Bayesian Neural Networks with Experience Replay for Continual Learning

Anuj Srivastava Debmalya Chatterjee Shubh Goel Utkarsh Gupta

1. Introduction

Continual learning models aim to adapt to new tasks and evolving environments over time. However, when trained sequentially on datasets with varying distributions, neural networks often face catastrophic forgetting, where previously acquired knowledge is lost. Methods that address catastrophic forgetting fall largely into two categories: experience replay and regularization, or a combination of both.

In experience replay, a coreset of data from previous tasks is maintained in a memory buffer, which is used to finetune the continually updating model to avoid forgetting previous tasks. Gradient episodic memory (GEM) (Lopez-Paz & Ranzato, 2022) uses this idea to store a subset of the observed examples from a particular task at the end of each episode to be used later to prevent gradient updates from deviating from their previous values. (Buzzega et al., 2020) also store logits, and minimize divergence between stored and updated logits in addition to the loss for new tasks, and (Yan et al., 2022) use intermediate layer features in addition. (Shin et al., 2017) and (Farquhar & Gal, 2019) also evaluate learning generative models to sample data from previous tasks for replay.

In the regularization-based approaches (Kirkpatrick et al., 2017), (Zenke et al., 2017), significant changes to the representation learned for previous tasks are prevented. This can be performed through regularizing the objective function or is directly enforced on weight parameters. Typically, this objective function is engineered to represent the importance of each parameter.

Regularization can be generalized to Bayesian inference since prior distributions can specify the regularizer, and continual learning can be formulated as a posterior update: $p(\theta|\mathcal{D}_{1:t}) \propto p(\theta|\mathcal{D}_{1:t-1})p(\mathcal{D}_t|\theta)$, where θ are model parameters and $\mathcal{D}_t$ is the t^{th} task. (Ebrahimi et al., 2020) use variational inference to approximate the posterior and adapt the learning rate proportional to the uncertainty (measured as the standard deviation) in the posterior probability distribution of the weights in the network. (Li et al., 2021) maintains a Gaussian mixture model by adding a mixture component for each new task, while continually merging components to reduce the mixture-size.

Further, (Farquhar & Gal, 2019) present a framework combining Bayesian continual learning with experience replay using random coresets. (Farquhar & Gal, 2019), (Kessler et al., 2023), and (Henning et al., 2021) evaluate such combinations using coresets that are selected from the original datasets by simple reservoir sampling. (Yan et al., 2022) learn Bayesian sparse networks and fill their memory buffer with loss-aware reservoir sampling, i.e., class-balanced samples with diverse loss values.

Contributions. In this work, we investigate the impact of Bayesian coresets and pseudocoresets for experience replay in continual learning, as an alternative to using random samples. Specifically, we examine their role in mitigating catastrophic forgetting when combined with recursive Bayesian updates under the framework of Bayesian Continual Learning. Bayesian coresets (Campbell & Beronov, 2019) are weighted subsets $\widetilde{\mathcal{D}}$ of the original dataset $\mathcal{D}$ that result in a posterior distribution closest to the original, by minimizing the following KL-divergence: $\min_{\widetilde{\mathcal{D}}} D_{KL}(\, p(\theta|\widetilde{\mathcal{D}}) \,||\, p(\theta|\mathcal{D})\,)$. (Manousakas et al., 2020) generalize this to Bayesian pseudocoresets where $\widetilde{\mathcal{D}}$ can be a synthetic dataset, albeit for shallow models only. Recent advancements by (Tiwary et al., 2024) enable pseudocoreset construction for Bayesian neural networks using contrastive divergence. Building on these foundations, we propose a simple algorithm for generating both Bayesian coresets and pseudocoresets. Our implementation and experiments are publicly available here.

2. Method

We build upon the codebase of (Ebrahimi et al., 2020) which is our primary baseline (UCB, Uncertainty-guided Continual learning with Bayesian NNs). In UCB, for task t, a variational approximation (diagonal Gaussian) $q_t(\theta)$ is optimized for $\min_{q_t} D_{KL}(\, q_t(\theta) \,||\, p(\theta|\mathcal{D}_t)\,)$, and q_t is initialized with q_{t-1}. Additionally, the relative learning rate of each parameter scaled proportional to its estimated standard deviation (hence "uncertainty-guided") – so as to encourage utilizing less important parameters for subsequent tasks. We do not include this in our proposed method.

We denote by Bayesian continual learning (BCL) our own baseline. In BCL, q_t is optimized for $\min_{q_t} D_{KL}(\; q_t(\theta) \;||\; p(\theta|\mathcal{D}_{1:t})\;)$, by approximating $p(\theta|\mathcal{D}_{1:t}) \approx \frac{1}{Z} p(\mathcal{D}_t|\theta) q_{t-1}(\theta)$, where Z is the appropri-

ate normalizing factor. Contrary to UCB, where the prior is fixed (standard normal), in BCL the prior for task t is the posterior from task $t-1$.

As part of our ablation study, we use BCL with uniformly sampled coresets (BCL+UC). The coreset size is fixed at 5% of dataset size for each previous task in a replay buffer. After each epoch in task t, we train one epoch on the coresets of each task from 1 to $t-1$.

As a skyline, we modify the above to sample a different random subset for the replay buffer instead of a fixed coreset. This avoids overfitting to a much smaller coreset, giving us a rough upper bound for replay-based methods. For this, we sample subsets of the same size as BCL+UC (5%) to avoid increasing training time. We call this method BCL+USPE (uniform sampling per epoch).

In our proposed method, we use Bayesian coresets (BCL+BC) or pseudocoresets (BCL+BPC) in the replay buffer. (Campbell & Beronov, 2019) developed an algorithm to generate Bayesian coresets for linear models, but their complexity varied quadratically with coreset size – as they iteratively add the next best sample to the coreset and each iteration involves posterior estimation on the coreset. We simplify this by uniformly sampling a subset $S \subset \mathcal{D}_t$, and then optimizing weights w_i of each sample $(x_i, y_i) \in S$ to obtain the coreset. The weights contribute to the log-likelihoods as: $\log p(S, w|\theta) = \sum_i w_i \log p(y_i|x_i, \theta)$.

Essentially, we need to minimize the following objective:

$$\min_w D_{KL}(\, p(\theta|S, w) \,||\, p(\theta|\mathcal{D}_t) \,). \tag{1}$$

We generate the coreset for task t after training on D_t, thus we can use $q_t(\theta)$ instead of $p(\theta|\mathcal{D}_t)$:

$$\min_w D_{KL}(\, p(\theta|S, w) \,||\, q_t(\theta) \,). \tag{2}$$

Since $p(\theta|S, w)$ is intractable we approximate it with a transient VI network q', to get the following objective:

$$\min_{w,q'} D_{KL}(\, q'(\theta) \,||\, p(\theta|S, w) \,) + D_{KL}(\, q'(\theta) \,||\, q_t(\theta) \,). \tag{3}$$

We apply bayes rule to expand the coreset posterior:

$$p(\theta|S, w) = \frac{p(S, w|\theta) q_{t-1}\theta}{\int p(S, w|\theta') q_{t-1}\theta' d\theta'} \tag{4}$$

With this expansion, we can rewrite the first term in Equation (3) as: $D_{KL}(q'(\theta)||p(\theta|S, w)) = D_{KL}(q'||q_{t-1}) - \mathbb{E}_q[\log p(S, w|\theta)] + \log \mathbb{E}_{q_{t-1}}[\log p(S, w|\theta)]$. We use Monte Carlo samples to compute the expectations, and use `logsumexp` of PyTorch[1] for the final term. This method is denoted by BCL+BC.

[1] https://pytorch.org/docs/stable/generated/torch.logsumexp.html

Algorithm 1 BCL+UC/BC/BPC

Input: Datasets for tasks $\mathcal{D}_{1:T}$, Prior $q_0(\theta)$, number of epochs n_e.
for $t = 1$ **to** T **do**
  Set prior $p = q_{t-1}$
  Train BNN
  for $e = 1$ **to** n_e **do**
    Train one epoch: $\min_{q_t} D(q_t(\theta)||p(\theta|\mathcal{D}_t))$
    for $u = 1$ **to** t **do**
      Train one epoch: $\min_{q_t} D(q_t(\theta)||p(\theta|C_u))$
    end for
  end for
  Generate coreset C_t: sample randomly for UC / by Equation (3) for BC/BPC
end for

By making the inputs of the coreset x_i trainable, we are able to obtain synthetic data, i.e. pseudocoresets. This method is denoted by BCL+BPC. Our algorithm is summarized in Algorithm 1.

3. Experiments

3.1. Experimental Setup

Datasets. We evaluate the performance of our method in the case of class incremental continual learning of a single or two randomly alternating datasets where each task covers only a subset of the classes in the dataset. We use Split MNIST (LeCun et al., 1998) with 5 tasks (5-Split MNIST) and 2 tasks (2-Split MNIST), Permuted MNIST and Alternating CIFAR10 and CIFAR100 (Krizhevsky & Hinton, 2009) with simillar experimental settings as (Ebrahimi et al., 2019). In 5-Split MNIST we learn the digits 0-9 in five tasks with 2 classes at a time in 5 pairs of $0/1$, $2/3$, $4/5$, $6/7$, and $8/9$. In Permuted MNIST, each task is considered as a random permutation of the original MNIST pixels. Following (Ebrahimi et al., 2019), we learn a sequence of 10 random permutations and report average accuracy at the end. For the Alternating CIFAR10 and CIFAR100 experiment, we randomly alternate between class incremental learning of CIFAR10 and CIFAR100. Both datasets are divided into 5 tasks each with 2 and 20 classes per task, respectively.

Baselines. We compare our method with the following approaches in the Continual Learning literature:

- **UCB** (Ebrahimi et al., 2019): Uncertainty-guided Continual Bayesian Neural Networks (UCB), where the learning rate is adapted according to the uncertainty defined in the probability distribution of the weights in networks.
- **PR-BbB** (Henning et al., 2021): Posterior Meta-

Replay which proposed a framework where task-conditioned posterior parameter distributions are continually learned and compressed in a hypernetwork.

- **VCL** (Nguyen et al., 2017): Variational Continual Learning, a framework for continual learning that fuses online variational inference (VI) and recent advances in Monte Carlo VI for neural networks.
- **ProtoCL** (Zhang et al., 2019): Prototypical Bayesian Continual Learning in which the authors explicilty modeled the generative process of the data.

The results for UCB (Ebrahimi et al., 2019) are obtained using their provided code run on reduced number of epochs (due to computational bottleneck and to ensure even comparison with our method). For the other baselines, the results are reported from their original work without re-implementation.

Training details. Following (Ebrahimi et al., 2019), we used stochastic gradient descent with a batch size of 64 and a learning rate of 0.01, decaying it by a factor of 0.3 once the loss plateaued. We trained each task for 5, 10, 20, and 20 epochs for 5-Split MNIST, Permuted MNIST, 2-Split MNIST and Alternating CIFAR$10/100$ respectively. No pre-trained model is used in all experiments. For MNIST datasets we use MLP with one hidden layer of 1200 neurons and a classification layer. For CIFAR$10/100$ dataset we use ResNet18 (He et al., 2016). For experience replay use a buffer size of 5%.

Performance Metric. We report the average test classification accuracy across all tasks once the model has finished sequentially learning. Formally,

$$ACC = \frac{1}{n}\sum_{i=1}^{n} R_{i,n} \tag{5}$$

where $R_{i,n}$ is the test classification accuracy on task i after sequentially finishing learning the n^{th} task where n is the total number of tasks.

3.2. Results

In Table 3.2 we summarize the performance of our proposed method and the baselines on various datasets. In Figure 1, we plot the accuracy on task i after the completion of task j of the Permuted MNIST dataset for our proposed methods. Note that for PR-BbB (Henning et al., 2021) we report the results of their small MLP (2 layers 100 unit each) and ResNet18 models on Split MNIST and CIFAR100 dataset respectively to ensure fair comparison. For CIFAR10, they have only reported results for ResNet32 which we mention here as it is.

Split MNIST. We observe that just using Bayesian Continuous Learning (BCL) outperforms every other baseline in all the variants of of Split MNIST dataset. Note that we get this performance using significantly less number of epochs for training as compared to other baselines. PR-BbB (Henning et al., 2021), ProtoCL (Zhang et al., 2019) and VCL (Nguyen et al., 2017) which are also based on Variational Inference techniques use 60, 50 and 120 epochs respectively for training on 5-Split MNIST Dataset

Alternating CIFAR10 and CIFAR100. We outperform UCB (Ebrahimi et al., 2019) on the alternating CIFAR10/100 dataset. Other baseline methods have not reported results for this specific task; instead, their evaluations have been conducted separately on the Split CIFAR10 and Split CIFAR100 datasets. To enable a meaningful comparison, we present the following metrics for UCB (Ebrahimi et al., 2019) and our proposed method:

$$ACC_{\texttt{CIFAR10}} = \frac{1}{5}\sum_{i=1}^{n} R_{i,n}\mathbf{1}_{\texttt{CIFAR10}}(i) \tag{6}$$

$$ACC_{\texttt{CIFAR100}} = \frac{1}{5}\sum_{i=1}^{n} R_{i,n}\mathbf{1}_{\texttt{CIFAR100}}(i) \tag{7}$$

where $\mathbf{1}_A(x)$ equals 1 if $x \in A$ (see Performance Metric). Note that due to computational bottleneck we trained our models for 20 epochs ($\approx$ 24 hours on NVIDIA GeForce GTX 1080) on combined CIFAR10/100. In contrast, PR-BbB (Henning et al., 2021) use 60 epochs on CIFAR10 and 200 epochs on CIFAR100, while ProtoCL (Zhang et al., 2019) use 50 epochs and 200 epochs on CIFAR10 and CIFAR100 respectively. Despite the reduced training duration, our method outperforms ProtoCL (Zhang et al., 2019) on both the CIFAR10 dataset and CIFAR100 dataset. Additionally, we approach the performance of PR-BbB (Henning et al., 2021) on the CIFAR10 dataset. These results highlight the effectiveness of our approach in mitigating catastrophic forgetting, even on complex datasets.

Effect of Experience Replay. We observe that using replay buffer of previous task's dataset along with BCL improved the performance in general. BCL+USPE has the highest performance across all datasets which is intuitive since it uniformly samples a different subset of pervious task's data every epoch while training on the current task. Thus in expectation, the model goes through the full dataset of previous tasks. Additionally, it has to maintain the full data of all the previous tasks which is memory inefficient. On the other hand BCL+(US/BC/BPC) use a fixed subset of a particular task for replaying in the future. We see BCL+BPC and BCL+BC have simillar performance as BCL+UC on the 2-Split MNIST and 5-Split MNIST datasets. However these are easy tasks where the scope of improvement is low. On a more difficult task like Permuted MNIST , they have a slighly better performnce. This is also observed in the CIFAR dataset where BCL+BC has a noticeable improvement in the performance over BCL+UC. Interestingly, BCL

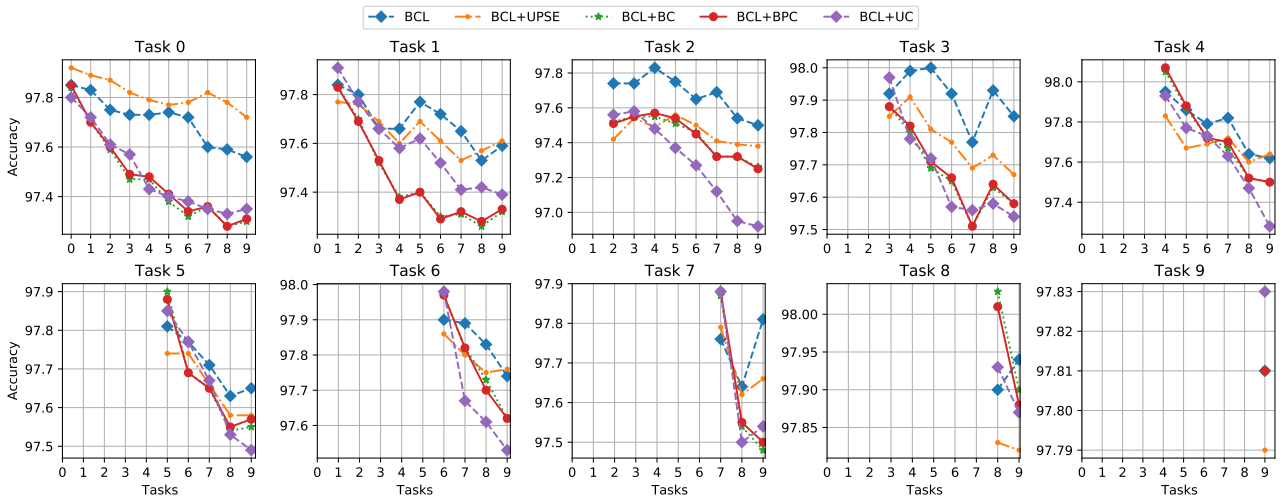

Figure 1. Accuracy on Permuted MNIST tasks for our proposed continual learning methods. For each task i, we have plotted accuracy on it after training of task j is completed.

Model	2-Split MNIST	5-Split MNIST	Permuted MNIST	CIFAR10/100	CIFAR10	CIFAR100
UCB	97.88%	98.45%	93.51%	35.58%	63.63%	7.53%
PR-BbB	N/A	98.75%	94.41%	N/A	**92.61%**	**84.97%**
ProtoCL	N/A	94.78%	N/A	N/A	57.91%	25.4%
VCL	N/A	98.4%	95.5%	N/A	N/A	N/A
BCL	98.81%	99.56%	**97.71%**	**48.28%**	68.83%	27.73%
BCL+USPE	**99.12%**	**99.65%**	97.66%	46.45%	73.47%	19.42%
BCL+UC	99.03%	99.63%	97.47%	37.38%	60.55%	14.21%
BCL+BC	99.05%	99.62%	97.53%	43.36%	70.48%	16.05%
BCL+BPC	99.05%	99.62%	97.54%	38.12%	64.68%	9.61%

Table 1. Comparison of average final accuracies across datasets and approaches. BCL+BC performs better than or at par with BCL+UC

peforms signfificantly better compared to its coreset-based counterparts on CIFAR dataset. This may be attributed to the use of ResNet18 architecture having close to 11M parameters. Retraining such a large model on a small coreset may have induced overfitting, thereby interfering with the effective learning of the current task.

4. Conclusion

We experimented with the use of Bayesian coresets (BC) and pseudocoresets (BPC) instead of uniformly sampled coresets (UC) in the Bayesian continual learning with experience replay setting, and found that generally BCs perform better than or at par with UCs. Our work sets a foundation for better utilization of Bayesian models in continual learning, by showing they can also be used to obtain better coresets for the replay buffer.

Contrary to our expectation, we found that BCL without experience replay outperforms replay-based methods on the CIFAR10/100 and PermutedMNIST tasks. We suspect this may be due to either interference from training on past coresets not allowing convergence on the current task, or simply overfitting on the coreset.

Future work. BPCs underperformed compared to BCs, despite being more expressive. This may be due to them being harder to optimize; future work may look at more sophisticated objectives for BPCs, such as (Tiwary et al., 2024). Our study relies on variational inference for approximating the posteriors. Further investigations could explore use of Markov chain Monte Carlo sampling methods for constructing BCs/BPCs as in (Tiwary et al., 2024).

An important aspect we could not test given limited resources is the effect of the coreset size on different methods, and whether BCs would have a bigger advantage on smaller coreset sizes. Further improvements may be made by better sampling of the initial subset in BCs/BPCs. This is done in (Campbell & Beronov, 2019) at the expense of complexity being quadratic in coreset size.

References

Buzzega, P., Boschini, M., Porrello, A., Abati, D., and CALDERARA, S. Dark experience for general continual learning: a strong, simple baseline. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*,

volume 33, pp. 15920–15930. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/b704ea2c39778f07c617f6b7ce480e9e-Paper.pdf.

Campbell, T. and Beronov, B. Sparse variational inference: Bayesian coresets from scratch. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/7bec7e63a493e2d61891b1e4051ef75a-Paper.pdf.

Ebrahimi, S., Elhoseiny, M., Darrell, T., and Rohrbach, M. Uncertainty-guided continual learning with bayesian neural networks. arxiv 2019. *arXiv preprint arXiv:1906.02425*, 2019.

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

Farquhar, S. and Gal, Y. A unifying bayesian view of continual learning. *CoRR*, abs/1902.06494, 2019. URL http://arxiv.org/abs/1902.06494.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 770–778, 2016.

Henning, C., Cervera, M., D'Angelo, F., Oswald, J. V., Traber, R., Ehret, B., Kobayashi, S., Grewe, B. F., and Sacramento, J. Posterior meta-replay for continual learning. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021.

Kessler, S., Cobb, A., Rudner, T. G. J., Zohren, S., and Roberts, S. J. On sequential bayesian inference for continual learning. *Entropy*, 25(6), 2023. ISSN 1099-4300. doi: 10.3390/e25060884. URL https://www.mdpi.com/1099-4300/25/6/884.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., 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. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114 (13):3521–3526, March 2017. ISSN 1091-6490. doi: 10.1073/pnas.1611835114. URL http://dx.doi.org/10.1073/pnas.1611835114.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Li, H., Barnaghi, P., Enshaeifar, S., and Ganz, F. Continual learning using bayesian neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(9): 4243–4252, 2021. doi: 10.1109/TNNLS.2020.3017292.

Lopez-Paz, D. and Ranzato, M. Gradient episodic memory for continual learning, 2022. URL https://arxiv.org/abs/1706.08840.

Manousakas, D., Xu, Z., Mascolo, C., and Campbell, T. Bayesian pseudocoresets. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 14950–14960. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/ab452534c5ce28c4fbb0e102d4a4fb2e-Paper.pdf.

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

Shin, H., Lee, J. K., Kim, J., and Kim, J. Continual learning with deep generative replay. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/0efbe98067c6c73dba1250d2beaa81f9-Paper.pdf.

Tiwary, P., Shubham, K., Kashyap, V. V., and AP, P. Bayesian pseudo-coresets via contrastive divergence. In *The 40th Conference on Uncertainty in Artificial Intelligence*, 2024. URL https://openreview.net/forum?id=SHsR2VOOKv.

Yan, Q., Gong, D., Liu, Y., van den Hengel, A., and Shi, J. Q. Learning bayesian sparse networks with full experience replay for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 109–118, June 2022.

Zenke, F., Poole, B., and Ganguli, S. Continual learning through synaptic intelligence. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of*

Machine Learning Research, pp. 3987–3995. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/zenke17a.html.

Zhang, M., Wang, T., Lim, J. H., Kreiman, G., and Feng, J. Variational prototype replays for continual learning. *arXiv preprint arXiv:1905.09447*, 2019.