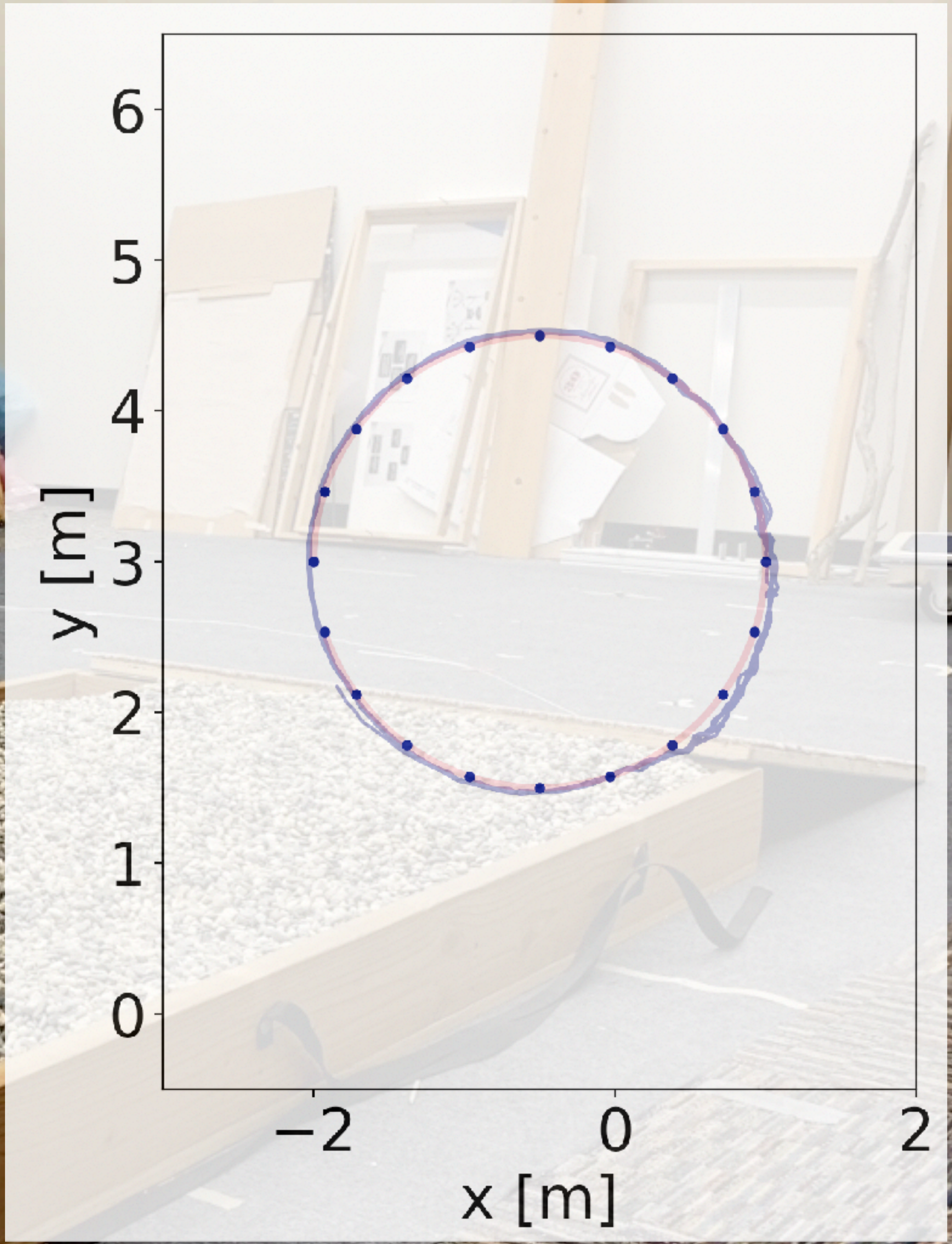
▷ Test OoD: Rope pulling



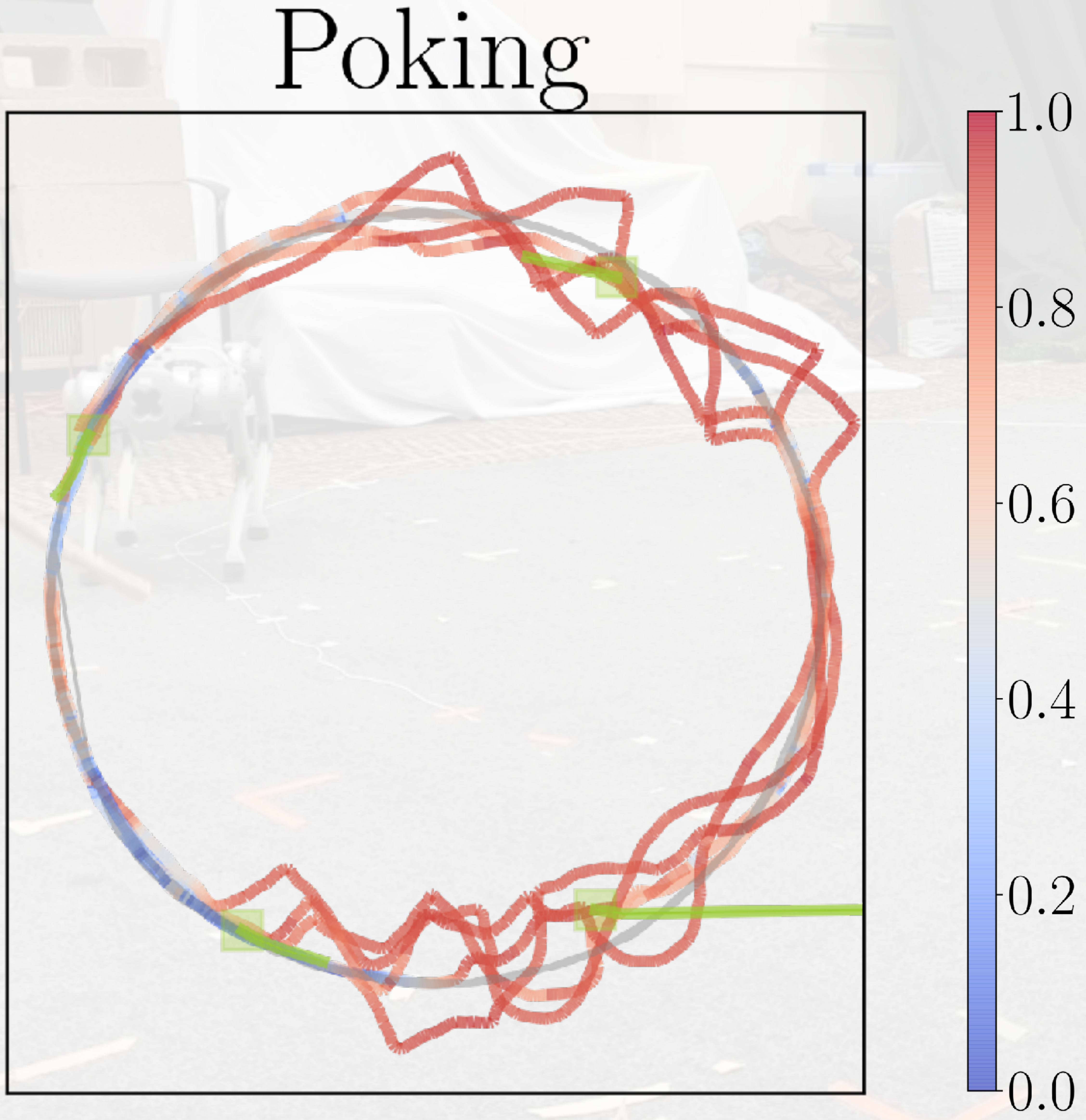


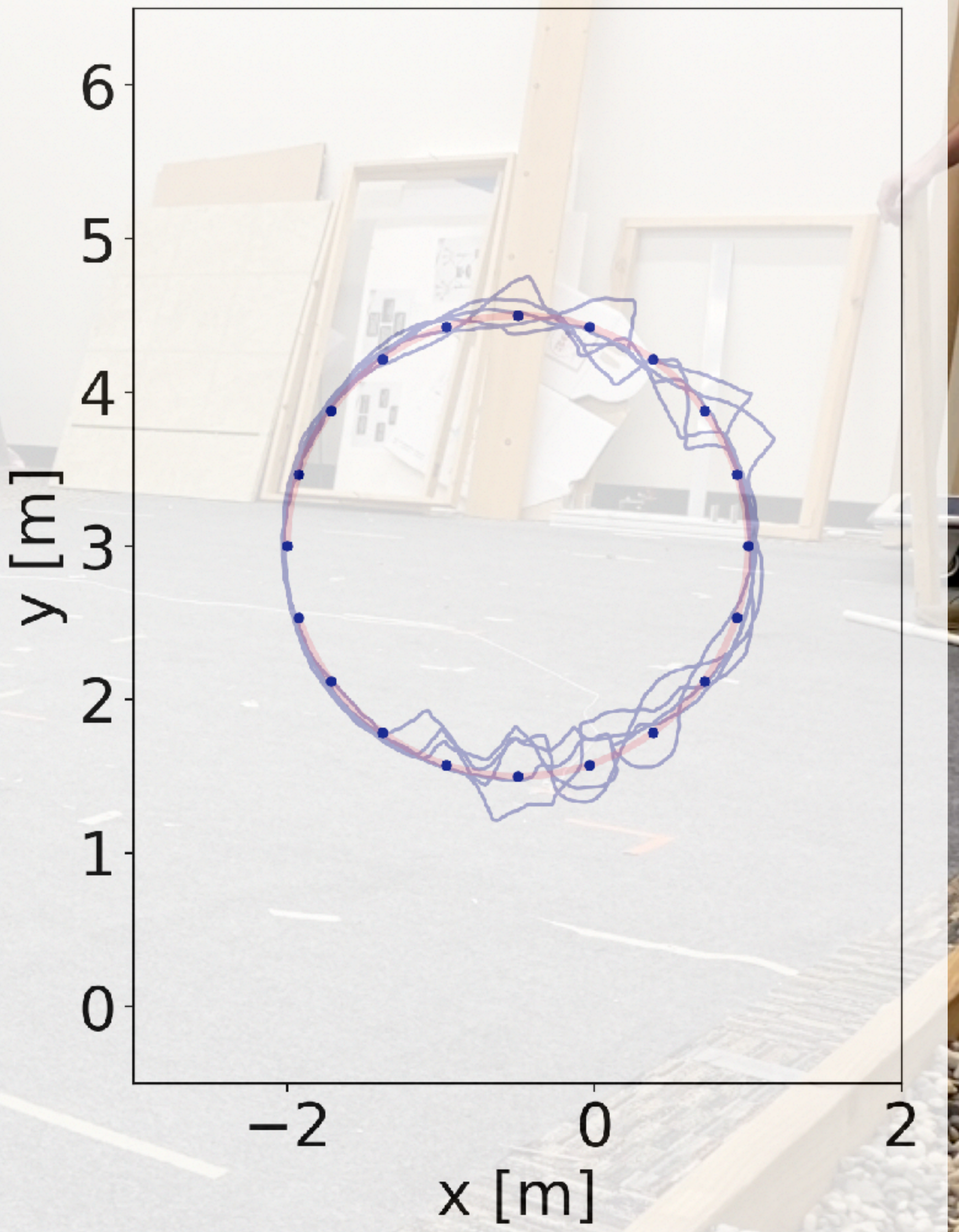
Test OoD: Rocky terrain





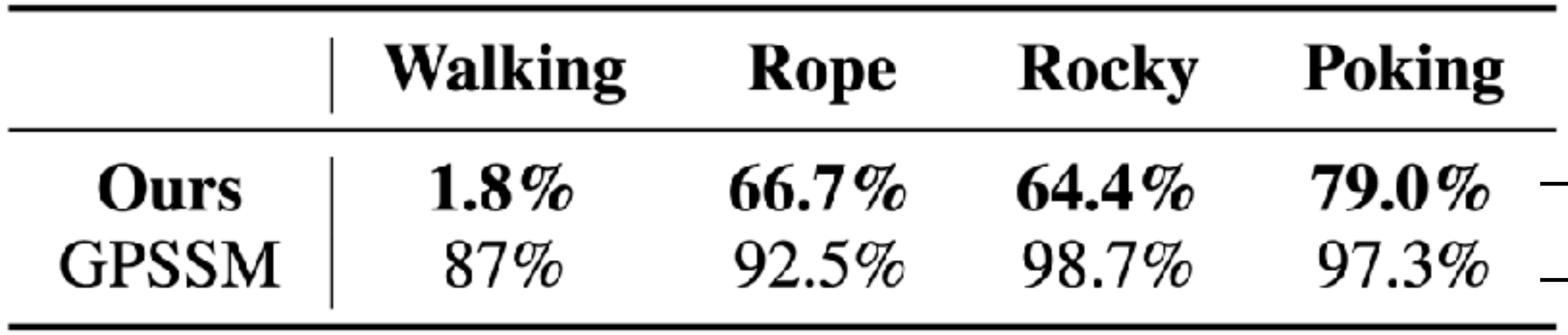
▹ Test OoD: Poking


Poking



Conclusions

- Do simulation-informed kernels help for more accurate OoD detection?



	Walking	**Rope**	**Rocky**	**Poking**
Ours	**1.8%**	**66.7%**	**64.4%**	**79.0%**
GPSSM	87%	92.5%	98.7%	97.3%

→ Informed

→ Not informed

- Novel prediction-based OoD detection method based on GPSSMs
- System-agnostic framework for embedding prior information into the GP kernel
- Sample-efficient medium/long-term predictive model, scalable to higher dimensions
- Code accessible soon

- Future work
    - Use OoD detection for decision-making on-the-fly
    - Integrate state-predictions with stochastic MPC for indoors navigation
    - Integrate non-Gaussian observations, e.g., on-board vision and lidar
    - OoD metric is delayed H steps