Evaluating Sampling Methods for Generative Replay in Continual Learning

Elliot Tower
Department of Computer Science
University of Massachusetts Amherst
Amherst, MA 01003
etower@umass.edu

Hava Siegelmann
Department of Computer Science
University of Massachusetts Amherst
Amherst, MA 01003
hava@cs.umass.edu

Abstract

Catastrophic forgetting is a pervasive problem for continual learning with neural networks. Generative replay serves to alleviate this issue by replaying to the model generated samples similar to previously seen data. In this work we investigate the the effectiveness of existing sampling methods in terms of test accuracy, forgetting, memory overhead, and computational efficiency. We additionally propose a novel method, Cross Entropy Sampling, which replays samples whose cross entropy loss has increased the most after foreseen model updates. Testing with the state-of-the-art generative replay model, our sampling method results in a 3% increase in accuracy and reduced forgetting across all hyperparameters, with only minor computational overhead and no additional parameters.

1 Introduction

Catastrophic forgetting is a pervasive characteristic of artificial neural networks wherein prior information is rapidly forgotten as new information is continually learned. This phenomena does not occur in human learning. Instead, the hippocampus region in the brain is believed to enable both memory retention and stability through reactivating neural structures (Dhawan, 2020) activated during the original memory procession. The artificial neural network (ANN) equivalents to this mechanism are replay models.

A simple method of adding replay into ANNs is to store the data from all previously encountered tasks, a method known as "exact" or "experience" replay. This naive approach leads to an enormous amount of memory use, and is not biologically plausible (Parisi et al., 2018).

Generative replay uses a generative neural network to create synthetic samples which are similar to the previously seen training data, requiring no explicit storage of data. This approach is inspired by the hippocampal mechanisms in biological learning, especially given the brain's limited storage capabilities.

Continual learning remains an area of extensive focus in neural network research due to the dynamic nature of biological neural networks as compared to the static input/output nature of artificial neural networks (Parisi et al., 2018). Life is far from static, and any increase in performance for continual learning could have widespread ripples in both the efficacy and utility of ANNs. While many ANNs are able to perform well in Task Independent Learning (Task-IL), classifying samples in their original learning window, performance worsens significantly in the Class Independent Learning (Class-IL) setting, distinguishing classes between *different* learning episodes (van de Ven and Tolias, 2019). The Brain-Inspired Replay (BI-R) model is one of the only ANNs which maintains acceptable performance in the difficult Class-IL scenario without explicitly storing data (van de Ven et al., 2020).

2 Related Work

Though some methods introduce alternative sampling methods (Wu et al., 2019; Hayes et al., 2019; Aljundi et al., 2019), most generative replay methods utilize uniform random sampling (Parisi et al., 2018; Hayes et al., 2021). As of writing there are no published works comparing these sampling methods. The best performing generative replay model, BI-R only utilizes random sampling; in this paper we test existing sampling methods as well as our own novel methods, in hopes of improving performance.

The BI-R model featured four biologically inspired improvements to the standard generative replay model (Shin et al., 2017). Internal re-

play allows replay to occur during the internal/hidden layers, inspired by the brain's replaying of higher level representations rather than exact inputs. Replay-through-feedback integrates the generator into the main model via backward/feedback connections, inspired by the brain's non-linear processing. Conditional replay is the creation of specific class modes through a Gaussian mixture (as opposed to a normal prior), allowing the model to generate specific examples, in the same way that humans can choose what they recall from memory. Lastly, context-dependent gating is the inhibition of different subsets of neurons for every layer during the generative backward feedback, simulating the mechanisms in the brain which prevent previous pathways from being accidentally overwritten.

Maximally Interfered Retrieval (MIR) is a recent approach for selectively replaying certain generated samples, both for generative replay and experience replay (Aljundi et al., 2019). In this method, instead of randomly replaying all generating samples, they choose samples for which the model's prediction distribution would be most affected by new incoming samples (had it been trained on these new samples in isolation, without replay). In order to calculate this, a copy is made of the model and samples are classified by the model both before and after training the model solely on the new incoming samples. Samples are chosen which maximize the KL divergence between the predictions before and after updating the model. The intuition behind this is that these samples were most affected by not replaying data, and replaying them ensures they are not 'forgotten'.

3 Proposed Approach

Our approach combines BI-R (van de Ven et al., 2020) with an improved version of the MIR sample selection method (Aljundi et al., 2019). Starting from the BI-R model, we implement various sample selection techniques, including a modified version of MIR.

Specifically, we use the ground truth labels y^* for generated data, which were not available in MIT but are available in the case of BI-R due to the generator having a separate Gaussian mode for each class. This allows us to measure the model's exact loss on the data before and after update, rather than approximating it by measuring the change in prediction distribution. KL divergence penalizes the model for all distribution changes, even if the correct class prediction is not affected, or even increased.

The generators for these two models also work at different abstraction levels: BI-R generates hidden or intermediate representations, whereas MIR generates input level images. It is notably more difficult to generate high quality input level images than intermediate representations, which is a limitation of the MIR approach.

3.1 Problem Formulation

For original classifier $f_{\theta'}$ with latent vector z and generator g_γ, we define $y_{pre} = f_{\theta'}(g_\gamma(z))$. For updated classifier f_{θ^v} we define $\hat{y} = f_{\theta^v}(g_\gamma(z))$. The objective is to find the given feature space data points $z \in Z$ which maximize the difference in loss before and after parameter update:

$$\max_Z \mathcal{L}\big(f_{\theta^v}(g_\gamma(Z)), y^*\big) - \mathcal{L}\big(f_{\theta'}(g_\gamma(Z)), y^*\big) \tag{1}$$

Because the generator in MIR does not provide access to ground-truth labels y^*, they approximate Equation 1 using KL-Divergence, D_{KL}:

$$\underset{Z}{max} \sum_{z \in Z} [D_{KL}(y_{pre} \parallel \hat{y}) - \alpha H(y_{pre})] \tag{2}$$

MIR additionally minimizes an entropy penalty H, encouraging the model to choose samples which it is confident in, and a regularization encouraging variety in samples. In implementation, samples are added iteratively, checking that the latent vector distance between a potential sample z and all currently selected samples $z_1 \dots z_n$ is beyond a threshold ϵ.

In our approach, we calculate Equation 1 directly, using the ground-truth class labels y^*. We perform our sample selection in batch operations in order to parallelize and avoid inefficient iterative code. For a given curation multiplier c and replay batch size b, our model generates $c * b$ samples and replays only the top-b samples, sorting to optimize variety in samples along with cross-entropy loss (Equation 1). Sample variety is done batch-wise using an optimized PyTorch function to calculate pairwise euclidean distance between all latent vectors in the generated batch, sorting by samples with the largest average distance.

4 System, Task, Environment and Behavior

The **system** which we are using to conduct our experiments is the currently best performing generative replay model. The **task** is the Class Independent Learning (Class-IL) setting of continual learning. The **environment** consists of the CIFAR-100 dataset (Krizhevsky, 2009) and the PyTorch machine learning framework (Paszke et al., 2019). The **behavior** we are interested in is average accuracy, per-task test accuracy, and quality of generated samples.

5 Research Questions & Hypotheses

Given that generative replay models do not store data explicitly, and rely on generating samples for each task, we set out to answer three main research questions:

R1. Which classes should we generate samples from? Does the class distribution matter?

R2. Which generated samples should we replay? Is every sample equally helpful?

R3. How does sample selection method affect the scaling of generative replay models?

Following these research questions, we devise three initial hypotheses:

H1. It is favorable to generate samples from previously-seen classes which the model most commonly confuses new data for.

H2. Class variety in replay is important: model performance will degrade if distribution is too far from uniform.

H3. Selectively replaying a higher-quality subset of samples to the model will lead to higher performance than replaying all generated samples.

H4. Selective replay increases performance when scaling to larger batch and replay batch sizes.

6 Methodology

We tested the following basic methods for determining which classes to generate samples for:

1. **Random Sampling**: Select classes to generate samples for randomly.
2. **Uniform Sampling**: Generate an equal number of samples for each class (uniform distribution).
3. **Softmax Sampling**: Before learning Task T, we pass the real samples from Task T into the classifier which has learned the last T-1 tasks. Then, we average the classifier's prediction distribution over the entire batch, and sample using this distribution. The intuition is we want to teach the model to distinguish new samples from the classes which it confuses them for the most.

Further, we tested the following **"Look-Ahead"** sampling methods, inspired by MIR (Aljundi et al., 2019). These methods classify the generated samples with the current model, make a copy of the model and train it using only data from the new task (no replay), and classifies the same generated samples using this updated model.

4. **Misclassified Sampling**: This method replays generated samples which the updated model assigned highest probability of belonging to the new incoming classes. The generated data cannot belong to these classes, so the intuition is that this method trains the model not to confuse samples from old classes with the newly learned class. A similar method to this was used as one of the components of the loss function in MIR.
5. **Cross-Entropy Sampling**: This method measures the cross-entropy loss for each generated sample, replaying samples for which the loss increased the most after update. This penalizes decreases in assigned probability for the correct class.
6. **KL-Divergence Sampling**: This sampling method approximates the change in loss by measuring the Kullback–Leibler distance between predicted output distributions. This penalizes any change in predicted class distribution, and does not require the ground-truth class label for generated samples.

Finally, we apply a **Variety Sampling** augmentation to all Look-Ahead sampling methods. This ensures that the selected samples are sufficiently different from each other. We implement this by calculating the pairwise Euclidean distance between the VAE's latent vector for each venerated

Task	Random	Uniform	Softmax	Missed	Cross Entropy	Cross Entropy Variety	Best
1	**0.034**	0.02	0.024	0.024	0.025	**0.056**	Cross Entropy Variety
2	0.046	**0.058**	0.041	0.042	0.042	**0.084**	Cross Entropy Variety
3	0.056	0.055	0.048	0.071	0.06	**0.094**	Cross Entropy Variety
4	0.075	**0.106**	0.084	0.112	0.102	**0.135**	Cross Entropy Variety
5	0.07	0.069	0.06	0.084	**0.102**	0.092	Cross Entropy
6	0.125	0.12	0.107	0.126	**0.157**	**0.145**	Cross Entropy
7	0.246	0.288	0.233	0.287	0.305	**0.33**	Cross Entropy Variety
8	0.232	0.259	0.241	0.282	**0.301**	0.281	Cross Entropy
9	0.479	0.442	0.458	0.441	**0.479**	0.456	Cross Entropy
10	0.655	**0.667**	0.645	0.633	0.646	0.652	Uniform
AVG	0.2018	0.2084	0.1941	0.2102	0.2219	**.2325**	Cross Entropy Variety

Table 1: Accuracy of Sample Methods on CIFAR-100 Experiment (256 batch size)

Task	Random	Uniform	Softmax	Missed	Cross Entropy	Cross Entropy Variety	Best
1	0.9059	0.9154	**0.913**	0.8837	0.8927	0.9069	Softmax
2	0.8898	0.8898	0.8737	**0.9079**	0.8712	0.8869	Missed
3	0.9098	0.9072	0.9349	**0.9381**	0.9354	0.9184	Missed
4	0.9673	0.9753	0.9703	**0.9789**	0.9708	0.9718	Missed
5	**0.9904**	0.9874	0.9834	0.9743	0.9856	0.9854	Random
AVG	0.9326	0.935	0.935	**0.9366**	0.9312	0.9339	Missed

Table 2: Accuracy of Sample Methods on Split MNIST Experiment (128 batch size).

sample, and selecting those with the highest average distance.

7 Research Design

Testing our first three sample selection methods gives us insight into the first research question R1. Softmax sampling (3) directly tests hypothesis H1, although poor performance may be due to a lack of class variety, as hypothesized by H2.

We answer the second research question R2 by performing ablation studies: replaying different proportions of the generated samples to the model. If performance is maximized by replaying all the generated samples to the model, this indicates that H3 is false.

We answer the third research question R3 through varying batch size and replay batch size, and empirically testing the performance of our method as compared with the base BI-R model.

8 Results

8.1 Sampling Methods

In order to determine the most effective sample selection method, we ran fixed-seed class-incremental learning experiments on two standard datasets: CIFAR-100 (split into 10 tasks, with 10 classes per task) and split-MNIST (split into 5 tasks, with 2 classes per task). Default hyperparameters from BI-R were used. (van de Ven et al., 2020)

As we can see in Table 1, the best performing method on CIFAR-100 Class-IL was Cross Entropy Variety. This method resulted in a final model test accuracy increase from 20.18% to 23.25%, as compared to the standard random sample selection used in BI-R. Compared with the other sampling methods, it had the best performance on 5/10 tasks, with its non-variety counterpart performing best on 4/10 tasks. The addition of variety resulted in a significant 200% increase in accuracy for task 1 and 2, and a roughly 3% increase in tasks 3 and 4. In the remaining tasks, the methods perform similarly, until uniform sampling performs best for the final task.

For Split MNIST, the Misclassified sample method performed the best on 3/5 tasks, being outperformed on the first task by Softmax, and on the final task by Random. However, all these methods perform very similarly, with a variance of 93.12% to 93.66%. This makes sense as this is the easiest scenario, and there is much less variance between the 10 classes of black and white handwritten digits in MNIST versus the 100 different classes of color images in CIFAR-100. This suggests that the benefits of different sampling methods are most relevant for difficult tasks such as CIFAR-100 Class IL.

Method	Batch Size	Replay Batch Size	Accuracy ↑	FID ↓	IS ↑
Random	256	256	21.32%	382.21	40.88
Cross-entropy Variety	256	256	**23.32%** (+2.00)	**355.18**	**41.17**
Random	512	512	22.44%	337.07	38.06
Cross-entropy Variety	512	512	**25.12%** (+2.68)	**304.36**	**36.88**

Table 3: Performance of random sampling and Cross-entropy Variety sampling across batch and replay batch sizes (averaged over 10 random seeds). Fréchet Inception Distance (FID) and Inception Score (IS) measure generated sample quality. Results averaged over 10 random seeds.

Random	371.16
Uniform	375.23
Softmax	345.62
Misclassified	402.63
Cross-Entropy (CE)	365.92
CE-Variety	361.50
CE-Variety (incl. new class)	**341.84**

Table 4: Fréchet Inception Distance (FID) of Sample Methods on CIFAR-100 Experiment. (lower is better)

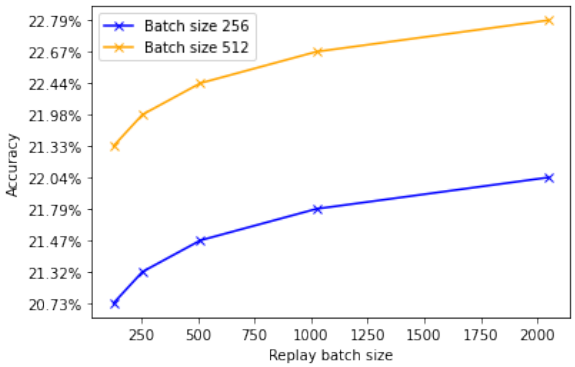


Figure 1: Accuracy of base BI-R model across batch size and replay batch sizes. Results averaged over 10 random seeds.

Fréchet Inception Distance (Heusel et al., 2017) measures the overall quality of generated samples and accuracy to the original dataset. A lower score means higher quality images, and closer distribution in variance to the original dataset. As we can see in Table 4, the Cross-Entropy Variety sampling method performed significantly better than all other sample methods besides Softmax, which was slightly behind it. However, Softmax achieved high quality samples at the cost of accuracy, performing worse than the baseline of random sampling.

An unexpected finding was that excluding the predicted probabilities for new classes decreased accuracy of Cross-entropy Variety by 0.63%. Our intuition is this penalizes the model for predicting generated samples as the new class, because the generated samples are all guaranteed to be one of the previously seen classes.

8.2 Scalability

In order to put the scalability of our proposed sample method in context, we first test scalability of the base BI-R model using random sampling. van de Ven et al. find performance to be decent even when scaled down to very small replay batch sizes, but did not cite experiments with larger batch size or replay batch sizes than 256.

As shown in Figure 1, BI-R performance continues to improve when scaled to larger batch sizes and replay batch sizes. Though we were unable to perform experiments with larger batch sizes than 512 due to computational limitations, it is clear that this hyperparameter is more impactful than replay batch size. Even pairing 256 batch size with a replay batch size of 2048 does not surpass performance of 512 batch size with the smallest replay batch size of 128. These results provide support for hypothesis H4.

As we can see in Table 3, our Cross-entropy Variety sampling method results in higher sample quality as indicated by both FID and IS, as well as increased average accuracy across all tasks by 2% and 2.68% for 256 and 512 batch sizes, respectively. Increased performance gain with larger batch sizes supports our hypothesis H4. Impressively, smallest scaling of Cross-entropy Variety at 256 batch size outperforms the largest scaling of BI-R (batch size 512; replay batch size 2048) by 0.53%. These results provide evidence for hypothesis H3.

Table 5 tests the performance of Cross-entropy Variety when replaying different sized subsets of the generated samples. We use the term 'cura-

Curation Multiplier	Batch Size	Replay Batch Size	Total Generated Samples	Accuracy ↑	FID ↑
1	256	1024	1024	21.79%	349.69
2	256	512	1024	22.64%	343.52
4	256	256	1024	**23.41%**	346.65
1	512	2048	2048	21.93%	313.97
2	512	1024	2048	22.02%	312.13
4	512	512	2048	**24.22%**	313.10

Table 5: Performance of Cross-entropy Variety using varying subsets of generated samples. Performance is maximized by selectively replaying only a small portion of the generated samples. Results averaged over 5 random seeds.

tion multiplier' to indicate how many extra samples are generated for the model to choose from: for a given eplay batch size r and curation multiplier c, a total of $r * c$ samples are generated, from which the model replays the top-r samples, according to some metric. Sorting metrics for each sampling methods are outlined in section 6. Results show that increasing curation multiplier directly correlates with higher performance, both with batch size 256 and 512. This provides evidence in support of hypothesis H3, though it is unclear whether or not this increased performance is due to increased variety in samples or samples which are more likely to be forgotten, as the Cross-entropy Variety method optimizes for both objectives.

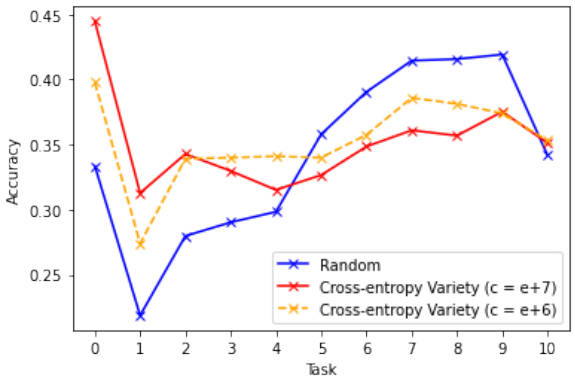


Figure 2: Accuracy of BI-R + SI using random sampling versus Cross-entropy Variety sampling. Random sampling yields 34.18% average accuracy, compared with 35.31% from Cross-entropy Variety with hyperparameter c = e+6. Setting c = e+7 leads to 35.12% average accuracy, but increases task 1 performance from 27% to 31%. Results averaged over 10 random seeds.

8.3 Synaptic Intelligence

As cited in the original paper, Synaptic Intelligence (SI) led to a significant increase in accuracy, from 21% to 35% (van de Ven et al., 2020). This component adds an additional term to the loss function which approximates performance on prior tasks, discouraging the the model from updates which would cause forgetting in prior tasks.

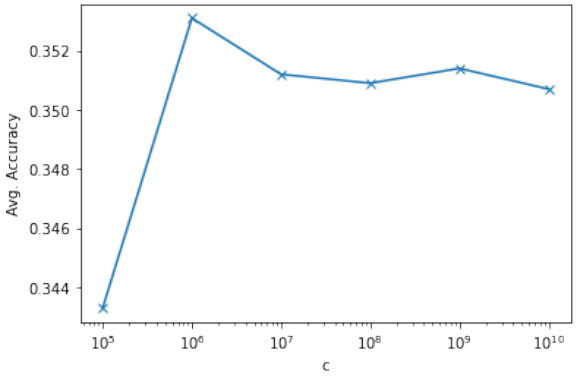
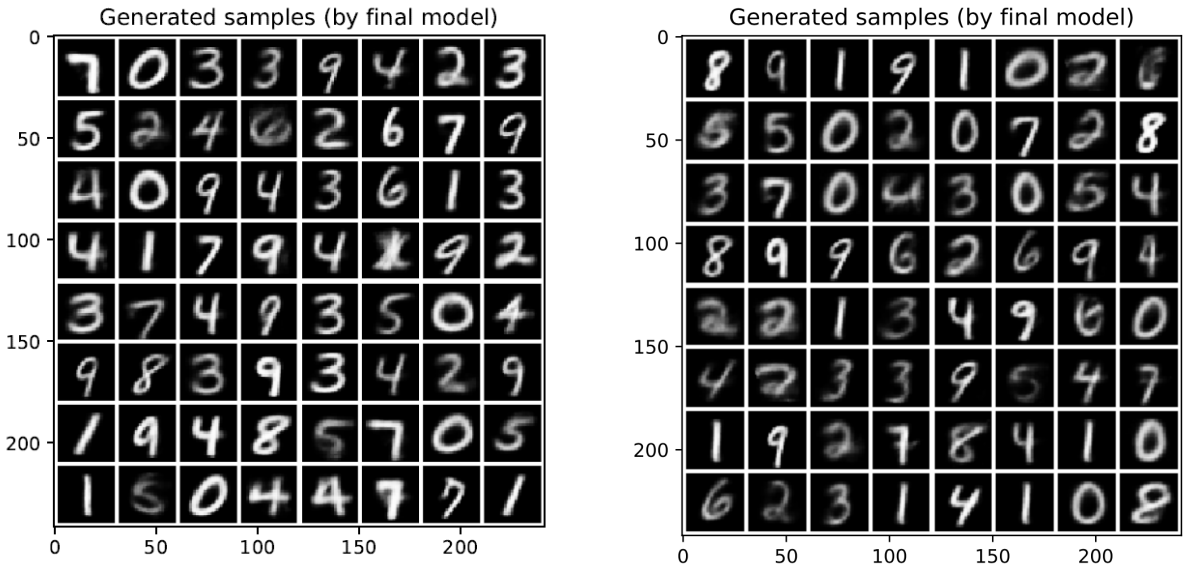


Figure 3: Grid search results for SI hyperparameter c, controlling weight of SI loss versus regular loss. Performance across c values differs from base BI-R (van de Ven et al., 2020). Results averaged over 10 random seeds.

Shown in Figure 2, our Cross-entropy Variety sampling method greatly reduces forgetting, allowing the model to maintain high accuracy across all previously learned tasks, with a minimum of 31.43%. With base BI-R using random sampling, accuracy drops over ten points from the average, down to 21.88%, equivalent to the average accuracy without SI. This shows that SI is limited in its reduction of forgetting. Cross-entropy Variety also increases overall accuracy from 34.18% to 35.12%.

Figure 3 shows that optimal performance with Cross-entropy Variety sampling is obtained using a c value of e+6, as compared with e+8 used in BI-R. Though e+6 reaches a maximum average performance of 35.31, this comes at the cost of



(a) Best sample quality: Cross-entropy Variety (341 FID)

(b) Worst sample quality: Misclassified (402 FID)

Figure 4: Comparison of sample quality from Cross-entropy Variety and Misclassified sampling methods. Samples in (a) exhibit increased clarity and decreased blur. Lower FID score is better.

additional forgetting for earlier tasks, leaving performance of task 1 at 27.41 We choose e+7, as this maintains strong average performance while more effectively reducing forgetting on earlier tasks.

As smaller values of c bias performance towards recently learned classes, the fact that Cross-entropy Variety even at e+6 maintains better reduction in forgetting when compared with BI-R at e+8 clearly shows that our method is providing additional reduction in catastrophic forgetting, and does not require as heavy biasing towards performance on previously seen tasks through SI.

8.4 Sample Quality

Figure 4 shows 64 samples generated by two of our sample selection methods: Cross-entropy Variety and Misclassified. As we can see, images on the left (a) are noticably more clear and bright, whereas images on the right (b) are blurry and more dim.

We expected the class distribution for Softmax to be skewed, as that is its key motivation, but we were surprised by extend to which it shifted the distribution: it produced just two 6's and two 8's, as compared with twelve 4's. Misclassified method does not explicitly bias by class like Softmax, but it still exhibited skewed distributions, as we can see by counting the occurrences of each class in Figure 4 (b): there are just three 6's, along with nine 2's and 9's.

We found our qualitative evaluations correlated with the FID automatic metric, with Cross-entropy Variety scoring 341.84, as compared with 402.63 from Misclassified (lower is better). As shown in Table 4, the addition of variety only results in a minor decrease in FID (4.42) indicating that the use of cross-entropy loss is the main component leading to increased sample quality.

An unexpected finding is that sample quality does not directly correlate with model performance. This is exemplified by our Softmax sample selection method which results in 0.77% worse accuracy than the baseline of random selection, but produces significantly higher quality samples, reducing FID by 25.54 (see Table 1 and Table 4).

9 Conclusion

Our work offers an in-depth investigation into sample selection methods for generative replay models. We propose an efficient sample selection method, Cross-entropy Variety, and empirically demonstrate its effectiveness as compared with its inspiration, Maximally Interfered Retrieval (MIR) (Aljundi et al., 2019). Combining this sample method with state-of-the-art generative replay model Brain-Inspired Replay, we achieve increased performance of 2.68% despite no additional parameters (van de Ven et al., 2020). Combined with Synaptic Intelligence (SI), our method results in a 0.94% increase in average ac-

curacy to 35.12%, while boosting each individual task accuracy to above 31%, as compared with a minimum of 21% with BI-R + SI.

We additionally test the scalability of BI-R, quantifying the performance gains from increasing batch size and replay batch size, and finding our proposed sample selection method results in significantly better scaling. We test many alternative sample selection methods and use ablation studies toidentify key mechanisms responsible for our propose method's performance gains.

We use qualitative analysis in order to investigate the relationship between generated sample quality and model performance, across a variety of sample selection methods. Automatic metrics such as FID confirm our qualitative findings, and indicate that high quality samples do not correlate with superior model performance.

A key finding of our work is quantifying the effect of sample selection with respect to replay batch size, finding that from a given batch of generated samples, b, the best performance is gained by selectively replaying just a quarter of these generated samples, with a replay batch size of $\frac{1}{4}b$. This shows that it is favorable to be more selective with the samples that are replayed to the model, rather than increasing the total number of samples to replay. We believe the mechanism behind this effect to be a combination of variance in generated sample quality (i.e., many samples are low-quality and detrimental to model performance when replayed) and utility of replaying samples which are maximally interfered, and would otherwise be forgotten.

10 Future Work

A future direction of work is modifying Softmax sampling (3) to normalize probability (pushing it closer to a uniform distribution), as the failure of this method may be either due to the falsity of H1 or the superveniance of H2 over H1. This modification could potentially reveal a sweet spot between optimizing for samples which the model confuses new data for, and relative representation of previously seen classes. We would also like to test the performance of Cross-Entropy Variety when using smaller replay batch sizes, in comparison with the same experiment conducted by van de Ven et al.. We would also like to do further qualitative analysis, and error analysis to see which kinds of images it is mistaking for which classes, and visualize samples from CIFAR-100. We would alaso like to experiment with varying the variety weight hyperparameter, as we were only able to test four additional values for two random seeds each.

Before publication, we would also like to compare performance explicitly with other methods, at the very least MIR. Initial testing indicated that using KL divergence rather than cross-entropy loss decreased performance significantly, but this needs to be rigorously proven. We would also like to quantify forgetting, as is done in many other continual learning and generatve replay papers. Comparison with recent models such as SLDA (Hayes and Kanan, 2019) and Generative Classification (van de Ven et al., 2021) would also be relevant, as BI-R is no longer the state-of-the-art for the task of Class-IL on CIFAR-100. We would also like to get final results on Split-MNIST and Permuted-MNIST, as well as on new datasets such as CORe50. Measuring results on exact replay would be beneficial as well, as Aljundi et al. make a convincing argument showing the efficacy of MIR on both exact and generative replay.

Finally, we would also like to collect exact measurements on runtime and memory usage, in order to better quantify the computational overhead induced by our sampling methods. Further qualitative analysis to visualize which examples were maximally interfered by which new training data, as shown in (Aljundi et al., 2019), would be very insightful as well.

References

Aljundi, R., Caccia, L., Belilovsky, E., Caccia, M., Charlin, L., and Tuytelaars, T. (2019). Online continual learning with maximally interfered retrieval.

Dhawan, A. (2020). Memory reactivation and its effect on exercise performance and heart rate. *Frontiers in Sports and Active Living*, 2:20.

Hayes, T., Kafle, K., Shrestha, R., Acharya, M., and Kanan, C. (2019). Remind your neural network to prevent catastrophic forgetting.

Hayes, T. L. and Kanan, C. (2019). Lifelong machine learning with deep streaming linear discriminant analysis. *CoRR*, abs/1909.01520.

Hayes, T. L., Krishnan, G. P., Bazhenov, M., Siegelmann, H. T., Sejnowski, T. J., and Kanan, C. (2021). Replay in Deep Learning: Current Approaches and Missing Biological Elements. *Neural Computation*, 33(11):2908–2950.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Klambauer, G., and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a nash equilibrium. *CoRR*, abs/1706.08500.

Krizhevsky, A. (2009). Learning multiple layers of features from tiny images. Technical report.

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. (2018). Continual lifelong learning with neural networks: A review. *CoRR*, abs/1802.07569.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Shin, H., Lee, J. K., Kim, J., and Kim, J. (2017). Continual learning with deep generative replay. *CoRR*, abs/1705.08690.

van de Ven, G., Siegelmann, H., and Tolias, A. (2020). Brain-inspired replay for continual learning with artificial neural networks. *Nature Communications*, 11:4069.

van de Ven, G. M., Li, Z., and Tolias, A. S. (2021). Class-incremental learning with generative classifiers. *CoRR*, abs/2104.10093.

van de Ven, G. M. and Tolias, A. S. (2019). Three scenarios for continual learning. *ArXiv*, abs/1904.07734.

Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., and Fu, Y. (2019). Large scale incremental learning. *CoRR*, abs/1905.13260.