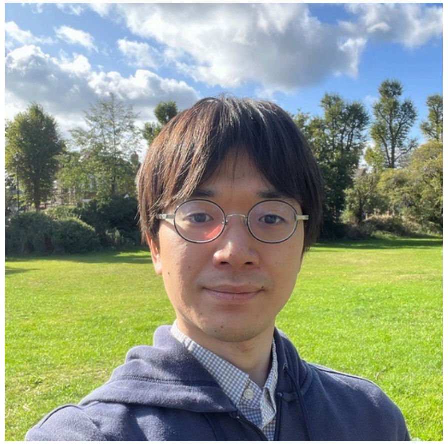
On “Modernising” Sparse Gaussian Processes

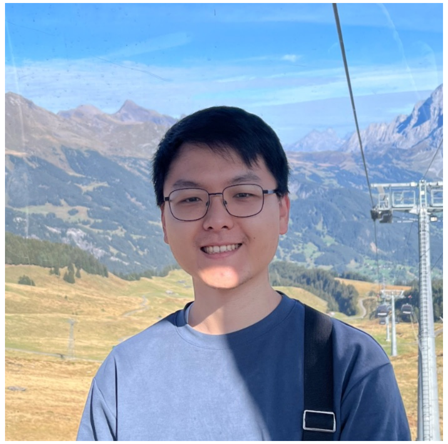
Yingzhen Li

yingzhen.li@imperial.ac.uk

Wenlong Chen



Naoki Kiyohara



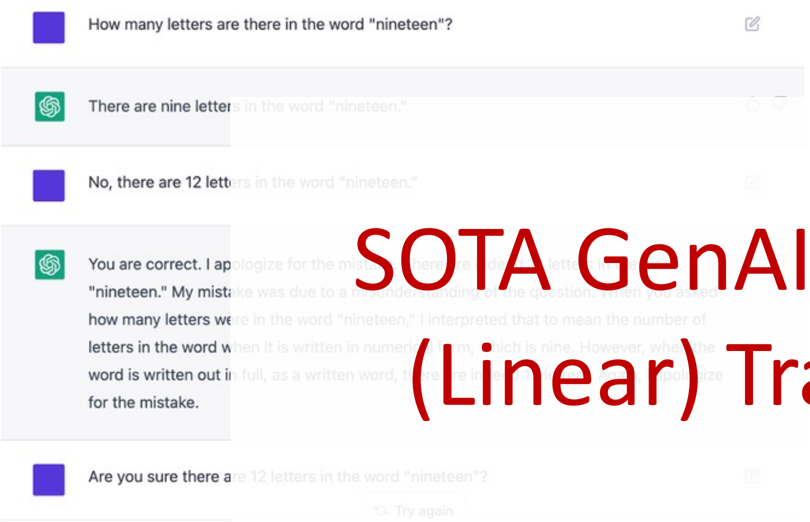
Harrison Bo Hua Zhu

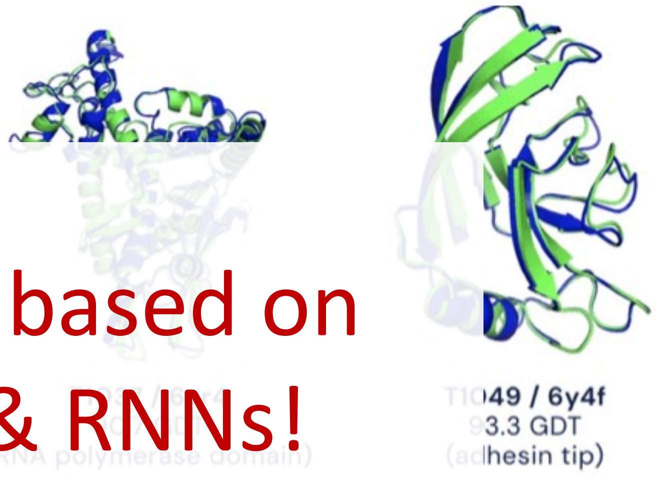
Based on the following paper and preprint:
Chen and Li, ICLR 2023
“Calibration Transformers via Sparse Gaussian Processes”

Chen et al., https://arxiv.org/abs/2502.08736
“Recurrent Memory for Online Interdomain Gaussian Processes”

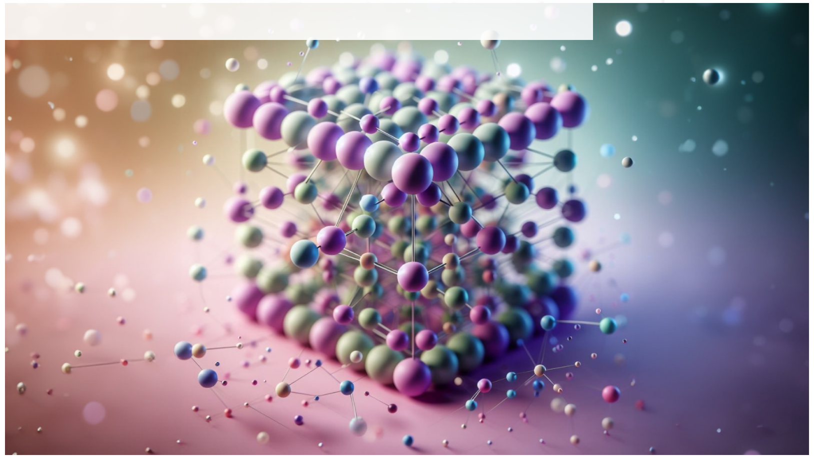
Generative AI BOOM

State-of-the-art AI by the end of Mar 2025





SOTA GenAI models are based on (Linear) Transformers & RNNs!



ChatGPT (OpenAI), FLUX (Black Forest Labs), AlphaFold (DeepMind), MatterGen (MSR)

Ask LLMs for Decision Making?

Users are hardly convinced by high accuracy only!



They want:

- Recommended decision suggestions with convincing reasoning processes
- Risk and uncertainty analysis for the recommended solutions

Bayesian Inference

$\pi(W) = p(W|data)$

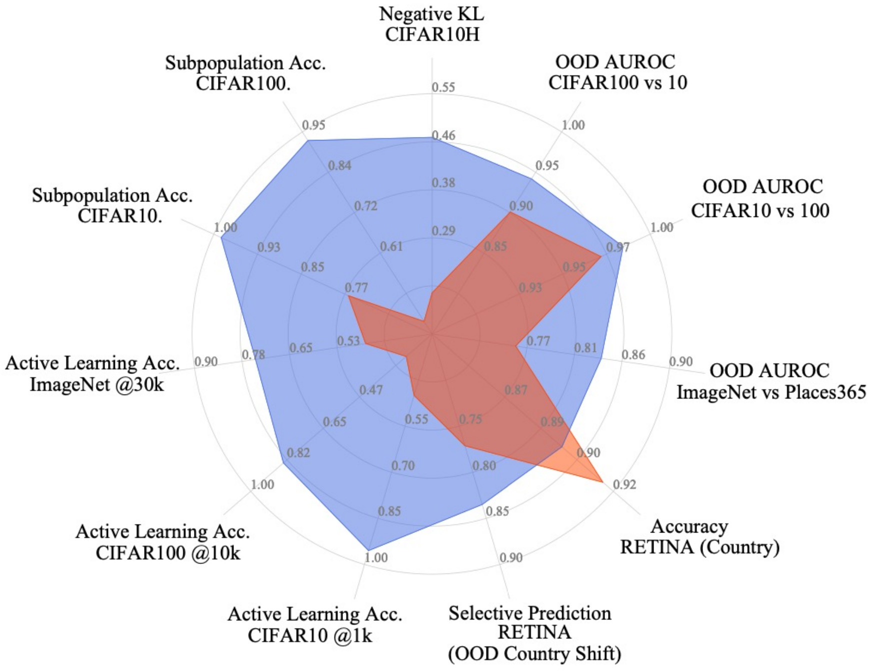
$$P(W|\,data) = \frac{P(W)P(data\,|\,W)}{P(data)}$$

- $P(W)$: prior distribution
- $P(data\,|\,W)$: likelihood of W given $data$
- $P(W|\,data)$: posterior distribution of W given $data$
- $P(data)$: marginal likelihood/model evidence

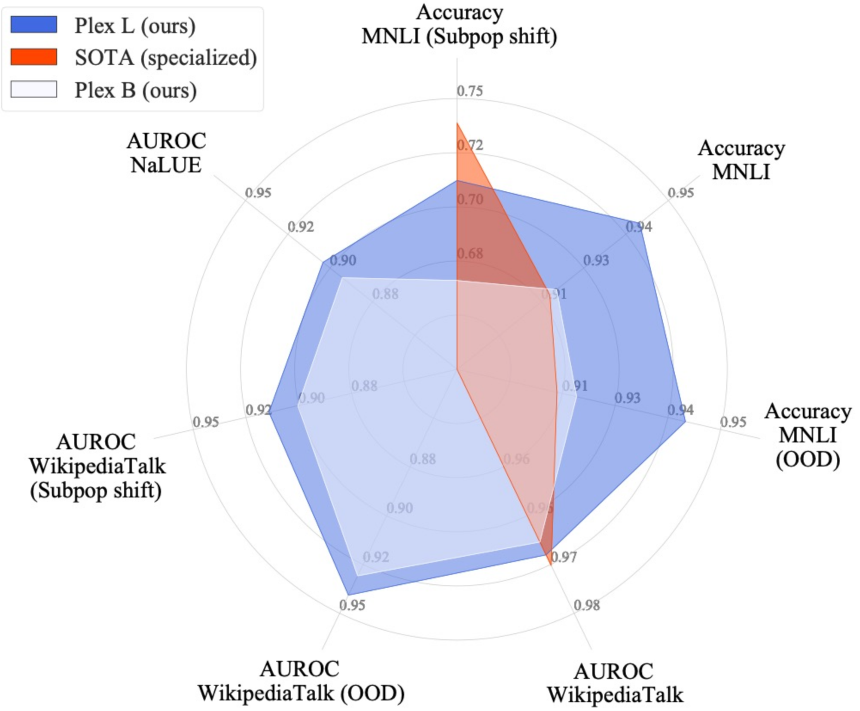
$$P(data) = \int P(W)P(data\,|\,W)$$

Image courtesy of Sebastian Nowozin
Re-use of the image for any other purpose is not allowed

Transformer + Weight-Space Bayesian Inference?



ViT-Plex-L (325M)



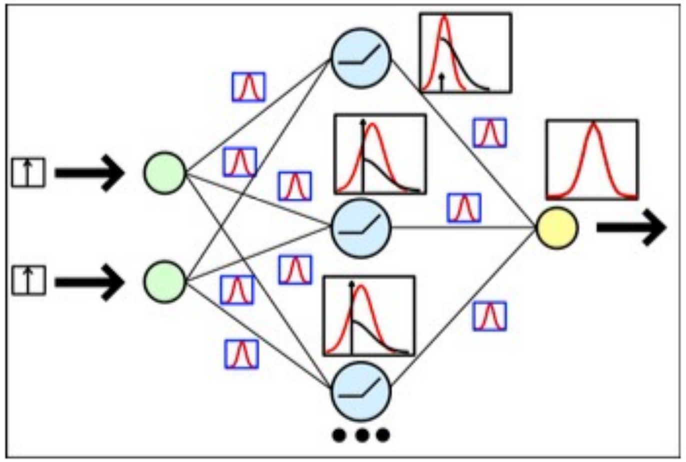
T5-Plex-L (880M)

Major Challenge: running accurate Bayesian inference on billions of weights!

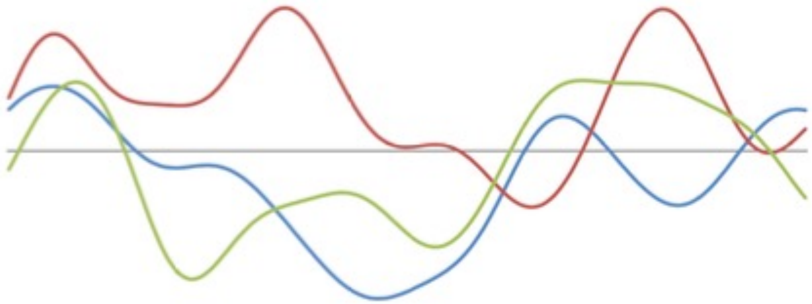
(not going to be solved anytime soon…)

Tran et al. Plex: Towards Reliability using Pretrained Large Model Extensions. arXiv 2207.07411

Weight-Space → Function-Space

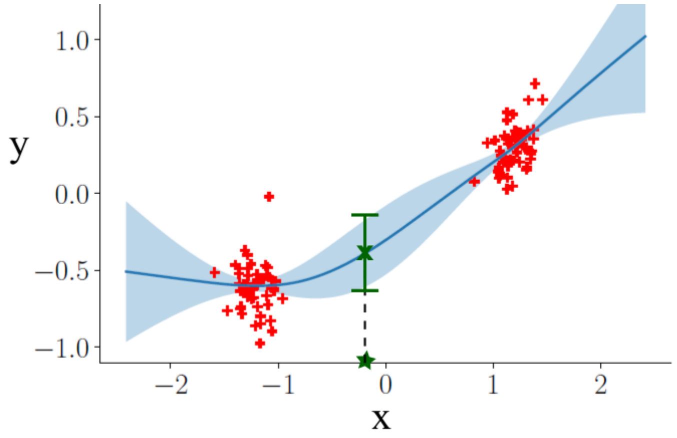


(a) weight space view



(b) function space view

- $W \sim q(W) \Leftrightarrow f \sim q_{BNN}(f)$
- In practice we care more about predictive mean & variance (which is quantifying the function-space behaviour)



Gaussian Processes Prior

$$f(\cdot) \sim GP(m(\cdot), k(\cdot, \cdot))$$

Prior over functions: Gaussian distribution over infinite number of random variables indexed by $\{x\}$

(marginal) $f_X \sim \mathcal{N}(m_X, K_{XX})$ $[K_{XX}]_{ij} = k(x_i, x_j)$

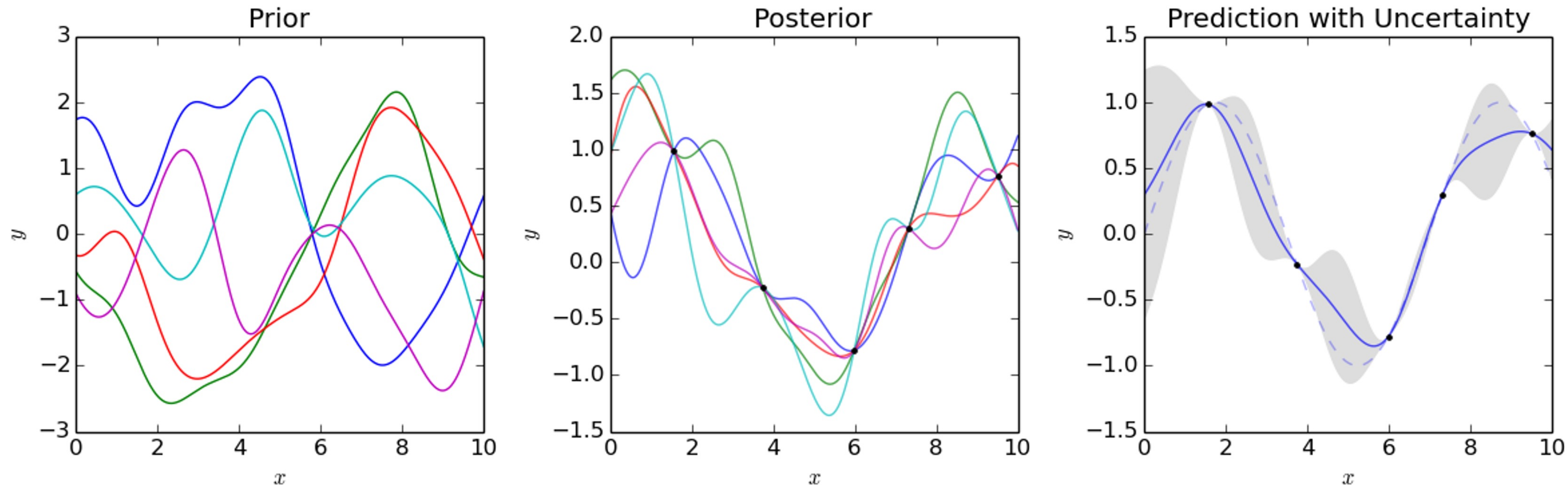


Fig credit: Richard Wilkinson's GPSS 2019 lecture

Sparse Variational Gaussian Process (SVGP) 101

$f \sim GP(0, k(\cdot,\cdot)) \quad \Rightarrow \quad$ Prior: $\quad p(\mathbf{f_X}) = \mathcal{N}(\mathbf{0}, \mathbf{K_{XX}}) \quad [\mathbf{K_{XX}}]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$

Exact posterior inference requires inverting $\mathbf{K_{XX}}$ which has $O(N^3)$ cost!

Inducing Variables: $\mathbf{u_Z} = f(\mathbf{Z}) \sim \mathcal{N}(\mathbf{0}, \mathbf{K_{ZZ}}) \Rightarrow$ Augmented Prior: $p(\mathbf{f_X}, \mathbf{u_Z}) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K_{XX}} & \mathbf{K_{XZ}} \\ \underbrace{\mathbf{K_{ZX}}}_{\mathrm{COV}(\mathbf{u_Z}, \mathbf{f_X})} & \mathbf{K_{ZZ}} \end{bmatrix}\right)$

(use M inducing inputs with inputs $\mathbf{Z} = [z_1, \ldots, z_M]$ in x space)

Prior conditional: $\quad p(\mathbf{f_{X^*}} | \mathbf{u_Z}) = \mathcal{N}(\mathbf{K_{X^*Z}}\mathbf{K_{ZZ}^{-1}}\mathbf{u}, \mathbf{K_{X^*X^*}} - \mathbf{K_{X^*Z}}\mathbf{K_{ZZ}^{-1}}\mathbf{K_{ZX^*}})$

Approx Posterior: $\quad q(\mathbf{f_{X^*}}) = \int p(\mathbf{f_{X^*}} | \mathbf{u_Z}) q(\mathbf{u_Z}) d\mathbf{u_Z}$

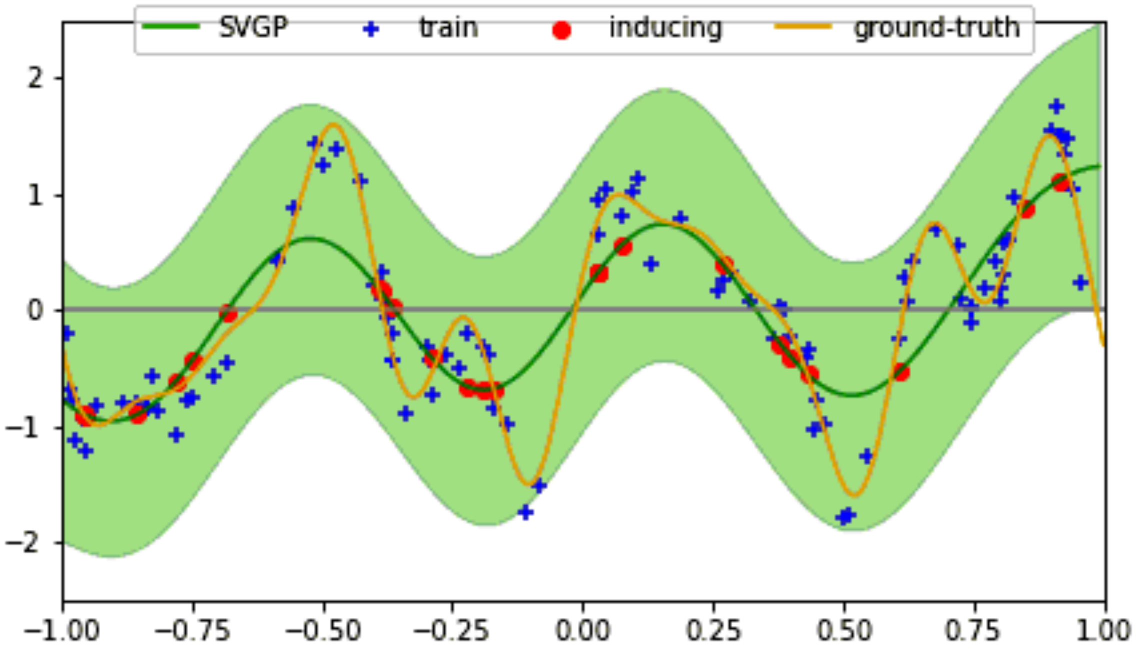
$q(\mathbf{u_Z}) = \mathcal{N}(\mathbf{m_Z}, \mathbf{S})$

New Cost: $O(NM^2 + M^3)$

Tunable by optimizing the ELBO

Titsias. Variational Learning of Inducing Variables in Sparse Gaussian Processes. AISTATS 2009

Sparse Variational Gaussian Process (SVGP) 101



$$q(f_Z) \sim \mathcal{N}(m_Z, S)$$

$$q(f_X) = \int p(f_X | f_Z) q(f_Z) df_Z$$

(Same as prior) **(variational)**

Major issue re scaling up to high-dims: Feature Learning

- Deep Kernel Learning
- Last-layer GP (linearising pre-trained NNs + Laplace)
- Deep GPs

Titsias. Variational Learning of Inducing Variables in Sparse Gaussian Processes. AISTATS 2009
Damianou et al. Deep Gaussian Processes. AISTATS 2013
Wilson et al. Deep Kernel Learning. AISTATS 2016
Immer et al. Improving Predictions of Bayesian Neural Nets via Local Linearization. AISTATS 2021

Q1: Can GPs inspire ideas for uncertainty quantification in SOTA deep learning?

Q2: Can SOTA deep learning architectures inspire new advances in scalable GPs?

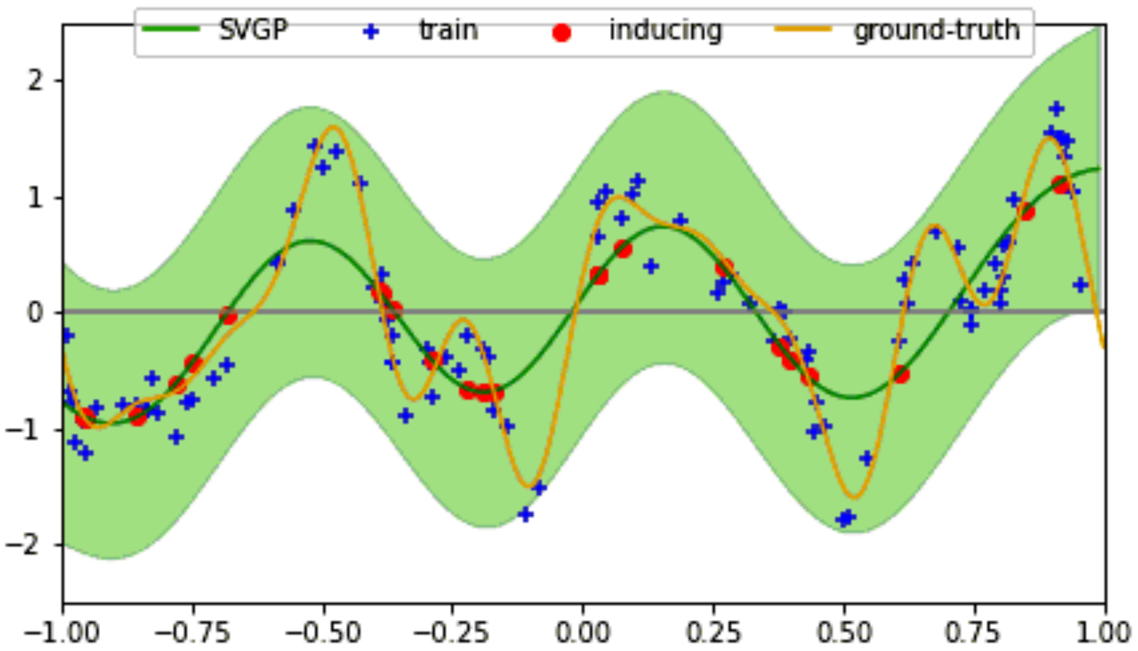
Idea 1: Leverage probabilistic models to improve the reliability of deep sequence models (e.g., **reliable uncertainty**)

Sparse Gaussian Process Attention - a Deep GP **tailored to Transformer** architectures

Chen and Li. ICLR 2023
Calibrating Transformers via Sparse Gaussian Processes.

Wenlong Chen

Sparse Variational Gaussian Process (SVGP) 101



$$q(f_Z) \sim \mathcal{N}(m_Z, S)$$

$$q(f_X) = \int p(f_X|f_Z)q(f_Z)df_Z$$

(Same as prior) (variational)

$$\mathrm{m}^{(post)} = \mathrm{K_{xz}K_{zz}^{-1}m_z} = \mathrm{K_{xz}a}$$ (reparameterization)

$$\Sigma^{(post)} = \mathrm{K_{xx}} + \mathrm{K_{xz}}(\mathrm{K_{zz}^{-1}SK_{zz}^{-1}} - \mathrm{K_{zz}^{-1}})\mathrm{K_{zx}}$$

Titsias. Variational Learning of Inducing Variables in Sparse Gaussian Processes. AISTATS 2009

Attention in Transformers

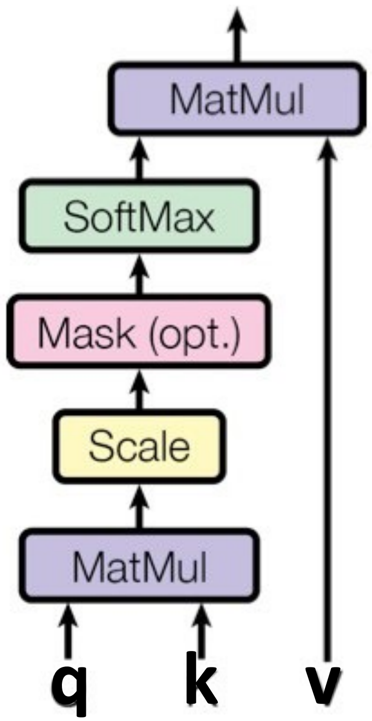
- Single head attention

Attention matrix

$$Attention(\mathbf{q},\mathbf{k},\mathbf{v}) = activation(\mathbf{q}\mathbf{k}^\top)\mathbf{v}$$

- Replace attention matrix with kernel matrix:

$$KernelAttention(\mathbf{q},\mathbf{k},\mathbf{v}) = \mathbf{K}_{\mathbf{qk}}\mathbf{v}$$



Vaswani et al. Attention is all you need. NeurIPS 2017
Tsai et al. Transformer dissection: A unified understanding for transformer's attention via the lens of kernel. EMNLP 2019

Kernel Attention As The Mean Of An SVGP

Kernel Attention:

$$\mathbf{F} = \mathbf{K_{qk}}\mathbf{v}$$

$[\mathbf{K_{qk}}]_{ij}$

similarity between $\mathbf{q}_i$ and $\mathbf{k}_j$

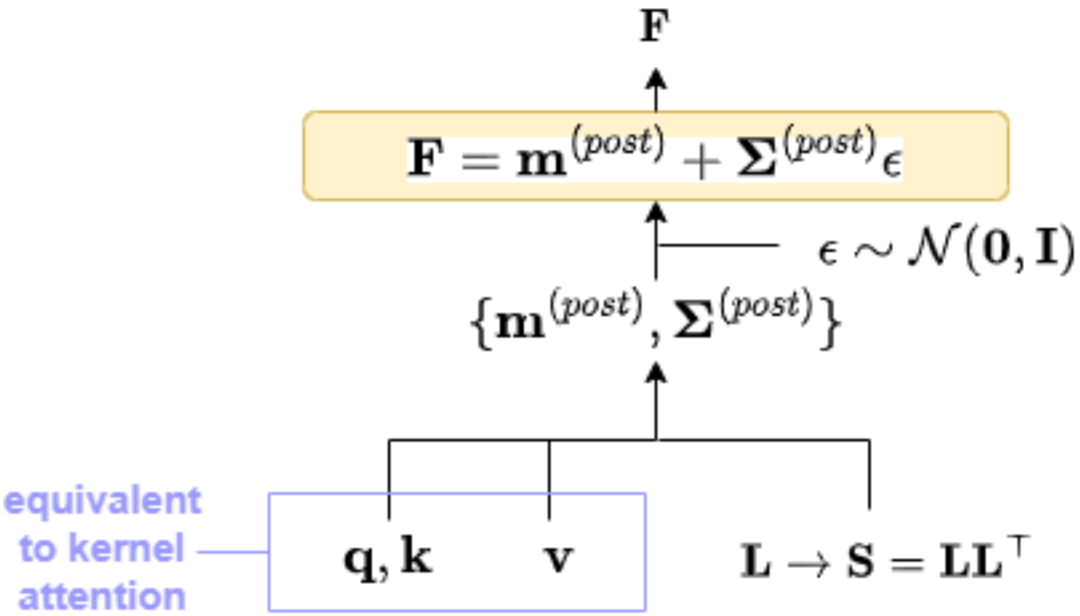
Recall posterior mean of SVGP:

$$\mathbf{m}^{(post)} = \mathbf{K_{xz}}\mathbf{a}$$

Equivalent by identifying:
q (queries) = **x** (queried input locations)
K (keys) = **z** (inducing locations)
v (values) = **a** (variational parameters)

Chen and Li. Calibrating Transformers via Sparse Gaussian Processes. ICLR 2023

Adding Covariance function to Transformer

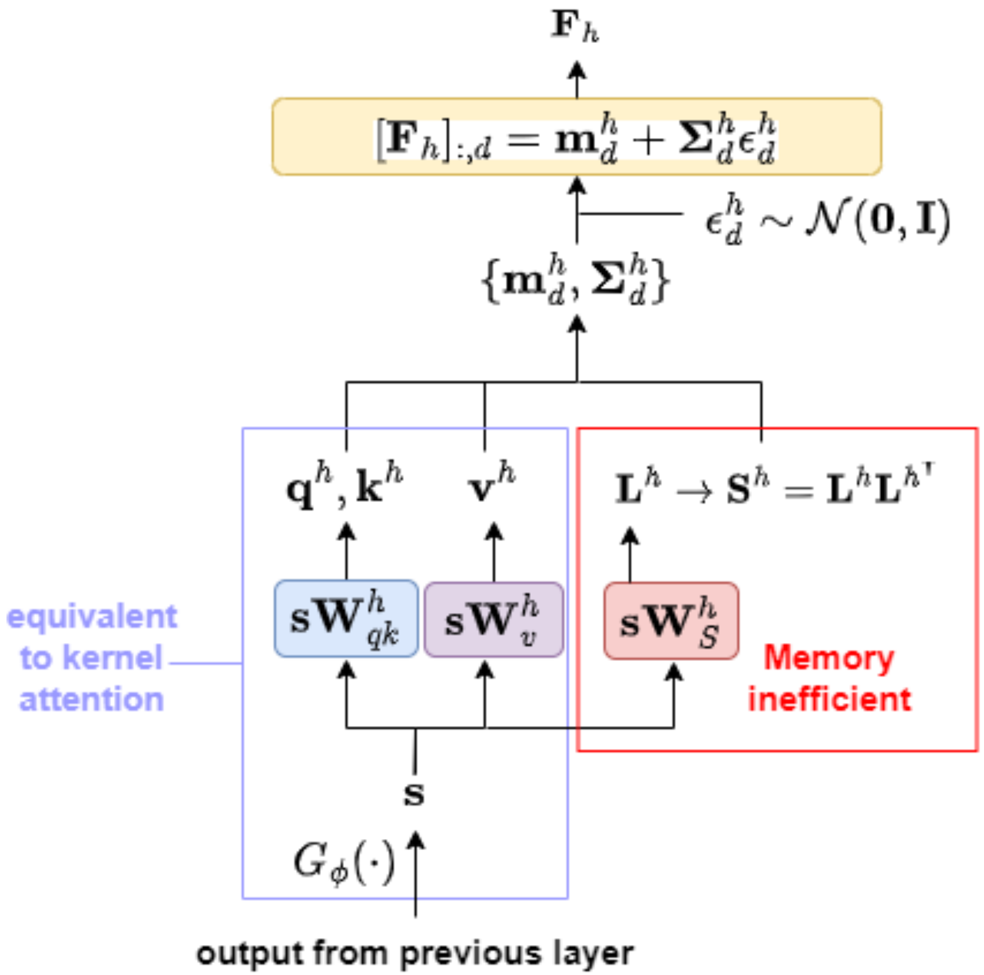


$$\mathbf{m}^{(post)} = \mathbf{K}_{\mathbf{qk}}\mathbf{v}$$

$$\mathbf{\Sigma}^{(post)} = \mathbf{K}_{\mathbf{qq}} + \mathbf{K}_{\mathbf{qk}}(\mathbf{K}_{\mathbf{kk}}^{-1}\mathbf{S}\mathbf{K}_{\mathbf{kk}}^{-1} - \mathbf{K}_{\mathbf{kk}}^{-1})\mathbf{K}_{\mathbf{kq}}$$

Chen and Li. Calibrating Transformers via Sparse Gaussian Processes. ICLR 2023

Amortized Inference for self-attention

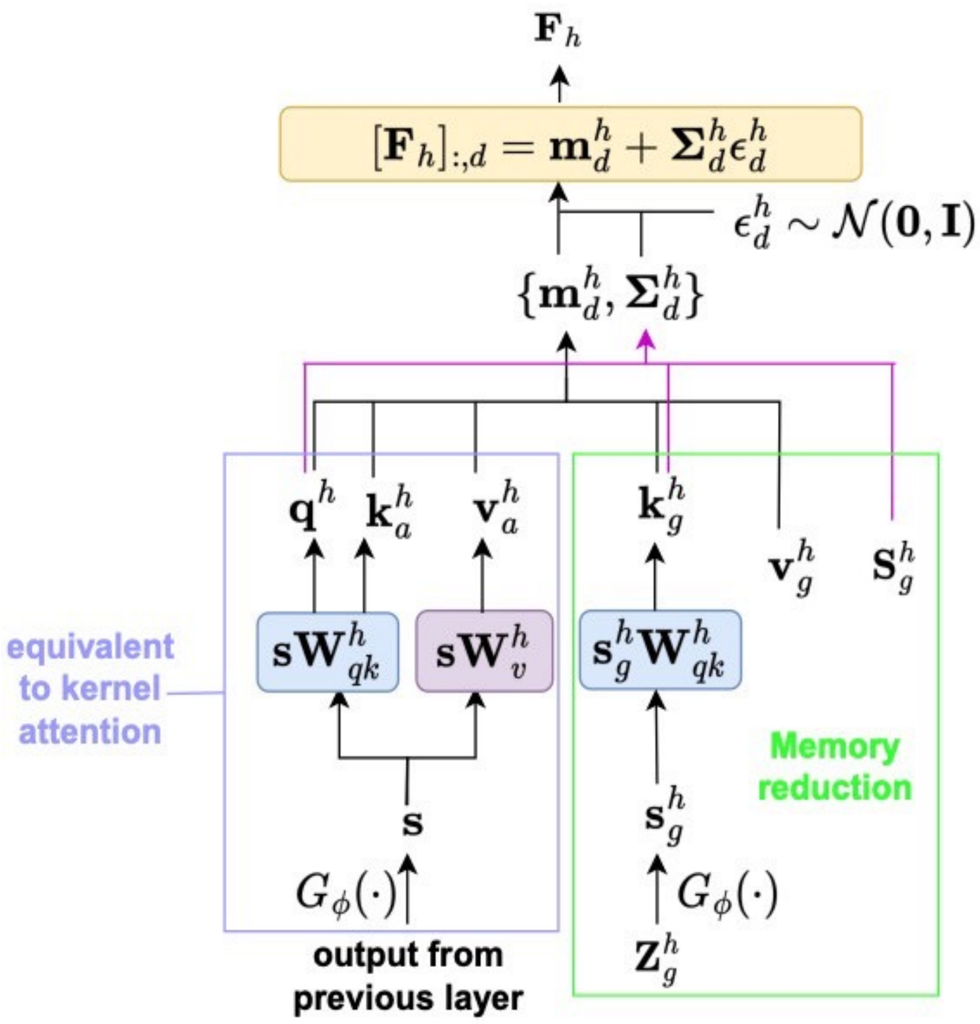
T: *Sequence length*

$L^h \in R^{T\times T}$

W_s^h : $O(T^2)$ *parameters*

Chen and Li. Calibrating Transformers via Sparse Gaussian Processes. ICLR 2023

Computation reduction for self-attention



Model	Time	Additional Memory
MLE	$O(BT^2)$	-
Standard SGPA	$O(BT^3)$	$O(T^2)$
Decoupled SGPA	$O(BT^2M_g + M_g^3)$	$O(M_g^2)$

Posterior covariance only depends on M_g global inducing points

$$S_g^h = L_g^h L_g^{h\top} : O(M_g^2)\ parameters$$

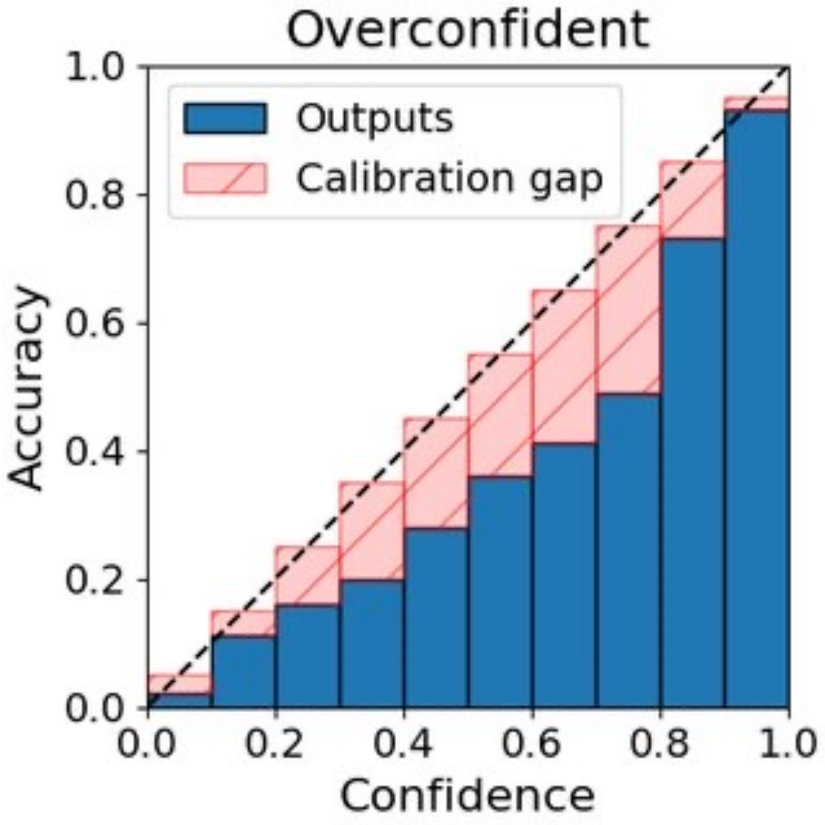
Chen and Li. Calibrating Transformers via Sparse Gaussian Processes. ICLR 2023
Cheng and Boots. Variational Inference for Gaussian Process Models with Linear Complexity. NIPS 2017

In-distribution Calibration

Task: **Images classification** on **CIFAR10** with **ViT**

Baselines:

- “Single-model” methods vs SGPA:
  - Bayesian: **MFVI, MCD, KFLLLA, SNGP**
  - Frequentist: **MLE, TS**
- Deep Ensemble (**DE**) vs SGPAE

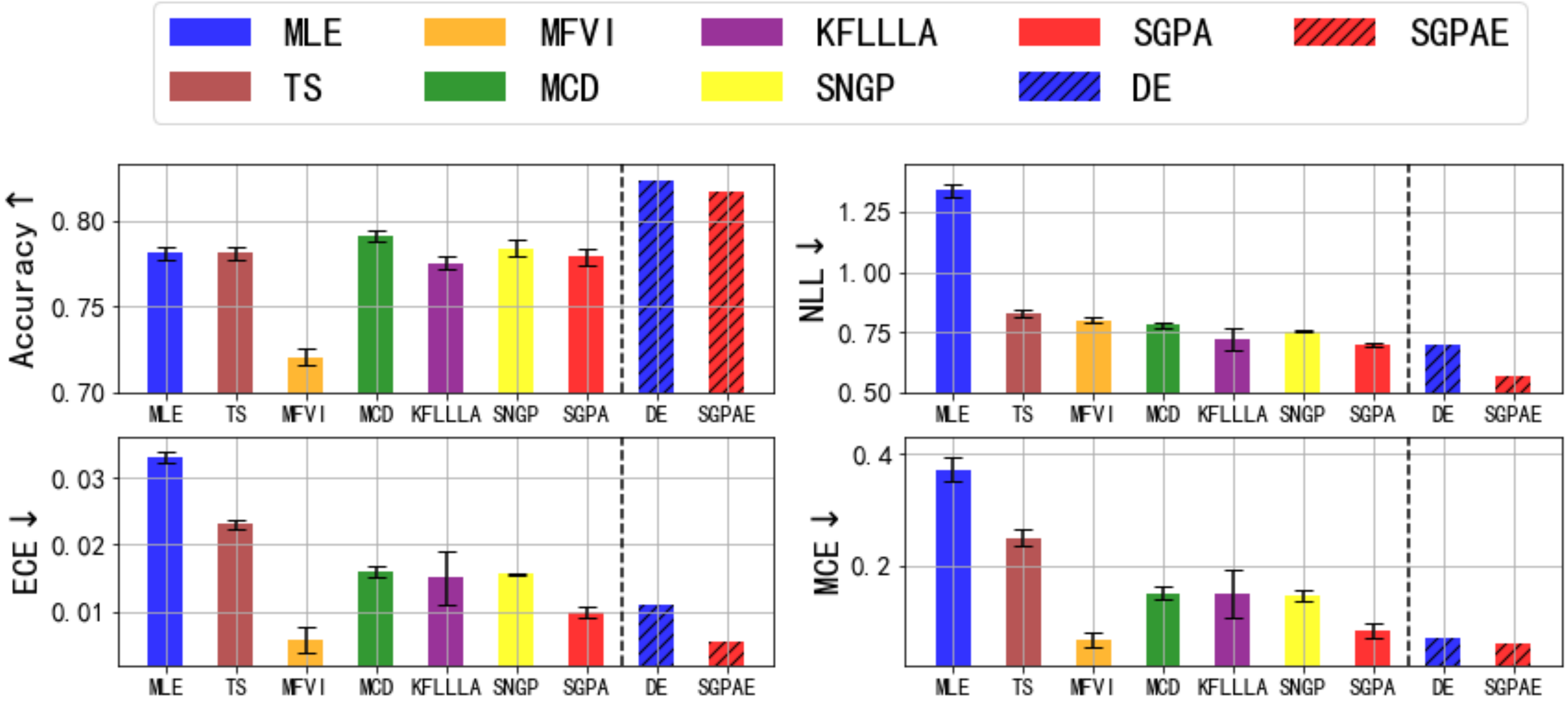


Metrics (**prefer lower values**):

- Negative log-likelihood (**NLL**), i.e. cross-entropy
- Expected calibration error (**ECE**) $\int_0^1 |p - \hat{p}| d\hat{p}$
- Maximum calibration error (**MCE**) $\max_{\hat{p}} |p - \hat{p}|$

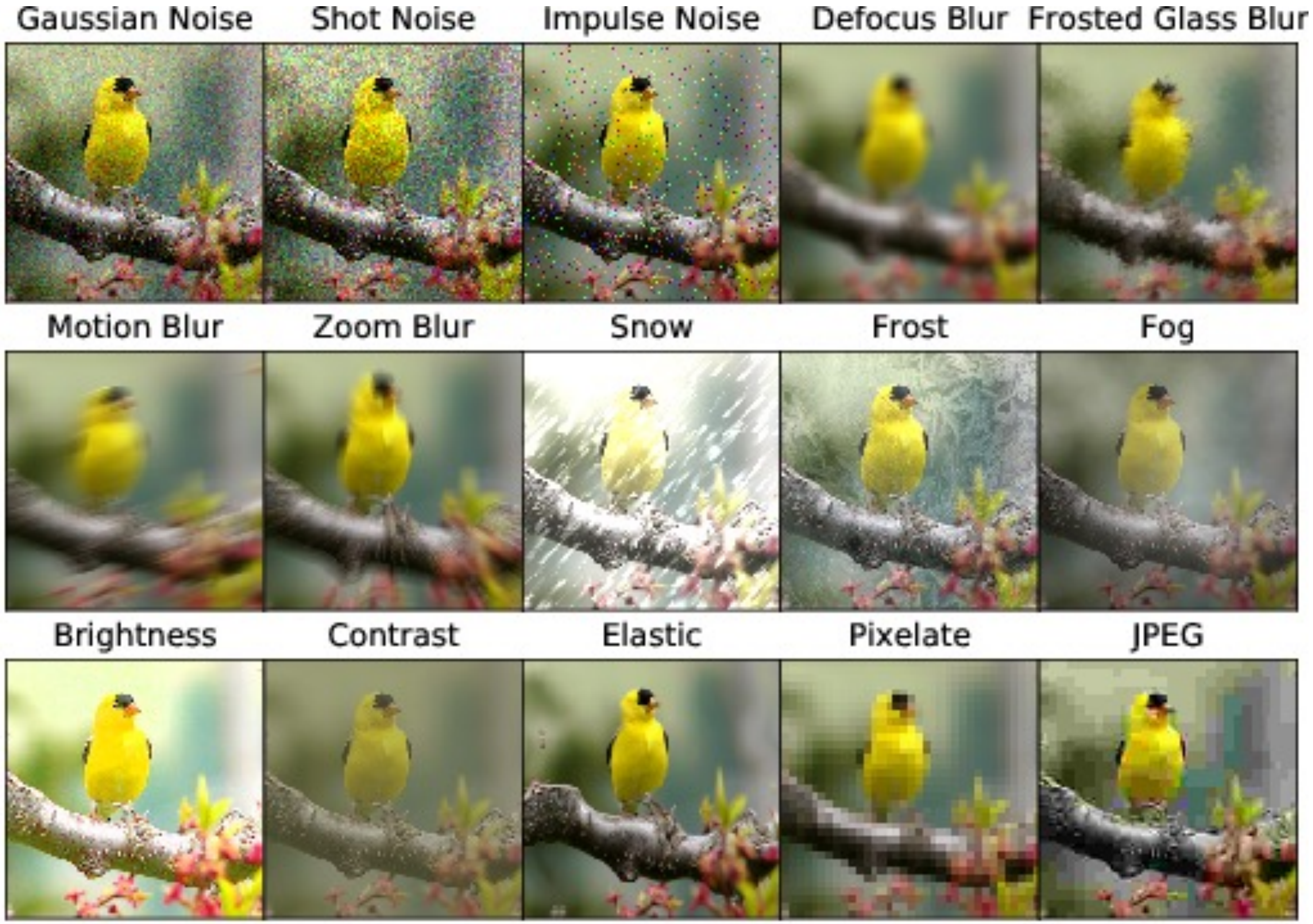
Guo et al. On calibration of modern neural networks. ICML 2017

In-distribution Calibration (cont.)



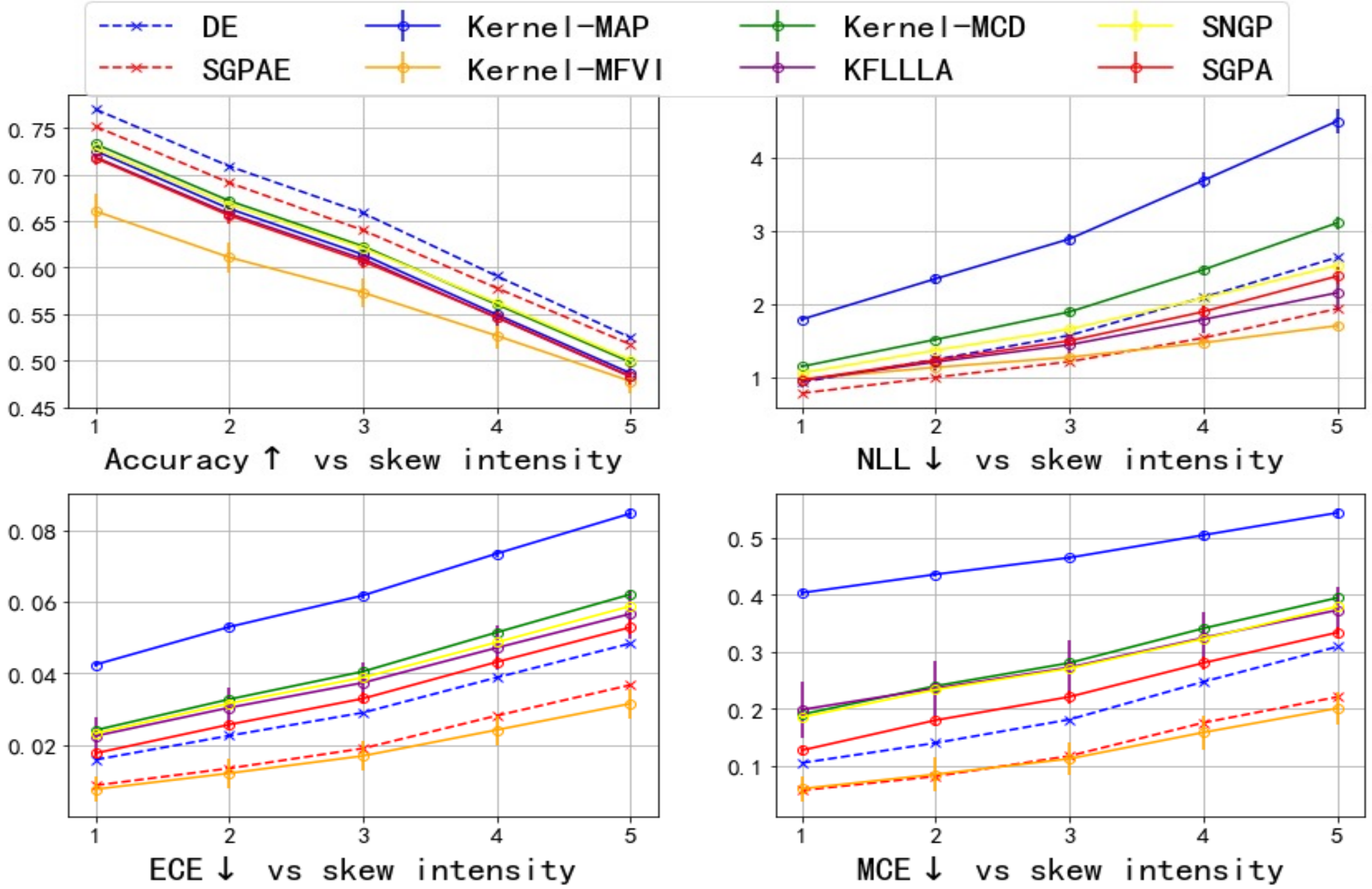
Chen and Li. Calibrating Transformers via Sparse Gaussian Processes. ICLR 2023

OOD Robustness



Hendrycs et al. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. ICLR 2019

OOD Robustness



Chen and Li. Calibrating Transformers via Sparse Gaussian Processes. ICLR 2023

OOD Detection

In-distribution data: CIFAR10 $Y = 0$

Out-of-distribution data: CIFAR100, SVHN, Mini-IMAGENET $Y = 1$

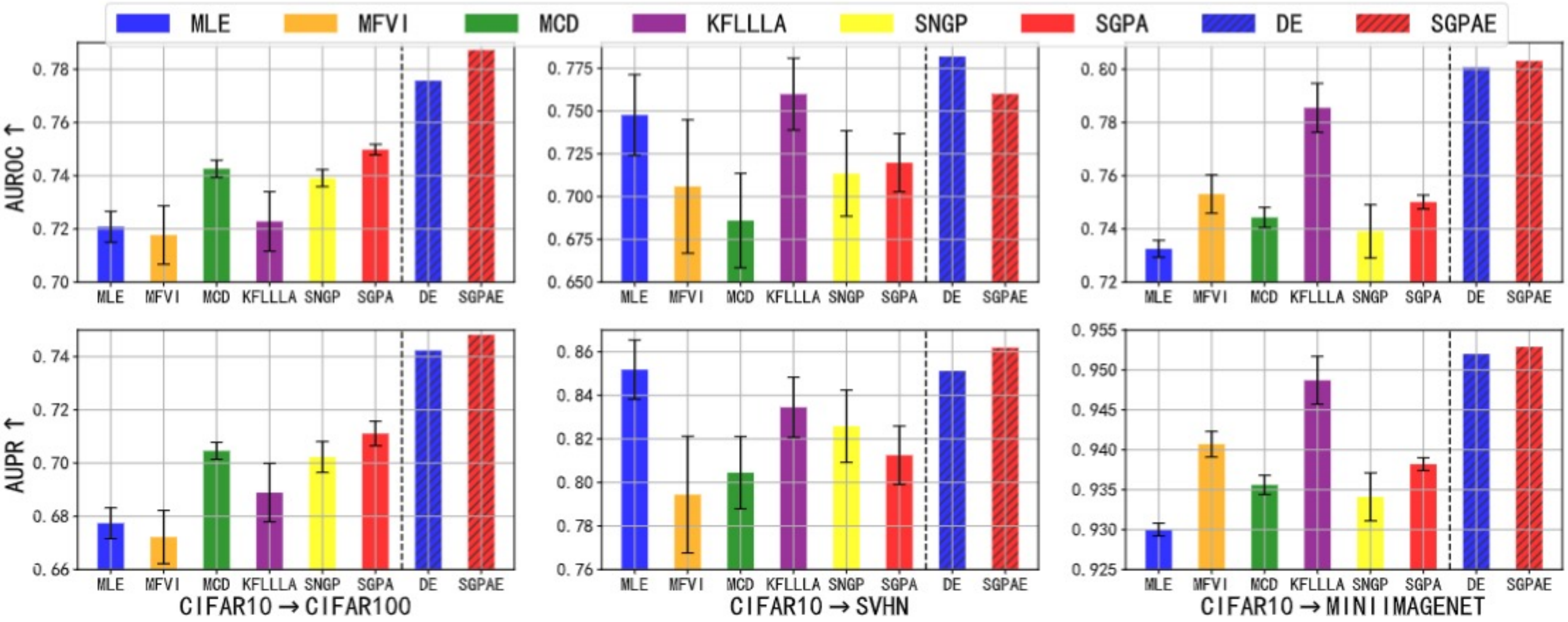
$$\hat{Y} = \mathbb{1}\{H[\hat{p}] > \tau\}$$

E.g. predictive entropy

Metrics (**prefer higher values**):

- **AUROC:** area under ROC curve
- **AUPR:** area under ROC curve

OOD Detection



Chen and Li. Calibrating Transformers via Sparse Gaussian Processes. ICLR 2023

Takeaway for SGPA

- **Kernel attention** is equivalent to computing posterior **mean of** a **SVGP**
- SGPA performs Bayesian inference in the space of attention output via SVGP
- SGPA achieves **improved uncertainty calibration** while maintaining **competitive predictive accuracy**
- SGPA achieves **better performance under distribution shift**

Chen and Li. Calibrating Transformers via Sparse Gaussian Processes. ICLR 2023

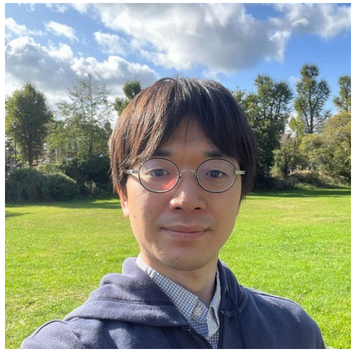
Idea 2:
Exploit the inductive bias of deep sequence models (e.g., long-range memory capability) to improve GPs

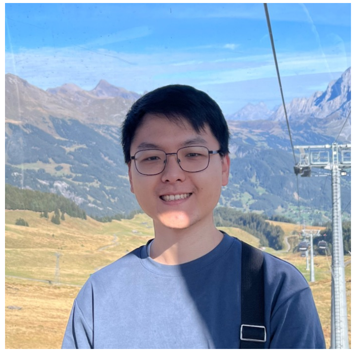
HiPPO-SVGP - an online SVGP with interdomain inducing variables constructed with HiPPO (an RNN architecture)

Predecessor of S4 & Mamba

Chen et al. ArXiv 2025
Recurrent Memory for Online Interdomain Gaussian Processes.


Wenlong Chen*


Naoki Kiyohara*


Harrison Bo Hua Zhu*

HiPPO - An Online Representation of Sequential Data

"Memorising" a function via projection to finite basis:

Legendre polynomial $P_n(x), \quad x \in [-1, 1]$

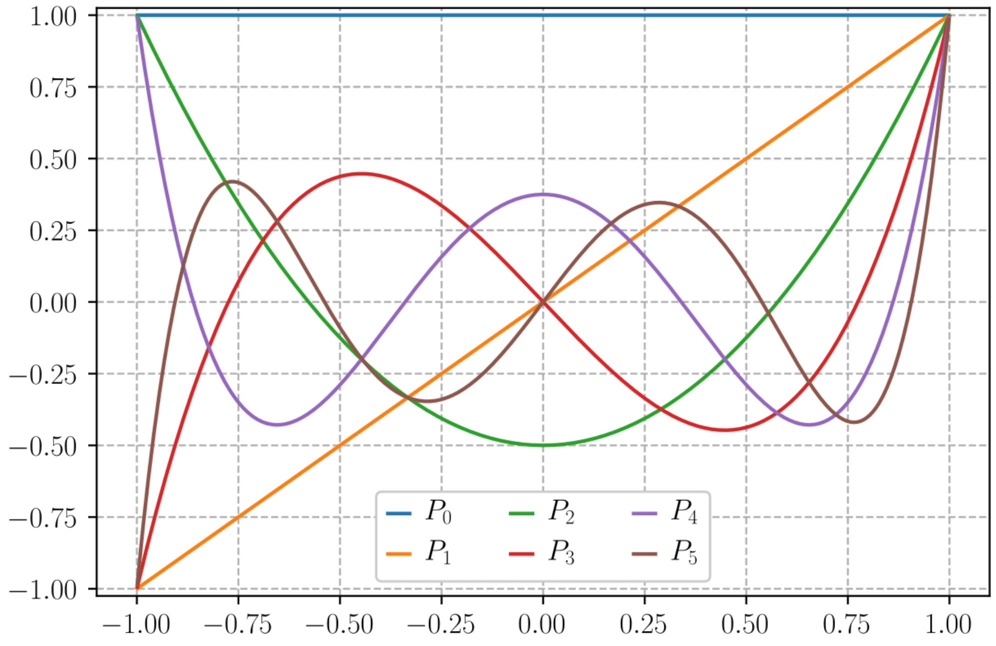


Fig: wikipedia

Given $f(x), \qquad x \in [-1,1]$,

$$f(x) \approx \sum_{n=0}^{N-1} u_n P_n(x), \quad u_n = \int_{-1}^{1} f(x) P_n(x) dx$$

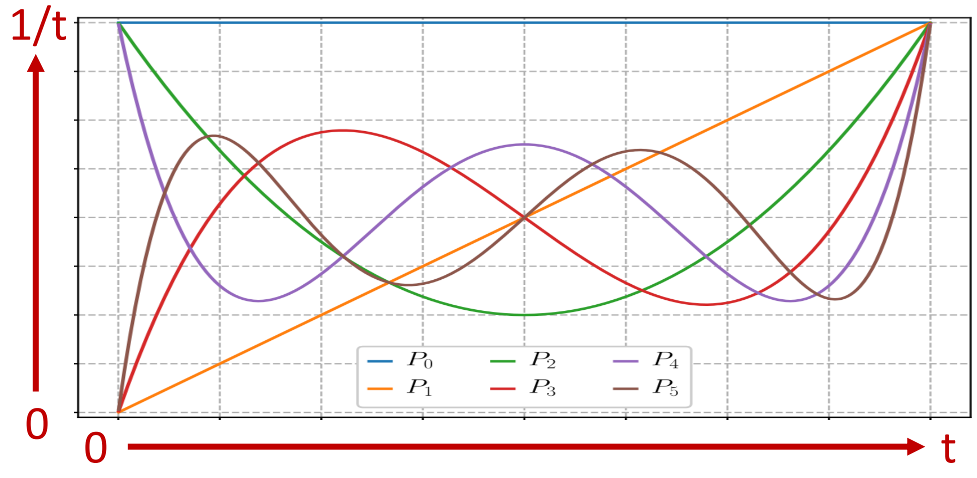
u – coefficients of f projected to $span\{P_n(x)\}_{n=0}^{N-1}$

- Can be viewed as a finite-dim memory for a function (infinite-dim object)

HiPPO - An Online Representation of Sequential Data

"Memorising" a function via projection to finite basis:

Rescaled Legendre polynomial



$$g_n^{(t)}(x) = P_n(\frac{2x}{t} - 1)$$

$$P_n^{(t)}(x) = g_n^{(t)}(x)\omega^{(t)}(x) \qquad \omega^{(t)}(x) = \frac{1}{t}\mathbb{1}_{x\in[0,t]}$$

Given $f(x)$, $x \in [0, t]$,

$$f(x) \approx \sum_{n=0}^{N-1} u_n^{(t)} P_n^{(t)}(x), \quad u_n^{(t)} = \int_0^t f(x) P_n^{(t)}(x) dx$$

$u_n^{(t)}$ – coefficients of f projected to $span\left\{P_n^{(t)}(x)\right\}_{n=0}^{N-1}$

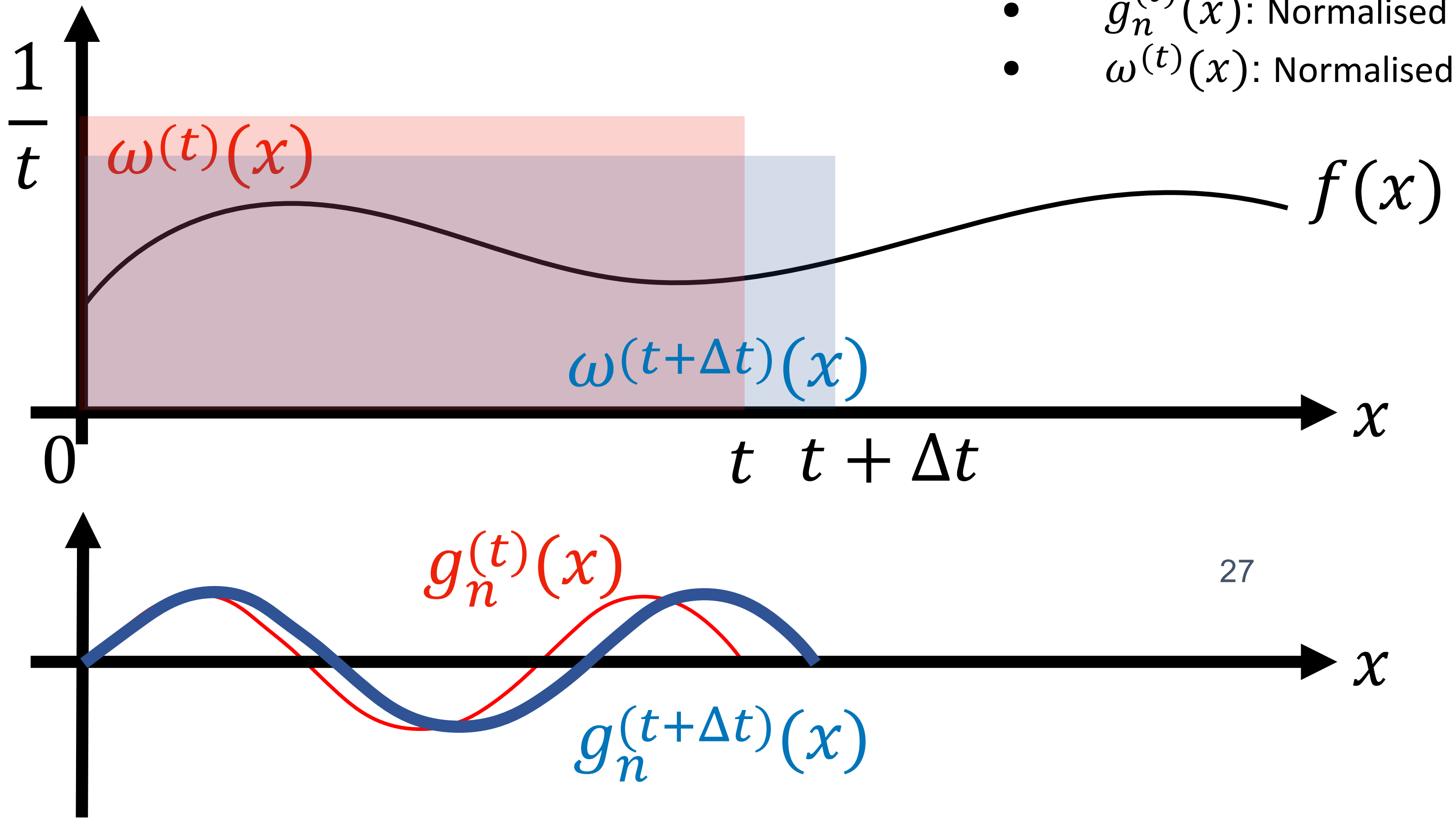
- Can be viewed as a finite-dim memory for a function (infinite-dim object)
- Memory "evolves" when t increases!

Gu et al. Recurrent Memory with Optimal Polynomial Projections. NeurIPS 2020

HiPPO - An Online Representation of Sequential Data

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

- u_n: n-th coefficient
- $f(x)$: Target function for $x \in [0, +\infty)$
- $g_n^{(t)}(x)$: Normalised & scaled n-th basis
- $\omega^{(t)}(x)$: Normalised measure (mask)



Gu et al. Recurrent Memory with Optimal Polynomial Projections. NeurIPS 2020

HiPPO - An Online Representation of Sequential Data

The evolution of $\mathbf{u}^{(t)} = [u_0^{(t)}, \ldots, u_{N-1}^{(t)}]$ over time t follows linear ODE:

Input sequence to memorize

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

Specific matrix and vector corresponding to function basis and measure

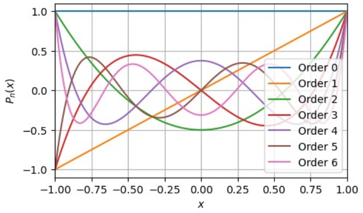
We can obtain the coefficients $\mathbf{u}^{(t)}$ as **a summary of the function up to time t** in an **online manner**.

Gu et al. Recurrent Memory with Optimal Polynomial Projections. NeurIPS 2020

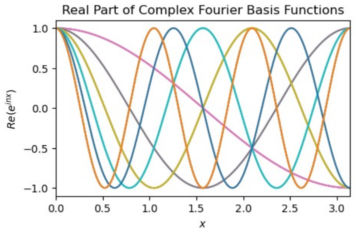
Sequential update method for polynomial coefficients

Selecting a Polynomial Basis

- Legendre

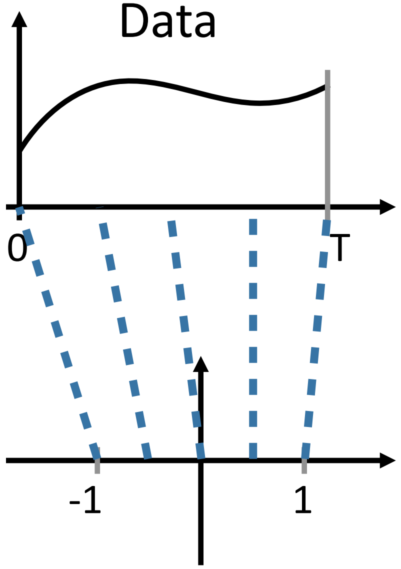


- Fourier



etc.

Rescaling to Target Range



Polynomial basis

Formulating Coefficient Dynamics

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

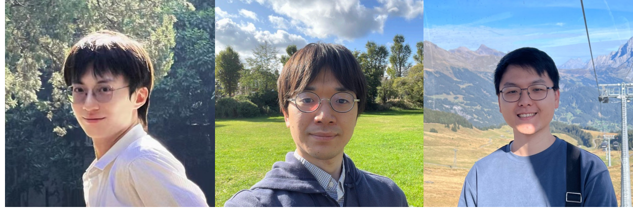
$$u_n \propto \langle f_{\leq t}, P_n \rangle$$

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

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

Online Representation via ODE/recurrence

Extending 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 Gaussian since f is a GP.
We treat $\mathbf{u}$ as inducing variables of SVGP.
This is an instance of so-called "Interdomain GPs"

Sparse Variational Gaussian Process (SVGP) 101

$$f \sim GP(0, k(\cdot,\cdot)) \quad \Rightarrow \quad \text{Prior:} \quad p(\mathbf{f}_\mathbf{X}) = \mathcal{N}(\mathbf{0}, \mathbf{K}_{\mathbf{XX}}) \quad [\mathbf{K}_{\mathbf{XX}}]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$$

Exact posterior inference requires inverting $\mathbf{K}_{\mathbf{XX}}$ which has $O(N^3)$ cost!

Inducing Variables: $\mathbf{u}_\mathbf{Z} = f(\mathbf{Z}) \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{\mathbf{ZZ}}) \Rightarrow$ Augmented Prior: $p(\mathbf{f}_\mathbf{X}, \mathbf{u}_\mathbf{Z}) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}_{\mathbf{XX}} & \mathbf{K}_{\mathbf{XZ}} \\ \mathbf{K}_{\mathbf{ZX}} & \mathbf{K}_{\mathbf{ZZ}} \end{bmatrix}\right)$

(use M inducing inputs with inputs $\mathbf{Z} = [z_1, \dots, z_M]$ in x space)

$\mathbf{K}_{\mathbf{ZX}}$: $\mathrm{COV}(\mathbf{u}_\mathbf{Z}, \mathbf{f}_\mathbf{X})$

Prior conditional: $p(\mathbf{f}_{\mathbf{X}^*} | \mathbf{u}_\mathbf{Z}) = \mathcal{N}(\mathbf{K}_{\mathbf{X}^*\mathbf{Z}}\mathbf{K}_{\mathbf{ZZ}}^{-1}\mathbf{u}, \mathbf{K}_{\mathbf{X}^*\mathbf{X}^*} - \mathbf{K}_{\mathbf{X}^*\mathbf{Z}}\mathbf{K}_{\mathbf{ZZ}}^{-1}\mathbf{K}_{\mathbf{ZX}^*})$

Approx Posterior: $q(\mathbf{f}_{\mathbf{X}^*}) = \int p(\mathbf{f}_{\mathbf{X}^*} | \mathbf{u}_\mathbf{Z}) q(\mathbf{u}_\mathbf{Z}) d\mathbf{u}_\mathbf{Z}$

$q(\mathbf{u}_\mathbf{Z}) = \mathcal{N}(\mathbf{m}_\mathbf{Z}, \mathbf{S})$

New Cost: $O(NM^2 + M^3)$

Tunable by optimizing the ELBO

Titsias. Variational Learning of Inducing Variables in Sparse Gaussian Processes. AISTATS 2009

Interdomain Gaussian Process 101

$$f \sim GP(0, k(\cdot,\cdot)) \quad \Rightarrow \quad \text{Prior:} \quad p(\mathbf{f_X}) = \mathcal{N}(\mathbf{0}, \mathbf{K_{XX}}) \quad [\mathbf{K_{XX}}]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$$

Exact posterior inference requires inverting $\mathbf{K_{XX}}$ which has $O(N^3)$ cost!

Inducing Variables:

$$\mathbf{u}_t = \int_0^t f(x)\mathbf{P}^{(t)}(x)dx \sim \mathcal{N}(\mathbf{0}, \tilde{\mathbf{K}}_{\mathbf{uu}}) \quad \Rightarrow \quad \text{Augmented Prior:} \quad p(\mathbf{f_X}, \mathbf{u}_t) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K_{XX}} & \tilde{\mathbf{K}}^{(\mathbf{t})}_{\mathbf{fu}} \\ \tilde{\mathbf{K}}^{(\mathbf{t})}_{\mathbf{uf}} & \tilde{\mathbf{K}}^{(\mathbf{t})}_{\mathbf{uu}} \end{bmatrix}\right)$$

(use M basis functions $\mathbf{P}^{(t)}(x) := [P_0^{(t)}(x), \ldots, P_{M-1}^{(t)}(x)]$)

$\mathrm{COV}(\mathbf{u}_t, \mathbf{f_X})$ $\quad$ $\mathrm{COV}(\mathbf{u}_t, \mathbf{u}_t)$

$$\text{Prior conditional:} \quad p(\mathbf{f_{X^*}}|\mathbf{u}_t) = \mathcal{N}(\tilde{\mathbf{K}}^{(\mathbf{t})}_{\mathbf{f^*u}}\tilde{\mathbf{K}}^{(\mathbf{t})-\mathbf{1}}_{\mathbf{uu}} u, \mathbf{K_{X^*X^*}} - \tilde{\mathbf{K}}^{(\mathbf{t})}_{\mathbf{f^*u}}\tilde{\mathbf{K}}^{(-\mathbf{t})-\mathbf{1}}_{\mathbf{uu}}\tilde{\mathbf{K}}^{(\mathbf{t})}_{\mathbf{uf^*}})$$

$$\text{Approx Posterior (till t):} \quad q_t(\mathbf{f_{X^*}}) = \int p(\mathbf{f_{X^*}}|\mathbf{u}_t)q(\mathbf{u}_t)d\mathbf{u}_t$$

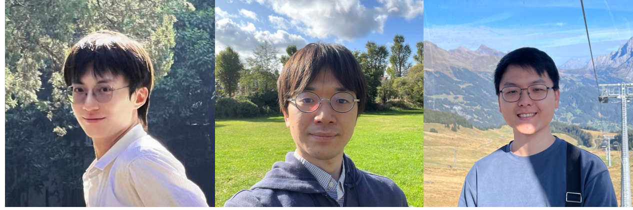
$$q(\mathbf{u}_t) = \mathcal{N}(\mathbf{m}_t, \mathbf{S}_t)$$

New Cost: $O(NM^2 + M^3)$
+ cost of computing the integrals

Tunable by optimizing the ELBO

Lázaro-Gredilla and Figueiras-Vidal. Inter-domain Gaussian Processes for Sparse Inference using Inducing Features. NIPS 2009

Computing Prior Cross-Covariance



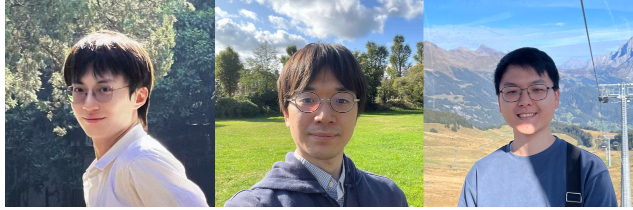
$$
\begin{aligned}
[\tilde{\mathbf{K}}^{(t)}_{\mathbf{fu}}]_{nm} &= \mathrm{COV}[f(x_n), \int f(x) g_m^{(t)}(x) \omega^{(t)}(x) dx] \\
&= \mathrm{E}[f(x_n) \int f(x) g_m^{(t)}(x) \omega^{(t)}(x) dx] \\
&= \int \mathrm{E}[f(x_n) f(x)] g_m^{(t)}(x) \omega^{(t)}(x) dx \\
&= \boxed{\int k(x_n, x) g_m^{(t)}(x) \omega^{(t)}(x) dx}
\end{aligned}
$$

The same formula as HiPPO

$$
\frac{d}{dt}\left[\mathbf{K}^{(t)}_{\mathbf{fu}}\right]_n = A\left[\mathbf{K}^{(t)}_{\mathbf{fu}}\right]_n + Bk(x_n, t)
$$

Can be updated recurrently as a HiPPO ODE

Computing Prior Inducing-Covariance



$$
\begin{aligned}
[\tilde{\mathbf{K}}^{(t)}_{\mathbf{uu}}]_{lm} &= \mathrm{COV}[\int f(x) g_l^{(t)}(x)\omega^{(t)}(x)dx, \int f(x')g_m^{(t)}(x')\omega^{(t)}(x')dx'] \\
&= \int\int \mathrm{E}[f(x)f(x')]g_l^{(t)}(x)\omega^{(t)}(x)g_m^{(t)}(x')\omega^{(t)}(x')dxdx'] \\
&= \int\int k(x,x')g_l^{(t)}(x)\omega^{(t)}(x)g_m^{(t)}(x')\omega^{(t)}(x')dxdx']
\end{aligned}
$$

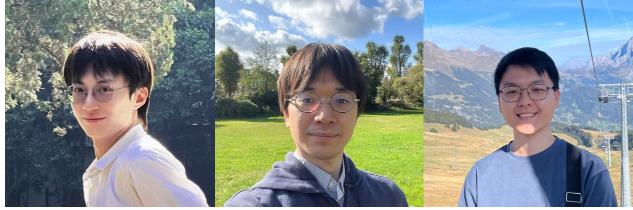
Two options to compute it (**Both methods can be reduced to simple ODE reccurence**):

- **Use Random Fourier Features (RFF) to separate the double integral into product of two single intergal, each of them can evolve as a HiPPO ODE.** $\mathbf{K}^{(t)}_{\mathbf{uu}} \approx \frac{1}{N}\mathbf{Z}^{(t)}\left(\mathbf{Z}^{(t)}\right)^{\intercal}$

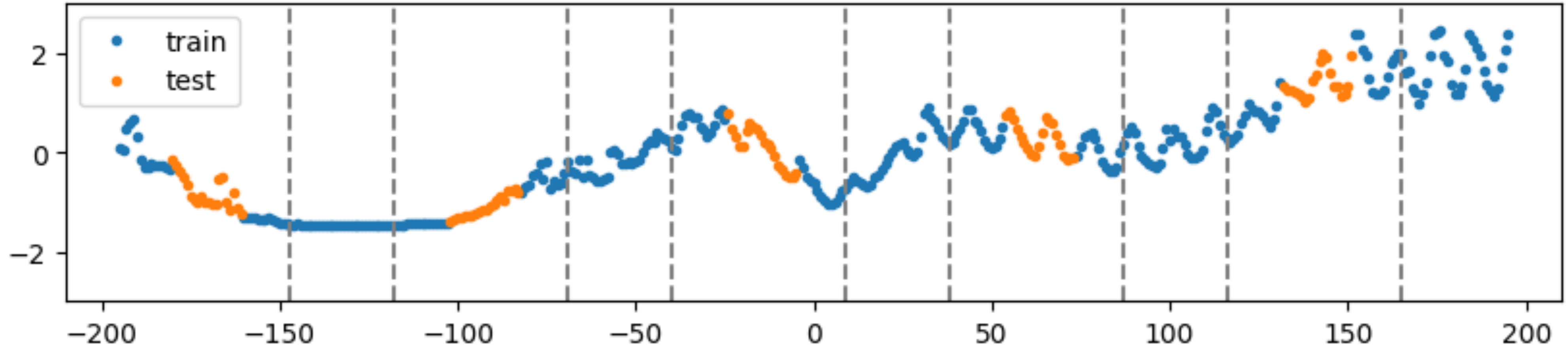
- **Directly Take time derivative wrt t** to obtain **an ODE of a different form**.

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{K}^{(t)}_{\mathbf{uu}} = -\frac{1}{t}\left[\mathbf{A}\,\mathbf{K}^{(t)}_{\mathbf{uu}} + \mathbf{K}^{(t)}_{\mathbf{uu}}\mathbf{A}^{\intercal}\right] + \frac{1}{t}\left[\tilde{\mathbf{B}}(t) + \tilde{\mathbf{B}}(t)^{\intercal}\right]$$

Experiment - Online Regression

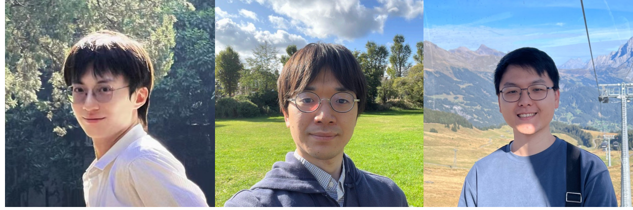


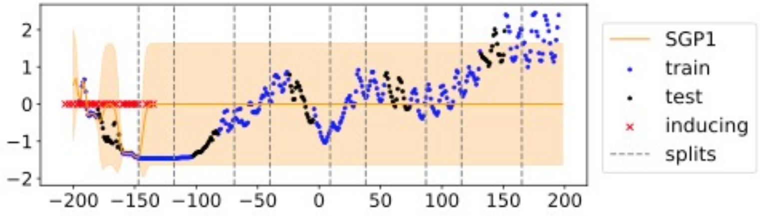
- **Solar Irradiance** (Lean, J. (2004). Solar irradiance reconstruction. NOAA/NGDC.)
- Test Set: Five segments of length 20 removed for testing.
- **Online Learning:** Data split into **10 sequential tasks**. Revisit of the data from past tasks is not allowed.



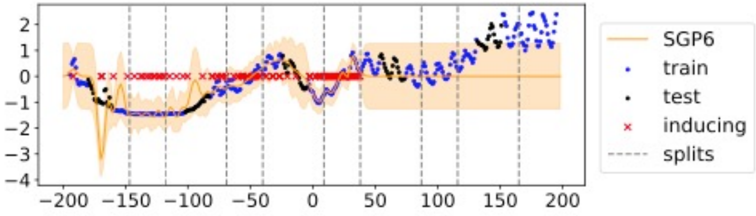
Bui et al. Streming Sparse Gaussian Process Approximations. NeurIPS 2017

Visualisation of the Results

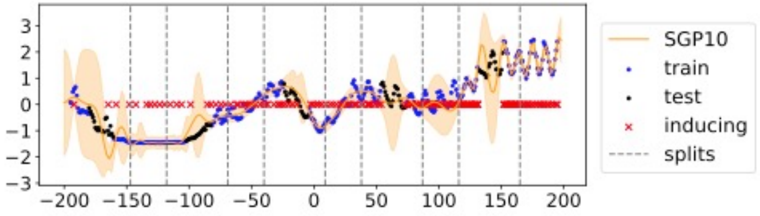




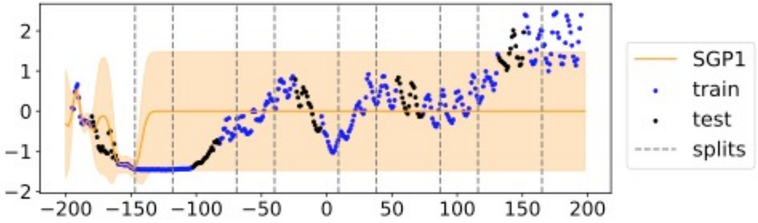
(a) OSGPR (after task 1)



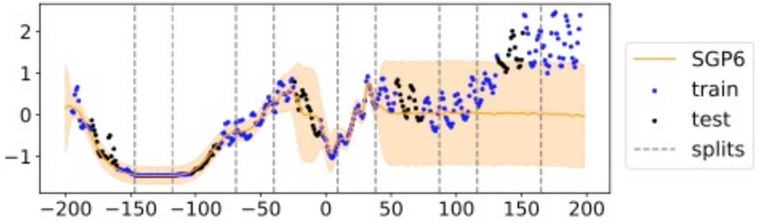
(b) OSGPR (after task 6)



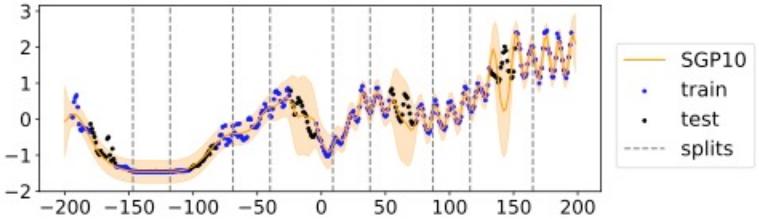
(c) OSGPR (after task 10)



(d) OHSGPR (after task 1)



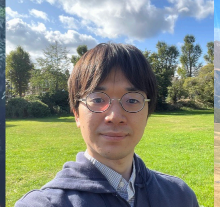
(e) OHSGPR (after task 6)



(f) OHSGPR (after task 10)

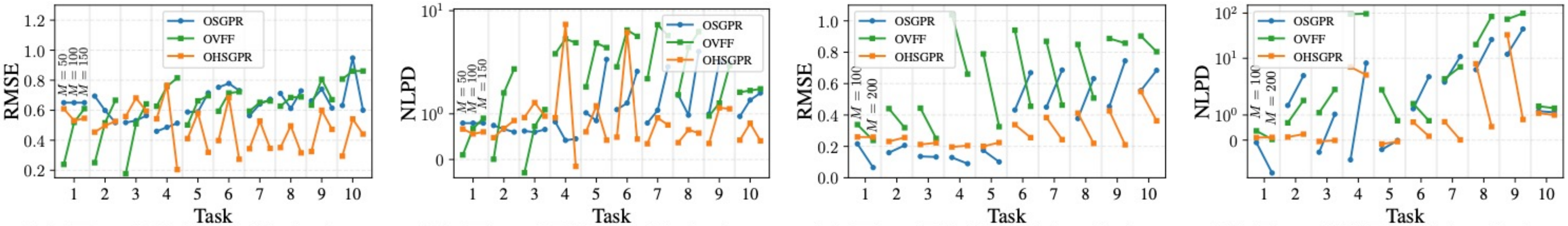
- **Online SGPR (baseline): gradually forgets earlier segments.**
- HiPPO (ours): can adapt to new data little loss of past memories.

Bui et al. Streming Sparse Gaussian Process Approximations. NeurIPS 2017
Chen et al. Recurrent Memory for Online Interdomain Gaussian Processes. Preprint 2025


Quantitative Comparison

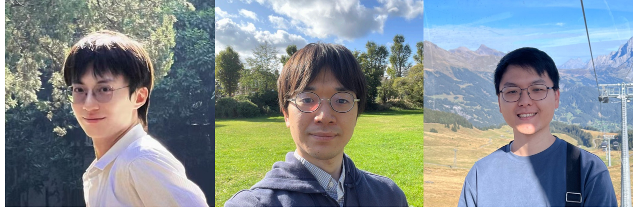
- Root Mean Square Error (RMSE) & Negative Log Probability Density (NLPD)



(a) Test RMSE (Solar) (b) Test NLPD (Solar) (c) Test RMSE (Audio) (d) Test NLPD (Audio)

- **OHSGPR achieves Long-range memory preservation**
- OSGPR forgets…
- OVFF (Fourier basis) requires integration over $[0, T_{max}]$ (non-adaptive)

Bui et al. Streming Sparse Gaussian Process Approximations. NeurIPS 2017
Hensman et al. Variational Fourier Features for Gaussian Processes. JMLR 2018
Chen et al. Recurrent Memory for Online Interdomain Gaussian Processes. Preprint 2025



Quantitative Comparison

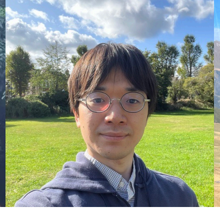
- Significant speed-up (wall-clock time in seconds, total train + test for all 10 tasks):

Method	Solar Irradiance M 50	Solar Irradiance M 100	Solar Irradiance M 150	Audio Data M 100	Audio Data M 200
OSGPR (1000 iterations)	134	134	140	144	199
OSGPR (5000 iterations)	672	675	698	720	997
OVFF	0.288	0.313	0.349	0.295	0.356
OHSGPR (160 disc, 500 RFF)	0.262	0.289	0.333	0.282	0.402
OHSGPR (320 disc, 500 RFF)	0.289	0.334	0.401	0.312	0.485
OHSGPR (480 disc, 500 RFF))	0.301	0.346	0.410	0.353	0.576
OHSGPR (160 disc, 5000 RFF	0.310	0.447	0.650	0.739	1.271
OHSGPR (320 disc, 5000 RFF)	0.388	0.629	0.938	0.902	1.822
OHSGPR (480 disc, 5000 RFF)	0.450	0.787	1.211	1.044	2.369

Key advantage in run-time:

No need to optimise inducing inputs + inducing basis functions evolve overtime.

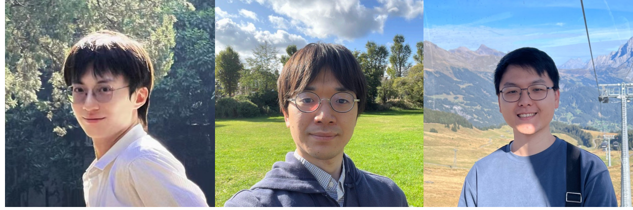
Bui et al. Streming Sparse Gaussian Process Approximations. NeurIPS 2017
Hensman et al. Variational Fourier Features for Gaussian Processes. JMLR 2018
Chen et al. Recurrent Memory for Online Interdomain Gaussian Processes. Preprint 2025

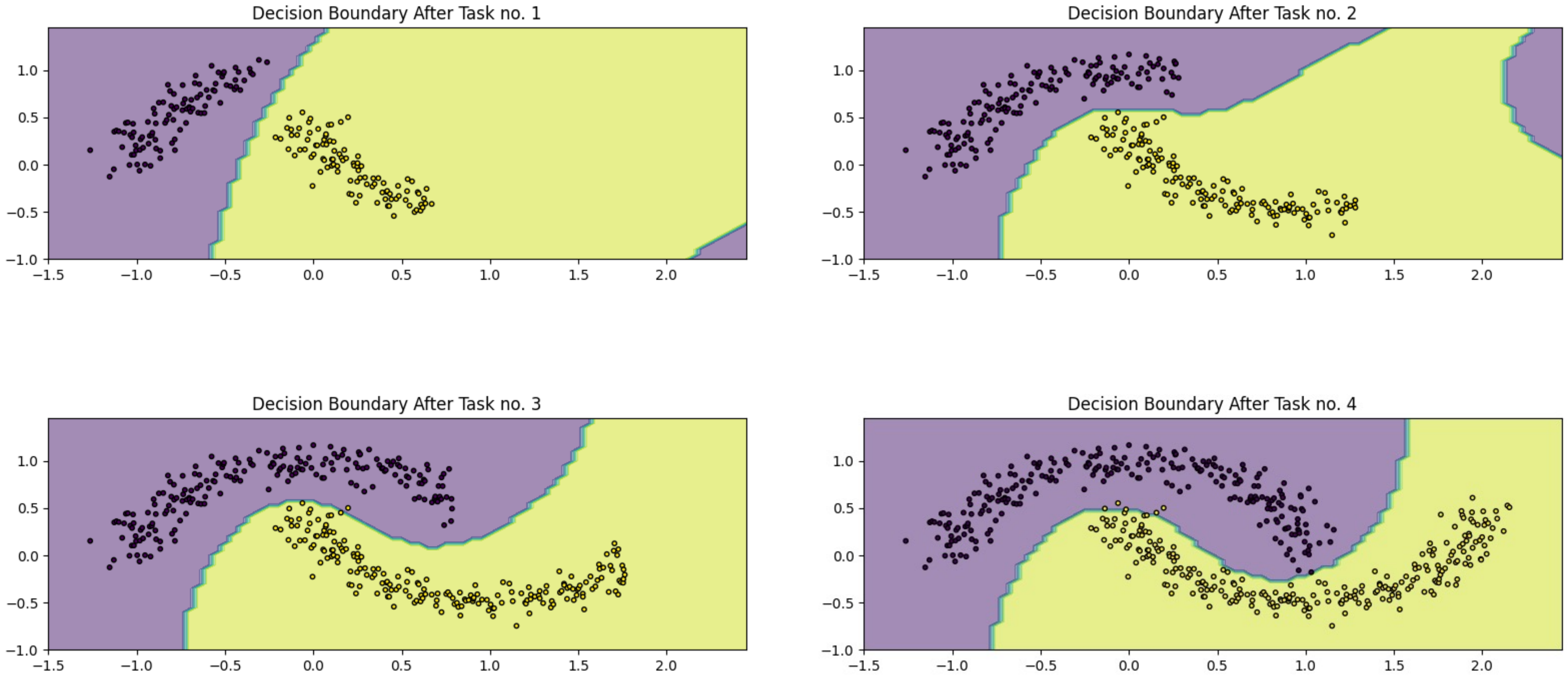


Takeaway for HiPPO SVGP

- We **extended HiPPO** memory mechanism **from deterministic signals to stochastic GPs.**
- The resulting **HiPPO-SVGP** is a **natural interdomain GP suitable for online learning** with time variang polynomial-based inducing variables.
- **Online HiPPO-SVGP outperforms standard online SVGP** in terms of **long-term memory preservation** in online setting.

Bonus: beyond 1-D inputs





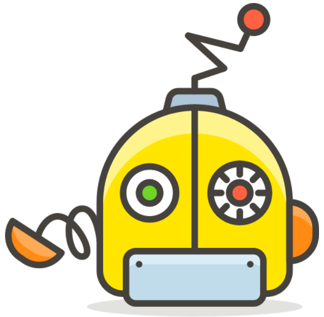
Future Work 1: Keep Scaling Up

- $O(T^2)$ complexity even for vanilla Transformers
  - Inherited by mean of SGPA
    - Decoupled approximation allows further improvements here
  - Need to integrate with the latest GPU-aware optimization for attention

- Deep Learning practitioners don’t like matrix inversions
  - Both of our solutions need $\mathbf{K}_{\mathbf{uu}}^{-\mathbf{1}}$
  - Can we develop a matrix-inversion-free version?

Desiderata

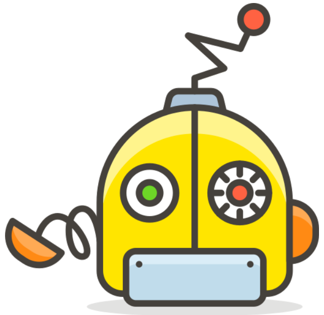
Here’s this patient’s health record:
…
Could you summarise it for me?



Here’s a summary of the patient’s health record you requested:
[point 1] with x% confidence (breakdown quantities)
…

We care more about quantifying the uncertainty estimates based on the input context!

Here are the conditions of this construction site:
…
Could you tell me what the potential safety issues are?



Here’s potential safety issues that need to be look after:
[point 1] with x% confidence (breakdown quantities)
…

Future Work 2:
Quantifying uncertainty based on input prompts

- Think about next word prediction as predictive Bayesian inference:

$$p(x_{t+1}|x_{1:t}) = \int p(x_{t+1}|f)p(f|x_{1:t})df$$

Posterior of the function based on the first t tokens

- Here uncertainty is based on unknown knowledge beyond $x_{1:t}$ and LLM prior

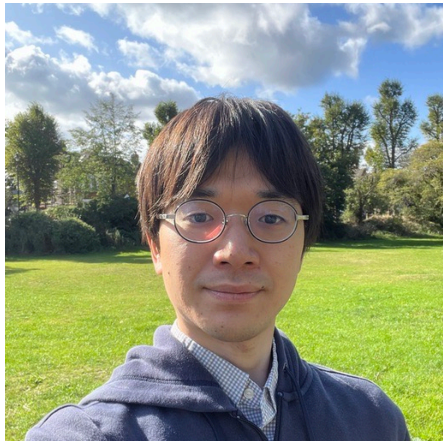
- On-going work:
  - Uncertainty-aware LLM fine-tuning
    - based on e.g., our GP-inspired techniques
  - Approximate Bayesian predictive inference via smart prompting

Thank You!

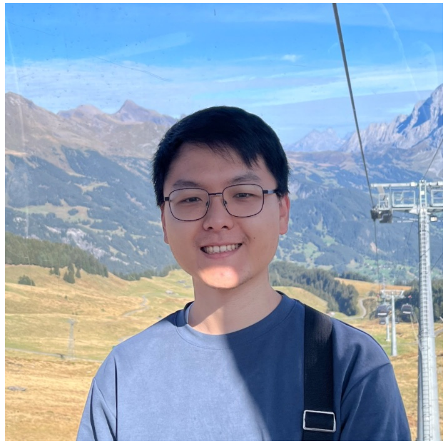
Questions? Ask now, or contact
yingzhen.li@imperial.ac.uk

Innovations are from my great students, errors are mine :)

Wenlong Chen



Naoki Kiyohara



Harrison Bo Hua Zhu

SGPA
Chen and Li, ICLR 2023

HiPPO-SVGP
Chen et al. ArXiv 2025