IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 30, 2021, accepted February 25, 2021, date of publication February 26, 2021, date of current version March 9, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3062747

Incremental Kernel Principal Components Subspace Inference With Nyström Approximation for Bayesian Deep Learning

YONGGUANG WANG, SHUZHEN YAO, AND TIAN XU
School of Computer Science and Engineering, Beihang University, Beijing 100191, China
Corresponding author: Yongguang Wang (wangyongguang@buaa.edu.cn)

ABSTRACT As the state-of-the-art technology of Bayesian inference, based on low-dimensional principal components analysis (PCA) subspace inference methods can provide approximately accurate predictive distribution and well calibrated uncertainty. However, the main problem of PCA method is that it is a linear subspace feature extractor, and it cannot effectively represent the nonlinearly high-dimensional parameter space of deep neural networks (DNNs). Firstly, in this paper, in order to solve the main problem of the linear characteristics of PCA in high-dimensional space, we apply kernel PCA to extract higher-order statistical information in parameter space of DNNs. Secondly, to improve the efficiency of subsequent computation, we propose a strictly ordered incremental kernel PCA (InKPCA) subspace of parameter space within stochastic gradient descent (SGD) trajectories. In the proposed InKPCA subspace, we employ two approximation inference methods: elliptical slice sampling (ESS) and variational inference (VI). Finally, to further improve the memory efficiency of computing the kernel matrix, we apply Nyström approximation to determine the suitable size of subsets in the original datasets. The novelty of this paper is that it is the first time to apply the proposed InKPCA subspace with Nyström approximation for Bayesian inference in DNNs, and the results show that it can produce more accurate predictions and well-calibrated predictive uncertainty in regression and classification tasks of deep learning.

INDEX TERMS Bayesian deep learning, incremental kernel PCA, elliptical slice sampling, variational inference, Nyström approximation.

I. INTRODUCTION

In key fields where safety is at stake, such as medical diagnoses and self-driving vehicles. Uncertainty estimates of deep learning models are very important for decision making to help to prevent dangerous accidents. However, deep learning models are usually miscalibrated and overconfident in predictions [1], it is very useful to add a credible uncertainty estimates to the predicted values [2].

Bayesian methods were regarded as gold standard, they could provide probabilistic uncertainty estimates and were once widely used for inference in machine learning models [3]–[5]. Unfortunately, Bayesian methods are inefficient in millions of parameters and high-dimensional parameter space, which are extensively existing in deep learning domain with high-dimensional datasets and complex parameter space in neural network architectures. The characteristic of Bayesian approaches limits their wide application in deep learning field.

In order to solve the problems mentioned above, the existing method adopts principal components analysis (PCA) method to construct a low-dimensional subspace of parameter space for Bayesian inference [6]. In reality, there are two most popular techniques for dimensionality reduction: PCA and Linear Discriminant Analysis (LDA, also named as Fisher Discriminant Analysis-FDA) [7]. LDA's limitation is to search for those vectors in the underlying space that can best discriminate among classes (rather than those vectors that can best describe the data) [8], whereas PCA achieves the data in its entirety for principal components analysis without paying any particular attention to the underlying class structure. By comparing the characteristics of PCA and LDA, PCA-based dimensionality reduction technology meets the requirements of constructing weight subspace from the weight in its entirety.

The associate editor coordinating the review of this manuscript and approving it for publication was Liangxiu Han.

However, PCA is a linear subspace feature extractor and has somewhat reached a performance barrier due to linear assumptions of the underlying generative phenomena, the subspace obtained by it is inadequate for high-dimensional parameter space in deep learning models [9], and resulting in the inability to obtain more accurate results. In this paper, in order to solve the main problem of linear PCA in extracting nonlinearly high-dimensional parameter space features in deep learning models, we apply kernel PCA(KPCA) to extract higher-order statistical information from parameter space of deep neural networks (DNNs). Studies have shown that KPCA always obtain better performance than PCA [10]. Although KPCA can extract non-linear features in high-dimensional space, it increases the space and time complexity compared to PCA, and the computation of standard KPCA results in a space complexity of $O(n^2)$ and a time complexity of $O(n^3)$.

In order to improve the computational efficiency problem caused by the introduction of KPCA, there are batch-based modeling method and the incremental approach.

The first class is the batch-based modeling method, which requires entire training data for estimating KPCA. For example, kernel Hebbian algorithm (KHA) can compute KPCA without storing the kernel matrix by kernelizing the generalized Hebbian algorithm and can deal with largescale datasets with high dimensionality. However, KHA has a scalar parameter which leads to slow convergence during training phase [11].

The second class is the incremental approach. In incremental version of KPCA (InKPCA), singular value decomposition is utilized to update an eigen feature space for incoming data, and it is time-consuming in processing high-dimensional data [12], [13]. Furthermore, InKPCA can apply rank one updates (ROU) algorithm to deal with the eigendecomposition of kernel matrix, but it needs to save entire collected samples for evaluating on a new sample [14]. Meanwhile, Nyström method can approximate the eigendecomposition of the Gram matrix and can further reduce the space complexity to $O(m^2n)$ [15]. And in some cases, the traditional InKPCA with ROU algorithm needs to be improved to adapt to ordered samples.

The above is mainly to introduce how to effectively obtain the subspace. However, where does the subspace exist? Existing studies believe that the subspace lies in the trajectories of the parameter optimization method stochastic gradient descent (SGD) [6], [16]. As a standard deep neural networks (DNNs) training optimization algorithm, SGD can escape local minima and converge to a better minimum, even if the loss function of DNNs is not differentiable everywhere [17]. More importantly, when PCA or InKPCA method has successfully held the subspace of the SGD trajectories, the obtained subspace contains different high-performance models and can efficiently approximate the posterior distribution over the weights of the neural network.

Since performing Bayesian inference on DNNs requires the posterior distribution $p(\theta|D)$, the posterior distribution for DNNs cannot be calculated analytically or even efficiently sampled from, so our goal is to approximate it over the parameter θ with dataset D in the obtained low-dimensional PCA or InKPCA subspace. There are stochastic approximation and deterministic approximation methods, which can be used to verify the advantages and disadvantages of the generated subspace.

Firstly, for stochastic approximation methods of $p(\theta|D)$, existing methods such as Hamiltonian Monte Carlo (HMC) makes Markov chain Monte Carlo (MCMC) once become a gold standard for posterior inference with neural networks [18], however, a limitation of HMC methods is the required intensive gradient computation when evaluating the of log-posterior on the full data and HMC approaches are not applicable to large datasets. So stochastic gradient HMC (SGHMC) [19] is proposed to use stochastic estimates of the gradient to avoid the costly fully gradient computation, and SGHMC applies second-order Langevin dynamics with a friction term to counteract the effects of the noisy gradient to maintain the target distribution $p(\theta|D)$ as the invariant distribution for posterior inference. Meanwhile, stochastic gradient Langevin dynamics (SGLD) applies first order Langevin dynamics to the stochastic gradient framework [19], [20]. Although SGHMC and SGLD asymptotically sample from the posterior along with step sizes infinitely tends to zero, using finite learning rates schedule to train the neural network, which leads to inaccurate approximate solutions [21]. Meanwhile, Elliptical Slice Sampling (ESS) as a new MCMC algorithm for performing in models with multivariate Gaussian priors. Its key properties are: 1) it has simple and generic code applicable to many models, 2) it has no free parameters, 3) it works well for a variety of Gaussian process based models [22]. In this paper, the constructed subspace exists in the parameter space of DNNs model with relatively low training loss, there are extensively reasonable priors exist in the subspace, and the hypothesis with multivariate Gaussian priors is a good choice for Bayesian inference, so we choose ESS to perform posterior inference in the constructed subspace.

Secondly, for deterministic approximation approaches of $p(\theta|D)$, the purpose is to fit a Gaussian variational posterior as a practical variational inference (VI) to approximate neural network weights [23]. Alternatively, VI approaches can be realized by reparameterization trick for training deep models with latent variable [24], [25].

Next, we will perform the approximate posterior inference methods ESS and VI within the constructed InKPCA subspace for predictive uncertainty estimates in regression and classification tasks. Of course, there are many other methods for uncertainty estimates, such as the existing SGD-based studies include stochastic weight average (SWA) and SWA-Gaussian (SWAG), these methods form an approximate Gaussian posterior over weights [26], or subspaces containing low-loss curves between independently trained

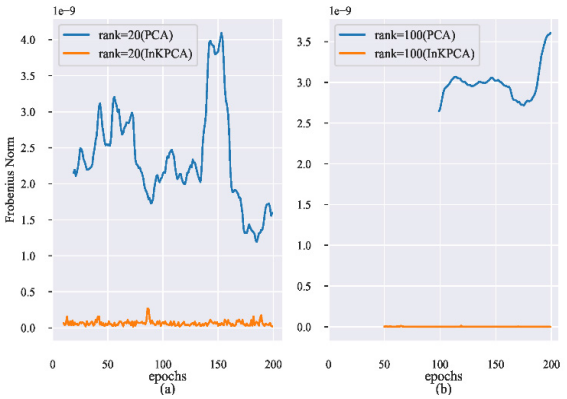

FIGURE 1. The Frobenius norms between PCA subspace and the proposed InKPCA subspace with rank = 20 and rank = 100 respectively.

solutions [27]. Alternatively, the application of a block Kronecker factored (KFAC) approximation to the Hessian matrix for Laplace approximations [28].

In this paper, with the training of the DNNs, the deviation vectors of weights are collected in strict order and the estimation of KPCA does not require the entire deviation vectors of weights. So it is different from the batch-based modeling method, which requires entire training data for estimating KPCA. And our purpose is to obtain the principal deviation vectors of weights and then construct weights posterior.

So we propose an improved InKPCA subspace with ROU algorithm with strictly ordered deviation vectors of weights, and then Nyström approximation is adopted to further reduce the space complexity. Finally, we perform approximate Bayesian inference in the constructed InKPCA subspace of the deep learning models [6], several steps are as follows.

Firstly, KPCA is applied to achieve higher-order statistical information in parameter space of DNNs. Secondly, to improve calculation efficiency of KPCA, strictly ordered incremental method and Nyström approximation are adopted to construct a low-dimensional InKPCA parameter subspace. Finally, posterior inference over weights is performed in the proposed InKPCA subspace and then Bayesian model averaging can be implemented by sampling weight parameters in InKPCA subspace to achieve the uncertainty estimates of deep learning.

In Fig.1, we mainly compare the Frobenius norm between the proposed InKPCA subspace and PCA subspace in a regression task, which is realized in section IV-A. The results show that Frobenius norm of the proposed InKPCA subspace is improved at least 100 times than that of PCA subspace, which means that the proposed InKPCA subspace is more representative and accurate than PCA subspace for original parameter space.

To summarize, our contributions are as follows:

1) We are the first to replace PCA with KPCA in extracting higher-order statistical information in parameter space of DNNs.
2) We propose an improved InKPCA approach to solve the computational efficiency problem. Firstly, the proposed InKPCA applies incremental algorithm ROU to compute the eigendecomposition of kernel matrix, which is generated by the kernelization of strictly ordered deviation vectors. Secondly, we use Nyström approximation to determine the suitable size of subsets to further improve the memory efficiency.
3) The experimental results show that the proposed InKPCA method can not only provide higher or comparable accuracy of the regression and classification tasks, but also better calibrated uncertainty estimates for DNNs.

The remainder of this paper is organized as follows. Section II presents the definition of InKPCA, Bayesian model averaging and approximate inference methods. Section III describes the details of our approaches.

Section IV implements two approximate inference approaches VI and ESS within the proposed InKPCA subspace on a regression task in the section IV-A, on several UCI regression tasks in the section IV-B, and on CIFAR-10 and CIFAR-100 image classification tasks in the section IV-C. Section V is the conclusion.

II. BAYESIAN INFERENCE WIHTIN INKPCA SUBSPACE

In this section we introduce the definition of InKPCA subspace in section II-A, Bayesian model averaging in section II-B and approximation inference approaches in section II-C.

A. DEFINITION OF INKPCA SUBSPACE

Assuming model M has weight parameters $w \in \mathbb{R}^d$ and a likelihood function $p_M(D|w)$ with dataset D. We can implement Bayesian inference in a n-dimensional subspace S, which can be defined as follows [6]:

$$\begin{aligned} S &= \left\{w|w = \hat{w} + \theta_1 v_1 + \cdots + \theta_n v_n\right\} \\ &= \left\{w|w = \hat{w} + P * \theta\right\} \end{aligned} \tag{1}$$

where projection matrix $P = (v_1, \cdots, v_n)^{\mathrm{T}} \in \mathbb{R}^{d\times n}$, fixed $\hat{w} \in \mathbb{R}^d, \theta = (\theta_1, \cdots, \theta_n)^{\mathrm{T}} \in \mathbb{R}^n$, and $*$ represents the multiplication of two elements.

According to the viewpoint of Izmailov *et al.* [6], the projection matrix $P = (v_1, \cdots, v_n)^{\mathrm{T}}$ is generated by running PCA based on truncated randomized singular value decomposition (SVD) [29] on the deviation matrix and using the first n principal components $v_1, \cdots, v_n$ to define the subspace P. In this paper, however, the generation process of subspace P is defined as follows: after the model is trained in a new epoch, a new deviation v_{n+1}^{T} is generated and added to $(v_1, \cdots, v_n)^{\mathrm{T}}$, and then InKPCA is applied to the $(n+1)$ deviation vectors $(v_1^{\mathrm{T}}, \cdots, v_n^{\mathrm{T}}, v_{n+1}^{\mathrm{T}})$ to produce n principal components deviation vectors $(v_1, \cdots, v_n)^{\mathrm{T}}$,which are served as the new projection matrix P.

And the new model M has the likelihood function:

$$p(D|\theta) = p_M(D|w = \hat{w} + P * \theta) \tag{2}$$

Equation (2) is not a reparameterization of the primitive model, θ serves as parameters of low-dimensional subspace

P, the subspace model can be parameterized by θ to share different functional properties with M, then we perform Bayesian inference over subspace parameters θ.

B. BAYESIAN MODEL AVERAGING

In testing phase of deep learning, in order to implement Bayesian model averaging on test dataset x, y, we can perform a Monte Carlo estimate of the integral as follows:

$$\begin{aligned} p(y|D,x) &= \int p_M(y|x, w = \hat{w} + P * \theta) p(\theta|D) d\theta \\ &\approx \frac{1}{n} \sum_{i=1}^{n} p_M(y|x, w = \hat{w} + P * \theta_i), \theta_i \sim p(\theta|D) \end{aligned} \tag{3}$$

where P represents the obtained projection matrix, w stands for transforming sampled n samples θ into original space as $w = \hat{w} + P * \theta$. θ_i means posterior sampling from $p(\theta|D)$.

To overcome the posterior concentration in the subspace, the temperature hyperparameter T is applied to scale the likelihood. Our purpose is to maximize the log tempered posterior respect to parameter θ described as follows [6]:

$$\log p(\theta|D) \propto \frac{1}{T} \log p(D|\theta) + \log p(\theta)$$

By adjusting $T = 1$ we can get true posterior and as $T \to \infty$ posterior approaches the prior $p(\theta)$,which is viewed as a regularizer in optimization.

In order to sample θ_i, we use deterministic approximation approach VI and stochastic approximation method ESS to estimate the integral for Bayesian model averaging in (3).

C. APPROXIMATE INFERENCE APPROACHES

Performing Bayesian inference on DNNs requires the posterior distribution $p(\theta|D)$. However, the posterior for DNNs cannot be calculated analytically or even efficiently sampled from. As stated in the introduction section, deterministic approximation approach VI and stochastic approximation method ESS are adopted for approximating $p(\theta|D)$.

Firstly, VI addresses this problem by approximating $p(\theta|D)$ with a more tractable distribution $q(\theta|\alpha)$, the purpose of VI is to find the parameter α of the distribution $q(\theta|\alpha)$ on the weights θ that can minimise the Kullback-Leibler (KL) divergence with the true Bayesian posterior $p(\theta|D)$ on the weights θ:

$$\begin{aligned} \alpha^* &= \arg\min_{\alpha} \mathrm{KL}\left[q(\theta|\alpha)||p\left(\theta|D\right)\right] \\ &= \arg\min_{\alpha} \int q(\theta|\alpha) \log \frac{q(\theta|\alpha)}{p\left(\theta\right) p\left(D|\theta\right)} \\ &= \arg\min_{\alpha} \left(\mathrm{KL}\left[q(\theta|\alpha)||p\left(\theta\right)\right] - \mathrm{E}_{q(\theta|\alpha)}\left[\log p\left(D|\theta\right)\right]\right) \end{aligned} \tag{4}$$

The resulting cost function of (4) is known as the variational free energy [3], [23], and the negative variational free energy is also called Evidence Lower Bound (ELBO), which can be expressed as:

$$F(D,\alpha) = \mathrm{E}_{q(\theta|\alpha)}\left[\log p\left(D|\theta\right)\right] - \mathrm{KL}\left[q(\theta|\alpha)||p\left(\theta\right)\right] \tag{5}$$

According to (4) and (5), maximising the cost function ELBO $F(D,\alpha)$ is equivalent to minimising the KL divergence between the approximate distribution $q(\theta|\alpha)$ and the true posterior $p(\theta|D)$. In practice, not requiring a closed form of KL divergence allows many combinations of prior $p\left(\theta\right)$ and variational posterior. We can approximate the exact ELBO $F(D,\alpha)$ as follows:

$$F(D,\alpha) \approx \log p(D|\theta_i) p(\theta_i) - \sum_{i=1}^{n} \log q(\theta_i|\alpha) \tag{6}$$

where θ_i denotes the ith Monte Carlo sample drawn from the variational posterior $q(\theta_i|\alpha)$.

Secondly, ESS as a MCMC algorithm can perform inference in models with multivariate Gaussian priors, and it has three key advantages:

1) simple and generic enough code applicable to abundant models;
2) no free parameters;
3) especially work well for Gaussian Process based models, which is very matched for our algorithms.

Ultimately, these sampling methods can achieve mixtures of Gaussian predictive distributions for regression tasks and categorical distributions for classification tasks.

Finally, we summarize the Bayesian inference within the proposed InKPCA subspace in Algorithm 1, which includes four steps: (1) construct an InKPCA subspace; (2) rebuild kernel matrix with Nyström approximation method; (3) perform posterior inference within InKPCA subspace; (4) form a Bayesian model averaging. Adopting ESS and VI to approximately sample parameters from subspace for posterior inference.

Algorithm 1 InKPCA Subspace Inference With Nyström Approximation

Input: data D; model M;

1. Construct an Incremental Kernel PCA subspace, i.e. using Algorithm 2 (section III-A).
2. Rebuild kernel matrix with Nyström approximation method, i.e. using Algorithm 3 (section III-B).
3. Perform VI and ESS posterior inference within the constructed InKPCA subspace (section III-C).
4. Form a Bayesian model averaging(section II-B).

III. INKPCA SUBSPACE CONSTRUCTION AND APPROXIMATE INFERENCE WITHIN THE SUBSPACE

In section II we introduce the definition of InKPCA subspace and Bayesian inference approaches in InKPCA subspace. In this section we present how to construct the proposed InKPCA subspace in section III-A and III-B, and how to perform approximate inference methods VI and ESS within the constructed InKPCA subspace in section III-C.

A. INKPCA SUBSPACE WITHIN SGD TRAJECTORIES

Izmailov *et al.* [6] proposed to run SGD from a pre-trained solution and capture w_i of weights at end of each of T epochs, $w_{\text{swa}}=(n^* w_{\text{swa}} + w_{\text{i}})/(n+1)$ is obtained according to a certain update frequency c, and then store deviations $d_i = w_{\text{swa}} - w_i$ for the last M epochs to form a deviation matrix D, which is comprised of M vectors $d_1,\ldots, d_M$, M ($M = 20$ in [6]) is determined by the available amount of memory. Izmailov *et al.* performed PCA based on randomized SVD on the deviation matrix D and used the first n principal componets $(v_1, \cdots, v_{\text{n}})^{\text{T}}$ to define a PCA subspace.

Meanwhile, Schölkopf*et al.* [10] presented that KPCA as a nonlinear subspace feature extractor can achieve the higher-order statistical information. To improve the computation efficiency of KPCA, strictly ordered incremental method and Nyström approximation are adopted to construct a low-dimensional InKPCA parameter subspace, we adopt the proposed InKPCA to deal with the new deviation vector.

Assuming we have stored the deviation vector $d_i = w_{\text{swa}} - w_i$ and M is determined by the capacity of memory. The proposed InKPCA is applied to decompose the kernel matrix of D and generate L_{nys}andU_{nys}, where L_{nys}andU_{nys} represent that the eigenvalues and eigenvectors are approximated by Nyström method. The construction process of InKPCA subspace is presented in Algorithm 2.

Algorithm 2 Subspace Construction With InKPCA

Input: w_0: pretrained weights; η: learning rate; T: number of steps; c: moment update frequency; D:deviation matrix; M: maximum number of columns in deviation matrix; r: rank of InKPCA approximation; P: projection matrix for subspace.

Output: w_{swa} , P

$w_{\text{swa}} \leftarrow w_0$
for $i \leftarrow 1, 2, \ldots T$ **do**
  $w_i \leftarrow w_{i-1} - \eta \nabla_w L(w_{i-1})$
if MOD$(i, c) = 0$ **then**
  $n \leftarrow i/c$
  $w_{\text{swa}} \leftarrow (n * w_{\text{swa}} + w_{\text{i}})/(n+1)$
  If NUM_COLS$(D) \geq M$ **then**
    REMOVE_COL$(D, w_i - w_{\text{swa}})$
  APPEND_COL$(D, w_i - w_{\text{swa}})$
  $L_{\text{nys}}, U_{\text{nys}} =$ InKPCA(D)
$P = U_{\text{nys}} * \sqrt{L_{\text{nys}}}$
return $\hat{w} = w_{\text{swa}}$, P

B. INKPCA BASED ON RANK ONE UPDATES AND NYSTRÖM APPROXIMATION

Modern deep learning models are usually high-dimensional parameter space, there are two main problems we should handle in Bayesian inference:

1) how to extract more valuable non-linear information from high-dimensional parameter space;

2) how to effectively handle a large number of parameters obtained after multiple epochs training.

For the first problem, kernel method is a practical way to handle non-linear information. For the second problem, we can reduce the number of parameters and find several representative parameters to stand for the entire parameter space, such as PCA subspace method and so on. To further improve the memory efficiency, there is usually a Nyström approximation for incremental calculation in computing huge kernel matrix [15]. This is because Nyström approximation can efficiently evaluate and determine the suitable size of subsets in the original datasets and resulting in further gains in memory efficiency.

Although the kernel method can extract non-linear information from high-dimensional space, it increases the difficulty of processing the kernelized high-dimensional parameter space. The incremental method of KPCA was firstly introduced for singular value decomposition to update an eigenfeature space, this method is time consuming when facing high-dimensional data [12], [13] . A novel based on ROU algorithm for InKPCA is proposed to efficiently compute the eigendecomposition of the kernel matrix [14].

The accuracy of Nyström approximation has been verified through extensive experiments and research, including comparison with other approaches such as random Fourier features, symmetric positive semi-definite Laplacian and kernel matrices [30].

A simple incremental variant of Nyström kernel regularized least squares controls the regularization and computation at the same time, the Nyström approximation approach uses Cholesky ROU formulas for kernel ridge regression [31]. Due to the main problem of kernel based predictors, such as Gaussian processes and Support Vector Machines, is that they are costly to find the solution scales as $O(n^3)$ [32].

However, Algorithm 3 is more generalized than their method. Nyström computational regularization method has been proven capable of achieving optimal bounds in the large scale statistical learning setting and earned much better time complexity on kernel classification and kernel ridge regression.

In this paper, we suppose that the parameters have zero mean in feature space, so the mean does not need to be updated.

We adopt a trick to ensure that the currently processed data is the latest deviation vector, so we need to keep the arriving data in strict order (SO). Assuming $X_{m+1}^{SO} = \{x_1, x_2, \ldots, x_m, x_{m+1}\}$, where $x_1 \prec x_2 \cdots \prec x_m \prec x_{m+1}$, and x_{m+1} is the latest data to be processed. Where $k_{i,j} = k(x_i, x_j)$ denotes the value of radial basis kernel function(RBF) between data x_i and x_j, $K_{m,m}^{SO}$ represents for kernel matrix of the first m samples of X_m^{SO}.

Our purpose is to expand an additional column and row of kernel matrix $K_{m,m}^{SO}$ to $K_{m+1,m+1}^{SO}$,we follow [14]:

$$v_1 = [\, A^{\text{T}} \quad \frac{1}{2} k_{m+1,m+1} \,]^{\text{T}}$$

$$v_2 = [\,A^{\mathrm{T}} \quad \frac{1}{4}k_{m+1,m+1}\,]^{\mathrm{T}}$$
$$\sigma = 4/k_{m+1,m+1}$$

So the identity equation is:

$$K^{SO}_{m+1,m+1} = \begin{bmatrix} K^{SO}_{m,m} & A \\ A^T & k_{m+1,m+1} \end{bmatrix} = \begin{bmatrix} K^{SO}_{m,m} & \mathbf{0}_m \\ \mathbf{0}_m^T & \frac{1}{4}k_{m+1,m+1} \end{bmatrix} + \sigma v_1 v_1^T - \sigma v_2 v_2^T \tag{7}$$

where $A = [k_{1,m+1}k_{2,m+1}\cdots k_{m,m+1}]^T$, $\mathbf{0}_m$ is a column vector of zeros, $v_1 v_1^T$ is a Gram matrix, $K^{SO}_{m+1,m+1}$ is a symmetric positive definite matrix.

In Algorithm 3, we devise an improved ROU method with strictly ordered dataset in two steps, without adjusting the mean. And the improved ROU algorithm is not reshuffled the order of the deviation vectors to ensure that the current deviation vector is the object to be processed. The function ROU(σ, v, L^{SO}, U^{SO}) updates the eigenvalues L^{SO} and eigenvectors U^{SO} with perturbation σvv^{T}.

Assuming the kernel matrix $K_{m,m}$ has the eigenvalues L^{SO} and eigenvectors U^{SO}, the corresponding Nyström approximation function $NYS(L^{SO}, U^{SO})$ produces approximate eigenvalues L_{nys} and eigenvectors U_{nys} described in (8) as follows:

$$L_{\mathrm{nys}} := \frac{n}{m}L^{SO}, \quad U_{\mathrm{nys}} := \sqrt{\frac{m}{n}}K_{n,m}U^{SO}(L^{SO})^{-1} \tag{8}$$

In Algorithm 3, each iteration adds an extra column of kernel values to $K_{n,m}$ corresponding to the new vector, and then calculates the rescaling (8), we finally achieve the approximation of kernel matrix $\tilde{K} = U_{\mathrm{nys}}L_{\mathrm{nys}}U_{\mathrm{nys}}^{\mathrm{T}}$. The time complexity in (8) is $O(m^2 n)$.

Algorithm 3 Incremental Eigendecomposition of Kernel Matrix With Nyström Approximation (InKPCA)

Input: vectors $X^{SO}_{m+1} = \{x_1, x_2, ...x_{\mathrm{m}}, x_{\mathrm{m+1}}\}$; row vector of eigenvalues L^{SO} and matrix of eigenvectors U^{SO} of $K^{SO}_{m,m}$; kernel function $k(\bullet, \bullet)$.
Output: L_{nys}, U_{nys} of $K^{SO}_{m+1,m+1}$

1. $L^{SO} \leftarrow [L^{SO}k_{m+1,m+1}/4]$
2. $U^{SO} \leftarrow \begin{bmatrix} U^{SO} & \mathbf{0}_m \\ \mathbf{0}_m^T & k_{m+1,m+1}/4 \end{bmatrix}$
3. $\sigma \leftarrow 4/k_{m+1,m+1}$
4. $v_1 \leftarrow [\,k_{1,m+1}k_{2,m+1}\cdots k_{m+1,m+1}/2\,]$
5. $v_2 \leftarrow [\,k_{1,m+1}k_{2,m+1}\cdots k_{m+1,m+1}/4\,]$
6. $L^{SO}, U^{SO} \leftarrow ROU(\sigma, v_1, L^{SO}, U^{SO})$
7. $L^{SO}, U^{SO} \leftarrow ROU(-\sigma, v_2, L^{SO}, U^{SO})$
8. $L_{\mathrm{nys}}, U_{\mathrm{nys}} \leftarrow NYS(L^{SO}, U^{SO})$

The selection of the kernel function is required to satisfy Mercer's theorem, that is the Gram matrix of the kernel function in the data space is a semi-positive definite matrix. In addition to RBF, the other commonly used kernel functions include linear kernel function, polynomial kernel function, Matérn kernel, exponential kernel and so on.

C. IMPLEMENTING APPROXIMATE INFERENCE METHODS WITHIN INKPCA SUBSPACE

The section III-A and III-B introduce that InKPCA subspace comes from SGD trajectories and we can apply ROU algorithm and Nyström approximation to effectively extract more representative subspaces. In this section we will present that how to implement approximate inference methods VI and ESS within the constructed InKPCA subspace.

For implementing Bayesian inference approach VI within the InKPCA subspace, several steps should be followed.

First of all, we should achieve the parameter subspace S of DNNs model, which is constructed with the fixed $\hat{w}$ and projection matrix P based on Algorithm 2. The weight subspace S of DNNs model can be reconstructed as follows:

$$S = \hat{w} + P^{\mathrm{T}} * \theta \tag{9}$$

where P^{T} represents the transposition of P and θ denotes the projection parameter, which is obtained by sampling from variational posterior distribution $q(\theta|\mu_q, \sigma_q) = N(\mu_q, \sigma_q^2 I_r)$, and r signifies the rank of InKPCA approximation. A reasonable choice of prior $p(\theta|\mu_p, \sigma_p)$ is $N(0,\sigma_p^2 I_r)$ [16], and σ_p is prior standard deviation. Due to the shift parameter $\hat{w}$ in constructing the InKPCA subspace S in (9), the prior $p(\theta|\mu_p, \sigma_p)$ will revolve around a set of good solutions [6].

Secondly, when Bayesian inference approach VI is performed within InKPCA subspace, we need to consider an empirical cost function negative ELBO in (5), which consists of two parts: KL divergence and negative log-likelihood (NLL).

For item KL divergence in negative ELBO, we apply the most common and fully factorized Gaussians for the prior and variational posterior distributions. So we only need to calculate the KL divergence between $q(\theta|\mu_q, \sigma_q)$ and $p(\theta|\mu_p, \sigma_p)$ in one dimension, which is described as follows:

$$\begin{aligned} &KL\left[q(\theta|\mu_q, \sigma_q) \parallel p(\theta|\mu_p, \sigma_p)\right] \\ &= KL\left[N(\mu_q, \sigma_q^2) \parallel N(0, \sigma_p^2)\right] \\ &= \frac{1}{n}\sum_{i=1}^{n}(\log\frac{\sigma_{p,i}}{\sigma_{q,i}} + \frac{\sigma_{q,i}^2}{2\sigma_{p,i}^2} + \frac{\mu_{q,i}^2}{2\sigma_{p,i}^2} - \frac{1}{2}) \end{aligned} \tag{10}$$

where n denotes the number of samples.

For item NLL in negative ELBO, θ can be sampled from $N(\mu_q, \sigma_q^2 I_r)$, and then θ is assigned to (9) to achieve the parameter weight of DNNs model, and finally $\mathrm{NLL} = -\mathrm{E}_{q(\theta|\mu_q,\sigma_q)}\left[\log p\,(D|\theta)\right]$ can be obtained.

At last, VI model with cost function negative ELBO can be optimized as the same as DNNs models. The difference between VI model and DNNs model is that VI model is an optimized DNNs model, and the optimization objects of VI model are mean μ_q and standard deviation σ_q of variational distribution $q(\theta|\mu_q, \sigma_q)$ for Bayesian inference, which

TABLE 1. Results of RMSE for inference methods on UCI regression datasets.

Dataset	N	D	SGD	SWAG	PCA+ESS	PCA+VI	InKPCA+ESS	InKPCA+VI
boston	506	13	3.986 ± 0.975	3.517 ± 0.981	3.453 ± 0.953	3.457 ± 0.951	**3.241** ± 0.021	3.242 ± 0.011
concrete	1030	8	5.194 ± 0.446	5.233 ± 0.417	5.194 ± 0.448	5.142 ± 0.418	5.438 ± 0.240	**4.980** ± 0.066
energy	768	8	1.602 ± 0.275	1.594 ± 0.273	1.598 ± 0.274	**1.587** ± 0.272	1.793 ± 0.038	1.780 ± 0.050
naval	11934	16	**0.001** ± 0.000	**0.001** ± 0.000	**0.001** ± 0.000	**0.001** ± 0.000	**0.001** ± 0.000	**0.001** ± 0.000
yacht	308	6	0.973 ± 0.374	0.973 ± 0.375	**0.972** ± 0.375	0.973 ± 0.375	1.117 ± 0.135	1.232 ± 0.020
elevators	16599	18	0.103 ± 0.035	0.088 ± 0.001	0.089 ± 0.002	**0.088** ± 0.001	0.092 ± 0.048	0.093 ± 0.005
keggD	48827	20	0.132 ± 0.017	0.129 ± 0.029	0.129 ± 0.028	0.128 ± 0.028	**0.128** ± 0.004	0.171 ± 0.263
keggU	63608	27	0.186 ± 0.034	0.160 ± 0.043	0.160 ± 0.043	0.160 ± 0.043	**0.149** ± 0.045	**0.149** ± 0.045
protein	45730	9	0.436 ± 0.011	0.415 ± 0.018	0.425 ± 0.017	0.418 ± 0.021	**0.414** ± 0.025	0.483 ± 0.077
skillcraft	3338	19	0.288 ± 0.014	0.293 ± 0.015	0.293 ± 0.015	0.293 ± 0.015	0.454 ± 0.157	**0.259** ± 0.005
pol	15000	26	3.900 ± 6.003	3.110 ± 0.070	3.755 ± 6.107	2.499 ± 0.684	**2.121** ± 0.032	2.760 ± 1.217

can be used for uncertainty estimates of DNNs model. However, original DNNs model is trained by optimizing its weight parameters, which can be considered as a deterministic optimization process.

As mentioned above, in order to sample a reasonable θ in VI, assuming $\theta \sim q(\theta|\mu_q, \sigma_q) = N(\mu_q, \sigma_q^2 I_r)$. For stochastic approximation method ESS, however, which is an easy and efficient tool to obtain the sample θ.

The only inputs required by ESS method are initial state θ_0, a routine prior that can sample from $N(0, \sigma_p^2 I_{r-1})$, and log-likelihood function $\log L$. And the outputs of ESS method are to produce an updated θ and $\log L$. However, function $\log L$ can be described by negative loss function of DNNs model, whose weight parameters in (9) are generated by θ.

IV. EXPERIMENTS

In this section we present the approximate Bayesian inference methods ESS and VI within the proposed InKPCA subspace. The experiments in this paper are based on a single Nvidia GTX 1080Ti GPU and PyTorch deep learning platform.

In the section IV-A, the results show that the proposed InKPCA subspace with Nyström approximation can produce good and variance sensitive predictive uncertainties on regression tasks. In the section IV-B, we conduct a quantitative evaluation on 11 UCI regression datasets, which include 6 large datasets keggdirected, keggundirected, elevators, skillcraft, protein, pol, and 5 small datasets boston, concrete, energy, naval and yacht. The number N and dimension D of these UCI datasets are described in Table 1.

In the section IV-C, we apply the proposed InKPCA subspace inference method to large-scale image classification on CIFAR-10 and CIFAR-100 datasets. The CIFAR-10 dataset consists of 60,000 32×32 color images in 10 classes, each class has 6,000 images, including 50,000 training images and 10,000 test images. The CIFAR-100 dataset has 100 classes, each class has 600 color images with a size of 32×32, of which 500 images are used as the training set and 100 images are used as the test set. The results demonstrate that Bayesian inference method ESS or VI in the proposed InKPCA subspace outperforms the other advanced Bayesian inference in DNNs.

A. VISUALIZATION REGRESSION UNCERTAINTY

Our purpose is to show that when we move away from the data, the predicted uncertainty should increase, because there are many possible functions can fit the data. However, improving the accuracy of the model is equally important.

The experiment settings follows the settings of literature [6]. That the fully-connected architecture has [200,50,50,50] neurons in hidden layers respectively. At the beginning of the architecture has two inputs, x and x^2, the last layer of the architecture has a single real value output $y = f(x)$. The 400 training data points generated by the defined architecture with random weights, and the training points are uniformly sampled in intervals $[-7.2, -4.8]$, $[-1.2, 1.2]$, $[4.8, 7.2]$. Gaussian noise is added to the outputs $y = f(x) + \epsilon(x)$, which are trained with red circles in Fig.2.

We train the above defined neural network and achieve a SWA solution [26], and construct three subspaces: a 100-dimensional InKPCA subspace, 100-dimensional PCA subspace and a 2-dimensional curve subspace. Then we implement VI and ESS inference methods in these representative subspaces. For VI model, we set the standard deviation $\sigma_p = 5$ of prior $p(\theta|0, \sigma_p)$ and initialize the standard deviation $\sigma_q = 1$ of variational distribution $q(\theta|0, \sigma_q)$. For ESS model, we set the standard deviation $\sigma_p = 5$ of the routine prior $N(0, \sigma_p^2 I_{r-1})$.The predictive distribution and the uncertainty in regression are showed in Fig.2, where red circles represent for data, predictive mean is expressed by dark blue line and shaded region stands for ± 3 standard deviations of predictive mean.

In the top row of Fig.2, we view the predictive distributions for VI approach applied in each of 3 subspaces. Compared with PCA subspace and Curve subspace, InKPCA subspace captures more concentrated and accurate predictive means with dark blue line models. At the same time, when models are far away from points in the second interval $[-1.2, 1.2]$, (a) has gained more predictive uncertainty than (b) and (c). At the beginning of points in third interval $[4.8, 7.2]$, there is a convex which is explicitly showed in (a), which may denote the predictive uncertainty of InKPCA subspace is more sensitive than (b) and (c) between data and no data. At the end of the third interval, the predictive uncertainty of PCA subspace and Curve subspace expand faster than InKPCA subspace.

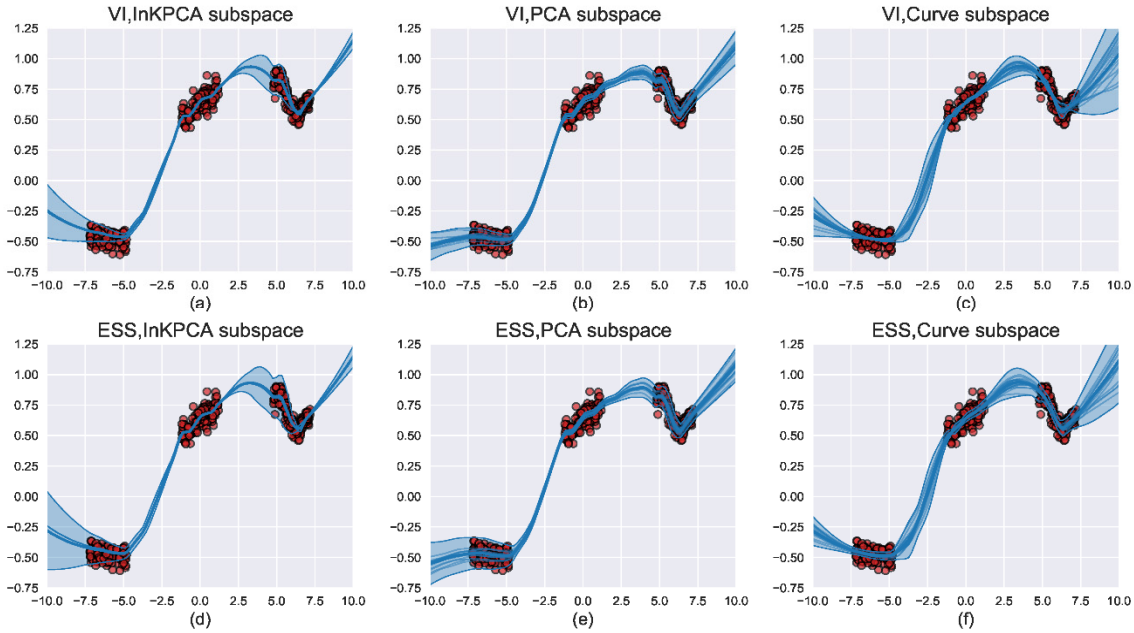

FIGURE 2. Visualizing uncertainty of predictive distribution in regression. red circles for Data, dark blue line for predictive mean, light blue lines for sampled posterior functions, shaded region for ±3 standard deviations about the mean.

For ESS method employed in the 3 subspaces, we can achieve the same conclusion in (d),(e) and (f) of Fig.2. In additional experiments, when changing a small variance value, InKPCA subspace is more sensitive to predictive uncertainty than PCA subspace and Curve subspace.

However, by comparing the Bayesian inference methods VI and ESS, we find that ESS obtains more predictive uncertainties than VI, the reason is that the VI method underestimates the uncertainty of the model [33].

B. UCI REGRESSION

We next compare InKPCA-based subspace inference methods (i.e. ESS and VI) on UCI large and small regression datasets with other approximate Bayesian inference approaches. And for VI model, we set the standard deviation $\sigma_p = 1$ of prior $p(\theta|0,\sigma_p)$ and initialize the standard deviation $\sigma_q = 3$ of variational distribution $q(\theta|0,\sigma_q)$. For ESS model, we set the standard deviation $\sigma_p = 1$ of the routine prior $N(0,\sigma_p^2 I_{r-1})$.

For the 6 large UCI regression datasets, we follow the experimental scheme in literature [34]. That is for the number of training examples N $>$ 6000, we apply a fully-connected DNNs with a [D-1000-1000-500-50-2] architecture, where D represents for the dimension of the datasets. And ReLU is chosen as the activation function and two outputs include predictive mean $\mu(x, w)$ and predictive variance $\sigma^2(x, w, s)$.The variance at x is $\sigma^2(x, w, s) = s^2 + \sigma_w^2(x)$, where $\sigma_w^2(x)$ is the variance and comes from the last layer of network, and s^2 is the learned global noise variance. The learning rate is 10^{-3},batch size is 400 and the training epochs are 200. For N $\leq$ 6000, we employ a fully-connected DNNs with a [D-1000-500-50-2] architecture, and only 100 epochs are needed to train it and the learning rate is 5×10^{-4}. The learning rate is 10^{-4} for keggD dataset. In subspace we employ the Normal distribution prior and the variance is 1.0. In order to ensure the variance s and σ are positive, we apply softplus parameterizations as a trick.

For the 5 small UCI regression datasets containing boston, concrete, energy, naval and yacht, we adhere to literature [35] and apply a fully-connected network, which has a single hidden layer with 50 units. And we use neural network to output a variance as heteroscedastic uncertainty. The learning rate and weight decay are manually tuned and the batch size is N/10,where N is the size of dataset. We employ neural networks with a Bayesian final layer based on SGD training in literature [36], and we compare InKPCA subspace inference methods (i.e. ESS and VI) with other approaches, such as PCA subspace inference methods, SGD, and SWAG in Literature [6].

In this section, we mainly measure the quality of prediction results of UCI regression tasks through RMSE and 95% prediction confidence interval. In testing phase, when the test data X_{i} is input to the trained neural network, we get two outputs: predictive mean μ_{i} and predictive variance σ_{i}^2. And assuming Y_{i} is the target. One of the metrics is $\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{\mathrm{i}=1}^{\mathrm{i}=n}(Y_{\mathrm{i}} - \mu_{\mathrm{i}})^2}$.

The other metric is 95% prediction confidence interval, which means that 95% of the sample means $\mu = \{\mu_1, \cdots, \mu_{\mathrm{n}}\}$ will fall within 2 standard error range, the mathematical formula is $P(\mu - 1.96\frac{\sigma}{\sqrt{n}} < Y < \mu + 1.96\frac{\sigma}{\sqrt{n}}) = 0.95$. Our goal is to figure out if the predictive mean μ_{i} is in this interval, and the

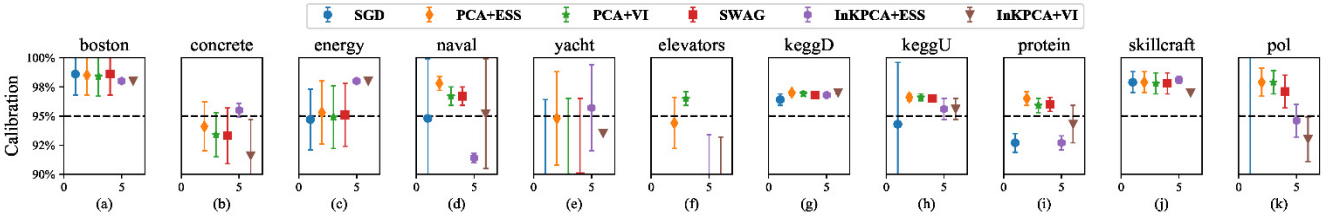

FIGURE 3. Coverage of 95% prediction interval on UCI regression dataset. In most cases, InKPCA subspace inference produces closer to 95% coverage than models trained using PCA subspace inference or SGD or SWAG.

TABLE 2. Result of accuracy for inference on cifar-10 and cifar-100 classification datasets.

Dataset	Model	SWAG	SWA	KFAC-Laplace	PCA+VI	PCA+ESS	InKPCA+VI	InKPCA+ESS
CIFAR-10	VGG-16	93.60 ± 0.10	93.61 ± 0.11	92.65 ± 0.20	93.61 ± 0.02	93.66 ± 0.08	93.61 ± 0.01	**93.66** ± 0.04
CIFAR-10	PRN-164	96.03 ± 0.02	96.09 ± 0.08	95.49 ± 0.06	95.96 ± 0.13	95.98 ± 0.09	**96.13** ± 0.00	96.07 ± 0.00
CIFAR-10	WRN28x10	96.32 ± 0.03	96.46 ± 0.04	96.17 ± 0.00	96.38 ± 0.05	96.32 ± 0.08	96.45 ± 0.02	**96.46** ± 0.01
CIFAR-100	VGG-16	74.77 ± 0.09	74.30 ± 0.22	72.38 ± 0.23	74.83 ± 0.08	74.62 ± 0.37	**74.84** ± 0.09	74.63 ± 0.23
CIFAR-100	PRN-164	79.90 ± 0.50	80.19 ± 0.52	78.51 ± 0.05	80.52 ± 0.18	80.54 ± 0.13	**80.62** ± 0.00	80.59 ± 0.00
CIFAR-100	WRN28x10	82.23 ± 0.19	82.40 ± 0.16	80.94 ± 0.41	82.63 ± 0.26	82.49 ± 0.23	**82.63** ± 0.22	82.47 ± 0.20

discriminant formula is $calib_i = \{(Y_i - \mu_i) < 1.96 * \sigma_i\} * \{(Y_i - \mu_i) > -1.96 * \sigma_i\}$,which means that if μ_i is in the interval, then $calib_i$ is *True*; else is *False*. So n predictive means produce n bool values to form an array $calib = array(calib_1, \cdots, calib_n)$, and then calculate the average value by the number of *True* values. The average value is the desired $calibration = \frac{\text{number of } True \text{ values}}{n}$, and then the mean and variance of *calibration* is obtained through 3 independent experiments with different seed.

Different datasets are well calibrated through the 95% confidence interval, the coverage of the prediction interval closer to 95% indicates that the better performance of the model. We plot the coverage of 95% prediction intervals on UCI regression datasets in Fig.3. In most cases, ESS or VI in InKPCA subspace produces the closest results to 95% coverage. For example, ESS in InKPCA subspace surpasses SGD, SWAG, and PCA subspace in (a), (b), (h), and (k), VI in InKPCA subspace outperforms SGD, SWAG, and PCA subspace in (i) and (j). Meanwhile, InKPCA-based subspace inference method VI is comparable to SGD method in (d), and exceeding PCA in (d).

However, PCA-based subspace inference methods generate better results, which are closer to 95% than InKPCA subspace in (c) and (f). Such as ESS in PCA subspace produces better results than the other approaches in (e). The conclusion is similar to RMSE in Table 1, and the Bayesian inference may be related to the dimensional information of the datasets, which provides a direction for future research.

And we summarize the RMSE in Table 1 and find that Bayesian inference in InKPCA subspace outperforms PCA subspace in boston, concrete, keggD, keggU, protein, skillcraft and pol datasets in RMSE (the lower is the better), and is comparable to SGD, SWAG and PCA in naval datasets. However, PCA subspace inference method achieves better RMSE results than InKPCA subspace in energy, ychat, and elevators datasets.

C. IMAGE CLASSIFICATION

In this section, we test the Bayesian inference approaches ESS and VI in InKPCA subspace on different advanced convolutional neural networks (CNNs) models and benchmark datasets. And for VI model, we set the standard deviation $\sigma_p = 1$ of prior $p(\theta|0,\sigma_p)$ and initialize the standard deviation $\sigma_q = 1$ of variational distribution $q(\theta|0,\sigma_q)$. For ESS model, we set the standard deviation $\sigma_p = 1$ of the routine prior $N(0,\sigma_p^2 I_{r-1})$.We follow the experimental framework in literature [26] and apply $T = 5000$ for temperature in the image classification experiments.

We report the mean and standard deviation over 3 independent runs of the inference methods ESS and VI in InKPCA subspace on high-performance CNNs models, such as VGG-16, PreResNet-164 (PRN-164), and WideResNet 28 × 10 (WRN 28 × 10), and the datasets are CIFAR-10, and CIFAR-100. The results show that the InKPCA-based subspace inference methods outperform PCA-based subspace inference approaches and other baselines including SGD and SWAG. The results are presented in Table 2 and bolded numbers represent the best result compared to other methods.

V. CONCLUSION

Although Bayesian approaches can provide probabilistic uncertainty estimates for machine learning models. Bayesian methods are inefficient in deep learning domain with high-dimensional parameter space. To overcome this challenge, the high-performance PCA-based subspace inference method has been proposed and produced the state-of-the-art practical results when comparing with other methods, such as Curve subspace, SGD, SWAG and so on [6].

However, PCA method is a linear subspace feature extractor and cannot extract nonlinear features from the high-dimensional parameter space of DNNs. In this paper, we propose to apply KPCA to extract higher-order statistical information of parameter subspace within SGD trajectories.

In order to improve the computational efficiency of the kernel matrix, we present an improved InKPCA, which applies rank one updates algorithm to carry out the strictly ordered incremental eigendecomposition of kernel matrix, and employ Nyström approximation to determine the suitable size of subsets to further improve the memory efficiency.

Through extensive experiments, the experimental results show that in most datasets, the inference method VI or ESS in the constructed InKPCA subspace not only has higher accuracy, but also is more effective and variance-sensitive to express uncertainty.

Importantly, the proposed InKPCA subspace is flexible to explore different subspaces and approximate inference approaches. And there are many promising directions for future research. For example, the construction of InKPCA subspace can be based on Fast Geometric Ensembling (FGE) [37], which can connect the optima of loss functions by simple curves. One could apply Bayesian theory to InKPCA for automatically choosing the dimensionality of subspace [38]. InKPCA subspace inference approach can also be applied to the problems of input dimensionality in Bayesian optimization and probabilistic model-based reinforcement learning. In addition to the RBF kernel function selected in this paper, one could try other kernel functions and devise algorithms to find the optimal kernel parameters. Due to kernel method is associated with every training vector and not sparse, one could reduce the number of example vectors by approximating the covariance matrix in feature space [39].

As an efficient nonlinear subspace feature extractor, the proposed InKPCA subspace with Nyström approximation is a low-dimensional, scalable and interpretable approach for Bayesian deep learning.

REFERENCES

[1] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. ICML*, vol. 70, 2017, pp. 1321–1330,
[2] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" in *Proc. NIPS*, Long Beach, CA, USA, Dec. 2017, pp. 5580–5590.
[3] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *Proc. ICML*, vol. 37, 2015, pp. 1613–1622.
[4] C. MacKay and J. David, *Information Theory, Inference and Learning Algorithms*. New York, NY, USA: Cambridge Univ. Press, 2002. [Online]. Available: https://www.cis.hut.fi/Opinnot/T-61.182/2004/slides/ch1.pdf
[5] D. P. Kingma, T. Salimans, and M. Welling, "Variational dropout and the local reparameterization trick," in *Proc. NIPS*, vol. 2, Dec. 2015, pp. 2575–2583.
[6] P. Izmailov, W. J. Maddox, P. Kirichenko, T. Garipov, D. Vetrov, and A. G. Wilson, "Subspace inference for Bayesian deep learning," in *Proc. ICML*, 2019, pp. 1169–1179.
[7] A. M. Martinez and A. C. Kak, "PCA versus LDA," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 2, pp. 228–233, Feb. 2002.
[8] R. A. Fisher, "The statistical utilization of multiple measurements," *Ann. Eugenics*, vol. 8, no. 4, pp. 376–386, Aug. 1938.
[9] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics Intell. Lab. Syst.*, vol. 2, no. 4, pp. 37–52, 1986.
[10] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, Jul. 1998.
[11] K. I. Kim, M. O. Franz, and B. Scholkopf, "Iterative kernel principal component analysis for image modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 9, pp. 1351–1366, Sep. 2005.
[12] T. Chin and D. Suter, "Incremental kernel PCA for efficient non-linear feature extraction," in *Proc. BMVC*, Edinburgh, Scotland, Sep. 2006, pp. 4–7.
[13] T.-J. Chin and D. Suter, "Incremental kernel principal component analysis," *IEEE Trans. Image Process.*, vol. 16, no. 6, pp. 1662–1674, Jun. 2007.
[14] F. Hallgren and P. Northrop, "Incremental kernel PCA and the Nyström method," 2018, *arXiv:1802.00043*. [Online]. Available: https://arxiv.org/abs/1802.00043
[15] C. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," in *Proc. NIPS*, Denver, CO, USA, 2001, pp. 682–688.
[16] W. Maddox, T. Garipov, P. Izmailov, D. Vetrov, and A. G. Wilson, "A simple baseline for Bayesian uncertainty in deep learning," in *Proc. NIPS*, Vancouver, BC, Canada, Dec. 2019, pp. 13132–13143.
[17] B. Léon, "Stochastic gradient learning in neural networks," *Proc. Neuro-Nimes*, vol. 91, no. 8, p. 12, Jan. 1991.
[18] R. M. Neal, "Bayesian learning for neural networks," *IEEE Trans. Neural Netw.*, vol. 8, no. 2, p. 456, Mar. 1997.
[19] T. Chen, E. B. Fox, and C. Guestrin, "Stochastic gradient Hamiltonian Monte Carlo," in *Proc. ICML*, Beijing, China, vol. 70, 2014, pp. 1321–1330.
[20] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient Langevin dynamics," in *Proc. ICML*, Bellevue, WA, USA, Jun. 2011, pp. 681–688.
[21] S. Mandt, M. D. Hoffman, and D. M. Blei, "Stochastic gradient descent as approximate Bayesian inference," *J. Mach. Learn. Res.*, vol. 18, pp. 1–35, Apr. 2017.
[22] I. Murray, P. R. Adams, and D. J. MacKay, "Elliptical slice sampling," *J. Mach. Learn. Res.*, vol. 9, pp. 541–548, Mar. 2010.
[23] A. Graves, "Practical variational inference for neural networks," in *Proc. 24th Int. Conf. Neural Inf. Process. Syst.*, Dec. 2011, pp. 2348–2356.
[24] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. ICLR*, Dec. 2014, pp. 1–14.
[25] C. Louizos and M. Welling, "Multiplicative normalizing flows for variational Bayesian neural networks," in *Proc. ICML*, vol. 70, Aug. 2017, pp. 2218–2227.
[26] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson, "Averaging weights leads to wider optima and better generalization," in *Proc. UAI*, Monterey, CA, USA, Aug. 2018, pp. 876–885.
[27] T. Garipov, P. Izmailov, D. Podoprikhin, D. P. Vetrov, and A. G. Wilson, "Loss surfaces, mode connectivity, and fast ensembling of DNNs," in *Proc. NIPS*, Red Hook, NY, USA, vol. 31, 2018, pp. 8789–8798.
[28] H. Ritter, A. Botev, and D. Barber, "Online structured Laplace approximations for overcoming catastrophic forgetting," in *Proc. NIPS*, Red Hook, NY, USA, vol. 31, 2018, pp. 3738–3748.
[29] N. Halko, P. G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM Rev.*, vol. 53, no. 2, pp. 217–288, Jan. 2011.
[30] T. Yang, Y.-F. Li, M. Mahdavi, R. Jin, and Z.-H. Zhou, "Method vs random Fourier features: A theoretical and empirical comparison," in *Proc. NIPS*, 2012, pp. 476–484.
[31] W. Li, M. Liu, and D. Zhang, "Subspace learning for large-scale SVMs," 2020, *arXiv:2002.08937*. [Online]. Available: https://arxiv.org/abs/2002.08937
[32] A. Rudi, R. Camoriano, and L. Rosasco, "Less is more: Computational regularization," in *Proc. NIPS*, 2015, pp. 1657–1665.
[33] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *J. Amer. Stat. Assoc.*, vol. 112, no. 518, pp. 859–877, Apr. 2017.
[34] A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing, "Deep kernel learning," in *Proc. ICML*, Cadiz, Spain, vol. 51, May 2016, pp. 370–378.
[35] T. D. Bui, "Deep Gaussian processes for regression using approximate expectation propagation," in *Proc. ICML*, New York, NY, USA, vol. 48, Jun. 2016, pp. 1472–1481.
[36] C. Riquelme, G. Tucker, and J. Snoek, "Deep Bayesian bandits showdown," in *Proc. ICLR*, 2017, pp. 1–27.
[37] T. Garipov, P. Izmailov, D. Podoprikhin, D. P. Vetrov, and A. G. Wilson, "Loss surfaces, mode connectivity, and fast ensembling of DNNs," in *Proc. NIPS*, Red Hook, NY, USA 2018, pp. 8803–8812.
[38] T. P. Minka, "Automatic choice of dimensionality for PCA," in *Proc. NIPS*, Denver, CO, USA, 2000, pp. 598–604.
[39] M. E. Tipping and C. C. Nh, "Sparse kernel principal component analysis," in *Proc. NIPS*, Denver, CO, USA, 2001, pp. 633–639.

YONGGUANG WANG received the B.S. degree from the Hubei University of Medicine, China, in 2011, and the M.S. degree from Beijing Jiaotong University, China, in 2015. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Beihang University. His research interests include deep learning, uncertainty quantification, and software safety.

SHUZHEN YAO received the B.S., M.S., and Ph.D. degrees from Beihang University, in 1986, 1989, and 2008, respectively. She was a Visitor Scholar with the University of Illinois at Chicago, from 2005 to 2006. She is currently a Professor with the School of Computer Science and Engineering, Beihang University. Her research interests include Petri net theory and software safety.

TIAN XU received the B.S. degree from Wuhan University, China, in 2018. He is currently pursuing the master's degree with the School of Computer Science and Engineering, Beihang University. His main research interests include software safety and Petri net application.