Variational Continual Learning for Regression with Exponential Prior

Candidate number: 1088143

1 Introduction

Real-world environments are dynamic, which necessitate intelligent systems to continually adapt to evolving environments. This motivated the study of *continual learning*, which is characterised by learning from a continuous stream of data while retaining previously acquired knowledge. Formally, the goal in this setting is to learn the model parameters $\boldsymbol{\theta}$ from a set of sequentially arriving datasets $\mathcal{D}_t = \{\boldsymbol{x}^{(n)}, y_t^{(n)}\}_{n=1}^{N_t}$, where each $\mathcal{D}_t$ may stem from changing input domains or are associated with different tasks Nguyen et al. [2017]. A critical challenge in continual learning is *catastrophic forgetting*, where adaptation to a new task results in drastic performance degradation on the old tasks.

While continual learning has seen its successful application in many classification tasks (e.g., visual classification), it is much less developed for regression De Lange et al. [2021]. Moreover, most Bayesian continual learning methods rely on standard Gaussian priors over $\boldsymbol{\theta}$. Yet, depending on the application area (e.g., genomics, where datasets are very high-dimensional but the signal-to-noise ratio is often known), we may have different *prior* knowledge on the feature sparsity for different tasks, which we would like to encode into the prior Cui et al. [2022]. These gaps motivated us to propose two novel extensions to the seminal Variational Continual Learning (VCL) framework: First, we aim to evaluate the effectiveness of VCL on both classification tasks and their *regression* counterparts. Second, we empirically compare the influence of a heavy-tailed, sparsity-inducing prior, namely the *exponential* distribution, with the standard Gaussian prior over network parameters.

2 Related Work

Continual Learning. We refer the readers to Wang et al. [2024] for a comprehensive overview on the numerous approaches to continual learning, and focus our discussion on *regularisation-based* methods. Their main approach is to add regularisation terms to the loss function, penalising changes to parameters deemed important to previous tasks. This is naturally formulated in a Bayesian framework: At task t, the posterior $p(\boldsymbol{\theta} \mid \mathcal{D}_{1:t})$ encodes its task-specific knowledge. Then, new tasks are learned by treating the current posterior as a prior for subsequent parameter updates. However, exact posterior evaluation is often intractable, especially with Bayesian Neural Networks (BNNs). This necessitates approximations q_t for the true posterior. Two common posterior approximation strategies are online *Laplace approximation* (LA) and *Variational Inference* (VI). The former approximates $p(\boldsymbol{\theta} \mid \mathcal{D}_{1:t})$ as a multivariate Gaussian with local gradient information Kirkpatrick et al. [2017]. The latter focuses on minimising the KL divergence between the approximation family and the true posterior (see Lemma 2). It is used in VCL and other works including CLAW Adel et al. [2019], KCL Derakhshani et al. [2021], VAR-GP Kapoor et al. [2021], and S-FSVI Rudner et al. [2022].

Weight-space Priors in BNNs. Isotropic Gaussian distributions, which have covariance matrices of the form $\Sigma = \sigma^2 I$, have been the most widely adopted priors for BNNs despite their well-documented limitations Vladimirova et al. [2019], Wenzel et al. [2020]. MacKay et al. demonstrated that in the infinite limit, such networks converge to Gaussian Processes (GPs), undermining the intention of using BNNs for their greater expressivity[1] over GPs. Consequently, more flexible alternatives have been sought. Examples include heavy-tailed priors like Laplace priors Williams [1995], Student-t priors Neklyudov et al. [2018] and the sparsity-inducing horseshoe prior Carvalho et al. [2009].

[1] Here, the ability to approximate different distributions over the function space in their respective predictives.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

3 Variational Continual Learning

Following the aforementioned Bayesian framework for continual learning, Nguyen et al. proposed VCL, which integrates online VI Ghahramani and Attias [2000] and Monte Carlo VI Blundell et al. [2015] for BNNs, with a small episodic memory to mitigate catastrophic forgetting. Their approximate Bayesian inference procedure is as follows: We first set the prior $p(\boldsymbol{\theta})$. Then, the posterior distribution after seeing T datasets, $p(\boldsymbol{\theta} \mid \mathcal{D}_{1:T})$, is recursively recovered by applying Bayes rule, as shown in Lemma 1. Further, Nguyen et al. extended VCL by selecting a coreset C_t at each task t, which is a small representative subset of past data points used to "refresh" the model's memory on old tasks. Iterative parameter updates are then performed using the posterior from non-coreset data and the likelihood from the new coreset. These computations are formulated in Lemma 3. To approximate $p(\boldsymbol{\theta} \mid \mathcal{D}_{1:T})$, they use a multivariate Gaussian prior $q_0(\boldsymbol{\theta}) = p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}; 0, I)$ with Gaussian mean-field approximation $q_t(\boldsymbol{\theta}) = \prod_{d=1}^{D} \mathcal{N}(\theta_{t,d}; \mu_{t,d}, \sigma_{t,d}^2)$, where $D = \dim(\boldsymbol{\theta})$. This offers several algebraic conveniences such as an efficiently computable closed-form KL divergence.

4 Proposed Extensions

Reformulating classification as regression. To adapt the original VCL to regression, we treat one-hot encoded class labels as continuous multi-dimensional targets in $\mathbb{R}^C$, where C is the number of classes. Correspondingly, we replace the categorical likelihood with Gaussian likelihood. Thus, we now have $\boldsymbol{y} \sim p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{y}; f_{\boldsymbol{\theta}}(\boldsymbol{x}), \sigma_y^2 I)$, where $f_{\boldsymbol{\theta}}$ denotes the output function of the model parametrised by $\boldsymbol{\theta}$. We follow the standard assumption of constant observation noise variance σ_y^2 for all targets. This in turn changes the expected log-likelihood in the variational evidence lower bound, and thus the loss function for VCL (7). As shown in Lemma 5, this is equivalent to replacing the task-specific loss (induced by the expected log-likelihood) from cross-entropy to MSE loss.

Replacing Gaussian prior with Exponential prior. Compared to the standard Gaussian distribution (see Figure 2 in Section B.2), the exponential distribution imposes stronger penalisation on large weights (due to faster weight decay), but still permits a few large weights (due to heavier tails). This balances sparsity and flexibility in feature selection Fortuin [2022]. Formally, we let $q_0(\boldsymbol{\theta}) = p(\boldsymbol{\theta}) = \prod_{d=1}^{D} \text{Exp}(\theta_d; \lambda)$, where independence across the parameters is assumed. To reconcile this with our Gaussian mean-field approximation for posteriors, we estimate the prior's scale λ^{-1} by first-absolute-moment matching:

$$\widehat{\lambda^{-1}} = \mathbb{E}[|\mathcal{N}(\mu_0, \sigma_0^2)|] = \left[\sigma_0 \sqrt{\frac{2}{\pi}} \exp\left(-\frac{\mu_0^2}{2\sigma_0^2} \right) + \mu_0 \left(1 - 2\Phi\left(-\frac{\mu_0}{\sigma_0} \right) \right) \right] \tag{1}$$

where Φ is the standard normal CDF. As a shorthand notation[2], henceforth we write $\hat{\lambda} := (\widehat{\lambda^{-1}})^{-1}$. For $\mu_0 = 0, \sigma_0 = 1$, this reduces to $\widehat{\lambda^{-1}} = \sqrt{2/\pi}$. Consequently, the initial KL divergence in (7) changes. To handle the non-negative support of q_0, we introduce the *capped*-KL divergence between $q_1 = \prod \mathcal{N}(\mu_{1,d}, \sigma_{1,d}^2)$ and q_0, with the subscript $t = 1$ in $\mu_{1,d}$ and $\sigma_{1,d}$ suppressed for brevity:

$$\overline{\text{KL}}(q_1 || q_0) := \sum_{d=1}^{D} \left\{ -\frac{1}{2} \log\left(2\pi\sigma_d^2\right) - \frac{1}{2} - \log \hat{\lambda} + \frac{\hat{\lambda}\sigma_d}{\sqrt{2\pi}} \exp\left(-\frac{\mu_d^2}{2\sigma_d^2} \right) + \hat{\lambda}\mu_d \left(1 - \Phi\left(-\frac{\mu_d}{\sigma_d} \right) \right) \right\} \tag{2}$$

. In fact, this is equivalent to $\text{KL}(q_1 || \bar{q}_0)$, where $\bar{q}_0(\boldsymbol{\theta}) := \prod \lambda \exp\left(-\lambda\theta_d \cdot \mathbb{1}[\theta_d \geqslant 0]\right)$. We formalise these and present the full derivations in Section B.2.

5 Experiments

We focus on the SplitMNIST experiment from Nguyen et al. [2017]. Our codebase can be found at: `https://anonymous.4open.science/r/VCL-ext-E2F0/`. We re-implemented VCL using PyTorch, with the hyperparameter setting of our experiments summarised in Table 2 (Section C.1). This is set to align with Nguyen et al.'s experimental setup where applicable. We explicitly tuned the coreset configurations via grid search over the coreset selection algorithm (random sampling, K-center method) and coreset sizes (0, 50, 100, 200). The optimal thus chosen configuration, K-center

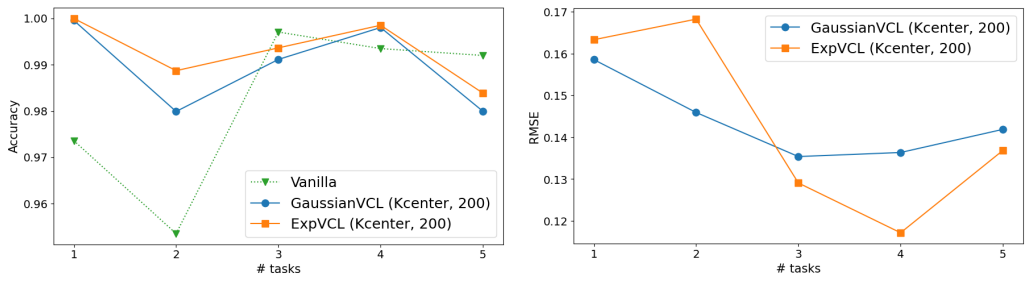
[2]We clarify so because in general, $\mathbb{E}[1/g(X)] \neq 1/\mathbb{E}[g(X)]$.

coresets of size 200, is consistent with Nguyen et al.'s observations. We compare three models: `Vanilla` (non-variational baseline with no coreset), `GaussianVCL` (with $\mathcal{N}(0, I)$ prior), and `ExpVCL` (with $\prod \text{Exp}(\hat{\lambda})$ prior). Each model is trained and tested on both the original classification tasks and their regression counterparts. For classification, we evaluate our models by their test set accuracy across all tasks (Figure 1a and 3a). For regression, we use RMSE for evaluation (Figure 1b and 3b). Table 1 shows our numerical results. We discuss the limitations of our methodology in Section C.2.

6 Results

Model	Lifetime Acc.	Final Acc.	Lifetime RMSE	Final RMSE
`Vanilla`	0.9816 ± 0.0175	0.9819 ± 0.0164	0.9667 ± 0.0346	0.9522 ± 0.0531
`GaussianVCL`	0.9917 ± 0.0071	0.9897 ± 0.0085	0.1330 ± 0.0113	0.1436 ± 0.0084
`ExpVCL`	$\mathbf{0.9942 \pm 0.0059}$	$\mathbf{0.9930 \pm 0.0060}$	$\mathbf{0.1323 \pm 0.0206}$	$\mathbf{0.1429 \pm 0.0197}$

Table 1: Test results (mean $\pm$ std) across all tasks. The best values in each column are highlighted in bold (highest for accuracy, lowest for RMSE). These slightly surpass Nguyen et al.'s results.



(a) Classification (b) Regression

Figure 1: Final test results on all observed tasks in SplitMNIST. The tasks are in order: 0/1, 2/3, 4/5, 6/7, 8/9.

6.1 Results Analysis

Endorsing Nguyen et al.'s results, `Vanilla` shows the worst performance across all metrics in both settings. Figure 1a reveals its particularly unstable learning pattern, characterised by abrupt accuracy improvements at task 3; and in Figure 1b, we omit its results as it fails catastrophically. This suggests an inherent limitation in its ability to balance plasticity and memory stability compared to the VCL approach. Yet, matching Nguyen et al.'s remarks, all models' performance dropped at task 2. Possibly, the models only "remembered" the shared, indistinguishable features of digits '2' and '3' (e.g., the top curvatures). Nonetheless, for classification, `ExpVCL` alleviated the forgetting behaviour to some extent, achieving a test accuracy of 98.87% at task 2, versus the 97.99% for `GaussianVCL`. Moreover, `ExpVCL` outperformed `GaussianVCL` on all classification tasks (see Figure 1a and 3a), and performed the best on average in both settings (see Table 1). However, the task-wise regression results in Figure 1b provide a different perspective: `GaussianVCL` performed the best in the earlier tasks, especially task 2. This echos the fundamental bias-variance trade-off in feature selection Munson and Caruana [2009]: In classification, the exponential prior's sparsity-inducing, L1-like regularisation, may excel by pruning irrelevant features (e.g., background pixels); whereas for regression, the Gaussian prior's L2-regularisation may preserve small but important weights, thus better modelling the smoothness of continuous targets Williams [1995]. Hence, prior selection should be task-dependent.

7 Conclusions

Building upon the VCL framework proposed by Nguyen et al., our work presents two key extensions: 1) reformulation of binary classification tasks as regression tasks within the VCL framework; and 2) replacement of the standard Gaussian prior with an exponential prior over network parameters, while retaining Gaussian mean-field posterior approximations. In this respect, we also introduce the novel *capped*-KL divergence for the initial optimisation phase. Empirically, we identify the exponential prior as a promising alternative to the standard Gaussian prior, especially for classification tasks. For future studies, one could consider addressing those limitations discussed in Section C.2, and exploring other non-Gaussian distributions (e.g., the Gamma distribution) for the prior, the approximation family, or even the hyperprior Tsuchida et al. [2019] over prior parameters.

References

T. Adel, H. Zhao, and R. E. Turner. Continual learning with adaptive weights (claw). *arXiv preprint arXiv:1911.09514*, 2019.

C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015.

C. M. Carvalho, N. G. Polson, and J. G. Scott. Handling sparsity via the horseshoe. In *Artificial intelligence and statistics*, pages 73–80. PMLR, 2009.

T. Cui, A. Havulinna, P. Marttinen, and S. Kaski. Informative bayesian neural network priors for weak signals. *Bayesian Analysis*, 17(4):1121–1151, 2022.

M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385, 2021.

M. M. Derakhshani, X. Zhen, L. Shao, and C. Snoek. Kernel continual learning. In *International Conference on Machine Learning*, pages 2621–2631. PMLR, 2021.

V. Fortuin. Priors in bayesian deep learning: A review. *International Statistical Review*, 90(3): 563–591, 2022.

Z. Ghahramani and H. Attias. Online variational bayesian learning. In *Slides from talk presented at NIPS workshop on Online Learning*, 2000.

S. Kapoor, T. Karaletsos, and T. D. Bui. Variational auto-regressive gaussian processes for continual learning. In *International Conference on Machine Learning*, pages 5290–5300. PMLR, 2021.

J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

F. C. Leone, L. S. Nelson, and R. Nottingham. The folded normal distribution. *Technometrics*, 3(4): 543–550, 1961.

D. J. MacKay et al. Introduction to gaussian processes. *NATO ASI series F computer and systems sciences*, 168:133–166, 1998.

M. A. Munson and R. Caruana. On feature selection, bias-variance, and bagging. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 144–159. Springer, 2009.

K. Neklyudov, D. Molchanov, A. Ashukha, and D. Vetrov. Variance networks: When expectation does not meet your expectations. *arXiv preprint arXiv:1803.03764*, 2018.

C. V. Nguyen, Y. Li, T. D. Bui, and R. E. Turner. Variational continual learning. *arXiv preprint arXiv:1710.10628*, 2017.

T. G. Rudner, F. B. Smith, Q. Feng, Y. W. Teh, and Y. Gal. Continual learning via sequential function-space variational inference. In *International Conference on Machine Learning*, pages 18871–18887. PMLR, 2022.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1): 1929–1958, 2014.

R. Tsuchida, F. Roosta, and M. Gallagher. Richer priors for infinitely wide multi-layer perceptrons. *arXiv preprint arXiv:1911.12927*, 2019.

M. Vladimirova, J. Verbeek, P. Mesejo, and J. Arbel. Understanding priors in bayesian neural networks at the unit level. In *International Conference on Machine Learning*, pages 6458–6467. PMLR, 2019.

L. Wang, X. Zhang, H. Su, and J. Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

F. Wenzel, K. Roth, B. S. Veeling, J. Świątkowski, L. Tran, S. Mandt, J. Snoek, T. Salimans, R. Jenatton, and S. Nowozin. How good is the bayes posterior in deep neural networks really? *arXiv preprint arXiv:2002.02405*, 2020.

P. M. Williams. Bayesian regularization and pruning using a laplace prior. *Neural computation*, 7(1): 117–143, 1995.

Appendix

A Background

A.1 Theoretical Derivations for the Original VCL

The following results are due to Nguyen et al. [2017].

Lemma 1 (True posterior)**.** Given $\mathcal{D}_t = \{\boldsymbol{x}^{(n)}, y_t^{(n)}\}_{n=1}^{N_t}$, $\mathcal{X}_t = \{\boldsymbol{x}^{(n)}\}_{n=1}^{N_t}$, and $\mathcal{Y}_t = \{y^{(n)}\}_{n=1}^{N_t}$ the posterior distribution after seeing T datasets can be computed using Bayes' rule:

$$\begin{aligned} p(\boldsymbol{\theta} \mid \mathcal{D}_{1:T}) &\propto p(\boldsymbol{\theta}) \prod_{t=1}^{T} \prod_{n_t=1}^{N_t} p(y_t^{(n_t)} \mid \boldsymbol{\theta}, \boldsymbol{x}_t^{(n_t)}) \\ &= p(\boldsymbol{\theta}) \prod_{t=1}^{T} p(\mathcal{Y}_t \mid \boldsymbol{\theta}, \mathcal{X}_t) \\ &\propto p(\boldsymbol{\theta} \mid \mathcal{D}_{1:T-1}) p(\mathcal{Y}_T \mid \boldsymbol{\theta}, \mathcal{X}_T) \end{aligned} \tag{3}$$

To lighten notation, henceforth we write $p(\mathcal{D}_t \mid \boldsymbol{\theta}) := p(\mathcal{D}_t \mid \boldsymbol{\theta}, \mathcal{X}_t)$.

Lemma 2 (KL divergence minimisation in Online Variational Inference)**.** Given the approximation family $\mathcal{Q}$, the approximate posterior is chosen as:

$$q_t(\boldsymbol{\theta}) = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}\left(q_t(\boldsymbol{\theta}) || \frac{1}{Z_t} q_{t-1}(\boldsymbol{\theta}) p(\mathcal{D}_t \mid \boldsymbol{\theta}) \right), \quad \text{for } t = 1, 2, \ldots, T \tag{4}$$

where Z_t is the intractable normalising constant of $p_t^*(\boldsymbol{\theta}) = q_{t-1}(\boldsymbol{\theta}) p(\mathcal{D}_t \mid \boldsymbol{\theta})$, and the zeroth approximate distribution is defined to be the prior, $q_0(\boldsymbol{\theta}) = p(\boldsymbol{\theta})$.

Lemma 3 (Posterior decomposition using coresets)**.** Given $\mathcal{D}_t$ as defined in Lemma 1, and coreset $C_t \subseteq \mathcal{D}_t$, the following holds by Bayes' rule:

$$p(\boldsymbol{\theta} \mid \mathcal{D}_{1:t}) \propto \underbrace{p(\boldsymbol{\theta} \mid \mathcal{D}_{1:t} \setminus C_t)}_{\substack{\text{posterior from data} \\ \text{not in new coreset}}} \underbrace{p(C_t \mid \boldsymbol{\theta})}_{\substack{\text{likelihood from data} \\ \text{in new coreset}}} \approx \tilde{q}_t(\boldsymbol{\theta}) p(C_t \mid \boldsymbol{\theta}) \tag{5}$$

where $\tilde{q}_t(\boldsymbol{\theta})$ is the variational distribution that approximates the contribution to the posterior from the non-coreset data points.

The posterior from the non-coreset can in turn be computed via the following recursion:

$$\begin{aligned} p(\boldsymbol{\theta} \mid \mathcal{D}_{1:T} \setminus C_t) &= p(\boldsymbol{\theta} \mid \mathcal{D}_{1:t-1} \setminus C_{t-1}) p(C_{t-1} \setminus C_t \mid \boldsymbol{\theta}) p(\mathcal{D}_t \setminus C_t \mid \boldsymbol{\theta}) \\ &\approx \tilde{q}_{t-1}(\boldsymbol{\theta}) p(\mathcal{D}_t \cup C_{t-1} \setminus C_t \mid \boldsymbol{\theta}) \end{aligned} \tag{6}$$

Definition 4 (Loss function for VCL)**.** VCL minimises the following objective:

$$\mathcal{L}_{\mathrm{VCL}}^t(q_t(\boldsymbol{\theta})) = \sum_{n=1}^{N_t} \mathbb{E}_{\boldsymbol{\theta} \sim q_t(\boldsymbol{\theta})} \Big[\log p(y_t^{(n)} \mid \boldsymbol{\theta}, \boldsymbol{x}_t^{(n)}) \Big] - \mathrm{KL}(q_t(\boldsymbol{\theta}) || q_{t-1}(\boldsymbol{\theta})) \tag{7}$$

which is equivalent to maximising the negative of the variational evidence lower bound to the online marginal likelihood. As usual, one may use simple Monte Carlo to estimate the expected log-likelihood, together with the local reparametrisation trick to compute the gradients.

B Proposed Extensions

Unless otherwise cited or referring to standard definitions, the following results are based on our own derivations.

B.1 Additional Details on Extension 1: Changing the task

Lemma 5 (Task-specific loss in VCL for Regression)**.** Given $\boldsymbol{y} \sim \mathcal{N}(f_\theta(\boldsymbol{x}), \sigma_y^2 I)$, maximising the expected log-likelihood is equivalent to minimising the Mean Squared Error (MSE).

Proof. The probability density function of a multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix Σ is given by:

$$p(\boldsymbol{y} \mid \boldsymbol{\theta}, \boldsymbol{x}) = (2\pi)^{-D/2}|\Sigma|^{-1/2} \cdot \exp\left(-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{y}-\boldsymbol{\mu})\right)$$

For $\boldsymbol{\mu} = f_{\boldsymbol{\theta}}(\boldsymbol{x}), \Sigma = \sigma_y^2 I_D$, this becomes:

$$p(\boldsymbol{y} \mid \boldsymbol{\theta}, \boldsymbol{x}) = (2\pi\sigma_y^2)^{-D/2} \exp\left(-\frac{1}{2\sigma_y^2}\|\boldsymbol{y} - f_{\boldsymbol{\theta}}(\boldsymbol{x})\|^2\right)$$

So the log-likelihood is given by:

$$\log p(\boldsymbol{y} \mid \boldsymbol{\theta}, \boldsymbol{x}) = -\frac{D}{2}\log(2\pi\sigma_y^2) - \frac{1}{2\sigma_y^2}\|\boldsymbol{y} - f_{\boldsymbol{\theta}}(\boldsymbol{x})\|^2 = -\frac{1}{2\sigma_y^2}\|\boldsymbol{y} - f_{\boldsymbol{\theta}}(\boldsymbol{x})\|^2 + \text{const.}$$

Therefore

$$\arg\max_{\boldsymbol{\theta}} \log p(\boldsymbol{y} \mid \boldsymbol{\theta}, \boldsymbol{x}) = \arg\min_{\boldsymbol{\theta}} \|\boldsymbol{y} - f_{\boldsymbol{\theta}}(\boldsymbol{x})\|^2$$

□

B.2 Additional Details on Extension 2: Changing the prior

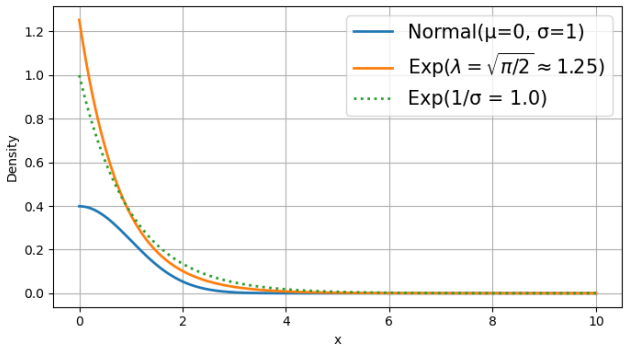


Figure 2: Comparison of the probability density functions of the $\mathcal{N}(0,1)$ Gaussian and Exp($\hat{\lambda}$) exponential prior, where $\hat{\lambda}$ is computed following Equation (1). With this estimator, the exponential distribution aligns more closely with the curvature of the Gaussian distribution in the middle region (around one or two σs away from zero), compared to the naïve scale-matching estimator $1/\sigma$. It also induces stronger regularisation through accelerating the weight decay near zero.

Definition 6 (Exponential PDF)**.** A univariate exponential distribution with rate parameter λ (and scale parameter $1/\lambda$), written Exp(λ), has probability density function:

$$p(x) = \lambda e^{-\lambda x}, \quad x > 0 \tag{8}$$

Definition 7 (*Cap* of exponential distribution)**.** Given a univariate Exponential $p = \text{Exp}(\lambda)$, we define its *cap* function as

$$\bar{p}(x) = \lambda \exp\left(-\lambda x \cdot \mathbb{1}_{x \geqslant 0}\right) = \begin{cases} \lambda e^{-\lambda x}, & x \geqslant 0 \\ \lambda, & x < 0 \end{cases} \tag{9}$$

Definition 8 (*Capped*-KL divergence)**.** Consider distributions p and q. We define the *capped*-KL divergence from q to p as

$$\overline{\text{KL}}(q||p) = \text{KL}(q||\bar{p}) \tag{10}$$

That is, the KL divergence from q to the *cap* function of p, as defined in Equation 9.

Lemma 9 (*Capped*-KL between Univariate Gaussian and Exponential)**.** Consider univariate distributions $q = \mathcal{N}(\mu, \sigma^2)$ and $p = \text{Exp}(\lambda)$. The *capped*-KL divergence from q to p is given by:

$$\text{KL}(q||p) = -\frac{1}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2} - \log\lambda + \frac{\lambda\sigma}{\sqrt{2\pi}} \cdot \exp\left(-\frac{\mu^2}{2\sigma^2}\right) + \lambda\mu\left(1 - \Phi\left(-\frac{\mu}{\sigma}\right)\right) \tag{11}$$

where Φ is the standard normal CDF.

Proof. We have $p(x)$ defined in Equation (8), $\bar{p}(x)$ defined in Equation (9), and

$$q(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Note that the standard KL divergence $\text{KL}(q||p)$ is infinite, as $\text{support}(q) = \mathbb{R} \nsubseteq \mathbb{R}^+ = \text{support}(p)$. Hence, we turn to its *capped* counterpart given in Definition 8:

$$\overline{\text{KL}}(q||p) = \text{KL}(q||\bar{p}) = \int_{-\infty}^{\infty} q(x) \log \frac{q(x)}{\bar{p}(x)}\, dx = \int_{-\infty}^{\infty} q(x)\log q(x)\, dx - \int_{-\infty}^{\infty} q(x)\log\bar{p}(x)\, dx$$

The first term is given by:

$$\begin{aligned}
\int_{-\infty}^{\infty} q(x)\log q(x)\, dx &= \mathbb{E}_q[\log q(x)] \\
&= \mathbb{E}_q\left[-\frac{1}{2}\log\left(2\pi\sigma^2\right) - \frac{(x-\mu)^2}{2\sigma^2}\right] \\
&= -\frac{1}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}\mathbb{E}_q[X^2] && \text{(by linearity of expectation)} \\
&= -\frac{1}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}\cdot\sigma^2 && (\mathbb{E}[X^2] = \mathbb{V}[X] - \mathbb{E}[X]^2 = \sigma^2) \\
&= -\frac{1}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2}
\end{aligned}$$

The second term is given by:

$$\begin{aligned}
\int_{-\infty}^{\infty} q(x)\log\bar{p}(x)\, dx &= \int_{-\infty}^{\infty} q(x)\big(\log\lambda - \lambda x\cdot\mathbb{1}_{x\geqslant 0}\big)\, dx \\
&= (\log\lambda)\cdot\int_{-\infty}^{\infty} q(x)\, dx - \lambda\int_{0}^{\infty} x\cdot q(x)\, dx && (q(x) \text{ is a PDF}) \\
&= \log\lambda - \lambda\int_{0}^{\infty} x\cdot q(x)\, dx \\
&= \log\lambda - \frac{\lambda}{\sqrt{2\pi\sigma^2}}\int_{0}^{\infty} x\cdot\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) dx \\
&= \log\lambda - \frac{\lambda}{\sqrt{2\pi\sigma^2}}\cdot\sigma\left(\sigma\exp\left(-\frac{\mu^2}{2\sigma^2}\right) + \mu\sqrt{2\pi}\left(1 - \Phi\left(-\frac{\mu}{\sigma}\right)\right)\right) \\
&= \log\lambda - \frac{\lambda\sigma}{\sqrt{2\pi}}\cdot\exp\left(-\frac{\mu^2}{2\sigma^2}\right) - \lambda\mu\left(1 - \Phi\left(-\frac{\mu}{\sigma}\right)\right)
\end{aligned}$$

For the second-last equality, the integral $I := \int_0^\infty x \cdot \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)\, dx$ is computable via integration by substitution: Let $z = \frac{x-\mu}{\sigma}$, then $x = \mu + \sigma z$ and $dx = \sigma\, dz$.

Substituting these into the integral, we get:

$$\begin{aligned} I &= \int_{-\frac{\mu}{\sigma}}^\infty (\mu + \sigma z) \cdot \exp\left(-\frac{z^2}{2}\right) \sigma\, dz \\ &= \sigma \cdot \left[\mu \int_{-\frac{\mu}{\sigma}}^\infty \exp\left(-\frac{z^2}{2}\right)\, dz + \sigma \int_{-\frac{\mu}{\sigma}}^\infty z \exp\left(-\frac{z^2}{2}\right)\, dz\right] \end{aligned}$$

We compute the two integrals respectively:

$$\int_{-\frac{\mu}{\sigma}}^\infty \exp\left(-\frac{z^2}{2}\right)\, dz = \sqrt{2\pi} \int_{-\frac{\mu}{\sigma}}^\infty \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right)\, dz = \sqrt{2\pi} \int_{-\frac{\mu}{\sigma}}^\infty \mathcal{N}(z; 0, 1)\, dz = \sqrt{2\pi}\left(1 - \Phi\left(-\frac{\mu}{\sigma}\right)\right)$$

The second integral is again computable via integration by substitution, with $u = z^2/2$, so $du = z\, dz$:

$$\int_{-\frac{\mu}{\sigma}}^\infty \exp\left(-\frac{z^2}{2}\right)\, dz = \int_{\frac{\mu^2}{2\sigma^2}}^\infty \exp\left(-u\right) du = \exp\left(-\frac{\mu^2}{2\sigma^2}\right)$$

Combining these gives the desired results.

□

Lemma 10 (*Capped*-KL between independent multivariate Gaussian and Exponential)**.** Consider multivariate distributions $q(\boldsymbol{\theta}) = \prod_d^D \mathcal{N}(\mu_d, \sigma_d^2)$ and $p(\boldsymbol{\theta}) = \prod_d^D \text{Exp}(\lambda)$. The capped-KL divergence from q to p is given by

$$\overline{\text{KL}}(q||p) = \sum_{d=1}^D \overline{\text{KL}}(q_d||p_d) \tag{12}$$

where each $\overline{\text{KL}}(q_d||p_d)$ is given by

$$\overline{\text{KL}}(q_d||p_d) = -\frac{1}{2}\log\left(2\pi\sigma_d^2\right) - \frac{1}{2} - \log\lambda + \frac{\lambda\sigma_d}{\sqrt{2\pi}} \cdot \exp\left(-\frac{\mu_d^2}{2\sigma_d^2}\right) + \lambda\mu_d\left(1 - \Phi\left(-\frac{\mu_d}{\sigma_d}\right)\right)$$

Proof. Given mutually independent parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_D)$, the joint distributions factorise as:

$$q(\boldsymbol{\theta}) = \prod_{d=1}^D q_d(\theta_d), \quad p(\boldsymbol{\theta}) = \prod_{d=1}^D p_d(\theta_d).$$

The KL divergence then decomposes additively:

$$\begin{aligned} \overline{\text{KL}}(q||p) &= \mathbb{E}_q\left[\log \frac{\prod_{d=1}^D q_d(\theta_d)}{\prod_{d=1}^D \bar{p}_d(\theta_d)}\right] = \mathbb{E}_q\left[\log \prod_{d=1}^D \frac{q_d(\theta_d)}{\bar{p}_d(\theta_d)}\right] \\ &= \mathbb{E}_{q_d(\theta_d)}\left[\sum_{d=1}^D \log \frac{q_d(\theta_d)}{\bar{p}_d(\theta_d)}\right] \\ &= \sum_{d=1}^D \mathbb{E}_{q_d(\theta_d)}\left[\log \frac{q_d(\theta_d)}{\bar{p}_d(\theta_d)}\right] \qquad \text{(by linearity of expectation)} \\ &= \sum_{d=1}^D \overline{\text{KL}}(q_d||p_d) \end{aligned}$$

Here, each $\overline{\text{KL}}(q_d||p_d)$ is given in Equation (11) (Lemma 9), with each μ replaced with the corresponding μ_d. □

Definition 11 (Absolute Moment Matching Estimator of λ^{-1})**.** Consider $q = \mathcal{N}(\mu, \sigma^2)$ and $p = \text{Exp}(\lambda)$. We define the *absolute moment matching estimator* for the scale parameter λ^{-1} as:

$$\widehat{\lambda^{-1}} := \mathbb{E}_q[|X|] = \sigma\sqrt{\frac{2}{\pi}}\exp\left(-\frac{\mu^2}{2\sigma^2}\right) + \mu\left(1 - 2\Phi\left(-\frac{\mu}{\sigma}\right)\right) \tag{13}$$

Proof. For a random variable $X \sim q$,

$$\mathbb{E}_q[|X|] = \sigma\sqrt{\frac{2}{\pi}}\exp\left(-\frac{\mu^2}{2\sigma^2}\right) + \mu\left(1 - 2\Phi\left(-\frac{\mu}{\sigma}\right)\right) \tag{14}$$

This can be derived via integration similar to the proof for Lemma 9. Alternatively, note that if $X \sim \mathcal{N}(\mu, \sigma^2)$, then $|X|$ follows a folded normal distribution with the same parameters, whose expected value (as given above) is an established result Leone et al. [1961].

Since $\text{Exp}(\lambda)$ is supported on the interval $[0, \infty)$ and has expectation $1/\lambda$, matching the first absolute moment (i.e., expected absolute value) of p and q, we get:

$$\mathbb{E}_{X\sim p}[|X|] = \mathbb{E}_{X\sim p}[X] = \lambda^{-1} = \mathbb{E}_{X\sim q}[|X|]$$

with the RHS defined in Equation (14). □

C Additional Details on Experiments

C.1 Experimental Setup

Hyperparam	Value
depth	2
input dim.	784
hidden dim.	256
activation fn.	ReLU
# MC samples	10
# epochs	100
batch size	256
learning rate	1e-3
patience	5
early stop threshold	1e-4

Table 2: Hyperparameter setting

Code attribution. Our implementation is partially adapted from the original VCL implementation by Nguyen et al., which is available at: `https://github.com/nvcuong/variational-continual-learning/`. In particular, we reproduced their coreset selection algorithms to ensure faithful replication of their experimental setup. The remainder, however, is mostly based on our own understanding of their presented work in Nguyen et al. [2017].

Notes on implementation. If we do allow negative weights in the model (which we do in these experiments), we should assign random signs (e.g., from a standard Gaussian distribution)[3] when initialising $\boldsymbol{\theta}$. This also helps stabilise training. Ultimately, it is the form of regularisation imposed by an exponential prior that we desire, rather than a restriction on the domain of $\boldsymbol{\theta}$.

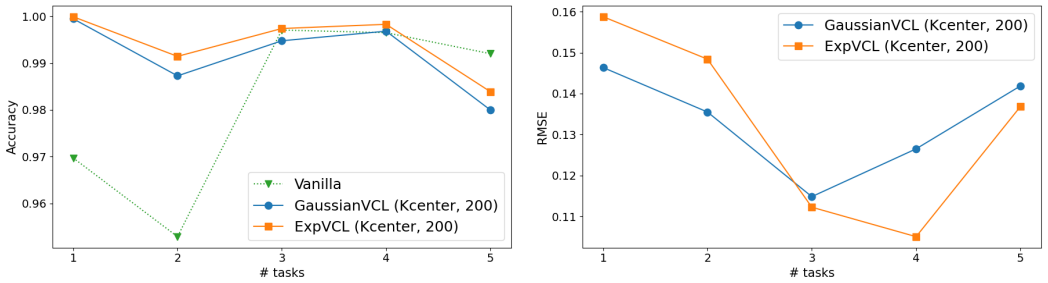
C.2 Limitations in Methodology

Our methodology poses several limitations, mostly due to computational and time constraints. Importantly, we only reproduced the SplitMNIST experiment which comprises only binary classification

[3]We are thus effectively sampling $\boldsymbol{\theta}_0$ from a double-exponential, or Laplacian distribution.

tasks. Given more time, we should generalise our investigation to multiclass classification such as the PermutedMNIST experiment, and standard regression tasks. The experimentation approach that follows from Nguyen et al. [2017] also has several inherent limitations. One facet is the evaluation metrics used. To assess the effectiveness of VCL for catastrophic forgetting, we could use more specific metrics like *backward transfer* for measuring memory stability, and *forward transfer* for learning plasticity Wang et al. [2024]. Certain hyperparameters (e.g., model width, initial prior parameters) and architectural components (e.g., dropout Srivastava et al. [2014]) that influence the models' forgetting behaviour also deserve further exploration.

C.3 Additional Experimental Results



(a) Classification

(b) Regression

Figure 3: Mean test results on all observed tasks for the SplitMNIST dataset. Again, RMSE results for the Vanilla model are omitted as they fail catastrophically.