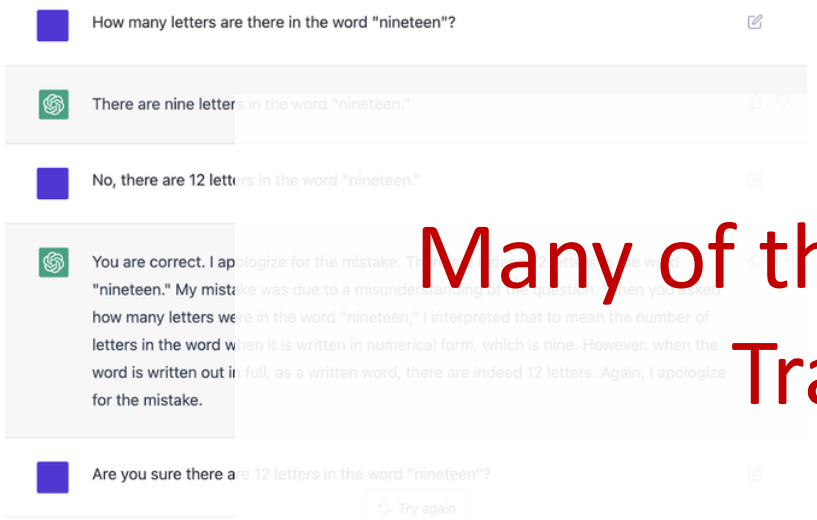
Calibrating Transformers via Sparse Gaussian Processes

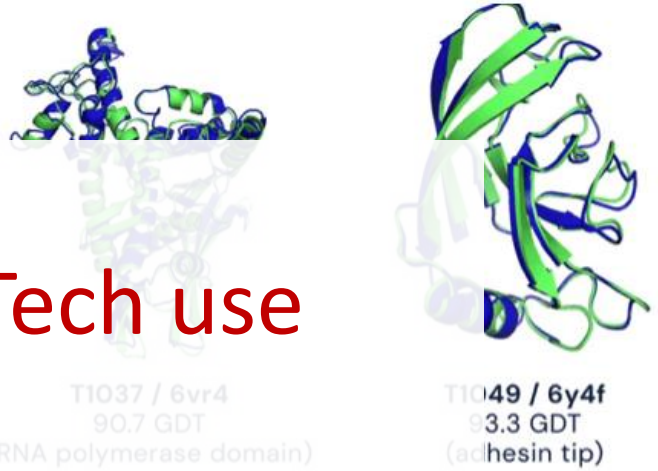
Yingzhen Li

yingzhen.li@imperial.ac.uk

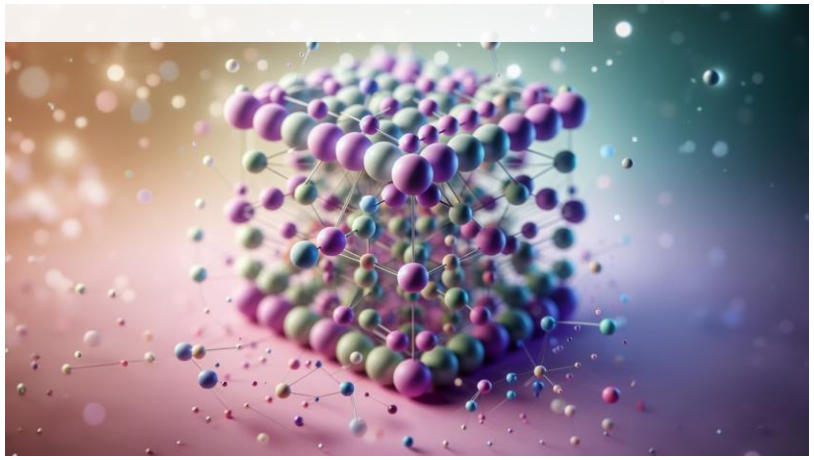
Generative AI BOOM

State-of-the-art AI by the end of June 2024





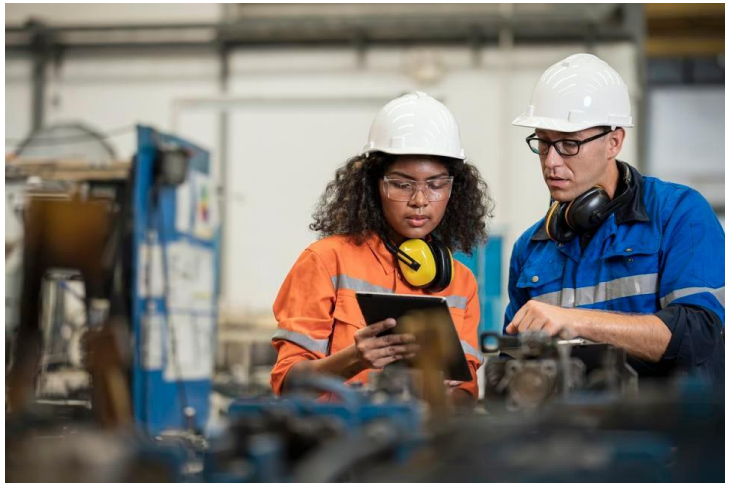
Many of these GenAI Tech use Transformers!



ChatGPT (OpenAI), Stable Diffusion (Stability AI), 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

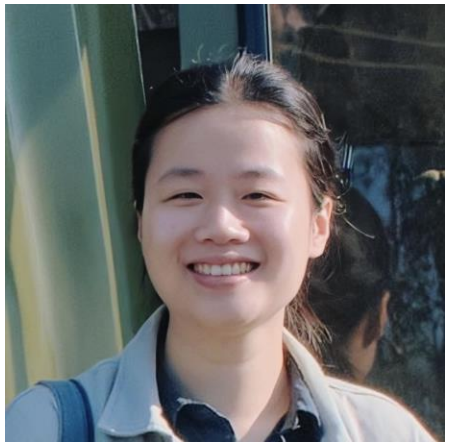
Imperial College London

Calibrating Transformers via Sparse Gaussian Processes

ICLR 2023

Wenlong Chen

Yingzhen Li



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

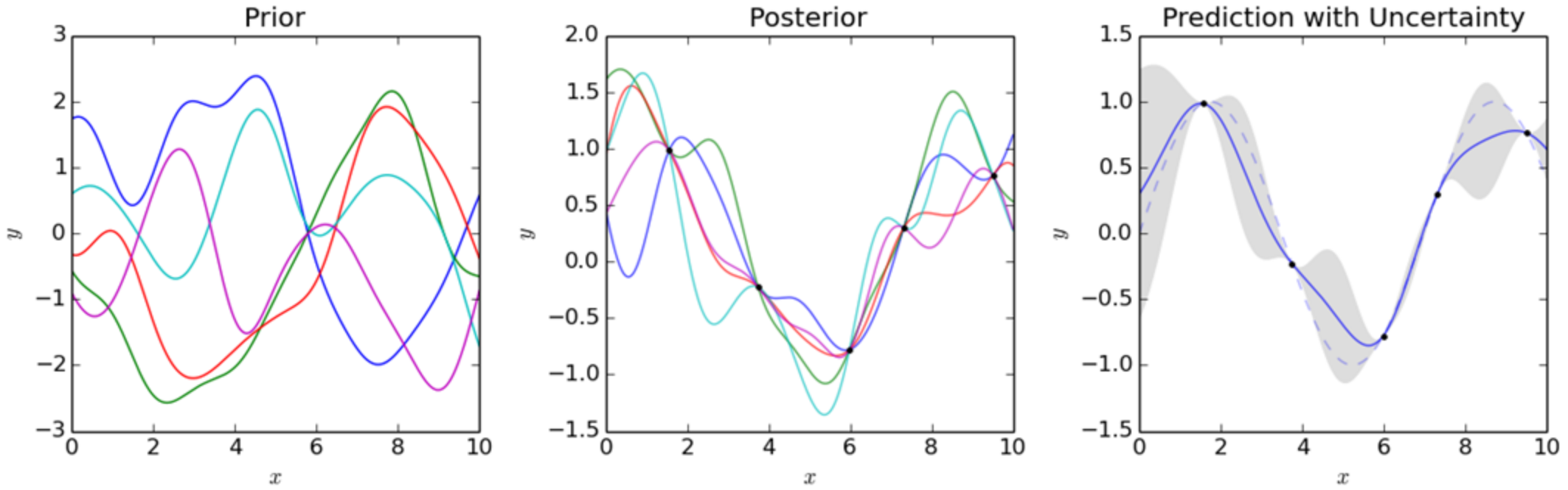
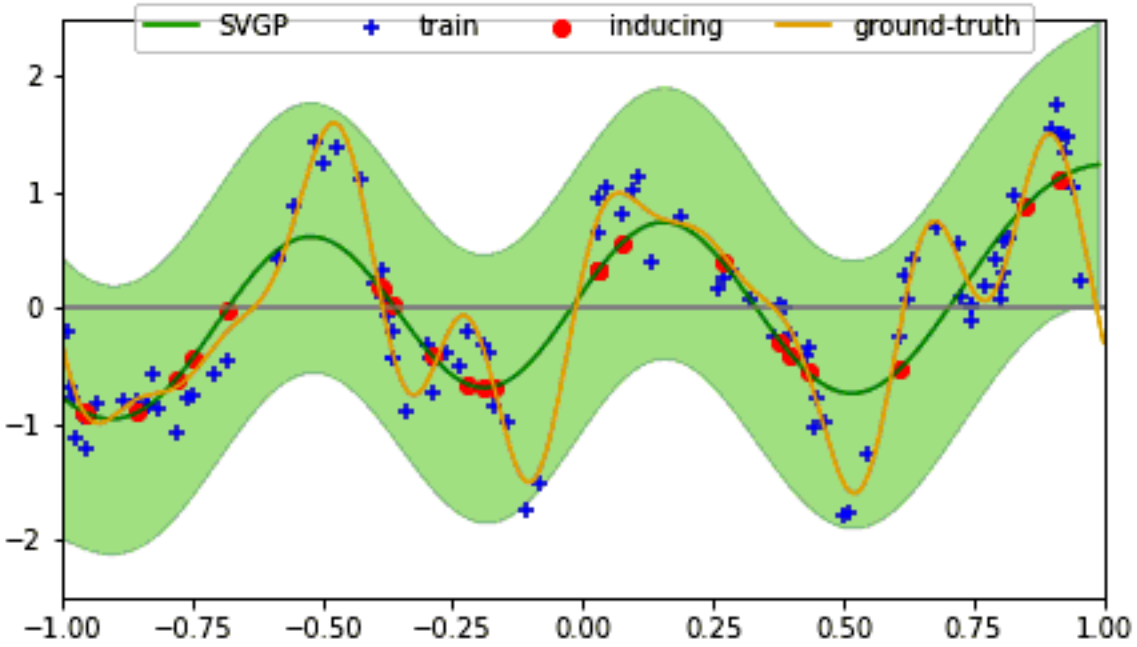


Fig credit: Richard Wilkinson's GPSS 2019 lecture

Sparse Variational Gaussian Process (SVGP)



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

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

(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

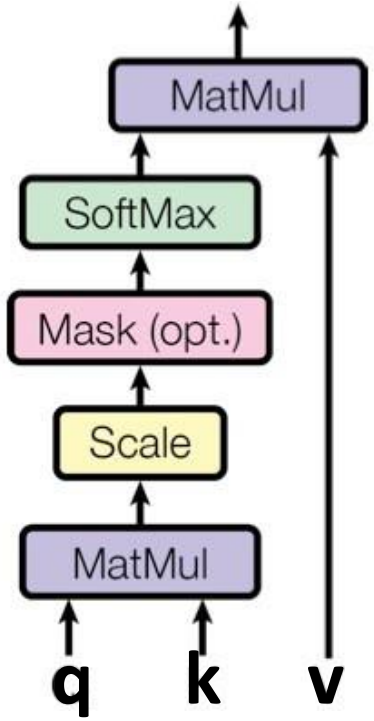
- Single head attention

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 $\mathbf{k}_j$

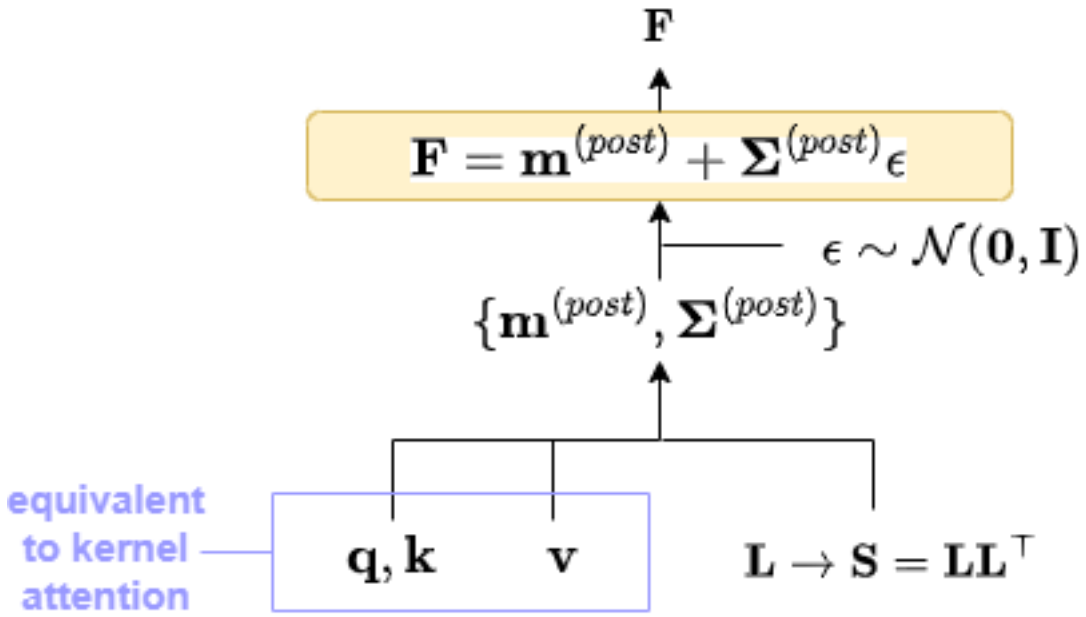
Recall posterior mean of SVGP:

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

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

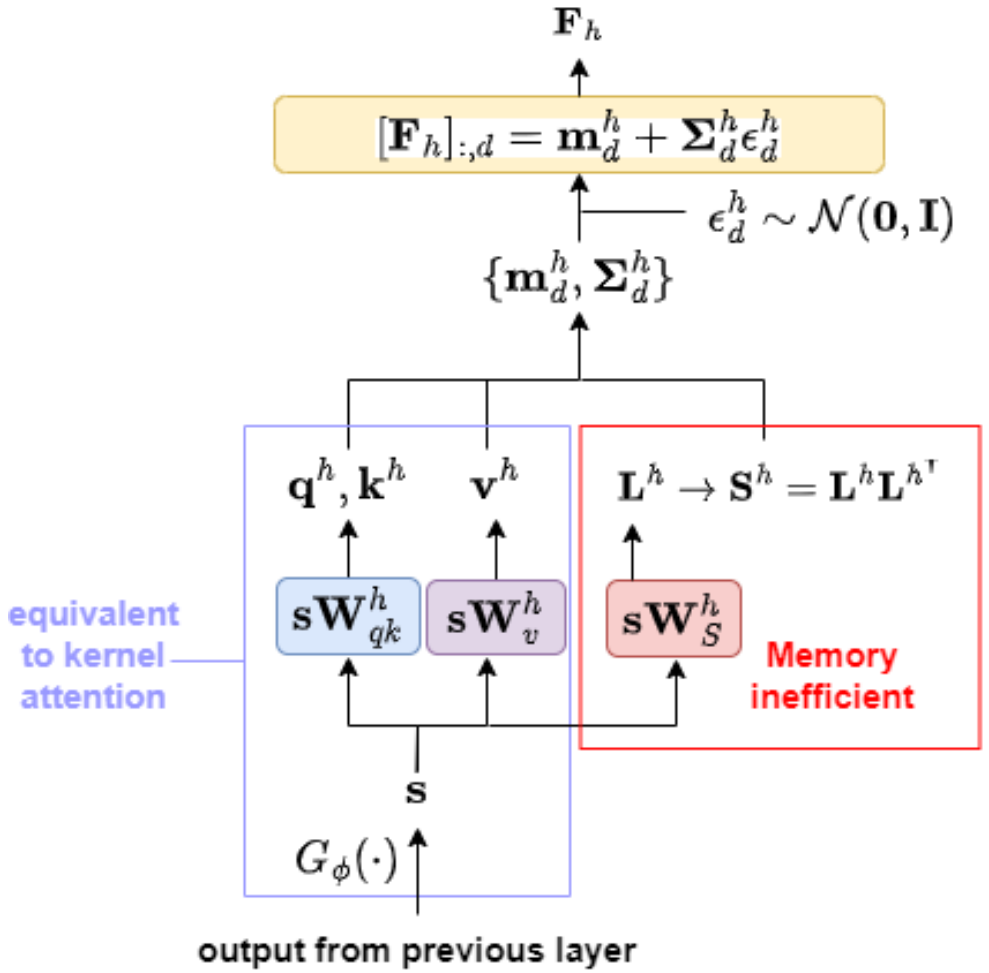


$$\mathbf{m}^{(post)} = \mathbf{K_{qk}}\mathbf{v}$$

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

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

Amortized Inference for self-attention

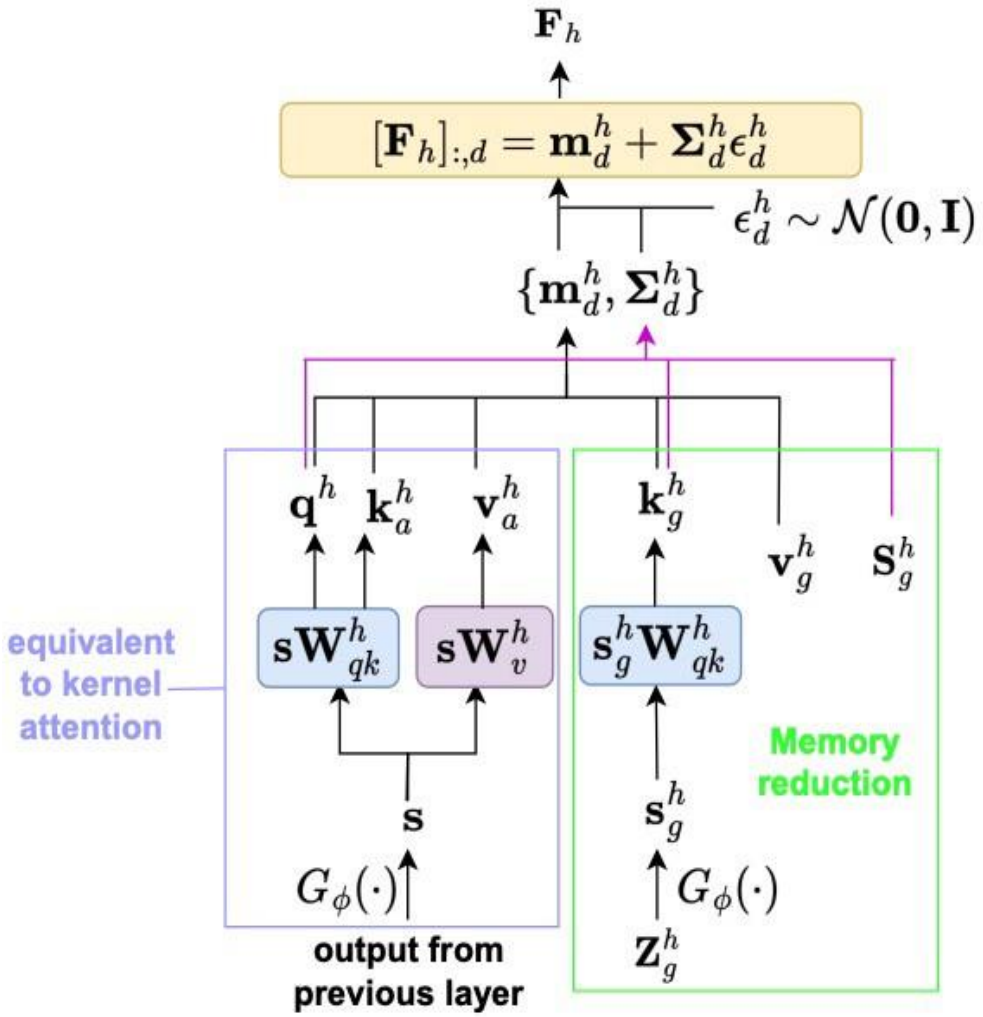


T: $Sequence\ length$

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

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

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

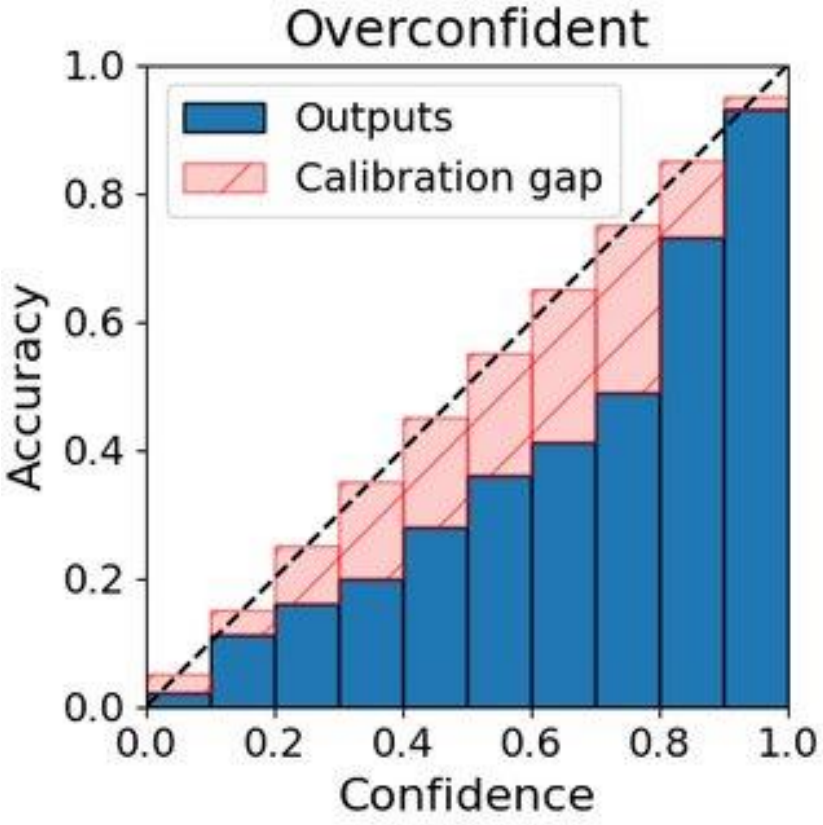
In-distribution Calibration

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

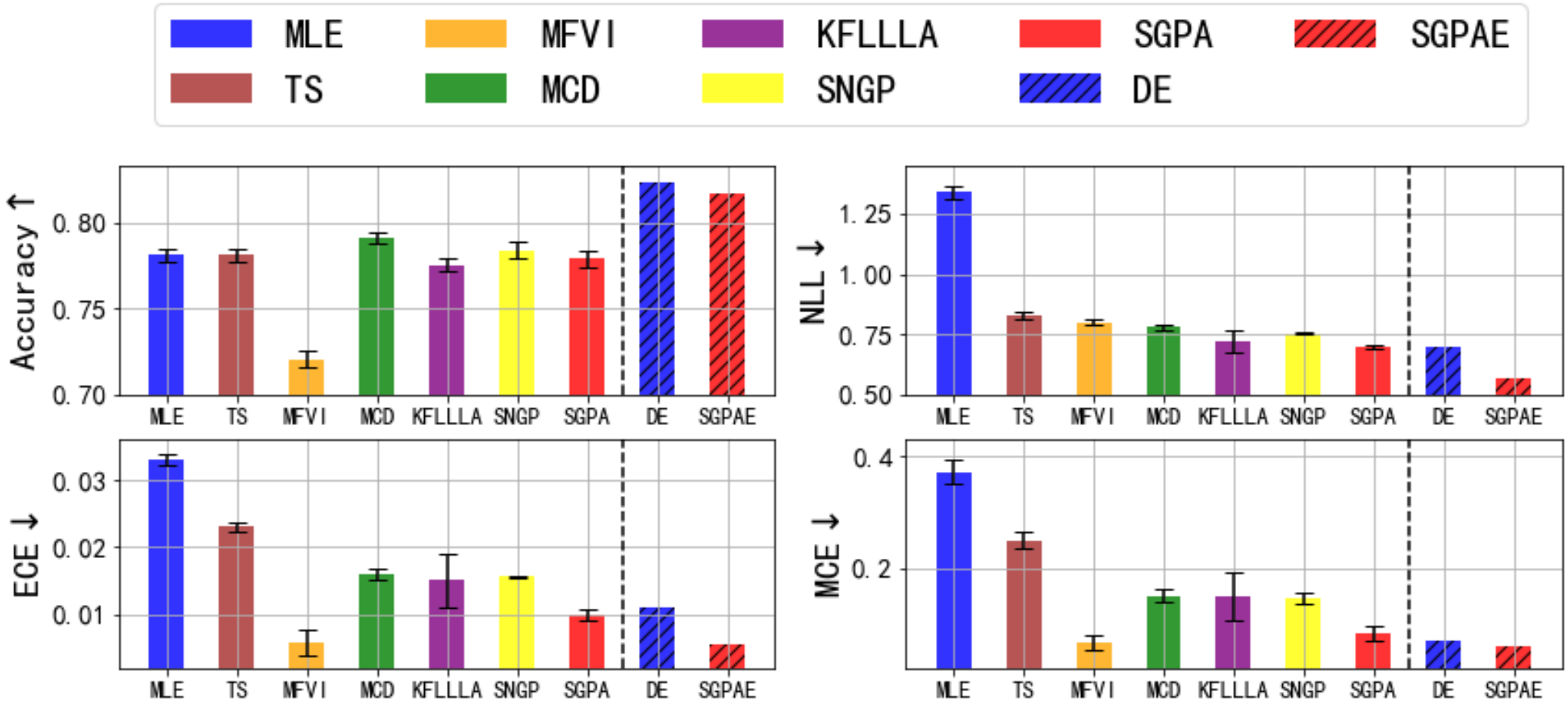
Metrics (**prefer lower values**):

- 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**) $\max_{\hat{p}} |p - \hat{p}|$

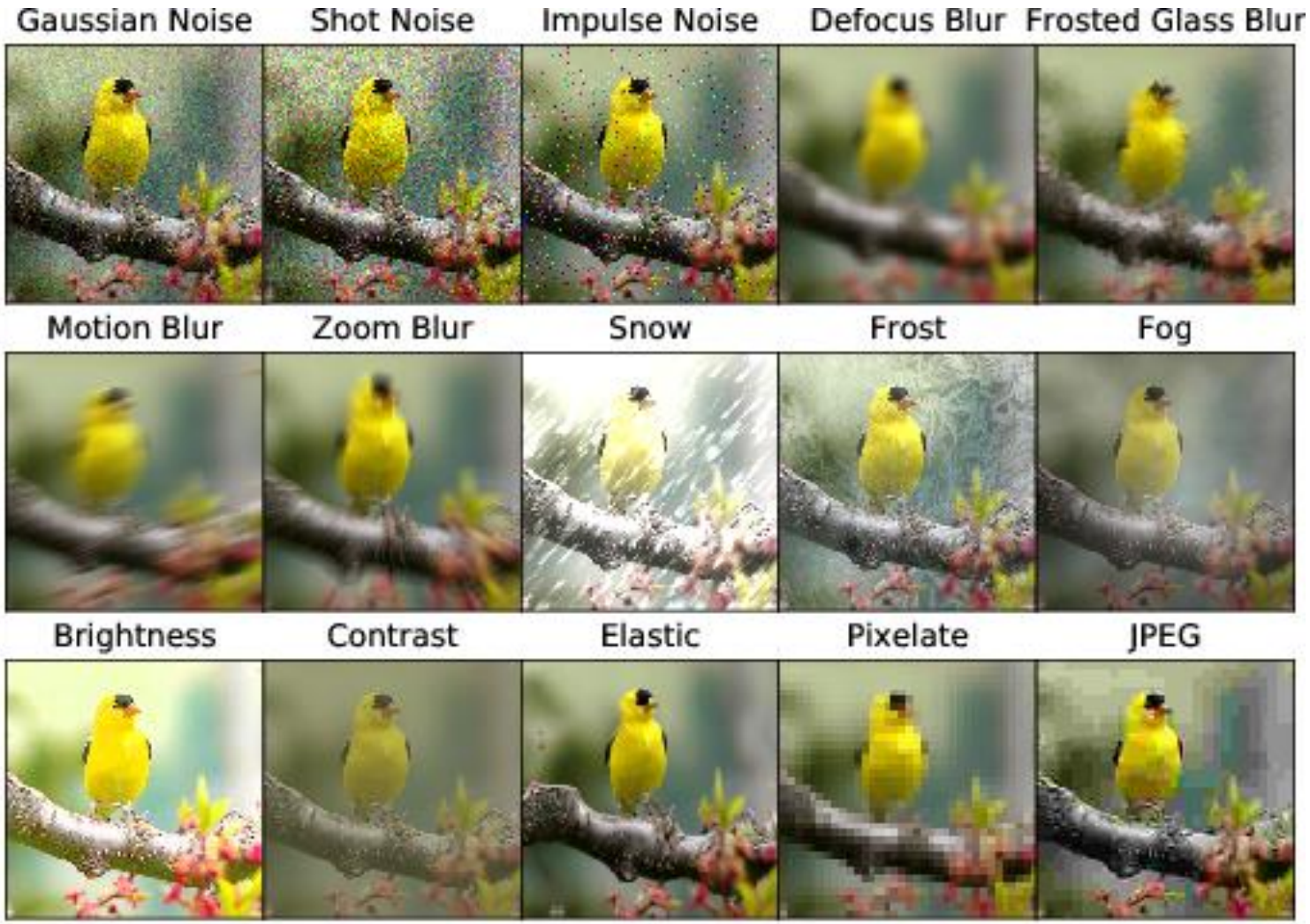


Guo et al. On calibration of modern neural networks. ICML 2017

In-distribution Calibration (cont.)

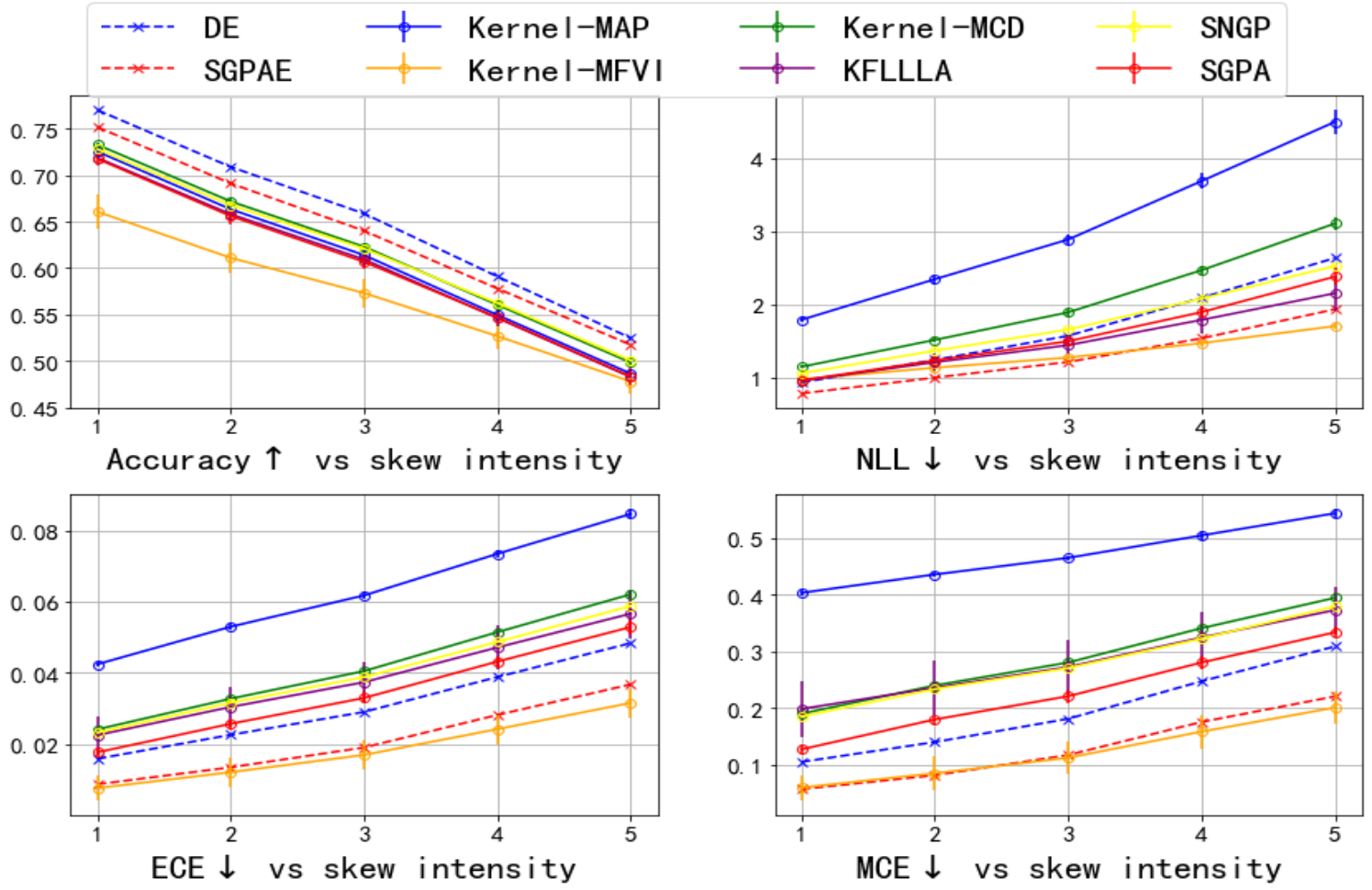


Out-of-distribution (OOD) Robustness



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

Out-of-distribution (OOD) Robustness (cont.)



Out-of-distribution detection

In-distribution data: CIFAR10 $Y = 0$

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

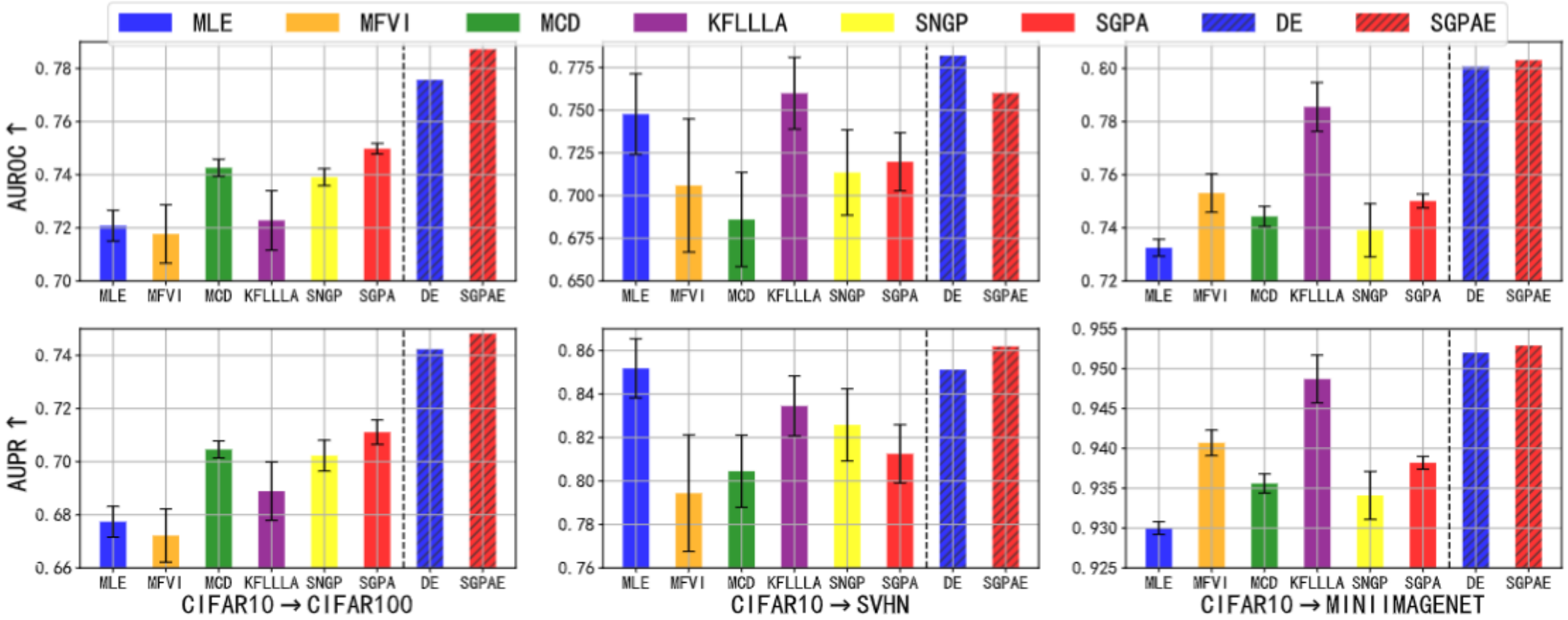
$$\hat{Y} = \mathbb{1}\{H[\hat{p}] > \tau\}$$

E.g. predictive entropy

Metrics (**prefer higher values**):

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

Out-of-distribution detection (cont.)

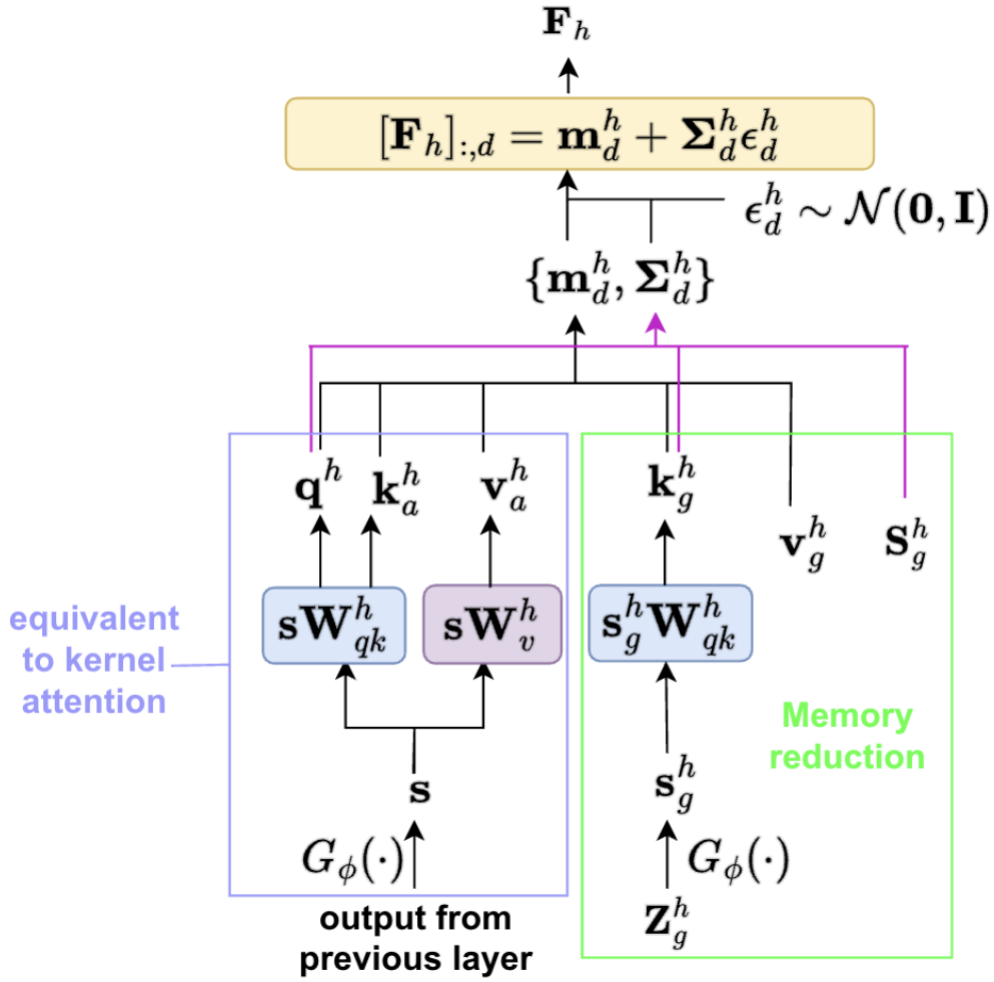


Summary

- **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**

On-Going Work 1: Keep Scaling Up

- $O(T^2)$ complexity as a major issue even for vanilla Transformers
  - Inherited by our solution in mean part
  - Decoupled approximation allows further improvements here
- Deep Learning practitioners don't like matrix inversions
  - Our solution needs matrix inversion for every head in every attention layer
  - Can we develop a matrix-inversion-free version?



On-Going 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: SGPA in this scenario
  - Think about auto-regressive token generation as (approximate) sequential Bayesian inference

Interested in some tutorials in Bayesian ML? (Materials also available at yingzhenli.net)

Approximate Inference
Tutorial @ NeurIPS 2020

Bayesian Neural Network
Tutorial @ ProbAI 2022

Thank You!

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