NEURAL INFORMATION PROCESSING SYSTEMS

Scan QR code for paper

Manuscript Location: https://arxiv.org/pdf/2502.08736

Recurrent Memory for Online Interdomain Gaussian Processes

Wenlong Chen[1,*], Naoki Kiyohara[1,2,*], Harrison Bo Hua Zhu[3,1,*], Jacob Curran-Sebastian[3], Samir Bhatt[3,1], Yingzhen Li[1]

[1]Imperial College London [2]Canon Inc. [3]University of Copenhagen

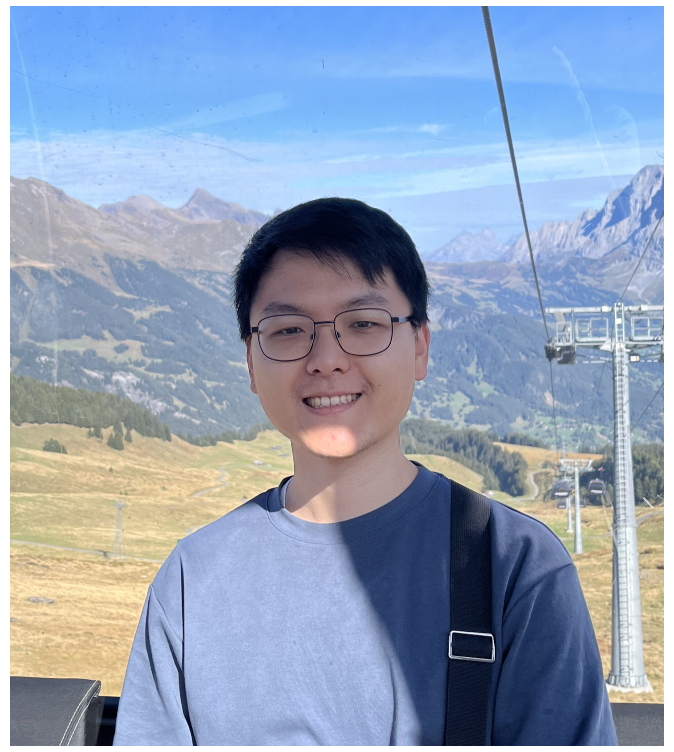
IMPERIAL Canon UNIVERSITY OF COPENHAGEN

Research Team

IMPERIAL Canon UNIVERSITY OF COPENHAGEN

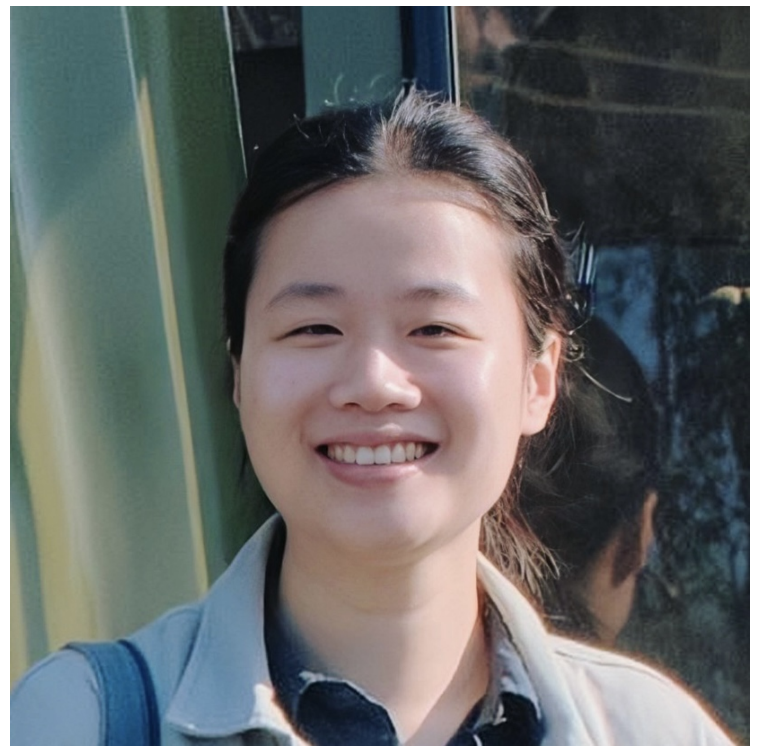

Wenlong Chen*


Naoki Kiyohara*


Harrison Bo Hua Zhu*


Jacob Curran-Sebastian


Samir Bhatt


Yingzhen Li

*Equal Contribution

Motivation

Long-term Memory in Online Learning

- Many real-world tasks involve streaming data (e.g., sensor feeds, sequential measurements, climate data).
- We need **an online method that**
  - **incrementally updates** rather than requiring re-training from scratch,
  - and **have capability of modeling uncertainty.**
- **Regression Tasks:** $y_t \sim p(y_t \mid f(t)), \quad f \sim$ Function Prior

The Gaussian processes (GPs) are elegant solutions, but…

Why is this hard?

(1) We Can't Keep All the Data

- Computational costs for exact GP inference scales as $\mathcal{O}(n^3)$
  → **infeasible for large streaming data.**

(2) Sparse Approximation Can Lead to Forgetting

- Typical sparse GPs rely on a small set of M inducing points to compress the data distribution to give $\mathcal{O}(M^3 + nM^2)$

- As new data arrive, inducing points shift, causing older ***memory*** to be lost.

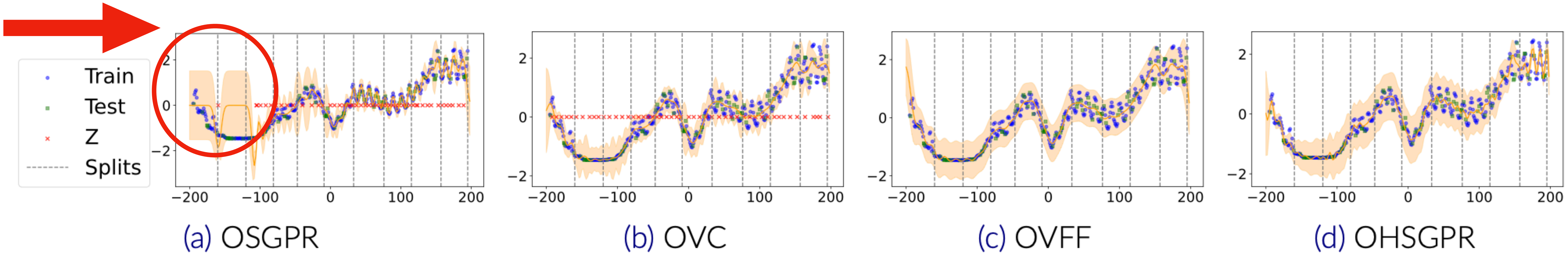


Figure 1. Predictive mean ±2 standard deviation of OSGPR, OVC, OVFF, and OHSGPR after task 10 of the Solar dataset. M = 50 inducing variables are used.

Background - Sparse Variational Gaussian Processes

- **Classical GP:** With n datapoints $(x_i, y_i)_{i=1}^n$,

$$f \sim GP(0,k) \Rightarrow \text{Prior: } \mathbf{f} \sim N(0, \mathbf{K_{ff}})$$

$$\text{Posterior: } f(x_*) \,|\, \mathbf{f} \sim N(\mathbf{K_{*f}}\mathbf{K_{ff}^{-1}}\mathbf{y}, \mathbf{K_{**}} - \mathbf{K_{*f}}\mathbf{K_{ff}^{-1}}\mathbf{K_{f*}})$$

$\mathcal{O}(n^3)$

- **Sparse Variational Gaussian Processes (SGPR, Titsias 2009; SVGP Hensman et al. 2013 & 2015):**

$$f \sim GP(0,k) \Rightarrow \text{Prior: } \mathbf{f} \sim N(0, \mathbf{K_{ff}})$$

Inducing points

Inducing Variables: $\mathbf{u} = f(\mathbf{Z}) \sim N(0, \mathbf{K_{uu}})$

$\mathcal{O}(n_* M^2 + M^3)$

$$\text{Conditioning: } f(x_*) \,|\, \mathbf{u} \sim N(\mathbf{K_{*u}}\mathbf{K_{uu}^{-1}}\mathbf{u}, \mathbf{K_{**}} - \mathbf{K_{*u}}\mathbf{K_{uu}^{-1}}\mathbf{K_{u*}})$$

Free Form: $N(\mathbf{m}, \mathbf{S})$

$$\text{Approx Posterior: } q(f(x_*)) = \int p(f(x_* \,|\, \mathbf{u})q(\mathbf{u}) \, \mathrm{d}\mathbf{u}$$

- **Optimise $\mathbf{m}, \mathbf{S}, \mathbf{Z}$ and other model hyperparameters:**

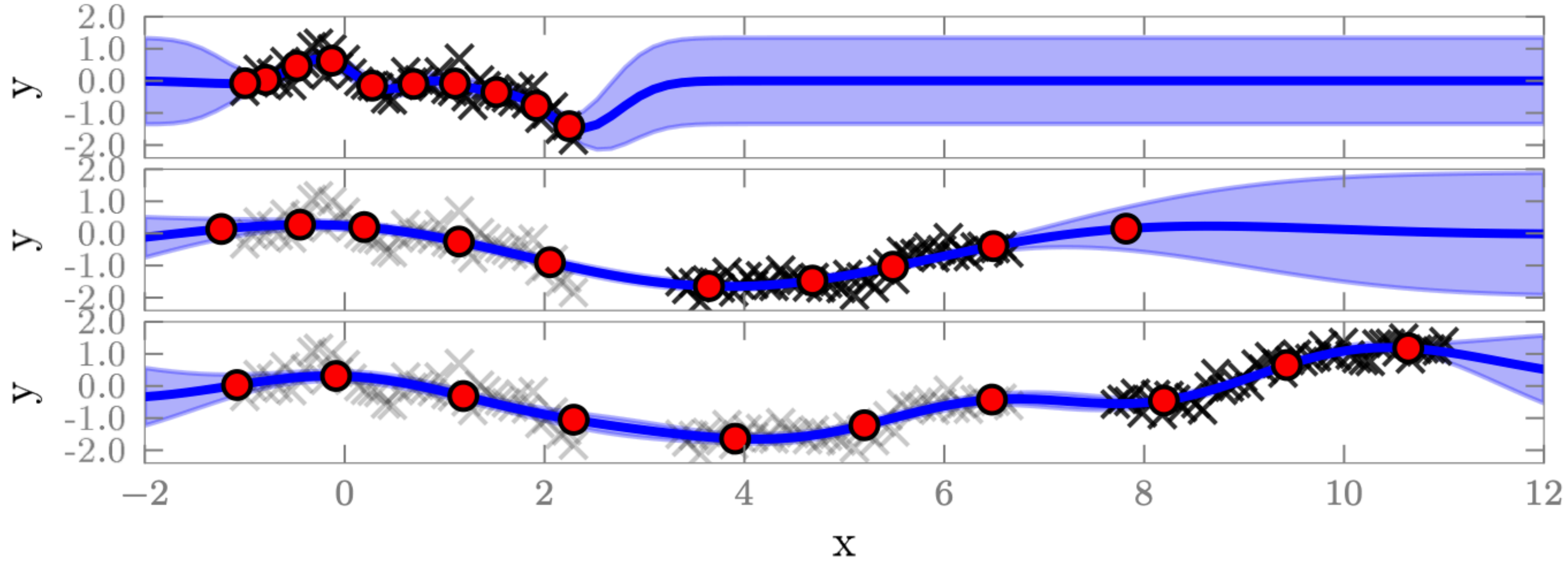
KL(Approx Posterior||True Posterior) $\Rightarrow$ Tractable objective function + optimal $\mathbf{m}, \mathbf{S}$
(when we use Gaussian likelihood)

Titsias, M. K. (2009). Variational learning of inducing variables in sparse Gaussian processes. In AISTATS.
Hensman, J., Fusi, N., & Lawrence, N. D. (2013). Gaussian processes for bigdata. In UAI.
Hensman, J., Matthews, A. G. d. G. T., Filippone, M., & Ghahramani, Z. (2015). Scalable variational Gaussian process classification. In AISTATS.

Background - Online SGPR

(Bui et al., NIPS 2017)

- **Process incoming data in small batches or one sample at a time.**
- **Keep the previous approximate posterior** $q_{\text{old}}(\mathbf{u}) = \mathcal{N}\left(\mathbf{m}_{\text{old}}, \mathbf{S}_{\text{old}}\right)$ and **update it** as new data arrives.
- The update acts as a **"correction" to the prior distribution**, so **storing all past data is unnecessary**.
- **Maximise the incremental ELBO** to refine the approximation with each new batch.



$$\sum_{i=1}^{n_2} \mathbb{E}_{q_2(f_i)} \left[\log p_{t_2}\left(y_i \middle| f_i\right)\right] - \mathrm{KL}\left[q_{t_2}\left(\mathbf{u}_{t_2}\right) \middle\| p_{t_2}\left(\mathbf{u}_{t_2}\right)\right] + \mathrm{KL}\left[\tilde{q}_{t_2}\left(\mathbf{u}_{t_1}\right) \middle\| p_{t_1}\left(\mathbf{u}_{t_1}\right)\right] - \mathrm{KL}\left[\tilde{q}_{t_2}\left(\mathbf{u}_{t_1}\right) \middle\| q_{t_1}\left(\mathbf{u}_{t_1}\right)\right]$$

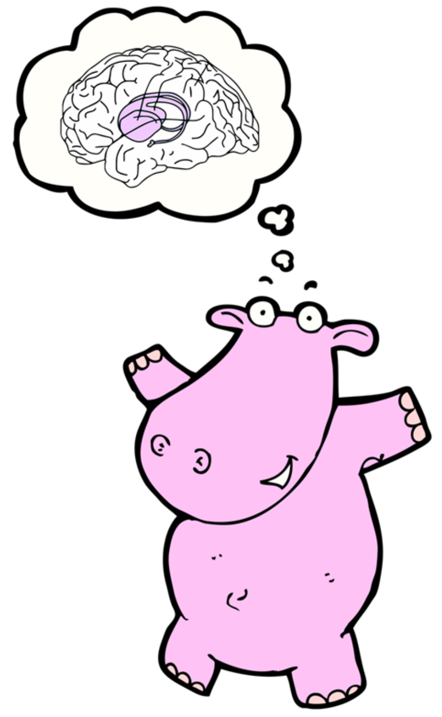
ELBO for the new data **Incorporation of the previous posterior**

This approach may still lose the old historical information -> insufficient inducing points, correction term only based on previous time task

Thang D. Bui, Cuong V. Nguyen, and Richard E. Turner (2017). Streaming sparse Gaussian process approximations. In Advances in Neural Information Processing Systems

Our approach

Leveraging HiPPO & Interdomain GPs



High Order Polynomial Projection Operators (HiPPO) (A. Gu et al., NeurIPS 2020)

Core Idea: Maintains an online representation of a time series by projecting onto polynomial bases with online manner.

Key Benefit: Updates the coefficient vector in O(1) per time step, enabling efficient long-term memory.

Interdomain Gaussian Processes (M. Lázaro-Gredilla & A. Figueiras-Vidal NIPS 2009)

Core Idea: Inducing variables are located in the **transformed function space** using an integral: $u_m = \int f(x)\phi_m(x)\mathrm{d}x$, where ϕ_m are basis functions.

By extending HiPPO's input from a deterministic signal to a Gaussian process, we naturally arrive at an interdomain formulation.

Obtaining sequential data representation

A Orthogonal Polynomial Expansion Approach

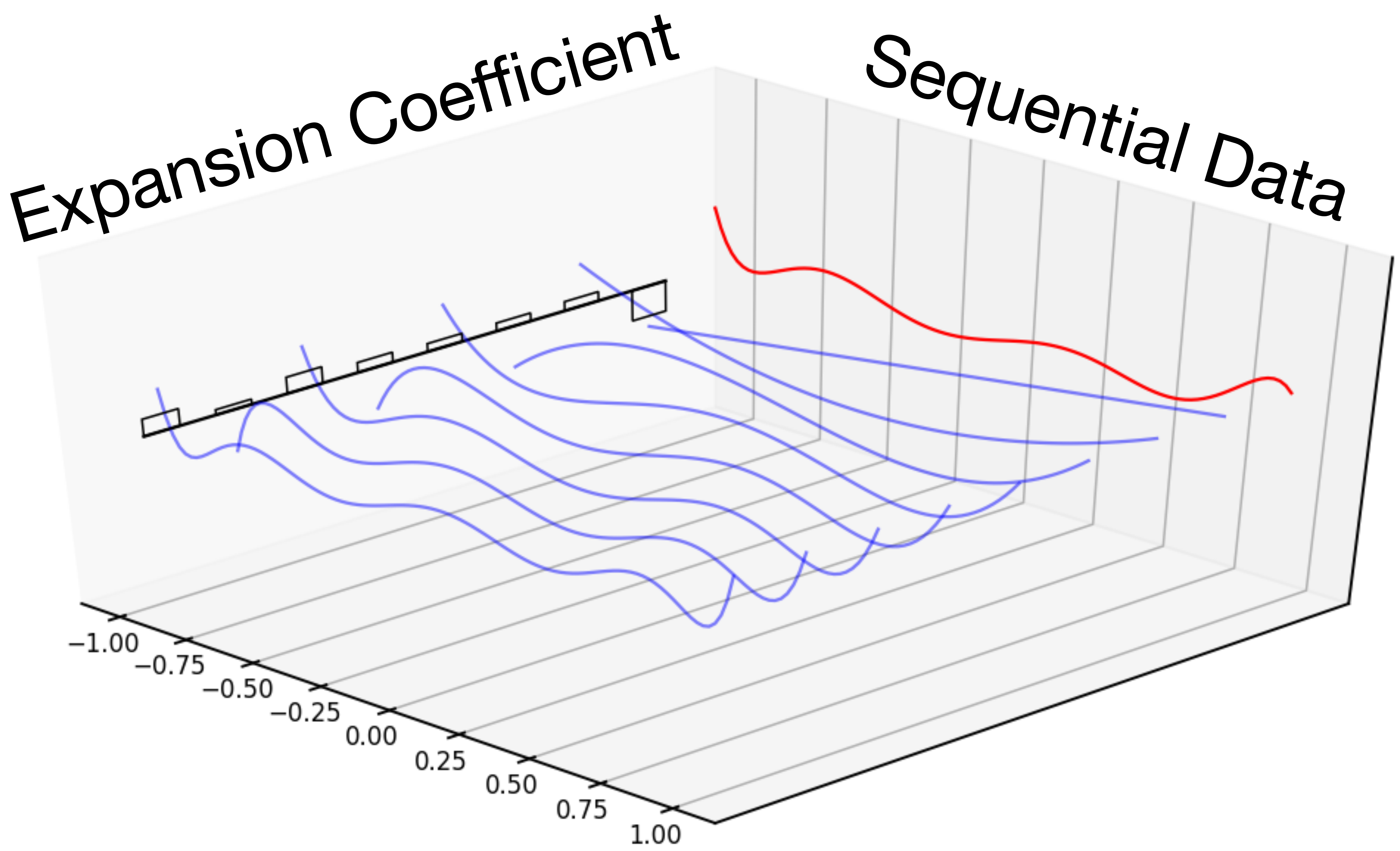


Fig. from blog post “Annotated S4”

Legendre polynomial $x \in [-1,1]$

$$f(x) = \sum_{n=0}^{\infty} u_n P_n(x), P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} \left(x^2 - 1\right)^n$$

Fourier polynomial $x \in [0,T]$

$$f(x) = \sum_{n=-\infty}^{\infty} u_n P_n(x), P_n(x) = \exp\left(\frac{2\pi i n x}{T}\right)$$

Chebyshev polynomial $x \in [-1,1]$

$$f(x) = \sum_{n=0}^{\infty} u_n P_n(x), P_n(x) = \cos(n \arccos(x))$$

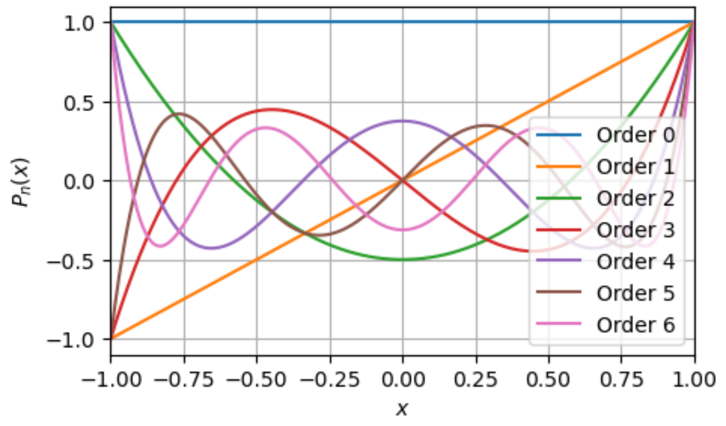
We can obtain <u>coefficients</u> via inner product $u_n \propto \langle f, P_n \rangle$,

BUT we need a more efficient, sequential approach for sequential process.

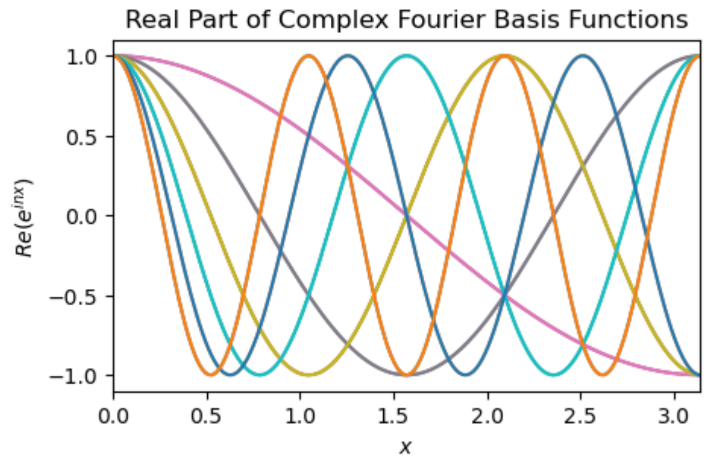
Sequential update method for polynomial coefficients

Selecting a Polynomial Basis

- Legendre

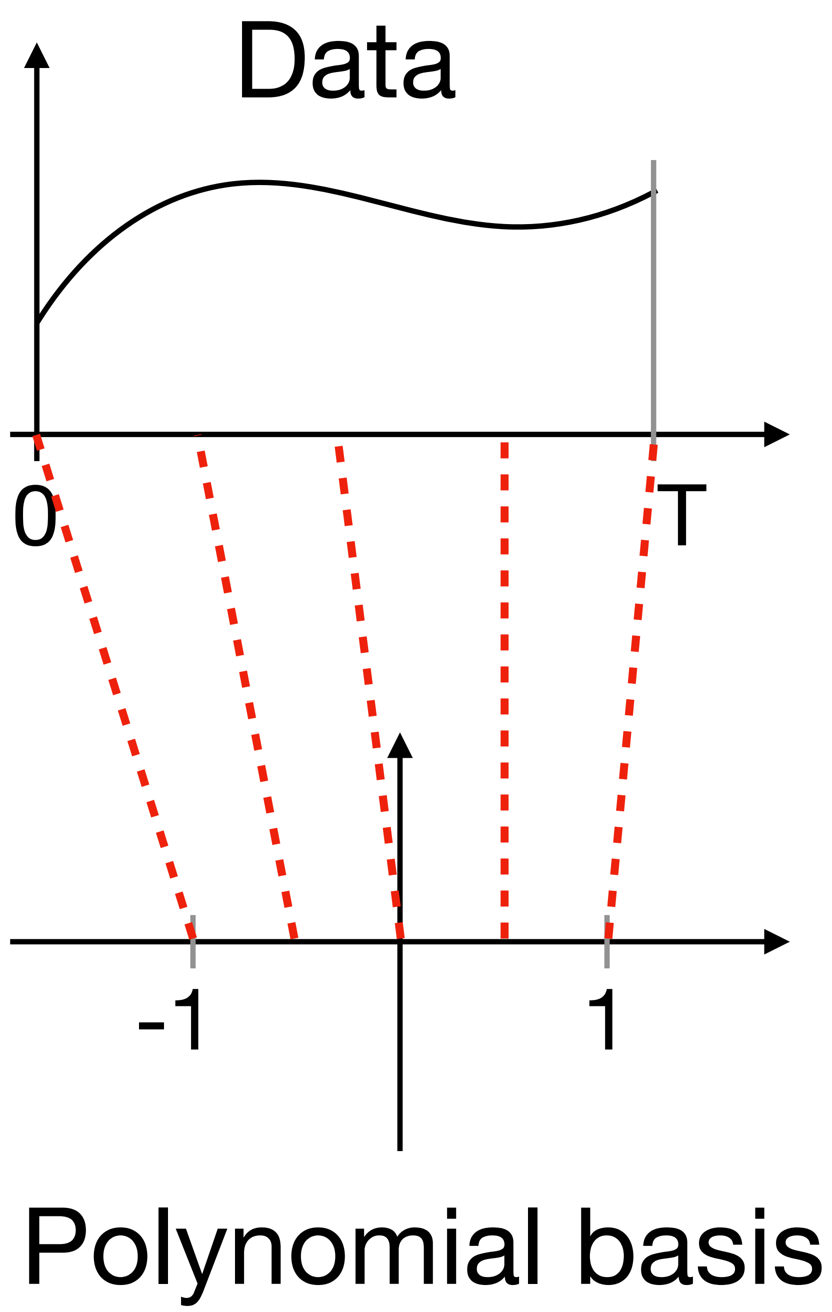


- Fourier



etc.

Rescaling to Target Range



Formulating Coefficient Dynamics

$$f(x,t) \simeq \sum_{n=0}^{N-1} c_n(t) P_n(x,t)$$

$$u_n \propto \langle f_{\le t}, P_n \rangle$$

$$\mathbf{u}(t) = \begin{pmatrix} u_0(t) \\ u_1(t) \\ \vdots \\ u_{N-1}(t) \end{pmatrix}$$

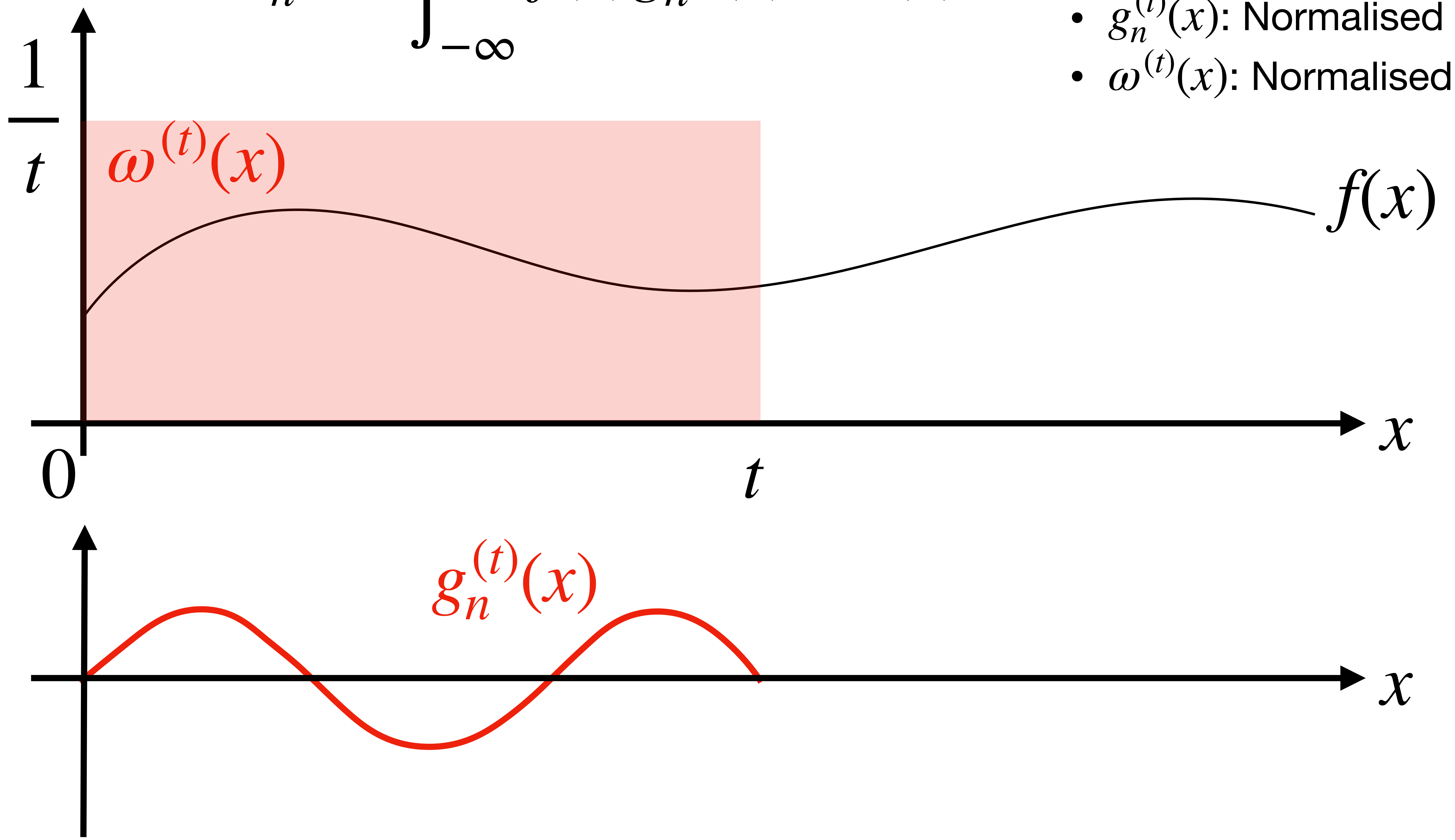
$$\frac{d}{dt}\mathbf{u}(t) = \dots$$

$$\simeq A(t)\mathbf{u}(t) + B(t)f(t)$$

Intuitive Visualisation of the Problem Setting

$$u_n^{(t)} = \int_{-\infty}^{\infty} f(x) g_n^{(t)}(x) \omega^{(t)}(x) dx$$

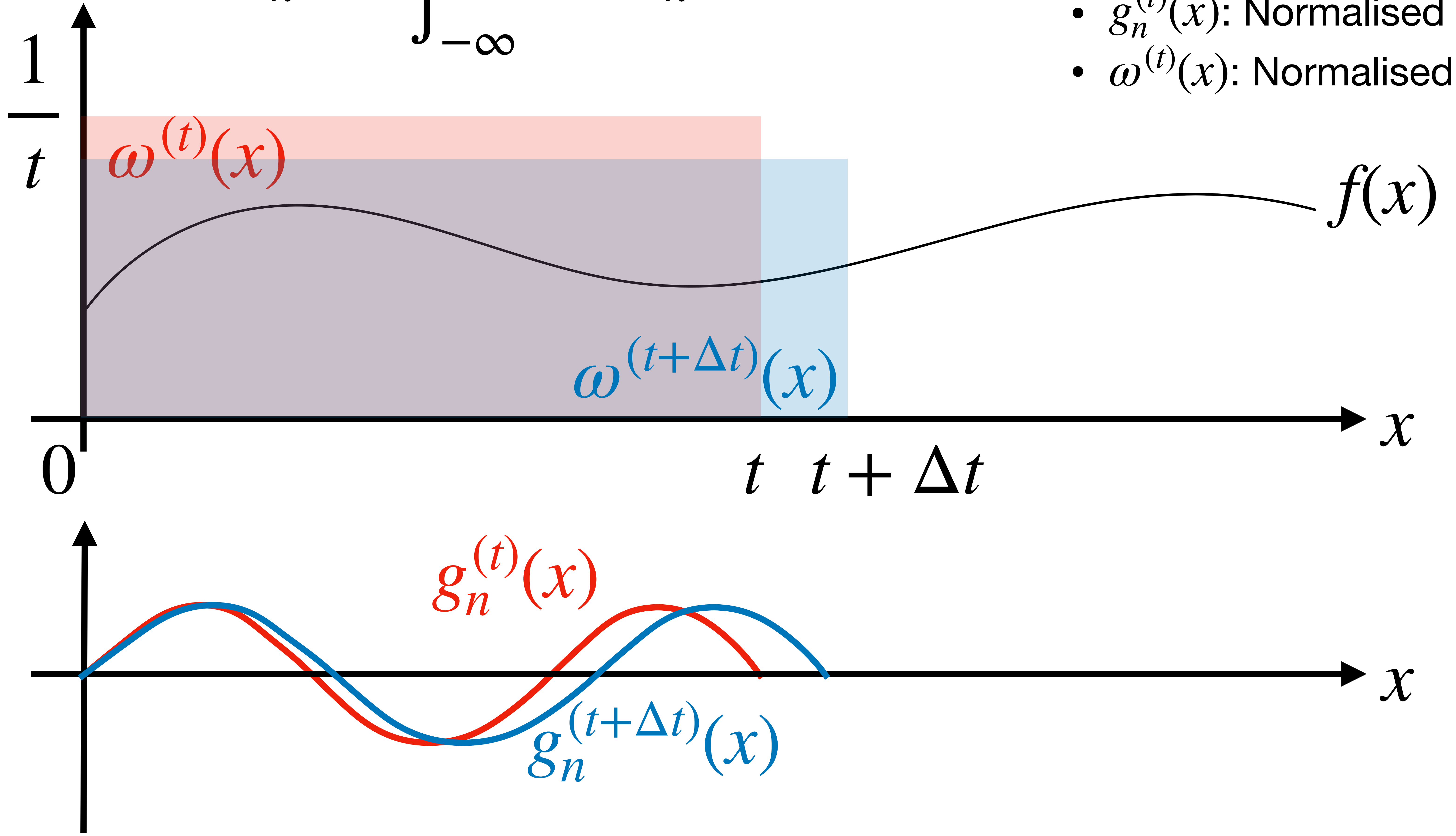
- u_n: n-th coefficient
- $f(x)$: Target function
- $g_n^{(t)}(x)$: Normalised & scaled n-th basis
- $\omega^{(t)}(x)$: Normalised measure (mask)



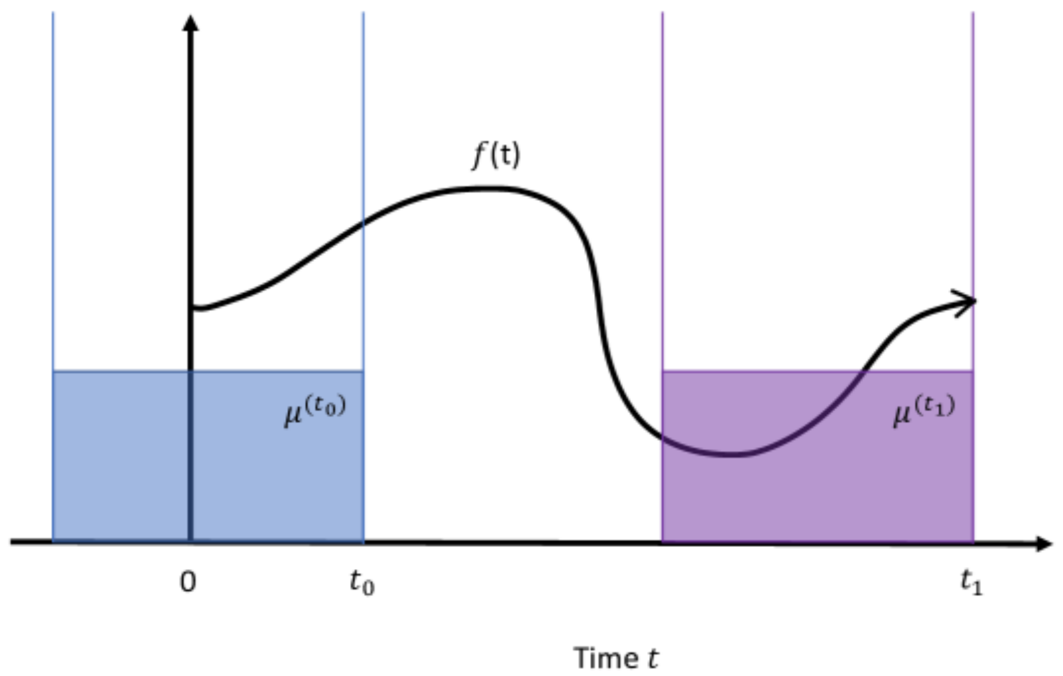
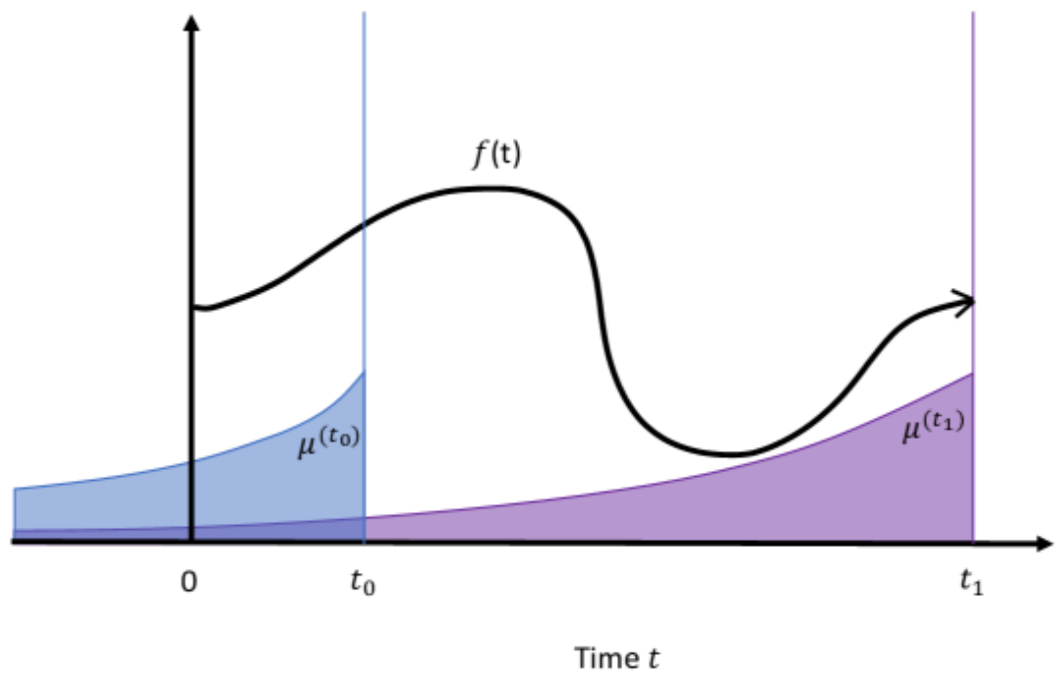
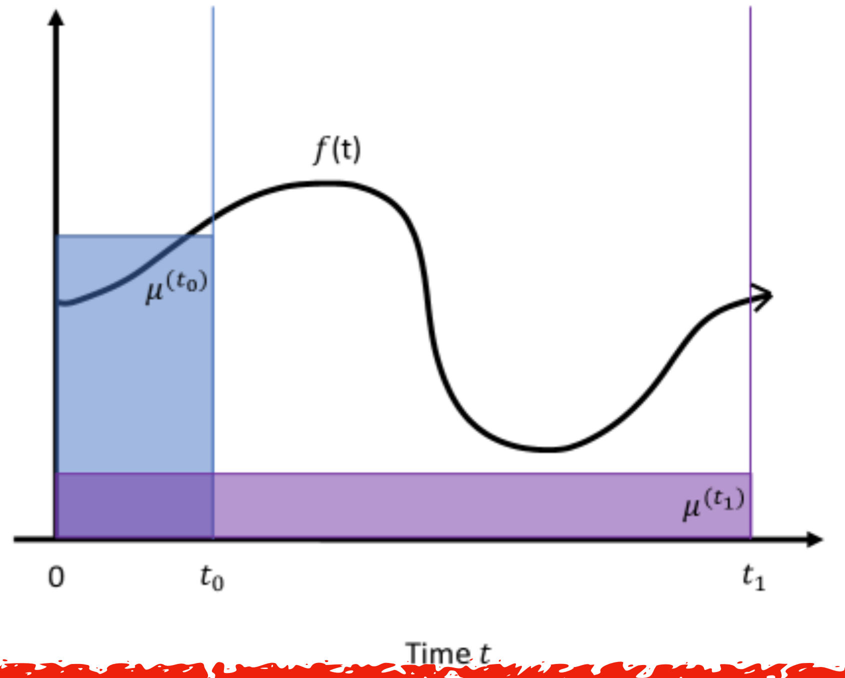
Intuitive Visualisation of the Problem Setting

$$u_n^{(t)} = \int_{-\infty}^{\infty} f(x) g_n^{(t)}(x) \omega^{(t)}(x) dx$$

- u_n: n-th coefficient
- $f(x)$: Target function
- $g_n^{(t)}(x)$: Normalised & scaled n-th basis
- $\omega^{(t)}(x)$: Normalised measure (mask)

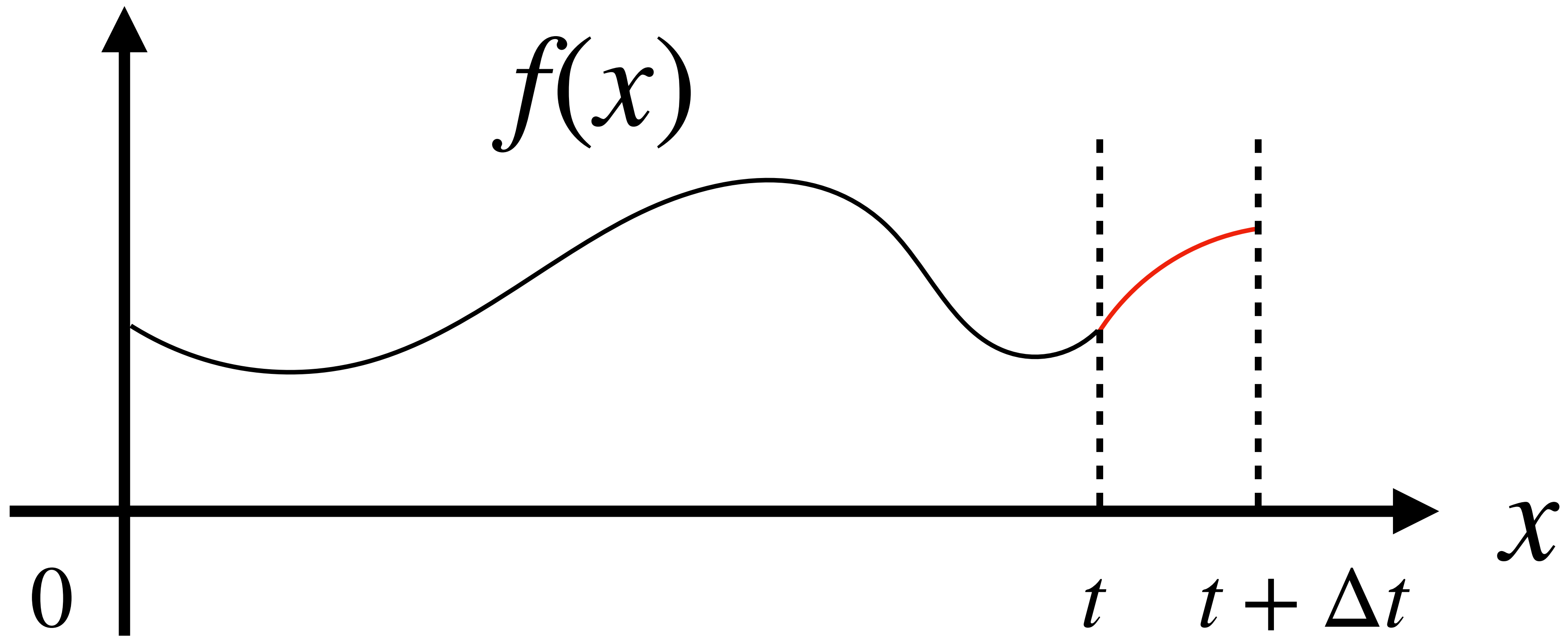


HiPPO variations

	Translated	Weighted by e^{x-t}	Scaled
			
Legendre	✔ (HiPPO-LegT)	Non-orthogonal*	✔ (HiPPO-LegS)
Laguerre	Non-orthogonal*	✔ (HiPPO-LagT)	Non-orthogonal*
Fourier	✔	Non-orthogonal*	Not derived
Chebyshev	✔	Non-orthogonal*	Not derived

We choose HiPPO-LegS as our initial example because it assigns uniform importance to the entire past, from the beginning to the present

Formulating coefficient dynamics



$$f(x) = \sum_{n=0}^{\infty} u_n g_n(x), \qquad g_n^{(t)}(x) = (2n+1)^{1/2} P_n\left(\frac{2x}{t} - 1\right),$$

$$\omega^{(t)}(x) = \frac{1}{t}\mathbb{I}_{[0,t)}(x)$$

Let's derive dynamics of the coefficient u_n.

$$\frac{\mathrm{d}u_n^{(t)}(x)}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t}\int f(x) g_n^{(t)}(x)\omega^{(t)}(x)\ \mathrm{d}x = \int f(x)\frac{\partial g_n^{(t)}(x)}{\partial t}\omega^{(t)}(x)\ \mathrm{d}x + \int f(x) g_n^{(t)}(x)\frac{\partial \omega^{(t)}(x)}{\partial t}\ \mathrm{d}x$$

Using property of Legendre polynomial basis

$$(x+1)P_n'(x) = nP_n + (2n-1)P_{n-1} + (2n-3)P_{n-2} + \ldots,$$

it can be written as

$$\frac{\partial}{\partial t} g_n^{(t)}(x) = -t^{-1}(2n+1)^{\frac{1}{2}}\left[n(2n+1)^{-\frac{1}{2}} g_n^{(t)}(x) + (2n-1)^{\frac{1}{2}} g_{n-1}^{(t)}(x) + (2n-3)^{\frac{1}{2}} g_{n-2}^{(t)}(x) + \ldots\right]$$

$$\frac{\partial}{\partial t}\omega^{(t)}(\cdot) = -t^{-2}\mathbb{I}_{[0,t]} + t^{-1}\delta_t$$
$$= t^{-1}\left(-\omega^{(t)}(\cdot) + \delta_t\right)$$

Formulating coefficient dynamics

Proof (by ChatGPT)

1) $\dfrac{\partial}{\partial t} g_n^{(t)}(x)$

Recall

$$g_n^{(t)}(x) = \sqrt{2n+1}\, P_n(\xi)\,, \qquad \xi = \frac{2x}{t} - 1.$$

By the chain rule,

$$\frac{\partial}{\partial t} g_n^{(t)}(x) = \sqrt{2n+1}\, P_n'(\xi)\, \frac{\partial}{\partial t}\left(\frac{2x}{t} - 1\right) = -\frac{1}{t}\sqrt{2n+1}\,(\xi+1)\,P_n'(\xi),$$

because $\partial_t(2x/t - 1) = -(2x/t^2) = -(\xi+1)/t$.

Use the Legendre identity

$$(\xi+1)P_n'(\xi) = n\,P_n(\xi) + (2n-1)P_{n-1}(\xi) + (2n-3)P_{n-2}(\xi) + \cdots = n\,P_n(\xi) + \sum_{j=1}^{n}\big(2n-2j+1\big)\,P_{n-j}(\xi).$$

Convert $P_k(\xi)$ back to the $g_k^{(t)}$ basis via $P_k(\xi) = g_k^{(t)}(x)/\sqrt{2k+1}$. Then

$$\boxed{\frac{\partial}{\partial t} g_n^{(t)}(x) = -\frac{1}{t}\sqrt{2n+1}\left[\frac{n}{\sqrt{2n+1}}\, g_n^{(t)}(x) + \sum_{j=1}^{n}\sqrt{2(n-j)+1}\; g_{n-j}^{(t)}(x)\right]}$$

which matches the pattern on your slide:

$$-\,t^{-1}(2n{+}1)^{1/2}\Big[n(2n{+}1)^{-1/2}g_n^{(t)}(x) + (2n{-}1)^{1/2}g_{n-1}^{(t)}(x) + (2n{-}3)^{1/2}g_{n-2}^{(t)}(x) + \cdots\Big].$$

Formulating coefficient dynamics

$$u_n^{(t)}(x) = \int f(x) g_n^{(t)} \omega^{(t)}(x) \mathrm{d}x$$

$$\frac{\mathrm{d}u_n^{(t)}(x)}{\mathrm{d}t} = \int f(x) \frac{\partial g_n^{(t)}(x)}{\partial t} \omega^{(t)}(x) \ \mathrm{d}x + \int f(x) g_n^{(t)}(x) \frac{\partial \omega^{(t)}(x)}{\partial t} \ \mathrm{d}x$$

$$= \int f(x) \left[-t^{-1}(2n+1)^{\frac{1}{2}} \left[n(2n+1)^{-\frac{1}{2}} g_n^{(t)}(x) + (2n-1)^{\frac{1}{2}} g_{n-1}^{(t)}(x) + (2n-3)^{\frac{1}{2}} g_{n-2}^{(t)}(x) + \dots \right] \right] \omega^{(t)}(x) \mathrm{d}x$$

$$+ \int f(x) g_n^{(t)}(x) \left[t^{-1} \left(-\omega^{(t)}(x) + \delta_t \right) \right] \mathrm{d}x$$

$$= -\frac{n}{t} u_n^{(t)}(x) - \frac{(2n+1)^{\frac{1}{2}}(2n-1)^{\frac{1}{2}}}{t} u_{n-1}^{(t)}(x) + \cdots - u_n^{(t)}(x) + \frac{1}{t} f(t) g_n^{(t)}(t)$$

$$= -\frac{1}{t} \sum_{k=0}^{n} \left[(2n+1)^{\frac{1}{2}}(2k+1)^{\frac{1}{2}} \right] u_k^{(t)}(x) + \frac{1}{t} f(t) g_n^{(t)}(t)$$

$$g_n^{(t)}(t) = (2n+1)^{\frac{1}{2}} P_n(1) = (2n+1)^{\frac{1}{2}}$$

$$= -\frac{1}{t} \sum_{k=0}^{n} \left[(2n+1)^{\frac{1}{2}}(2k+1)^{\frac{1}{2}} \right] u_k^{(t)}(x) + \frac{1}{t} f(t) (2n+1)^{\frac{1}{2}}$$

Coefficient dynamics

$$\frac{d}{dt}\mathbf{u}^{(t)} = A(t)\mathbf{u}^{(t)} + B(t)f(t)$$

$$\mathbf{u}(t) = \begin{pmatrix} u_0(t) \\ u_1(t) \\ \vdots \end{pmatrix}, \quad A_{nk}(t) = \begin{cases} -\frac{1}{t}(2n+1)^{1/2}(2k+1)^{1/2} & \text{if } n > k \\ -\frac{1}{t}(n+1) & \text{if } n = k, \\ 0 & \text{if } n < k \end{cases} \quad B_n(t) = \frac{1}{t}(2n+1)^{\frac{1}{2}}$$

Now, by solving this simple ODE, we can obtain coefficients for polynomial expansion with online manner

Summary of the HiPPO's Idea

By simply solving the linear ODE:

Input sequence to memorise

$$\frac{d}{dt}\mathbf{u}^{(t)} = A(t)\mathbf{u}^{(t)} + B(t)f(t)$$

Specific matrix and vector corresponding to measure and basis

We can obtain the coefficients $\mathbf{u}^{(t)}$ for corresponding measure and basis.

Expanding the Deterministic Input to HiPPO to $f \sim \mathrm{GP}(0,k)$

The m-th polynomial coefficient $u_m^{(t)} = \int f(x) g_m^{(t)}(x) \omega^{(t)}(x) dx$

Turning deterministic f into stochastic $f \sim \mathrm{GP}(0, k)$

$p(\mathbf{u})$ is now multivariate normal distribution since f is from Gaussian process.
We regard $p(\mathbf{u})$ as inducing variables' distribution.

This is an instance of so-called "Interdomain GPs"

Computing Predictive Distribution

Sparse Variational Gaussian Processes (SVGP)

With non-conjugate Gaussian likelihood, $q(\mathbf{u}) = N(\mathbf{m}, \mathbf{S}) \Rightarrow q(f(x_*)) = \int p(f(x_*)\,|\,\mathbf{u})q(\mathbf{u})d\mathbf{u}$ (**SVGP**). With conjugate Gaussian likelihood, we can obtain the analytical optimal (**SGPR)**:

$$m_* = \frac{1}{\sigma^2}\mathbf{K}_{*\mathbf{u}}\left(\mathbf{K}_{\mathbf{uu}} + \frac{1}{\sigma^2}\mathbf{K}_{\mathbf{uf}}\mathbf{K}_{\mathbf{fu}}\right)^{-1}\mathbf{K}_{\mathbf{uf}}\,\mathbf{y},$$

$$\mathrm{var}[f_*] = \mathbf{K}_{**} - \mathbf{K}_{*\mathbf{u}}\left(\mathbf{K}_{\mathbf{uu}} + \frac{1}{\sigma^2}\mathbf{K}_{\mathbf{uf}}\mathbf{K}_{\mathbf{fu}}\right)^{-1}\mathbf{K}_{\mathbf{u}*}\,.$$

Cross-covariance: $\left[\mathbf{K}^{(t)}_{\mathbf{fu}}\right]_{nm} = \mathrm{COV}\left[f(x_n), \int f(x)g^{(t)}_m(x)\omega^{(t)}(x)dx\right]$

Inducing covariance $\left[\mathbf{K}^{(t)}_{\mathbf{uu}}\right]_{nm} = \mathrm{COV}\left[\int f(x)g^{(t)}_n(x)\omega^{(t)}(x)dx, \int f(x)g^{(t)}_m(x)\omega^{(t)}(x)dx\right]$

How can we compute these covariances?

Computing Predictive Distribution

Sparse Variational Gaussian Processes (SVGP)

$$\left[\mathbf{K}_{\mathbf{fu}}^{(t)}\right]_{nm} = \mathrm{COV}\left[f(x_n), \int f(x) g_m^{(t)}(x) \omega^{(t)}(x) dx\right]$$

$$= \mathbb{E}\left[f\left(x_n\right) \int f(x) g_m^{(t)}(x) \omega^{(t)}(x) dx\right]$$

$$= \int E\left[f\left(x_n\right) f(x)\right] g_m^{(t)}(x) \omega^{(t)}(x) dx$$

$$= \int k\left(x_n, x\right) g_m^{(t)}(x) \omega^{(t)}(x) dx \quad \Rightarrow \quad \frac{d}{dt}\left[\mathbf{K}_{\mathbf{fu}}^{(t)}\right]_n = A\left[\mathbf{K}_{\mathbf{fu}}^{(t)}\right]_n + Bk(x_n, t)$$

The same formula as HiPPO

This can be computed with the ODE Solver

Computing Predictive Distribution

Sparse Variational Gaussian Processes (SVGP)

$$
\begin{aligned}
\left[\tilde{\mathbf{K}}_{\mathrm{uu}}^{(t)}\right]_{\ell m} &= \mathrm{COV}\left[\int f(x) g_\ell^{(t)}(x)\omega^{(t)}(x)dx, \int f(x')\, g_m^{(t)}(x')\omega^{(t)}(x')dx'\right] \\
&= \iint E\left[f(x)f(x')\right] g_\ell^{(t)}(x)\omega^{(t)}(x) g_m^{(t)}(x')\omega^{(t)}(x') dx dx' \\
&= \iint k(x, x')\, g_\ell^{(t)}(x)\omega^{(t)}(x) g_m^{(t)}(x')\omega^{(t)}(x') dx dx'\,.
\end{aligned}
$$

We have two options to compute it:

- Use random Fourier features (RFF) to decouple the correlated integral.
- As is done in the HiPPO formulation, take time derivative wrt t to obtain similar ODE as HiPPO's.

Both methods can be reduced to a simple ODE computation.

Computing Predictive Distribution

With RFF

$$[\tilde{\mathbf{K}}_{\mathbf{uu}}^{(t)}]_{\ell m} = \iint k(x, x')\, g_\ell^{(t)}(x)\omega^{(t)}(x) g_m^{(t)}(x')\omega^{(t)}(x') dx dx' .$$

HIPPO-ODEs again!

Bochner's Theorem

$$k(x, x') = \mathbb{E}_{p(w)}\left[\cos(wx)\ \cos(wx') + \sin(wx)\ \sin(wx')\right],$$

$$w \sim p(w)$$

$$Z_{w,\ell}^{(t)} = \int \cos(wx) g_\ell^{(t)}(x)\omega^{(t)}(x)\,\mathrm{d}x, \qquad Z_{w,\ell}^{\prime(t)} = \int \sin(wx) g_\ell^{(t)}(x)\omega^{(t)}(x)\,\mathrm{d}x,$$

$$\mathbf{Z}_w^{(t)} = \left[Z_{w,1}^{(t)}, \cdots, Z_{w,M}^{(t)}\right]^\intercal, \quad \mathbf{Z}_w^{\prime(t)} = \left[Z_{w,1}^{\prime(t)}, \cdots, Z_{w,M}^{\prime(t)}\right]^\intercal .$$

Collecting N Monte Carlo samples, we have

$$\mathbf{Z}^{(t)} = \left[\mathbf{Z}_{w_1}^{(t)} \quad \mathbf{Z}_{w_2}^{(t)} \quad \cdots \quad \mathbf{Z}_{w_N}^{(t)} \quad \mathbf{Z}_{w_1}^{\prime(t)} \quad \mathbf{Z}_{w_2}^{\prime(t)} \quad \cdots \quad \mathbf{Z}_{w_N}^{\prime(t)}\right],$$

$$\Rightarrow \mathbf{K}_{\mathbf{uu}}^{(t)} \approx \frac{1}{N}\mathbf{Z}^{(t)}\left(\mathbf{Z}^{(t)}\right)^\intercal .$$

Computing Predictive Distribution

Sparse Variational Gaussian Processes (SVGP)

With non-conjugate Gaussian likelihood, $q(\mathbf{u}) = N(\mathbf{m}, \mathbf{S}) \Rightarrow q(f(x_*)) = \int p(f(x_*) \,|\, \mathbf{u}) q(\mathbf{u}) d\mathbf{u}$ (**SVGP**).

With conjugate Gaussian likelihood, we can obtain the analytical optimal (**SGPR**):

$$m_* = \frac{1}{\sigma^2} \mathbf{K}_{*\mathbf{u}} \left(\mathbf{K}_{\mathbf{uu}} + \frac{1}{\sigma^2} \mathbf{K}_{\mathbf{uf}} \mathbf{K}_{\mathbf{fu}} \right)^{-1} \mathbf{K}_{\mathbf{uf}} \mathbf{y},$$

$$\mathrm{var}[f_*] = \mathbf{K}_{**} - \mathbf{K}_{*\mathbf{u}} \left(\mathbf{K}_{\mathbf{uu}} + \frac{1}{\sigma^2} \mathbf{K}_{\mathbf{uf}} \mathbf{K}_{\mathbf{fu}} \right)^{-1} \mathbf{K}_{\mathbf{u}*} .$$

Cross-covariance: $\left[\mathbf{K}_{\mathbf{fu}}^{(t)}\right]_{nm} = \mathrm{COV}\left[f(x_n), \int f(x) g_m^{(t)}(x) \omega^{(t)}(x) dx \right]$

Inducing covariance $\left[\mathbf{K}_{\mathbf{uu}}^{(t)}\right]_{nm} = \mathrm{COV}\left[\int f(x) g_n^{(t)}(x) \omega^{(t)}(x) dx, \int f(x) g_m^{(t)}(x) \omega^{(t)}(x) dx \right]$

Both of the covariances can be computed using the simple ODE.
But we still need batch computation in this formulation.
→ How can we update with online manner?

Updating $q(\mathbf{u})$ in Online Manner

Online ELBO (Bui et al., NIPS 2017)

$$\sum_{i=1}^{n_2} \mathbb{E}_{q_2(f_i)} \left[\log p_{t_2}\left(y_i \middle| f_i\right) \right] - \mathrm{KL}\left[q_{t_2}\left(\mathbf{u}_{t_2}\right) \middle\| p_{t_2}\left(\mathbf{u}_{t_2}\right) \right] + \mathrm{KL}\left[\tilde{q}_{t_2}\left(\mathbf{u}_{t_1}\right) \middle\| p_{t_1}\left(\mathbf{u}_{t_1}\right) \right] - \mathrm{KL}\left[\tilde{q}_{t_2}\left(\mathbf{u}_{t_1}\right) \middle\| q_{t_1}\left(\mathbf{u}_{t_1}\right) \right]$$

ELBO for the new data — Incorporation of the previous posterior

- The Gaussian process exhibits long-term memory like HiPPO
- The correction regularizes for the likelihood and other model parameters against the past

HIPPO-SVGP in Multidimensional Input Setting

- Suppose there is a time order for the first batch of training points with inputs $\{\mathbf{x}_n^{(1)}\}_{n=1}^{N_1}$, such **that** $\mathrm{x}_i^{(1)}$ **appears after** $\mathbf{x}_j^{(1)}$ if $i > j$,
- We further assume $\mathbf{x}_i$ appears at time index $i\Delta t$ (i.e., $\mathbf{x}(i\Delta t) = \mathbf{x}_i^{(1)}$), where Δt is a user-specified constant step size.
- We can again obtain interdomain prior covariance matrices via HiPPO recurrence. For example, a forward Euler method applied to the ODE for $\mathbf{K}_{\mathbf{fu}}^t$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[\mathbf{K}_{\mathbf{fu}}^{(t)}\right]_{n,:} = \mathbf{A}(t)\left[\mathbf{K}_{\mathbf{fu}}^{(t)}\right]_{n,:} + \mathbf{B}(t)k\left(\mathbf{x}_n, t\right),$$

yields

$$[\mathbf{K}_{\mathbf{fu}}^{((i+1)\Delta t)}]_{n,:} = [\mathbf{I} + \Delta t\mathbf{A}(i\Delta t)][\mathbf{K}_{\mathbf{fu}}^{(i\Delta t)}]_{n,:} + \Delta t\mathbf{B}(i\Delta t)k\left(\mathbf{x}_n^{(1)}, \mathbf{x}_i^{(1)}\right).$$

HIPPO-SVGP in Multidimensional Input Setting

Heuristic approaches on ordering of points:

- Option 1: Random permutations
- Option 2: Kernel distance minimization (OHSVGP-k) $x_i = \text{argmin}_{x \in X} k(x, x_{i-1})$
  - Have closer points in the kernel distance helps gets "smoother" signal for the kernel matrix ODEs
- Option 3: Oracle, based on how task data is created
- But many problems have natural ordering e.g. text data

HIPPO-SVGP in High-Dimensional Output Setting

We use SVGPVAE from Jazbec et al. (2021) to give us **OHSVGPVAE:**

- **Likelihood:** $p(y \mid \varphi_\theta(f(t)))$ with a **decoder** network $\varphi_\theta : \mathbb{R}^{d_{\text{latent}}} \to \mathbb{R}^{d_y}$
- **Latent Variables:** $f(t) \sim$ **Multi-Output GP**
- **Posterior:** HIPPO-SVGP formulation for $q(f(t))$ with $q(\mathbf{u}) \equiv q(\mathbf{u} \mid \phi(y))$, where we have an encoder network $\phi : \mathbb{R}^{d_y} \to \mathbb{R}^{d_{\text{latent}}}$. $q(\mathbf{u})$ is defined according to Jazbec et al. (2021)
- **ELBO:** Follows Jazbec et al. (2021), but for the online setting we also use Elastic Weight Consolidation (EWC; Kirkpatrick et al. (2017)) on the encoder and decoder network weights

Kirkpatrick et al. (2017). Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences.
Metod Jazbec, Matt Ashman, Vincent Fortuin, Michael Pearce, Stephan Mandt, and Gunnar Ratsch (2021). Scalable Gaussian process variational autoencoders. In AISTATS.

Summary of our Approach

Classical Online SVGP vs. HiPPO SVGP

1. Classical Online SVGP

- Inducing point location is a parameter:
  It requires gradient-based optimization for each batch.
- As the location is not fixed, **it is not guaranteed that the prediction equals to that of batch SVGP**.
  - This results in the loss of old history information.

2. Our HiPPO SVGP

- "Locations" are replaced by time-varying polynomial bases: **No free parameter to “relocate”, and $q(\mathbf{u})$ is updated in closed form** due to conjugacy.
  - Cross-covariance and inducing covariance can be computed via the simple ODE.
- As we don’t parameterise the location, **the prediction is guaranteed to be the same distribution as batch SGPR**.

Experiments

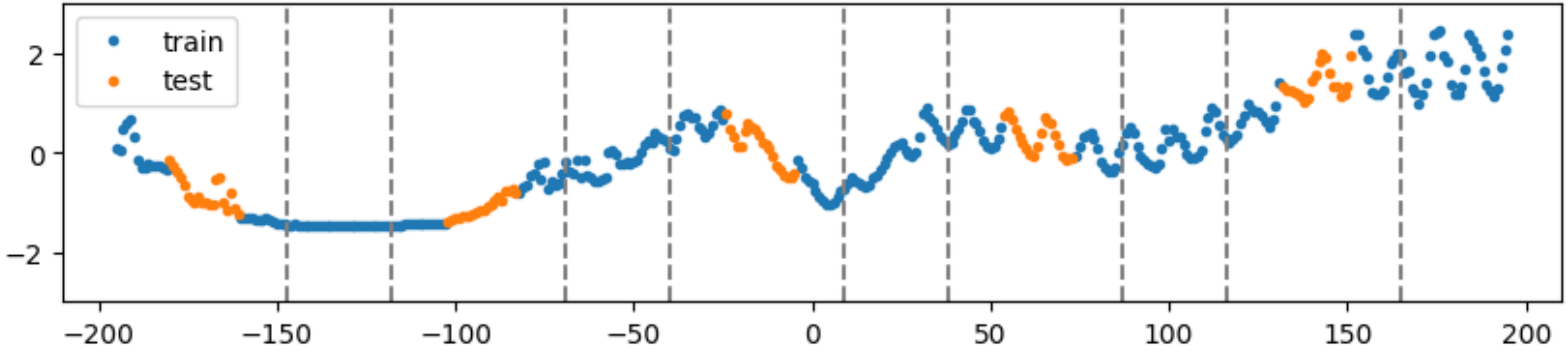
Experimental Setup

Solar Irradiance (Lean, J. (2004). Solar irradiance reconstruction. NOAA/NGDC.)

Test Set: Five segments of length 20 removed for testing.

Online Learning: Data split into 10 sequential tasks.

Objective: To show OHSGPR's efficiency in streaming data without revisiting old tasks.



Baselines:

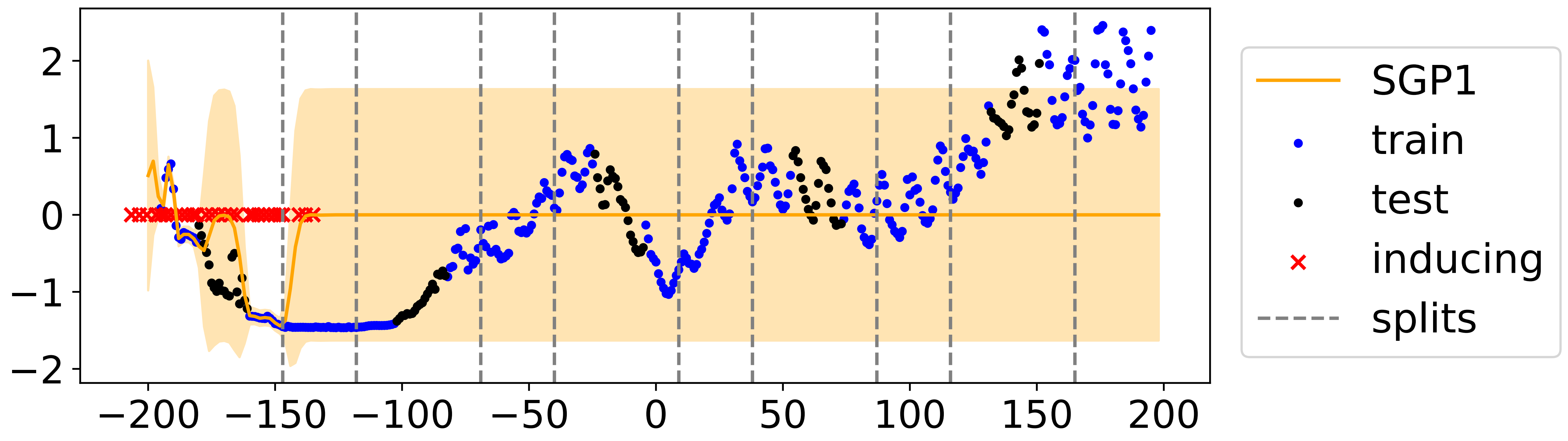
- With the online GP correction term, we call them OSVGP/OSGPR etc…
- Variational Fourier Features (**VFF**):
  - Also inter-domain approach with Fourier eigenfunctions on bounded Euclidean domain
  - Requires computing covariances as integrals over a **predefined interval** covering the **whole range of the time indices from all tasks** (including unobserved ones) ⇒ impractical for real world problems
- Online Variational Conditioning (**OVC**)
  - Uses pivoted Cholesky to initialize the inducing points
  - Basically a greedy variance hunter - get points that span the major directions of the kernel matrix
  - Either we fix (OVC) or optimize them (OVC-opt)

James Hensman, Nicolas Durrande, and Arno Solin (2018). Variational Fourier features for Gaussian processes. JMLR
Wesley J. Maddox, Samuel Stanton, and Andrew Gordon Wilson (2021). Conditioning sparse variational Gaussian processes for online decision-making. NeurIPS

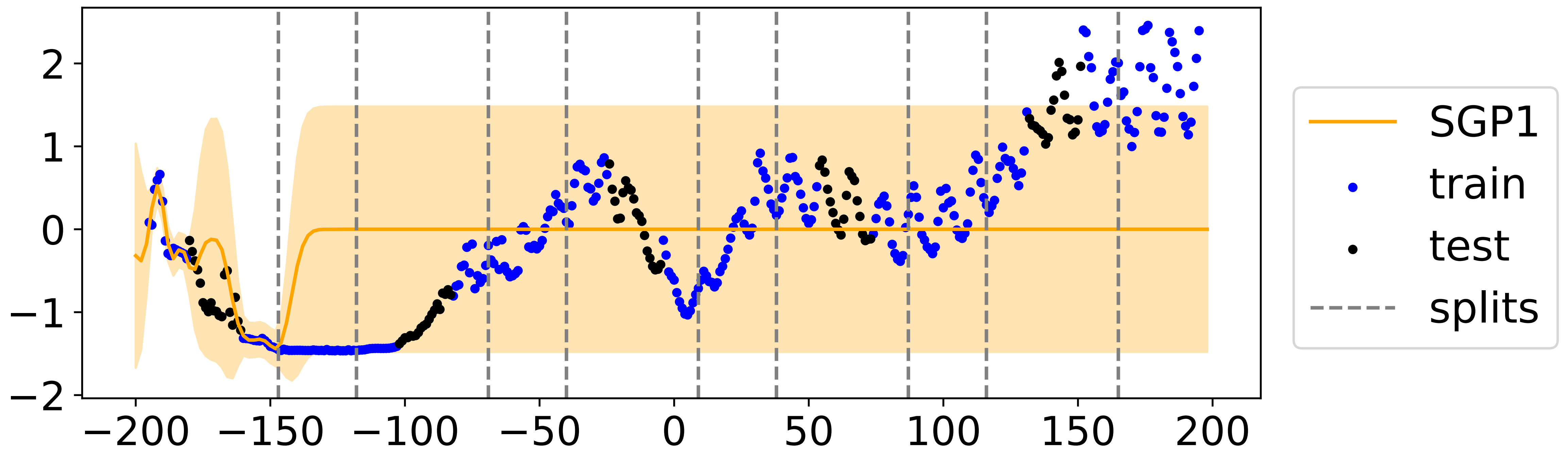
Visualisation of the Results

Task #1

Online SGPR (baseline)



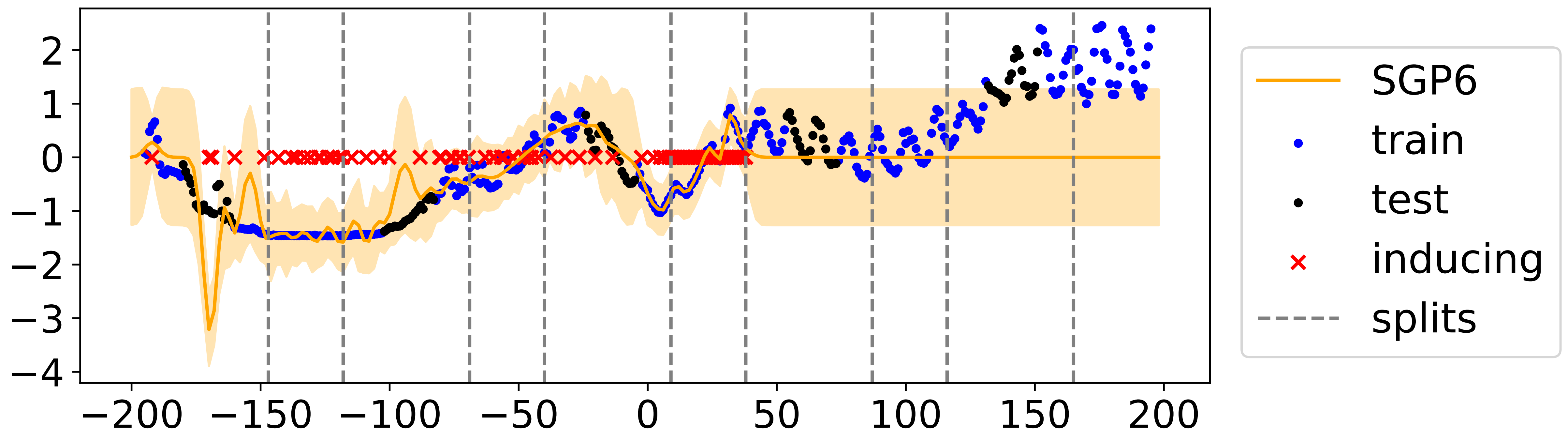
Online HiPPO SGPR (ours)



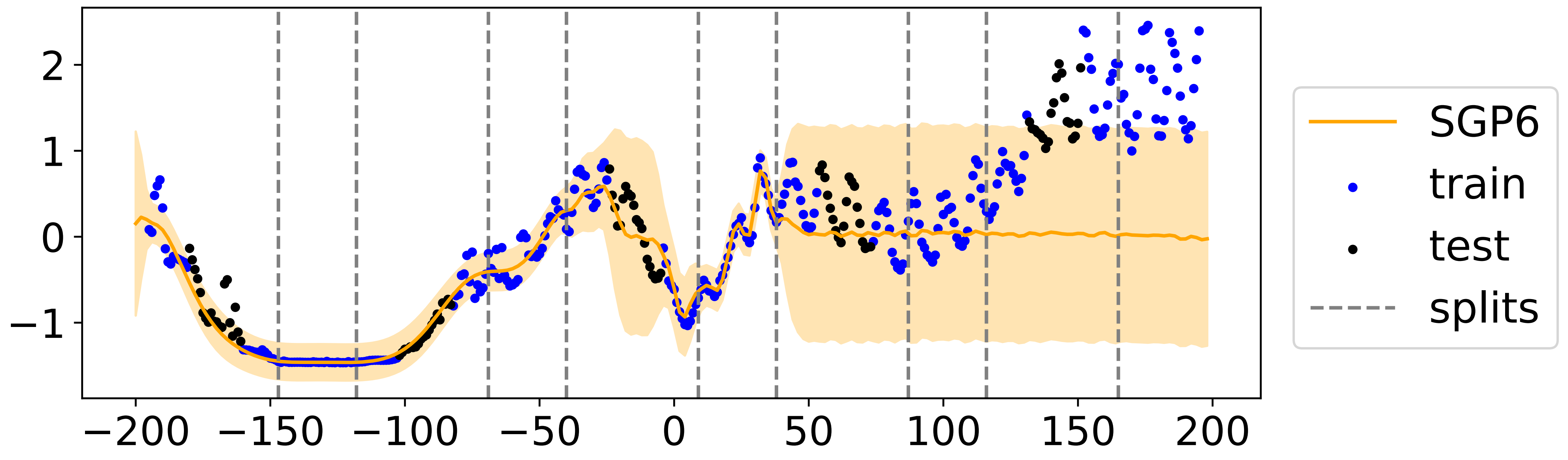
Visualisation of the Results

Task #6

Online SGPR (baseline)

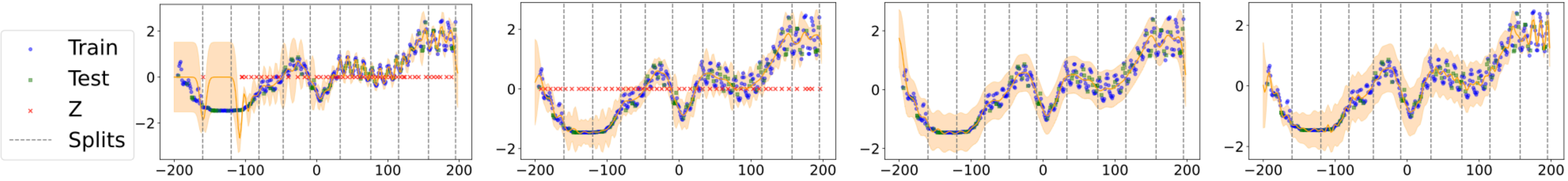


Online HiPPO SGPR (ours)



Visualisation of the Results

Task #10



(a) OSGPR (b) OVC (c) OVFF (d) OHSGPR

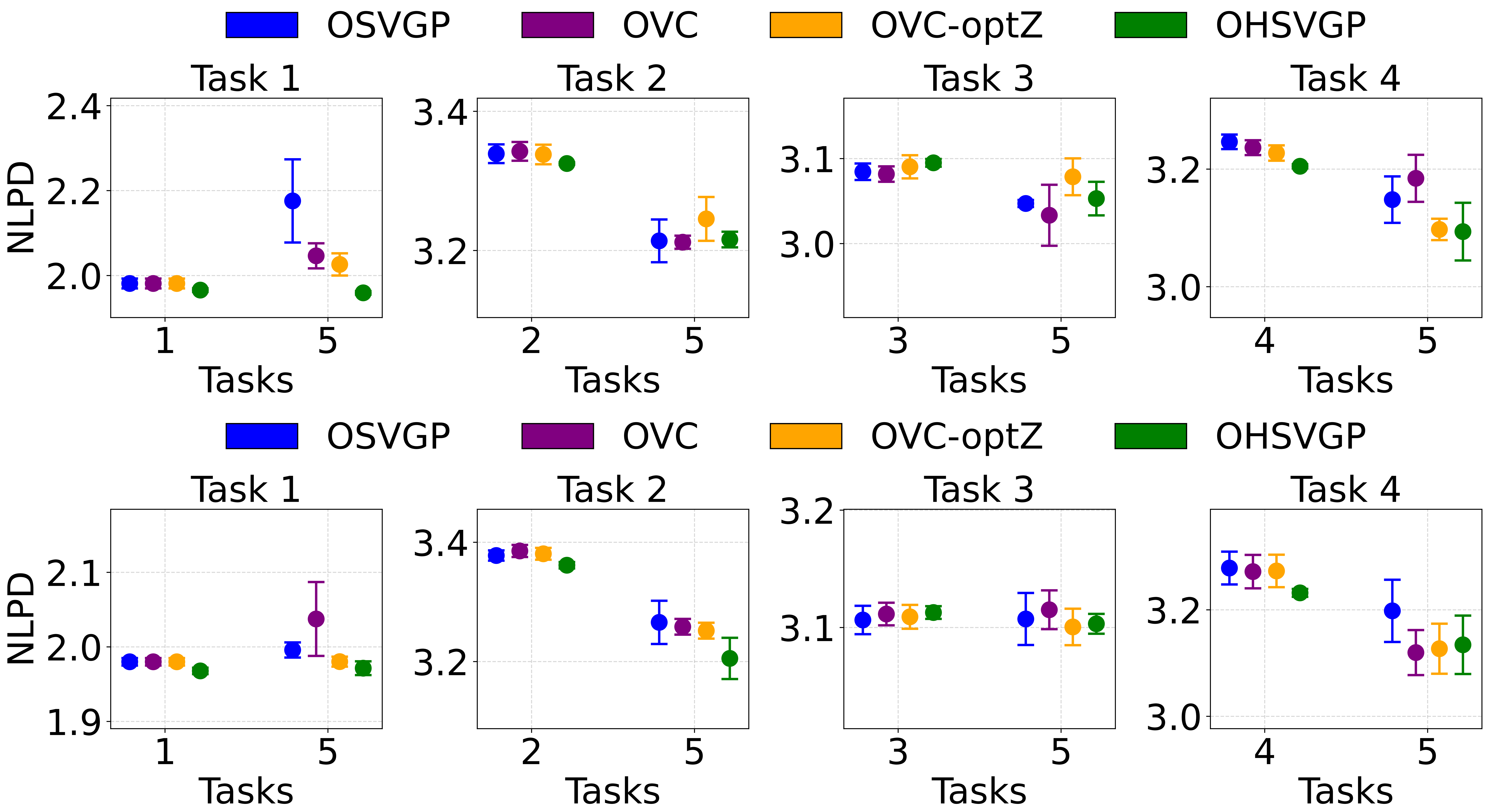
Figure 1. Predictive mean ±2 standard deviation of OSGPR, OVC, OVFF, and OHSGPR after task 10 of the Solar dataset. M = 50 inducing variables are used.

Our method can adapt to new data, and there is little loss of past memories.

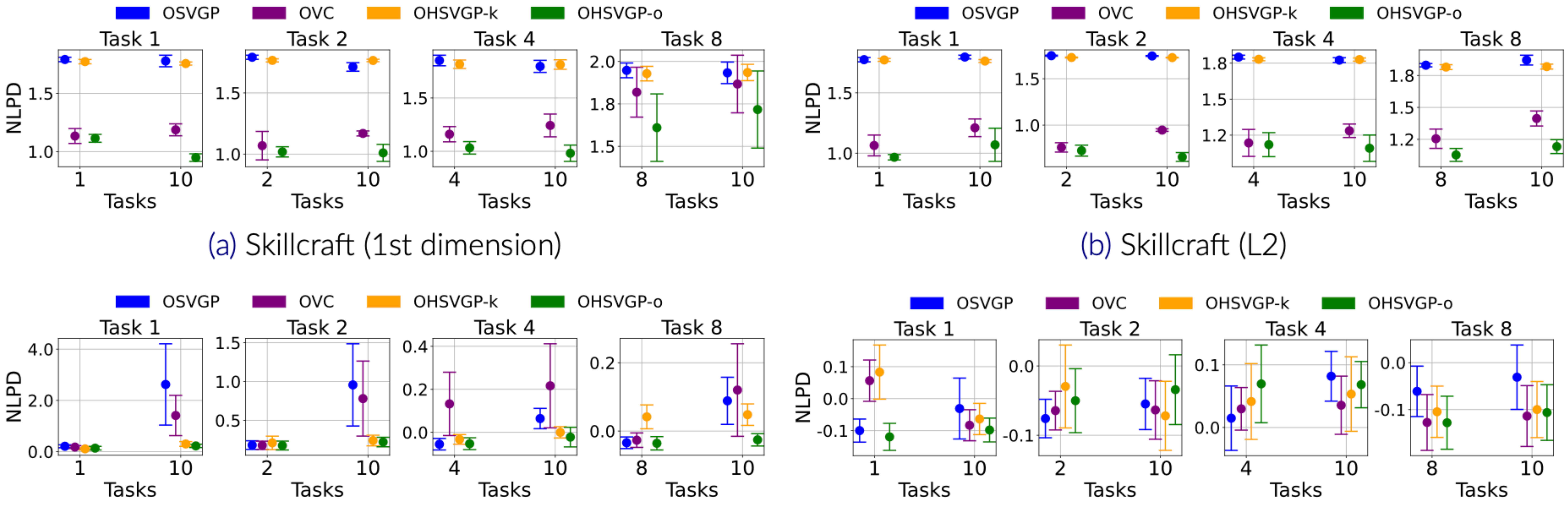
Quantitative Comparison - COVID Modeling

M=15

M=30



Quantitative Comparison - Multidimensional Input



(a) Skillcraft (1st dimension)

(b) Skillcraft (L2)

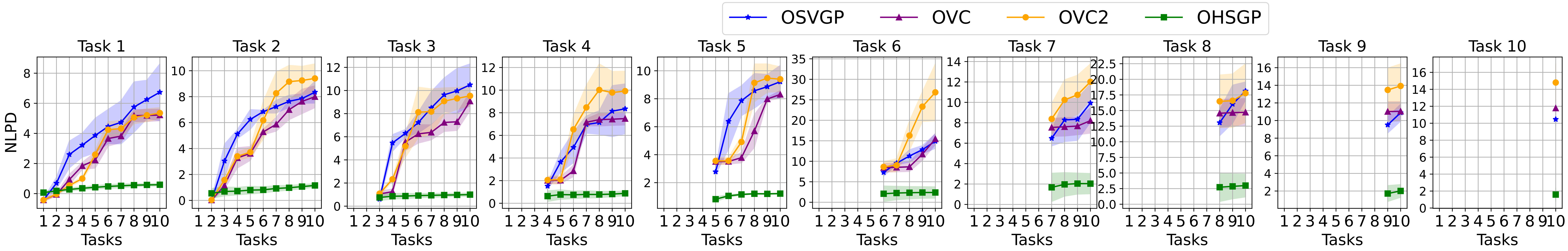
(c) Powerplant (1st dimension)

(d) Powerplant (L2)

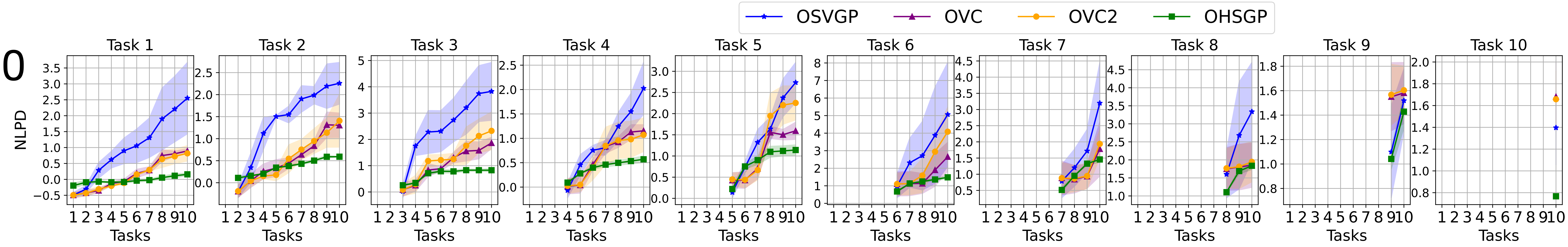
Figure 2. Test set NLPD after continually learning Task i and after learning all the tasks for $i = 1, 2, 4, 8$. Tasks are created by splitting Powerplant and Skillcraft datasets with inputs sorted either according to the 1st input dimension or L2 distance to the origin.

Quantitative Comparison - GPVAE Fitted on Climate Data

M=50



M=100



Computational Cost

Table 1: Wall-clock accumulated runtime for learning all the tasks on a single NVIDIA RTX3090 GPU in seconds, of different models for time series prediction experiments.

	Solar Irradiance		**Audio Data**		**COVID**	
Method	M		M		M	
	50	150	100	200	15	30
OSGPR/OSVGP	140	149	144	199	525	530
OVC	0.450	0.620	0.558	0.863	345	360
OVFF	0.327	0.354	0.295	0.356	-	-
OHSGPR/OHSVGP	0.297	0.394	0.392	0.655	370	380

Under fixed kernel parameters, **Online SGPR requires gradient-based optimisation for each new batch**, whereas **Online HiPPO SGPR only evolves its ODE**.

Making it dramatically faster overall.

Summary of the Experimental Results

Qualitative Findings

- Online SVGP (baseline) gradually forgets earlier segments as it shifts inducing points.
- Online HiPPO SVGP (proposed) retains stable predictions for both recent and older time windows.

Quantitative Performance

- Our method shows consistently lower RMSE/NLPD for larger number tasks, confirming stronger long-term memory than other online methods.

Computational Efficiency

- Both methods are sparse, but Online HiPPO SVGP uses fast ODE-based updates and avoids gradient based update of inducing points' location.

Online HiPPO SVGP mitigates forgetting and maintains scalability, outperforming the baseline in accuracy and memory retention.

Summary

- **We extended the HiPPO memory mechanism from deterministic signals to GPs.**
- **Achieved a natural interdomain GP formulation with polynomial-based inducing variables.**

Key Benefits:

- **Maintains long-term memory in online setting** and **solves the trade-off between memorizing and forgetting**.
- Remains **computationally feasible** in a streaming setting.

Limitations:

- The performance is limited by the expressive power of the selected polynomial basis.
- Numerical stability.