A SIMPLE BASELINE FOR STABLE AND PLASTIC NEURAL NETWORKS

Étienne Künzel[1] **, Achref Jaziri**[1] **, Visvanathan Ramesh**[1,2]
[1] Department of Computer Science and Mathematics, Goethe University,
[2] HessianAI, Frankfurt am Main, Germany
{jaziri, vramesh}@em.uni-frankfurt.de

ABSTRACT

Continual learning in computer vision requires that models adapt to a continuous stream of tasks without forgetting prior knowledge, yet existing approaches often tip the balance heavily toward either plasticity or stability. We introduce RDBP, a simple, low-overhead baseline that unites two complementary mechanisms: ReLUDown, a lightweight activation modification that preserves feature sensitivity while preventing neuron dormancy, and Decreasing Backpropagation, a biologically inspired gradient-scheduling scheme that progressively shields early layers from catastrophic updates. Evaluated on the Continual ImageNet benchmark, RDBP matches or exceeds the plasticity and stability of state-of-the-art methods while reducing computational cost. RDBP thus provides both a practical solution for real-world continual learning and a clear benchmark against which future continual learning strategies can be measured.

1 INTRODUCTION

Continual learning in computer vision tackles the fundamental challenge of enabling models to adapt to a continuous stream of visual information rather than to a single static dataset. Such systems must *continuously* integrate new concepts while retaining the features and representations learned from previous tasks. This requires balancing plasticity, the ability to acquire new knowledge as data distributions shift, with stability, the capacity to preserve previously learned representations and avoid catastrophic forgetting. The plasticity–stability dilemma is especially critical in real-world applications, where access to past data may be restricted by memory constraints or privacy regulations, making rehearsal-based methods impractical (Mermillod et al., 2013; Smith et al., 2023).

Although many class-incremental methods report strong end-to-end performance in a continual learning setting, it often remains unclear whether those gains arise from enhanced plasticity, superior stability, or both. Recent analyses show that numerous class-incremental algorithms prioritize stability so heavily that the feature extractor trained on the initial tasks alone can be as effective as that of the full incremental model (Kim & Han, 2023). These observations underscore the need for both a more granular understanding and analyses of continual learning dynamics and the establishment of clear, simple baselines that achieve a balanced trade-off between plasticity and stability.

A key motivation for our work is the *lack of a straightforward and simple baseline* in the stability–plasticity literature. Most existing approaches skew toward optimizing either plasticity or stability in isolation. Moreover, prevalent methods for boosting plasticity such as weight perturbations that inadvertently erase previously acquired knowledge (Dohare et al., 2024; Lee et al., 2024), or adaptive activation functions (Delfosse et al., 2021) that introduce extra parameters and complex operations often complicate the incorporation of stability mechanisms and increase computational overhead.

Some recent efforts have attempted to address this by leveraging multi-scale feature encodings from pre-trained backbones, structure-wise distillation losses to preserve hierarchical representations, and bespoke stability–plasticity normalization layers to balance adaptation and retention (Jung et al., 2023; Elsayed & Mahmood, 2024). These methods have demonstrated notable performance improvements across various continual learning benchmarks, showcasing their ability to mitigate forgetting while enabling efficient task adaptation. However, despite their effectiveness, these solutions incur significant architectural complexity, rely on auxiliary distillation objectives, and depart from the simplicity desired for a baseline framework. In this work, we explore a simple yet effective approach to continual learning that preserves high plasticity without undermining stable representations or imposing heavy computational costs. Specifically, we introduce:

- **ReLUDown**, a lightweight activation modification that dynamically scales activations to enhance adaptability while leaving earlier-layer features intact;
- **Decreasing Backpropagation (DBP)**, a gradient-scheduling scheme inspired by biological memory consolidation processes, which progressively decreases gradient flow through lower layers to protect established knowledge.

Collectively, ReLUDown and DBP, forming our **RDBP** baseline, present a low-complexity framework for addressing the stability-plasticity dilemma. RDBP not only provides a practical solution for class-incremental continual learning but also serves as a clear, lightweight benchmark for comparing future continual learning approaches.

2 Related Work

Continual learning, also referred to as lifelong learning (Parisi et al., 2019), requires models to learn from a sequence of non-stationary data streams while jointly addressing both network plasticity and stability (Kim & Han, 2023). Traditional neural networks, when trained sequentially on multiple tasks, often experience catastrophic forgetting (Riemer et al., 2018), where previously learned representations are overwritten, and their ability to adapt effectively to new tasks is significantly reduced.

Maintaining **plasticity** ensures that models can adapt their parameters in response to a change in the data distribution (Berariu et al., 2021). Two main phenomena impact the network's plasticity: first, shifts in the pre-activation distribution during training can cause neurons to become "dead" (Lin et al., 2015), "dormant" (Sokar et al., 2023), or effectively linearized (Lyle et al., 2024), reducing the network's expressive capacity; second, unchecked growth in parameter norms sharpens the loss landscape, driving activation and normalization layers toward saturation and diminishing sensitivity to gradient updates (Damian et al., 2022). To counteract these effects, three main categories of methods have emerged. One line of work periodically resets or overwrites selected weights or layers to increase the network's ability to learn new patterns (Dohare et al., 2024; He & Liu, 2025). A second approach focuses on activation-level interventions using adaptive activation functions or carefully calibrated noise injections to preserve plasticity without altering the underlying training dynamics (Delfosse et al., 2021; Kuo et al., 2021). Finally, architectural expansion methods introduce additional modules or subnetworks that specialize in handling new tasks, which allows the base model to retain existing representations intact (Liu et al., 2025).

Maintaining **stability** is essential not only to prevent catastrophic forgetting but also to facilitate efficient transfer to related tasks (Weiss et al., 2016; Zhuang et al., 2020). One common strategy is to retain and replay a small set of representative examples from past tasks, thereby reinforcing old knowledge whenever the network trains on newly introduced data (Shin et al., 2017). Other methods dynamically adjust the structure of the network, either by dedicating subnetworks to each task or by growing new modules on demand, to protect previously learned features (Yoon et al., 2017). Regularization-based approaches, such as Elastic Weight Consolidation, impose penalties on updates to parameters deemed important for earlier tasks, using measures like the Fisher information matrix to identify and protect critical weights (Schwarz et al., 2018).

Recent works tackle the stability-plasticity dilemma in various contexts through task sequencing, alternating update schemes, and transfer learning. For example, Jaziri et al. (2024) propose multiple subspace expansions during training, which alternate between weight regularization for better stability and adaptive activations for higher plasticity in a curriculum learning setting. While Scialom et al. (2022) employs continual fine-tuning of pre-trained models, refining large-scale features without full retraining. Integrated frameworks further combine regularization, distillation, and specialized normalization to balance adaptation and retention (Jung et al., 2023). Yet, these methods often add architectural or computational overhead, underscoring the need for a simple baseline like RDBP proposed in this work.

3 RDBP: Simple Baseline for the Stability-Plasticity Dilemma

We present RDBP, that unites two complementary mechanisms to tackle the plasticity–stability dilemma. First, it employs a modified activation function, ReLUDown, designed to preserve plasticity. Second, it integrates a Decreasing Backpropagation (DBP) update scheme, which gradually decreases gradient flow through lower layers to protect previously learned weights. Together, these components form an easy-to-implement baseline that requires no auxiliary memory buffers or complex architectural additions.

3.1 MAINTAINING PLASTICITY: RELUDOWN

For our baseline, we focus on plasticity approaches that preserve existing representations to ensure compatibility with stability mechanism. Methods that reset or expand parameters tend to impact model stability or simply defer plasticity challenges to newly added modules which can be computationally expensive in the long run. Instead, activation-based techniques maintain sensitivity to new inputs without erasing prior knowledge. Adaptive activation functions (Delfosse et al., 2021) maintain plasticity over long sequences but introduce extra computation and unstable preactivation distributions, making them hard to combine with stability mechanisms. To address these issues, we propose a static activation alternative that retains the benefits of adaptive activations.

We derive our static activation function from the standard ReLU, which fails to maintain plasticity due to the preactivation distribution gradually shifting into the negative domain, causing neurons to become dormant (Lyle et al., 2024). This issue can be effectively addressed by introducing a linear component with a non-zero gradient for the range from the hinge point $d < 0$ to $-\infty$. This approach preserves the non-linearity of the activation function while ensuring non-zero gradients on both sides of the zero-encoding, thereby enabling the model to implicitly stabilize its preactivation distribution by the standard training process. The activation function is displayed in the appendix and can be formulated as follows:

$$f(x) = max(0, x) - max(0, -x + d), d < 0$$

3.2 MAINTAINING STABILITY: DECREASEBACKPROPAGATION

Modeling the interaction between long-term and short-term memory has been a longstanding topic of research (Melton et al., 1970). Long and short-term memory can be modeled as discrete systems (Izquierdo et al., 1999) or as points along a continuum (Başar & Düzgün, 2016), each comes with its own advantages and disadvantages. Our proposed solution DecreaseBackPropagation (DBP) relies on a continual modeling approach and transfers this principle to neural networks by adapting the backpropagation process. Instead of applying traditional backpropagation uniformly across all layers, our method gradually reduces the influence of backpropagation on earlier layers as the network encounters new tasks.

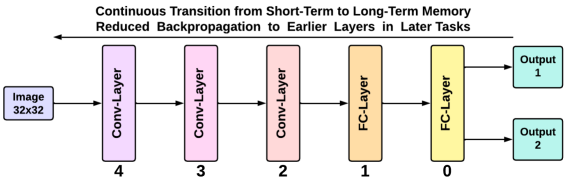


Figure 1: Schematic illustration of the DBP algorithm

The standard backpropagation formula is adapted using an annealing process where the gradient anneals to a fraction dependent on the decrease Factor f and the Layer l(numbered from front to back see Figure 1) with increasing Task Number n, the higher the speed factor a, the faster the algorithm anneals:

$$\text{bp}_{decrease}(n, l, f, a) = \text{bp}_{standard} \times (1 - (l \times f) + (l \times f) \times a^{-n})$$

This strategy emulates long-term memory by allowing earlier layers to encode frequently occurring patterns while maintaining adaptability in the later layers through reduced restrictions, making them flexible for the specific task.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP AND METRICS

We evaluate our models on an adapted version of the ImageNet Dataset (Deng et al., 2009) called Continual Imagenet (Dohare et al., 2024), where each task is binary classification between two ImageNet classes. Each model has only has access to the data of the current task. We contrast the proposed method with Continual Backpropagation, which prioritizes network plasticity (Dohare et al., 2024) and Generative Replay, which prioritizes preserving prior task knowledge(Shin et al., 2017). For RDBP, we chose a hinge point of $d = -3$, a decrease factor of $f = 0.15$ and a speed factor of $a = 1.005$. All networks use 3 convolution layers followed by two fully connected layers. For further information about the baselines, hyperparameters and evaluation variables see the Appendix.

We assess plasticity by measuring the accuracy of the model in each newly learned task immediately after training in a continual sequence. The classification head is reset after each task and stored. Stability is quantified as the mean accuracy over the previous ten tasks, evaluated without any further weight updates to the network and the classification head of the matching task.

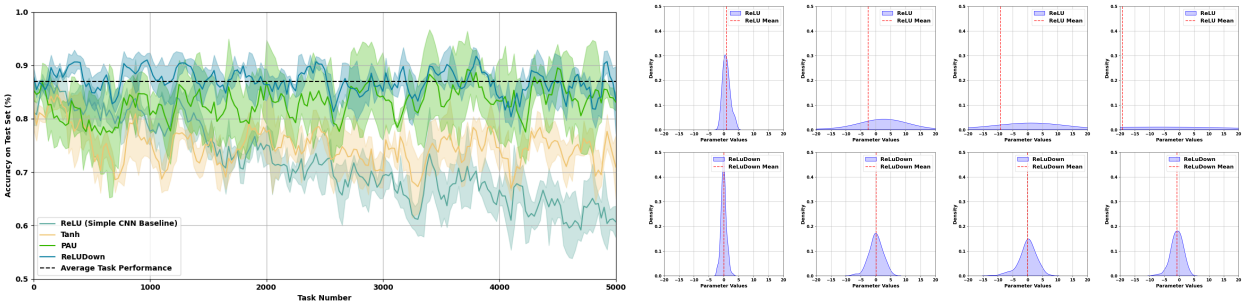


Figure 2: **Left**: Current Task Performance of a CNN with different Activation Functions: ReLU, Tanh, ReLUDown, PAU and the Average Performance of a CNN fully reset at each task **Right**: Preactivation Distribution for Task 100, 1000, 3000, 5000 of the CNN with the ReLU(top) and ReLUDown(bottom)

4.2 Results and Discussion

In Figure 2, we illustrate the impact of various activation functions on the neural network's ability to maintain plasticity (left panel). The results demonstrate that the network utilizing the ReLU activation function exhibits a decline in performance, despite similar task complexity. In contrast, networks employing the ReLUDown or Padé Activation Unit(Delfosse et al., 2021) activation functions are able to maintain their plasticity over time. Additionally, our proposed activation function achieves this while also reducing training time by approximately 20% compared to the Padé Activation Units (PAU)(see Appendix A.5.3).
The right panel of Figure 2 provides insights into the preactivation distribution. In the top row, we can see the network with the ReLU activation function experiences a shift in the preactivation distribution shift. Conversely, for the ReLUDown, the preactivation distribution converges toward a normal distribution, thereby mitigating one of the main mechanisms of plasticity loss: "Linearization and preactivation distribution shift" noted by Lyle et al. (2024).

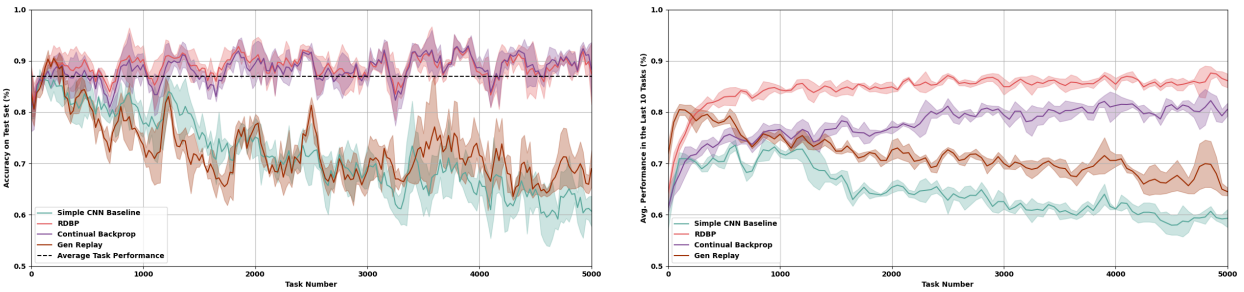


Figure 3: Performance of a CNN with a ReLU, RDBP, Continual Backpropagation and Generative Replay. **Left**:Current Task Performance to evaluate Plasticity. **Right**: Average Performance in the last 10 Tasks to evaluate Stability

In Figure 3, we compare our approach RDBP with Standard CNN, Continual Backpropagation, and Generative Replay in terms of network plasticity (left) and stability(right).
For plasticity, RDBP performs similarly to Continual Backpropagation with both of their performances fluctuating around the performance level network trained on each task independently oscillate around the single-task training benchmark (black dotted line), demonstrating preserved plasticity and effective transfer learning in some cases. While Continual Backpropagation achieves this with neuron resets our approach achieves similar results without additional computations. When using Generative Replay with a ReLU activation function, the classifier still loses plasticity like the Simple CNN baseline, though slightly more slowly, likely due to increased variance in the training data, which makes it harder for the classifier to overfit to individual tasks.
For stability, RDBP showcases an increasing performance with progressing training due to the fact that in earlier tasks the gradients are not as reduced as in later tasks. Continual Backpropagation maintains plasticity by resetting parameters, yet remains stable, likely due to a high maturity threshold preserving knowledge of the last ten tasks, but suffers catastrophic forgetting for older tasks. Generative Replay shows good task maintenance for the first tasks, which then decreases due to the decline in plasticity, underlining the importance of considering both plasticity and stability when evaluating continual learning methods.

5 Conclusion and Future Work

We introduce ReLUDown, a drop-in activation that increases plasticity in continuous learning by preserving the dynamics of the standard network, avoiding pre-activation changes and requiring no parameter changes. Paired with Decreasing Backpropagation (DBP), which gradually attenuates gradient updates in early layers to mirror a continuum between short- and long-term memory and enhance stability, these two components form RDBP. RDBP balances plasticity and stability in an incremental task setting, all without extra memory, architectural tweaks, or added complexity. We hypothesize that this algorithm will perform effectively in settings where the underlying task structure remains consistent. This is particularly relevant in domains where representations obtained from convolutional layers are transferable across tasks (i.e., object classification). For future work, we will systematically benchmark RDBP and various continual learning methods across a wider spectrum of incremental paradigms, including classes, tasks, and domain incremental settings. Moreover, for applications where gradient attenuation alone cannot fully prevent forgetting, we will investigate hybrid schemes that couple RDBP with stronger stability modules, such as generative replay buffers or adaptive network expansion, to dynamically reinforce memory retention without sacrificing plasticity.

Acknowledgments

This work was supported by the KIBA Project (Künstliche Intelligenz und diskrete Beladeoptimierungsmodelle zur Auslastungssteigerung im Kombinierten Verkehr - KIBA) under reference number 45KI16E051, funded by the German Ministry of Transportation.

References

Erol Başar and Aysel Düzgün. The brain as a working syncytium and memory as a continuum in a hyper timespace: Oscillations lead to a new model. *International Journal of Psychophysiology*, 103:199–214, 2016.

Tudor Berariu, Wojciech Czarnecki, Soham De, Jorg Bornschein, Samuel Smith, Razvan Pascanu, and Claudia Clopath. A study on the plasticity of neural networks. *arXiv preprint arXiv:2106.00042*, 2021.

Alex Damian, Eshaan Nichani, and Jason D Lee. Self-stabilization: The implicit bias of gradient descent at the edge of stability. *arXiv preprint arXiv:2209.15594*, 2022.

Quentin Delfosse, Patrick Schramowski, Martin Mundt, Alejandro Molina, and Kristian Kersting. Adaptive rational activations to boost deep reinforcement learning. *arXiv preprint arXiv:2102.09407*, 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. Ieee, 2009.

Shibhansh Dohare, J Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A Rupam Mahmood, and Richard S Sutton. Loss of plasticity in deep continual learning. *Nature*, 632(8026):768–774, 2024.

Mohamed Elsayed and A Rupam Mahmood. Addressing loss of plasticity and catastrophic forgetting in continual learning. *arXiv preprint arXiv:2404.00781*, 2024.

Zhiqiang He and Zhi Liu. Plasticity-aware mixture of experts for learning under qoe shifts in adaptive video streaming, 2025. URL https://arxiv.org/abs/2504.09906.

Iván Izquierdo, Jorge H Medina, Mônica RM Vianna, Luciana A Izquierdo, and Daniela M Barros. Separate mechanisms for short-and long-term memory. *Behavioural Brain Research*, 103(1):1–11, 1999.

Achref Jaziri, Etienne Künzel, and Visvanathan Ramesh. Mitigating the stability-plasticity dilemma in adaptive train scheduling with curriculum-driven continual dqn expansion. *arXiv preprint arXiv:2408.09838*, 2024.

Dahuin Jung, Dongjin Lee, Sunwon Hong, Hyemi Jang, Ho Bae, and Sungroh Yoon. New insights for the stability-plasticity dilemma in online continual learning. *arXiv preprint arXiv:2302.08741*, 2023.

Dongwan Kim and Bohyung Han. On the stability-plasticity dilemma of class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20196–20204, 2023.

Nicholas I Kuo, Mehrtash Harandi, Nicolas Fourrier, Christian Walder, Gabriela Ferraro, Hanna Suominen, et al. Plastic and stable gated classifiers for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3553–3558, 2021.

Hojoon Lee, Hyeonseo Cho, Hyunseung Kim, Donghu Kim, Dugki Min, Jaegul Choo, and Clare Lyle. Slow and steady wins the race: Maintaining plasticity with hare and tortoise networks. *arXiv preprint arXiv:2406.02596*, 2024.

Zhouhan Lin, Roland Memisevic, and Kishore Konda. How far can we go without convolution: Improving fully-connected networks. *arXiv preprint arXiv:1511.02580*, 2015.

Jiashun Liu, Johan Obando-Ceron, Aaron Courville, and Ling Pan. Neuroplastic expansion in deep reinforcement learning, 2025. URL https://arxiv.org/abs/2410.07994.

Clare Lyle, Zeyu Zheng, Khimya Khetarpal, Hado van Hasselt, Razvan Pascanu, James Martens, and Will Dabney. Disentangling the causes of plasticity loss in neural networks. *arXiv preprint arXiv:2402.18762*, 2024.

Arthur W Melton, KH Pribram, and DE Broadbent. Short-and long-term postperceptual memory: Dichotomy or continuum? In *Biology of Memory*, pp. 3–5. Academic Press, 1970.

Martial Mermillod, Aurélia Bugaiska, and Patrick Bonin. The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects, 2013.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural networks*, 113:54–71, 2019.

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910*, 2018.

Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In *International Conference on Machine Learning*, pp. 4528–4537. PMLR, 2018.

Thomas Scialom, Tuhin Chakrabarty, and Smaranda Muresan. Fine-tuned language models are continual learners. *arXiv preprint arXiv:2205.12393*, 2022.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in Neural Information Processing Systems*, 30, 2017.

James Seale Smith, Junjiao Tian, Shaunak Halbe, Yen-Chang Hsu, and Zsolt Kira. A closer look at rehearsal-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2410–2420, 2023.

Ghada Sokar, Rishabh Agarwal, Pablo Samuel Castro, and Utku Evci. The dormant neuron phenomenon in deep reinforcement learning. In *International Conference on Machine Learning*, pp. 32145–32168. PMLR, 2023.

Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big Data*, 3:1–40, 2016.

Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547*, 2017.

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.

A Appendix

A.1 Evaluation Setting

The Continual ImageNet benchmark uses a downsampled 32×32 version of ImageNet, consisting of 1,000 classes with 700 images per class(600 for training and 100 for testing). Each binary classification task involves training on 1,200 images and testing on 200 images. Models are trained over 250 epochs using mini-batches of 100. We trained for 5 runs of 5000 tasks each.

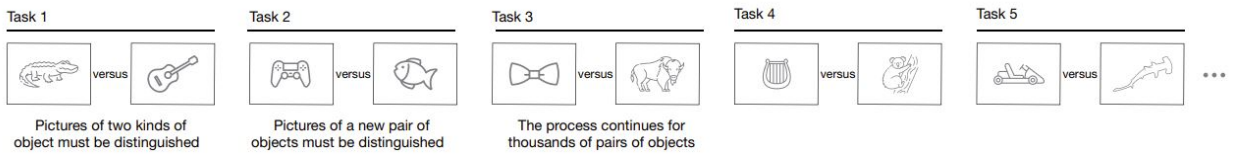


Figure 4: Continual ImageNet (Dohare et al., 2024)

A.2 Example Images

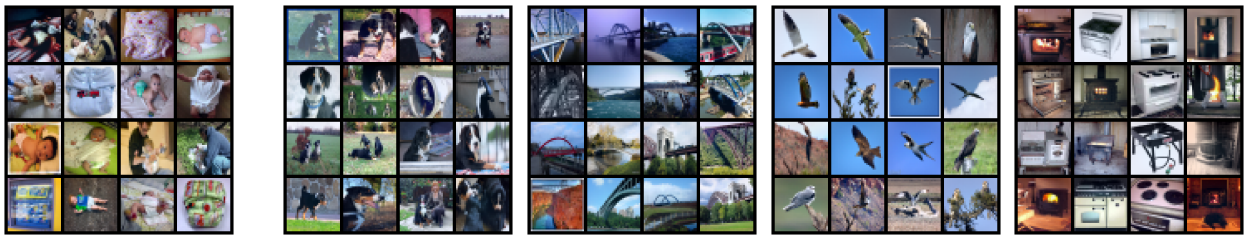


Figure 5: Example Images of the downsampled ImageNet Dataset. : Baby, Dog, Bridge, Bird, Oven

A.3 Activation Functions

Figure 6 illustrates the activation functions used in our experiments. The first is the standard Rectified Linear Unit, followed by the hyperbolic tangent function. The third is the initialization of the Pade Activation Unit, which we chose to initialize as a ReLU, but then adapts during training.
Lastly, our proposed activation function, ReLUDown, is shown with three distinct hinge points from which we only used d=-3.

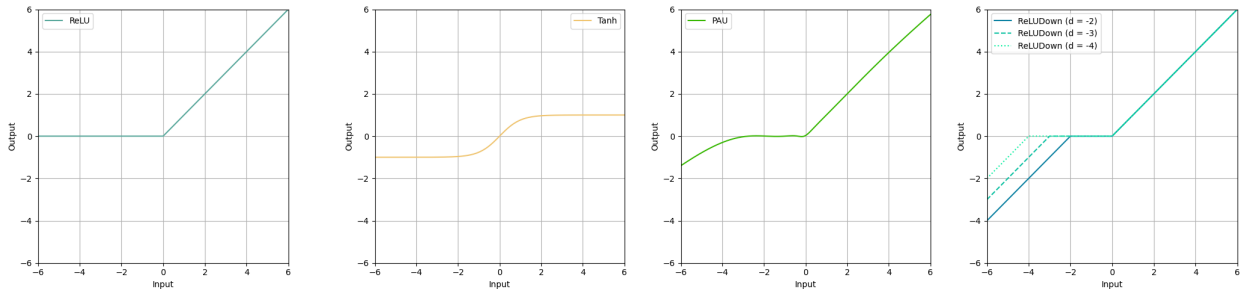


Figure 6: Activation Function used in the Experiments: ReLU, Tanh, PAU, ReLUDown

A.4 Decrease Backpropagation

In Figure 7, we observe a decline in backpropagation activity for our algorithm (RDBP). All layers start with the same gradient factor of one, making it able to learn the basic structures of the environment. With further training, the gradient factor is reduced. After Task 1000, the gradient factor stabilizes, with the first convolutional layer receiving only 25% of the backpropagation signal compared to a standard network. In contrast, the classification head maintains full adaptability with no reduction in gradient flow.
It will be interesting to investigate how altering the hyperparameters for the Speed Factor and Decrease Factor affects the algorithm's performance in retaining and acquiring knowledge, as well as to explore the potential impact of alternative annealing strategies.

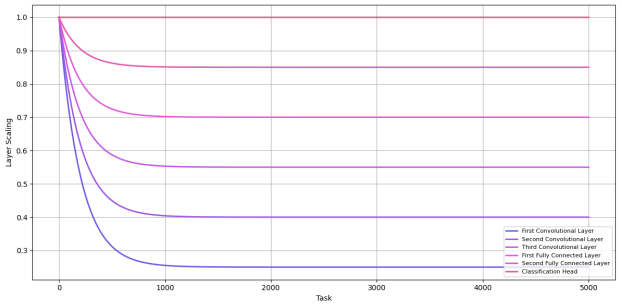


Figure 7: Gradient Factor for the different Layers of the DBP algorithm with a Speed Factor of $a = 1.005$ and a Decrease Factor of $f = 0.15$ based on the task.

A.5 RESULTS

A.5.1 PLASTICITY

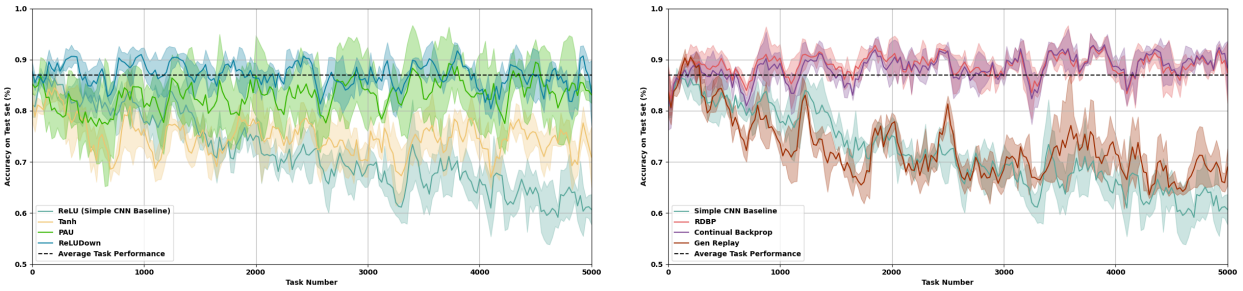


Figure 8: Current Task Performance of a CNN to evaluate plasticity with different Activation Functions: ReLU, Tanh, ReLUDown, PAU on the left and RDBP, Continual Backpropagation, and Generative Replay on the right. The average Performance of a CNN fully reset at each task is included as a black dotted line.

A.5.2 STABILITY

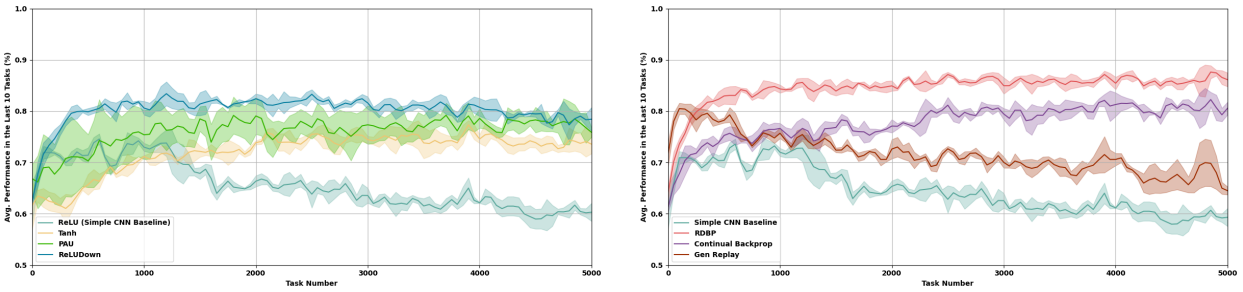


Figure 9: Average Performance in the last 10 Tasks to evaluate the Stability of a CNN with different Activation Functions: ReLU, Tanh, ReLUDown, PAU on the left and RDBP, Continual Backpropagation, and Generative Replay on the right.

A.5.3 TIME

Table 10 presents the relative training times of various activation functions and continual learning strategies, expressed as percentage increases compared to a baseline convolutional network using ReLU.

The Tanh activation function introduces a negligible computational overhead. In contrast, PAUs result in a substantial 60% increase in training time, attributed to the additional trainable parameters that are continuously updated during learning.

The ReLUDown activation function leads to a 34% increase in training time relative to standard ReLU. We hypothesize that the relatively low training time of the convolutional network with ReLU is due to the predominantly zero gradients of dormant neurons in later tasks, a problem our activation function does not suffer.

Algorithm	Function	Rel. Train Time
Conv-Net	ReLU	0%
Conv-Net	Tanh	1%
Conv-Net	PAU	60%
Conv-Net	ReLUDown	34%
Decrease Backprop.	ReLUDown	35%
Continual Backprop.	ReLU	26%
Generative Replay	ReLU	81%

Figure 10: Training time per task for different algorithms.

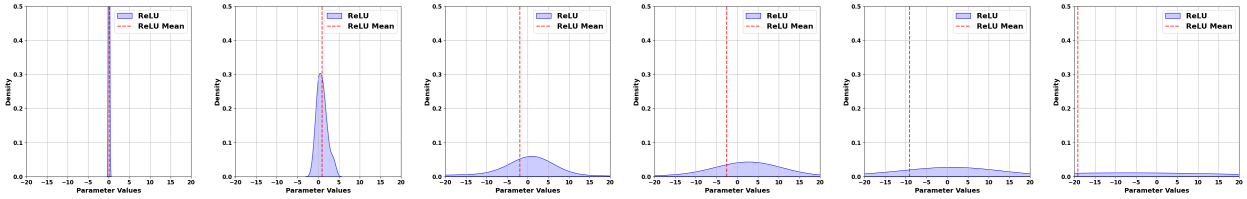
Applying Decrease Backpropagation in conjunction with ReLUDown does not result in any further significant overhead beyond that introduced by ReLUDown alone.

Finally, Generative Replay using a VAE incurs the highest computational cost, increasing training time by 81% compared to standard ReLU. This is due to the necessity of training a generative model alongside the classifier.

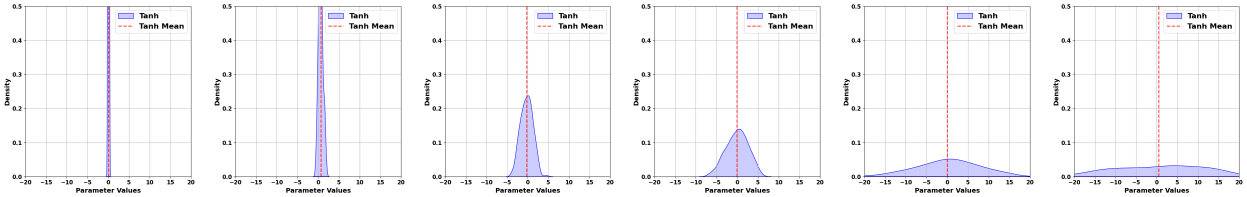
A.5.4 PREACTIVATION DISTRIBUTIONS OF ACTIVATION FUNCTIONS

In this section, we visualize and compare the preactivation distributions observed across ReLU, Tanh, PAU, and ReLUDown(Ours). This analysis provides insights into how each function responds to a shifting input distribution. We took representative distributions for Task 0, 100, 500, 1000, 3000 and 5000.
When using the ReLU activation function, the preactivation distribution exhibits a consistent shift toward negative values and becomes increasingly dispersed. As a result, most activations are zero with zero gradients. This worsens adaptation to new tasks and learning performance.
Although Tanh performs worse than ReLU on the initial tasks, it retains more plasticity over time, allowing it to outperform ReLU in later tasks. This is reflected in its preactivation distribution, which stays at a mean of zero with a growing standard deviation.
The preactivation distribution of the Network with PAUs stays around a zero mean but displays fluctuations of the distribution shape.
ReLUDown begins with a similar preactivation distribution as other activation functions. However, unlike them, its distribution gradually converges toward a near-normal shape, exhibiting only minor fluctuations in both mean and standard deviation.

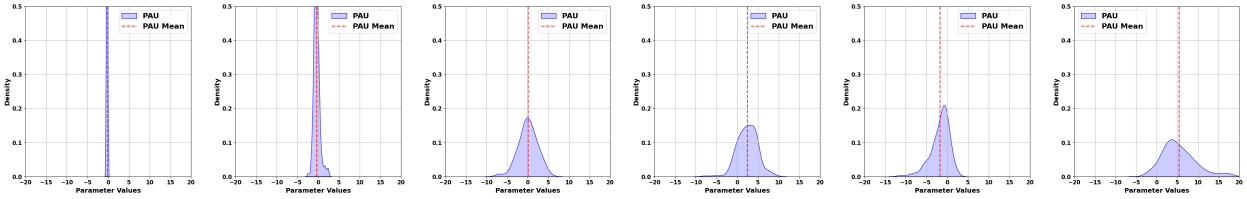
A.5.5 RELU



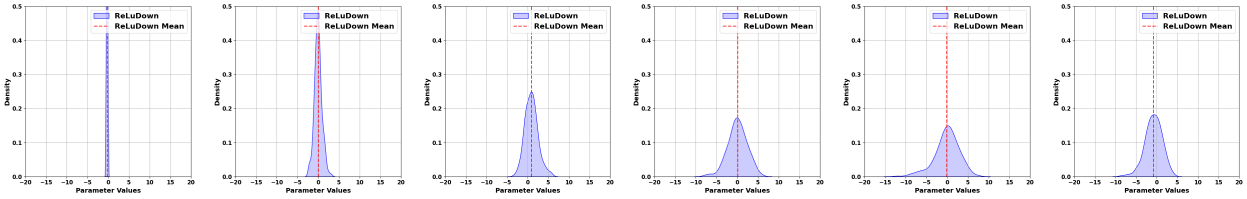
A.5.6 TANH



A.5.7 PAU



A.5.8 RELUDOWN



A.6 NETWORK AND HYPERPARAMETERS

Network Layers		
Layer	**Output Shape**	**Activation**
Input	$3 \times 32 \times 32$	–
Conv2d (3, 32, 5)	$32 \times 28 \times 28$	ReLU, Tanh, PAU, ReLUDown
MaxPool2d (2, 2)	$32 \times 14 \times 14$	–
Conv2d (32, 64, 3)	$64 \times 12 \times 12$	ReLU, Tanh, PAU, ReLUDown
MaxPool2d (2, 2)	$64 \times 6 \times 6$	–
Conv2d (64, 128, 3)	$128 \times 4 \times 4$	ReLU, Tanh, PAU, ReLUDown
MaxPool2d (2, 2)	$128 \times 2 \times 2$	–
Flatten	$128 \times 4 = 512$	–
Linear (512, 128)	128	ReLU, Tanh, PAU, ReLUDown
Linear (128, 128)	128	ReLU, Tanh, PAU, ReLUDown
Linear (128, 2)	2	ReLU, Tanh, PAU, ReLUDown
Loss Function: Cross Entropy Loss		
Parameter	**Symbol**	**Value**
Step Size	-	0.010
Weight Decay	-	0.000
Momentum	-	0.900
ReLuDown		
Parameter	**Symbol**	**Value**
Hinge Point	d	3
Decrease Backpropagation		
Parameter	**Symbol**	**Value**
Decrease Factor	f	0.150
Speed factor	a	1.005
Padé Activation Units		
Parameter	**Symbol**	**Value**
Nominator polynomial	m	5
Denominator polynomial	n	4
Initialization Shape	-	ReLU
Continual Backpropagation		
Parameter	**Symbol**	**Value**
Replacement Rate	-	0.0001
Decay Rate	-	0.99
Maturity Treshold	-	100
Initialization	-	Kaiming
Utilization Type	-	Contribution
Generative Replay: Variational Autoencoder		
Parameter	**Symbol**	**Value**
Step Size	-	0.01
Replay Percentage	-	1/6
Layer	**Output Shape**	**Activation**
Input (Image + Label)	$4 \times 32 \times 32$	–
Conv2d (4, 32, 4, 2, 1)	$32 \times 16 \times 16$	ReLU
Conv2d (32, 64, 4, 2, 1)	$64 \times 8 \times 8$	ReLU
Conv2d (64, 128, 4, 2, 1)	$128 \times 4 \times 4$	ReLU
Flatten	2048	–
FC $\rightarrow \mu, \log \sigma^2$	2×128	–
z + Label Embedding	130	–
FC $\rightarrow$ 2048	2048	–
Unflatten	$128 \times 4 \times 4$	–
ConvT (128 $\rightarrow$ 64)	$64 \times 8 \times 8$	ReLU
ConvT (64 $\rightarrow$ 32)	$32 \times 16 \times 16$	ReLU
ConvT (32 $\rightarrow$ 3)	$3 \times 32 \times 32$	Sigmoid