Self-Attention through Kernel-Eigen Pair Sparse Variational Gaussian Processes

Yingyi Chen [*1] Qinghua Tao [*1] Francesco Tonin [2] Johan A.K. Suykens [1]

Abstract

While the great capability of Transformers significantly boosts prediction accuracy, it could also yield overconfident predictions and require calibrated uncertainty estimation, which can be commonly tackled by Gaussian processes (GPs). Existing works apply GPs with symmetric kernels under variational inference to the attention kernel; however, omitting the fact that attention kernels are in essence asymmetric. Moreover, the complexity of deriving the GP posteriors remains high for large-scale data. In this work, we propose Kernel-Eigen Pair Sparse Variational Gaussian Processes (KEP-SVGP) for building uncertainty-aware self-attention where the asymmetry of attention kernels is tackled by Kernel SVD (KSVD) and a reduced complexity is acquired. Through KEP-SVGP, *i)* the SVGP pair induced by the two sets of singular vectors from KSVD w.r.t. the attention kernel fully characterizes the asymmetry; *ii)* using only a small set of adjoint eigenfunctions from KSVD, the derivation of SVGP posteriors can be based on the inversion of a diagonal matrix containing singular values, contributing to a reduction in time complexity; *iii)* an evidence lower bound is derived so that variational parameters and network weights can be optimized with it. Experiments verify our excellent performances and efficiency on in-distribution, distribution-shift and out-of-distribution benchmarks.

1. Introduction

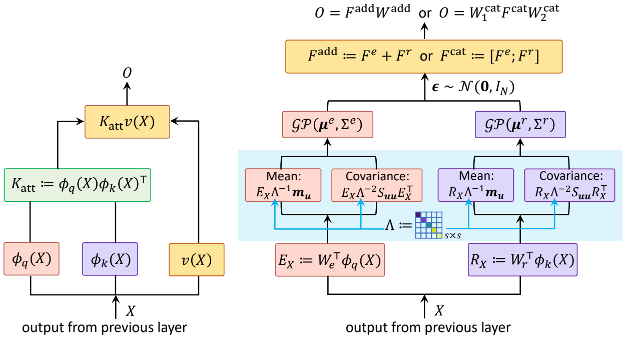
In recent years, Transformers (Vaswani et al., 2017) stand out among deep learning models, achieving state-of-the-art performances and excelling in feature learning in diverse applications (Brown et al., 2020; Dosovitskiy et al., 2021; Touvron et al., 2021; Wu et al., 2022). However, the large architecture capacities of Transformers could also lead to overconfident predictions (Guo et al., 2017; Mukhoti et al., 2020) with risks of robustness-related issues in safety-critical applications (Moon et al., 2020; Zhu et al., 2023) where reliable uncertainty quantification can help. Bayesian approaches allowing rich probabilistic interpretations of model predictions have been well studied on modern neural networks (Blundell et al., 2015; Gal & Ghahramani, 2016; Kendall & Gal, 2017; Salimbeni & Deisenroth, 2017; Geifman et al., 2019; Zhang et al., 2020), where posterior inferences are often conducted in weight spaces (Foong et al., 2020; Ritter et al., 2021; Coker et al., 2022). In Transformers, uncertainty estimation with Variational Inference (VI) (Graves, 2011) is relatively less studied. Existing works include the studies using VI on layer weights (Tran et al., 2019; Xue et al., 2021), attention matrix (Fan et al., 2020; Cinquin et al., 2022) and attention outputs (Liu et al., 2020; Chen & Li, 2023), vital in providing reliable predictions.

Gaussian processes (GPs) (Rasmussen & Williams, 2006) serve as principal tools for uncertainty estimation within Bayesian inference. Though GPs provide posterior distributions in closed forms, they are intractable for large datasets, e.g., long-sequence data for Transformers, as time complexity to the posterior GPs scale as $\mathcal{O}(N^3)$ where N is the number of training samples. Sparse Variational Gaussian Process (SVGP) (Titsias, 2009) deploying VI is proposed as an efficient alternative to classical GP. It conducts posterior approximation based on a small set of s "inducing points (variables)" yielding a reduction of time complexity from $\mathcal{O}(N^3)$ to $\mathcal{O}(Ns^2)$. Recently, SVGPs are utilized (Chen & Li, 2023) on attention outputs for uncertainty estimation. However, we underscore that the self-attention kernel is in essence asymmetric (Tsai et al., 2019; Wright & Gonzalez, 2021; Chen et al., 2023), whereas SVGPs can only be characterized with symmetric kernels, resulting in a nontrivial gap in capturing the intrinsic rationale.

Chen et al. (2023) casts the asymmetric self-attention kernel in the framework of Kernel Singular Value Decomposition (KSVD) (Suykens, 2016; Tao et al., 2023), which fully characterizes the asymmetry of the attention kernel through two sets of projection outputs w.r.t. both right and left singular vectors and can be efficiently optimized through an auxiliary loss. In this paper, we propose Kernel-Eigen Pair Sparse

[*] Equal contribution [1] ESAT-STADIUS, KU Leuven, Belgium [2] LIONS, EPFL, Switzerland (most of the work was done at ESAT-STADIUS, KU Leuven). Correspondence to: Yingyi Chen <yingyi.chen@esat.kuleuven.be, chenyingyi076@gmail.com>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).


O
$K_{\text{att}}v(X)$
$K_{\text{att}} \coloneqq \phi_q(X)\phi_k(X)^\top$
$\phi_q(X)$
$\phi_k(X)$
$v(X)$
X
output from previous layer
$O = F^{\text{add}}W^{\text{add}}$ or $O = W_1^{\text{cat}}F^{\text{cat}}W_2^{\text{cat}}$
$F^{\text{add}} \coloneqq F^e + F^r$ or $F^{\text{cat}} \coloneqq [F^e; F^r]$
$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, I_N)$
$\mathcal{GP}(\boldsymbol{\mu}^e, \Sigma^e)$
$\mathcal{GP}(\boldsymbol{\mu}^r, \Sigma^r)$
Mean: $E_X\Lambda^{-1}\boldsymbol{m_u}$
Covariance: $E_X\Lambda^{-2}S_{\boldsymbol{uu}}E_X^\top$
Mean: $R_X\Lambda^{-1}\boldsymbol{m_u}$
Covariance: $R_X\Lambda^{-2}S_{\boldsymbol{uu}}R_X^\top$
$\Lambda \coloneqq$
$s \times s$
$E_X \coloneqq W_e^\top\phi_q(X)$
$R_X \coloneqq W_r^\top\phi_k(X)$
X
output from previous layer

(a) Canonical Transformer

(b) KEP-SVGP

Figure 1. Illustration of canonical self-attention and our KEP-SVGP in one layer. (a) The attention kernel K_{att} in canonical self-attention is induced by two different feature maps ϕ_q, ϕ_k related to queries and keys; hence K_{att} is in essence asymmetric. (b) KEP-SVGP consists of one SVGP pair induced by the two sets of projection outputs based on ϕ_q, ϕ_k from KSVD w.r.t. K_{att}, which fully characterizes the asymmetry of self-attention in the posterior. The posteriors are now approximated based on the inversion of a diagonal matrix Λ containing top s singular values, thereby of time complexity $\mathcal{O}(s)$.

Variational Gaussian Processes (KEP-SVGP) for building uncertainty-aware self-attention where the asymmetry of attention kernel is tackled by KSVD and a reduced time complexity in computing the SVGPs posterior is also acquired. Specifically, through KEP-SVGP:

- *Our SVGP pair induced by the left and right singular vectors of KSVD w.r.t. the attention kernel matrix fully characterizes the asymmetry in the posterior.* This SVGP pair is obtained by setting the pair of adjoint eigenfunctions (Schmidt, 1907; Stewart, 1993) w.r.t. the asymmetric attention kernel to formulate "inducing variables", which is a technique named kernel-eigen features for SVGPs (Lázaro-Gredilla & Figueiras-Vidal, 2009; Leibfried et al., 2020). Note that this technique has not yet been explored for large-architecture models, such as Transformers.

- *We reduce the time complexity of the matrix inversion in SVGPs posterior approximation from $\mathcal{O}(s^3)$ to $\mathcal{O}(s)$.* By using the singular vectors-induced SVGP pair, the posterior is now approximated based on the inversion of a truncated singular value matrix, e.g., corresponding to top-s singular values, which is a diagonal matrix.

- *An evidence lower bound (ELBO) tailored for KEP-SVGP is derived,* so that variational parameters and the network weights can be jointly optimized with ELBO.

- *KEP-SVGP's efficacy and efficiency are experimentally verified* on in-distribution, distribution-shift and out-of-distribution benchmarks without sacrificing accuracy.[1]

2. Related Work

Uncertainty Estimation Uncertainty estimation aims at providing confidence scores to the predictions, hence helping with both interpretability and trustworthiness of the models, and benefits downstream tasks, including failure prediction (Hendrycks & Gimpel, 2017; Li et al., 2024), model calibration (Guo et al., 2017), OOD detection (Zhu et al., 2023), etc. MSP (Hendrycks & Gimpel, 2017) takes the maximum softmax probabilities from the softmax label distribution in deep neural network (DNN) classifiers as confidence scores. Temperature Scaling (Guo et al., 2017) is a post-hoc model calibration method which optimizes a single temperature parameter within the softmax in the DNN classifier on a validation set to calibrate predictions. Monte-Carlo Dropout (MC Dropout) (Gal & Ghahramani, 2016) casts dropout training in DNNs as approximate Bayesian inference in deep Gaussian processes, hence modelling uncertainty with dropout NNs. Kronecker-factored last layer Laplace approximation (KFLLLA) (Kristiadi et al., 2020) calibrates uncertainty on a ReLU network with asymptotic confidence of a last-layer Gaussian-approximated binary ReLU classifier with Laplace approximation. Deep Ensembles (Lakshminarayanan et al., 2017) ensembles DNNs for well-calibrated uncertainty estimates. Note that all the above mentioned methods can be easily implemented in different network architectures, such as Transformers.

Deep Gaussian Processes Damianou & Lawrence (2013) introduces a deep hierarchy of GPs, where each layer is a GP or GP latent variable model. Rather than combining with DNNs, Damianou & Lawrence (2013) utilizes deep GP hierarchies and are limited to small datasets. Aitchison et al. (2021) also focuses on the deep hierarchy of GPs by encompassing Damianou & Lawrence (2013) into deep kernel processes, where each GP layer with isotropic covariance kernel is connected to Wishart distribution and thus is fully characterized by the Gram matrix. Milsom et al. (2024) proposes convolutional deep kernel machine (C-DKM). Despite the naming, it belongs to deep GPs operating on Gram matrices (Aitchison et al., 2021) but focuses on convolutional kernels. Another category is deep kernel learning (DKL). Wilson et al. (2016) proposes stochastic variational deep kernel learning (SV-DKL), which introduces an extra GP layer to DNN backbones, where the DKL is in the sense of applying kernel machines on top of a DNN, which is agnostic in architectures. SV-DKL requires two-step training, i.e., the backbone pretraining and the finetuning with the extra GP layer. Ober et al. (2021) further empirically investigates Wilson et al. (2016). Discussions on KEP-SVGP's connections to deep GPs are provided in Section 4.1.

Bayesian Transformers Tran et al. (2019) and Xue et al. (2021) perform variational inference (VI) on Transformers'

[1] Code is at https://github.com/yingyichen-cyy/KEP-SVGP.

layer weights, which can underfit the data, verified both empirically (Chen & Li, 2023) and theoretically (Foong et al., 2020; Coker et al., 2022). Fan et al. (2020) and Cinquin et al. (2022) perform VI on attention matrix where their experimental settings can be too restrictive for complex problems (Chen & Li, 2023). VI can be also performed on attention outputs (Liu et al., 2020; Chen & Li, 2023). Liu et al. (2020) fits an extra GP layer over the last layer output, which can be treated as a deep kernel learning in Transformer. Chen & Li (2023) proposes Sparse Gaussian Process Attention (SGPA) by formulating the self-attention with one SVGP. Our KEP-SVGP is different from SGPA in *i)* we use two SVGPs with two kernel-eigen features, hence taking the asymmetry in self-attention into consideration, while SGPA considers one vanilla SVGP; *ii)* we manage to reduce time complexity for self-attention to linear with the number of input data N, while SGPA remains N^2. More discussions on the comparisons of time complexity between KEP-SVGP and SGPA are provided in Section 4.1.

3. Background

3.1. Sparse Variational Gaussian Processes

Gaussian Processes A GP (Rasmussen & Williams, 2006) represents a distribution, denoted by $\mathcal{GP}$, over real-valued functions $f(\cdot) : \mathcal{X} \to \mathbb{R}$ defined on an input domain $\mathcal{X} \subset \mathbb{R}^d$. A GP prior is characterized through two real-valued functions: a mean function $\mu(\cdot) : \mathcal{X} \to \mathbb{R}$ which is often set to zero without loss of generality, and a symmetric positive-definite covariance function parameterized by a kernel function $\kappa(\cdot,\cdot) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. When evaluating a GP at any finite number of inputs $X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]^\top$, $\boldsymbol{x}_i \in \mathcal{X}$, we obtain a Gaussian marginal distribution of function values $\boldsymbol{f} := [f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_N)]^\top \in \mathbb{R}^N$, that is,

$$\text{Prior: } f(\cdot) \sim \mathcal{GP}(0, \kappa(\cdot,\cdot)) \quad \Rightarrow \quad \boldsymbol{f} \sim \mathcal{N}(\mathbf{0}, K_{XX}), \tag{1}$$

with $K_{XX} := [\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)] \in \mathbb{R}^{N \times N}$. The training data is $(X, \boldsymbol{y}) := \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ with $\boldsymbol{y} = [y_1, \ldots, y_N]^\top$, $y_i \in \mathbb{R}$ being the given outputs to the inputs X. With the likelihood $\boldsymbol{y}|\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{f}, \sigma^2 I_N)$ being Gaussian, the posterior is also a GP. Given the test inputs X^*, the posterior predictive distribution of $\boldsymbol{f}^*$ is

$$\begin{aligned}\mathrm{p}(\boldsymbol{f}^*|X^*, X, \boldsymbol{y}) = \mathcal{N}\big(&K_{X^*X}(K_{XX} + \sigma^2 I_N)^{-1}\boldsymbol{y}\,, \\ &K_{X^*X^*} - K_{X^*X}(K_{XX} + \sigma^2 I_N)^{-1}K_{XX^*}\big)\,.\end{aligned} \tag{2}$$

However, (2) is intractable for large-scale data as the inversion of an $N \times N$ matrix is of time complexity $\mathcal{O}(N^3)$.

Sparse Variational Gaussian Processes SVGPs (Titsias, 2009) variationally approximate GP posteriors with a small set of s supports, i.e., $(Z, \boldsymbol{u}) := \{(\boldsymbol{z}_m, u_m)\}_{m=1}^s$, $\boldsymbol{z}_m \in \mathcal{X}$, $u_m = f(\boldsymbol{z}_m) \in \mathbb{R}$ where the "inducing variables" $\boldsymbol{u}$ are evaluated at the "inducing points" Z. In SVGPs, the mean $\boldsymbol{\mu}_{\boldsymbol{u}}$ is set to zero without loss of generality and the covariance matrix is $K_{ZZ} := [\kappa(\boldsymbol{z}_i, \boldsymbol{z}_j)] \in \mathbb{R}^{s \times s}$. Other than considering the marginal distribution $\mathrm{p}(\boldsymbol{u}) = \mathcal{N}(\mathbf{0}, K_{ZZ})$, SVGPs give a variational distribution $\mathrm{q}(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{m}_{\boldsymbol{u}}, S_{\boldsymbol{uu}})$, $\boldsymbol{m}_{\boldsymbol{u}} \in \mathbb{R}^s$, $S_{\boldsymbol{uu}} \in \mathbb{R}^{s \times s}$ (Leibfried et al., 2020). Thus, a marginal distribution over $\boldsymbol{f}$ can be obtained by $\mathrm{q}(\boldsymbol{f}) = \int \mathrm{p}(\boldsymbol{f}|\boldsymbol{u})\mathrm{q}(\boldsymbol{u})\,\mathrm{d}\boldsymbol{u}$, which corresponds to the posterior whose distribution is also Gaussian:

$$\begin{aligned}\mathrm{q}(\boldsymbol{f}) = \mathcal{N}\big(&K_{XZ}K_{ZZ}^{-1}\boldsymbol{m}_{\boldsymbol{u}}, \\ &K_{XX} - K_{XZ}K_{ZZ}^{-1}(K_{ZZ} - S_{\boldsymbol{uu}})K_{ZZ}^{-1}K_{ZX}\big)\,,\end{aligned} \tag{3}$$

where the kernel function values are $K_{XZ} := [\kappa(\boldsymbol{x}_i, \boldsymbol{z}_j)] \in \mathbb{R}^{N \times s}$, $K_{ZX} := K_{XZ}^\top$. In inference, the approximate posterior distribution evaluated at test inputs can then be obtained with (3). The optimization of SVGPs proceeds to maximize the evidence lower bound (ELBO) $\mathbb{E}_{\mathrm{q}(\boldsymbol{f})}\left[\log \mathrm{p}\left(\boldsymbol{y}|\boldsymbol{f}\right)\right] - \mathrm{KL}\left(\mathrm{q}(\boldsymbol{u})\|\mathrm{p}(\boldsymbol{u})\right)$ for the variational parameters $\boldsymbol{m}_{\boldsymbol{u}}$ and $S_{\boldsymbol{uu}}$ in the variational distribution $\mathrm{q}(\boldsymbol{u})$. Detailed derivations are given in Appendix A.1.

Kernel-Eigen Features for SVGPs Further in SVGPs, the "inducing variables" $\boldsymbol{u}$ can be alternatively chosen as a linear functional on $f(\cdot)$, which is called the "inter-domain GPs" (Lázaro-Gredilla & Figueiras-Vidal, 2009; Leibfried et al., 2020), such that

$$u_m = \int f(\boldsymbol{x})\phi_m(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}, \quad m = 1, \ldots, s, \tag{4}$$

where $\{\phi_m(\cdot)\}_{m=1}^s$ are the "inducing features" through the real-valued function $\phi_m(\cdot) : \mathcal{X} \to \mathbb{R}$. In particular, let $\phi_m(\cdot) := \nu_m(\cdot)$ be chosen as the m-th eigenfunction of the symmetric kernel $\kappa(\cdot,\cdot)$ with eigenvalue λ_m, that is,

$$\lambda_m \nu_m(\cdot) = \int \kappa(\cdot, \boldsymbol{x})\nu_m(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}, \quad m = 1, \ldots, s. \tag{5}$$

When evaluating the SVGP prior over a finite set $X \subset \mathcal{X}$ and its inducing points Z, the chosen eigenfunction $\nu_m(\cdot)$ for the "inducing features" $\phi_m(\cdot)$ in (4) leads to an eigenvalue problem, which corresponds to the finite case of the integral equations w.r.t. the symmetric kernel function $\kappa(\cdot,\cdot)$ in (5) (Williams & Seeger, 2000):

$$K_{XX}H = H\Lambda, \tag{6}$$

where $H := [\boldsymbol{\nu}_1, \ldots, \boldsymbol{\nu}_s] \in \mathbb{R}^{N \times s}$ contains the eigenvectors to the top-s nonzero eigenvalues of the kernel matrix K_{XX}, i.e., $\Lambda = \mathrm{diag}\{\lambda_1, \ldots, \lambda_s\}$. With $\mathrm{q}(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{m}_{\boldsymbol{u}}, S_{\boldsymbol{uu}})$, the posterior distribution (3) in SVGPs with kernel-eigen features is then yielded as:

$$\begin{aligned}&\text{Prior: } \begin{pmatrix}\boldsymbol{f}\\ \boldsymbol{u}\end{pmatrix} \sim \mathcal{GP}\left(\mathbf{0}, \begin{bmatrix}K_{XX} & H\Lambda \\ \Lambda H^\top & \Lambda\end{bmatrix}\right) \\ &\qquad\qquad\qquad \Downarrow \\ &\mathrm{q}(\boldsymbol{f}) = \mathcal{N}\big((H\Lambda)\Lambda^{-1}\boldsymbol{m}_{\boldsymbol{u}}, \\ &\quad K_{XX} - (H\Lambda)\Lambda^{-1}(\Lambda - S_{\boldsymbol{uu}})\Lambda^{-1}(\Lambda H^\top)\big)\,.\end{aligned} \tag{7}$$

Remark 3.1. By the compact SVD $K_{XX} = \sum_{i=1}^{R} \lambda_i \boldsymbol{\nu}_i \boldsymbol{\nu}_i^\top$ with R the rank of K_{XX}, the covariance of the posterior in (7) can be written as $K_{XX} - H\Lambda H^\top + HS_{\boldsymbol{uu}}H^\top = U\Lambda_U U^\top + HS_{\boldsymbol{uu}}H^\top$, where $\Lambda_U = \mathrm{diag}\{\lambda_{s+1}, \ldots, \lambda_R\}$ are the smallest non-zero $(R - s)$ eigenvalues, and columns of U are the corresponding eigenvectors. By the Eckart-Young theorem (Eckart & Young, 1936), $H\Lambda H^\top$ is the best rank-s approximation to K_{XX}. For low-rank matrices, such as the self-attention (Wang et al., 2020), $\|U\Lambda_U U^\top\|_F^2$ is small, motivating to a faster-to-compute approximate posterior by $\tilde{\mathrm{q}}(\boldsymbol{f}) \sim \mathcal{N}\left(H\boldsymbol{m}_{\boldsymbol{u}}, HS_{\boldsymbol{uu}}H^\top\right)$. Validity of this approximation is numerically verified in Appendix D.6.

Recall that regular SVGPs (3) give the posterior involving the inversion on $K_{ZZ} \in \mathbb{R}^{s\times s}$, and thereby have a time complexity of $\mathcal{O}(s^3)$. In contrast, with kernel-eigen features in SVGPs, the empirical covariance matrix w.r.t. $\boldsymbol{u}$ becomes diagonal, i.e., Λ, hence the time complexity of the matrix inversion is $\mathcal{O}(s)$, leading to a greater improvement in efficiency. Detailed derivations are in Appendix A.2.

3.2. Self-Attention as Asymmetric Kernel Machine

Self-Attention corresponds to Asymmetric Kernel Let the input data sequence be $\{\boldsymbol{x}_i\}_{i=1}^N$, $\boldsymbol{x}_i \in \mathcal{X}$, self-attention formulates the queries $q(\boldsymbol{x}_i) = W_q\boldsymbol{x}_i$, $W_q \in \mathbb{R}^{d_q\times d}$, keys $k(\boldsymbol{x}_i) = W_k\boldsymbol{x}_i$, $W_k \in \mathbb{R}^{d_k\times d}$, and values $v(\boldsymbol{x}_i) = W_v\boldsymbol{x}_i$, $W_v \in \mathbb{R}^{d_v\times d}$, commonly with $d_q = d_k$. As pointed out in Tsai et al. (2019), the attention matrix can be interpreted as a kernel matrix with entries depicting the asymmetric similarities between queries and keys:

$$\kappa_{\mathrm{att}}(\boldsymbol{x}_i, \boldsymbol{x}_j) := \mathrm{softmax}(\langle W_q\boldsymbol{x}_i, W_k\boldsymbol{x}_j\rangle / \sqrt{d_k}), \quad (8)$$

where $\kappa_{\mathrm{att}}(\cdot,\cdot) : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ is the kernel yielding attention matrix $K_{\mathrm{att}} := [\kappa_{\mathrm{att}}(\boldsymbol{x}_i, \boldsymbol{x}_j)] \in \mathbb{R}^{N\times N}$. As $W_q \neq W_k$ generally, we have $\langle W_q\boldsymbol{x}_i, W_k\boldsymbol{x}_j\rangle \neq \langle W_q\boldsymbol{x}_j, W_k\boldsymbol{x}_i\rangle$, so that the attention matrix is essentially asymmetric with $K_{ij} \neq K_{ji}$. The canonical self-attention output in each head is denoted as $O := [\boldsymbol{o}_1, \ldots, \boldsymbol{o}_N]^\top \in \mathbb{R}^{N\times d_v}$ with

$$\boldsymbol{o}_i = \sum\nolimits_{j=1}^{N} v(\boldsymbol{x}_j)\kappa_{\mathrm{att}}(\boldsymbol{x}_i, \boldsymbol{x}_j), \quad i = 1, \ldots, N. \quad (9)$$

Kernel-based approaches have become popular in studying the attention (Choromanski et al., 2021; Nguyen et al., 2022; Chi et al., 2022; Nguyen et al., 2023). However, they resort to the techniques with symmetric kernels where the inputs are $\{q(\boldsymbol{x}_i)\}_{i=1}^N$, $\{k(\boldsymbol{x}_j)\}_{i=1}^N$. Differently, the following Chen et al. (2023) works directly on the asymmetric kernel function and is grounded on the original input $\{\boldsymbol{x}_i\}_{i=1}^N$.

Self-Attention with Kernel SVD Chen et al. (2023) formulates the self-attention mechanism with KSVD (Suykens, 2016; Tao et al., 2023) which allows asymmetric kernels, and derives a primal-dual framework to represent the attention outputs and the optimization. The asymmetric kernel for self-attention is introduced as $\kappa_{\mathrm{att}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle\phi_q(\boldsymbol{x}_i), \phi_k(\boldsymbol{x}_j)\rangle$ with $\phi_q(\cdot) : \mathcal{X} \to \mathbb{R}^p$ and $\phi_k(\cdot) : \mathcal{X} \to \mathbb{R}^p$ related to queries and keys, respectively. The primal-dual representations of self-attention with KSVD give:

$$\begin{aligned} &\text{Primal:} \quad \begin{cases} e(\boldsymbol{x}) = W_e^\top \phi_q(\boldsymbol{x}), \\ r(\boldsymbol{x}) = W_r^\top \phi_k(\boldsymbol{x}), \end{cases} \\ &\text{Dual:} \quad \begin{cases} e(\boldsymbol{x}) = \sum_{j=1}^N \boldsymbol{h}_{r_j}\kappa_{\mathrm{att}}(\boldsymbol{x}, \boldsymbol{x}_j), \\ r(\boldsymbol{x}) = \sum_{i=1}^N \boldsymbol{h}_{e_i}\kappa_{\mathrm{att}}(\boldsymbol{x}_i, \boldsymbol{x}), \end{cases} \end{aligned} \quad (10)$$

where $e(\boldsymbol{x}), r(\boldsymbol{x}) \in \mathbb{R}^s$ are the projections related to queries and keys, whose variances are maximized under KSVD as shown in the objective (11). Primal variables $W_e, W_r \in \mathbb{R}^{p\times s}$ serve as the projection weights, and dual variables $H_e := [\boldsymbol{h}_{e_1}, \ldots, \boldsymbol{h}_{e_N}]^\top$, $H_r := [\boldsymbol{h}_{r_1}, \ldots, \boldsymbol{h}_{r_N}]^\top \in \mathbb{R}^{N\times s}$ are column-wisely the left and right singular vectors of the attention matrix K_{att}. Note that the canonical self-attention outputs in (9) corresponds to the dual representation of the projection score $e(\boldsymbol{x})$ in (10) once setting $\boldsymbol{h}_{r_j} := v(\boldsymbol{x}_j)$.

To fully exploit the asymmetry in self-attention kernel matrix, Chen et al. (2023) proposes Primal-Attention, which concatenates both projections $e(\boldsymbol{x})$, $r(\boldsymbol{x})$ w.r.t. right and left singular vectors, such that $F_i := [e(\boldsymbol{x}_i); r(\boldsymbol{x}_i)] = [W_e^\top\phi_q(\boldsymbol{x}_i); W_r^\top\phi_k(\boldsymbol{x}_i)]$. With the KKT conditions, the stationary solutions to KSVD yield a zero-value objective, as proved in Lemma 4.2 in Chen et al. (2023). Thus, the KSVD optimization in Primal-Attention can be flexibly implemented by minimizing an auxiliary regularization loss:

$$\begin{aligned} \min_{W_e, W_r, \Lambda} \mathcal{L}_{\mathrm{KSVD}} := \Big[&- \tfrac{1}{2}\textstyle\sum_{i=1}^N e(\boldsymbol{x}_i)^\top\Lambda^{-1}e(\boldsymbol{x}_i) \\ &- \tfrac{1}{2}\textstyle\sum_{j=1}^N r(\boldsymbol{x}_j)^\top\Lambda^{-1}r(\boldsymbol{x}_j) + \mathrm{Tr}(W_e^\top W_r)\Big]^2, \end{aligned} \quad (11)$$

seeking for the projections with maximal variances w.r.t. the two sets of singular vectors, where $\Lambda \in \mathbb{R}^{s\times s}$ is a positive diagonal matrix of the top-s singular values. Primal-Attention avoids computing the attention kernel matrix in the dual by deploying the primal representations with greater efficiency. Details of Chen et al. (2023) are in Appendix C.

4. KEP-SVGP for Self-Attention

In this section, the method of KEP-SVGP is illustrated in Section 4.1, where two branches of SVGPs with the adjoint kernel-eigen feature pair are considered based on KSVD to capture asymmetry in self-attention. Section 4.2 provides the optimization of KEP-SVGP.

4.1. Kernel-Eigen Pair SVGP

In (SV)GPs, the kernel function is required to be symmetric, whereas the attention in Transformers is in essence asymmetric. As shown in Chen et al. (2023), the asymmetric kernel in self-attention can be fully characterized by two sets of projections under the KSVD framework. To variationally model the outputs with the asymmetric attention kernel, we use the pair of adjoint eigenfunctions in the integral equations w.r.t. the attention kernel for the kernel-eigen features, so as to formulate the "inducing variables" in SVGPs.

Pair of Adjoint Eigenfunctions for Self-Attention With asymmetric $\kappa_{\text{att}}(\cdot,\cdot)$ (8), its adjoint eigenfunctions $\nu_{e_m}(\cdot)$, $\nu_{r_m}(\cdot)$ regarding the eigenvalue λ_m (Schmidt, 1907) satisfy

$$\begin{aligned}\lambda_m \nu_{e_m}(\cdot) &= \textstyle\int \kappa_{\text{att}}(\cdot,\boldsymbol{z})\nu_{r_m}(\boldsymbol{z})\,\mathrm{d}\boldsymbol{z},\\ \lambda_m \nu_{r_m}(\cdot) &= \textstyle\int \kappa_{\text{att}}(\boldsymbol{x},\cdot)\nu_{e_m}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}.\end{aligned} \tag{12}$$

The finite-sample cases to the integrals in (12) correspond to the compact SVD on an asymmetric attention kernel matrix $K_{\text{att}} \in \mathbb{R}^{N\times N}$, i.e., KSVD (Tao et al., 2023), where the approximation to $\nu_{e_m}(\cdot)$, $\nu_{r_m}(\cdot)$ leads to the left and right singular vectors. To differentiate from the symmetric cases, eigenvalue λ_m in (12) with asymmetric kernels are named as singular values (Stewart, 1993). In self-attention, this yields the shifted eigenvalue problem (Lanczos, 1958; Suykens, 2016) w.r.t. the attention matrix K_{att} (Chen et al., 2023):

$$K_{\text{att}} H_r = H_e \Lambda, \quad K_{\text{att}}^\top H_e = H_r \Lambda, \tag{13}$$

where $\Lambda = \operatorname{diag}\{\lambda_1,\ldots,\lambda_s\}$ contains the top-s nonzero singular values of K_{att}, $H_e, H_r \in \mathbb{R}^{N\times s}$ are kernel-eigen features defined in Section 3.2. However, the kernel matrix in (13) is asymmetric, hence inconsistent with the symmetry requirements of SVGPs in (6).

Two sets of integral equations w.r.t. a pair of symmetric kernels can be introduced to equivalently characterize the integral equations in (12) (Schmidt, 1907; Stewart, 1993):

$$\begin{aligned}\lambda_m^2 \nu_{e_m}(\cdot) &= \textstyle\int \kappa_e(\cdot,\boldsymbol{z})\nu_{e_m}(\boldsymbol{z})\,\mathrm{d}\boldsymbol{z},\\ \lambda_m^2 \nu_{r_m}(\cdot) &= \textstyle\int \kappa_r(\boldsymbol{x},\cdot)\nu_{r_m}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x},\end{aligned} \tag{14}$$

where $\kappa_e(\cdot,\boldsymbol{z}) := \int \kappa_{\text{att}}(\cdot,\boldsymbol{y})\kappa_{\text{att}}(\boldsymbol{z},\boldsymbol{y})\,\mathrm{d}\boldsymbol{y}$, $\kappa_r(\boldsymbol{x},\cdot) := \int \kappa_{\text{att}}(\boldsymbol{y},\boldsymbol{x})\kappa_{\text{att}}(\boldsymbol{y},\cdot)\,\mathrm{d}\boldsymbol{y}$ correspond to two symmetric and positive definite kernels and λ_m^2 is the eigenvalue in the induced two sets of integral equations. Thus, we can deploy a SVGP pair with kernel-eigen features (5) to variationally model the self-attention outputs.

Given the training data sequences in self-attention, the finite-sample cases to integrals (14) give two eigendecompositions w.r.t. symmetric kernels κ_e, κ_r:

$$(K_{\text{att}} K_{\text{att}}^\top) H_e = H_e \Lambda^2,\; (K_{\text{att}}^\top K_{\text{att}}) H_r = H_r \Lambda^2. \tag{15}$$

The asymmetric attention matrix K_{att} can be fully characterized by the symmetric $K_{\text{att}} K_{\text{att}}^\top$ and $K_{\text{att}}^\top K_{\text{att}}$ with kernel-eigen feature pair H_e, H_r serving as the "inducing features" in the resulting two SVGPs. Detailed derivations of (15) are given in Appendix B.1.

SVGPs with Kernel-Eigen Features Pair In basic setups of (SV)GPs, the processes are of single-output as in Section 3.1 with $f(\cdot): \mathcal{X} \to \mathbb{R}$. For the multi-dimensional attention outputs, we consider independent multi-output Gaussian processes (Leibfried et al., 2020) where a separate SVGP is specified for each output dimension (Salimbeni & Deisenroth, 2017; Chen & Li, 2023). Following the KSVD framework with Primal-Attention setups (Chen et al., 2023), we model the s-dimensional attention outputs with the two sets of projections for capturing the asymmetry, denoted as $F^e_{[d]} := F^e[:,d]$, $F^r_{[d]} := F^r[:,d] \in \mathbb{R}^N$, $d = 1,\ldots,s$, w.r.t. $e(\boldsymbol{x})$, $r(\boldsymbol{x})$ in (10), respectively. Therefore, with (15), a SVGP pair w.r.t. the two symmetric kernels $K_{\text{att}} K_{\text{att}}^\top$ and $K_{\text{att}}^\top K_{\text{att}}$ induced by the asymmetric K_{att} is established. With the kernel-eigen feature pair H_e, H_r on $\boldsymbol{f}^e_{[d]}, \boldsymbol{f}^r_{[d]} \in \mathbb{R}^N$, similar as (7) with Remark 3.1 our SVGPs are attained as follows:

$$\begin{aligned}
&\text{Prior: } \begin{pmatrix}\boldsymbol{f}^e_{[d]}\\ \boldsymbol{u}^e_{[d]}\end{pmatrix} \sim \mathcal{GP}\left(\mathbf{0}, \begin{bmatrix} K_{\text{att}} K_{\text{att}}^\top & H_e \Lambda^2\\ \Lambda^2 H_e^\top & \Lambda^2\end{bmatrix}\right)\\
&\qquad\qquad\Downarrow\\
&\tilde{\mathrm{q}}(\boldsymbol{f}^e_{[d]}) = \mathcal{N}\Big(\underbrace{E_X \Lambda^{-1} \boldsymbol{m}_{\boldsymbol{u},[d]}}_{\boldsymbol{\mu}^e := \boldsymbol{m}^e_{[d]}}, \underbrace{E_X \Lambda^{-2} S_{\boldsymbol{uu},[d]} E_X^\top}_{\Sigma^e := L^e_{[d]} L^e_{[d]}{}^\top}\Big)\\
&\text{Prior: } \begin{pmatrix}\boldsymbol{f}^r_{[d]}\\ \boldsymbol{u}^r_{[d]}\end{pmatrix} \sim \mathcal{GP}\left(\mathbf{0}, \begin{bmatrix} K_{\text{att}}^\top K_{\text{att}} & H_r \Lambda^2\\ \Lambda^2 H_r^\top & \Lambda^2\end{bmatrix}\right)\\
&\qquad\qquad\Downarrow\\
&\tilde{\mathrm{q}}(\boldsymbol{f}^r_{[d]}) = \mathcal{N}\Big(\underbrace{R_X \Lambda^{-1} \boldsymbol{m}_{\boldsymbol{u},[d]}}_{\boldsymbol{\mu}^r := \boldsymbol{m}^r_{[d]}}, \underbrace{R_X \Lambda^{-2} S_{\boldsymbol{uu},[d]} R_X^\top}_{\Sigma^r := L^r_{[d]} L^r_{[d]}{}^\top}\Big)
\end{aligned} \tag{16}$$

with variational distributions on $\boldsymbol{u}^e_{[d]}, \boldsymbol{u}^r_{[d]} \in \mathbb{R}^s$:

$$\boldsymbol{u}^e_{[d]}, \boldsymbol{u}^r_{[d]} \sim \mathcal{N}(\boldsymbol{m}_{\boldsymbol{u},[d]}, S_{\boldsymbol{uu},[d]}), \tag{17}$$

where $E_X := [e(\boldsymbol{x}_i),\ldots,e(\boldsymbol{x}_N)]^\top \in \mathbb{R}^{N\times s}$ and $R_X := [r(\boldsymbol{x}_i),\ldots,r(\boldsymbol{x}_N)]^\top \in \mathbb{R}^{N\times s}$ are the projection matrices w.r.t. right and left singular vectors of KSVD in (10), and $\boldsymbol{m}_{\boldsymbol{u}} \in \mathbb{R}^{s\times s}$, $S_{\boldsymbol{uu}} \in \mathbb{R}^{s\times s\times s}$ are the variational parameters with $\boldsymbol{m}_{\boldsymbol{u},[d]} := \boldsymbol{m}_{\boldsymbol{u}}[:,d] \in \mathbb{R}^s$, $S_{\boldsymbol{uu},[d]} := S_{\boldsymbol{uu}}[:,:,d] \in \mathbb{R}^{s\times s}$ corresponding to the d-th output dimension. Note that we set $\boldsymbol{u}^e_{[d]}$,$\boldsymbol{u}^r_{[d]}$ from the same variational distribution since the priors in SVGP pair in (16) share the same marginal distribution of the "inducing variables". Detailed derivations of our SVGP pair (16) are given in Appendix B.2.

Based on the approximate posteriors in (16), the outputs of the two SVGPs are obtained by the reparameterization trick (Salimbeni & Deisenroth, 2017):

$$F^e_{[d]} = \boldsymbol{m}^e_{[d]} + L^e_{[d]}\boldsymbol{\epsilon}, \quad F^r_{[d]} = \boldsymbol{m}^r_{[d]} + L^r_{[d]}\boldsymbol{\epsilon}, \tag{18}$$

with $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I_N)$, where $\boldsymbol{m}^e_{[d]}$, $\boldsymbol{m}^r_{[d]}$ are the means in (16) and $L^e_{[d]} := E_X \Lambda^{-1} L_{\boldsymbol{uu},[d]}$, $L^r_{[d]} := R_X \Lambda^{-1} L_{\boldsymbol{uu},[d]}$ are Cholesky factors of the approximate posterior covariances Σ^e, Σ^r in (16), and $L_{\boldsymbol{uu},[d]}$ is the Cholesky factor of $S_{\boldsymbol{uu},[d]}$.

Merging the SVGPs Utilizing the SVGP pair in (16) to preserve the asymmetric $K_{\rm att}$, we propose two schemes to merge two SVGPs outputs in (18):

$$\begin{aligned} \text{Addition:} \quad & F_{[d]} := F^e_{[d]} + F^r_{[d]} \in \mathbb{R}^N, \\ \text{Concatenation:} \quad & F_{[d]} := [F^e_{[d]}; F^r_{[d]}] \in \mathbb{R}^{2N}. \end{aligned} \tag{19}$$

To align with the d_v dimensions in standard Transformer architectures (9), our s-dimensional outputs are applied with linear projections, similar to Chen et al. (2023). Specifically, the final outputs $O \in \mathbb{R}^{N\times d_v}$ of the attention layer are: $O := F^{\rm add}W^{\rm add}$ for the addition, $O := W_1^{\rm cat}F^{\rm cat}W_2^{\rm cat}$ for the concatenation, where $F = [F_{[1]}, \ldots, F_{[s]}]$ is the output matrix of our merged SVGPs with projection matrices $W^{\rm add} \in \mathbb{R}^{s\times d_v}$, $W_1^{\rm cat} \in \mathbb{R}^{N\times 2N}$ and $W_2^{\rm cat} \in \mathbb{R}^{s\times d_v}$. Both schemes can be applied to data with fixed sequence lengths common in computer vision tasks, while for those with varying sequence lengths common in language modelling, we turn to the addition scheme. Note that for the concatenation scheme, we can replace $W_1^{\rm cat}$ with $AB^\top$, $A \in \mathbb{R}^{N\times s}$, $B \in \mathbb{R}^{2N\times s}$ to maintain the linear complexity with N. More details can be found in Appendix D.2.

Discussions on Time Efficiency The time complexity of MSP (Hendrycks & Gimpel, 2017), i.e., the canonical Transformer with softmax self-attention, is $\mathcal{O}(BN^2)$, where B is the batch size, N is the data sequence length. Recently, Chen & Li (2023) employs SVGPs for self-attention where Bayesian inference is performed in the attention output space to calibrate uncertainty. Chen & Li (2023) proposes: *i)* standard SGPA, which is based on regular SVGPs, with a time complexity of $\mathcal{O}(BN^3)$; *ii)* decoupled SGPA, which sets s global "inducing points", with time complexity $\mathcal{O}(BN^2s + s^3)$. However, decoupled SGPA still scales quadratically w.r.t. the sequence length and needs the matrix inversion with $\mathcal{O}(s^3)$. In contrast, our posterior distribution in (16) involves the matrix multiplication with $\mathcal{O}(BNs^2)$ and the inversion of the $s \times s$ diagonal matrix Λ with only $\mathcal{O}(s)$, leading to $\mathcal{O}(BNs^2 + Bs)$. To alignment with the hidden dimensions in standard Transformers, after (19), our concatenation merging scheme takes the matrix multiplication of $\mathcal{O}(N^2s)$. In practice, we commonly have $s < N$ with s being distinctively smaller than N, so that, omitting the effect of batch size, *the main time complexities in MSP, SGPA, and KEP-SVGP scale as $\mathcal{O}(N^2)$, $\mathcal{O}(N^2s)$, and $\mathcal{O}(Ns^2)$ for addition scheme, $\mathcal{O}(N^2s)$ for concatenation scheme, respectively.* In practice, we pertain considerable efficiency advantages for both our merging schemes, as we by default apply KEP-SVGP to the last layer, yielding better performances, instead of all layers as SGPA. Experiments on training time efficiency are in Table 4 and Appendix D.4.

Discussions on Connections to Deep GPs KEP-SVGP proposes a variational self-attention mechanism performing Bayesian inference in the outputs of multi-head attention blocks. **Relevance:** When multiple self-attention layers in a Transformer are replaced by KEP-SVGP, the model can be categorized as a specialized deep GPs; when only the last self-attention layer is replaced by KEP-SVGP, the model can be categorized as a stochastic variational model upon deep kernel learning with Transformer backbones. **Differences:** *i)* As attention matrix is asymmetric and cannot be simply treated as a GP covariance matrix, we propose two GP branches to capture the asymmetry. Current literature in deep GPs has not considered this asymmetric case observed in transformers. *ii)* Rather than adding an extra GP layer on top of a DNN (Wilson et al., 2016; Ober et al., 2021), KEP-SVGP replaces the last self-attention layer without imposing extra layers for variational modelling. *iii)* Compared to the two-step training in SV-DKL (Wilson et al., 2016), KEP-SVGP trains from scratch. *iv)* C-DKM (Milsom et al., 2024) is computationally demanding due to the large Gram matrices in each layer, while KEP-SVGP is efficient. Hence, some deep GPs methods are not comparable to KEP-SVGP in transformers due to different setups. Experiments on SV-DKL are in Section 5 with transformer for fair comparisons.

4.2. Optimization of KEP-SVGP

In optimization, we derive the ELBO objective for training the variational parameters involved in our SVGP pair. For Transformers with $N_{\rm h}$ heads in L attention layers applied with KEP-SVGP, we denote $\{F^l \in \mathbb{R}^{N\times(N_{\rm h}d_v)}\}_{l=1}^L$ as the output of the l-th KEP-SVGP layer following the convention of linearly concatenating the heads. Since single-output SVGPs (Leibfried et al., 2020) are employed as explained in (18), we can perform the variational inference on each output dimension of the attention heads before concatenating them, rather than the inference directly on the multi-head attention. With $\{\boldsymbol{u}^{l,n_{\rm h}}\}_{l=1,n_{\rm h}=1}^{L,N_{\rm h}}$ for the SVGPs in our KEP-SVGP, the ELBO is formulated as:

$$\begin{aligned} \max_{\Theta,\{\boldsymbol{m}_{\boldsymbol{u}},S_{\boldsymbol{uu}}\}} & \mathcal{L}_{\rm ELBO} := \mathbb{E}_{\mathrm{q}(F^L|F^0)}\left[\log \mathrm{p}(Y|F^L)\right] \\ & - \sum_{l=1}^{L}\sum_{n_{\rm h}=1}^{N_{\rm h}} \mathbb{E}_{\mathrm{q}(F^{l-1})}\left[\mathrm{KL}\big(\mathrm{q}(\boldsymbol{u}^{l,n_{\rm h}}|F^{l-1})\|\mathrm{p}(\boldsymbol{u}^{l,n_{\rm h}}|F^{l-1})\big)\right] \end{aligned} \tag{20}$$

where Θ denotes all network weights including W_e, W_r and Λ, Y contains the labels of the input data $F^0 := X_{\rm in}$. In (20), the first item in $\mathcal{L}_{\rm ELBO}$ corresponds to the objective of the learning task, such as the cross-entropy loss, while the second term of the Kullback–Leibler divergence balances the distance between the prior and variational distribution of the inducing variables $\boldsymbol{u}^{l,n_{\rm h}}$. Since $\boldsymbol{u}^e$, $\boldsymbol{u}^r$ share the same marginal prior and variational distributions conditioned on each F^{l-1}, we can consider one $\boldsymbol{u}$ for the KL divergence term. The KL divergence involved in the ELBO of our KEP-SVGP is depicted in Proposition 4.1, with detailed derivations in Appendix B.3.

Table 1. Mean and standard deviations on CIFAR-10, CIFAR-100, IMDB, CoLA benchmarks. Experimental results are reported over five trials, with the best mean results shown in bold. ACC, AUROC, FPR95, ECE and Brier are percentages, AURC is $\times 10^3$, NLL is $\times 10$.

Dataset	Method	ACC/MCC ↑	AURC ↓	AUROC ↑	FPR95 ↓	ECE ↓	NLL ↓	Brier ↓
CIFAR-10 (Krizhevsky et al., 2009)	MSP (Hendrycks & Gimpel, 2017)	83.50±0.43	42.60±1.84	86.15±0.35	66.51±2.19	12.87±0.29	11.13±0.40	28.62±0.74
	Temperature Scaling (Guo et al., 2017)	83.50±0.43	40.47±1.63	86.55±0.36	65.10±2.23	9.50±0.25	6.70±0.20	26.05±0.68
	MC Dropout (Gal & Ghahramani, 2016)	83.69±0.51	41.36±1.45	86.18±0.28	66.49±1.96	12.48±0.43	10.35±0.41	28.09±0.73
	KFLLLA (Kristiadi et al., 2020)	83.54±0.45	40.12±1.65	86.70±0.50	63.13±1.75	**1.51**±0.18	**5.08**±0.10	23.75 0.57
	SV-DKL (Wilson et al., 2016)	83.82±0.58	39.78±1.91	86.57±0.38	65.02±1.33	11.32±0.55	7.88±0.57	27.03±0.96
	KEP-SVGP (ours)	84.70±0.61	35.15±2.65	87.20±0.65	64.93±1.41	10.60±0.45	8.00±0.56	25.45±1.05
	Deep Ensembles (Lakshminarayanan et al., 2017)	86.43	27.76	88.64	60.72	9.98	7.40	22.89
	KEP-SVGP Ensembles (ours)	**87.62**	**22.56**	**89.64**	**56.70**	8.19	5.61	**20.08**
CIFAR-100 (Krizhevsky et al., 2009)	MSP (Hendrycks & Gimpel, 2017)	52.82±0.53	229.25±4.41	82.01±1.93	75.45±0.83	30.97±0.61	33.21±1.13	74.89±1.03
	Temperature Scaling (Guo et al., 2017)	52.82±0.53	223.83±4.18	82.47±0.34	71.79±0.97	17.36±0.68	21.82±0.54	64.92±0.84
	MC Dropout (Gal & Ghahramani, 2016)	53.37±0.62	224.01±4.82	81.44±0.17	75.12±1.12	30.09±0.80	31.78±1.22	73.51±1.24
	KFLLLA (Kristiadi et al., 2020)	51.35±0.64	263.59±4.65	79.62±0.19	71.86±1.30	37.98±4.61	27.21±1.66	82.84±3.82
	SV-DKL (Wilson et al., 2016)	53.00±0.71	228.10±8.35	81.41±0.63	72.22±1.63	**1.59**±0.22	**17.28**±0.31	60.07±0.87
	KEP-SVGP (ours)	55.02±0.83	209.75±6.20	81.71±0.30	74.03±0.90	27.80±0.57	28.38±0.45	69.91±1.12
	Deep Ensembles (Lakshminarayanan et al., 2017)	59.65	165.27	83.84	71.40	24.41	22.93	62.01
	KEP-SVGP Ensembles (ours)	**62.45**	**144.63**	**84.56**	**70.68**	21.01	19.64	**56.63**
IMDB (Maas et al., 2011)	MSP (Hendrycks & Gimpel, 2017)	88.17±0.52	35.27±3.04	82.29±0.87	71.41±1.57	4.01±1.36	3.10±0.26	17.88±0.95
	Temperature Scaling (Guo et al., 2017)	88.17±0.52	35.27±3.04	82.29±0.87	71.08±1.55	**1.05**±0.70	2.89±0.12	17.40±0.80
	MC Dropout (Gal & Ghahramani, 2016)	88.34±0.65	34.62±3.17	82.24±0.83	71.65±2.03	2.66±1.84	2.97±0.27	17.47±1.19
	KFLLLA (Kristiadi et al., 2020)	88.17±0.52	35.20±3.01	82.31±0.86	71.07±1.51	19.13±0.73	4.38±0.07	25.88±0.63
	SV-DKL (Wilson et al., 2016)	88.86±1.04	59.84±18.90	73.20±5.56	69.91±3.68	7.31±8.27	3.38±0.82	19.62±5.18
	SGPA (Chen & Li, 2023)	88.36±0.75	33.14±3.46	82.78±0.44	70.85±2.46	5.52±0.46	3.40±0.10	18.05±0.81
	KEP-SVGP (ours)	89.01±0.14	30.69±0.69	83.22±0.31	68.15±0.95	3.72±0.81	3.00±0.13	16.56±0.25
	Deep Ensembles (Lakshminarayanan et al., 2017)	89.57	28.69	83.45	67.69	2.42	**2.68**	**15.60**
	KEP-SVGP Ensembles (ours)	**89.68**	**27.79**	**83.56**	**67.54**	3.43	2.84	15.68
CoLA (Warstadt et al., 2019)	MSP (Hendrycks & Gimpel, 2017)	26.93±1.38	205.47±7.62	64.55±0.86	89.86±1.29	23.84±2.23	14.45±2.83	52.15±2.43
	Temperature Scaling (Guo et al., 2017)	26.93±1.38	205.46±7.61	64.55±0.91	90.09±0.77	18.98±3.33	8.72±1.18	47.59±2.85
	MC Dropout (Gal & Ghahramani, 2016)	26.41±1.87	203.93±8.34	65.15±0.76	88.58±0.53	23.33±2.16	13.74±2.64	51.35±2.43
	KFLLLA (Kristiadi et al., 2020)	26.90±1.31	204.31±8.57	64.60±0.96	90.06±0.74	**2.51**±1.09	**5.94**±0.04	**40.52**±0.38
	SV-DKL (Wilson et al., 2016)	26.65±1.38	235.76±10.03	62.14±1.36	89.94±1.79	18.13±8.92	8.07±1.86	48.45±6.41
	SGPA (Chen & Li, 2023)	26.15±1.12	210.03±6.30	64.18±0.68	90.35±1.47	16.48±0.79	8.76±0.34	45.77±0.54
	KEP-SVGP (ours)	30.54±1.61	186.66±8.50	65.16±0.86	88.39±0.83	15.89±3.48	8.54±1.66	43.55±2.99
	Deep Ensembles (Lakshminarayanan et al., 2017)	27.35	184.96	67.02	87.93	22.82	12.45	49.45
	KEP-SVGP Ensembles (ours)	**31.02**	**164.06**	**67.88**	**85.18**	14.96	7.40	40.68

Proposition 4.1. *The Kullback–Leibler divergence in the ELBO objective* (20) *is equal to*

$$\begin{aligned}&\tfrac{1}{2}\sum_{d=1}^{s}\Big[\mathrm{Tr}(\Lambda^{-2}S_{\boldsymbol{uu},[d]})+\boldsymbol{m}_{\boldsymbol{u},[d]}^{\top}\Lambda^{-2}\boldsymbol{m}_{\boldsymbol{u},[d]}\\&+\log\tfrac{|\Lambda^2|}{|S_{\boldsymbol{uu},[d]}|}-s\Big]\end{aligned} \tag{21}$$

where $\Lambda \in \mathbb{R}^{s\times s}$ *is diagonal whose inversion is of* $\mathcal{O}(s)$.

The training objective of KEP-SVGP is $\min -\mathcal{L}_{\text{ELBO}} + \eta\mathcal{L}_{\text{KSVD}}$, where $\eta > 0$ is the regularization constant. In our objective, we also incorporate loss $\mathcal{L}_{\text{KSVD}}$ in KSVD given in (11), ensuring that H_e, H_r in (15) are kernel-eigen features defined in Section 3.2 for SVGPs. Monte-Carlo sampling is used to compute $\mathcal{L}_{\text{ELBO}}$, where function values are generated iteratively by passing through each layer with the reparameterization trick (Kingma & Welling, 2013).

5. Experiments

Datasets and Baselines We conduct empirical evaluations on benchmarks including *i)* computer vision: CIFAR-10, CIFAR-100 (Krizhevsky et al., 2009); *ii)* language modelling: IMDB sentiment analysis (Maas et al., 2011), CoLA linguistic acceptability prediction (Warstadt et al., 2019). We compare our KEP-SVGP with *i)* single-model methods: maximum softmax probability score (MSP) (Hendrycks & Gimpel, 2017), Temperature Scaling (Guo et al., 2017), Monte-Carlo Dropout (MC Dropout) (Gal & Ghahramani, 2016), Kronecker-factored last layer Laplace approximation (KFLLLA) (Kristiadi et al., 2020), and SGPA (Chen & Li, 2023); *ii)* ensemble method: we compare our KEP-SVGP Ensembles with Deep Ensembles (Lakshminarayanan et al., 2017). In the experiments, we set the concatenation merging scheme for computer vision datasets, and the addition merging scheme for language modelling datasets. Unless specified, we replace the last-layer self-attention with KEP-SVGP, as this simple setup already achieves better performances with improved efficiency.

Training Configurations and Evaluation Metrics For both CIFAR-10, CIFAR-100, we train 7-layer Vision Transformer (ViT) (Dosovitskiy et al., 2021), optimized by Adam with batch size 128 and a cosine learning rate initialized with 10^{-3} for 300 epochs. Following Chen & Li (2023), for IMDB, we adopt one-layer Transformer, trained with batch size 32 with a cosine learning rate initialized with 10^{-3} for 20 epochs; for CoLA, we adopt a two-layer Transformer trained with batch size 32, an initial cosine learning rate of 5×10^{-4} for 50 epochs. For evaluations, in addition to test accuracy (ACC), we consider a variety of metrics

Table 2. Performance under distribution shift. The averaged results for 15 kinds of corruption under five different levels of perturbation severity are reported.

Method	ACC ↑	AURC ↓	AUROC ↑	FPR95 ↓	ECE ↓	NLL ↓	Brier ↓
			CIFAR-10-C				
MSP	69.17	151.07	78.71	77.98	24.75	24.27	53.99
MC Dropout	69.26	150.11	78.70	78.01	24.26	22.79	53.39
KFLLLA	69.17	146.55	79.59	75.11	**6.94**	**9.72**	**43.17**
SV-DKL	68.95	150.90	79.10	76.70	22.43	16.45	51.94
KEP-SVGP (ours)	69.71	144.90	79.40	77.13	22.03	18.30	50.70
Deep Ensembles	**73.93**	**114.14**	81.47	73.98	20.14	17.40	44.82
KEP-SVGP Ensembles (ours)	73.67	115.65	**81.54**	**73.85**	18.34	14.17	43.42
			CIFAR-100-C				
MSP	39.19	394.14	76.64	79.52	40.53	48.98	96.31
MC Dropout	39.62	389.91	76.67	79.27	39.73	47.29	95.12
KFLLLA	38.00	419.64	76.41	77.18	27.43	31.33	88.19
SV-DKL	40.17	379.79	77.84	76.40	**19.48**	**27.75**	**78.76**
KEP-SVGP (ours)	39.69	391.65	76.57	78.93	37.82	43.83	93.13
Deep Ensembles	**46.33**	**312.35**	**78.90**	76.73	33.40	35.91	82.98
KEP-SVGP Ensembles (ours)	46.31	315.69	78.76	**76.38**	31.21	33.03	80.93

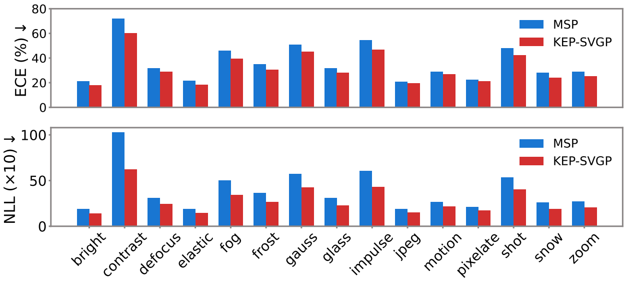


Figure 2. Comparisons of our KEP-SVGP with MSP under distribution shift. Performance on 15 types of corruption under the severity level of 5 is reported, where models are trained on CIFAR-10/ViT and tested on CIFAR-10-C.

widely used for failure prediction and uncertainty calibration, presenting more comprehensive analyses: *i)* failure prediction: the area under risk coverage curves (AURC) (Geifman & El-Yaniv, 2017), the area under the receiver operating characteristic curve (AUROC) (Davis & Goadrich, 2006), FPR95 that returns FPR at 95% TPR; *ii)* uncertainty calibration: expected calibration error (ECE) (Naeini et al., 2015), negative predictive log-likelihood (NLL), and Brier score (Brier, 1950). More implementation details including choices of s, η, kernel functions, Monte-Carlo sampling for ELBO during inference are given in Appendix D.1.

5.1. Comparison Results

Uncertainty Awareness on In-distribution Data In Table 1, we evaluate the in-distribution performances on four benchmarks where results on CoLA are measured with Matthew correlation coefficient (MCC) (Matthews, 1975; Warstadt et al., 2019). Among single-model methods, KFLLLA and Temperature Scaling are two post-hoc approaches designed for calibration and show good performances w.r.t. ECE, NLL and Brier. However, the performance of KFLLLA is not stable across tasks, e.g., it has the highest ECE on CIFAR-100, while not able to improve much regarding the failure prediction metrics. Our KEP-SVGP has consistently better performance across tasks. Compared with MSP, our method not only improves upon ACC/MCC, but significantly reduces the AURC, NLL, and other comparing metrics. Although SV-DKL using two-step training shows better performance in some metrics on CIFAR-100, KEP-SVGP has overall better performance regarding all benchmarks. Notably, compared to the latest method for uncertainty calibration on Transformers (SGPA), KEP-SVGP distinctively surpasses its performances w.r.t. both failure prediction and calibration metrics, together with significantly improved efficiency as later shown in Table 4. Deep Ensembles integrates five independently trained models and achieves more advantageous results than single-model methods. Compared to Deep Ensembles, KEP-SVGP Ensembles still outperforms it on all datasets and w.r.t. most metrics. Moreover, KEP-SVGP can serve as a good complement for further improvement in calibration, given in Appendix D.3.

Table 3. OOD detection performance with AUROC (%) and AUPR (%). The average results over five trials are reported.

ID	CIFAR-10			CIFAR-100		
OOD	SVHN	CIFAR-100	LSUN	SVHN	CIFAR-10	LSUN
		AUROC ↑				
MSP	86.56	81.50	87.48	75.83	67.14	74.97
MC Dropout	86.56	81.67	88.19	76.62	67.54	74.94
KFLLLA	75.95	75.67	80.00	72.81	65.37	71.25
SV-DKL	75.48	76.81	82.02	74.35	65.72	72.03
KEP-SVGP (ours)	84.75	82.32	91.50	79.98	67.51	78.22
Deep Ensembles	**90.74**	85.22	90.25	79.49	70.09	77.93
KEP-SVGP Ensembles (ours)	88.15	**85.36**	**93.24**	**84.16**	**70.44**	**81.28**
		AUPR ↑				
MSP	81.34	83.30	89.08	65.85	68.42	78.74
MC Dropout	81.89	83.50	89.69	67.03	68.93	78.81
KFLLLA	66.58	78.51	83.22	58.98	67.50	74.42
SV-DKL	64.68	78.79	84.71	59.63	68.83	74.91
KEP-SVGP (ours)	79.05	84.07	92.77	71.57	68.83	81.65
Deep Ensembles	**87.79**	86.75	91.62	71.38	71.30	82.05
KEP-SVGP Ensembles (ours)	84.35	**86.78**	**94.29**	**77.69**	**71.68**	**84.70**

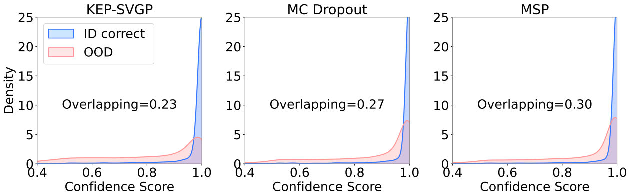


Figure 3. KEP-SVGP leads to better confidence separation between ID correct and OOD samples.

Robustness on Distribution-shift Data We consider CIFAR-10-C and CIFAR-100-C (Hendrycks & Dietterich, 2019) in Table 2, which are corrupted data of CIFAR-10/100 containing 15 types of input corruptions over 5 levels of corruption severity. Models are trained on clean CIFAR-10/100 and evaluated on corrupted CIFAR-10/100-C. As Temperature Scaling is designed for in-distribution uncertainty only, we omit its comparison herein. Among almost all single-model methods, KEP-SVGP is with best ACC. Although KFLLLA has good calibration metrics on both datasets, its ACC is equal to or lower than that of MSP, indicating that its good calibration performance comes at the cost of ACC.

Table 4. Comparisons of performance and efficiency on a single NVIDIA Tesla V100 SXM2 32 GB. Results are reported over five trials.

Method	Time Complexity	CIFAR-10			CIFAR-100			IMDB			CoLA		
		ACC ↑	NLL ↓	s/Epoch	ACC ↑	NLL ↓	s/Epoch	ACC ↑	NLL ↓	s/Epoch	MCC ↑	NLL ↓	s/Epoch
MSP	$\mathcal{O}(N^2)$	78.11±0.10	13.40±0.07	29.58	52.16±0.50	43.90±0.42	29.76	88.17±0.52	3.10±0.26	16.65	26.93±1.38	14.45±2.83	23.09
SGPA	$\mathcal{O}(N^2 s)$	77.87±0.12	6.97±0.02	137.28	53.02±0.36	25.64±0.41	288.04	88.36±0.75	3.40±0.10	1662.36	26.15±1.12	8.76±0.34	28.05
KEP-SVGP (ours)	$\mathcal{O}(Ns^2)$	**78.27**±0.30	**6.29**±0.06	30.97	**56.26**±0.70	**20.10**±1.10	32.21	**89.01**±0.14	**3.00**±0.13	32.30	**30.54**±1.61	**8.54**±1.66	23.95

Table 5. Ablation on the asymmetry of our SVGP pair on CoLA. Results are reported over five trials.

Pairing	MCC ↑	AURC ↓	AUROC ↑	FPR95 ↓	ECE ↓	NLL ↓	Brier ↓
MSP	26.93	205.47	64.55	89.86	23.84	14.45	52.15
$F^e_{[d]} + F^e_{[d]}$	28.71	199.48	64.05	89.36	21.85	13.15	49.99
$F^r_{[d]} + F^r_{[d]}$	28.10	196.65	64.66	90.01	18.99	10.76	47.68
$F^e_{[d]} + F^r_{[d]}$	**29.31**	**189.08**	**64.88**	**88.51**	**16.08**	**9.02**	**44.16**

SV-DKL is a strong method on CIFAR-100-C, however, its ACC on CIFAR-10-C is even lower than the MSP baseline, which is less desirable. KEP-SVGP has good ACC with more stable uncertainty metrics. Figure 2 shows the comparisons to MSP built upon canonical softmax-based Transformer where KEP-SVGP improves the model robustness under various corruptions. With ensembles, Deep Ensembles is a classical method admitted with strong robustness against distribution shift. Notably, KEP-SVGP Ensembles provides better results than Deep Ensembles on more metrics, especially on CIFAR-10-C. As we are not aiming at achieving SOTAs in all metrics/setups, current results can verify KEP-SVGP's robustness under distribution shift.

Out-of-distribution Detection An effective estimator with reliable confidence is expected to well separate out-of-distribution (OOD) data and misclassified in-distribution (ID) data from correct predictions. Hence, we consider OOD detection, where CIFAR-10/100 are taken as the ID data with evaluations on each other, SVHN (Netzer et al., 2011) and LSUN (Yu et al., 2015). In Table 3, KEP-SVGP can overall boost the AUROC and AUPR over baselines. Note that though KFLLLA has good performance on the tasks of in-distribution calibration and distribution-shift datasets, it cannot handle OOD detection well, while our method maintains its effectiveness. Moreover, our KEP-SVGP Ensembles achieves the best performance almost in all cases, e.g., the boost on CIFAR-100→SVHN is 4.67% in AUROC, 6.31% in AUPR over Deep Ensembles, which is quite substantial. Figure 3 shows that KEP-SVGP leads to less confidence overlap between OOD and correct ID compared with MSP, MC Dropout, which is desirable.

5.2. Time Complexity

Table 4 gives the training time efficiency discussed in Section 4.1. We adopt same architectures in Chen & Li (2023) on all datasets for fair comparisons with SGPA. Compared to the latest SVGPs-based counterpart SGPA, KEP-SVGP significantly reduces the computational time over all benchmarks, which is consistent with the analytical results on time complexity. Compared to MSP, it is reasonable that KEP-SVGP takes comparable or sometimes slightly longer training time as $\mathcal{O}(Ns^2)$ can be sometimes larger than $\mathcal{O}(N^2)$ with the chosen rank s. KEP-SVGP distinctively outperforms MSP on all datasets and metrics, only with a slightly extra training time. More results are in Appendix D.4.

5.3. Ablation Study

In Table 5, we investigate the effectiveness of leveraging our SVGPs for characterizing the asymmetric self-attention. We adopt a two-layer Transformer with all layers substituted by KEP-SVGP, which can also investigate our model applied to deep layers. We consider addition scheme for merging the outputs of SVGPs, i.e., $F^e_{[d]} + F^r_{[d]}$, and compare with the symmetric cases, which only include a single SVGP under the same architecture, i.e., either $F^e_{[d]} + F^e_{[d]}$ or $F^r_{[d]} + F^r_{[d]}$. Results show that we achieve performance gains by leveraging our SVGP pair conceived from the asymmetry on the self-attention kernel. Moreover, the 2-layer KEP-SVGP performs slightly inferior than transformers with only the last attention layer replaced by KEP-SVGP in Table 1. This can be due to the fact that shallow layer may not necessarily enjoy a low-rank property (Chen et al., 2023). Nevertheless, all variants of our 2-layer KEP-SVGP in Table 5 clearly outperform the MSP baseline, further verifying the effectiveness of our SVGPs, especially the asymmetry in our KEP-SVGP. More ablation results are provided in Appendix D.5.

6. Conclusion

In this work, we propose a novel variational modelling for realizing more reliable self-attention outputs through two branches of SVGPs, which leverages the pair of adjoint eigenfunctions w.r.t. the asymmetric attention kernel to formulate a pair of kernel-eigen features for "inducing features" in SVGPs. First, we fully characterize the intrinsic asymmetry of the attention kernel by utilizing the adjoint pair of SVGPs. Second, by deploying KSVD into SVGPs, we manage to reduce the time complexity of posterior processes approximation significantly. Third, we tailor the ELBO for optimizing the variational parameters in our model. Experiments verify our enhanced reliability and efficiency. To the best of our knowledge, this is the first variational inference modelling of Transformers with the asymmetry in attention kernel addressed.

Acknowledgements

This work is jointly supported by the European Research Council under the European Union's Horizon 2020 research and innovation program/ERC Advanced Grant E-DUALITY (787960), iBOF project Tensor Tools for Taming the Curse (3E221427), Research Council KU Leuven: Optimization framework for deep kernel machines C14/18/068, KU Leuven Grant CoE PFV/10/002, The Research Foundation–Flanders (FWO) projects: GOA4917N (Deep Restricted kernel Machines: Methods and Foundations), Ph.D./Postdoctoral grant, the Flemish Government (AI Research Program), EU H2020 ICT-48 Network TAILOR (Foundations of Trustworthy AI-Integrating Reasoning, Learning and Optimization), Leuven.AI Institute.

Impact Statement

In this work, we provide a new variational inference modelling for realizing uncertainty-aware self-attention modules in Transformers relying on Sparse Variational Gaussian Processes and Kernel SVD. Compared to methods in literature for self-attention within the Bayesian inference framework, our method has a low time complexity, hence faster computation processes. Therefore, we provide a more energy friendly method whose advantages lie in requiring less power consumption during model training. Moreover, as we propose a new self-attention mechanism, it can be adopted in many deep learning systems with Transformers as backbones. However, we should be aware that an undue trust in deep learning models when applying to real-life scenarios can lead to unexpected consequences. Considering that our uncertainty-aware attention maintains good robustness in failure prediction analysis, distribution-shift and out-of-distribution benchmarks, we still see opportunities in anticipating and mitigating the risks due to real-world uncertainty during applications in advance.

References

Aitchison, L., Yang, A., and Ober, S. W. Deep kernel processes. In *International Conference on Machine Learning*, pp. 130–140, 2021.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural network. In *International Conference on Machine Learning*, pp. 1613–1622, 2015.

Brier, G. W. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, 1950.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33: 1877–1901, 2020.

Chen, W. and Li, Y. Calibrating transformers via sparse Gaussian processes. In *International Conference on Learning Representations*, 2023.

Chen, Y., Tao, Q., Tonin, F., and Suykens, J. A. Primal-attention: Self-attention through asymmetric kernel SVD in primal representation. *Advances in Neural Information Processing Systems*, 2023.

Chi, T.-C., Fan, T.-H., Ramadge, P. J., and Rudnicky, A. Kerple: Kernelized relative positional embedding for length extrapolation. *Advances in Neural Information Processing Systems*, 35:8386–8399, 2022.

Choromanski, K. M., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., Hawkins, P., Davis, J. Q., Mohiuddin, A., Kaiser, L., Belanger, D. B., Colwell, L. J., and Weller, A. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021.

Cinquin, T., Immer, A., Horn, M., and Fortuin, V. Pathologies in priors and inference for Bayesian transformers. In *the Fourth Symposium on Advances in Approximate Bayesian Inference*, 2022.

Coker, B., Bruinsma, W. P., Burt, D. R., Pan, W., and Doshi-Velez, F. Wide mean-field Bayesian neural networks ignore the data. In *International Conference on Artificial Intelligence and Statistics*, 2022.

Damianou, A. and Lawrence, N. D. Deep Gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pp. 207–215, 2013.

Davis, J. and Goadrich, M. The relationship between precision-recall and ROC curves. In *International Conference on Machine Learning*, pp. 233–240, 2006.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Eckart, C. and Young, G. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

Fan, X., Zhang, S., Chen, B., and Zhou, M. Bayesian attention modules. *Advances in Neural Information Processing Systems*, 33:16362–16376, 2020.

Foong, A., Burt, D., Li, Y., and Turner, R. On the expressiveness of approximate inference in Bayesian neural networks. *Advances in Neural Information Processing Systems*, 33:15897–15908, 2020.

Gal, Y. and Ghahramani, Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, 2016.

Geifman, Y. and El-Yaniv, R. Selective classification for deep neural networks. *Advances in Neural Information Processing Systems*, 30, 2017.

Geifman, Y., Uziel, G., and El-Yaniv, R. Bias-reduced uncertainty estimation for deep neural classifiers. *International Conference on Learning Representations*, 2019.

Graves, A. Practical variational inference for neural networks. *Advances in Neural Information Processing Systems*, 24, 2011.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330. PMLR, 2017.

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. *International Conference on Learning Representations*, 2019.

Hendrycks, D. and Gimpel, K. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017.

Kendall, A. and Gal, Y. What uncertainties do we need in Bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems*, 30, 2017.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kristiadi, A., Hein, M., and Hennig, P. Being Bayesian, even just a bit, fixes overconfidence in ReLU networks. In *International Conference on Machine Learning*, pp. 5436–5446, 2020.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. *Technical Report*, 2009.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, 30, 2017.

Lanczos, C. Linear systems in self-adjoint form. *The American Mathematical Monthly*, 65(9):665–679, 1958.

Lázaro-Gredilla, M. and Figueiras-Vidal, A. Inter-domain Gaussian processes for sparse inference using inducing features. *Advances in Neural Information Processing Systems*, 22, 2009.

Leibfried, F., Dutordoir, V., John, S., and Durrande, N. A tutorial on sparse Gaussian processes and variational inference. *arXiv preprint arXiv:2012.13962*, 2020.

Li, Y., Chen, Y., Yu, X., Chen, D., and Shen, X. SURE: SUrvey REcipes for building reliable and robust deep networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Liu, J., Lin, Z., Padhy, S., Tran, D., Bedrax Weiss, T., and Lakshminarayanan, B. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in Neural Information Processing Systems*, 33:7498–7512, 2020.

Maas, A., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. Learning word vectors for sentiment analysis. In *Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, 2011.

Matthews, B. W. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451, 1975.

Milsom, E., Anson, B., and Aitchison, L. Convolutional deep kernel machines. *International Conference on Learning Representations*, 2024.

Moon, J., Kim, J., Shin, Y., and Hwang, S. Confidence-aware learning for deep neural networks. In *International Conference on Machine Learning*, pp. 7034–7044, 2020.

Mukhoti, J., Kulharia, V., Sanyal, A., Golodetz, S., Torr, P., and Dokania, P. Calibrating deep neural networks using focal loss. *Advances in Neural Information Processing Systems*, 33:15288–15299, 2020.

Naeini, M. P., Cooper, G., and Hauskrecht, M. Obtaining well calibrated probabilities using Bayesian binning. In *AAAI Conference on Artificial Intelligence*, volume 29, 2015.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. 2011.

Nguyen, T., Pham, M., Nguyen, T., Nguyen, K., Osher, S., and Ho, N. Fourierformer: Transformer meets generalized fourier integral theorem. *Advances in Neural Information Processing Systems*, 35:29319–29335, 2022.

Nguyen, T. M., Nguyen, T. M., Ho, N., Bertozzi, A. L., Baraniuk, R., and Osher, S. A primal-dual framework for transformers and neural networks. In *International Conference on Learning Representations*, 2023.

Ober, S. W., Rasmussen, C. E., and van der Wilk, M. The promises and pitfalls of deep kernel learning. In *Uncertainty in Artificial Intelligence*, pp. 1206–1216, 2021.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. Deep contextualized word representations. *Conference of the North American Chapter of the Association for Computational Linguistics*, 2018.

Rasmussen, C. E. and Williams, C. K. *Gaussian processes for machine learning*. Springer, 2006.

Ritter, H., Kukla, M., Zhang, C., and Li, Y. Sparse uncertainty representation in deep learning with inducing weights. *Advances in Neural Information Processing Systems*, 34:6515–6528, 2021.

Salimbeni, H. and Deisenroth, M. Doubly stochastic variational inference for deep Gaussian processes. *Advances in Neural Information Processing Systems*, 30, 2017.

Schmidt, E. Zur theorie der linearen und nichtlinearen integralgleichungen. *Mathematische Annalen*, 63(4):433–476, 1907.

Stewart, G. W. On the early history of the singular value decomposition. *SIAM Review*, 35(4):551–566, 1993.

Suykens, J. A. SVD revisited: A new variational principle, compatible feature maps and nonlinear extensions. *Applied and Computational Harmonic Analysis*, 40(3): 600–609, 2016.

Tao, Q., Tonin, F., Patrinos, P., and Suykens, J. A. Nonlinear SVD with asymmetric kernels: feature learning and asymmetric Nyström method. *arXiv preprint arXiv:2306.07040*, 2023.

Titsias, M. Variational learning of inducing variables in sparse Gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pp. 567–574, 2009.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pp. 10347–10357, 2021.

Tran, D., Dusenberry, M., Van Der Wilk, M., and Hafner, D. Bayesian layers: A module for neural network uncertainty. *Advances in Neural Information Processing Systems*, 32, 2019.

Tsai, Y.-H. H., Bai, S., Yamada, M., Morency, L.-P., and Salakhutdinov, R. Transformer dissection: An unified understanding for transformer's attention via the lens of kernel. In *Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4344–4353, 2019.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

Wang, S., Li, B. Z., Khabsa, M., Fang, H., and Ma, H. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

Warstadt, A., Singh, A., and Bowman, S. R. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019.

Williams, C. and Seeger, M. Using the Nyström method to speed up kernel machines. In Leen, T., Dietterich, T., and Tresp, V. (eds.), *Advances in Neural Information Processing Systems*, volume 13, 2000.

Wilson, A. G., Hu, Z., Salakhutdinov, R. R., and Xing, E. P. Stochastic variational deep kernel learning. *Advances in Neural Information Processing Systems*, 29, 2016.

Wright, M. A. and Gonzalez, J. E. Transformers are deep infinite-dimensional non-Mercer binary kernel machines. *arXiv preprint arXiv:2106.01506*, 2021.

Wu, H., Wu, J., Xu, J., Wang, J., and Long, M. Flowformer: Linearizing transformers with conservation flows. In *International Conference on Machine Learning*, pp. 24226–24242, 2022.

Xue, B., Yu, J., Xu, J., Liu, S., Hu, S., Ye, Z., Geng, M., Liu, X., and Meng, H. Bayesian transformer language models for speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 7378–7382. IEEE, 2021.

Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., and Xiao, J. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

Zhang, R., Li, C., Zhang, J., Chen, C., and Wilson, A. G. Cyclical stochastic gradient MCMC for Bayesian deep learning. In *International Conference on Learning Representations*, 2020.

Zhu, F., Cheng, Z., Zhang, X.-Y., and Liu, C.-L. Openmix: Exploring outlier samples for misclassification detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12074–12083, 2023.

A. Analytical Derivations of SVGPs

A.1. Derivations of (3): Variational Marginal Distribution in SVGPs

As an efficient alternative to the classical GPs (2), SVGPs (Titsias, 2009) variationally approximate the Gaussian posterior distribution with a small set of s supports, i.e., $(Z, \boldsymbol{u}) := \{(\boldsymbol{z}_m, u_m)\}_{m=1}^{s}$, $\boldsymbol{z}_m \in \mathcal{X}$, $u_m = f(\boldsymbol{z}_m) \in \mathbb{R}$, commonly with $s \ll N$, where Z are named "inducing points" and $\boldsymbol{u}$ are "inducing variables". In SVGPs, the mean $\boldsymbol{\mu}_{\boldsymbol{u}}$ is often set to zero and the covariance matrix is given by $K_{\boldsymbol{uu}} := [\kappa(\boldsymbol{z}_i, \boldsymbol{z}_j)] \in \mathbb{R}^{s\times s}$. The joint GP and the conditional GP of $f(\cdot)$ conditioned on $\boldsymbol{u}$ are as follows:

$$
\begin{gathered}
\text{Prior: } \begin{pmatrix} f(\cdot) \\ \boldsymbol{u} \end{pmatrix} \sim \mathcal{GP}\left(\boldsymbol{0}, \begin{bmatrix} \kappa(\cdot,\cdot) & \boldsymbol{\kappa}_{\cdot\boldsymbol{u}} \\ \boldsymbol{\kappa}_{\boldsymbol{u}\cdot} & K_{\boldsymbol{uu}} \end{bmatrix}\right) \\
\Downarrow \\
f(\cdot)|\boldsymbol{u} \sim \mathcal{GP}\left(\boldsymbol{\kappa}_{\cdot\boldsymbol{u}} K_{\boldsymbol{uu}}^{-1}\boldsymbol{u},\, \kappa(\cdot,\cdot) - \boldsymbol{\kappa}_{\cdot\boldsymbol{u}} K_{\boldsymbol{uu}}^{-1}\boldsymbol{\kappa}_{\boldsymbol{u}\cdot}\right)
\end{gathered}
\tag{22}
$$

with $\boldsymbol{\kappa}_{\cdot\boldsymbol{u}} := [\kappa(\cdot, \boldsymbol{z}_1), \ldots, \kappa(\cdot, \boldsymbol{z}_s)]$, $\boldsymbol{\kappa}_{\boldsymbol{u}\cdot} := [\kappa(\boldsymbol{z}_1, \cdot), \ldots, \kappa(\boldsymbol{z}_s, \cdot)]^\top$, and $\boldsymbol{\kappa}_{\boldsymbol{u}\cdot} = \boldsymbol{\kappa}_{\cdot\boldsymbol{u}}^\top$. Then, the corresponding conditional distribution of the function values $\boldsymbol{f}$ on $\boldsymbol{u}$ is as follows:

$$
\mathrm{p}(\boldsymbol{f}|\boldsymbol{u}) = \mathcal{N}(K_{XZ}K_{ZZ}^{-1}\boldsymbol{u},\, K_{XX} - K_{XZ}K_{ZZ}^{-1}K_{ZX}),
$$

where $K_{XZ} := [\kappa(\boldsymbol{x}_i, \boldsymbol{z}_j)] \in \mathbb{R}^{N\times s}$, $K_{ZX} := [\kappa(\boldsymbol{z}_i, \boldsymbol{x}_j)] \in \mathbb{R}^{s\times N}$ and $K_{ZZ} = K_{\boldsymbol{uu}} \in \mathbb{R}^{s\times s}$.

In addition to considering the marginal distribution $\mathrm{p}(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{0}, K_{ZZ})$, SVGPs provide a variational distribution $\mathrm{q}(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{m}_{\boldsymbol{u}}, S_{\boldsymbol{uu}})$ where $\boldsymbol{m}_{\boldsymbol{u}} \in \mathbb{R}^s$, $S_{\boldsymbol{uu}} \in \mathbb{R}^{s\times s}$ (Leibfried et al., 2020). The variational marginal distribution of $\boldsymbol{f}$ is given by $\mathrm{q}(\boldsymbol{f}) = \int \mathrm{p}(\boldsymbol{f}|\boldsymbol{u})\mathrm{q}(\boldsymbol{u})\,\mathrm{d}\boldsymbol{u}$, which is still Gaussian and corresponds to the approximate posterior (3) in Section 3.1:

$$
\mathrm{q}(\boldsymbol{f}) = \mathcal{N}\left(K_{XZ}K_{ZZ}^{-1}\boldsymbol{m}_{\boldsymbol{u}},\, K_{XX} - K_{XZ}K_{ZZ}^{-1}(K_{ZZ} - S_{\boldsymbol{uu}})K_{ZZ}^{-1}K_{ZX}\right). \tag{23}
$$

In inference, the posterior distribution of $\boldsymbol{f}^*$ evaluated at test inputs X^* is then given by

$$
\mathrm{q}(\boldsymbol{f}^*|X^*, Z) = \mathcal{N}\left(K_{X^*Z}K_{ZZ}^{-1}\boldsymbol{m}_{\boldsymbol{u}},\, K_{X^*X^*} - K_{X^*Z}K_{ZZ}^{-1}(K_{ZZ} - S_{\boldsymbol{uu}})K_{ZZ}^{-1}K_{ZX^*}\right).
$$

In SVGPs, the evidence lower-bound (ELBO) objective involves the variational parameters $\boldsymbol{m}_{\boldsymbol{u}}$, $S_{\boldsymbol{uu}}$ in the variational distribution $\mathrm{q}(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{m}_{\boldsymbol{u}}, S_{\boldsymbol{uu}})$ and is used for the training

$$
\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{\mathrm{q}(\boldsymbol{f})}\left[\log \mathrm{p}(\boldsymbol{y}|\boldsymbol{f})\right] - \mathrm{KL}\left(\mathrm{q}(\boldsymbol{u})||\mathrm{p}(\boldsymbol{u})\right).
$$

The derivation of this classical ELBO is provided in (30) in Appendix B.3. More details on SVGPs can refer to Titsias (2009); Leibfried et al. (2020).

A.2. Derivations of (7): SVGPs with Kernel-Eigen Features

With (4), let $\phi_m(\cdot) := \nu_m(\cdot)$ be chosen as the eigenfunction corresponding to the m-th largest eigenvalue λ_m w.r.t. the kernel function $\kappa(\cdot,\cdot)$, i.e., $\int \kappa(\cdot, \boldsymbol{x})\nu_m(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \lambda_m\nu_m(\cdot)$ in (5). In this setup, the $\boldsymbol{\kappa}_{\cdot\boldsymbol{u}}$, $\boldsymbol{\kappa}_{\boldsymbol{u}\cdot}$, $K_{\boldsymbol{uu}}$ of the prior GP in (22) are updated as follows (Leibfried et al., 2020):

$$
\begin{aligned}
\boldsymbol{\kappa}_{\cdot\boldsymbol{u}}[m] = \boldsymbol{\kappa}_{\boldsymbol{u}\cdot}[m] &= \lambda_m\nu_m(\cdot), \\
K_{\boldsymbol{uu}} &= \mathrm{diag}\{\lambda_1, \ldots, \lambda_s\},
\end{aligned}
\tag{24}
$$

where $\boldsymbol{\kappa}_{\cdot\boldsymbol{u}}[m]$, $\boldsymbol{\kappa}_{\boldsymbol{u}\cdot}[m]$ are the m-th entry of $\boldsymbol{\kappa}_{\cdot\boldsymbol{u}}$ and $\boldsymbol{\kappa}_{\boldsymbol{u}\cdot}$ respectively. The derivations of (24) are provided in the following.

The cross-covariance $\boldsymbol{\kappa}_{\cdot\boldsymbol{u}}$ is a vector-valued function with s outputs. The scalar-valued function $\boldsymbol{\kappa}_{\cdot\boldsymbol{u}}[m]$ corresponding to the output index m is computed as:

$$
\begin{aligned}
\boldsymbol{\kappa}_{\cdot\boldsymbol{u}}[m] &= \mathbb{E}\left[(f(\cdot) - 0)\,(u_m - 0)\right] = \mathbb{E}\left[f(\cdot)\left(\int f(\boldsymbol{x})\nu_m(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\right)\right] \\
&= \int \mathbb{E}\left[f(\cdot)f(\boldsymbol{x})\right]\nu_m(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \int \kappa(\cdot, \boldsymbol{x})\nu_m(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \lambda_m\nu_m(\cdot).
\end{aligned}
$$

Similarly, the m-th output of the cross-covariance $\boldsymbol{\kappa}_{\boldsymbol{u}\cdot}$ is the attained as:

$$\begin{aligned}\boldsymbol{\kappa}_{\boldsymbol{u}\cdot}[m] &= \mathbb{E}\left[(u_m - 0)\,(f(\cdot) - 0)\right] = \mathbb{E}\left[\left(\int f(\boldsymbol{x})\nu_m(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\right) f(\cdot)\right] \\ &= \int \mathbb{E}\left[f(\boldsymbol{x})f(\cdot)\right]\nu_m(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \int \kappa(\boldsymbol{x},\cdot)\nu_m(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \\ &= \int \kappa(\cdot,\boldsymbol{x})\nu_m(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \lambda_m\nu_m(\cdot),\end{aligned}$$

where the fifth equation holds with a symmetric $\kappa(\cdot,\cdot)$. The covariance $K_{\boldsymbol{uu}}$ is an $s \times s$ matrix with the $[i,j]$-th entry computed by

$$\begin{aligned}K_{\boldsymbol{uu}}[i,j] &= \mathbb{E}\left[(u_i - 0)\,(u_j - 0)\right] = \mathbb{E}\left[\left(\int f(\boldsymbol{x})\nu_i(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\right)\left(\int f(\boldsymbol{x}')\nu_j(\boldsymbol{x}')\,\mathrm{d}\boldsymbol{x}'\right)\right] \\ &= \int\int \mathbb{E}\left[f(\boldsymbol{x})f(\boldsymbol{x}')\right]\nu_i(\boldsymbol{x})\nu_j(\boldsymbol{x}')\,\mathrm{d}\boldsymbol{x}'\mathrm{d}\boldsymbol{x} = \int\int \kappa(\boldsymbol{x},\boldsymbol{x}')\nu_i(\boldsymbol{x})\nu_j(\boldsymbol{x}')\,\mathrm{d}\boldsymbol{x}'\mathrm{d}\boldsymbol{x} \\ &= \int \nu_i(\boldsymbol{x})\int \kappa(\boldsymbol{x},\boldsymbol{x}')\nu_j(\boldsymbol{x}')\,\mathrm{d}\boldsymbol{x}'\mathrm{d}\boldsymbol{x} = \int \nu_i(\boldsymbol{x})\lambda_j\nu_j(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = \lambda_j\int \nu_i(\boldsymbol{x})\nu_j(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \\ &= \lambda_i \text{ if } i = j \text{ else } 0.\end{aligned} \tag{25}$$

The last equation in (25) establishes since eigenfunctions are orthonormal systems: $\int \nu_i(\boldsymbol{x})\nu_j(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ equals one if i equals j, and is zero otherwise. When evaluating the SVGP prior over a finite set $X \subset \mathcal{X}$ and its inducing points Z, we have the finite case of the integral equations w.r.t. the symmetric kernel function $\kappa(\cdot,\cdot)$ in (5) (Williams & Seeger, 2000) as

$$K_{XX}H = H\Lambda,$$

where $H := [\boldsymbol{\nu}_1,\ldots,\boldsymbol{\nu}_s] \in \mathbb{R}^{N\times s}$ contains the eigenvectors to the top-s nonzero eigenvalues of the kernel matrix K_{XX}, i.e., $\Lambda = \text{diag}\{\lambda_1,\ldots,\lambda_s\}$. By substituting (24) together with its finite sample case (6) into the GP prior in (22), with $\text{q}(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{m}_{\boldsymbol{u}}, S_{\boldsymbol{uu}})$, the posterior distribution corresponding to (23) is then formulated as

$$\begin{gathered}\text{Prior: } \begin{pmatrix}\boldsymbol{f}\\ \boldsymbol{u}\end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix}K_{XX} & H\Lambda \\ \Lambda H^\top & \Lambda\end{bmatrix}\right) \\ \Downarrow \\ \text{q}(\boldsymbol{f}) = \mathcal{N}\left((H\Lambda)\Lambda^{-1}\boldsymbol{m}_{\boldsymbol{u}},\, K_{XX} - (H\Lambda)\Lambda^{-1}(\Lambda - S_{\boldsymbol{uu}})\Lambda^{-1}(\Lambda H^\top)\right).\end{gathered} \tag{26}$$

This completes the derivation of (7) in the paper.

B. Analytical Derivations of KEP-SVGP

B.1. Derivations of (15): Two Eigenvalue Problems Induced by KSVD on the Asymmetric Attention Kernel Matrix

From the KSVD on the asymmetric attention kernel matrix K_{att}, we have $K_{\text{att}} = H_e\Lambda H_r^\top$, where the left and right singular vectors suffice $H_e^\top H_e = I$, $H_r^\top H_r = I$ with $\boldsymbol{h}_{e,i} := H_e[:,i] = [\boldsymbol{h}_{e_1}[i],\ldots,\boldsymbol{h}_{e_N}[i]]^\top \in \mathbb{R}^N$ and $\boldsymbol{h}_{r,i} := H_r[:,i] = [\boldsymbol{h}_{r_1}[i],\ldots,\boldsymbol{h}_{r_N}[i]]^\top \in \mathbb{R}^N$, $i = 1,\ldots,N$. With KSVD, we have the shifted eigenvalue problem in (13), which gives

$$\begin{aligned}\left(K_{\text{att}}K_{\text{att}}^\top\right)H_e &= K_{\text{att}}\left(H_e\Lambda H_r^\top\right)^\top H_e = K_{\text{att}}H_r\Lambda H_e^\top H_e = K_{\text{att}}H_r\Lambda\left(H_e^\top H_e\right) = K_{\text{att}}H_r\Lambda = H_e\Lambda^2, \\ \left(K_{\text{att}}^\top K_{\text{att}}\right)H_r &= \left(H_e\Lambda H_r^\top\right)^\top K_{\text{att}}H_r = H_r\Lambda H_e^\top\left(K_{\text{att}}H_r\right) = H_r\Lambda H_e^\top\left(H_e\Lambda\right) = H_r\Lambda\left(H_e^\top H_e\right)\Lambda = H_r\Lambda^2.\end{aligned} \tag{27}$$

More explanations on the shifted eigenvalue problem from SVD can refer to the Lanczos decomposition in Lanczos (1958), Theorem 3.2 in Chen et al. (2023), Proposition 3.1 in Tao et al. (2023). This completes the derivation of (15) in the paper.

B.2. Derivations of (16): SVGP Pair on the Asymmetric Attention Kernel

Within the framework of KSVD (Suykens, 2016; Tao et al., 2023) w.r.t. the self-attention (Chen et al., 2023), we have the equivalence between the primal and dual model representation for the projection matrices w.r.t. right and left singular vectors of KSVD in (10), such that

$$E_X := W_e^\top\phi_q(X) \overset{(10)}{=} K_{\text{att}}H_r \overset{(13)}{=} H_e\Lambda, \quad R_X := W_r^\top\phi_k(X) \overset{(10)}{=} K^\top H_e \overset{(13)}{=} H_r\Lambda, \tag{28}$$

where $E_X := e(X) = [e(\boldsymbol{x}_i), \dots, e(\boldsymbol{x}_N)]^\top \in \mathbb{R}^{N\times s}$, $R_X := R_X = [r(\boldsymbol{x}_i), \dots, r(\boldsymbol{x}_N)]^\top \in \mathbb{R}^{N\times s}$ are the projection matrices w.r.t. right and left singular vectors of KSVD in (10). Here, we only consider the output of the d-th dimension. We consider the approximate posterior GP w.r.t. the symmetric kernel $K_{\text{att}}K_{\text{att}}^\top$ and $K_{\text{att}}^\top K_{\text{att}}$ given the distribution on the inducing points $\boldsymbol{u}^e_{[d]}, \boldsymbol{u}^r_{[d]} \sim \mathcal{N}(\boldsymbol{m}_{\boldsymbol{u},[d]}, S_{\boldsymbol{uu},[d]})$, where $\boldsymbol{m}_{\boldsymbol{u},[d]} := \boldsymbol{m}_{\boldsymbol{u}}[:, d] \in \mathbb{R}^s$ and $S_{\boldsymbol{uu},[d]} := S_{\boldsymbol{uu}}[:, :, d] \in \mathbb{R}^{s\times s}$ correspond to the d-th output dimension for the variational parameters $\boldsymbol{m}_{\boldsymbol{u}} \in \mathbb{R}^{s\times s}$, $S_{\boldsymbol{uu}} \in \mathbb{R}^{s\times s\times s}$.

According to the formulations of SVGPs in (7) and (26), the mean of the posterior process in the first set of SVGP in (16) is attained as

$$\boldsymbol{\mu}^e := (H_e\Lambda^2)\Lambda^{-2}\boldsymbol{m}_{\boldsymbol{u},[d]} = (H_e\Lambda)\Lambda^{-1}\boldsymbol{m}_{\boldsymbol{u},[d]} = E_X\Lambda^{-1}\boldsymbol{m}_{\boldsymbol{u},[d]}, \tag{29}$$

with the covariance matrix

$$\begin{aligned}\Sigma^e &:= K_{\text{att}}K_{\text{att}}^\top - (H_e\Lambda^2)\Lambda^{-2}(\Lambda^2 - S_{\boldsymbol{uu},[d]})\Lambda^{-2}(\Lambda^2 H_e^\top)\\ &= K_{\text{att}}K_{\text{att}}^\top - (H_e\Lambda^2)\Lambda^{-2}(\Lambda^2 H_e^\top) + (H_e\Lambda)\Lambda^{-1}S_{\boldsymbol{uu},[d]}\Lambda^{-1}(H_e\Lambda)^\top\\ &= \underbrace{K_{\text{att}}K_{\text{att}}^\top - H_e\Lambda^2 H_e^\top}_{\approx 0 \text{ (see Remark 3.1)}} + E_X\Lambda^{-1}S_{\boldsymbol{uu},[d]}\Lambda^{-1}E_X^\top\\ &\approx E_X\Lambda^{-1}S_{\boldsymbol{uu},[d]}\Lambda^{-1}E_X^\top\\ &= E_X\Lambda^{-2}S_{\boldsymbol{uu},[d]}E_X^\top.\end{aligned}$$

The attention matrix is commonly low-rank (Wang et al., 2020; Chen et al., 2023), so we motivate to utilize the fast-to-compute approximate posterior as given by Remark 3.1. Numerical evidence is also provided in Appendix D.6, verifying the validity of Remark 3.1 in our work. Therefore, we have the approximate distribution of the posterior process as

$$\tilde{\mathrm{q}}(\boldsymbol{f}^e_{[d]}) = \mathcal{N}\left(E_X\Lambda^{-1}\boldsymbol{m}_{\boldsymbol{u},[d]},\, E_X\Lambda^{-2}S_{\boldsymbol{uu},[d]}E_X^\top\right).$$

Similarly, based on (16) and (7), the mean of the posterior process w.r.t. the symmetric kernel $K_{\text{att}}^\top K_{\text{att}}$ is

$$\boldsymbol{\mu}^r := (H_r\Lambda^2)\Lambda^{-2}\boldsymbol{m}_{\boldsymbol{u},[d]} = (H_r\Lambda)\Lambda^{-1}\boldsymbol{m}_{\boldsymbol{u},[d]} = R_X\Lambda^{-1}\boldsymbol{m}_{\boldsymbol{u},[d]},$$

as given in (16), and the corresponding covariance is

$$\begin{aligned}\Sigma^r &:= K_{\text{att}}^\top K_{\text{att}} - (H_r\Lambda^2)\Lambda^{-2}(\Lambda^2 - S_{\boldsymbol{uu},[d]})\Lambda^{-2}(\Lambda^2 H_r^\top)\\ &= K_{\text{att}}^\top K_{\text{att}} - (H_r\Lambda^2)\Lambda^{-2}(\Lambda^2 H_r^\top) + (H_r\Lambda)\Lambda^{-1}S_{\boldsymbol{uu},[d]}\Lambda^{-1}(H_r\Lambda)^\top\\ &= \underbrace{K_{\text{att}}^\top K_{\text{att}} - H_r\Lambda^2 H_r^\top}_{\approx 0 \text{ (see Remark 3.1)}} + R_X\Lambda^{-1}S_{\boldsymbol{uu},[d]}\Lambda^{-1}R_X^\top\\ &\approx R_X\Lambda^{-1}S_{\boldsymbol{uu},[d]}\Lambda^{-1}R_X^\top\\ &= R_X\Lambda^{-2}S_{\boldsymbol{uu},[d]}R_X^\top,\end{aligned}$$

hence yielding the approximate distribution of the posterior process as

$$\tilde{\mathrm{q}}(\boldsymbol{f}^r_{[d]}) = \mathcal{N}\left(R_X\Lambda^{-1}\boldsymbol{m}_{\boldsymbol{u},[d]},\, R_X\Lambda^{-2}S_{\boldsymbol{uu},[d]}R_X^\top\right).$$

This completes the derivation of (16) in the paper.

B.3. Derivations of (20): The ELBO Objective of KEP-SVGP

For the optimization of our approximate posterior distribution, we follow the spirit of deep Gaussian Processes in Salimbeni & Deisenroth (2017); Damianou & Lawrence (2013). Hence the Transformers applied with KEP-SVGP can be viewed as a sparse approximation to a deep GPs with kernels in each layer. To proceed our derivations, we firstly recall the formulations of the ELBO involving $\boldsymbol{y}, \boldsymbol{f}, \boldsymbol{u}$:

$$\begin{aligned}
\log \mathrm{p}(\boldsymbol{y}, \boldsymbol{f}, \boldsymbol{u}) &= \log \int \underbrace{\mathrm{p}(\boldsymbol{y}|\boldsymbol{f}, \boldsymbol{u})}_{\text{likelihood}} \underbrace{\mathrm{p}(\boldsymbol{f}, \boldsymbol{u})}_{\text{GP Prior}} \, \mathrm{d}\boldsymbol{f}\mathrm{d}\boldsymbol{u} = \log \int \mathrm{q}(\boldsymbol{f}, \boldsymbol{u}) \frac{\mathrm{p}(\boldsymbol{y}|\boldsymbol{f}, \boldsymbol{u})\mathrm{p}(\boldsymbol{f}, \boldsymbol{u})}{\mathrm{q}(\boldsymbol{f}, \boldsymbol{u})} \, \mathrm{d}\boldsymbol{f}\mathrm{d}\boldsymbol{u} \\
&\overset{\text{Jensen's inequality}}{\geq} \int \mathrm{q}(\boldsymbol{f}, \boldsymbol{u}) \log \left(\frac{\mathrm{p}(\boldsymbol{y}|\boldsymbol{f}, \boldsymbol{u})\mathrm{p}(\boldsymbol{f}, \boldsymbol{u})}{\mathrm{q}(\boldsymbol{f}, \boldsymbol{u})} \right) \mathrm{d}\boldsymbol{f}\mathrm{d}\boldsymbol{u} \\
&= \int \mathrm{q}(\boldsymbol{f}, \boldsymbol{u}) \log \left(\frac{\mathrm{p}(\boldsymbol{y}, \boldsymbol{f}, \boldsymbol{u})}{\mathrm{q}(\boldsymbol{f}, \boldsymbol{u})} \right) \mathrm{d}\boldsymbol{f}\mathrm{d}\boldsymbol{u} = \mathbb{E}_{\mathrm{q}(\boldsymbol{f}, \boldsymbol{u})} \left[\log \frac{\mathrm{p}(\boldsymbol{y}, \boldsymbol{f}, \boldsymbol{u})}{\mathrm{q}(\boldsymbol{f}, \boldsymbol{u})} \right]. \qquad (30)
\end{aligned}$$

Similar to the doubly stochastic variational inference framework (Salimbeni & Deisenroth, 2017) for the ELBO derivation, let $\{F^l \in \mathbb{R}^{N \times (N_\text{h} d_v)}\}_{l=1}^L$ be the output of the l-th KEP-SVGP layer, where L is the number of layers and N_h is the number of heads. Our model follows the convention of concatenating these multiple heads in canonical self-attention. As is shown in (16) and (17), $\boldsymbol{u}^e$, $\boldsymbol{u}^r$ share the same marginal prior and variational distribution, hence we only need to optimize one set of variational parameters $\{\boldsymbol{m}_{\boldsymbol{u}}, S_{\boldsymbol{u}\boldsymbol{u}}\}$ for each KEP-SVGP layer. In this manner, we consider $\{\boldsymbol{u}^{l,n_\text{h}}\}_{l=1,n_\text{h}=1}^{L,N_\text{h}}$ for the L attention layers with N_h heads, and the resulting process can be characterized with the joint density:

$$\mathrm{p}\left(Y, \{F^l\}_{l=1}^L, \{\boldsymbol{u}^{l,n_\text{h}}\}_{l=1,n_\text{h}=1}^{L,N_\text{h}} | F^0\right) = \underbrace{\mathrm{p}\left(Y|F^L\right)}_{\text{likelihood}} \underbrace{\prod\nolimits_{l=1}^L \mathrm{p}\left(F^l | \{\boldsymbol{u}^{l,n_\text{h}}\}_{n_\text{h}=1}^{N_\text{h}}, F^{l-1}\right) \mathrm{p}\left(\{\boldsymbol{u}^{l,n_\text{h}}\}_{n_\text{h}=1}^{N_\text{h}})|F^{l-1}\right)}_{\text{GP Prior}}, \qquad (31)$$

where we define $F^0 := X_\text{in}$ as the inputs to the Transformer. The variational posterior of $(\{F^l\}_{l=1}^L, \{\boldsymbol{u}^{l,n_\text{h}}\}_{l=1,n_\text{h}=1}^{L,N_\text{h}})$ is then:

$$\mathrm{q}\left(\{F^l\}_{l=1}^L, \{\boldsymbol{u}^{l,n_\text{h}}\}_{l=1,n_\text{h}=1}^{L,N_\text{h}} | F^0\right) = \prod\nolimits_{l=1}^L \mathrm{p}(F^l | \{\boldsymbol{u}^{l,n_\text{h}}\}_{n_\text{h}=1}^{N_\text{h}}, F^{l-1}) \mathrm{q}\left(\{\boldsymbol{u}^{l,n_\text{h}}\}_{n_\text{h}=1}^{N_\text{h}})|F^{l-1}\right), \qquad (32)$$

where $\mathrm{q}(\{\boldsymbol{u}^{l,n_\text{h}}\}_{n_\text{h}=1}^{N_\text{h}})|F^{l-1})$ is the variational distribution, and $\mathrm{q}(F^l|\{\boldsymbol{u}^{l,n_\text{h}}\}_{n_\text{h}=1}^{N_\text{h}}, F^{l-1}) = \mathrm{p}(F^l|\{\boldsymbol{u}^{l,n_\text{h}}\}_{n_\text{h}=1}^{N_\text{h}}, F^{l-1})$ is also assumed as in Chen & Li (2023). Moreover, we also follow the assumption in Chen & Li (2023) on the conditional independency for each head across layers. When considering $\{\boldsymbol{u}^{l,n_\text{h}}\}_{n_\text{h}=1}^{N_\text{h}}$ from each head, we then have the factorizations:

$$\mathrm{p}\left(\{\boldsymbol{u}^{l,n_\text{h}}\}_{n_\text{h}=1}^{N_\text{h}})|F^{l-1}\right) = \prod\nolimits_{n_\text{h}=1}^{N_\text{h}} \mathrm{p}\left(\boldsymbol{u}^{l,n_\text{h}}|F^{l-1}\right), \quad \mathrm{q}\left(\{\boldsymbol{u}^{l,n_\text{h}}\}_{n_\text{h}=1}^{N_\text{h}})|F^{l-1}\right) = \prod\nolimits_{n_\text{h}=1}^{N_\text{h}} \mathrm{q}\left(\boldsymbol{u}^{l,n_\text{h}}|F^{l-1}\right). \qquad (33)$$

With the prerequisites derived above, we now proceed to formulate the ELBO in our KEP-SVGP:

$$\begin{aligned}
&\mathcal{L}_\text{ELBO} \overset{(30)}{=} \mathbb{E}_{\mathrm{q}\left(\{F^l\}_{l=1}^L, \{\boldsymbol{u}^{l,n_\text{h}}\}_{l=1,n_\text{h}=1}^{L,N_\text{h}}|F^0\right)} \left[\log \frac{\mathrm{p}\left(Y, \{F^l\}_{l=1}^L, \{\boldsymbol{u}^{l,n_\text{h}}\}_{l=1,n_\text{h}=1}^{L,N_\text{h}}|F^0\right)}{\mathrm{q}\left(\{F^l\}_{l=1}^L, \{\boldsymbol{u}^{l,n_\text{h}}\}_{l=1,n_\text{h}=1}^{L,N_\text{h}}|F^0\right)} \right] \\
&\overset{(31),(32)}{=} \mathbb{E}_{\mathrm{q}\left(\{F^l\}_{l=1}^L, \{\boldsymbol{u}^{l,n_\text{h}}\}_{l=1,n_\text{h}=1}^{L,N_\text{h}}|F^0\right)} \left[\log \frac{\mathrm{p}\left(Y|F^L\right) \prod_{l=1}^L \mathrm{p}\left(F^l|\{\boldsymbol{u}^{l,n_\text{h}}\}_{n_\text{h}=1}^{N_\text{h}}, F^{l-1}\right) \mathrm{p}\left(\{\boldsymbol{u}^{l,n_\text{h}}\}_{n_\text{h}=1}^{N_\text{h}})|F^{l-1}\right)}{\prod_{l=1}^L \mathrm{p}(F^l|\{\boldsymbol{u}^{l,n_\text{h}}\}_{n_\text{h}=1}^{N_\text{h}}, F^{l-1}) \mathrm{q}\left(\{\boldsymbol{u}^{l,n_\text{h}}\}_{n_\text{h}=1}^{N_\text{h}})|F^{l-1}\right)} \right] \\
&\overset{(33)}{=} \mathbb{E}_{\mathrm{q}\left(\{F^l\}_{l=1}^L, \{\boldsymbol{u}^{l,n_\text{h}}\}_{l=1,n_\text{h}=1}^{L,N_\text{h}}|F^0\right)} \left[\log \mathrm{p}\left(Y|F^L\right)\right] + \mathbb{E}_{\mathrm{q}\left(\{F^l\}_{l=1}^L, \{\boldsymbol{u}^{l,n_\text{h}}\}_{l=1,n_\text{h}=1}^{L,N_\text{h}}|F^0\right)} \left[\log \frac{\prod_{l=1,n_\text{h}=1}^{L,N_\text{h}} \mathrm{p}\left(\boldsymbol{u}^{l,n_\text{h}}|F^{l-1}\right)}{\prod_{l=1,n_\text{h}=1}^{L,N_\text{h}} \mathrm{q}\left(\boldsymbol{u}^{l,n_\text{h}}|F^{l-1}\right)} \right] \\
&= \mathbb{E}_{\mathrm{q}(F^L|F^0)} \left[\log \mathrm{p}\left(Y|F^L\right)\right] + \sum\nolimits_{l=1}^L \sum\nolimits_{n_\text{h}=1}^{N_\text{h}} \mathbb{E}_{\mathrm{q}(F^{l-1})} \mathbb{E}_{\mathrm{q}\left(\boldsymbol{u}^{l,n_\text{h}}|F^{l-1}\right)} \left[\log \frac{\mathrm{p}\left(\boldsymbol{u}^{l,n_\text{h}}|F^{l-1}\right)}{\mathrm{q}\left(\boldsymbol{u}^{l,n_\text{h}}|F^{l-1}\right)} \right] \\
&= \mathbb{E}_{\mathrm{q}(F^L|F^0)} \left[\log \mathrm{p}\left(Y|F^L\right)\right] - \sum\nolimits_{l=1}^L \sum\nolimits_{n_\text{h}=1}^{N_\text{h}} \mathbb{E}_{\mathrm{q}(F^{l-1})} \left[\mathrm{KL}\left(\mathrm{q}\left(\boldsymbol{u}^{l,n_\text{h}}|F^{l-1}\right) \| \, \mathrm{p}\left(\boldsymbol{u}^{l,n_\text{h}}|F^{l-1}\right)\right)\right], \qquad (34)
\end{aligned}$$

where $\mathrm{q}(F^L|F^0) = \int \prod_{l=1}^L \mathrm{p}(F^l|\{\boldsymbol{u}^{l,n_\text{h}}\}_{n_\text{h}=1}^{N_\text{h}}, F^{l-1}) \mathrm{q}(\{\boldsymbol{u}^{l,n_\text{h}}\}_{n_\text{h}=1}^{N_\text{h}}|F^{l-1}) \, \{\mathrm{d}\boldsymbol{u}^{l,n_\text{h}}\}_{l=1,n_\text{h}=1}^{L,N_\text{h}} \{\mathrm{d}F^l\}_{l=1}^{L-1}$. In this regard, the first term in the above ELBO can be estimated using Monte-Carlo samples layer-wise with the reparameterization trick (Kingma & Welling, 2013). As introduced in Section 4.1, we consider the independent multi-output Gaussian Processes (Leibfried et al., 2020) by specifying separate s single-output SVGPs. Hence, an independent SVGP is formulated for each of the output dimension, i.e., $\mathrm{q}(\boldsymbol{u}_{[d]}^{l,n_{\mathrm{h}}}|F^{l-1}) = \mathcal{N}(\boldsymbol{m}_{\boldsymbol{u},[d]}, S_{\boldsymbol{uu},[d]})$, $\mathrm{p}(\boldsymbol{u}_{[d]}^{l,n_{\mathrm{h}}}|F^{l-1}) = \mathcal{N}(0, \Lambda^2)$. Thus, the KL-divergence term in (34) for each head index by n_{h} in each layer indexed by l can be expressed in the following form:

$$\begin{aligned}
&\mathrm{KL}\left(\mathrm{q}\left(\boldsymbol{u}^{l,n_{\mathrm{h}}}|F^{l-1}\right) \| \,\mathrm{p}\left(\boldsymbol{u}^{l,n_{\mathrm{h}}}|F^{l-1}\right)\right) = \sum\nolimits_{d=1}^{s} \mathrm{KL}\left(\mathrm{q}\left(\boldsymbol{u}_{[d]}^{l,n_{\mathrm{h}}}|F^{l-1}\right) \| \,\mathrm{p}\left(\boldsymbol{u}_{[d]}^{l,n_{\mathrm{h}}}|F^{l-1}\right)\right) \\
&\overset{(16),(17)}{=} \frac{1}{2}\sum\nolimits_{d=1}^{s} \left[\mathrm{Tr}(\Lambda^{-2} S_{\boldsymbol{uu},[d]}) + \boldsymbol{m}_{\boldsymbol{u},[d]}^{\top} \Lambda^{-2} \boldsymbol{m}_{\boldsymbol{u},[d]} + \log \frac{|\Lambda^2|}{|S_{\boldsymbol{uu},[d]}|} - s\right],
\end{aligned}$$

where s is the number of output dimensions of $\boldsymbol{u}_{[d]}^{l,n_{\mathrm{h}}} \in \mathbb{R}^s$, and also the number of "inducing points" for the SVGPs. This completes the derivation of (20).

C. More Background Materials

In this section, we recall Suykens (2016); Chen et al. (2023) so as to provide a better understanding of KEP-SVGP.

SVD under LS-SVM framework (Suykens, 2016) Let $X \in \mathbb{R}^{N \times M}$ be the data matrix, we first define two sources of data, corresponding to the rows and columns of the data matrix X respectively: $\{\boldsymbol{x}_i := X^{\top}\epsilon_i\}_{i=1}^{N}$, $\{\boldsymbol{z}_j := X\varepsilon_j\}_{j=1}^{M}$ where $\epsilon_i \in \mathbb{R}^N$, $\varepsilon_j \in \mathbb{R}^M$ denote standard basis vectors, that is, the column vectors of the identity matrices I_N and I_M, respectively. The operations from X to $\{\boldsymbol{x}_i \in \mathbb{R}^M\}_{i=1}^{N}$ and $\{\boldsymbol{z}_j \in \mathbb{R}^N\}_{j=1}^{M}$ can be denoted by **two linear feature maps**:

$$\varphi(\boldsymbol{x}_i) := C^{\top}\boldsymbol{x}_i = C^{\top}X^{\top}\epsilon_i, \quad \psi(\boldsymbol{z}_j) := \boldsymbol{z}_j = X\varepsilon_j, \tag{35}$$

where $\varphi : \mathbb{R}^M \to \mathbb{N}$, $\psi : \mathbb{R}^N \to \mathbb{R}^N$ and $C \in \mathbb{R}^{M \times N}$ is a compatibility matrix so that $\boldsymbol{x}_i$, $\boldsymbol{z}_j$ can be compared with each other after applying he feature maps. Now we consider the following constrained optimization problem, i.e., primal problem, which aims at maximizing the projection variances w.r.t. rows and columns data of X:

$$\begin{aligned}
\max_{w,v,e_i,r_j} \quad & J = \frac{1}{2\gamma}\sum\nolimits_{i=1}^{N} e_i^2 + \frac{1}{2\gamma}\sum\nolimits_{j=1}^{M} r_j^2 - w^{\top}v \\
\text{s.t.} \quad & e_i = w^{\top}\varphi(\boldsymbol{x}_i),\ i = 1,\ldots,N, \\
& r_j = v^{\top}\psi(\boldsymbol{z}_j),\ j = 1,\ldots,M,
\end{aligned} \tag{36}$$

where $w, v \in \mathbb{R}^N$ are the projection weights, $e_i, r_j \in \mathbb{R}$ are the projection scores, and $\gamma \in \mathbb{R}$ is the regularization coefficient. This projection variances maximization in (36) exactly follows the spirit of singular value decomposition (SVD).

KSVD problem for self-attention (Chen et al., 2023) Let $\{\boldsymbol{x}_i \in \mathbb{R}^d\}_{i=1}^{N}$ be the input data sequence. The asymmetric self-attention kernel matrix K_{att} can be formulated by $K_{\text{att}} := [\kappa_{\text{att}}(\boldsymbol{x}_i, \boldsymbol{x}_j)] \in \mathbb{R}^{N \times N}$ where $\kappa_{\text{att}}(\boldsymbol{x}_i, \boldsymbol{x}_j) := \langle \phi_q(\boldsymbol{x}_i), \phi_k(\boldsymbol{x}_j) \rangle$ with two feature maps ϕ_q, ϕ_k related to queries and keys, corresponding to (35). Chen et al. (2023) gives the **primal problem** of KSVD of the self-attention mechanism as follows:

$$\begin{aligned}
\max_{W_e,W_r,\Lambda} \quad & J = \frac{1}{2}\sum\nolimits_{i=1}^{N} e(\boldsymbol{x}_i)^{\top}\Lambda^{-1}e(\boldsymbol{x}_i) + \frac{1}{2}\sum\nolimits_{j=1}^{N} r(\boldsymbol{x}_j)^{\top}\Lambda^{-1}r(\boldsymbol{x}_j) - \mathrm{Tr}\left(W_e^{\top}W_r\right) \\
\text{s.t.} \quad & e(\boldsymbol{x}_i) = W_e^{\top}\phi_q(\boldsymbol{x}_i),\ i = 1,\ldots,N, \\
& r(\boldsymbol{x}_j) = W_r^{\top}\phi_k(\boldsymbol{x}_j),\ j = 1,\ldots,N,
\end{aligned} \tag{37}$$

which corresponds to the primal problem in (36). Here, $W_e, W_r \in \mathbb{R}^{p \times s}$ are the projection weights, $\phi_q(\cdot), \phi_k(\cdot) \colon \mathbb{R}^d \to \mathbb{R}^p$ are the feature maps, $e(\boldsymbol{x}_i) = W_e^{\top}\phi_q(\boldsymbol{x}_i) \in \mathbb{R}^s$, $r(\boldsymbol{x}_j) = W_r^{\top}\phi_k(\boldsymbol{x}_j) \in \mathbb{R}^s$ are the projection scores, and $\Lambda \in \mathbb{R}^{s \times s}$ is the regularization coefficient which is a positive diagonal matrix. J in (37) maximizes the projection variances of $W_e^{\top}\phi_q(\boldsymbol{x}_i)$, $W_r^{\top}\phi_k(\boldsymbol{x}_j)$ regarding queries and keys, and involves a regularization term coupling the projections.

With Lagrangian duality and KKT conditions, the **dual problem** of (37) is

$$K_{\text{att}}H_r = H_e\Lambda, \quad K_{\text{att}}^{\top}H_e = H_r\Lambda \tag{38}$$

where $\Lambda \in \mathbb{R}^{s \times s}$ is a positive diagonal matrix serves as singular values of K_{att}, and $H_e = [\boldsymbol{h}_{e_1}, \ldots, \boldsymbol{h}_{e_N}]^{\top} \in \mathbb{R}^{N \times s}$, $H_r = [\boldsymbol{h}_{r_1}, \ldots, \boldsymbol{h}_{r_N}]^{\top} \in \mathbb{R}^{N \times s}$ are the dual variables serving as the left and right singular vectors, respectively. Then the projection scores can be either represented in the primal using explicit feature maps or in the dual using kernel functions:

$$\text{Primal:} \ \begin{cases} e(\boldsymbol{x}) = W_e^{\top}\phi_q(\boldsymbol{x}) \\ r(\boldsymbol{x}) = W_r^{\top}\phi_k(\boldsymbol{x}) \end{cases}, \quad \text{Dual:} \ \begin{cases} e(\boldsymbol{x}) = \sum_{j=1}^{N} \boldsymbol{h}_{r_j}\kappa_{\text{att}}(\boldsymbol{x}, \boldsymbol{x}_j) \\ r(\boldsymbol{x}) = \sum_{i=1}^{N} \boldsymbol{h}_{e_i}\kappa_{\text{att}}(\boldsymbol{x}_i, \boldsymbol{x}). \end{cases}$$

According to Lemma 4.2 in Chen et al. (2023), the solutions H_e, H_r, Λ to the dual problem (38) lead to the zero-value objective J in (37). Therefore, we set $\mathcal{L}_{\mathrm{KSVD}} := J^2$ in (11) as our KSVD regularization loss. By minimizing $\mathcal{L}_{\mathrm{KSVD}}$ to zero through SGD-based optimizers, we realize SVD on K_{att} in self-attention in an approximate way.

D. More Experiment Details

First of all, in addition to reporting good performance on in-distribution datasets, we would like to provide some rationales behind KEP-SVGP's good performance in distribution-shift robustness and out-of-distribution detection, which are two common tasks for Bayesian models:

- With KSVD, we use the pair of adjoint eigenfunctions of the attention kernel as the "inducing features" in two SVGPs with two benefits: *i)* eigenfunctions span an orthogonal system seeking informative feature spaces; *ii)* eigenfunctions in our KSVD promote low-rank property to attention, where noisy patterns could be filtered out.

- Recall that the distribution-shift data corrupts the clean data with shifts, while out-of-distribution (OOD) data are from another different distribution. *i)* As distribution shifts can serve as noisy patterns, the low-rank property can help filtering out these noises. *ii)* As the feature space of OOD data can be largely different from that of the in-distribution data, the informative features by the eigenfunctions can help distinguish such differences.

D.1. More Details on Experimental Setups

All experiments presented in this work are implemented with PyTorch, which can be conducted on a single NVIDIA GeForce RTX 2070 SUPER GPU. Our implementation is available at https://github.com/yingyichen-cyy/KEP-SVGP.

Experiments on CIFAR-10 and CIFAR-100 For both CIFAR-10 and CIFAR-100, we randomly split the original training set into 90% training and 10% validation set, leading to a training set of 45K samples and a validation set of 5K. The test set is of 10K samples. For both datasets, we use 7-layer ViT (Dosovitskiy et al., 2021) where the 32×32 input images are tokenized with patches of size 4×4, the embedding dimension is 384, the hidden dimension is 384, the number of heads is 12, the dropout ratio is 0.1, and the classification token is turned off. For all KEP-SVGPs on these two datasets, we set the regularization constant of KSVD loss in our objective $\min -\mathcal{L}_{\mathrm{ELBO}} + \eta \mathcal{L}_{\mathrm{KSVD}}$ as $\eta = 10$, set the rank for KSVD as $s = 10$. In our experimental section, we choose the feature maps related to the cosine similarity kernel on queries and keys as in Chen et al. (2023). All models are trained from scratch with ADAM optimizer (Kingma & Ba, 2015), except the post-hoc methods including Temperature Scaling and KEFLLLA, for 300 epochs with 5 warm-up epochs. The batch size is 128, and a cosine learning rate schedule is utilized with a learning rate of 10^{-3} and minimum learning rate of 10^{-5}. Ensemble methods are based on the models trained independently over 5 trials. More specifically, deep Ensembles is implemented by the mean ensembling of 5 independently trained transformers, which is the referred "regular transformer-based model along with ensembling techniques". KEP-SVGP Ensembles does the same for 5 independently trained KEP-SVGP transformers. The best models are selected with the best validation accuracy. During inference, for MC Dropout and our KEP-SVGP, predictive uncertainty is estimated using 10 Monte Carlo samples. Note that SGPA (Chen & Li, 2023) is very time and memory consuming with 7-layer architectures, therefore we do not include it in the comparisons on CIFAR datasets in Table 1. However, we do include SGPA in Table 4 with all models trained with the same architectures as done in its original paper (Chen & Li, 2023) for fair comparisons: 5-layer ViT on CIFAR-10, 6-layer ViT on CIFAR-100.

Experiments on IMDB We randomly split the IMDB original training set into 35K as training and 5K as validation, the test set is of 10K samples. IMDB is with the maximum sequence length of 512. Following Chen & Li (2023), we use 1-layer Transformer (Vaswani et al., 2017) where the embedding dimension is 128, the hidden dimension is 128, the number of heads is 8, and the dropout ratio is 0.1. For our KEP-SVGP with addition scheme (19) for merging of SVGPs outputs, we set KSVD regularization constant as $\eta = 10$, the KSVD rank as $s = 10$, with feature maps related to the cosine similarity kernel on queries and keys (Chen et al., 2023). We train all models with ADAM optimizer, except for the post-hoc methods including Temperature Scaling and KFLLLA, for 20 epochs with 5 warm-up epochs, a batch size of 32, and a initial learning rate 10^{-3} which decays to 10^{-4} following a cosine learning rate decay. Ensemble methods are based on the models trained independently over 5 trials. The best models are selected with the best validation accuracy. During inference, for MC Dropout, SGPA and our KEP-SVGP, predictive uncertainty is estimated with 10 Monte Carlo samples.

Table 6. Comparisons on $W_1^{\text{cat}} \in \mathbb{R}^{N\times 2N}$ and $AB^\top$, $A \in \mathbb{R}^{N\times s}$, $B \in \mathbb{R}^{2N\times s}$ on CIFAR-10. Forward time (s) is on a single V100.

Concatenation	Forward Time (s)	ACC ↑	AURC ↓	AUROC ↑	FPR95 ↓	ECE ↓	NLL ↓	Brier ↓
$W_1^{\text{cat}} \in \mathbb{R}^{N\times 2N}$	0.08±0.03	84.70±0.61	**35.15**±2.65	**87.20**±0.65	**64.93**±1.41	10.60±0.45	**8.00**±0.56	**25.45**±1.05
$AB^\top$, $A \in \mathbb{R}^{N\times s}$, $B \in \mathbb{R}^{2N\times s}$	**0.05**±0.02	**84.71**±0.10	36.01±1.01	86.63±0.31	67.72±1.08	**10.53**±0.22	8.03±0.32	25.57±0.33

Table 7. KEP-SVGP serves as a good complement to other methods for improving calibration. Experimental results are reported over five trials, with the best mean results shown in bold. ACC, AUROC, FPR95, ECE and Brier are percentages, AURC is $\times 10^3$, NLL is $\times 10$.

Dataset	Method	ACC/MCC ↑	AURC ↓	AUROC ↑	FPR95 ↓	ECE ↓	NLL ↓	Brier ↓
IMDB (Maas et al., 2011)	MSP (Hendrycks & Gimpel, 2017)	88.17±0.52	35.27±3.04	82.29±0.87	71.41±1.57	4.01±1.36	3.10±0.26	17.88±0.95
	Temperature Scaling (TS) (Guo et al., 2017)	88.17±0.52	35.27±3.04	82.29±0.87	71.08±1.55	**1.05**±0.70	2.89±0.12	17.40±0.80
	KEP-SVGP (ours)	89.01±0.14	**30.69**±0.69	**83.22**±0.31	**68.15**±0.95	3.72±0.81	3.00±0.13	16.56±0.25
	KEP-SVGP+TS (ours)	**89.01**±0.14	30.71±0.76	**83.22**±0.36	68.38±0.60	1.19±0.19	**2.73**±0.03	**16.21**±0.16
CoLA (Warstadt et al., 2019)	MSP (Hendrycks & Gimpel, 2017)	26.93±1.38	205.47±7.62	64.55±0.86	89.86±1.29	23.84±2.23	14.45±2.83	52.15±2.43
	KFLLLA (Kristiadi et al., 2020)	26.90±1.31	204.31±8.57	64.60±0.96	90.06±0.74	**2.51**±1.09	5.94±0.04	40.52±0.38
	KEP-SVGP (ours)	30.54±1.61	186.66±8.50	65.16±0.86	88.39±0.83	15.89±3.48	8.54±1.66	43.55±2.99
	KEP-SVGP+KFLLLA (ours)	**31.22**±**1.63**	**185.58**±6.57	**65.49**±1.31	**87.97**±2.12	5.81±2.19	**5.78**±0.04	**38.98**±0.44

Experiments on CoLA This dataset provides an in-distribution training with 8551 samples and a in-distribution test of 527. Following Chen & Li (2023), we use 2-layer Transformer (Vaswani et al., 2017) where the embedding dimension is 128, the hidden dimension is 256, the number of heads is 4. For the input embedding, we adopt ELMO-style representation (Peters et al., 2018). For our KEP-SVGP with addition merging scheme in (19), we set KSVD regularization constant as $\eta = 1$, the KSVD rank as $s = 5$, with feature maps ϕ_q, ϕ_k in (10) related to the cosine similarity kernel (Chen et al., 2023). We train all models with ADAM optimizer, except for the post-hoc methods including Temperature Scaling and KFLLLA, for 50 epochs with 5 warm-up epochs, a batch size of 32, and a initial learning rate 5×10^{-4} which decays to 10^{-5} following a cosine learning rate decay. Ensemble methods are based on the models trained independently over 5 trials. During inference, for MC Dropout, SGPA and our KEP-SVGP, predictive uncertainty is estimated using 10 Monte Carlo samples.

D.2. More Details on Concatenation Weights

For the concatenation merging scheme in Section 4.1, we can replace $W_1^{\text{cat}} \in \mathbb{R}^{N\times 2N}$ with $AB^\top$ where $A \in \mathbb{R}^{N\times s}$, $B \in \mathbb{R}^{2N\times s}$ to maintain the overall linear time complexity with N when computing KEP-SVGP. We provide empirical evaluations on comparing these two concatenation weight schemes in Table 6. By utilizing $AB^\top$, comparable performances are obtained at less computational cost. However, we still adopt W_1^{cat} for all experiments in this paper so as to maintain good failure prediction and calibration performances.

D.3. More Details on Calibration Improvement

We discuss the further potentials of KEP-SVGP here. The benefits of our KEP-SVGP can beyond the existing evaluations, as the key idea of fully utilizing the asymmetry in self-attention is compatible with other calibration method and thus serves as a good complement for a further boost in performances, as demonstrated in Table 7. Specifically, we find that

- KEP-SVGP+TS on IMDB: Temperature Scaling (TS) is a post-hoc method tuning the temperature scale inside the softmax probability in the classification head. As KEP-SVGP directly models the attention blocks, it can be combined with TS, leading to an improved calibration.
- KEP-SVGP+KFLLLA on CoLA: KFLLLA is a post-hoc method working on the last linear layer in the classification head. Hence, fine-tuning the classification head with KFLLLA on pre-trained KEP-SVGP transformer improves both failure prediction and calibration metrics.

D.4. More Details on Time Efficiency

In addition to Table 4, we also report forward time in seconds during training in Table 8 so as not to include the time taken by the optimizer. It can be seen that we reach the same conclusion as in Section 5.2 that KEP-SVGP distinctively outperforms MSP on all datasets with only a slightly extra forward time during training.

Table 8. Performance and forward time (s) on a single NVIDIA Tesla V100 SXM2 32 GB. Results are reported over five trials.

Method	Time Complexity	CIFAR-10		CIFAR-100		IMDB		CoLA	
		ACC ↑	Forward Time (s)	ACC ↑	Forward Time (s)	ACC ↑	Forward Time (s)	MCC ↑	Forward Time (s)
MSP (Hendrycks & Gimpel, 2017)	$\mathcal{O}(N^2)$	78.11±0.10	**0.02**±0.01	52.16±0.50	**0.02**±0.02	88.17±0.52	**0.001**±0.0	26.93±1.38	0.40±0.33
SGPA (Chen & Li, 2023)	$\mathcal{O}(N^2 s)$	77.87±0.12	0.15±0.0	53.02±0.36	0.27±0.0	88.36±0.75	0.63±0.07	26.15±1.12	0.47±0.37
KEP-SVGP (ours)	$\mathcal{O}(Ns^2)$	**78.27**±0.30	**0.02**±0.02	**56.26**±0.70	0.03±0.01	**89.01**±0.14	0.01±0.0	**30.54**±1.61	**0.37**±0.41

Table 9. Ablation on KSVD regularization constant η and the merging scheme of the SVGP pair on CIFAR-10.

η	ACC ↑	AURC ↓	AUROC ↑	FPR95 ↓	ECE ↓	NLL ↓	Brier ↓
			$[F^e_{[d]}; F^r_{[d]}]$				
10	**84.70**	**35.43**	87.17	**62.55**	**10.71**	**7.95**	**25.32**
1	83.77	39.28	87.08	66.36	12.09	9.67	27.64
0.1	83.78	37.39	**87.86**	64.12	12.04	9.62	27.46
0.01	82.60	44.60	86.17	67.13	13.12	11.16	29.85
			$F^e_{[d]} + F^r_{[d]}$				
10	84.04	38.99	86.64	66.04	11.64	9.39	27.02
1	84.45	37.32	86.97	65.02	11.30	8.80	26.27
0.1	83.56	39.74	87.03	63.63	12.22	9.49	27.84
0.01	81.46	49.71	85.81	66.51	13.94	11.24	31.35

D.5. Additional Ablations

There are two main hyper-parameters in KEP-SVGP: the regularization constant of KSVD loss η, the rank of KSVD s. For the rank s, we adopt the default settings in Chen et al. (2023) and set $s \in \{5, 10\}$. As η balances $\mathcal{L}_{\text{ELBO}}$ and $\mathcal{L}_{\text{KSVD}}$, it is of importance for KEP-SVGP. Therefore, we provide the ablation of η with rank $s = 10$ fixed on CIFAR-10 (Krizhevsky et al., 2009), and also the ablation on the addition and concatenation merging schemes given in (19) in Table 9, so as to give a guideline of the choice of schemes.

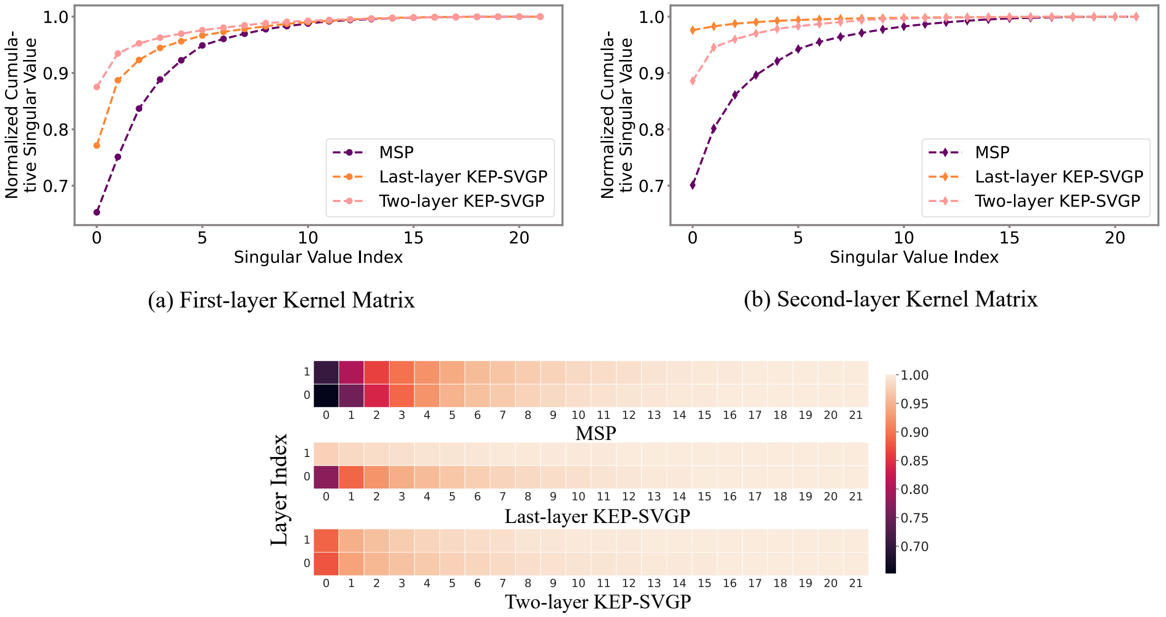
It can be seen in Table 9 that the concatenation merging scheme has an overall better performance among all metrics than the addition scheme. The possible reason can be that positive and negative output values from the two SVGPs branches can cancel each other out when employing the addition scheme, while the concatenation scheme can preserve the outputs information. However, when dealing with language modelling tasks, we employ the addition scheme since it is sequence length independent while most of the language datasets are with varying sequence length. We also find that $\eta = 10$ returns better performances than choosing a small one. A larger η can help the model to conduct effective KSVD in an early stage, contributing to the construction of more accurate SVGPs branches and thereby leading to better overall performances.

D.6. Low-rank Property in Attention Kernel Matrix

To further validate the approximate posterior distributions described in Remark 3.1, we present empirical evidence delving into the low-rank property resided in the Transformer models. We adopt the two-layer Transformer on CoLA following the setups in Appendix D.1. Specifically, we consider 3 different models: *i)* MSP; *ii)* last-layer KEP-SVGP; *iii)* two-layer KEP-SVGP. All our methods here are with addition merging scheme. The spectrum analysis of the self-attention kernel matrix in each layer of each model is provided in the following Figure 4.

According to the results in Figure 4, we find that

- The attention matrix in the layers of Transformer has low-rank property, though the shallow layer may not desires the low-rank property as much as the deeper layer. This is consistent with the findings in Chen et al. (2023). (Figure 4(c))
- Most of the information ($> 95\%$ explained variance) of the attention matrix in both layers in the MSP baseline can be captured by the top-5 singular vectors. Thus, the hyperparameter $s = 5$ in KSVD of our method is reasonable and approaches the ground-truth rank of the attention kernel. Our method captures distinctively higher explained variances in the top singular vectors than the MSP baseline. (Figure 4(a), Figure 4(b))
- Applying KEP-SVGP only to the last layer also enhances the low-rank property of the attention in other layers, as shown in the comparisons between our last-layer KEP-SVGP and two-layer KEP-SVGP. (Figure 4(a), Figure 4(b))



(a) First-layer Kernel Matrix

(b) Second-layer Kernel Matrix

(c) Low-rank of Each Layer

Figure 4. Spectrum analysis of the self-attention kernel matrix on CoLA. Specifically, we consider the normalized cumulative singular values of the attention matrix of the two-layer Transformer models. (a) plots the spectrum results of the first-layer kernel matrix; (b) plots the spectrum results of the second-layer kernel matrix; (c) plots the normalized cumulative singular values w.r.t. singular value index of each layer, showing the low-rank property of the attention matrix of each model.

D.7. Additional Visualization on Distribution-shift Data

We provide comparisons of our KEP-SVGP with all other baselines under distribution shift in Figure 5. We report the mean AURC results of all 5 severity levels under 15 types of corruption on CIFAR-10-C (Hendrycks & Dietterich, 2019). All models are trained on CIFAR-10/ViT following the setups in Appendix D.1, and then tested on CIFAR-10-C. Our KEP-SVGP has overall better and stable AURC than all other comparing methods.

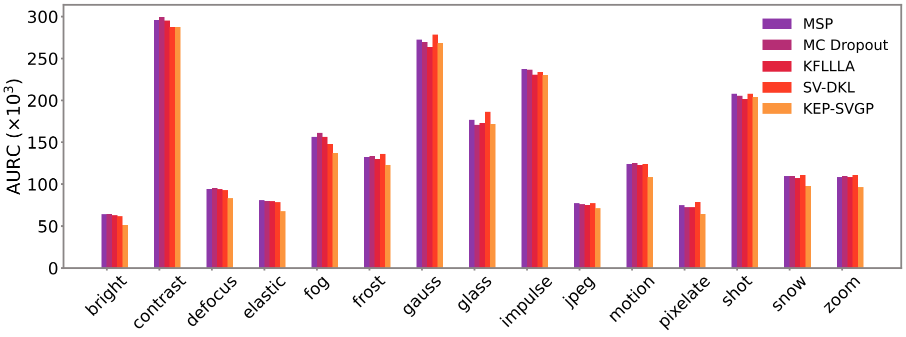


Figure 5. Comparisons of our KEP-SVGP with baselines under distribution shift. Mean AURC results of all 5 severity levels under 15 types of corruption are reported, where models are trained on CIFAR-10/ViT and tested on CIFAR-10-C.