Investigating the Role of Samples in Catastrophic Forgetting

Cedric Tschechtelin [1] **Mauro Vogel** [1] **Simon Bührer** [1] **Joshi Himanshu** [1]

Abstract

Continual learning tries to acquire and update knowledge incrementally, while facing challenges such as catastrophic forgetting. Replay-based strategies, such as Goldilocks sampling, focus on moderately learned examples to improve learning efficiency. In this study, we investigated three different approaches to replay sampling: sensitivity-aware sampling, weighted sample prioritizing, and confidence-based learning speed estimate. The results of our trials showed that these methods produced performance comparable to the Goldilocks technique, without obvious improvements. These results demonstrate the Goldilocks strategy's resilience.

1. Introduction

The field of *continual learning* (Wang et al., 2024; Delange et al., 2021; Hadsell et al., 2020; Parisi et al., 2019) focuses on enabling models to incrementally acquire and update knowledge throughout their lifetime. This capability lays the groundwork for AI systems to evolve in dynamic, real-world environments. However, *catastrophic forgetting* (McCloskey & Cohen, 1989; French, 1999) challenges continuous learning, as learning new tasks reduces performance on previously learned tasks.

Different approaches have been explored by researchers to deal with catastrophic forgetting. In general, these techniques can be categorized into five groups: regularization-based, representation-based, optimization-based, architecture-based, and replay-based approaches (Wang et al., 2024).

Replay-based methods have drawn a lot of interest. Recent developments include *Goldilocks* (Hacohen & Tuytelaars, 2024), a sampling strategy that takes advantage of the relationship between learning speed and forgetting. Specifically, they observed that samples learned at an intermediate speed are the most effective for rehearsal. Goldilocks sampling filters out samples that are learned too fast or too slow.

Understanding a model's sensitivity to its training data is just as important as its learning speed. The Memory-Perturbation Equation (MPE) (Nickl et al., 2024) is a recent work that uses Bayesian principles to quantify sensitivity to data disturbances. Existing sensitivity measures are unified and generalized by MPE, making it suitable for a variety of models and methods. Empirical results (Nickl et al., 2024) demonstrate that sensitivity estimates obtained during training can effectively predict generalization performance on unseen data, offering valuable information to develop more robust and adaptive learning systems.

Despite these developments, little research has been done comparing model sensitivity-based approaches with learning speed aproaches. Furthermore, there are still inefficiencies in the selection of samples for rehearsal. For example, determining learning speed often requires prolonged training.(Hacohen & Tuytelaars, 2024). We suggest three innovative sampling techniques to overcome these issues:

- **Confidence-Based Learning Speed Estimation:** We introduce an efficient method for estimating learning speed based on changes in model confidence over a small number of epochs.
- **Weighted Sample Prioritization:** Leveraging the correlation between learning speed and retention, we prioritize examples that are learned more slowly using a weighted sampling mechanism.
- **Sensitivity-Aware Sampling:** We incorporate model sensitivity, estimated through prediction variance and error, to identify critical examples for rehearsal.

These strategies are designed to optimize replay buffer sampling, addressing inefficiencies and boosting performance in continual learning scenarios.[1]

2. Methodology

Continual Learning with replay buffers is defined as the model f being trained on each task $\mathcal{D}_t$ for $E \in \mathbb{N}$ epochs.

[1]Eidgenössische Technische Hochschule, Zurich, Switzerland. Correspondence to: Cedric Tschechtelin <ctschecht@ethz.ch>, Joshi Himanshu <hjoshi@ethz.ch>, Mauro Vogel <mavogel@ethz.ch>, Simon Bührer <sbuehrer@ethz.ch>.

[1]Code is available at https://gitlab.ethz.ch/sbuehrer/dl2024.

During the training of task t, a fixed-size replay buffer $\mathcal{B}_t \subseteq \mathcal{D}_1 \cup \cdots \cup \mathcal{D}_{t-1}$ is maintained to store examples from the previously encountered tasks. By satisfying the restriction $|\mathcal{B}_t| \ll |\mathcal{D}_t|$, the buffer's size guarantees that only a limited portion of the previous task data is kept. In the case of no memory replay, i.e., $\mathcal{B}_t = \emptyset$, the model encounters catastrophic forgetting, where the performance on prior tasks deteriorates as new tasks are learned.

2.1. Experimental Setup

Unless otherwise stated, all methods are evaluated on the CIFAR-10 dataset (Krizhevsky, 2009), using ResNet-18 (He et al., 2015) as the base model. Optimization is carried out with stochastic gradient descent (SGD), employing a learning rate of 10^{-4}. For each task t, the model is trained for $E = 100$ epochs. The total number of tasks is set to $T = 2$. Algorithm 1 outlines the implementation of learning speed estimation and weighted sampling.

Algorithm 1 Confidence Based Learning Speed Estimation and Weighted Sampling

Input: Task data $\mathcal{D}_t = \{(x_i, y_i)\}$, model f, epochs E, buffer size $|\mathcal{B}|$
Initialize confidence matrix $C \in [0, 1]^{E \times |\mathcal{D}_t|}$
for $e = 1$ to E **do**
  Train f on $\mathcal{D}_t$ for one epoch
  for $i = 1$ to $|\mathcal{D}_t|$ **do**
    $C[e, i] \leftarrow c_e(x, y)$
  end for
end for
Compute learning speed $LS_{\text{conf}}(x, y)$
Assign weights $\omega(x_i, y_i)$
Normalize weights $P(x_i, y_i)$
Sample buffer $\mathcal{B}$ based on $P(x_i, y_i)$

2.2. Confidence-Based Learning Speed Estimation

Effectively assessing the learning speed of individual samples, which has a direct impact on their selection for the replay buffer, is a major difficulty in continual learning. The learning speed of an example (x, y) is defined by earlier research, such as Goldilocks (Hacohen & Tuytelaars, 2024), as follows:

$$LS_{Goldi}(x, y) = \frac{1}{E} \sum_{e=1}^{E} \mathbb{I}[f_e(x) = y],$$

where f_e represents the model f after the e-th epoch of training on task $\mathcal{D}_t$, and $\mathbb{I}[\cdot]$ is the indicator function that evaluates to 1 if the prediction $f_e(x)$ matches the true label y, and 0 otherwise. While this method is computationally efficient, it heavily relies on binary classification outcomes and struggles to accurately estimate learning speed for tasks with very few training epochs.

We propose a novel confidence-based learning speed estimation method to get over this restriction. Instead of relying on binary indicators, we use the model's confidence scores during training to compute a new measure of learning speed. Specifically, for an example (x, y), the learning speed is calculated as the mean confidence over all epochs:

$$LS_{\text{conf}}(x, y) = \frac{1}{E} \sum_{e=1}^{E} c_e(x, y),$$

where $c_e(x, y)$ represents the confidence score of the model for (x, y) at epoch e.

By adding up the confidence scores over all epochs, we effectively obtain the area under the confidence curve, which serves as a indicator of the learning progress.

2.3. Weighted Sample Prioritization

(Hacohen & Tuytelaars, 2024) observed a strong correlation between the mean learning speed and the percentage of examples remembered by the networks. This indicates that networks are more likely to remember quickly learned example. To exploit this, our replay buffer will use a sampling strategy where examples are randomly selected, but with weights proportional to the inverse of their learning speed — assigning higher weights to examples learned more slowly. The weight assigned to each example (x, y) is computed as:

$$\omega(x, y) = \frac{1}{(LS(x, y) + \epsilon)^{\alpha}}$$

where $LS(x, y)$ is the learning speed as defined in (Hacohen & Tuytelaars, 2024) and $\epsilon = 10^{-6}$ ensures numerical stability for very small learning speeds. The alpha parameter controls the strength of the weighting. To determine the sampling probabilities, the weights are normalized as:

$$P(x, y) = \frac{\omega(x, y)}{\sum_{(x', y')} \omega(x', y')}$$

By prioritizing examples with lower learning speeds, the model's exposure to challenging examples is increased in the hopes of improving its long-term retention and generalization capabilities.

2.4. Sensitivity-Aware Sampling

We sample the replay buffer based on the estimate of the sensitivity of the model to its training data. We estimate the sensitivity by multiplying the prediction variance by the error, as proposed in the MPE of (Nickl et al., 2024). In our method, we are approximating the deviation on the model's

output by using this formula:

$$\underbrace{f_i\left(\boldsymbol{\theta}_t^{\setminus i}\right) - f_i\left(\boldsymbol{\theta}_t\right)}_{\text{Deviation in the output}} \approx \underbrace{\nabla f_i\left(\boldsymbol{\theta}_t\right)^\top \nabla f_i\left(\boldsymbol{\theta}_t\right)}_{=v_{it},\ \text{prediction variance}} \underbrace{\left[\sigma\left(f_i\left(\boldsymbol{\theta}_t\right)\right) - y_i\right]}_{=e_{it},\ \text{prediction error}}.$$

Where $f_i(\theta_t)$ is the ith element of the model output with model parameters θ at time t, y_i the target label and σ is the activation function. $f_i(\theta_t^{\setminus i})$ are the model parameters after training the model on a training set without the i-th example.

Since SGD is our learning algorithm, we are using the sensitivity measure from Table 1 of (Nickl et al., 2024). Following this approximation, we are prioritizing the samples based on their effect on the model's output, i.e. we are extending our replay buffer only with samples that exceed (or deceed) a certain threshold of deviation, telling us how significant the sample is to the model's output and hence how important it is to be memorized in the buffer.

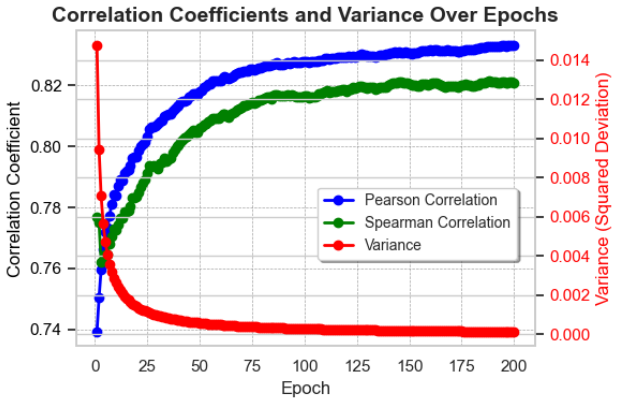


Figure 1. Pearson (blue) and Spearman (green) correlations between Goldilocks-based and Confidence-based learning speeds across all CIFAR-10 samples. The variance of the linear regression for learning speed samples is shown in red. Correlation increases with epochs, while variance decreases.

3. Results

3.1. Confidence-Based Learning Speed Estimation

After training for 200 epochs, distinct differences in learning speed behaviors between the Goldilocks-based and Confidence-based methods are observed. At the initial stage, following the first epoch ($E = 1, t = 1$), the correlation between the two methods is relatively low (Pearson $r = 0.75$, $p \leq 10^{-8}$), accompanied by high variance in the samples around the linear regression line ($v = 1.4 \cdot 10^{-2}$)[2].

As shown in Figure 1 for later epochs, the strategies increase in correlation and decrease in variance, By the end of the training process ($E = 200$, $t = 1$), a strong correlation between the methods is achieved (Pearson $r = 0.83$, $p \leq 10^{-8}$), and the variance significantly decreases to $\sigma^2 = 3 \cdot 10^{-4}$.

The observed increase in correlation and decrease in variance suggest that the learning speeds become more interchangeable as the number of epochs grows. However, discrepancies remain in the earlier stages of training, particularly at fewer epochs. To further evaluate the effectiveness of the two approaches, we analyze the test accuracy after the second task ($t = 2$).

According to the accuracy heatmaps displayed in Figure 6, there is no apparent increase in test accuracy using the confidence-based approach. When different learning speeds, epoch counts, and parameter settings are used, both techniques produce performance that is equivalent. This lack of distinction shows that model performance is not much impacted by the learning speed estimation approach selected.

[2]For a detailed visualization, the correlation scatterplot is presented in Figure 4 in the appendix.

For the statistical analysis, we tested the following hypotheses:

- Null Hypothesis (H_0): No significant difference in test accuracy between the Goldilocks and Confidence-based strategies.
- Alternative Hypothesis (H_a): A significant difference in test accuracy between the two strategies.

The results of the Wilcoxon Signed-Rank Test (Durango & Refugio, 2018) showed a p-value of 0.128 and a statistic of 95.5. We are unable to reject H_0, suggesting no significant difference, because $p > 0.05$. The results are summarized as Figure 5 for all buffer ratios and learning speeds.

3.2. Weighted Sample Prioritization

We first examine the effect of the α parameter on model performance and compare it to the baseline where samples are drawn uniformly from the replay buffer ($\alpha = 0$). Figure 7 show the performance of different α values over varying numbers of training epochs. The results indicate no significant advantage of the Weighted Sampling Prioritization Strategy compared to drawing samples uniformly from the buffer. For completeness, Figure 2 provides a heatmap comparison between the Weighted Sampling and Goldilocks Strategies, demonstrating that the latter consistently achieves superior performance, showing that the Weighted Strategy is outperformed by Goldilocks.

3.3. Sensitivity-Aware Sampling

We perform all evaluations on a buffer ratio of 20% and train for 2, 5, 10 and 15 epochs. The threshold indicates which

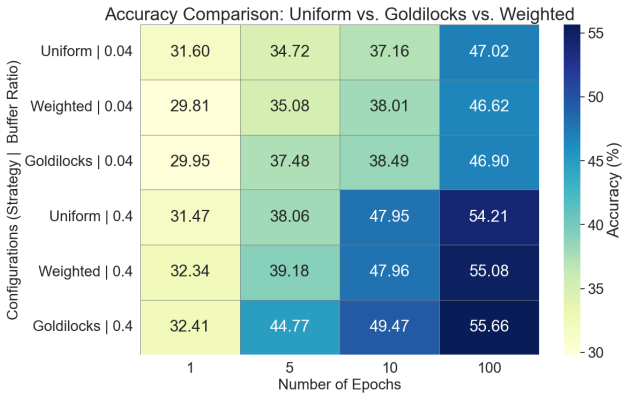


Figure 2. Heatmap showing the mean test accuracy of different strategies (Uniform, Weighted, and Goldilocks) at buffer ratios of 0.04 and 0.4 across varying numbers of epochs (1, 5, 10, and 100). For Goldilocks, the Confidence Strategy was used with the parameters quick 44% and slow 12% for a buffer ratio of 0.4, and quick 12% and slow 36% for 0.04, as identified as effective in (Hacohen & Tuytelaars, 2024). For the Weighted strategy, various well-performing alpha values were selected to optimize accuracy (compare to Figure 7).

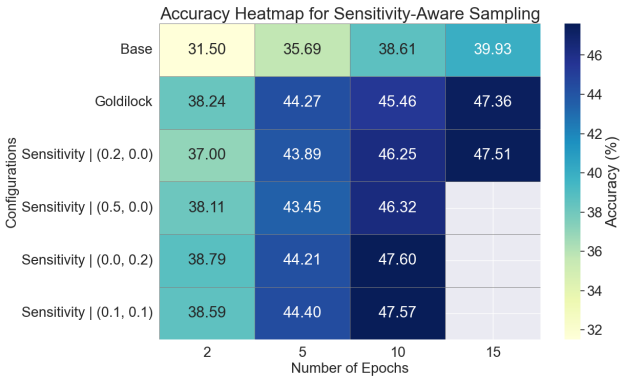


Figure 3. In this figure, one can detect similar results of test accuracy when using the sensitivity sampling method compared to the Goldilocks method (Hacohen & Tuytelaars, 2024). In this experiment we compare our method to Goldilocks method with both the slowest and fastest 10% removed. We tried out several attempts of choosing to remove the least 20%, least 50%, most 20% and to remove both the most 10% and the least 10% sensitive samples from the buffer.

percentile of sensitivity needs to be exceeded or deceeded, such that they are taken into the replay buffer. We compare the average test precision over all epochs with the precision of other strategies.

The results in 3 show that we are matching the performance of Goldilocks with a slight improvement in most of our experiments. There is a possibility that a certain correlation between high sensitivity of the model towards the samples and the time needed to learn samples exists. Interestingly, we are observing slightly better results when removing the most sensitive examples compared to removing the least sensitive examples.

4. Discussion

4.1. Confidence-Based Learning Speed Estimation

Despite variations in how learning speed is estimated the confidence-based strategy produces test accuracy results that are comparable with the Goldilocks-based approach. However, it requires significantly more memory. Given these trade-offs, we recommend continuing with the Goldilocks strategy for its lower memory consumption. While having classification scores for each epoch and sample is useful for visualization, we suggest for future works to optimize memory usage by iteratively updating the learning speed after each epoch.

4.2. Weighted Sample Prioritization

The Weighted Sampling Prioritization Strategy revealed no clear performance benefits compared to the baseline, where samples are drawn uniformly from the replay buffer. Compared to the Goldilocks Strategy, it performs worse.

4.3. Sensitivity-Aware Sampling

Due to performance constraints, we limited the training to a maximum of 15 epochs, as gradients are calculated $|\mathcal{B}|$ times per iteration. Extracting gradients of the model output with respect to parameters from the existing PyTorch computational graph proved challenging (Paszke et al., 2017), as noted in the original paper (Nickl et al., 2024). Given this issue and the negligible performance gain, we do not recommend the Sensitivity-Aware sampling method.

Further work should focus on saving the gradients during backpropagation of the loss to increase its performance. By resolving this issue, it would be easier to detect the long-term effect of the sensitivity towards the learning effect.

An interesting idea would be to weight the sensitivities of later epochs higher than those at the start of training, since deviations in the model output at a later stage should reveal more insights on the importance of examples. Additional work could focus on the correlation between learning time and sensitivity, which would benefit future research on this topic.

References

Delange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., and Tuytelaars, T. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021. ISSN 1939-3539. doi: 10.1109/tpami.2021.3057446. URL http://dx.doi.org/10.1109/TPAMI.2021.3057446.

Durango, A. and Refugio, C. An empirical study on wilcoxon signed rank test, 12 2018.

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

Hacohen, G. and Tuytelaars, T. Forgetting order of continual learning: Examples that are learned first are forgotten last, 2024. URL https://arxiv.org/abs/2406.09935.

Hadsell, R., Rao, D., Rusu, A. A., and Pascanu, R. Embracing change: Continual learning in deep neural networks. *Trends in Cognitive Sciences*, 24(12):1028–1040, 2020. ISSN 1364-6613. doi: https://doi.org/10.1016/j.tics.2020.09.004. URL https://www.sciencedirect.com/science/article/pii/S1364661320302199.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

Krizhevsky, A. Learning multiple layers of features from tiny images, 2009. URL https://api.semanticscholar.org/CorpusID:18268744.

McCloskey, M. and Cohen, N. J. Catastrophic interference in connectionist networks: The sequential learning problem, 1989. ISSN 0079-7421. URL https://www.sciencedirect.com/science/article/pii/S0079742108605368.

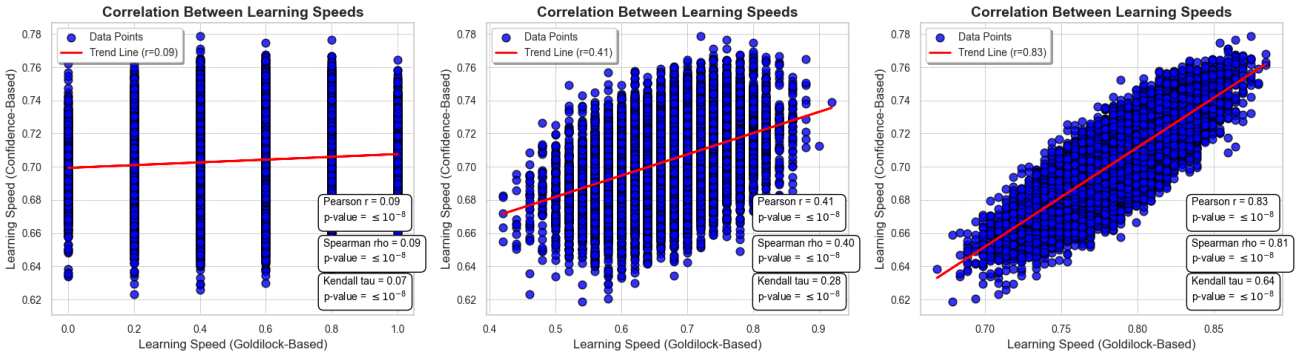
Nickl, P., Xu, L., Tailor, D., Möllenhoff, T., and Khan, M. E. The memory perturbation equation: Understanding model's sensitivity to data, 2024. URL https://arxiv.org/abs/2310.19273.

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. Continual lifelong learning with neural networks: A review. *Neural Netw*, 113:54–71, February 2019.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

Wang, L., Zhang, X., Su, H., and Zhu, J. A comprehensive survey of continual learning: Theory, method and application, 2024. URL https://arxiv.org/abs/2302.00487.

A. Confidence Based Learning Speed Estimation



(a) Epochs = 5 (b) Epochs = 50 (c) Epochs = 200

Figure 4. Scatterplots illustrating the correlation between Goldilock-based and Confidence-based learning speed estimations for CIFAR-10 samples. Statistical correlation measured with Pearson, Kendall, and Spearman. All p-values are extremely low (approaching zero with `float64` precision).

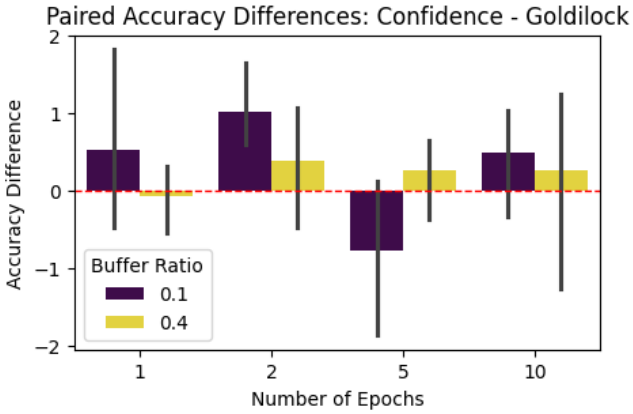
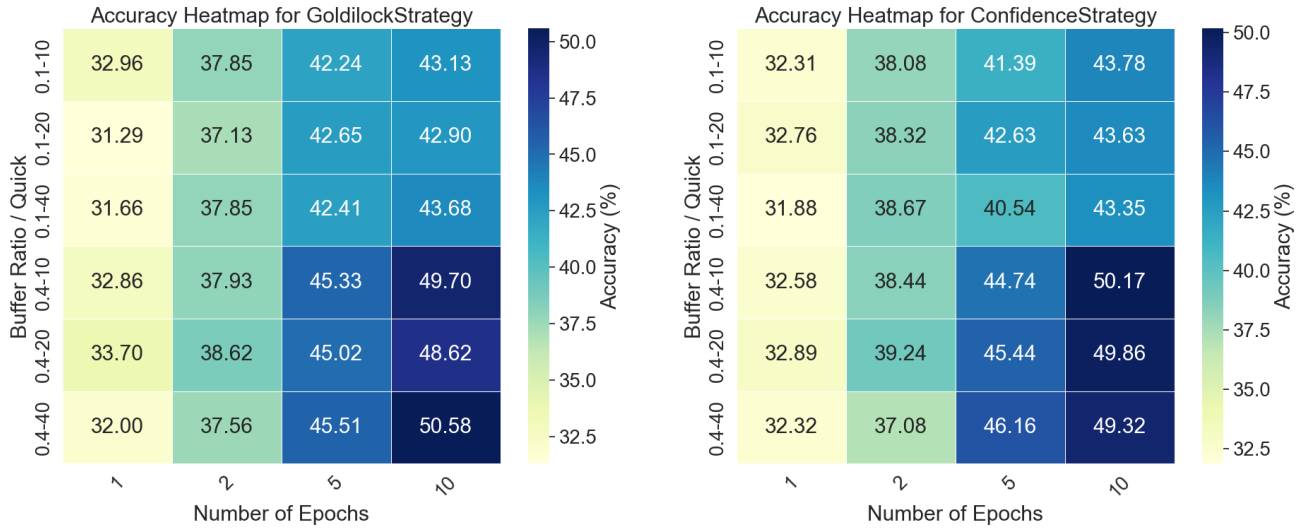


Figure 5. Bar plot showing the paired accuracy differences between the Confidence and Goldilocks strategies, calculated as Confidence - Goldilocks. The plot visualizes how the difference varies across different epoch counts and buffer ratios. The red dashed line at zero indicates no difference in accuracy between the two methods. The legend corresponds to the buffer ratio settings. The Wilcoxon Signed-Rank Test, with a statistic of 95.5 and a p-value of 0.128, shows no significant difference between the two strategies.

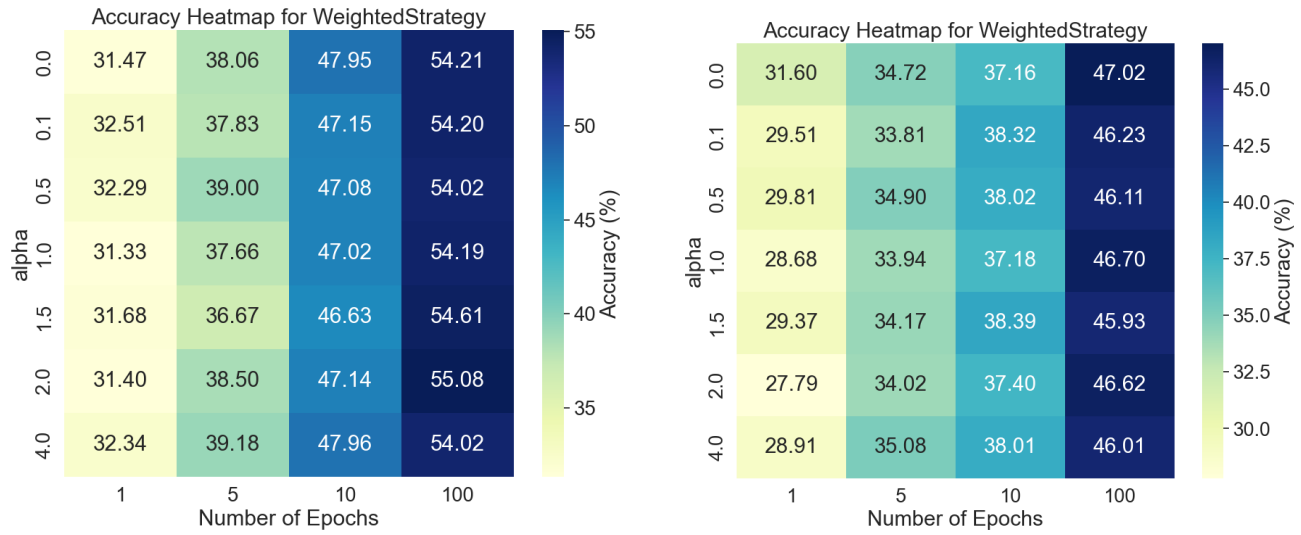


(a) Buffer ratio = 0.4

(b) Buffer ratio = 0.04

Figure 6. Heatmaps illustrating the mean test accuracy for various learning speeds, epoch counts, and buffer ratio settings. The top heatmap represents the results for the Goldilocks-based strategy, while the bottom heatmap shows the Confidence-based strategy. The accuracy values are averaged across different removal percentages of the slowest learning samples, with buffer ratios of 10% and 40%. The results indicate no significant performance difference between the two methods, as both exhibit similar accuracy trends under identical conditions.

B. Weighted Sample Prioritization



(a) Buffer ratio = 0.4

(b) Buffer ratio = 0.04

Figure 7. Heatmaps showing the mean test accuracy for various α values for the Weighted Sample Prioritization across different numbers of training epochs. The case $\alpha = 0$ represents the baseline scenario where samples are drawn uniformly from the replay buffer.

C. Authors' Contributions

Cedric Tschechtelin conceived and performed the experiments related to Sensitivity-Aware Sampling and contributed to the analysis and interpretation of the results. Cedric also wrote the corresponding sections of the report, including Section 2.4, Section 3.3, and Section 4.3.

Mauro Vogel designed and executed the experiments for Weighted Sample Prioritization and contributed to the analysis and interpretation of the results. Mauro was responsible for writing the corresponding report sections, including Section 2.3, Section 3.2, and Section 4.2.

Simon Bührer performed the experiments related to Confidence-Based Learning Speed Estimation and contributed to the analysis and interpretation of the results. Simon also wrote several key sections of the report, including Section 1, Section 2.1, Section 2.2, Section 3.1, and Section 4.1.

Joshi Himanshu - conceived and developed the Hybrid Sampling Strategy, which combines sensitivity and confidence metrics to enhance sample prioritization in continual learning.