Continual Learning via Sequential Function-Space Variational Inference

Tim G. J. Rudner [1] **Freddie Bickford Smith** [1] **Qixuan Feng** [1] **Yee Whye Teh** [1] **Yarin Gal** [1]

Abstract

Sequential Bayesian inference over predictive functions is a natural framework for continual learning from streams of data. However, applying it to neural networks has proved challenging in practice. Addressing the drawbacks of existing techniques, we propose an optimization objective derived by formulating continual learning as sequential function-space variational inference. In contrast to existing methods that regularize neural network parameters directly, this objective allows parameters to vary widely during training, enabling better adaptation to new tasks. Compared to objectives that directly regularize neural network predictions, the proposed objective allows for more flexible variational distributions and more effective regularization. We demonstrate that, across a range of task sequences, neural networks trained via sequential function-space variational inference achieve better predictive accuracy than networks trained with related methods while depending less on maintaining a set of representative points from previous tasks.

1. Introduction

Continual learning promises to enable applications of machine learning to settings with resource constraints, privacy concerns, or non-stationary data distributions. However, continual learning in deep neural networks remains a challenge. While progress has been made to mitigate "forgetting" of previously learned abilities, existing objective-based approaches to continual learning still fall short.

A popular family of objectives penalizes changes in parameters from one task to another (Ahn et al., 2019; Aljundi et al., 2018; Chaudhry et al., 2018; Kirkpatrick et al., 2017; Lee et al., 2017; Liu et al., 2018; Loo et al., 2020; Nguyen et al., 2018; Park et al., 2019; Ritter et al., 2018; Schwarz et al., 2018; Swaroop et al., 2019; Yin et al., 2020a;b; Zenke et al., 2017). However, explicitly regularizing parameters in this way may be ineffective, since parameters are only a proxy for a neural network's predictive function. For example, predictive functions defined by overparameterized neural networks may be obtained with several different parameter configurations, and small changes in a network's parameters may cause large changes in its predictions.

An alternative approach that addresses this shortcoming is to regularize the *predictive function* directly (Benjamin et al., 2019; Bui et al., 2017; Buzzega et al., 2020; Jung et al., 2018; Kapoor et al., 2021; Kim et al., 2018; Li and Hoiem, 2018; Moreno-Muñoz et al., 2019; Pan et al., 2020; Titsias et al., 2020). Existing function-space regularization methods represent the state of the art among objective-based approaches to continual learning (Kapoor et al., 2021; Pan et al., 2020; Titsias et al., 2020). Yet, as we demonstrate, these methods still leave room for improvement. For example, "functional regularization of the memorable past" (FROMP; Pan et al., 2020) uses a Laplace approximation and as such does not directly optimize variance parameters, while "functional regularization for continual learning" (FRCL; Titsias et al., 2020) is constrained to linear models.

To address these limitations, we frame continual learning as sequential function-space variational inference (S-FSVI) and adapt the variational objective proposed by Rudner et al. (2021) to the continual-learning setting. The resulting variational optimization objective has three key advantages over existing alternatives. First, it is expressed purely in terms of distributions over predictive functions, which allows greater flexibility than with parameter-space regularization methods (Figure 1). Second, unlike FROMP, it allows direct optimization of variational variance parameters. Third, unlike FRCL, it can be applied to fully-stochastic neural networks—not just to Bayesian linear models.

We demonstrate that S-FSVI outperforms existing objective-based continual learning methods—in some cases by a significant margin—on a wide range of task sequences, including single-head split MNIST, multi-head split CIFAR, and multi-head sequential Omniglot. We further present empirical results that showcase the usefulness of learned variational variance parameters and demonstrate that S-FSVI is less reliant on careful selection of datapoints that summarize past tasks than other methods.

[1]University of Oxford, Oxford, UK. Correspondence to: Tim G. J. Rudner <tim.rudner@cs.ox.ac.uk>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

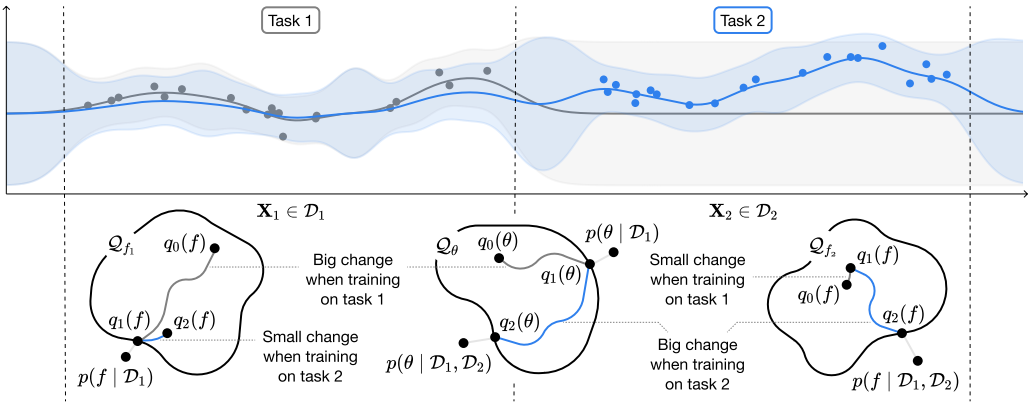


Figure 1. Schematic of how sequential function-space variational inference (S-FSVI) allows a Bayesian neural network to learn new tasks while maintaining previously learned abilities. *(Top: predictive distributions.)* On task 1, the model fits dataset D_1 by updating an initial distribution over parameters $q_0(\boldsymbol{\theta})$ to a variational posterior $q_1(\boldsymbol{\theta})$, which in turn induces a distribution over functions $q_1(f)$. On task 2, the variational objective encourages the posterior distribution over functions to match $q_1(f)$ on a small set of data points from task 1 while also fitting dataset D_2. The mean and two standard deviations of the distributions over functions learned on task 1 and task 2 are shown in grey and blue, respectively. *(Bottom: learning trajectories.)* On task 1, the distribution over functions changes by a large amount for inputs $\mathbf{X}_1$ (left) but by a small amount for inputs $\mathbf{X}_2$ (right). On task 2, the reverse is true. On both tasks, the change in the distribution over parameters (center) is decoupled from the changes in the distribution over functions (left, right).

2. Background

2.1. Continual Learning as Bayesian Inference

Consider a sequence of tasks indexed by t 2 f1, ..., Tg. Each task involves making predictions on a supervised dataset $D_t = (\mathbf{X}_t, \mathbf{y}_t)$. Continual learning is the problem of inferring a distribution over predictive functions that fits the whole collection of datasets fD1, ..., DTg as well as possible given access to only a single full dataset at a time.

Sequential Bayesian inference over predictive functions f provides a natural framework for this. Assuming we have a prior $p(f)$, the posterior distribution over f at task 1 is

$$p(f \mid D_1) = p(D_1 \mid f)p(f)/p(D_1). \tag{1}$$

For subsequent tasks t, the posterior can be expressed as

$$p(f \mid D_1, \ldots, D_t) \;/\; p(D_t \mid f)p(f \mid D_1, \ldots, D_{t-1}), \tag{2}$$

where the posterior after task $t-1$ is treated as the prior for task t. Given the intractibility of computing this posterior exactly, we need to use approximate inference.

2.2. Function-Space Variational Inference

Given a dataset $D = (\mathbf{X}, \mathbf{y})$, a prior $p(f)$ and a variational family Q_f, function-space variational inference (Burt et al., 2021; Matthews et al., 2016; Rudner et al., 2021; Sun et al., 2019) consists of finding the variational distribution $q(f)$ 2 Q_f that maximizes

$$\mathbb{E}_{q(f)}[\log p(\mathbf{y} \mid f(\mathbf{X}))] \quad \mathrm{D}_{\mathrm{KL}}(q(f) \,\|\, p(f)). \tag{3}$$

This variational optimization problem presents a trade-off between fitting the data and matching a prior over functions. To address the fact that the KL divergence between distributions over functions is not in general tractable, prior works have developed estimation procedures that allow turning Equation (3) into an objective function that can be used in practice (Rudner et al., 2021; Sun et al., 2019).

3. Continual Learning via Sequential Function-Space Variational Inference

The ideas presented in Section 2 provide a starting point for our method. To approximate the posterior in Equation (2) at task t, we would like to find a variational distribution $q_t(f)$ 2 Q_f that minimizes

$$\mathrm{D}_{\mathrm{KL}}(q_t(f) \,\|\, p_t(f \mid D_1, ..., D_t)), \tag{4}$$

which can equivalently be expressed as maximizing

$$\mathbb{E}_{q_t(f)}[\log p(\mathbf{y}_t \mid f(\mathbf{X}_t))] \quad \mathrm{D}_{\mathrm{KL}}(q_t(f) \,\|\, p_t(f \mid D_1, ..., D_{t-1})).$$

Since we do not have access to $p_t(f \mid D_1, ..., D_{t-1})$, we simplify the inference problem to maximizing the variational objective

$$\mathbb{E}_{q_t(f)}[\log p(\mathbf{y}_t \mid f(\mathbf{X}_t))] \quad \mathrm{D}_{\mathrm{KL}}(q_t(f) \,\|\, p_t(f)), \tag{5}$$

where for $t = 1$ we assume some prior $p_1(f)$ and for $t > 1$ the prior is given by the variational posterior distribution over functions inferred on the previous task. That is,

$$p_t(f) \doteq q_{t-1}(f).$$

While this objective is in general intractable for distributions over functions induced by neural networks with stochastic parameters, Rudner et al. (2021) proposed an approximation that makes this objective amenable to gradient-based optimization and scalable to large neural networks. To perform sequential function-space variational inference, we adapt the estimation procedure proposed by Rudner et al. (2021) to the continual-learning setting:

Proposition 1 (Sequential Function-Space Variational Inference (S-FSVI); adapted from Rudner et al. (2021))**.** *Let D_t be the number of model output dimensions for t tasks, let $f : X \quad \mathsf{R}^P \; ! \; \mathsf{R}^{D_t}$ be a mapping defined by a neural network architecture, let $\Theta \; 2 \; \mathsf{R}^P$ be a multivariate random vector of network parameters, and let $q_t(\theta) \doteq \mathcal{N}(\mu_t, \Sigma_t)$ and $q_{t\ 1}(\theta) \doteq \mathcal{N}(\mu_{t\ 1}, \Sigma_{t\ 1})$ be variational distributions over Θ. Additionally, let $\mathbf{X}_C$ denote a set of context points, and let $\bar{\mathbf{X}}_t \quad f\mathbf{X}_t \; [\; \mathbf{X}_C g$. Under a diagonal approximation of the prior and variational posterior covariance functions across output dimensions, the objective in Equation (5) can be approximated by*

$$
\begin{aligned}
&\mathcal{F}(q_t, q_{t\ 1}, \mathbf{X}_C, \mathbf{X}_t, \mathbf{y}_t) \\
&\doteq \mathsf{E}_{q_t(\theta)}[\log p(\mathbf{y}_t \, j \, f(\mathbf{X}_t; \theta))] \\
&\quad \sum_{k=1}^{D_t} \frac{1}{2} \Bigg(\log \frac{j[\mathbf{K}^{p_t}]_k j}{j[\mathbf{K}^{q_t}]_k j} \quad \frac{j \bar{\mathbf{X}}_t j}{D_t} + \mathrm{Tr}([\mathbf{K}^{p_t}]_k^{\ 1} [\mathbf{K}^{q_t}]_k) \\
&\quad + \Delta(\bar{\mathbf{X}}_t; \mu_t, \mu_{t\ 1})^{>} [\mathbf{K}^{p_t}]_k^{\ 1} \Delta(\bar{\mathbf{X}}_t; \mu_t, \mu_{t\ 1}) \Bigg),
\end{aligned} \tag{6}
$$

where

$$
\Delta(\bar{\mathbf{X}}_t; \mu_t, \mu_{t\ 1}) \doteq [f(\bar{\mathbf{X}}_t; \mu_t)]_k \quad [f(\bar{\mathbf{X}}_t; \mu_{t\ 1})]_k \tag{7}
$$

and

$$
\mathbf{K}^{p_t} \doteq \mathcal{J}(\bar{\mathbf{X}}_t, \mu_{t\ 1}) \Sigma_{t\ 1} \mathcal{J}(\bar{\mathbf{X}}_t, \mu_{t\ 1})^{>} \tag{8}
$$

$$
\mathbf{K}^{q_t} \doteq \mathcal{J}(\bar{\mathbf{X}}_t, \mu_t) \Sigma_t \mathcal{J}(\bar{\mathbf{X}}_t, \mu_t)^{>}, \tag{9}
$$

are covariance matrix estimates constructed from Jacobians $\mathcal{J}(\ , \mathbf{m}) \doteq \frac{\partial f(\ ; \Theta)}{\partial \Theta} j_{\Theta = \mathbf{m}}$ with $\mathbf{m} = f\mu_t, \mu_{t\ 1} g$.

Proof. See Appendix A. □

"Functional regularization for continual learning" (FRCL; Titsias et al., 2020) and "functional regularization of the memorable past" (FROMP; Pan et al., 2020) use objectives conceptually similar to the objective in Equation (5) and mathematically similar to the objective in Equation (6). To highlight the differences between the S-FSVI objective above and FROMP and FRCL, respectively, we make the relationship between these two methods and S-FSVI precise in the following two propositions.

Proposition 2 (Relationship between FROMP and S-FSVI)**.** *With the S-FSVI objective $\mathcal{F}$ defined as in Equation (6), let $\bar{\mathbf{X}}_t = \mathbf{X}_C$. Then, up to a multiplicative constant, the FROMP objective corresponds to the S-FSVI objective with the prior covariance given by a Laplace approximation about $\mu_{t\ 1}$ and the variational distribution given by a Dirac delta distribution $q_t^{\text{FROMP}}(\theta) \doteq \delta(\theta \quad \mu_t)$. Denoting the prior covariance under a Laplace approximation about $\mu_{t\ 1}$ by $\hat{\Sigma}_0(\mu_{t\ 1})$ so that $q_{t\ 1}^{\text{FROMP}}(\theta) \doteq \mathcal{N}(\mu_{t\ 1}, \hat{\Sigma}_0(\mu_{t\ 1}))$, the FROMP objective can be expressed as*

$$
\begin{aligned}
&\mathcal{L}^{\text{FROMP}}(q_t^{\text{FROMP}}, q_{t\ 1}^{\text{FROMP}}, \mathbf{X}_C, \mathbf{X}_t, \mathbf{y}_t) \\
&\quad = \mathcal{F}(q_t^{\text{FROMP}}, q_{t\ 1}^{\text{FROMP}}, \mathbf{X}_C, \mathbf{X}_t, \mathbf{y}_t) \quad \mathcal{V},
\end{aligned}
$$

where

$$
\mathcal{V} \doteq \quad \frac{1}{2} \sum_k \left(\log \frac{[\bar{\mathbf{K}}^{\hat{p}_t}]_k}{[\bar{\mathbf{K}}^{q_t}]_k} + \frac{[\bar{\mathbf{K}}^{q_t}]_k}{[\bar{\mathbf{K}}^{\hat{p}_t}]_k} \quad 1 \right),
$$

with $\bar{\mathbf{K}}$ denoting a covariance matrix under a block-diagonalization without inter-task dependence, and

$$
\bar{\mathbf{K}}^{\hat{p}_t} \doteq \text{block-diag}\Big(\mathcal{J}(\bar{\mathbf{X}}_t, \mu_{t\ 1}) \hat{\Sigma}_0(\mu_{t\ 1}) \mathcal{J}(\bar{\mathbf{X}}_t, \mu_{t\ 1})^{>} \Big).
$$

Proof. See Appendix A. □

Proposition 2 shows that the FROMP objective nearly corresponds to the S-FSVI objective but is missing the term in the S-FSVI objective (denoted by $\mathcal{V}$ above) that encourages learning variational variance parameters that accurately reflect the variance of the prior. This insight reflects a shortcoming of the FROMP objective. Unlike in the S-FSVI objective which allows optimization over Σ, the FROMP objective is restricted to covariance estimates given by the Laplace approximation.

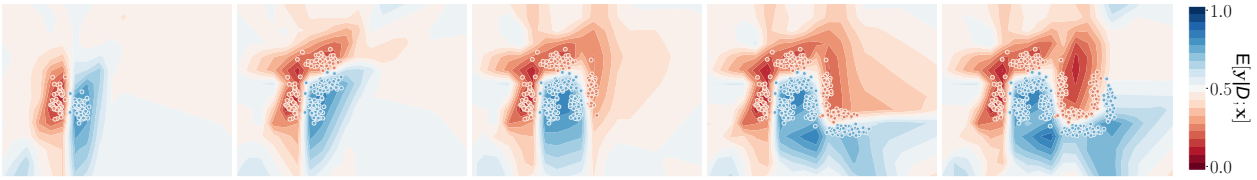
The FRCL objective can be related to the S-FSVI objective in a similar way:

Proposition 3 (Relationship between FRCL and S-FSVI)**.** *With the S-FSVI objective $\mathcal{F}$ defined as in Equation (6), let $\bar{\mathbf{X}}_t = \mathbf{X}_C$, and let $f^{\text{LM}}(\ ; \Theta) \doteq \Phi_\psi(\ ) \Theta$ be a Bayesian linear model, where $\Phi_\psi(\ )$ is a deterministic feature map parameterized by ψ. Then the FRCL objective corresponds to the S-FSVI objective for the model $f^{\text{LM}}(\ ; \Theta)$ plus an additional weight-space KL divergence penalty. That is, for $p_t(\theta) \doteq \mathcal{N}(\mathbf{0}, \mathbf{I})$, and $q_t(\theta) \doteq \mathcal{N}(\mu_t, \Sigma_t)$,*

$$
\begin{aligned}
&\mathcal{L}^{\text{FRCL}}(q_t^{\text{FRCL}}, q_{t\ 1}^{\text{FRCL}}, \mathbf{X}_C, \mathbf{X}_t, \mathbf{y}_t) \\
&\quad = \mathcal{F}(q_t^{\text{FRCL}}, q_{t\ 1}^{\text{FRCL}}, \mathbf{X}_C, \mathbf{X}_t, \mathbf{y}_t) + \mathsf{D}_{\text{KL}}(q_t(\theta) \, k \, p_t(\theta)).
\end{aligned} \tag{10}
$$

Proof. See Appendix A. □

Proposition 3 highlights that the FRCL objective is restricted to Bayesian linear models and does not regularize the deterministic parameters in the feature map as effectively as if they were variational parameters.



(a) Task 1 **(b)** Task 2 **(c)** Task 3 **(d)** Task 4 **(e)** Task 5

Figure 2. A practical demonstration of sequential function-space variational inference (S-FSVI) on a sequence of five binary-classification tasks with 2D inputs. The neural network infers a decision boundary between the two classes while maintaining high predictive uncertainty away from the data. The experimental setup is described in detail in Appendix C.

3.1. Simplified Sequential Function-Space VI

For ease of computation and to ensure scalability to large neural networks, we consider mean-field distributions $q_t^{\mathrm{MF}}(\boldsymbol{\theta})$ for all tasks, diagonalize the covariance matrix estimates $\mathbf{K}^{p_t}$ and $\mathbf{K}^{q_t}$ across input points in $\bar{\mathbf{X}}_t$, and let $(\mathbf{X}_B, \mathbf{y}_B)$ D_t be a mini-batch from the current dataset. This way, we obtain the simplified variational objective

$$\begin{aligned}
&\tilde{F}(q_t^{\mathrm{MF}}, q_t^{\mathrm{MF}}{}_1, \mathbf{X}_C, \mathbf{X}_B, \mathbf{y}_B) \\
&= \frac{1}{S}\sum_{i=1}^{S} \log p(\mathbf{y}_B \,|\, f(\mathbf{X}_B; h(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t, \boldsymbol{\epsilon}^{(i)}))) \\
&\quad \sum_{j=1}^{J\bar{\mathbf{X}}j}\sum_{k=1}^{D_t} \frac{1}{2}\Bigg(\log \frac{[\mathbf{K}^{p_t}]_{j,k}}{[\mathbf{K}^{q_t}]_{j,k}} + \frac{[\mathbf{K}^{q_t}]_{j,k}}{[\mathbf{K}^{p_t}]_{j,k}} \quad 1 \\
&\qquad + \frac{\left([f(\bar{\mathbf{X}}_t; \boldsymbol{\mu}_t)]_{j,k} \quad [f(\bar{\mathbf{X}}_t; \boldsymbol{\mu}_t \;\; {}_1)]_{j,k}\right)^2}{[\mathbf{K}^{p_t}]_{j,k}}\Bigg),
\end{aligned} \tag{11}$$

where $h(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t, \boldsymbol{\epsilon}^{(i)}) \doteq \boldsymbol{\mu}_t + \boldsymbol{\Sigma}_t \quad \boldsymbol{\epsilon}^{(i)}$ is a reparameterization of $\boldsymbol{\Theta}$ 2 R^P with $\boldsymbol{\epsilon}^{(i)} \quad N(\mathbf{0}, \mathbf{I}_P)$, S is the number of Monte Carlo samples, D_t is as defined before, and

$$\mathbf{K}^{p_t} \doteq \mathrm{diag}\left(J(\bar{\mathbf{X}}_t, \boldsymbol{\mu}_t \;\; {}_1)\boldsymbol{\Sigma}_t \;\; {}_1 J(\bar{\mathbf{X}}_t, \boldsymbol{\mu}_t \;\; {}_1)^{>}\right) \tag{12}$$

$$\mathbf{K}^{q_t} \doteq \mathrm{diag}\left(J(\bar{\mathbf{X}}_t, \boldsymbol{\mu}_t)\boldsymbol{\Sigma}_t J(\bar{\mathbf{X}}_t, \boldsymbol{\mu}_t)^{>}\right). \tag{13}$$

This simplified objective does not require matrix inversion, and the time and space complexity for gradient estimation and prediction scale linearly in the number of context points $\bar{\mathbf{X}}_t$ and network parameters. The context set $\mathbf{X}_C$ can be constructed from coresets containing representative points from previous tasks.

We provide an empirical comparison of the simplified S-FSVI, FROMP, and FRCL objectives in Section 5 to assess the extent to which the differences described above affect continual learning.

4. Related Work

There are three main (partially overlapping) categories of methods for continual learning in a deep neural network. Objective-based approaches modify the objective function used to train the neural network. Replay-based approaches summarize past tasks using either stored data or freshly generated synthetic data. Architecture-based approaches change the neural network's structure from one task to another. For extensive reviews, see De Lange et al. (2021) and Parisi et al. (2019). As sequential function-space variational inference (S-FSVI) centers around a new training objective, we focus on objective-based approaches in this review. (Like the methods reviewed below, S-FSVI does incorporate a form of replay in that it uses context points, but the primary interest is the training objective.)

For a neural network to retain abilities it has previously learned, its predictions on data associated with past tasks must not change significantly from one task to another. One way of achieving this is to include in the training objective a form of function-space regularization to discourage important changes in the network's predictions or internal representations. "Learning without forgetting" (Li and Hoiem, 2018) uses a modified cross-entropy loss that penalizes the difference between the predictions of the current network on the current task data and the predictions of the previous network on the current task data. "Less-forgetful learning" (Jung et al., 2018) employs the same method but uses squared Euclidean distance rather than the modified cross-entropy loss and applies it to the penultimate-layer representations rather than the network's predictions. "Keep and learn" (Kim et al., 2018) also uses internal representations as a basis for regularization. The method subsequently proposed by Benjamin et al. (2019) involves comparing the current network with all previous versions of the network and on data from all past tasks instead of with only the most recent network on data from the current task. Each pair of networks is compared by computing the Euclidean distance between the networks' predictions. "Dark experience replay" (Buzzega et al., 2020) extends this method to work in a setting where task boundaries are not clearly defined.

While these approaches mitigate forgetting, they do not explicitly account for predictive uncertainty, which is an issue if the neural network is a poor fit to the data. This deficiency is addressed by probabilistic approaches to function-space regularization, which encourage a network's predictions to

Table 1. Predictive accuracies of a selection of objective-based methods for continual learning. Results are reported for three task sequences: split MNIST (S-MNIST), split Fashion MNIST (S-FMNIST) and permuted MNIST (P-MNIST). In some cases, a multi-head setup (MH) is used; in others, a single-head setup (SH). Best results for identical network architectures are printed in boldface (exception: VAR-GP uses a non-parametric model). Best overall results are highlighted in gray. Each numerical entry denotes the mean accuracy across tasks at the end of training. Where possible, this accuracy is based on experiments repeated with different random seeds (10 repeats for S-FSVI), with both the mean value and standard error reported. All methods use the same architecture and coreset size unless indicated otherwise. See Appendix C for more experimental details. [1]Accuracies computed using the best coreset-selection method (either random or k-center). [2]Uses random coreset selection. [3]Requires a multi-head setup with task identifiers, including for permuted MNIST. This requirement explains the missing FRCL result for S-MNIST (SH). [4]Uses a larger MLP architecture (see Table 4 in appendix).[5]Evaluates the KL divergence at points sampled from the empirical data distribution of the current task. [6]Uses one sample per class as a coreset.

Method	S-MNIST (MH)	S-FMNIST (MH)	P-MNIST (SH)	S-MNIST (SH)
EWC (Kirkpatrick et al., 2017)	63.10%	—	84.00%	—
SI (Zenke et al., 2017)	98.90%	—	86.00%	—
VCL (Nguyen et al., 2018)[1]	98.40%	98.60% 0.04	93.00%	32.11% 1.16
VCL (no coreset)	97.00%	89.60% 1.75	87.50% 0.61	17.74% 1.20
FRCL (Titsias et al., 2020)[3]	97.80% 0.22	97.28% 0.17	94.30% 0.06	—
FROMP (Pan et al., 2020)	99.00% 0.04	99.00% 0.03	94.90% 0.04	35.29% 0.52
VAR-GP (Kapoor et al., 2021)	—	—	**97.20%** 0.08	90.57% 1.06
S-FSVI (**ours**)[2]	**99.54%** 0.04	**99.19%** 0.02	95.76% 0.02	**92.87%** 0.14
S-FSVI Ablation Study:				
S-FSVI (larger networks)[4]	**99.76%** 0.00	99.16% 0.03	**97.50%** 0.01	**93.38%** 0.10
S-FSVI (no coreset)[5]	99.62% 0.02	**99.54%** 0.01	84.06% 0.46	20.15% 0.52
S-FSVI (minimal coreset)[6]	—	—	89.59% 0.30	51.44% 1.22

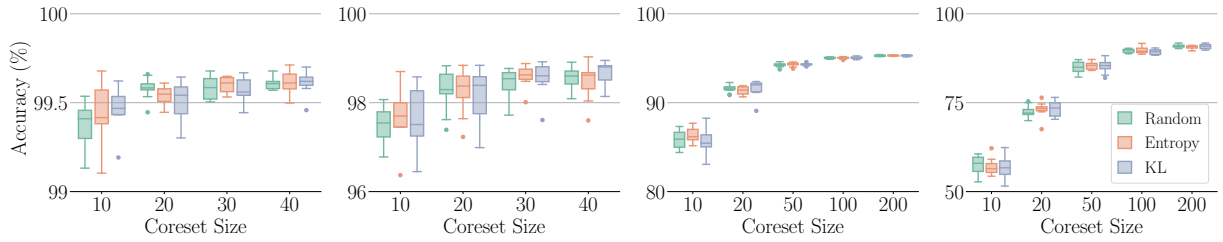
agree with a prior distribution over functions rather than with a single function. "Functional regularization for continual learning" (FRCL; Titsias et al., 2020) considers a network whose final layer is a Bayesian linear model. Based on the duality between parameter space and function space, the FRCL objective includes the KL divergence between predictive distributions at a selection of input points. This encourages similarity between the network's current predictive distribution and the distributions from past tasks. FRCL is theoretically appealing, building on a well-understood method for stochastic variational inference using inducing points, but is only applicable to Bayesian linear models. In contrast, "functional regularization of the memorable past" (FROMP; Pan et al., 2020) maintains a posterior distribution over all the parameters of a neural network. While FROMP achieves state-of-the-art performance on several continual-learning task sequences, it relies on a change in the underlying probabilistic model and uses a surrogate objective for optimization, which divorces it from function-space variational objectives. As we show, this results in suboptimal performance compared to sequential function-space variational inference, which maintains a stronger link to the underlying Bayesian approximation.

Although our focus is on methods for training deep neural networks, for completeness, we also note methods based on Gaussian processes (GPs). Incremental variational sparse GP regression (Cheng and Boots, 2016), streaming sparse GPs (Bui et al., 2017) and online sparse multi-output GP regression (Yang et al., 2019) built on the work of Csató and Opper (2002) and Csató (2002), and are effective approaches to continual learning for regression tasks. Continual multi-task GPs (Moreno-Muñoz et al., 2019) extend to multi-output settings with non-Gaussian likelihoods. The success of variational autoregressive GPs (VAR-GP; Kapoor et al., 2021) on continual learning for task sequences with image inputs gives reason for inclusion where relevant in Section 5. However, we note that VAR-GP scales poorly with the number of tasks: the time complexity for inference is cubic in the number of context points and hence in the number of tasks, which may limit its applicability to task sequences like sequential Omniglot. In contrast, the time complexity of S-FSVI is linear in the number of context points.

Also distinct from but related to our method are a number of objective-based approaches to continual learning that directly regularize the parameters of a neural network. We briefly discuss these approaches in Appendix D.

5. Empirical Evaluation

After visualizing how S-FSVI works in practice (Section 5.1), we compare S-FSVI's performance with that of existing objective-based methods for continual learning (Sections 5.2 to 5.4). For a comprehensive comparison, we evaluate S-FSVI on a range of task sequences used in related work. Aiming to use as strong baselines as possible, we report results taken directly from the literature in most cases (and mention when we do not). Reporting baselines in this way leaves gaps in our comparison: for each existing technique, results are available for only a subset of the task sequences we consider here (e.g., Pan et al. (2020) report results for split CIFAR but not sequential Omniglot, while Titsias et al. (2020) do the reverse).



(a) S-MNIST (MH) **(b)** S-FMNIST (MH) **(c)** P-MNIST (SH) **(d)** S-MNIST (SH)

Figure 3. Effect of the coreset size and coreset-selection method on the predictive accuracy of S-FSVI. Three coreset-selection methods are presented: sampling data points with uniform probability; sampling with probability proportional to model's predictive entropy; and sampling with probability proportional to the KL divergence between the posterior predictive distribution and the prior predictive distribution. Ten inducing points are used in each case. No coreset-selection method consistently yields higher accuracy.

Our evaluation pays attention to two factors important in the assessment of continual-learning methods: the use of task identifiers when making predictions, and the use of a coresets of data points to summarize past tasks (Farquhar and Gal, 2018). To provide some commentary on the first of these factors, we run an experiment that compares the performance of a single-head neural network (which does not use task identifiers) to that of a multi-head neural network (which uses task identifiers). Regarding the second factor, we explore how performance changes when the coreset size changes or a context set unrelated to previous tasks is used.

Details about the experimental setups (e.g., optimization routines and hyperparameter searches) can be found in Appendix C. Our code can be accessed at:

`https://timrudner.com/sfsvi-code`.

5.1. Illustrative Example

To provide intuition for how S-FSVI allows learning on new tasks while maintaining previously acquired abilities, we apply it to a task sequence based on easy-to-visualize synthetic 2D data, originally proposed by Pan et al. (2020). In this task sequence, each data point belongs to one of two classes, and more data points are revealed as the task sequence progresses. The data-generating process is assumed to reveal data from mostly non-overlapping subsets of the input space. The continual-learning problem is then to infer the decision boundary around data points revealed up to and including the current task without forgetting the decision boundary inferred on previous tasks. We use a single-head neural network.

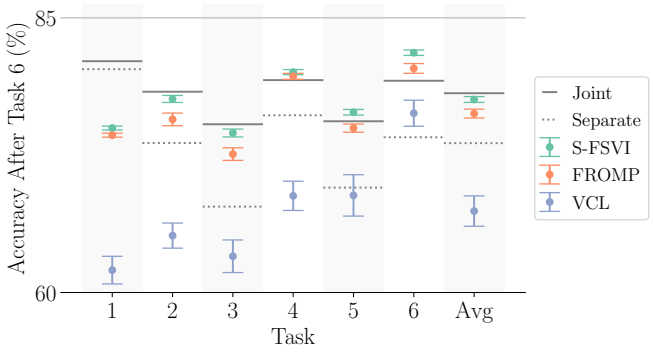
In Figure 2, we plot the model's posterior predictive distribution after training on each of five tasks. After training on task 1, the model has low predictive uncertainty close to the data points and high uncertainty (class probabilities around 0.5) everywhere else (Figure 2a). On task 2, S-FSVI seeks to match the distribution over functions inferred on the previous task while fitting the new set of data points. S-FSVI achieves this and expands the area in input space where the model is confident in its predictions (Figure 2b).

As more tasks and data are revealed, S-FSVI allows the model to continually explore the data space and infer the decision boundary while maintaining accurate, high-confidence predictions on data points in parts of the inputs space where it was previously trained on observed data. Finally, after training on five tasks, the model has inferred the decision boundary between the two classes, while maintaining high predictive uncertainty in parts of the input space where no data points have been observed yet (Figure 2e). The model maintains high predictive uncertainty away from the data, which makes it easier to learn on new tasks. This is unlike deterministic neural networks, which tend to make highly confident predictions in parts of the inputs space where no data has been observed, or on data points that lie outside of the distribution of the training data.

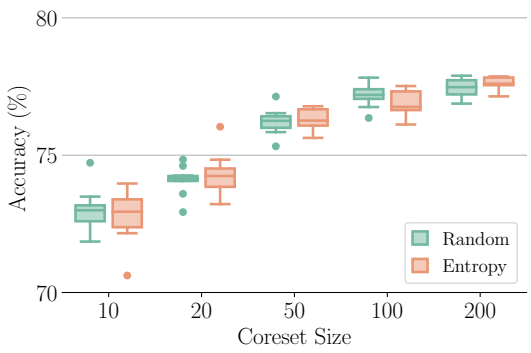
5.2. Split (Fashion) MNIST & Permuted MNIST

Having established some intuition for how S-FSVI works, we demonstrate how this translates to high predictive accuracy on three task sequences commonly used to evaluate continual-learning methods. First is split MNIST (S-MNIST), in which each task consists of binary classification on a pair of MNIST classes (0 vs. 1, 2 vs. 3, and so on). Second is split Fashion MNIST (S-FMNIST), which has the same structure but uses data from Fashion MNIST, posing a harder problem. Third is permuted MNIST (P-MNIST), in which each task consists of ten-way classification on MNIST images whose pixels have been randomly reordered. A multi-head setup (MH) with task identifiers provided at prediction time is the default for S-MNIST and S-FMNIST, while a single-head setup (SH) without task identifiers is standard for P-MNIST. In addition to running the default setup for all three task sequences, we run a single-head setup for S-MNIST.

With a standard configuration, S-FSVI outperforms all existing methods based on deep neural networks by a statistically



(a) Split CIFAR Accuracies After Training on All Tasks



(b) S-FSVI Accuracy as a Function of Coreset Size

Figure 4. Predictive accuracies of S-FSVI and related methods on split CIFAR. (a) Per-task and average accuracy after training on six tasks. The result of "joint" baseline is obtained using a model trained on data from all tasks at the same time. The accuracy at task t for the "separate" baseline is the accuracy of an independent model trained only on task t. We use the best performing method for each baseline: FROMP for "joint", S-FSVI for "separate". (b) Average accuracy after training on six tasks with different coreset sizes. "Random" coreset selection denotes uniform sampling from the training set. "Entropy" coreset selection denotes sampling from the training set with probability proportional to the entropy of the model's posterior predictive distribution.

significant margin on all task sequences (Table 1). As noted in Section 4, VAR-GP's conceptual connection to our method warrants its inclusion in our comparison. VAR-GP performs better than our standard configuration of S-FSVI on permuted MNIST, but this advantage disappears once a larger neural network is used with S-FSVI. Moreover, VAR-GP is unlikely to scale well to more challenging task sequences, such as those in Sections 5.3 and 5.4.

5.3. Sequential Omniglot

Sequential Omniglot (Lake et al., 2015; Schwarz et al., 2018) provides a more challenging task sequence than those considered in Section 5.2. It consists of 50 classification tasks, where the number of classes varies between the tasks (details in Appendix C). We find that S-FSVI produces better predictive accuracy than all available baselines, including FRCL, by a statistically significant margin (Table 2). To illustrate the stability of S-FSVI across long task sequences, we plot its mean accuracy over 50 tasks in Figure 5.

Table 2. Predictive accuracies of S-FSVI and related methods on sequential Omniglot. For S-FSVI and FRCL, the coreset consists of two data points per class. All baseline results are from Titsias et al. (2020). For all methods, the mean and standard deviation over five random task permutations are reported. [1]Li and Hoiem (2018). [2]Schwarz et al. (2018). [3]Schwarz et al. (2018). [4]Coreset selected using FRCL's "trace" method. [5]Details in Appendix C.

Method	Test Accuracy
Learning Without Forgetting[1]	62.06% 2.0
EWC	67.32% 4.7
Online EWC[2]	69.99% 3.2
Progress & Compress[3]	70.32% 3.3
FRCL[4]	81.47% 1.6
S-FSVI (**ours**)[5]	**83.29%** 1.2

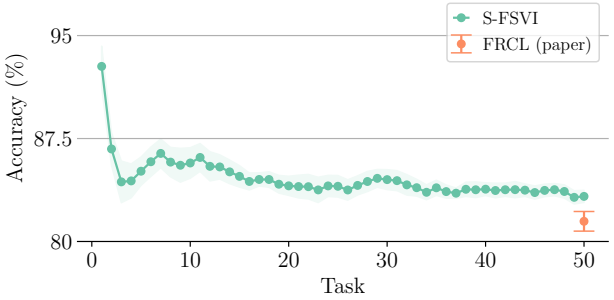
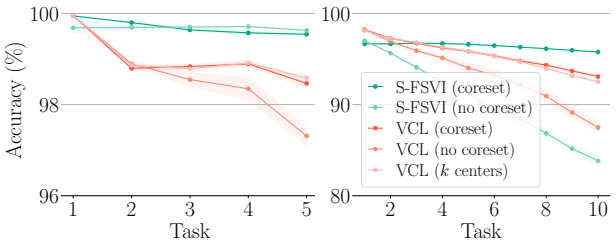


Figure 5. Predictive accuracies of S-FSVI and FRCL on sequential Omniglot. For S-FSVI, the accuracy shown at task t is the mean accuracy across all tasks up to that point (mean one standard error as computed across five permutations of the task order). We were unable to reproduce the result reported in Titsias et al. (2020) using the authors' code. However, we compare against the result from the paper (only the accuracy at task 50 is reported) here to provide a strong baseline.

5.4. Split CIFAR

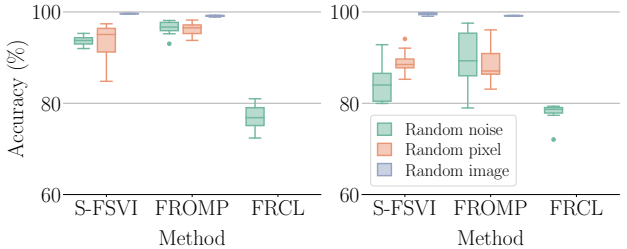
Moving beyond classification tasks on grayscale images, we evaluate S-FSVI on split CIFAR (Pan et al., 2020; Zenke et al., 2017). This uses the full CIFAR-10 dataset for the first task, followed by five ten-way classification tasks drawn from CIFAR-100. Our results show S-FSVI achieving higher accuracy on all tasks than FROMP and VCL after learning all six tasks (Figure 4a). Notably, on each task except the first, S-FSVI performs close to or better than two baselines: a model trained only on that task, and a model trained on all tasks jointly. The latter is a particularly strong baseline, because all data is available during training.

As in related work (Lopez-Paz and Ranzato, 2017; Pan et al., 2020), we compute the forward transfer (FT) and backward transfer (BT) for S-FSVI on split CIFAR. FT captures by



(a) S-MNIST (MH) **(b)** P-MNIST (SH)

Figure 6. Predictive accuracies of S-FSVI and parameter-space variational inference (VCL) on split MNIST and permuted MNIST. The accuracy shown at task t is the mean accuracy across all tasks up to that point (mean one standard error as computed across ten repetitions of the experiment). With a coreset, S-FSVI outperforms VCL on both task sequences. Without a coreset, S-FSVI performs poorly on permuted MNIST.



(a) S-MNIST (MH) **(b)** S-FMNIST (MH)

Figure 7. Predictive accuracies of S-FSVI, FROMP and FRCL on multi-head split (Fashion) MNIST without using coresets. Inducing inputs for evaluating the KL divergence are sampled according to three different sampling schemes derived from the current task's empirical data distribution (see Appendix C for details). Using S-FSVI with images sampled from the current task's training set significantly outperforms all other methods.

how much the accuracy on the current tasks increases as the number of past tasks increases; BT captures by how much the accuracy on the previous tasks increases as more tasks are observed (see Appendix C.6 for mathematical definitions). As well as having the best overall accuracy, S-FSVI significantly outperforms all baselines in terms of FT and has BT comparable to EWC and FROMP (Table 3).

Table 3. Forward transfer (FT) and backward transfer (BT) of S-FSVI and related methods on split CIFAR. All baseline results are from Pan et al. (2020). For all methods, the mean and standard error over five repeated experiments are reported. [1]Details in Appendix C.

Method	Test Accuracy	FT	BT
EWC	71.6% 0.4	0.2 0.4	**-2.3** 0.6
VCL	67.4% 0.6	1.8 1.4	-9.2 0.8
FROMP	76.2% 0.2	6.1 0.3	**-2.6** 0.4
S-FSVI (ours)[1]	**77.6%** 0.2	**7.3** 0.2	**-2.5** 0.2

5.5. Function- vs. Parameter-Space Inference

To demonstrate the importance of performing inference in function space, we compare how the accuracies of S-FSVI and VCL evolve from one task to another on split MNIST and permuted MNIST (Figure 6). We find that S-FSVI consistently outperforms VCL whose predictive performance steadily degrades suggesting that function-space inference may be more effective than parameter-space inference at transferring prior knowledge from one task to another, and that this may offset the information loss in the KL divergence between distributions over functions compared to the KL divergence between distributions over parameters.

5.6. Coreset Size and Selection

Similar to existing methods such as FROMP and FRCL, S-FSVI includes in the training objective a function-space regularization term that encourages matching the prior distribution over functions at a set of context points. Typically, this requires keeping a representative coreset of data points from each task, from which a context set can be constructed.

S-FSVI offers two benefits with respect to coresets. First, it is insensitive to which points get included in the coresets. Whereas existing methods often require expensive procedures to select important data points from previous tasks, Figures 3 and 4b show that S-FSVI achieves strong performance while only using randomly selected coresets. Second, S-FSVI does not require large coresets to perform well. On permuted MNIST, S-FSVI achieves better predictive accuracy than EWC and SI even if the coreset used for S-FSVI consists of only a single data point per class (Table 1). On the single-head version of split MNIST, a minimal coreset (one point per class, or two points per task) allows S-FSVI to outperform VCL and FROMP, both with coresets of 40 points per task (Table 1). In some multi-head settings, S-FSVI achieves state-of-the-art predictive accuracies with randomly-generated noise coresets (Table 1 and Figure 7).

6. Conclusion

We presented sequential function-space variational inference (S-FSVI), a method for continual learning in deep neural networks. We showed that S-FSVI improves on the predictive performance of existing objective-based continual learning methods—often by a significant margin—including on task sequences with high-dimensional inputs (split CIFAR) and large numbers of tasks (sequential Omniglot). Lastly, we demonstrated that—unlike existing function-space regularization methods—S-FSVI does not rely on careful coreset selection and, in multi-head settings, can achieve state-of-the-art performance even without coresets collected on previous tasks. We hope that this work will lead to future research into further improving function-space objectives for continual learning.

Acknowledgements

Tim G. J. Rudner and Freddie Bickford Smith are funded by the Engineering and Physical Sciences Research Council (EPSRC). Tim G. J. Rudner is also funded by the Rhodes Trust and by a Qualcomm Innovation Fellowship. We gratefully acknowledge donations of computing resources by the Alan Turing Institute.

References

Ahn, H., Cha, S., Lee, D., and Moon, T. (2019). Uncertainty-based continual learning with adaptive regularization. In *Advances in Neural Information Processing Systems*.

Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., and Tuytelaars, T. (2018). Memory aware synapses: learning what (not) to forget. In *European Conference on Computer Vision*.

Benjamin, A., Rolnick, D., and Kording, K. (2019). Measuring and regularizing networks in function space. In *International Conference on Learning Representations*.

Broderick, T., Boyd, N., Wibisono, A., Wilson, A. C., and Jordan, M. I. (2013). Streaming variational Bayes. In *Advances in Neural Information Processing Systems*.

Bui, T. D., Nguyen, C., and Turner, R. E. (2017). Streaming sparse Gaussian process approximations. In *Advances in Neural Information Processing Systems*.

Burt, D. R., Ober, S. W., Garriga-Alonso, A., and van der Wilk, M. (2021). Understanding variational inference in function-space. In *Symposium on Advances in Approximate Bayesian Inference*.

Buzzega, P., Boschini, M., Porrello, A., Abati, D., and Calderara, S. (2020). Dark experience for general continual learning: a strong, simple baseline. In *Advances in Neural Information Processing Systems*.

Chaudhry, A., Dokania, P., Ajanthan, T., and Torr, P. (2018). Riemannian walk for incremental learning: understanding forgetting and intransigence. In *European Conference on Computer Vision*.

Cheng, C.-A. and Boots, B. (2016). Incremental variational sparse Gaussian process regression. In *Advances in Neural Information Processing Systems*.

Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*. Wiley, New York.

Csató, L. (2002). *Gaussian processes: iterative sparse approximations*. PhD thesis, Aston University.

Csató, L. and Opper, M. (2002). Sparse on-line Gaussian processes. *Neural Computation*.

De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., and Tuytelaars, T. (2021). A continual learning survey: defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Ebrahimi, S., Elhoseiny, M., Darrell, T., and Rohrbach, M. (2020). Uncertainty-guided continual learning with Bayesian neural networks. In *International Conference on Learning Representations*.

Farquhar, S. and Gal, Y. (2018). Towards robust evaluations of continual learning. *ICML Workshop on Lifelong Learning: A Reinforcement Learning Approach*.

Ghahramani, Z. and Attias, H. (2000). Online variational Bayesian learning. In *NIPS Workshop on Online Learning*.

Honkela, A. and Valpola, H. (2003). On-line variational Bayesian learning. In *International Symposium on Independent Component Analysis and Blind Signal Separation*.

Jung, H., Ju, J., Jung, M., and Kim, J. (2018). Less-forgetful learning for domain expansion in deep neural networks. In *AAAI Conference on Artificial Intelligence*.

Kapoor, S., Karaletsos, T., and Bui, T. D. (2021). Variational auto-regressive Gaussian processes for continual learning. In *International Conference on Machine Learning*.

Kessler, S., Nguyen, V., Zohren, S., and Roberts, S. (2019). Hierarchical Indian buffet neural networks for Bayesian continual learning. *arXiv*.

Kim, H.-E., Kim, S., and Lee, J. (2018). Keep and learn: continual learning by constraining the latent space for knowledge preservation in neural networks. In *Medical Image Computing and Computer Assisted Intervention*.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N. C., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., and Hadsell, R. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*.

Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. (2015). Human-level concept learning through probabilistic program induction. *Science*.

Lee, S.-W., Kim, J.-H., Jun, J., Ha, J.-W., and Zhang, B.-T. (2017). Overcoming catastrophic forgetting by incremental moment matching. In *Advances in Neural Information Processing Systems*.

Li, Z. and Hoiem, D. (2018). Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Liu, X., Masana, M., Herranz, L., van de Weijer, J., López, A. M., and Bagdanov, A. D. (2018). Rotate your networks: better weight consolidation and less catastrophic forgetting. *International Conference on Pattern Recognition*.

Loo, N., Swaroop, S., and Turner, R. E. (2020). Generalized variational continual learning. In *International Conference on Learning Representations*.

Lopez-Paz, D. and Ranzato, M. A. (2017). Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*.

Matthews, A. G. d. G., Hensman, J., Turner, R., and Ghahramani, Z. (2016). On sparse variational methods and the Kullback-Leibler divergence between stochastic processes. In *International Conference on Artificial Intelligence and Statistics*.

Moreno-Muñoz, P., Artés-Rodríguez, A., and Álvarez, M. A. (2019). Continual multi-task Gaussian processes. *arXiv*.

Nguyen, C. V., Li, Y., Bui, T. D., and Turner, R. E. (2018). Variational continual learning. In *International Conference on Learning Representations*.

Pan, P., Swaroop, S., Immer, A., Eschenhagen, R., Turner, R., and Khan, M. E. E. (2020). Continual deep learning by functional regularisation of memorable past. In *Advances in Neural Information Processing Systems*.

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. (2019). Continual lifelong learning with neural networks: a review. *Neural Networks*.

Park, D., Hong, S., Han, B., and Lee, K. M. (2019). Continual learning by asymmetric loss approximation with single-side overestimation. In *International Conference on Computer Vision*.

Ritter, H., Botev, A., and Barber, D. (2018). Online structured Laplace approximations for overcoming catastrophic forgetting. In *Advances in Neural Information Processing Systems*.

Rudner, T. G. J., Chen, Z., and Gal, Y. (2021). Rethinking function-space variational inference in Bayesian neural networks. In *Symposium on Advances in Approximate Bayesian Inference*.

Sato, M.-A. (2001). Online model selection based on the variational Bayes. *Neural Computation*.

Schwarz, J., Czarnecki, W., Luketina, J., Grabska-Barwinska, A., Teh, Y. W., Pascanu, R., and Hadsell, R. (2018). Progress & compress: a scalable framework for continual learning. In *International Conference on Machine Learning*.

Shannon, C. E. and Weaver, W. (1949). *The Mathematical Theory of Communication*. University of Illinois Press, Urbana and Chicago.

Sun, S., Zhang, G., Shi, J., and Grosse, R. (2019). Functional variational Bayesian neural networks. In *International Conference on Learning Representations*.

Swaroop, S., Nguyen, C. V., Bui, T. D., and Turner, R. E. (2019). Improving and understanding variational continual learning. In *NeurIPS Workshop on Continual Learning*.

Titsias, M. K., Schwarz, J., de G. Matthews, A. G., Pascanu, R., and Teh, Y. W. (2020). Functional regularisation for continual learning with Gaussian processes. In *International Conference on Learning Representations*.

Yang, L., Wang, K., and Mihaylova, L. S. (2019). Online sparse multi-output Gaussian process regression and learning. *IEEE Transactions on Signal and Information Processing over Networks*.

Yin, D., Farajtabar, M., and Li, A. (2020a). SOLA: continual learning with second-order loss approximation. *arXiv*.

Yin, D., Farajtabar, M., Li, A., Levine, N., and Mott, A. (2020b). Optimization and generalization of regularization-based continual learning: a loss approximation viewpoint. *arXiv*.

Zenke, F., Poole, B., and Ganguli, S. (2017). Continual learning through synaptic intelligence. In *International Conference on Machine Learning*.

Supplementary Material

Table of Contents

A. Proofs

A.1. Variational Objective

Proposition 1 (Sequential Function-Space Variational Inference (S-FSVI); adapted from (Rudner et al., 2021)). *Let D_t be the number of model output dimensions for t tasks, let $f : \mathcal{X} \to \mathbb{R}^P \to \mathbb{R}^{D_t}$ be a mapping defined by a neural network architecture, let $\boldsymbol{\Theta} \sim \mathbb{R}^P$ be a multivariate random vector of network parameters, and let $q_t(\boldsymbol{\theta}) \doteq \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$ and $q_{t-1}(\boldsymbol{\theta}) \doteq \mathcal{N}(\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1})$ be variational distributions over $\boldsymbol{\Theta}$. Additionally, let $\mathbf{X}_C$ denote a sample of context points, and let $\bar{\mathbf{X}}_t \sim \{f(\mathbf{X}_t) \mid \mathbf{X}_C\}$. Under a diagonal approximation of the prior and variational posterior covariance functions across output dimensions, the objective in Equation (5) can be approximated by*

$$\begin{aligned} &\mathcal{F}(q_t, q_{t-1}, \mathbf{X}_C, \mathbf{X}_t, \mathbf{y}_t) \\ &\doteq \mathbb{E}_{q_t(\boldsymbol{\theta})}[\log p(\mathbf{y}_t \mid f(\mathbf{X}_t; \boldsymbol{\theta}))] \\ &\qquad \sum_{k=1}^{D_t} \frac{1}{2} \left(\log \frac{|[\mathbf{K}^{p_t}]_k|}{|[\mathbf{K}^{q_t}]_k|} - \frac{|\bar{\mathbf{X}}_t|}{D_t} + \mathrm{Tr}([\mathbf{K}^{p_t}]_k^{-1}[\mathbf{K}^{q_t}]_k) + \Delta(\bar{\mathbf{X}}_t; \boldsymbol{\mu}_t, \boldsymbol{\mu}_{t-1})^\top [\mathbf{K}^{p_t}]_k^{-1} \Delta(\bar{\mathbf{X}}_t; \boldsymbol{\mu}_t, \boldsymbol{\mu}_{t-1}) \right), \end{aligned} \tag{A.1}$$

where

$$\Delta(\bar{\mathbf{X}}_t; \boldsymbol{\mu}_t, \boldsymbol{\mu}_{t-1}) \doteq [f(\bar{\mathbf{X}}_t; \boldsymbol{\mu}_t)]_k - [f(\bar{\mathbf{X}}_t; \boldsymbol{\mu}_{t-1})]_k \tag{A.2}$$

and

$$\mathbf{K}^{p_t} \doteq \mathcal{J}(\bar{\mathbf{X}}_t, \boldsymbol{\mu}_{t-1}) \boldsymbol{\Sigma}_{t-1} \mathcal{J}(\bar{\mathbf{X}}_t, \boldsymbol{\mu}_{t-1})^\top \quad \text{and} \quad \mathbf{K}^{q_t} \doteq \mathcal{J}(\bar{\mathbf{X}}_t, \boldsymbol{\mu}_t) \boldsymbol{\Sigma}_t \mathcal{J}(\bar{\mathbf{X}}_t, \boldsymbol{\mu}_t)^\top, \tag{A.3}$$

are covariance matrix estimates constructed from Jacobians $\mathcal{J}(\cdot, \mathbf{m}) \doteq \frac{\partial f(\cdot; \boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}}|_{\boldsymbol{\Theta}=\mathbf{m}}$ with $\mathbf{m} = \{\boldsymbol{\mu}_t, \boldsymbol{\mu}_{t-1}\}$.

Proof. The results follows directly from the variational objective derived in (Rudner et al., 2021) when setting the prior to $p \doteq q_{t-1}$ and specifying the context set to be constructed from the coreset. □

A.2. Derivation of Correspondence to Other Function-Space Objectives

Proposition 2 (Relationship between FROMP and S-FSVI). *With the S-FSVI objective $\mathcal{F}$ defined as in Equation (6), let $\bar{\mathbf{X}}_t = \mathbf{X}_C$. Then, up to a multiplicative constant, the FROMP objective corresponds to the S-FSVI objective with the prior covariance given by a Laplace approximation about $\boldsymbol{\mu}_{t-1}$ and the variational distribution given by a Dirac delta distribution $q_t^{\text{FROMP}}(\boldsymbol{\theta}) \doteq \delta(\boldsymbol{\theta} - \boldsymbol{\mu}_t)$. Denoting the prior covariance under a Laplace approximation about $\boldsymbol{\mu}_{t-1}$ by $\hat{\boldsymbol{\Sigma}}_0(\boldsymbol{\mu}_{t-1})$ so that $q_{t-1}^{\text{FROMP}}(\boldsymbol{\theta}) \doteq \mathcal{N}(\boldsymbol{\mu}_{t-1}, \hat{\boldsymbol{\Sigma}}_0(\boldsymbol{\mu}_{t-1}))$, the FROMP objective can be expressed as*

$$\mathcal{L}^{\text{FROMP}}(q_t^{\text{FROMP}}, q_{t-1}^{\text{FROMP}}, \mathbf{X}_C, \mathbf{X}_t, \mathbf{y}_t) = \mathcal{F}(q_t^{\text{FROMP}}, q_{t-1}^{\text{FROMP}}, \mathbf{X}_C, \mathbf{X}_t, \mathbf{y}_t) - \mathcal{V},$$

where

$$\mathcal{V} \doteq \frac{1}{2} \sum_k \left(\log \frac{[\bar{\mathbf{K}}^{\hat{p}_t}]_k}{[\bar{\mathbf{K}}^{q_t}]_k} + \frac{[\bar{\mathbf{K}}^{q_t}]_k}{[\bar{\mathbf{K}}^{\hat{p}_t}]_k} - 1 \right),$$

with $\bar{\mathbf{K}}$ denoting a covariance matrix under a block-diagonalization without inter-task dependence, and

$$\bar{\mathbf{K}}^{\hat{p}_t} \doteq \text{block-diag}\left(\mathcal{J}(\bar{\mathbf{X}}_t, \boldsymbol{\mu}_{t-1}) \hat{\boldsymbol{\Sigma}}_0(\boldsymbol{\mu}_{t-1}) \mathcal{J}(\bar{\mathbf{X}}_t, \boldsymbol{\mu}_{t-1})^\top \right).$$

Proof. By Equation (8) in Pan et al. (2020), the FROMP objective function is given by

$$\begin{aligned}&L^{\text{FROMP}}(q_t^{\text{FROMP}}, q_{t\ 1}^{\text{FROMP}}, \mathbf{X}_C, \mathbf{X}_t, \mathbf{y}_t)\\ &\doteq \mathsf{E}_{q_t(\boldsymbol{\theta})}[\log p(\mathbf{y}_t \mid f(\mathbf{X}_t;\boldsymbol{\mu}_t))] + \sum_{k=1}^{t\ 1} \frac{\tau}{2} \left([f(\mathbf{X}_C;\boldsymbol{\mu}_t)]_k \quad [f(\mathbf{X}_C;\boldsymbol{\mu}_{t\ 1})]_k\right)^\top [\mathbf{K}^{\hat{p}_t}]_k^{\ 1} \left([f(\mathbf{X}_C;\boldsymbol{\mu}_t)]_k \quad [f(\mathbf{X}_C;\boldsymbol{\mu}_{t\ 1})]_k\right),\end{aligned} \tag{A.4}$$

with temperature parameter τ. The result follows directly from the definition of $\mathcal{F}(q_t^{\text{FROMP}}, q_{t\ 1}^{\text{FROMP}}, \mathbf{X}_C, \mathbf{X}_t, \mathbf{y}_t)$ and $\tau = 1$. □

Proposition 3 (Relationship between FRCL and S-FSVI)**.** *With the* S-FSVI *objective* $\mathcal{F}$ *defined as in Equation (6), let* $\bar{\mathbf{X}}_t = \mathbf{X}_C$*, and let* $f^{\text{LM}}(\ ;\boldsymbol{\Theta}) \doteq \Phi_\psi(\ )\boldsymbol{\Theta}$ *be a Bayesian linear model, where* $\Phi_\psi(\ )$ *is a deterministic feature map parameterized by* ψ*. Then the* FRCL *objective corresponds to the* S-FSVI *objective for the model* $f^{\text{LM}}(\ ;\boldsymbol{\Theta})$ *plus an additional weight-space KL divergence penalty. That is, for* $p_t(\boldsymbol{\theta}) \doteq \mathcal{N}(\mathbf{0}, \mathbf{I})$*, and* $q_t(\boldsymbol{\theta}) \doteq \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$*,*

$$L^{\text{FRCL}}(q_t^{\text{FRCL}}, q_{t\ 1}^{\text{FRCL}}, \mathbf{X}_C, \mathbf{X}_t, \mathbf{y}_t) = \mathcal{F}(q_t^{\text{FRCL}}, q_{t\ 1}^{\text{FRCL}}, \mathbf{X}_C, \mathbf{X}_t, \mathbf{y}_t) + \mathsf{D}_{\text{KL}}(q_t(\boldsymbol{\theta}) \,\|\, p_t(\boldsymbol{\theta})). \tag{A.5}$$

Proof. By Section 2.3 in Titsias et al. (2020), the FRCL objective function is given by

$$\begin{aligned}L^{\text{FRCL}}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t, \mathbf{X}_C, \mathbf{X}_t, \mathbf{y}_t) \doteq\ & \mathsf{E}_{q_t(\boldsymbol{\theta})}[\log p(\mathbf{y}_t \mid \Phi_\psi(\mathbf{X}_t)\boldsymbol{\theta})] \quad \mathsf{D}_{\text{KL}}(q_t(\boldsymbol{\theta}) \,\|\, p_t(\boldsymbol{\theta}))\\ & \mathsf{D}_{\text{KL}}(\tilde{q}_t(\tilde{f}(\mathbf{X}_{C_t};\boldsymbol{\theta})) \,\|\, \tilde{p}_t(\tilde{f}(\mathbf{X}_{C_t};\boldsymbol{\theta}))) \quad \sum_{k=1}^{t\ 1} \mathsf{D}_{\text{KL}}(?\ (\tilde{q}_k(\tilde{f}(\mathbf{X}_{C_k};\boldsymbol{\theta}))) \,\|\, \tilde{p}_k(\tilde{f}(\mathbf{X}_{C_k};\boldsymbol{\theta}))),\end{aligned} \tag{A.6}$$

with the inducing points associated with task k denoted by $\mathbf{X}_{C_k}$ and $?$ denoting the `stop-gradient` operator, whereas the S-FSVI objective for a Bayesian linear model is

$$\begin{aligned}&\mathcal{F}(q_t, q_{t\ 1}, \mathbf{X}_C, \mathbf{X}_t, \mathbf{y}_t) \doteq \mathsf{E}_{q_t(\boldsymbol{\theta})}[\log p(\mathbf{y}_t \mid \Phi_\psi(\mathbf{X}_t)\boldsymbol{\theta})]\\ &\quad \sum_{k=1}^{D_t} \frac{1}{2}\left(\log \frac{|[\mathbf{K}^{p_t}]_k|}{|[\mathbf{K}^{q_t}]_k|} \quad \frac{|\bar{\mathbf{X}}_t|}{D_t} + \mathrm{Tr}([\mathbf{K}^{p_t}]_k^{\ 1}[\mathbf{K}^{q_t}]_k) + \Delta(\bar{\mathbf{X}}_t;\boldsymbol{\mu}_t,\boldsymbol{\mu}_{t\ 1})^\top [\mathbf{K}^{p_t}]_k^{\ 1} \Delta(\bar{\mathbf{X}}_t;\boldsymbol{\mu}_t,\boldsymbol{\mu}_{t\ 1})\right),\end{aligned} \tag{A.7}$$

with

$$\Delta(\bar{\mathbf{X}}_t;\boldsymbol{\mu}_t,\boldsymbol{\mu}_{t\ 1}) \doteq [\Phi_\psi(\bar{\mathbf{X}}_t)\boldsymbol{\mu}_t]_k \quad [\Phi_\psi(\bar{\mathbf{X}}_t)\boldsymbol{\mu}_{t\ 1}]_k \tag{A.8}$$

and

$$\mathbf{K}^{p_t} \doteq \Phi_\psi(\bar{\mathbf{X}}_t)\boldsymbol{\Sigma}_{t\ 1}\Phi_\psi(\bar{\mathbf{X}}_t)^\top \qquad \mathbf{K}^{q_t} \doteq \Phi_\psi(\bar{\mathbf{X}}_t)\boldsymbol{\Sigma}_t\Phi_\psi(\bar{\mathbf{X}}_t)^\top. \tag{A.9}$$

Letting $\mathbf{X}_{C_k}$ be the context points associated with task k and letting $\bar{\mathbf{K}}$ denote a covariance matrix under a block-diagonalization without inter-task dependence, we define

$$\bar{\mathbf{K}}^{p_t} \doteq \text{block-diag}\left(\Phi_\psi(\bar{\mathbf{X}}_t)\boldsymbol{\Sigma}_{t\ 1}\Phi_\psi(\bar{\mathbf{X}}_t)^\top\right) \qquad \bar{\mathbf{K}}^{q_t} \doteq \text{block-diag}\left(\Phi_\psi(\bar{\mathbf{X}}_t)\boldsymbol{\Sigma}_t\Phi_\psi(\bar{\mathbf{X}}_t)^\top\right), \tag{A.10}$$

with diagonal entries $\{\mathbf{K}_1^{p_t}, \ldots, \mathbf{K}_t^{p_t}\}$ and $\{\mathbf{K}_1^{q_t}, \ldots, \mathbf{K}_t^{q_t}\}$, respectively, where each $\mathbf{K}_k^{p_t}$ is computed from task-specific context points $\mathbf{X}_{C_k}$. Fixing $[\boldsymbol{\mu}_{t\ 1}]_k = \mathbf{0}$ and $[\boldsymbol{\Sigma}_{t\ 1}]_k = \mathbf{I}_{M_k}$ for all $k \quad t$ with $M_k = |\mathbf{X}_{C_k}|$, as in Titsias et al. (2020), we then get

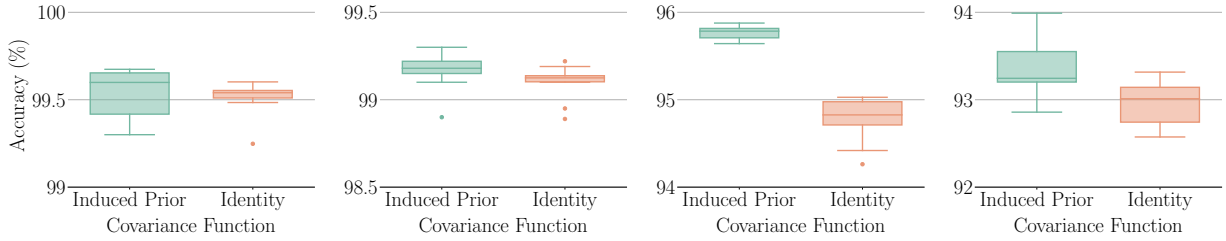
$$\bar{\mathbf{K}}_k^{p_t} = \Phi_\psi(\bar{\mathbf{X}}_{C_k})\Phi_\psi(\bar{\mathbf{X}}_{C_k})^\top \quad \forall k \quad t. \tag{A.11}$$

Considering $[\boldsymbol{\mu}_t]_k$ and $[\boldsymbol{\Sigma}_t]_k$ as fixed for all $k \quad t \quad 1$, as in Titsias et al. (2020), using the `stop-gradient` operator $?$, we can write the S-FSVI objective as

$$\begin{aligned}&\mathcal{F}(q_t, q_{t\ 1}, \mathbf{X}_C, \mathbf{X}_t, \mathbf{y}_t) \doteq \mathsf{E}_{q_t(\boldsymbol{\theta})}[\log p(\mathbf{y}_t \mid \Phi_\psi(\mathbf{X}_t)\boldsymbol{\theta})]\\ &\quad \mathsf{D}_{\text{KL}}(\tilde{q}_t(\tilde{f}(\mathbf{X}_{C_t};\boldsymbol{\theta})) \,\|\, \tilde{p}_t(\tilde{f}(\mathbf{X}_{C_t};\boldsymbol{\theta}))) \quad \sum_{k=1}^{t\ 1} \mathsf{D}_{\text{KL}}(?\ (\tilde{q}_k(\tilde{f}(\mathbf{X}_{C_k};\boldsymbol{\theta}))) \,\|\, \tilde{p}_k(\tilde{f}(\mathbf{X}_{C_k};\boldsymbol{\theta}))),\end{aligned} \tag{A.12}$$

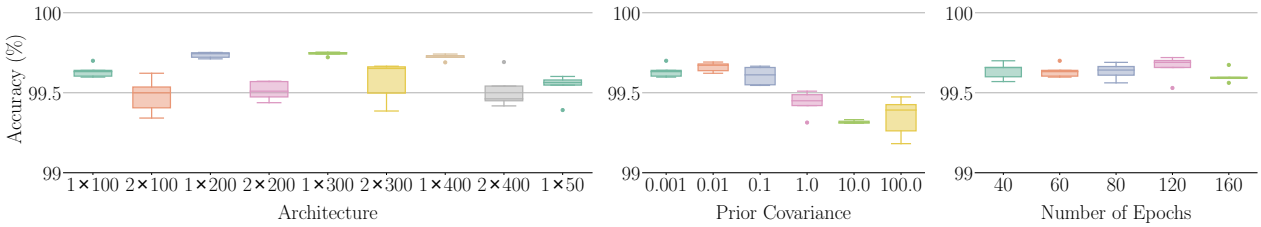
concluding the proof. □

B. Further Empirical Results

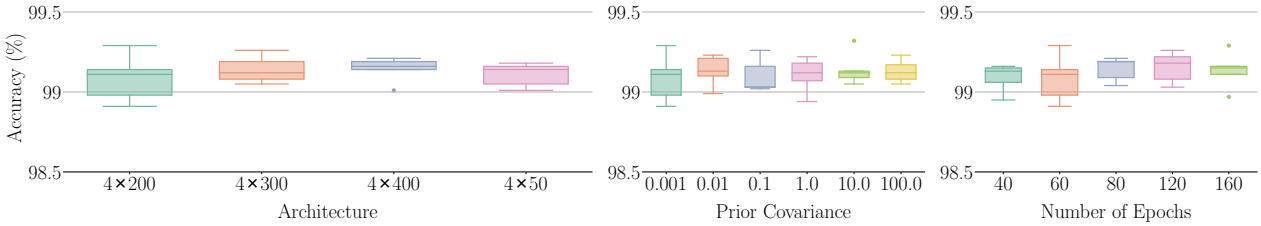


(a) S-MNIST (MH) **(b)** S-FMNIST (MH) **(c)** P-MNIST (SH) **(d)** S-MNIST (SH)

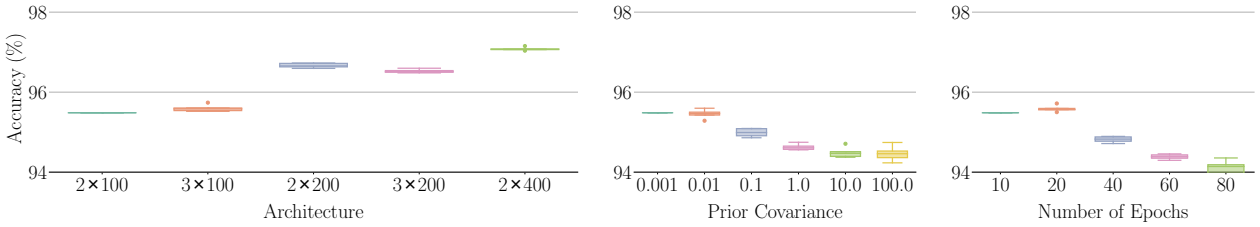
Figure 8. **Effect of Empirical Prior Covariance.** Comparison of predictive performance under the induced prior covariance function $\mathbf{K}^{p_t} = \text{diag}\left(J_{\boldsymbol{\mu}_{t-1}}(\mathbf{x})\boldsymbol{\Sigma}_t \; {}_1J_{\boldsymbol{\mu}_{t-1}}(\mathbf{x}^0)^\top\right)$ (left) vs. an identity covariance function (right).



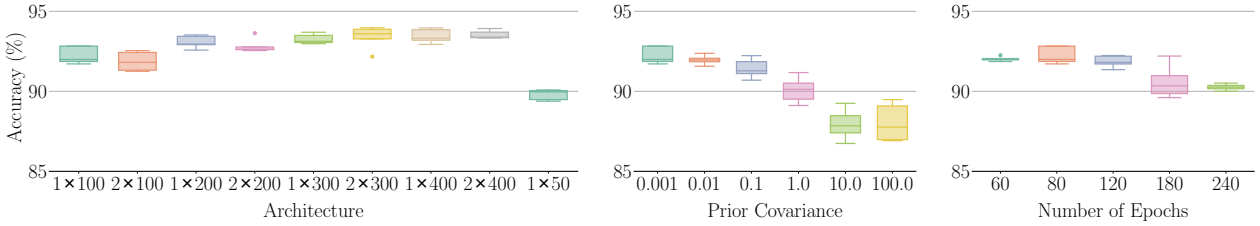
(a) S-MNIST (MH)



(b) S-FMNIST (MH)

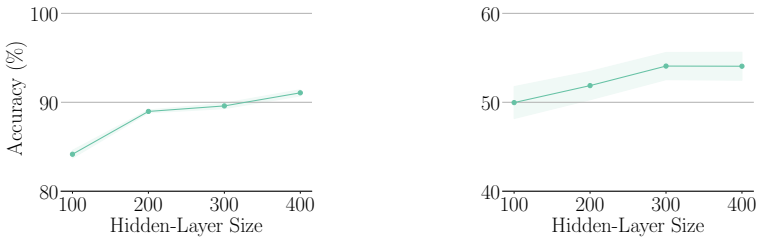


(c) P-MNIST (SH)



(d) S-MNIST (SH)

Figure 9. **Effect of Neural-Network Size, First-Task Prior Covariance, and the Number of Training Epochs.** We explore settings of neural-network size (e.g., 2 100 means a fully connected neural network with two hidden layers of size 100), initial prior covariance and number of training epochs for each task. To limit the computational resources required, we vary the values of one hyperparameter at a time instead of carrying out a full grid search.



(a) Permuted MNIST (SH) **(b)** Split MNIST (SH)

Figure 10. **Effect of Neural-Network Size under Minimal Coresets.** Predictive accuracy under S-FSVI on permuted MNIST (SH) and split MNIST (SH) as a function of network width, using only a minimal coreset of one sample per class, selected randomly.

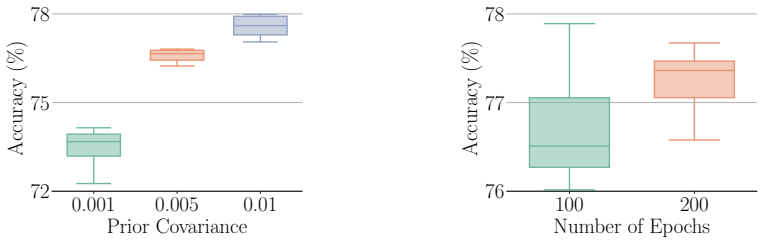


Figure 11. **Hyperparameter Search on Split CIFAR.** We explore settings of the initial first-task prior covariance and the number of epochs for the first task. To limit the computational resources required, we vary the values of one hyperparameter at a time instead of carrying out a full grid search.

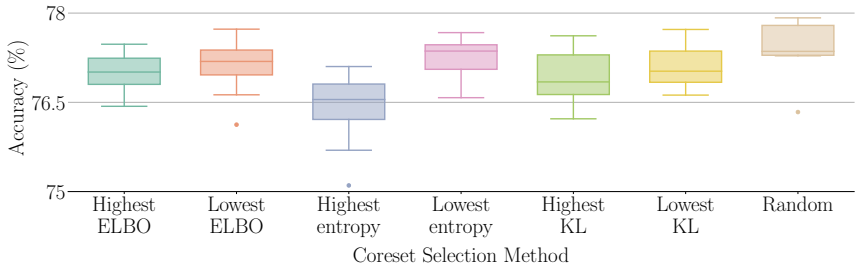


Figure 12. **Comparison of Different Coreset-Selection Methods on Split CIFAR.** For score-based coreset-selection methods, we first score each coreset point—using Equation (11) for ELBO scoring, using the predictive entropy for entropy scoring, and the KL divergence in Equation (11) for KL scoring—then sample context points from the coreset according to the probability mass function defined in Equation (C.13).

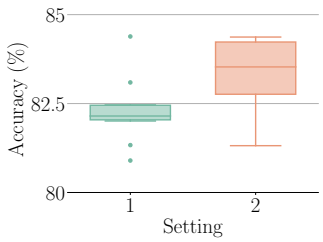


Figure 13. **Hyperparameter Search on Sequential Omniglot.** We compare two settings. In the first, we always sample one context point for each previous task from the context set at each gradient step. In the second, we sample a larger number of context points (with a budget of 60 samples per gradient step) from the context set when learning on the first 25 tasks.

C. Experimental Details

Our empirical evaluation centers around six sequences of classification tasks: a synthetic sequence of binary-classification tasks with 2D inputs; split MNIST; split Fashion MNIST; permuted MNIST; split CIFAR; and sequential Omniglot. With the exception of permuted MNIST, each of these task sequences can be tackled by a neural network with either a multi-head setup (MH) or a single-head setup (SH). In a multi-head setup, the neural network has a separate output layer (or head) for each task, and task identifiers are provided at test time in order to select the appropriate head. In a single-head setup, the neural network has just one output layer shared across all tasks, and task identifiers are not provided. In our experiments, we use multi-head setups for split Fashion MNIST, split CIFAR and sequential Omniglot, and single-head setups for the synthetic task sequence along with permuted MNIST. For split MNIST, we run both setups.

C.1. Illustrative Example

The task sequence shown in Figure 2 was created by Pan et al. (2020). Each of the five tasks in this sequence involves binary classification on 2D inputs, where the number of training examples per task is 3,600. Following Pan et al. (2020), we use a fully connected neural network with an input layer of size 2, two hidden layers of size 20 and an output layer of size 2. When running S-FSVI, we set the prior covariance as $\mathbf{\Sigma}_0 = 0.1$ and train the neural network for 250 epochs on each task. We use the Adam optimizer with an initial learning rate of 0.0005 ($\beta_1 = 0.9, \beta_2 = 0.999$) and a batch size of 128. The coreset is constructed by choosing 40 samples from the training data for each task. To evaluate the KL divergence between the posterior and the prior distributions over functions, for each previous task we sample 20 input points from the context set and generate another 30 samples by sampling each pixel uniformly from the range $[\ \ 4, 4]$. For example, when we train the model on task t *2* *f*$1, 2, 3, \ldots$*g*, we use $20(t \quad 1)$ samples chosen from the context set and $30t$ white-noise samples. The noise samples encourage the neural network to preserve high predictive uncertainty in regions far from the training data.

C.2. Task Sequences Based on (Fashion) MNIST

Split MNIST consists of five tasks, where each task is binary classification on a pair of MNIST classes. Split Fashion MNIST has the same form but uses data from Fashion MNIST. Permuted MNIST comprises ten tasks, where each task involves classifying images into the ten MNIST classes after the image pixels have been randomly reordered. Unless specified otherwise, the following setups apply to Figures 3, 6, 7 and 8 and Table 1.

Dataset. In all cases, 60,000 data samples are used for training and 10,000 data samples are used for testing. The input images are converted to floating-point numbers with values in the range $[0, 1]$.

Neural-Network Size & Coreset Size. To ensure fair comparison, all methods in Table 1 (unless where explicitly indicated otherwise) use the same neural-network size and (where applicable) coreset size. As in prior work (Pan et al., 2020; Titsias et al., 2020), we use fully connected neural networks, with two hidden layers of size 100 for permuted MNIST and two hidden layers of size 256 for split (Fashion) MNIST. In all cases, the ReLU activation function is applied to non-output units. For single-head setups, we use 200 coreset points; for multi-head setups, we use 40 points.

Coreset Selection. For S-FSVI with a coreset, when training on the first task, 40 context points are generated by sampling each pixel uniformly from the range $[0, 1]$; during training on subsequent tasks, 40 context points are chosen randomly from the context set. For S-FSVI without a coreset, 40 context points are chosen uniformly randomly from the training data of the current task (corresponding to the "Random" label in Figure 3).

Prior Distribution. For the first task, S-FSVI uses a prior distribution over functions with fixed mean and diagonal covariance. When using a coreset, the prior distribution is assumed to be Gaussian with zero mean and a diagonal covariance of magnitude 0.001. When not using a coreset, the prior distribution is assumed to be Gaussian with zero mean and a diagonal covariance of magnitude 100. The prior variance is optimized via hyperparameter selection on a validation set.

Optimization. We use the Adam optimizer with an initial learning rate of 0.0005 ($\beta_1 = 0.9, \beta_2 = 0.999$). The number of epochs on each task is 60 for split MNIST (MH), 60 for split Fashion MNIST (MH), 10 for permuted MNIST (SH) and 80 for split MNIST (SH). The batch size is 128.

Prediction. The predictive distribution used for computing the expected log-likelihood is estimated using five Monte Carlo samples.

Hyperparameter Selection. For "S-FSVI (optimized)" in Table 1, we used the optimized hyperparameters chosen on a

Table 4. Hyperparameter selection. Optimal values (in bold) were chosen based on validation-set accuracy. Standard errors were computed across ten random seeds.

Task Sequences	Number of Layers & Units	Magnitude of Prior Variance	Number of Epochs
Split MNIST (MH)	{**1**, 2} * {100, 200, 300, **400**}	{**0.001**, 0.01, 0.1, 1, 10, 100}	{40, **60**, 80, 120, 160}
Split Fashion MNIST (MH)	{**4**} * {50, **200**, 300, 400}	{**0.001**, 0.01, 0.1, 1, 10, 100}	{40, **60**, 80, 120, 160}
Permuted MNIST (SH)	{**2**} * {100, 200, 400, **500**}	{**0.001**, 0.01, 0.1, 1, 10, 100}	{10, **20**, 40, 60, 80}
Split MNIST (SH)	{**1**, 2} * {100, 200, 300, **400**}	{**0.001**, 0.01, 0.1, 1, 10, 100}	{60, **80**, 120, 160, 240}

validation set after exploring the configurations shown in Table 4. For cases where no configuration is significantly better than the rest, the default value given in Appendix C.2 is used.

C.3. Split CIFAR

Split CIFAR, as described in Pan et al. (2020), consists of six tasks. The first is ten-way classification on the full CIFAR-10 dataset. Each of the following five is also ten-way classification, with classes drawn from CIFAR-100. Following Pan et al. (2020), we use a neural network with four convolutional layers followed by two fully connected layers followed by multiple output heads (one for each task). For S-FSVI, we use the following setup: Adam optimizer with learning rate 0.0005, prior with covariance 0.01, random coreset selection, 200 coreset points per task, 50 context points at each task. We also use this setup (and a training duration of 2000 epochs) when training individual neural networks for the "separate" baseline.

C.4. Sequential Omniglot

Sequential Omniglot, as described in Schwarz et al. (2018), comprises 50 classification tasks. Each task is associated with an alphabet, and the number of characters (classes) varies between alphabets. Following Schwarz et al. (2018), we use a neural network with four convolutional layers followed by one fully connected layer. For S-FSVI, we use two coreset points per character, as used by Titsias et al. (2020). The coreset points are sampled from the training set with probability proportional to the entropy of the neural network's posterior predictive distribution. To limit memory usage, we draw no more than 25 context points from the context set at each gradient step after task 25. We use a learning rate of 0.001 and a prior covariance of 1.0. For the first task, the neural network trains for 200 epochs; for subsequent tasks, it trains for ten epochs per task. We use the same data augmentation and train-test split as Titsias et al. (2020).

C.5. Coreset-Selection Methods

We consider different distributions from which to sample points to be added to the coreset. For each of the scoring methods below, we use the scores to create a probability mass function from which points can be sampled.

Random. Points are sampled uniformly from the training data.

Predictive-Entropy Scoring. Points are scored according to the total predictive uncertainty (i.e., the predictive entropy) of the model. For a model with stochastic parameters $\mathbf{\Theta}$, pre-likelihood outputs $f(\mathbf{X}; \boldsymbol{\theta})$, and a likelihood function $p(\mathbf{y} \mid f(\mathbf{X}; \boldsymbol{\theta}))$, the predictive entropy is given by $\mathsf{H}(\mathsf{E}[p(\mathbf{y} \mid f(\mathbf{X}; \boldsymbol{\theta}))])$ (Cover and Thomas, 1991; Shannon and Weaver, 1949). The expectation is taken with respect to the model parameters. $\mathsf{H}(\,)$ is the entropy functional, and $\mathsf{I}(\mathbf{y}\ ;\ \mathbf{\Theta})$ is the mutual information between the model parameters and its predictions.

Evidence-Lower-Bound Scoring. Points are scored according to the value of the evidence lower bound (ELBO) given in Equation (11).

Kullback-Leibler-Divergence Scoring. Points are scored according to the value of the approximation to the function-space KL divergence given in Equation (11).

Score-Based Distributions. After scoring with the above methods, points are added to the coreset by sampling from one of the following probability mass functions:

$$\textbf{Lowest:} \quad \mathsf{P}(i) \doteq \frac{\bar{s}_i}{\sum_{j=1}^{N} \bar{s}_j} \qquad \text{and} \qquad \textbf{Highest:} \quad \mathsf{P}(i) \doteq \frac{s_i}{\sum_{j=1}^{N} s_j}, \tag{C.13}$$

where s_i is the score of i-th point, $\bar{s}_i = \max_{j=1}^{N} s_j \quad s_i$, and N is the number of candidate points.

C.6. Forward and Backward Transfer

In Table 3, we report forward and backward transfer metrics as defined in Pan et al. (2020). Backward transfer (BT) indicates the performance gain on past tasks when new tasks are learnt, while forward transfer (FT) quantifies how much knowledge from past tasks helps the learning of new tasks. Higher is better for both. For T tasks, let $R_{i,i}$ be the accuracy of model on task t_i after training on task t_i, and let R_i^{ind} be the accuracy of an independent model trained only on task t_i. Then

$$\text{BT} \doteq \frac{1}{T-1}\sum_{i=1}^{T-1} R_{T,i} - R_{i,i} \qquad \text{and} \qquad \text{FT} \doteq \frac{1}{T-1}\sum_{i=2}^{T} R_{i,i} - R_i^{\text{ind}}.$$

D. Further Related Work

Objective-based approaches to continual learning involve training a neural network using a specially designed objective function. Typically the objective includes a regularization term that penalizes changes in the neural network's configuration. Whereas in Section 4 we summarise methods that regularize in function space, here we cover methods that regularize directly in terms of the parameters of a neural network. Among these, most relevant to our work are those that approximate Bayesian updating, in which the posterior from the previous task forms the prior for the current task.

A key idea is shared between many methods for parameter-space regularization: for each parameter, apply a penalty on the difference between its current setting and its prior setting, weighted by a measure of the parameter's importance. Methods vary in how they measure importance. Variational continual learning (VCL; Nguyen et al., 2018; Swaroop et al., 2019), which extends the concept of online variational inference (Broderick et al., 2013; Ghahramani and Attias, 2000; Honkela and Valpola, 2003; Sato, 2001) to deep neural networks, uses the parameter covariance matrix of the model currently serving as the prior. Elastic weight consolidation (EWC; Kirkpatrick et al., 2017) and its successors (Chaudhry et al., 2018; Lee et al., 2017; Liu et al., 2018; Schwarz et al., 2018) use a Fisher information matrix computed on each task. Online structured Laplace (Ritter et al., 2018) and second-order loss approximation (Yin et al., 2020a) respectively use Kronecker-factored and low-rank Hessians. Synaptic intelligence (SI; Zenke et al., 2017) uses a cumulative sum of the gradient of the training objective with respect to the parameters. Memory-aware synapses (MAS; Aljundi et al., 2018) use the gradient of the model output with respect to the parameters.

Other related work on parameter-space regularization includes various modifications to VCL (Ahn et al., 2019; Kessler et al., 2019), uncertainty-guided continual learning in Bayesian neural networks (Ebrahimi et al., 2020), and a variation of SI known as asymmetric loss approximation with single-side overestimation (Park et al., 2019). There have also been efforts to conceptually unify some of the approaches outlined above: Loo et al. (2020) draws a link between VCL and online EWC; Chaudhry et al. (2018) combines EWC and SI in a single method; Yin et al. (2020b) generalizes EWC, online structured Laplace, SI and MAS.