Supplementary Materials for “Kernel Continual Learning”

Mohammad Mahdi Derakhshani [1] **Xiantong Zhen** [1 2] **Ling Shao** [2] **Cees G. M. Snoek** [1]

In Section A, we provide the detailed derivation of the Evidence Lower Bound (ELBO) of our variational random features for kernel continual learning. In Section B, we illustrate our proposed model in detail, explaining each part. Moreover, for all kernels, e.g., Linear, Polynomial, and Radial Basis Function, we discuss how the main architecture is changed accordingly. In Section C, all hyperparamters are listed in a table in order to reproduce the paper results. Finally, in Section D, we include additional ablation results on miniImageNet for the size of inference memory and number of Random Bases.

A. Derivation of Evidence Lower Bound

Our proposed objective in equation 10 is derived as follows:

$$\begin{aligned}\ln p(\mathbf{y}|\mathbf{x},\mathcal{D}_t\backslash\mathcal{C}_t) &= \ln\Big[\int p(\mathbf{y}|\mathbf{x},\boldsymbol{\omega},\mathcal{D}_t\backslash\mathcal{C}_t)p(\boldsymbol{\omega}|\mathcal{D}_t\backslash\mathcal{C}_t)d\boldsymbol{\omega}\Big]\\ &= \ln\Big[\int p(\mathbf{y}|\mathbf{x},\boldsymbol{\omega},\mathcal{D}_t\backslash\mathcal{C}_t)p(\boldsymbol{\omega}|\mathcal{D}_t\backslash\mathcal{C}_t)\frac{q_\phi(\boldsymbol{\omega}|\mathcal{C}_t)}{q_\phi(\boldsymbol{\omega}|\mathcal{C}_t)}d\boldsymbol{\omega}\Big]\\ &= \ln\Big[\mathbb{E}_{q_\phi(\boldsymbol{\omega}|\mathcal{C}_t)}\frac{p(\mathbf{y}|\mathbf{x},\boldsymbol{\omega},\mathcal{D}_t\backslash\mathcal{C}_t)p(\boldsymbol{\omega}|\mathcal{D}_t\backslash\mathcal{C}_t)}{q_\phi(\boldsymbol{\omega}|\mathcal{C}_t)}\Big].\end{aligned}\tag{1}$$

By applying Jensen’s inequality, we have

$$\begin{aligned}\ln p(\mathbf{y}|\mathbf{x},\mathcal{D}_t\backslash\mathcal{C}_t) &\geq \mathbb{E}_{q_\phi(\boldsymbol{\omega}|\mathcal{C}_t)}\Big[\ln\frac{p(\mathbf{y}|\mathbf{x},\boldsymbol{\omega},\mathcal{D}_t\backslash\mathcal{C}_t)p(\boldsymbol{\omega}|\mathcal{D}_t\backslash\mathcal{C}_t)}{q_\phi(\boldsymbol{\omega}|\mathcal{C}_t)}\Big]\\ &= \mathbb{E}_{q_\phi(\boldsymbol{\omega}|\mathcal{C}_t)}\Big[\ln p(\mathbf{y}|\mathbf{x},\boldsymbol{\omega},\mathcal{D}_t\backslash\mathcal{C}_t)+\ln\frac{p(\boldsymbol{\omega}|\mathcal{D}_t\backslash\mathcal{C}_t)}{q_\phi(\boldsymbol{\omega}|\mathcal{C}_t)}\Big]\\ &= \mathbb{E}_{q_\phi(\boldsymbol{\omega}|\mathcal{C}_t)}\Big[\ln p(\mathbf{y}|\mathbf{x},\boldsymbol{\omega},\mathcal{D}_t\backslash\mathcal{C}_t)\Big]+\mathbb{E}_{q_\phi(\boldsymbol{\omega}|\mathcal{C}_t)}\Big[\ln\frac{p(\boldsymbol{\omega}|\mathcal{D}_t\backslash\mathcal{C}_t)}{q_\phi(\boldsymbol{\omega}|\mathcal{C}_t)}\Big]\\ &= \mathbb{E}_{q_\phi(\boldsymbol{\omega}|\mathcal{C}_t)}\Big[\ln p(\mathbf{y}|\mathbf{x},\boldsymbol{\omega},\mathcal{D}_t\backslash\mathcal{C}_t)\Big]-D_{\mathrm{KL}}\big[q_\phi(\boldsymbol{\omega}|\mathcal{C}_t)\|p_\gamma(\boldsymbol{\omega}|\mathcal{D}_t\backslash\mathcal{C}_t)\big]\\ &= \mathcal{L}_{\mathrm{ELBO}}.\end{aligned}\tag{2}$$

B. Model Details

We provide the computational graph of our kernel continual learning with variational random features in Figure B.1. Our method consists of three networks. h_θ is the backbone network shared across different tasks to extract general features. f_ϕ and f_γ are two amortized networks to estimate the posterior and prior distributions over ω. q and p refer to posterior and prior distributions. r_x are features extracted over samples drawn from $D\backslash C$. These features are l2-normalized as well as average pooled over samples in the batch.

On the left, we show the posterior and the priors generated in the sequence of tasks. On the right, the inference model is depicted. To predict a label for a given query sample, first, the input images and its corresponding coreset are forwarded through h_θ and their features are computed: r_x^t and r_c^t. Next, we feed r_c^t through f_ϕ and estimate the posterior distribution

[1] AIM Lab, University of Amsterdam, The Netherlands [2] Inception Institute of Artificial Intelligence, UAE. Correspondence to: M. Derakhshani <m.m.derakhshani@uva.nl>, X. Zhen <x.zhen@uva.nl>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

over $q_\phi(\omega \mid \mathcal{C}_t)$. Then, we create the random Fourier bases ω_t by drawing samples from estimated posterior distribution. Having random bases for the current task t, ω_t, as well as r_x^t and r_c^t, random Fourier features related the query input $\psi(r_x^t)$ and coreset data $\psi(r_c^t)$ are estimated. Each kernel, K and $\tilde{K}$, is estimated using its corresponding random Fourier features and $\mathrm{k}\left(\mathbf{x}, \mathbf{x}'\right) = \psi(\mathbf{x})\psi(\mathbf{x}')^\top$. Based on Eq. (5) in the main manuscript, these two estimated kernels are used to predict the output labels for given query samples.

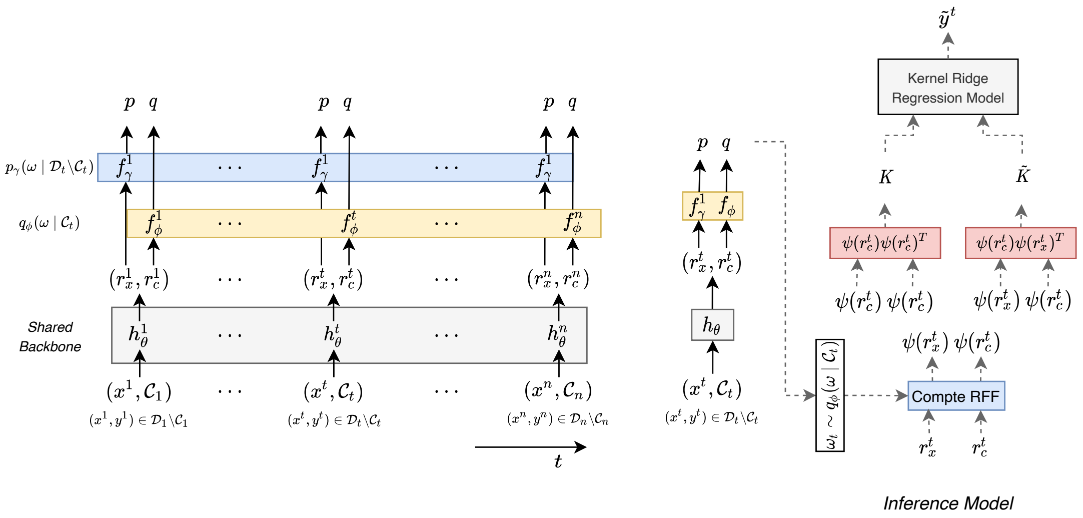
p q
$p_\gamma(\omega \mid \mathcal{D}_t \backslash \mathcal{C}_t)$
f_γ^1
$q_\phi(\omega \mid \mathcal{C}_t)$
f_ϕ^1
f_ϕ^t
f_ϕ^n
(r_x^1, r_c^1)
(r_x^t, r_c^t)
(r_x^n, r_c^n)
Shared Backbone
h_θ^1
h_θ^t
h_θ^n
$(x^1, \mathcal{C}_1)$
$(x^t, \mathcal{C}_t)$
$(x^n, \mathcal{C}_n)$
$(x^1, y^1) \in \mathcal{D}_1 \backslash \mathcal{C}_1$
$(x^t, y^t) \in \mathcal{D}_t \backslash \mathcal{C}_t$
$(x^n, y^n) \in \mathcal{D}_n \backslash \mathcal{C}_n$
t
f_γ^1 f_ϕ
h_θ
$\tilde{y}^t$
Kernel Ridge Regression Model
K
$\tilde{K}$
$\psi(r_c^t)\psi(r_c^t)^T$
$\psi(r_c^t)\psi(r_x^t)^T$
$\psi(r_c^t)$ $\psi(r_c^t)$
$\psi(r_x^t)$ $\psi(r_c^t)$
$\omega_t \sim q_\phi(\omega \mid \mathcal{C}_t)$
Compte RFF
r_x^t
r_c^t
Inference Model

Figure B.1. Kernel continual learning model with variational random features.

Note that for the variant of our variational random features with an uninformative prior. the prior network is removed and the prior is set to a standard Gaussian distribution. In addition, by using linear, polynomial, and radial basis function kernels, neither the prior network nor the posterior network is used.

C. Hyperparameters

We provide the detailed hyperparameter settings in Table C.1, which are used to to generate Figure 7 and Table 3 in the main paper for each dataset.

Table C.1. Hyperparameters used to generate results in Figure 7 and Table 3.

Method	Permuted MNIST	Rotated MNIST	Split CIFAR100	Split miniImageNet
Batch Size	10	10	10	10
Learning Rate (LR)	0.1	0.1	0.3	0.3
LR Decay Factor	0.8	0.8	0.95	0.95
Momentum	0.8	0.8	0.8	0.8
Dropout	0.5	0.5	0.02	0.02
Coreset Size	20	20	20	30
Number of Bases	1024	1024	2048	2048
Number of Tasks	20	20	20	20
Tau	0.01	0.01	0.01	0.01

D. Additional ablation results on miniImageNet

We provide additional ablation results on the miniImageNet dataset. We report the influence of the inference memory and the number of Random bases. Figure D.2 shows increasing the coreset size from 1 to 20, improves average accuracy consistently. It saturates between 20 to 40. By enlarging the coreset size to 50, model performance increases again. In Table 3 in the main paper the results related to miniImageNet are reported based on a coreset size of 30. Figure D.3 highlights how the number of random bases affects the average accuracy. Consistent with the findings in other datasets, the performance

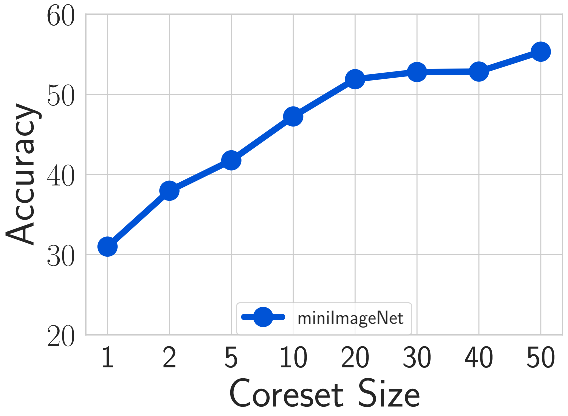

Figure D.2. **How much inference memory?** Enlarging the coreset size of the VRF kernel leads to improvement of performance on miniImageNet benchmark dataset. Coreset size 30 is chosen for conducting experiment in the main paper.

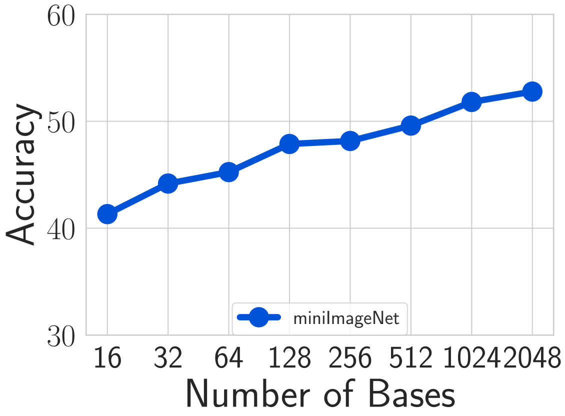

Figure D.3. **How many Random Bases?** In general, a larger number of random Fourier bases consistently improves performance on miniImageNet benchmark dataset. In the main paper, number of bases is set to be 2048.

increases with a larger number of random Fourier bases. Since miniImageNet is a challenging benchmark, we choose 2048 as the number of bases for all remaining experiments in the main paper.