Exploring the Limits of Feature Learning in Continual Learning

Jacopo Graldi[*] graldij@ethz.ch
Giulia Lanzillotta[†‡] giulia.lanzillotta@ai.ethz.ch
Lorenzo Noci[†]
Benjamin F. Grewe[§]
Thomas Hofmann[†]

Abstract

Despite the recent breakthroughs in deep learning, neural networks still struggle to learn continually in non-stationary environments, and the reasons are poorly understood. In this work, we perform an empirical study on the role of *feature learning* and *scale* on *catastrophic forgetting* by applying the precepts of the theory on neural networks scaling limits. We interpolate between lazy and rich training regimes, finding that the optimal amount of feature learning is modulated by task similarity. Surprisingly, our results consistently show that more feature learning increases catastrophic forgetting and that scale only helps when yielding more laziness. Supported by empirical evidence on a variety of benchmarks, our work provides the first unified understanding of the role of scale in the different training regimes and parameterizations for continual learning.

1 Introduction

The learning paradigm most adopted for modern neural networks (NN) has achieved impressive results on many hard benchmarks, recently beating all expectations with the introduction of large autoregressive language models [28]. However, this paradigm has been repeatedly and consistently shown to be ill-suited to deal with *non-stationarities* in the input data distribution. Interference between multiple tasks leads to a reduction of performance, termed *catastrophic forgetting* (CF), when the tasks are learned sequentially [21].

Research in continual learning has worked towards solutions to mitigate CF by adapting the learning paradigm [14, 39, 20, 7, 32]. Besides these solution-oriented studies, various studies have been focusing on the theoretical understanding of the mechanisms at the roots of catastrophic forgetting in simplified settings [1, 4, 19, 5, 6]. A question of particular theoretical interest has been to understand the impact of task similarity [18, 9], or overparameterization [8, 9] on catastrophic forgetting. Almost contemporaneously, other empirical studies have significantly contributed to understanding the causes of CF in modern neural networks [22, 23, 30].

This work presents an empirical study of *the role of feature learning and scale on catastrophic forgetting*. Differently from previous work, we follow the literature on scaling limits to control separately for feature learning and scale. More concretely, we interpolate between the two extremes of the parameterization spectrum, namely the Neural Tangent Parameterization (NTP) [13] – characterized by network parameters that hardly vary from initialization (*lazy regime*) – and the Maximal Update

[*]Dept. of Information Technology and Electrical Engineering, ETH Zurich, Switzerland.
[†]Dept. of Computer Science, ETH Zurich, Switzerland
[‡]ETH AI Center, Switzerland
[§]Institute of Neuroinformatics - University of Zurich and ETH Zurich, Switzerland

NeurIPS 2024 Workshop on Scalable Continual Learning for Lifelong Foundation Models.

Parameterization (μP) [36, 2] – where feature learning is preserved maximally at every layer (*rich regime*). By disentangling the model scale from the learning regime we are able to evaluate the contribution of each element to CF, thereby resolving the existing contradictions regarding the role of scale in CF.

Importantly for our work, Ramasesh et al. [31] investigated the role of scale in CF with and without pretraining, finding that scale helps when using a pretrained model, but not otherwise. Crucially, Mirzadeh et al. [24] and Mirzadeh et al. [25] have observed that increasing the width of a network – but not the depth – reduces CF on various benchmarks. Conversely, Wenger et al. [35] recently denounced that the results in [24] only hold for very short training, attributing the lower CF to a confounding lower training accuracy, and therefore questioning the role of width in CF. Additionally, Guha and Lakshman [10] theoretically formalized the diminishing returns of width in continual learning, further questioning the true effect of width on CF.

In this work, we find that increasing feature learning consistently leads to increased CF, and in general, we observe that *laziness is beneficial for forgetting*. In this light, we expose that *scale does not help CF* in general. In particular, scale helps CF solely when increasing the width of the model yields more laziness, thus explaining the contradictory results in the literature regarding the role of width in CF (e.g. [24] vs. [35]). Additionally, we *optimize the tradeoff of plasticity and stability* by finding the optimal amount of feature learning, which shows a surprising transfer across widths. Lastly, we show that task similarity modulates the effect of scale and laziness on CF.

2 Background and Metrics

Scaling Limits, Parameterizations, and Training Regimes The literature on scaling limits [26, 13, 17, 36] has provided several useful tools to deep learning theory. In particular, depending on the parameterization of the network forward function, one obtains different learning behaviors. For instance, the NTP used in the Neural Tangent Kernel (NTK) theory [13], leads in the infinite-width limit to a linearization of the dynamics of the network [17], and *the change in networks' parameters and features* converges to 0. By contrast, the μP achieves *maximal feature learning* in the infinite-width limit. Following the notation of Bordelon et al. [3], we consider here a simplified residual network of width N and L residual blocks, with the weights of layer l initialized as $W^l_{ij} \sim \mathcal{N}(0, \sigma_l^2)$. For an input $\mathbf{x} \in \mathbb{R}^D$, the preactivations of the first block are defined as $\mathbf{h}^1(\mathbf{x}) = \beta_0 \mathbf{W}^0 \mathbf{x}$. Conversely, the N-dimensional preactivations of the l-th block have a residual branch scaled by β_l, and the outputs $f(\mathbf{x}) \in \mathbb{R}$ are additionally inversely scaled by γ:

$$\mathbf{h}^{l+1}(\mathbf{x}) = \mathbf{h}^l(\mathbf{x}) + \beta_l \mathbf{W}^l \phi(\mathbf{h}^l(\mathbf{x})), \quad f(\mathbf{x}) = \frac{\beta_L}{\gamma} \mathbf{w}^L \cdot \phi(\mathbf{h}^L(\mathbf{x})). \tag{1}$$

The choice of how the scaling factors β_l and γ, the weights initialization variance σ_l^2, and the (possibly time-varying) learning rate $\eta(t)$ should scale as width N grows, differentiate the NTP and μP as per Tab. 1. Note that the NTP corresponds to the standard parameterization of PyTorch. Moreover, by varying the γ_0 parameter it is possible to smoothly interpolate between lazy ($\gamma_0 \to 0$) and rich ($\gamma_0 = 1$) regimes.[5]

Continual Learning and Catastrophic Forgetting In CL the models are typically evaluated on the *average test accuracy* after learning all tasks sequentially [24]. Catastrophic forgetting is evaluated after training on each new task as the average decrease in accuracy on the past tasks. Additionally, we measure the *average learning accuracy*, which is the average accuracy on the task after training on it.[6] In contrast to previous works, we introduce a novel metric to measure CF: the *Catastrophic Forgetting rate* (CFr). For a benchmark with T tasks, define the test accuracy on the task $i \in \{0, ..., T-1\}$ after training on task $t \in \{0, ..., T-1\}$ as $a_{t,i}$. Next, we define the CFr as

$$\text{CFr} = \frac{1}{T-1} \sum_{i=0}^{T-2} \frac{\max_{t \in \{i,...,T-2\}}(a_{t,i}) - a_{T-1,i}}{\max_{t \in \{i,...,T-2\}}(a_{t,i})}. \tag{2}$$

This metric is motivated by the observation mentioned above of Wenger et al. [35] regarding the training accuracy, and it decouples CF and the learning accuracy avoiding confounding effects.

[5] A gentle introduction into this alternative derivation can be found in the lecture notes of Pehlevan and Bordelon [29].

[6] See Appendix B.

3 *Lazy* and *Rich* Regimes in Continual Learning

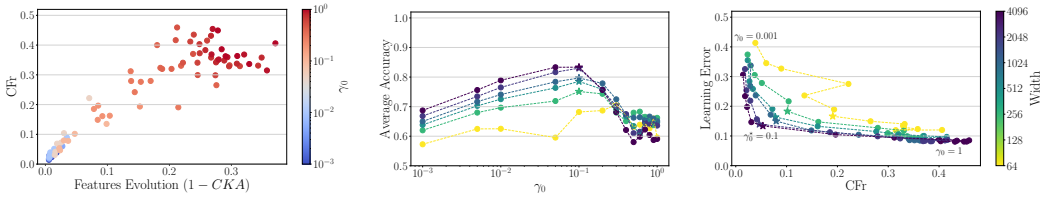
3.1 Increasing Feature Learning via γ_0 Variation Leads to Increased CF

We investigate the effect of feature learning on CF by varying the γ_0 parameter in the μP, which allows us to interpolate smoothly between the two regimes for models of various widths. This experimental methodology was already successfully employed to study phenomena tied to network training dynamics, such as grokking [16]. We execute these experiments on the Split-CIFAR10 continual learning benchmark (further details in Appendix C).

Firstly, we monitor how the internal representation of a batch of data coming from a task changes while training on the next tasks. In particular, we compare the activations at every residual block of the model against the activations of the same data after other tasks have been trained. We use CKA [15] as a similarity measure, and we average over all tasks. As expected, we observe that increasing the degree of feature learning increases the average evolution of the activations (i.e. $1-CKA$) (Fig. 1(a)). Crucially, we consistently see that, for all the widths, *higher features evolution corresponds to higher forgetting* (Fig. 1(a), 3). This analysis leads us to the following conclusion regarding the nature of feature learning in modern neural networks: **feature learning is destructive on non-stationary distributions and increases catastrophic forgetting** .

Knowing that more feature learning leads to higher learning accuracy [33] and, as just noted, higher forgetting, we hypothesize that the average accuracy might be highest at an intermediate level of feature learning. Interestingly on Split-CIFAR10 the optimal trade-off between forgetting and learning accuracy is achieved at a relatively low γ_0 value of $\gamma_0^\star \approx 0.1$ (Fig. 1(b)). It is even more noteworthy that **the optimal degree of feature learning – $\gamma_0^\star$ – shows a striking transfer property across widths**. Thus, one can optimize the performance of a CL pipeline by finding the optimal $\gamma_0^\star$ for low widths, which then transfers to larger scales, as recently proposed for non-CL-specific hyperparameters [37, 38, 3, 27].

In Fig. 1(c) we visualize the trade-off between learning accuracy and forgetting. It is interesting to observe that $\gamma_0^\star$ separates two distinct regimes, consistently across widths: below $\gamma_0^\star$ the learning error rapidly reduces with a small increase in forgetting, while above $\gamma_0^\star$ the learning error has saturated, and the increase in forgetting is far more pronounced. We interpret this $\gamma_0^\star$ as the minimum feature learning amount needed to effectively learn the task: more feature learning gives diminishing returns in learning accuracy and it harms the performance on old tasks.



(a) (b) (c)

Figure 1: Smooth interpolation between lazy ($\gamma_0 \to 0$) and rich regimes ($\gamma_0 = 1$) on the Split-CIFAR10 dataset. Points are the average over 5 random seeds. (a) Average evolution ($1-CKA$) of features at convergence and at later points in training that scales with CFr. (b) Average Accuracy for various widths and optimal $\gamma_0^\star \approx 0.1$; (c) tradeoff between learning error and catastrophic forgetting.

3.2 Scale Does Not Always Ameliorate CF

In the μP the magnitude of features is independent of the width of the model (assumed large enough width) [34]; this allows us to separate the effect of scale from the degree of feature learning. In fact, we clearly observe from Fig. 1 that when feature learning is abundant ($\gamma_0 \geq \gamma_0^\star$), width does not improve performance. In other words, our results show that **it is not true in general that scaling up decreases forgetting** . On the other hand, in the NTP this decoupling between scale and training regime is not possible, as scaling the width increases the laziness of the model, which, as we have seen before, reduces CFr (Fig. 4). Therefore, the findings of Mirzadeh et al. [24] are confirmed in the context of the NTK parameterization: in this case, we find that **scale reduces CF if it increases laziness in training**.

Through the perspective on parameterization and training regimes, we can also reconcile the criticism of Wenger et al. [35] towards Mirzadeh et al. [24] regarding the observation that the benefit of width disappears when the model is trained until convergence. Indeed, both results are consistent and can be explained by the same underlying mechanism of laziness and feature learning: training for a longer time increases the finite-width effects of the NTP, forcing the model out of the NTK lazy regime and into the feature learning regime, where width does not offer any advantage in terms of CFr.

While a comprehensive investigation on the effect of depth on CF is left for future work, we can speculate that the detrimental effect of depth on CF observed by Mirzadeh et al. [24] is explained within our observations as depth notably encourages NTK evolution [2] (i.e. more feature learning), which in turn is detrimental for CF.

3.3 The Interplay with Task Similarity

We include task similarity as another variable in our analyses. Inspired by previous works [9, 14], we use the permuted input MNIST dataset with 5 tasks consisting of the MNIST dataset with a random but fixed permutation of the pixels. The task similarity is varied by controlling the number of pixels that are permuted between tasks (further details in Appendix C).

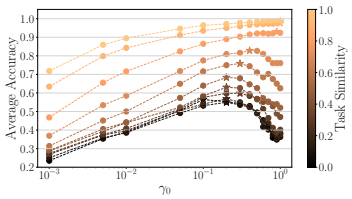


Figure 2: Results on the synthetic dataset Permuted Input MNIST. (a) Average accuracy for various γ_0 and task similarities; $\gamma_0^\star$ shifts with task similarity.

In Fig. 2 we plot the average accuracy as we vary the amount of feature learning on tasks of different similarity. We observe that task similarity is positively correlated with the value of γ_0 maximizing the average accuracy ($\gamma_0^\star$). This observation has an intuitive explanation. By decreasing the task similarity we inject more non-stationarity into the input data distribution: **by design, feature learning is optimal in stationary settings, whereas in non-stationary settings laziness provides crucial robustness against forgetting, thus shifting $\gamma_0^\star$ towards 0**.

Secondly, we inspect the test error on the first task, after training on all other tasks (Fig. 5(a)). We uncover a non-monotonic relationship between the error and task similarity. Similarly to prior work [9, 30] we observe the highest error for intermediate levels of task similarity. To the best of our knowledge, this behavior has never been observed in the feature learning regime and was only analyzed in fixed-features settings. Our findings show that the same trend is observed in lazy and rich regimes and that it is amplified with more feature learning.

Lastly, we observe that the laziness of the model is influenced by the task similarity: a lower task similarity induces higher feature evolution for a given γ_0 (Fig. 5(b)). This implies that the minimum width necessary to achieve lazy training in the practice depends on the task similarity. Therefore, a theoretical assumption such as laziness in the training dynamics holds in finite widths only in certain ranges of task similarity, and including assumptions on task similarity appears to be crucial in the theoretical models of CF.

In Appendix D.3 we study a more realistic kind of task similarity by varying the number of classes per task with the Split-TinyImagenet dataset.

4 Discussion and Conclusion

In this work, we study the role of feature learning and scale in CF. The powerful toolbox of network parameterizations, developed in the recent literature on scaling limits, allows us to decouple the entangled relationships between these aspects. We are thus able to resolve contradictions regarding the role of scale in CF, showing that scaling reduces CF only by increasing laziness, and thus its effect is tied to the parameterization. More surprisingly, our experimental results conclusively demonstrate the intrinsic inadequacy of modern feature learning in non-stationary environments.

These results encourage a broader reflection: the DL community has for years focused its efforts on optimizing feature learning for plasticity, i.e. better learning accuracy and transfer. However, a second fundamental aspect of learning is memory stability, which, as indicated by our work, is achieved in the current learning paradigm only at the expense of plasticity. Developing better solutions to CF may necessitate a fundamental rethinking of optimization strategies to enable both plasticity and stability without compromising either.

Acknowledgements

LN would like to acknowledge the support of a Google PhD research fellowship.

References

[1] Mehdi Abbana Bennani, Thang Doan, and Masashi Sugiyama. Generalisation guarantees for continual learning with orthogonal gradient descent. *arXiv preprint arXiv:2006.11942*, 2020.

[2] Blake Bordelon and Cengiz Pehlevan. Self-consistent dynamical field theory of kernel evolution in wide neural networks. *Advances in Neural Information Processing Systems*, 35:32240–32256, 2022.

[3] Blake Bordelon, Lorenzo Noci, Mufan Bill Li, Boris Hanin, and Cengiz Pehlevan. Depthwise hyperparameter transfer in residual networks: Dynamics and scaling limit. In *The Twelfth International Conference on Learning Representations*, 2023.

[4] Thang Doan, Mehdi Abbana Bennani, Bogdan Mazoure, Guillaume Rabusseau, and Pierre Alquier. A theoretical analysis of catastrophic forgetting through the ntk overlap matrix. In *International Conference on Artificial Intelligence and Statistics*, pages 1072–1080. PMLR, 2021.

[5] Itay Evron, Edward Moroshko, Rachel Ward, Nathan Srebro, and Daniel Soudry. How catastrophic can catastrophic forgetting be in linear regression? In *Conference on Learning Theory*, pages 4028–4079. PMLR, 2022.

[6] Itay Evron, Edward Moroshko, Gon Buzaglo, Maroun Khriesh, Badea Marjieh, Nathan Srebro, and Daniel Soudry. Continual learning in linear classification on separable data. In *International Conference on Machine Learning*, pages 9440–9484. PMLR, 2023.

[7] Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *International Conference on Artificial Intelligence and Statistics*, pages 3762–3773. PMLR, 2020.

[8] Daniel Goldfarb and Paul Hand. Analysis of catastrophic forgetting for random orthogonal transformation tasks in the overparameterized regime. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 2975–2993. PMLR, 25–27 Apr 2023. URL `https://proceedings.mlr.press/v206/goldfarb23a.html`.

[9] Daniel Goldfarb, Itay Evron, Nir Weinberger, Daniel Soudry, and PAul HAnd. The joint effect of task similarity and overparameterization on catastrophic forgetting — an analytical model. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=u3dHl287oB`.

[10] Etash Kumar Guha and Vihan Lakshman. On the diminishing returns of width for continual learning. In *Forty-first International Conference on Machine Learning*, 2024. URL `https://openreview.net/forum?id=Ld255Mbx9F`.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

[13] Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. 31, 2018. URL `https://proceedings.neurips.cc/paper_files/paper/2018/file/5a4be1fa34e62bb8a6ec6b91d2462f5a-Paper.pdf`.

[14] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[15] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMLR, 2019.

[16] Tanishq Kumar, Blake Bordelon, Samuel J Gershman, and Cengiz Pehlevan. Grokking as the transition from lazy to rich training dynamics. In *The Twelfth International Conference on Learning Representations*, 2024.

[17] Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *Advances in neural information processing systems*, 32, 2019.

[18] Sebastian Lee, Sebastian Goldt, and Andrew Saxe. Continual learning in the teacher-student setup: Impact of task similarity. In *International Conference on Machine Learning*, pages 6109–6119. PMLR, 2021.

[19] Sen Lin, Peizhong Ju, Yingbin Liang, and Ness Shroff. Theory on forgetting and generalization of continual learning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 21078–21100. PMLR, 23–29 Jul 2023. URL `https://proceedings.mlr.press/v202/lin23f.html`.

[20] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

[21] Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pages 109–165. Academic Press, 1989. doi: https://doi.org/10.1016/S0079-7421(08)60536-8. URL `https://www.sciencedirect.com/science/article/pii/S0079742108605368`.

[22] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Razvan Pascanu, and Hassan Ghasemzadeh. Understanding the role of training regimes in continual learning. *Advances in Neural Information Processing Systems*, 33:7308–7320, 2020.

[23] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Dilan Gorur, Razvan Pascanu, and Hassan Ghasemzadeh. Linear mode connectivity in multitask and continual learning. In *The Ninth International Conference on Learning Representations (ICLR 2022)*, 2021.

[24] Seyed Iman Mirzadeh, Arslan Chaudhry, Dong Yin, Huiyi Hu, Razvan Pascanu, Dilan Gorur, and Mehrdad Farajtabar. Wide neural networks forget less catastrophically. In *International conference on machine learning*, pages 15699–15717. PMLR, 2022.

[25] Seyed Iman Mirzadeh, Arslan Chaudhry, Dong Yin, Timothy Nguyen, Razvan Pascanu, Dilan Gorur, and Mehrdad Farajtabar. Architecture matters in continual learning. *arXiv preprint arXiv:2202.00275*, 2022.

[26] Radford M. Neal. *Bayesian Learning for Neural Networks*. Lecture Notes in Statistics. Springer New York, New York, NY, 1996.

[27] Lorenzo Noci, Alexandru Meterez, Thomas Hofmann, and Antonio Orvieto. Why do learning rates transfer? reconciling optimization and scaling limits for deep learning. *arXiv preprint arXiv:2402.17457*, 2024.

[28] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[29] Cengiz Pehlevan and Blake Bordelon. Lecture notes on infinite-width limits of neural networks. 2023.

[30] Vinay V Ramasesh, Ethan Dyer, and Maithra Raghu. Anatomy of catastrophic forgetting: Hidden representations and task semantics. *arXiv preprint arXiv:2007.07400*, 2020.

[31] Vinay Venkatesh Ramasesh, Aitor Lewkowycz, and Ethan Dyer. Effect of scale on catastrophic forgetting in neural networks. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=GhVS8_yPeEa`.

[32] Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. *arXiv preprint arXiv:2103.09762*, 2021.

[33] Nikhil Vyas, Yamini Bansal, and Preetum Nakkiran. Limitations of the ntk for understanding generalization in deep learning. *arXiv preprint arXiv:2206.10012*, 2022.

[34] Nikhil Vyas, Alexander Atanasov, Blake Bordelon, Depen Morwani, Sabarish Sainathan, and Cengiz Pehlevan. Feature-learning networks are consistent across widths at realistic scales. *Advances in Neural Information Processing Systems*, 36, 2024.

[35] Jonathan Wenger, Felix Dangel, and Agustinus Kristiadi. On the disconnect between theory and practice of overparametrized neural networks. *arXiv preprint arXiv:2310.00137*, 2023.

[36] Greg Yang and Edward J Hu. Feature learning in infinite-width neural networks. *arXiv preprint arXiv:2011.14522*, 2020.

[37] Greg Yang, Edward J Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer. *arXiv preprint arXiv:2203.03466*, 2022.

[38] Greg Yang, Dingli Yu, Chen Zhu, and Soufiane Hayou. Tensor programs vi: Feature learning in infinite-depth neural networks. *arXiv preprint arXiv:2310.02244*, 2023.

[39] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. PMLR, 2017.

Appendices

A Scaling Limits and Parameterizations

Inspired by seminal work of Neal [26], Jacot et al. [13] has shown that if a network has infinite width and a specific parameterization (the NTP), its training dynamics can be understood as a kernel method. The associated kernel, the Neural Tangent Kernel (NTK), stays constant during training, leading to a linearization of the dynamics of the network [17]. Importantly, Jacot et al. [13] showed that with the NTP *the change in networks' parameters and features* converges to 0, leading to what is called the *lazy training regime*.

In contrast to the NTK regime and its parameterization, Yang and Hu [36] have studied a more general parameterization that allows non-vanishing changes in the network's features as the width is scaled up. In particular, a novel parameterization – μP – achieves maximal feature learning in the infinite-width limit. This parameterization has been alternatively derived by Bordelon and Pehlevan [2], with the explicit possibility to smoothly interpolate (by varying the γ_0 parameter) between lazy ($\gamma_0 \to 0$) and rich ($\gamma_0 = 1$) regimes.

Tab. 1, together with the notation introduced above, defines the two parameterizations: NTP and μP.

Table 1: Branch and output scales, learning rate, and weight variance in the two parameterizations: NTP (PyTorch default) and μP.

	NTP	μ**P**
Branch Scale β_l	1	$\begin{cases} N^{-1/2}, & l > 0 \\ D^{-1/2}, & l = 0 \end{cases}$
Output Scale γ	1	$\gamma_0 N^{1/2}$
LR Schedule $\eta(t)$	$\eta_0(t)$	$\eta_0(t)\gamma_0^2 N$
Weight Variance σ_l^2	$\begin{cases} N^{-1}, & l > 0 \\ D^{-1}, & l = 0 \end{cases}$	1

B Continual Learning Metrics Details

Firstly, for a benchmark with T tasks, we define the test accuracy on the task $i \in \{0, ..., T-1\}$ after training on task $t \in \{0, ..., T-1\}$ as $a_{t,i}$. The used metrics are defined as follows:

- Average Accuracy:
$$\text{A} = \frac{1}{T}\sum_{i=0}^{T-1} a_{T-1,i}.$$
- Learning Accuracy:
$$\text{LA} = \frac{1}{T}\sum_{i=0}^{T-1} a_{i,i}.$$
- Catastrophic Forgetting rate:
$$\text{CFr} = \frac{1}{T-1}\sum_{i=0}^{T-2} \frac{\max_{t\in\{i,...,T-2\}}(a_{t,i}) - a_{T-1,i}}{\max_{t\in\{i,...,T-2\}}(a_{t,i})}.$$

C Experimental Details

Throughout all experiments in this work, we use an architecture of the ResNet family [11]. The base architecture is composed of 6 residual blocks, each with one convolutional layer with 64 channels

(i.e. the *width* of the convolutional model) that we scale up in our experiments. Each block has a batch-normalization layer [12] and ReLU activation function. The μP and NTP are normalized to be equivalent for the base-width architecture. All models are optimized without any regularization, using SGD without momentum and a learning rate which is optimal on the full dataset and at base width. The learning rate follows a cosine annealing scheduling which restarts at each task.

Split-CIFAR10

The Split-CIFAR10 dataset has 5 tasks of 2 classes each (i.e. the 10 classes of CIFAR10 are split into 5 tasks with non-overlapping classes), and as common for task-incremental learning benchmarks, the model uses a separate head for each task. We train each task for 5 epochs.

Permuted MNIST

Inspired by previous works [9, 14], we use the permuted input MNIST dataset with 5 tasks to investigate the impact of task similarity on CF. Specifically, each task of this benchmark consists of the MNIST dataset with a random but fixed permutation of the pixels.

In particular, we consider task similarity as the fraction of pixels that are not permuted between tasks, and the inner square of the image is permuted first. As an example, a task similarity of 1.0 corresponds to the original MNIST dataset; a task similarity of 0.0 corresponds to a dataset where each task permutes all pixels of the images; a task similarity of 0.5 corresponds to tasks where the middle square containing 50% of the pixels is permuted. These tasks are synthetic, and except for the untouched pixels, the tasks are not related to each other: this allows us to artificially cause CF by design. Each task is trained for 5 epochs, and we use the same architecture, hyperparameters search, and optimization as in the previous experiments. The only exception to the experiments on Split-CIFAR10 is the absence of BN layers (following the observation of Mirzadeh et al. [25]).

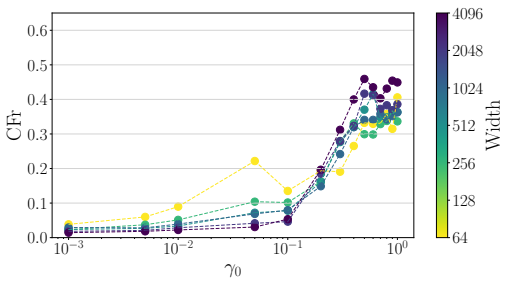
Split-TinyImagenet

Similar to Split-CIFAR10, the classes of TinyImagenet are split into tasks with non-overlapping classes. Throughout the experiments, we consider varying the number of tasks and classes-per-task: 5 tasks of 2 classes each (we denote it 5/2), 5/10, 5/40, as well as 20 tasks of 2 classes each (20/2), and 20/10. We use the same architecture, hyperparameters search, and optimization as in the Split-CIFAR10 experiments.

D Additional Results

D.1 Split-CIFAR10

In Fig. 3 further results for the μP on Split-CIFAR10, while in Fig. 4 for the NTP.



(a)

Figure 3: Split-CIFAR10. Varying the level of feature learning with γ_0 and its effect on forgetting.

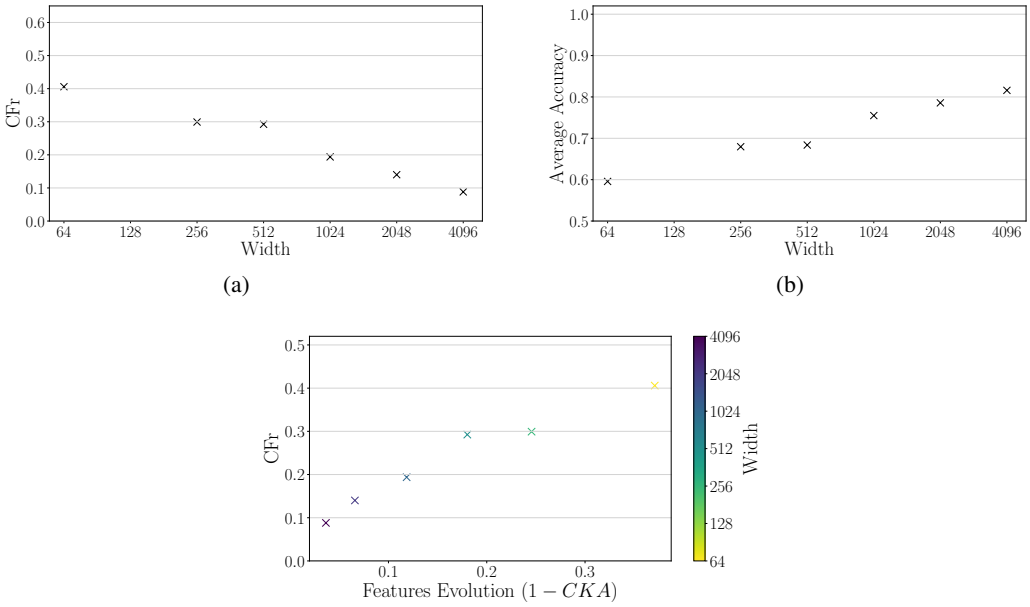


Figure 4: Split-CIFAR10. The effect of scaling the width in the NTP. Increasing the width increases laziness, which thus reduces forgetting.

D.2 Permuted MNIST

In Fig. 5 we report further insights into CF and feature evolution for the experiments with Permuted MNIST.

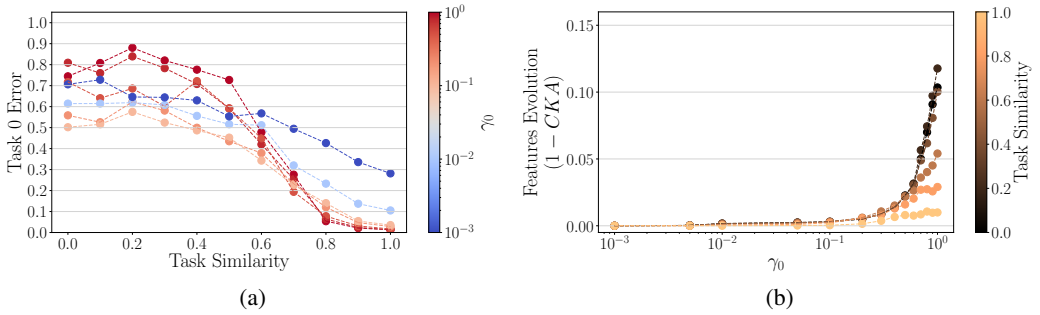


Figure 5: Permuted MNIST results. (a) Error on the first task after all tasks have been trained, for various γ_0 and task similarities. (b) Feature evolution between the converged representation and later representations $(1 - CKA)$ for various γ_0 and task similarities; reducing the task similarity increases the feature evolution.

D.3 Split-TinyImagenet

With the Split-TinyImagenet dataset, we investigate the impact of the number of classes per task on task similarity and CF. In particular, we consider varying the number of tasks and classes-per-task: 5 tasks of 2 classes each (we denote it 5/2), 5/10, 5/40, as well as 20 tasks of 2 classes each (20/2), and 20/10. We use the same architecture, hyperparameters search, and optimization as in the previous experiments.

Intuitively, a higher number of classes per task increases the similarity between tasks, however, it also increases the difficulty of the task. We observe that increasing the number of classes per task

breaks the monotonic relationship between CFr and feature learning observed on CIFAR10 – where increasing feature learning strictly increases forgetting (Fig. 6(a)). The number of tasks does not impact this observation, and for a low number of classes per task, the relation is still monotonic. This non-monotonicity in CFr is reflected also in the average accuracy, where the higher the number of classes, the better it is having a higher degree of feature learning for the overall performance (Fig. 6(b)).

The reason behind this (apparent) deviation from what we have seen so far, is due to the impact of the number of classes per task on the nature of the problem: increasing the number of classes per task increases the variety of the tasks, encouraging the model to learn right away "good" and general features, which happen to generalize well also for later tasks. In this scenario, training on the first task is intuitively similar to pretraining the model on a large dataset, and then deploying it on the CL dataset composed of the remaining $T - 1$ tasks [31].

To analyze the mechanisms of this observation and validate our argument, we first design a set of experiments to gain insights into the amount of feature evolution that occurs on the various tasks. In particular, this time, we compare the internal representations before and after training on a certain task, and we separate the evolution happening during the first task, and for later tasks (as an average). We observe that the feature evolution happening during the first task is significant and grows with the number of classes per task (Fig. 7(a)). On the other hand, and surprisingly, on average the feature evolution for all other tasks is lower for a higher number of classes per task and for γ_0 close to 1 (Fig. 7(b)). The two trends combined, in Fig. 7(c), show that increasing the number of classes together with a high γ_0, i.e. with maximal feature learning, encourages the model to learn features that are useful for later tasks, for which then we observe a *self-induced laziness*. Thus, a high variety of classes in the first task, coupled with feature learning, empowers the model to learn features that are useful for later tasks as well. This equates to a lazy behavior where no features are learned, thus yielding a lower forgetting.

In summary, despite an apparent deviation from the previous observations, feature learning keeps revealing itself as detrimental for forgetting: the improvement in CFr we observe for increased feature learning is caused by self-induced laziness on later tasks thanks to the features learned in the first task. On the contrary when later tasks keep showing feature learning, forgetting increases. The fact that more classes per task induces laziness is in line with the observations above, regarding the impact of laziness on the amount of feature evolution observed for a given γ_0 (or width, in the case of NTP).

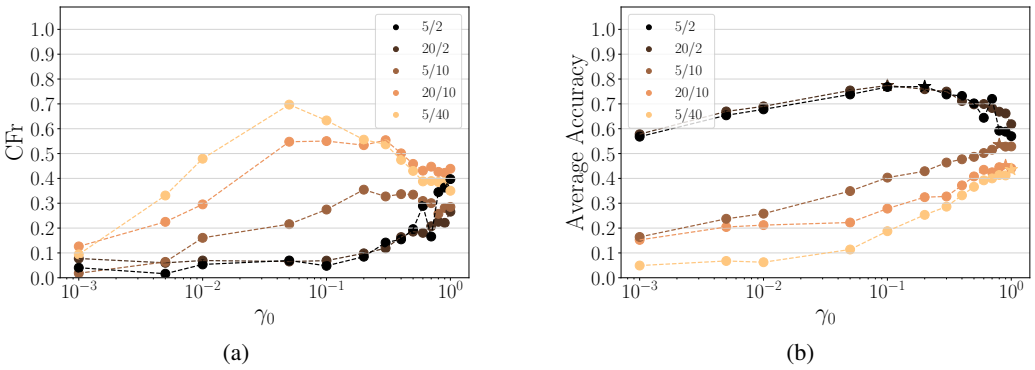


Figure 6: (a) CFr and (b) average accuracy on Split-TinyImagenet for varying γ_0 and various combinations of tasks and classes per task. Notation: 5/10 corresponds to 5 tasks with 10 classes each.

If we were to consider the effect of width in the case of many classes per task, we observe that width has a positive impact on forgetting, especially at intermediate levels of feature learning (Fig. 8(a)). This is reflected also in the average accuracy (Fig. 8(b)). In a similar investigation as above, we can inspect the CKA similarity of features before and after training on that task, distinguishing the first task from the later tasks (Fig. 8(c)). We observe that increasing the width does not modify the level of evolution during the first task, but it does impact the evolution during later tasks, which is further reduced, and hence the CFr observed is improved by width. The benefit of width for pretrained models was also found by Ramasesh et al. [31], thus providing further evidence of the *pretraining*

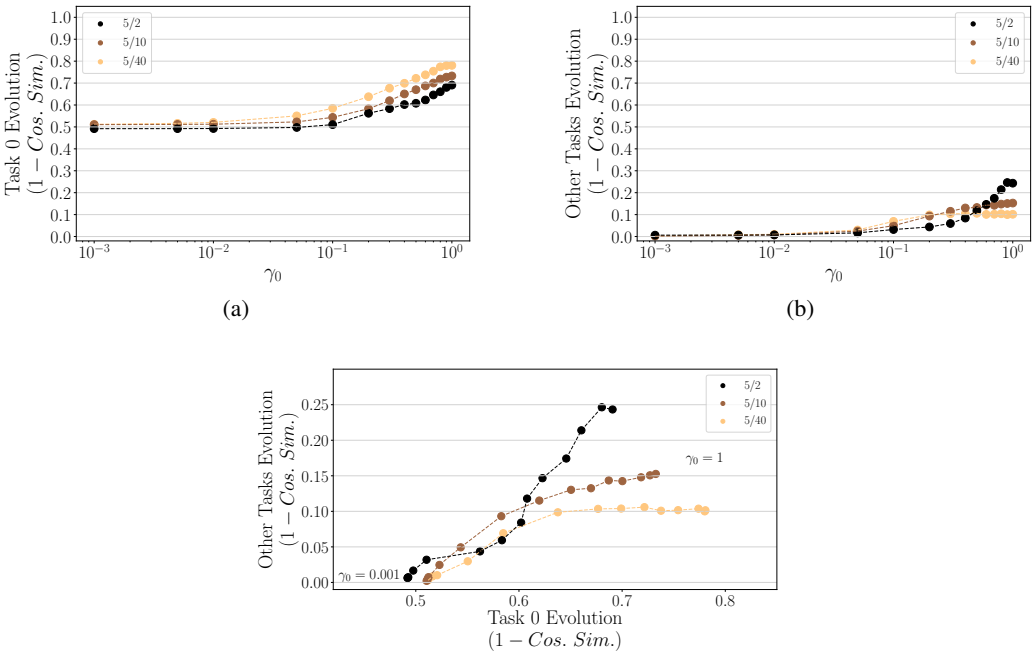


Figure 7: Investigations on the impact of the number of classes for Split-TinyImagenet with 5 tasks. (a) Features evolution (1− the cosine similarity of the features before and after training on that task) for the first task; (b) average features evolution for all other tasks; (c) features evolution for the first task vs. all other tasks.

effect discussed above. Indeed, we can further interpret the observations of [31], where width (or scale at large) helps only in the case of pretraining: without pretraining (or without the pretraining effect), the model is encouraged into the rich regime where width is not beneficial.

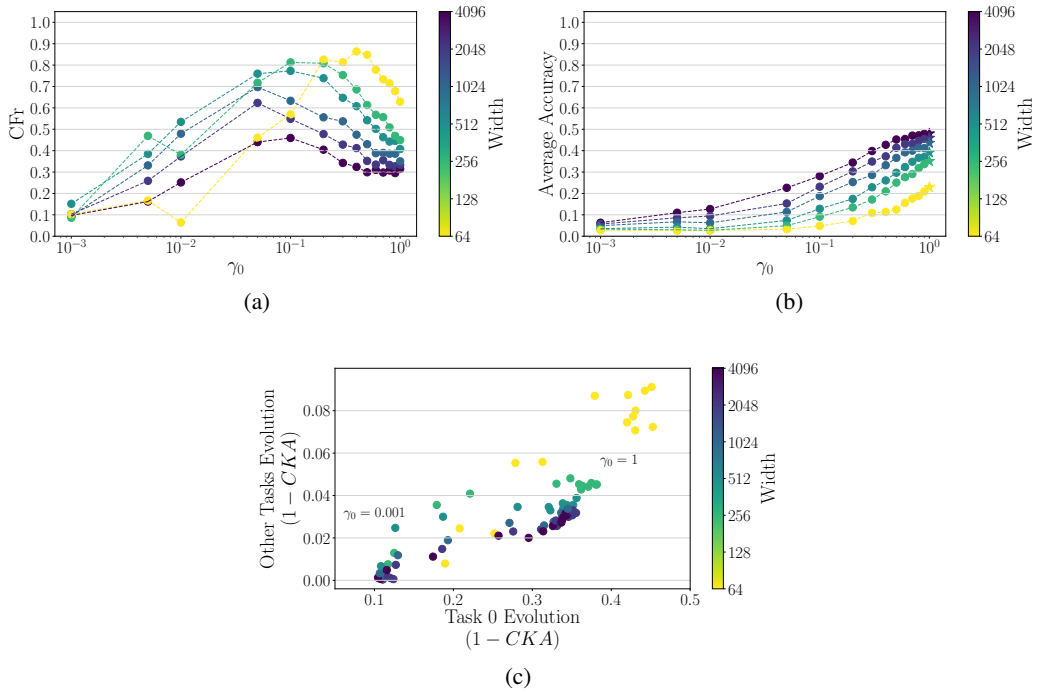


Figure 8: The role of width and γ_0 on Split-TinyImagenet with 5 tasks. (a) CFr; (b) average accuracy; (c) features evolution for the first task vs. all other tasks.