GENERALIZED VARIATIONAL CONTINUAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Continual learning deals with training models on new tasks and datasets in an online fashion. One strand of research has used probabilistic regularization for continual learning, with two of the main approaches in this vein being Online Elastic Weight Consolidation (Online EWC) and Variational Continual Learning (VCL). VCL employs variational inference, which in other settings has been improved empirically by applying likelihood-tempering. We show that applying this modification to VCL recovers Online EWC as a limiting case, allowing for interpolation between the two approaches. We term the general algorithm Generalized VCL (GVCL). In order to mitigate the observed overpruning effect of VI, we take inspiration from a common multi-task architecture, neural networks with task-specific FiLM layers, finding that this leads to significant performance gains. In the small-data regime, GVCL strongly outperforms existing baselines. In larger datasets, GVCL outperforms or is competitive with existing baselines in terms of accuracy, whilst also providing significantly better calibration.

1 INTRODUCTION

Continual learning methods enable learning when a set of tasks changes over time. This topic is of practical interest as many real-world applications require models to be regularly updated as new data is collected or new tasks arise. Standard machine learning models and training procedures fail in these settings (French, 1999), so bespoke architectures and fitting procedures are required.

This paper makes two main contributions to continual learning for neural networks. First, we develop a new regularization-based approach to continual learning. Regularization approaches adapt parameters to new tasks while keeping them close to settings that are appropriate for old tasks. Two popular approaches of this type are Variational Continual Learning (VCL) (Nguyen et al., 2018) and Online Elastic Weight Consolidation (Online EWC) (Kirkpatrick et al., 2017; Schwarz et al., 2018). The former is based on a variational approximation of a neural network's posterior distribution over weights, while the latter uses Laplace's approximation. In this paper, we propose Generalized Variational Continual Learning (GVCL) of which VCL and Online EWC are two special cases. Under this unified framework, we are able to combine the strengths of both approaches. GVCL is closely related to likelihood-tempered Variational Inference (VI), which has been found to improve performance in standard learning settings (Zhang et al., 2018; Osawa et al., 2019). We also see significant performance improvements in continual learning.

Our second contribution is to introduce an architectural modification to the neural network that combats the deleterious overpruning effect of VI (Trippe & Turner, 2018; Turner & Sahani, 2011). We analyze pruning in VCL and show how task-specific FiLM layers mitigate it. Combining this architectural change with GVCL further improves performance which exceeds or is within statistical error of strong baselines such as HAT (Serrà et al., 2018) and PathNet (Fernando et al., 2017).

The paper is organised as follows. Section 2 outlines the derivation of GVCL, shows how it unifies many continual learning algorithms, and describes why it might be expected to perform better than them. Section 3 introduces FiLM layers, first from the perspective of multi-task learning, and then through the lens of variational over-pruning, showing how FiLM layers mitigate this pathology of VCL. Finally, in Section 5 we test GVCL on a number of standard benchmarks, including ones with few samples, a regime that could benefit more from continual learning. We find that GVCL outperforms existing baselines on a variety of metrics, including raw accuracy, forwards and backwards transfer, and calibration error. Code will be made available upon acceptance.

2 GENERALIZED VARIATIONAL CONTINUAL LEARNING

In this section, we introduce Generalized Variational Continual Learning (GVCL) as a likelihood-tempered version of VCL, with further details in Appendix B. We show how GVCL recovers Online EWC. We also discuss further links between GVCL and the Bayesian cold posterior in Appendix C.

2.1 LIKELIHOOD-TEMPERING IN VARIATIONAL CONTINUAL LEARNING

Variational Continual Learning (VCL). Bayes' rule calculates a posterior distribution over model parameters θ based on a prior distribution $p(\theta)$ and some dataset $D_T = \{X_T, y_T\}$. Bayes' rule naturally supports online and continual learning by using the previous posterior $p(\theta|D_{T-1})$ as a new prior when seeing new data (Nguyen et al., 2018). Due to the intractability of Bayes' rule in complicated models such as neural networks, approximations are employed, and VCL (Nguyen et al., 2018) uses one such approximation, Variational Inference (VI). This approximation is based on approximating the posterior $p(\theta|D_T)$ with a simpler distribution $q_T(\theta)$, such as a Gaussian. This is achieved by optimizing the ELBO for the optimal $q_T(\theta)$,

$$\text{ELBO}_{\text{VCL}} = \mathbb{E}_{\theta\sim q_T(\theta)}[\log p(D_T|\theta)] - D_{\text{KL}}(q_T(\theta)\|q_{T-1}(\theta)), \tag{1}$$

where $q_{T-1}(\theta)$ is the approximation to the previous task posterior. Intuitively, this refines a distribution over weight samples that balances good predictive performance (the first expected prediction accuracy term) while remaining close to the prior (the second KL-divergence regularization term).

Likelihood-tempered VCL. Optimizing the ELBO will recover the true posterior if the approximating family is sufficiently rich. However, the simple families used in practice typically lead to poor test-set performance. Practitioners have found that performance can be improved by down-weighting the KL-divergence regularization term by a factor β, with $0 < \beta < 1$. Examples of this are seen in Zhang et al. (2018) and Osawa et al. (2019), where the latter uses a "data augmentation factor" for down-weighting. In a similar vein, sampling from "cold posteriors" in SG-MCMC has also been shown to outperform the standard Bayes posterior, where the cold posterior is given by $p_T(\theta|D) \propto p(\theta|D)^{\frac{1}{T}}, T < 1$ (Wenzel et al., 2020). Values of $\beta > 1$ have also been used to improve the disentanglement variational autoencoder learned models (Higgins et al., 2017). We down-weight the KL-divergence term in VCL, optimizing the β-ELBO[1],

$$\beta\text{-ELBO} = \mathbb{E}_{\theta\sim q_T(\theta)}[\log p(D_T|\theta)] - \beta D_{\text{KL}}(q_T(\theta)\|q_{T-1}(\theta)).$$

VCL is trivially recovered when $\beta = 1$. We will now show that surprisingly as $\beta \to 0$, we recover a special case of Online EWC. Then, by modifying the term further as required to recover the full version of Online EWC, we will arrive at our algorithm, Generalized VCL.

2.2 ONLINE EWC IS A SPECIAL CASE OF GVCL

We analyze the effect of KL-reweighting on VCL in the case where the approximating family is restricted to Gaussian distributions over θ. We will consider training all the tasks with a KL-reweighting factor of β, and then take the limit $\beta \to 0$, recovering Online EWC. Let the approximate posteriors at the previous and current tasks be denoted as $q_{T-1}(\theta) = \mathcal{N}(\theta; \mu_{T-1}, \Sigma_{T-1})$ and $q_T(\theta) = \mathcal{N}(\theta; \mu_T, \Sigma_T)$ respectively, where we are learning $\{\mu_T, \Sigma_T\}$. The optimal Σ_T under the β-ELBO has the form (see Appendix B),

$$\Sigma_T^{-1} = \frac{1}{\beta}\nabla_{\mu_T}\nabla_{\mu_T}\mathbb{E}_{q_T(\theta)}[-\log p(D_T|\theta)] + \Sigma_{T-1}^{-1}. \tag{2}$$

Now take the limit $\beta \to 0$. From Equation 2, $\Sigma_T \to 0$, so $q_T(\theta)$ becomes a delta function, and

$$\Sigma_T^{-1} = -\frac{1}{\beta}\nabla_{\mu_T}\nabla_{\mu_T}\log p(D_T|\theta = \mu_T) + \Sigma_{T-1}^{-1} = \frac{1}{\beta}H_T + \Sigma_{T-1}^{-1} = \frac{1}{\beta}\sum_{t=1}^{T} H_t + \Sigma_0^{-1}, \tag{3}$$

where H_T is the Tth task Hessian[2]. Although the learnt distribution $q_T(\theta)$ becomes a delta function (and not a full Gaussian distribution as in Laplace's approximation), we will see that a cancellation

[1] We slightly abuse notation by writing the likelihood as $p(D_T|\theta)$ instead of $p(y_T|\theta, X_T)$.

[2] The actual Hessian may not be positive semidefinite while Σ is, so here we refer to a positive semidefinite approximation of the Hessian.

of β factors in the β-ELBO will lead to the eventual equivalence between GVCL and Online EWC. Consider the terms in the β-ELBO that only involve μ_T:

$$\begin{aligned}\beta\text{-ELBO} &= \mathbb{E}_{\theta\sim q_T(\theta)}[\log p(D_T|\theta)] - \frac{\beta}{2}(\mu_T - \mu_{T-1})^\top \Sigma_{T-1}^{-1}(\mu_T - \mu_{T-1}) \\ &= \log p(D_T|\theta = \mu_T) - \frac{1}{2}(\mu_T - \mu_{T-1})^\top \left(\sum_{t=1}^{T-1} H_t + \beta\Sigma_0^{-1}\right)(\mu_T - \mu_{T-1}), \qquad (4)\end{aligned}$$

where we have set the form of Σ_{T-1} to be as in Equation 3. Equation 4 is an instance of the objective function used by a number of continual learning methods, most notably Online EWC[3] (Schwarz et al., 2018), Online-Structured Laplace (Ritter et al., 2018), and SOLA (Yin et al., 2020). These algorithms can be recovered by changing the approximate posterior class $\mathcal{Q}$ to Gaussians with diagonal, block-diagonal Kronecker-factored covariance matrices, and low-rank precision matrices, respectively.

Based on this analysis, we see that β can be seen as interpolating between VCL, with $\beta = 1$, and continual learning algorithms which use point-wise approximations of curvature as $\beta \to 0$. In Appendix A we explore how β controls the scale of the quadratic curvature approximation, picking up local structure for small β or global structure for large β.

2.3 Reweighting the quadratic term

As described above, the β-ELBO recovers instances of a number of existing second-order continual learning algorithms including Online EWC as special cases. However, the correspondence does not recover a key hyperparameter λ used by these methods that up-weights the quadratic regularization term, instead producing an implicit value of $\lambda = 1$, i.e. equal weight between tasks of equal sample count. In practice it is found that algorithms such as Online EWC perform best when $\lambda > 1$, typically $10 - 1000$. We modify the β-ELBO to incorporate an equivalent parameter by introducing λ into the standard KL-divergence for Gaussians, and allow $\lambda > 1$,

$$D_{\mathrm{KL}\lambda}(q_T\|q_{T-1}) := \tfrac{1}{2}\left(\lambda(\mu_T - \mu_{T-1})^\top \Sigma_{T-1}^{-1}(\mu_T - \mu_{T-1}) + \mathrm{Tr}(\Sigma_{T-1}^{-1}\Sigma_T) + \log|\tfrac{\Sigma_{T-1}}{\Sigma_T}| - d\right).$$

In the limit of $\beta \to 0$, our λ coincides with Online EWC's λ, if the tasks have the same number of samples. However, this form of λ has a problem: it increases the regularization strength of the initial prior Σ_0 on the mean parameter update. We empirically found that this negatively affects performance. We therefore propose a different version of λ, which only up-weights the "data-dependent" parts of Σ_{T-1}. We define,

$$\tilde{\Sigma}_{T,\lambda}^{-1} := \frac{\lambda}{\beta}\sum_{t=1}^{T} H_t + \Sigma_0^{-1} = \lambda(\Sigma_T^{-1} - \Sigma_0^{-1}) + \Sigma_0^{-1}.$$

In practice, it is necessary to clip negative values of $\Sigma_T^{-1} - \Sigma_0^{-1}$ to keep $\tilde{\Sigma}_{T,\lambda}^{-1}$ positive definite. This is only required because of errors during optimization. We then use a modified KL-divergence,

$$D_{\mathrm{KL}\tilde{\lambda}}(q_T\|q_{T-1}) = \tfrac{1}{2}\left((\mu_T - \mu_{T-1})^\top \tilde{\Sigma}_{T-1,\lambda}^{-1}(\mu_T - \mu_{T-1}) + \mathrm{Tr}(\Sigma_{T-1}^{-1}\Sigma_T) + \log|\tfrac{\Sigma_{T-1}}{\Sigma_T}| - d\right).$$

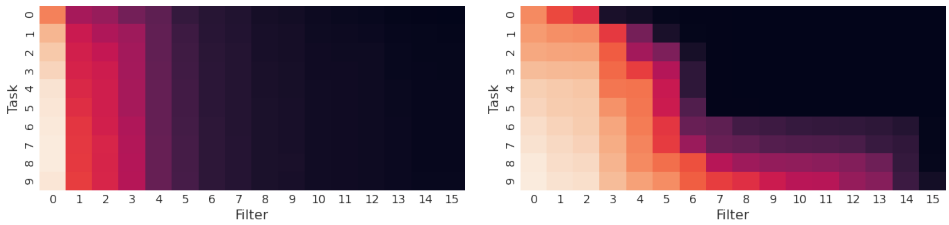
Note that in Online EWC, there is another parameter γ, that down-weights the previous Fisher matrices. As shown in Appendix B, we can introduce this hyperparameter by weighting the Trace term in the KL divergence by γ. However, we do not find that this is necessary in our algorithm. Combining everything, we have our objective for GVCL,

$$\mathbb{E}_{\theta\sim q_T(\theta)}[\log p(D_T|\theta)] - \beta D_{\mathrm{KL}\tilde{\lambda}}(q_T(\theta)\|q_{T-1}(\theta)).$$

3 FiLM Layers for Continual Learning

The Generalized VCL algorithm proposed in Section 2 is applicable to any model. Here we discuss a multi-task neural network architecture that is especially well-suited to GVCL when the task ID is known at both training and inference time: neural networks with task-specific FiLM layers.

[3]EWC uses the Fisher information, but our derivation results in the Hessian. The two matrices coincide when the model has near-zero training loss, as is often the case (Martens, 2020).



(a) GVCL (b) GVCL+ FiLM

Figure 1: Visualizations of deviation from the prior distribution for filters in the first layer of a convolutional networks trained on Hard-CHASY. Lighter colours indicate an active filter for that task. Models are trained either (a) sequentially using GVCL, or (b) sequentially with GVCL + FiLM. FiLM layers increase the number of active units.

3.1 Background to FiLM Layers

The most common architecture for continual learning is the multi-headed neural network. A shared set of body parameters act as the feature extractor. For every task, features are generated in the same way, before finally being passed to separate head networks for each task. This architecture does not allow for task-specific differentiation in the feature extractor, which is limiting (consider, for example, the different tasks of handwritten digit recognition and image recognition). FiLM layers (Perez et al., 2017) address this limitation by linearly modulating features for each specific task so that useful features can be amplified and inappropriate ones ignored. In fully-connected layers, the transformation is applied element-wise: for a hidden layer with width W and activation values h_i, $1 \leq i \leq W$, FiLM layers perform the transformation $h'_i = \gamma_i h_i + b_i$, before being passed on to the remainder of the network. For convolutional layers, transformations are applied filter-wise. Consider a layer with N filters of size $K \times K$, resulting in activations $h_{i,j,k}, 1 \leq i \leq N, 1 \leq j \leq W, 1 \leq k \leq H$, where W and H are the dimensions of the resulting feature map. The transformation has the form $h'_{i,j,k} = \gamma_i * h_{i,j,k} + b_i$. The number of required parameters scales with the number of filters, as opposed to the full activation dimension, making them computationally cheap and parameter-efficient. FiLM layers have previously been shown to help with fine-tuning for transfer learning (Rebuffi et al., 2017), multi-task meta-learning (Requeima et al., 2020), and few-shot learning (Perez et al., 2017). In Appendix E, we show how FiLM layer parameters are interpretable, with similarities between FiLM layer parameters for similar tasks in a multi-task setup.

3.2 Combining GVCL and FiLM Layers

It is simple to apply GVCL to models which utilize FiLM layers. Since these layers are specific to each task they do not need a distributional treatment or regularization as was necessary to support continual learning of the shared parameters. Instead, point estimates are found by optimising the GVCL objective function. This has a well-defined optimum unlike joint MAP training when FiLM layers are added (see Appendix D for a discussion). We might expect an improved performance for continual learning by introducing task-specific FiLM layers as this results in a more suitable multi-task model. However, when combined with GVCL, there is an additional benefit.

When applied to multi-head networks, VCL tends to prune out large parts of the network (Trippe & Turner, 2018; Turner & Sahani, 2011) and GVCL inherits this behaviour. This occurs in the following way: First, weights entering a node revert to their prior distribution due to the KL-regularization term in the ELBO. These weights then add noise to the network, affecting the likelihood term of the ELBO. To avoid this, the bias concentrates at a negative value so that the ReLU activation effectively shuts off the node. In the single task setting, this is often relatively benign and can even facilitate compression (Louizos et al., 2017; Molchanov et al., 2017). However, in continual learning the effect is pathological: the bias remains negative due to its low variance, meaning that the node is effective shut off from that point forward, preventing the node from re-activating. Ultimately, large sections of the network can be shut off after the first task and cannot be used for future tasks, which wastes network capacity (see Figure 1a).

In contrast, when using task-specific FiLM layers, pruning can be achieved by either setting the FiLM layer scale to 0 or the FiLM layer bias to be negative. Since there is no KL-penalty on these parameters, it is optimal to prune in this way. Critically, both the incoming weights and the bias of a pruned node can then return to the prior without adding noise to the network, meaning that the node can be re-activated in later tasks. The increase in the number of unpruned units can be seen in Figure 1b. In Appendix F we provide more evidence of this mechanism.

4 Related Work

Regularization-based continual learning. Many algorithms attempt to regularize network parameters based on a metric of importance. Section 2 shows how some methods can be seen as special cases of GVCL. We now focus on other related methods. Lee et al. (2018) proposed IMM, which is an extension to EWC which merges posteriors based on their Fisher information matrices. Ahn et al. (2019), like us, use regularizers based on the ELBO, but also measure importance on a per-node basis rather than a per-weight one. SI (Zenke et al., 2017) measures importance using "Synaptic Saliency," as opposed to methods based on approximate curvature.

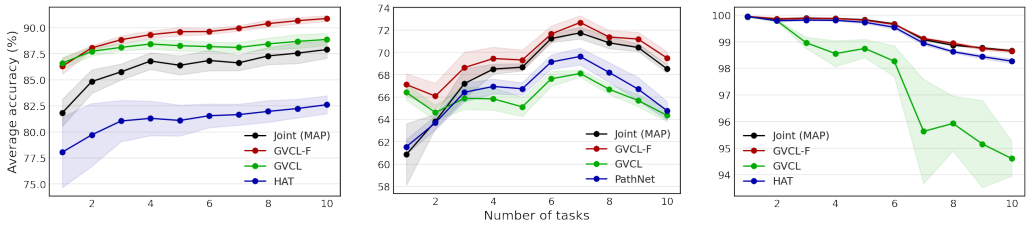
Architectural approaches to continual learning. This family of methods modifies the standard neural architecture by adding components to the network. Progressive Neural Networks (Rusu et al., 2016) adds a parallel column network for every task, growing the model size over time. PathNet (Fernando et al., 2017) fixes the model size while optimizing the paths between layer columns. Architectural approaches are often used in tandem with regularization based approaches, such as in HAT (Serrà et al., 2018), which uses per-task gating parameters alongside a compression-based regularizer. Adel et al. (2020) propose CLAW, which also uses variational inference alongside per-task parameters, but requires a more complex meta-learning based training procedure involving multiple splits of the dataset. GVCL with FiLM layers adds to this list of hybrid architectural-regularization based approaches. See Appendix G for a more comprehensive related works section.

5 Experiments

We run experiments in the small-data regime (Easy-CHASY and Hard-CHASY) (Section 5.1), on *Split*-MNIST (Section 5.1), on the larger Split CIFAR benchmark (Section 5.2), and on a much larger Mixed Vision benchmark consisting of 8 different image classification datasets (Section 5.3). In order to compare continual learning performance, we compare final average accuracy, forward transfer (the improvement on the current task as number of past tasks increases (Pan et al., 2020)) and backward transfer (the difference in accuracy between when a task is first trained and its accuracy after the final task (Lopez-Paz & Ranzato, 2017)). We compare to many baselines, but due to space constraints, only report the best-performing baselines in the main text. We also compare to two offline methods: an upper-bound "joint" version trained on all tasks jointly, and a lower-bound "separate" version with each task trained separately (no transfer). Further baseline results are in Appendix I. The combination of GVCL on task-specific FiLM layers (GVCL-F) outperforms baselines on the smaller-scale benchmarks and outperforms or performs within statistical error of baselines on the larger Mixed Vision benchmark. We also report calibration curves, showing that GVCL-F is well-calibrated. Full experimental protocol and hyperparameters are reported in Appendix H.

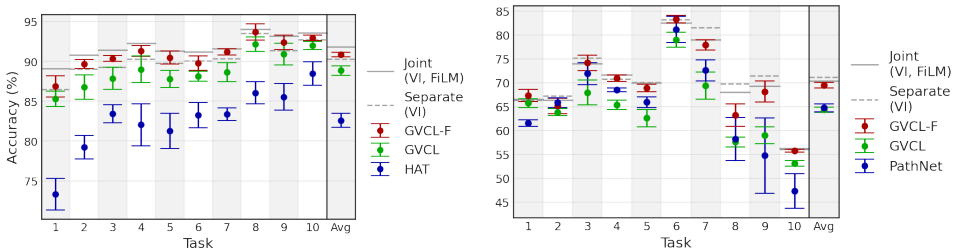
5.1 CHASY and *Split*-MNIST

The CHASY benchmark consists of a set of tasks specifically designed for multi-task and continual learning, with detailed explanation in Appendix J. It is derived from the HASYv2 dataset (Thoma, 2017), which consists of 32x32 handwritten latex characters. Easy-CHASY was designed to maximize transfer between tasks and consists of similar tasks with 20 classes for the first task, to 11 classes for the last. Hard-CHASY represents scenarios where tasks are very distinct, where tasks range from 18 to 10 classes. Both versions have very few samples per class. Testing our algorithm on these datasets tests two extremes of the continual learning spectrum. For these two datasets we use a small convolutional network comprising two convolutions layers and a fully connected layer. For our *Split*-MNIST experiment, in addition to the standard 5 binary classification tasks for *Split*-MNIST, we add 5 more binary classification tasks by taking characters from the KMNIST dataset



(a) Easy-CHASY (b) Hard-CHASY (c) *Split*-MNIST

Figure 2: Running average accuracy of Easy-CHASY, Hard-CHASY and *Split*-MNIST trained continually. GVCL-F and GVCL are compared to the best performing baseline algorithm. GVCL-F and GVCL both significantly outperform HAT on Easy-CHASY. On Hard-CHASY, GVCL-F still manages to perform as well joint MAP training, while GVCL performs as well as PathNet. In Split-MNIST, GVCL-F narrowly outperforms HAT, with both performing nearly as well as joint training.



(a) Easy-CHASY (b) Hard-CHASY

Figure 3: Accuracy of Easy-CHASY and Hard-CHASY trained models at the end of learning all 10 tasks continually. Performance of GVCL-F, GVCL and the best performing baselines (HAT and Pathnet) are compared to Joint and Separate training. GVCL-F again strongly outperforms the baselines and performs similar to the upper-bound VI joint training.

(Clanuwat et al., 2018). For these experiments we used a 2-layer fully-connected network, as in common in continual learning literature (Nguyen et al., 2018; Zenke et al., 2017).

Figure 2 shows the raw accuracy results. As the CHASY datasets have very few samples per class (16 per class, resulting in the largest task having a training set of 320 samples), it is easy to overfit. This few-sample regime is a key practical use case for continual learning as it is essential to transfer information between tasks. In this regime, continual learning algorithms based on MAP-inference overfit, resulting in poor performance. As GVCL-F is based on a Bayesian framework, it is not as adversely affected by the low sample count, achieving 90.9% accuracy on Easy-CHASY compared to 82.6% of the best performing MAP-based CL algorithm, HAT. Hard-CHASY tells a similar story, 69.1% compared to PathNet's 64.8%. Compared to the full joint training baselines, GVCL-F achieves nearly the same accuracy (Figure 3). The gap between GVCL-F and GVCL is larger for Easy-CHASY than for Hard-CHASY, as the task-specific adaptation that FiLM layers provide is more beneficial when tasks require contrasting features, as in Hard-CHASY. With *Split*-MNIST, GVCL-F also reaches the same performance as joint training, however it is difficult to distinguish approaches on this benchmark as many achieve near maximal accuracy.

5.2 *Split*-CIFAR

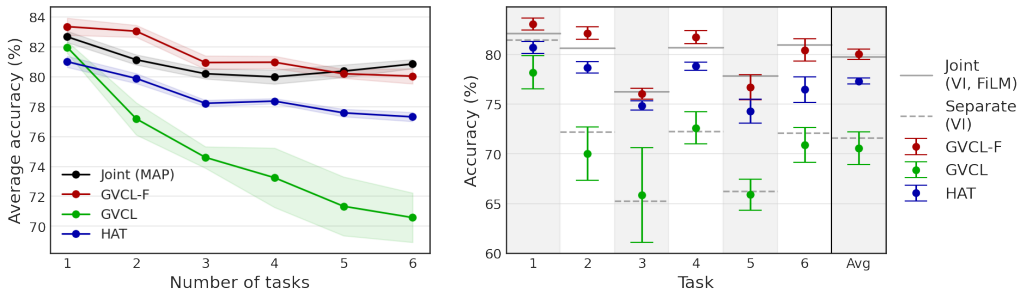
The popular *Split*-CIFAR dataset, introduced in Zenke et al. (2017), has CIFAR10 as the first task, and then 5 tasks as disjoint 10-way classifications from the first 50 classes of CIFAR100, giving a total of 6 tasks. We use the same architecture as in other papers (Zenke et al., 2017; Pan et al., 2020). Like with Easy-CHASY, jointly learning these tasks significantly outperforms networks separately trained on the tasks, indicating potential for forward and backward transfer in a continual learning algorithm. Results are in Figure 4. GVCL-F is able to achieve the same final accuracy as joint training with FiLM layers, achieving 80.0±0.5%, beating all baseline algorithms by at least 2%.

		GVCL-F	GVCL	HAT	PathNet	VCL	Online EWC
Easy-CHASY	ACC (%)	$\mathbf{90.9 \pm 0.3}$	88.9 ± 0.6	82.6 ± 0.9	82.4 ± 0.9	78.4 ± 1.0	73.4 ± 3.4
	BWT (%)	$\mathbf{0.2 \pm 0.1}$	-0.8 ± 0.4	-1.6 ± 0.6	0.0 ± 0.0	-4.1 ± 1.2	-8.9 ± 2.9
	FWT (%)	$\mathbf{0.4 \pm 0.3}$	-0.6 ± 0.5	$\mathbf{-0.2 \pm 1.4}$	-2.1 ± 0.9	-7.9 ± 0.8	-2.1 ± 0.5
Hard-CHASY	ACC (%)	$\mathbf{69.5 \pm 0.6}$	64.4 ± 0.6	62.5 ± 5.4	64.8 ± 0.8	45.8 ± 1.4	56.4 ± 1.7
	BWT (%)	$\mathbf{-0.1 \pm 0.1}$	-0.6 ± 0.2	-0.8 ± 0.4	$\mathbf{0.0 \pm 0.0}$	-11.9 ± 1.6	-7.1 ± 1.7
	FWT (%)	$\mathbf{-1.6 \pm 0.7}$	-6.3 ± 0.6	$\mathbf{-4.0 \pm 5.5}$	$\mathbf{-2.5 \pm 0.8}$	-13.5 ± 2.2	-3.7 ± 1.3
Split-MNIST (10 Tasks)	ACC (%)	$\mathbf{98.6 \pm 0.1}$	94.6 ± 0.7	98.3 ± 0.1	95.2 ± 1.8	92.4 ± 1.2	94.0 ± 1.4
	BWT (%)	$\mathbf{0.0 \pm 0.0}$	-4.0 ± 0.7	-0.2 ± 0.0	$\mathbf{0.0 \pm 0.0}$	-5.5 ± 1.1	-3.8 ± 1.4
	FWT (%)	$\mathbf{-0.1 \pm 0.1}$	$\mathbf{-0.0 \pm 0.0}$	-0.1 ± 0.1	-3.3 ± 1.8	-0.8 ± 0.1	-0.8 ± 0.1
Split-CIFAR	ACC (%)	$\mathbf{80.0 \pm 0.5}$	70.6 ± 1.7	77.3 ± 0.3	68.7 ± 0.8	56.2 ± 2.8	77.1 ± 0.2
	BWT (%)	$\mathbf{-0.3 \pm 0.2}$	-2.3 ± 1.4	$\mathbf{-0.1 \pm 0.1}$	$\mathbf{0.0 \pm 0.0}$	-19.5 ± 3.2	-0.5 ± 0.3
	FWT (%)	$\mathbf{8.8 \pm 0.5}$	1.3 ± 1.0	6.8 ± 0.2	-1.9 ± 0.8	4.1 ± 0.8	6.9 ± 0.3
Mixed Vision Tasks	ACC (%)	$\mathbf{80.0 \pm 1.2}$	49.0 ± 2.8	$\mathbf{80.3 \pm 1.0}$	76.8 ± 2.0	26.9 ± 2.1	62.8 ± 5.2
	BWT (%)	$\mathbf{-0.9 \pm 1.3}$	-13.1 ± 1.6	$\mathbf{-0.1 \pm 0.1}$	$\mathbf{0.0 \pm 0.0}$	-35.0 ± 5.6	-18.7 ± 5.8
	FWT (%)	$\mathbf{-4.8 \pm 1.6}$	-23.5 ± 3.4	$\mathbf{-5.8 \pm 1.0}$	-9.5 ± 2.0	-23.7 ± 3.8	$\mathbf{-4.8 \pm 0.7}$

Table 1: Performance metrics of GVCL-F and GVCL compared to baselines (more in Appendix I). GVCL-F obtains the best accuracy and backwards/forwards transfer on many datasets/architectures.

This confirms that our algorithm performs well in larger settings as well as the previous smaller-scale benchmarks, with minimal forgetting. While the backwards transfer metric for many of the best performing continual learning algorithms is near 0, GVCL-F has the highest forward transfer, achieving 8.5%.

GVCL consistently outperforms VCL, but unlike in the CHASY experiments, it does not outperform Online EWC. This also occurs in the Mixed Vision tasks considered next. Theoretically this should not happen, but GVCL's hyperparameter search found $\beta = 0.2$ which is far from Online EWC. We believe this is because optimizing the GVCL cost for small β is more challenging. However, since intermediate β settings result in more pruning, FiLM layers then bring significant improvement.



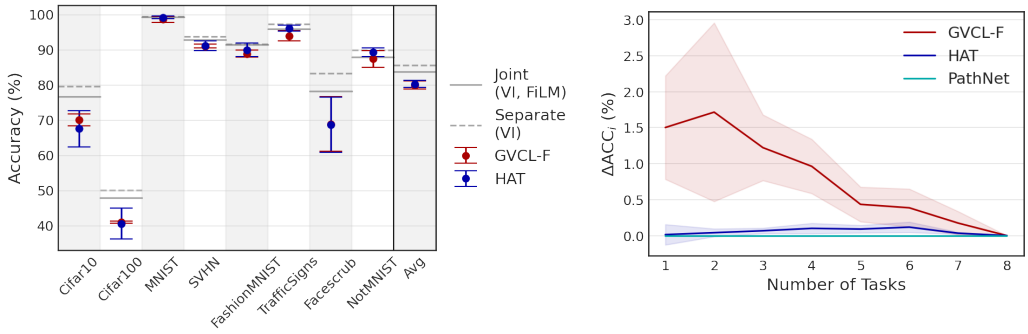
(a) Running average accuracy of *Split*-CIFAR

(b) Final accuracies on *Split*-CIFAR

Figure 4: Running average accuracy of *Split*-CIFAR and final accuracies after continually training on 6 tasks for GVCL-F, GVCL, and HAT. GVCL-F achieves the maximum amount of forwards transfer, and achieves close to the upper-bound joint performance.

5.3 Mixed Vision Tasks

We finally test on a set of mixed vision datasets, as in Serrà et al. (2018). This benchmark consists of 8 image classification datasets with 10-100 classes and a range of dataset sizes, with the order of tasks randomly permuted between different runs. We use the same AlexNet architecture as in Serrà et al. (2018). Average accuracies of the 8 tasks after continual training are shown in Figure 5. GVCL-F's final accuracy matches that of HAT, with similar final performances of 80.0±1.2% and 80.3±1.0% for the two methods, respectively. Figure 5b shows the relative accuracy of the model after training on intermediate tasks compared to its final accuracy. A positive relative accuracy after t tasks means that the method performs better on the tasks seen so far than it does on the same tasks after seeing all 8 tasks (Appendix H contains a precise definition). HAT achieves its continual

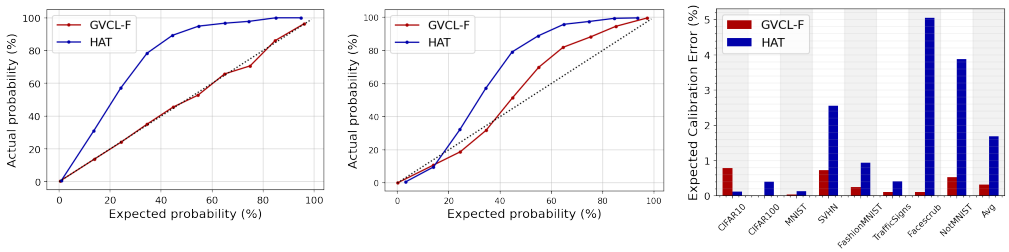


(a) Final accuracies after all tasks

(b) Relative accuracy after training on the ith task

Figure 5: (a) Average accuracy of mixed vision tasks at the end of training for GVCL-F and HAT. Both algorithms perform nearly equally well in this respect. (b) GVCL-F gracefully forgets, with higher intermediate accuracies, while HAT has a lower initial accuracy but does not forget.

learning performance by compressing earlier tasks, hindering their performance in order to reserve capacity for later tasks. In contrast, GVCL-F attempts to maximize the performance for early tasks, but allows performance to gradually decay, as shown by the gradually decreasing relative accuracy in Figure 5b. While both strategies result in good final accuracy, one could argue that pre-compressing a network in anticipation of future tasks which may or may not arrive is an impractical real-world strategy, as the number of total tasks may be unknown *a priori*, and therefore one does not know how much to compress the network. The approach taken by GVCL-F is then more desirable, as it ensures good performance after any number of tasks, and frees capacity by "gracefully forgetting".



(a) Cifar100 Calibration Curve

(b) Facescrub Calibration Curve

(c) ECE on individual tasks

Figure 6: Calibration curves and Expected Calibration Error for GVCL-F and HAT trained on the Mixed Vision Tasks benchmark. GVCL-F achieves much lower Expected Calibration Error, attaining a value averaged across all tasks of 0.3% compared to HAT's 1.7%.

Uncertainty calibration. As GVCL-F is based on a probabilistic framework, we expect it to have good uncertainty calibration compared to other baselines. We show this for the Mixed Vision tasks in Figure 6. Overall, the average Expected Calibration Error for GVCL-F (averaged over tasks) is 0.32%, compared to HAT's 1.69%, with a better ECE on 7 of the 8 tasks. These results demonstrate that GVCL-F is generally significantly better calibrated than HAT, which can be extremely important in decision critical problems where networks must know when they are likely to be uncertain.

6 Conclusions

We have developed a framework, GVCL, that generalizes Online EWC and VCL, and we combined it with task-specific FiLM layers to mitigate the effects of variational pruning. GVCL with FiLM layers outperforms strong baselines on a number of benchmarks, according to several metrics. Future research might combine GVCL with memory replay methods, or find ways to use FiLM layers when task ID information is unavailable.

REFERENCES

Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless Fowlkes, Stefano Soatto, and Pietro Perona. Task2Vec: Task Embedding for Meta-Learning. *arXiv:1902.03545 [cs, stat]*, February 2019. URL http://arxiv.org/abs/1902.03545. arXiv: 1902.03545.

Alessandro Achille, Giovanni Paolini, and Stefano Soatto. Where is the information in a deep neural network?, 2020.

Tameem Adel, Han Zhao, and Richard E. Turner. Continual Learning with Adaptive Weights (CLAW). *arXiv:1911.09514 [cs, stat]*, June 2020. URL http://arxiv.org/abs/1911.09514. arXiv: 1911.09514.

Hongjoon Ahn, Sungmin Cha, Donggyu Lee, and Taesup Moon. Uncertainty-based Continual Learning with Adaptive Regularization. *arXiv:1905.11614 [cs, stat]*, November 2019. URL http://arxiv.org/abs/1905.11614. arXiv: 1905.11614.

Alexander A. Alemi, Ben Poole, Ian Fischer, Joshua V. Dillon, Rif A. Saurous, and Kevin Murphy. Fixing a Broken ELBO. *arXiv:1711.00464 [cs, stat]*, February 2018. URL http://arxiv.org/abs/1711.00464. arXiv: 1711.00464.

Tarin Clanuwat, Mikel Bober-Irizar, A. Kitamoto, A. Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature. *ArXiv*, abs/1812.01718, 2018.

Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei Rusu, Alexander Pritzel, and Daan Wierstra. Pathnet: Evolution channels gradient descent in super neural networks. 01 2017.

Robert M. French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128–135, April 1999. ISSN 1364-6613. doi: 10.1016/S1364-6613(99)01294-2. URL http://www.sciencedirect.com/science/article/pii/S1364661399012942.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. β-VAE: LEARNING BASIC VISUAL CONCEPTS WITH A CONSTRAINED VARIATIONAL FRAMEWORK. pp. 22, 2017.

J. Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, J. Veness, G. Desjardins, Andrei A. Rusu, K. Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, C. Clopath, D. Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114:3521 – 3526, 2017.

Sang-Woo Lee, Jin-Hwa Kim, Jaehyun Jun, Jung-Woo Ha, and Byoung-Tak Zhang. Overcoming Catastrophic Forgetting by Incremental Moment Matching. *arXiv:1703.08475 [cs]*, January 2018. URL http://arxiv.org/abs/1703.08475. arXiv: 1703.08475.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient Episodic Memory for Continual Learning. *arXiv:1706.08840 [cs]*, November 2017. URL http://arxiv.org/abs/1706.08840. arXiv: 1706.08840.

Christos Louizos, Karen Ullrich, and Max Welling. Bayesian compression for deep learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 30*, pp. 3288–3298. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/6921-bayesian-compression-for-deep-learning.pdf.

James Martens. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 21(146):1–76, 2020. URL http://jmlr.org/papers/v21/17-678.html.

Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Variational dropout sparsifies deep neural networks. volume 70 of *Proceedings of Machine Learning Research*, pp. 2498–2507, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL http://proceedings.mlr.press/v70/molchanov17a.html.

Cuong V. Nguyen, Yingzhen Li, Thang D. Bui, and Richard E. Turner. Variational continual learning, 2018.

Manfred Opper and Cedric Archambeau. The variational gaussian approximation revisited. *Neural computation*, 21:786–92, 10 2008. doi: 10.1162/neco.2008.08-07-592.

Kazuki Osawa, Siddharth Swaroop, Anirudh Jain, Runa Eschenhagen, Richard E. Turner, Rio Yokota, and Mohammad Emtiyaz Khan. Practical Deep Learning with Bayesian Principles. *arXiv:1906.02506 [cs, stat]*, October 2019. URL http://arxiv.org/abs/1906.02506. arXiv: 1906.02506.

Pingbo Pan, Siddharth Swaroop, Alexander Immer, Runa Eschenhagen, Richard E. Turner, and Mohammad Emtiyaz Khan. Continual Deep Learning by Functional Regularisation of Memorable Past. *arXiv:2004.14070 [cs, stat]*, June 2020. URL http://arxiv.org/abs/2004.14070. arXiv: 2004.14070.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual Reasoning with a General Conditioning Layer. *arXiv:1709.07871 [cs, stat]*, December 2017. URL http://arxiv.org/abs/1709.07871. arXiv: 1709.07871.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. *arXiv:1705.08045 [cs, stat]*, November 2017. URL http://arxiv.org/abs/1705.08045. arXiv: 1705.08045.

James Requeima, Jonathan Gordon, John Bronskill, Sebastian Nowozin, and Richard E. Turner. Fast and Flexible Multi-Task Classification Using Conditional Neural Adaptive Processes. *arXiv:1906.07697 [cs, stat]*, January 2020. URL http://arxiv.org/abs/1906.07697. arXiv: 1906.07697.

Hippolyt Ritter, Aleksandar Botev, and David Barber. Online structured laplace approximations for overcoming catastrophic forgetting, 05 2018.

Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive Neural Networks. *arXiv:1606.04671 [cs]*, September 2016. URL http://arxiv.org/abs/1606.04671. arXiv: 1606.04671.

Jonathan Schwarz, Jelena Luketina, Wojciech M. Czarnecki, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & Compress: A scalable framework for continual learning. *arXiv:1805.06370 [cs, stat]*, July 2018. URL http://arxiv.org/abs/1805.06370. arXiv: 1805.06370.

Joan Serrà, Dídac Surís, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. *CoRR*, abs/1801.01423, 2018. URL http://arxiv.org/abs/1801.01423.

Alexander Smola, Vishy Vishwanathan, and Eleazar Eskin. Laplace propagation. 01 2003.

Siddharth Swaroop, Cuong V. Nguyen, Thang D. Bui, and Richard E. Turner. Improving and Understanding Variational Continual Learning. *arXiv:1905.02099 [cs, stat]*, May 2019. URL http://arxiv.org/abs/1905.02099. arXiv: 1905.02099.

Martin Thoma. The HASYv2 dataset. *arXiv:1701.08380 [cs]*, January 2017. URL http://arxiv.org/abs/1701.08380. arXiv: 1701.08380.

Brian Trippe and Richard Turner. Overpruning in Variational Bayesian Neural Networks. *arXiv:1801.06230 [stat]*, January 2018. URL http://arxiv.org/abs/1801.06230. arXiv: 1801.06230.

R. Turner and M. Sahani. Two problems with variational expectation maximisation for time-series models. 2011.

Florian Wenzel, Kevin Roth, Bastiaan S. Veeling, Jakub Świątkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How Good is the Bayes Posterior in Deep Neural Networks Really? *arXiv:2002.02405 [cs, stat]*, July 2020. URL http://arxiv.org/abs/2002.02405. arXiv: 2002.02405.

Dong Yin, Mehrdad Farajtabar, and Ang Li. SOLA: Continual Learning with Second-Order Loss Approximation. *arXiv:2006.10974 [cs, stat]*, June 2020. URL http://arxiv.org/abs/2006.10974. arXiv: 2006.10974.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual Learning Through Synaptic Intelligence. *arXiv:1703.04200 [cs, q-bio, stat]*, June 2017. URL http://arxiv.org/abs/1703.04200. arXiv: 1703.04200.

Guodong Zhang, Shengyang Sun, David Duvenaud, and Roger Grosse. Noisy Natural Gradient as Variational Inference. *arXiv:1712.02390 [cs, stat]*, February 2018. URL http://arxiv.org/abs/1712.02390. arXiv: 1712.02390.