Beyond Cosine Decay: On the effectiveness of Infinite Learning Rate Schedule for Continual Pre-training

Vaibhav Singh [*1 2] **Paul Janson** [*1 2] **Paria Mehrbod** [1 2] **Adam Ibrahim** [3 2] **Irina Rish** [3 2] **Eugene Belilovsky** [1 2]
Benjamin Thérien [3 2]

Abstract

The growing availability of unlabeled data offers both opportunities and challenges for training AI systems. Self-supervised learning (SSL) has emerged as a powerful method for extracting representations from such data, but existing techniques struggle to adapt to non-stationary, non-IID real-world data without forgetting prior knowledge. While recent works use a cosine annealing schedule for continual pre-training, this approach causes forgetting during re-warming and hasn't been compared to other SSL methods. In this work, we compare the cosine schedule with the recently proposed infinite learning rate schedule and find the latter to be more effective. Our extensive evaluation across image and language datasets shows that the infinite learning rate schedule is a flexible and robust alternative, performing well without needing a fixed iteration budget. It demonstrates stable and effective performance in both small and large-scale pre-training setups, retaining knowledge and adapting across tasks.

1. Introduction

Self-supervised pre-training (Balestriero et al., 2023) has driven the development of foundational models in vision (Radford et al., 2021; Oquab et al., 2023; Kirillov et al., 2023; Shang et al., 2024) and language (Bommasani et al., 2021; Achiam et al., 2023; Touvron et al., 2023; Zhao et al., 2023), widely applied across various domains. These models, known for their large parameter counts and extensive training, exhibit impressive general-purpose capabilities.

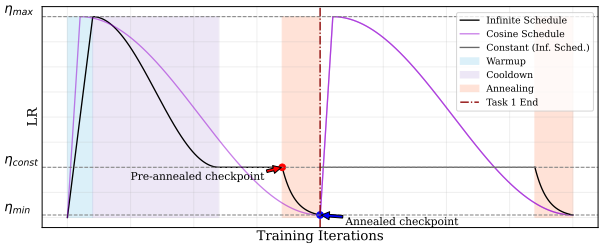


Figure 1. **Comparing Infinite Learning Rate Schedule with Repeated Cosine Annealing for Two-Task CL.** This comparison highlights the differences between the **infinite learning rate schedule** and the **cosine schedule**. The infinite schedule includes four phases: warmup, cooldown, constant, and annealing (see legend). The **vertical line** marks Task 1 completion. The infinite schedule offers two checkpointing options: the **pre-annealed checkpoint** at η_{const} and the **annealed checkpoint** at η_{min}, enabling flexibility for continual training. In contrast, the cosine schedule lacks the constant phase, limiting its adaptability for CL.

However, adapting foundation models to evolving data, such as new text (Soldaini et al., 2024; Li et al., 2024; Abadji et al., 2022; Kocetkov et al., 2022) and novel visual concepts (Prabhu et al., 2023; Seo et al., 2024), remains challenging due to high retraining costs and the risk of catastrophic forgetting (McCloskey & Cohen, 1989) caused by distributional shifts. While recent studies (Ke et al., 2023; Qiao & Mahdavi, 2024; Yıldız et al., 2024; Parmar et al., 2024) offer guidelines for continual pre-training in language modeling, seamless integration into existing pipelines is still lacking. In computer vision, traditional continual learning methods like regularization (Kirkpatrick et al., 2017; Li & Hoiem, 2017; Aljundi et al., 2018) and architectural modifications (Douillard et al., 2022; Yan et al., 2021) struggle to scale with modern foundation models due to limitations in generalizing to self-supervised learning objectives and large-scale datasets.

Most approaches for continually pre-training foundation models use a repeated cosine annealing schedule (Loshchilov & Hutter, 2017) with fixed duration (Gupta et al., 2023; Defazio et al., 2023; Ibrahim et al., 2024; Parmar et al., 2024; Guo et al., 2024). This schedule involves a warmup phase, followed by a cosine decay to the minimum value at the end of training (purple in Figure 1), assuming

[*]Equal contribution; authorship order among first authors was randomized. [1]Concordia University, Montréal, Canada [2]Mila – Quebec AI Institute, Montréal, Canada [3]DIRO, Université de Montréal, Montréal, Canada. Correspondence to: Paul Janson <paul.janson@mila.quebec>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

a terminal point. This limits future pre-training on new datasets and causes significant forgetting. Re-warming from the minimum value can also cause instability and exacerbate forgetting (Ibrahim et al., 2024). Recent works have explored more flexible *infinite learning rate* schedules (Zhai et al., 2022; Defazio et al., 2024; Hu et al., 2024; Shen et al., 2024; Hägele et al., 2024), which consist of four phases: warmup, decay (e.g., cosine, inverse square root), plateau, and rapid annealing (**black** line in Figure 1). These schedules, originally from data-scaling research, have been extended to continual learning (Garg et al., 2024; Ibrahim et al., 2024).

However, these works fail to address a key question: *How do these scheduling approaches perform under distribution shifts, i.e., non-IID data distributions?*[1] This is particularly relevant in practical applications where models must continuously adapt to data from diverse domains. For example, adapting an English model to incorporate German often leads to catastrophic forgetting, which severely impacts performance.

This work makes several key contributions:

- We present the first systematic study on learning rate schedules in non-IID self-supervised Continual Learning for both vision and language modalities.
- We show that infinite learning rate schedules, combined with experience replay, outperform several continual learning baselines in self-supervised settings (Sec C.1).
- We demonstrate that infinite learning rate schedules match or outperform repeated cosine annealing across large-scale sequential vision and language pre-training tasks, offering greater flexibility by not requiring a predefined dataset size (Sec 4.1, 4.2).
- We show that the **Infinite Cosine Schedule** is a robust alternative to repeated cosine decay, improving knowledge retention and adaptability in non-IID self-supervised learning scenarios across both vision and language tasks.

2. The Need for Infinite Learning Rate Scheduling: Why It Matters?

In this work, we investigate the effectiveness of the infinite cosine schedule (Ibrahim et al., 2024) compared to repeated cosine for the CPT of models under strong distribution shifts. We perform a comprehensive comparison across diverse self-supervised learning tasks in vision and language domains. Our extensive experiments show that infinite learning rate scheduling improves robustness to distribution shifts and outperforms cosine scheduling by removing the need to predefine training duration. We define Infinite Cosine Schedule as given in Ibrahim et al. (2024):

$$\text{Inf Cosine}(n) = \begin{cases} \frac{n}{N_w} \cdot \eta_{max}, & \text{if } n < N_w \\ \eta_{const} + \frac{\eta_{max} - \eta_{const}}{2} \cdot \left(1 + \cos\left(\pi \frac{n - N_w}{N_c - N_w}\right)\right), & \text{if } N_w < n \leq N_c \\ \eta_{const}, & \text{if } N_c < n \leq N_d \\ \eta_{const} \cdot \left(\frac{\eta_{min}}{\eta_{const}}\right)^{\frac{n - N_d}{t_a + N_d}} & \text{if } n > N_d \end{cases} \tag{1}$$

where n is the current training step, η_{max} and η_{min} denote the maximum and minimum learning rates respectively, and $N_w, N_c\ N_d$ denote number of warmup steps, cooldown steps, and decay steps respectively, each specifying the transition points between the phases. t_a denotes the amount of annealing steps required to achieve a converged checkpoint.

3. Experimental setup

Our experiments span vision and language domains focusing on significant distribution shifts across datasets $\mathcal{D}_0, \mathcal{D}_1, \ldots, \mathcal{D}_{N-1}$. We scale up to large-scale vision datasets with significant distribution shifts and demonstrate generalizability by continually pre-training LLMs across diverse distributions.

Continual pre-training of MAEs: We use Masked Autoencoders (MAE) (He et al., 2022) for vision pre-training, which masks image patches and reconstructs them using a Vision Transformer (ViT) (Dosovitskiy et al., 2020) encoder-decoder architecture. After pre-training, the encoder serves as a feature extractor for downstream tasks. Our vision pipeline uses three large-scale datasets ($N = 3$). ImageNet (Russakovsky et al., 2015) ($\mathcal{D}_0$) provides 1.28M object-centric images across 1,000 categories. Places2 (Zhou et al., 2017) ($\mathcal{D}_1$) introduces a distribution shift with 1M scene-understanding images. FireRisk (Shen et al., 2023) ($\mathcal{D}_2$) presents a substantial shift to remote sensing with 91K satellite images.

Evaluation: We measure task-specific performance and cross-task knowledge transfer using linear probing. After pre-training on each dataset $\mathcal{D}i$, we freeze the encoder and train linear classifiers $h\psi_i : \mathbb{R}^d \rightarrow \mathbb{R}^{c_i}$ optimized with cross-entropy loss.

Implementation: We use ViT-B/16 backbone with constant learning rate $\eta_{const} = 3.75e-5$ for infinite schedules. Experiments include replay buffers of size $B = 0.05 \times |\mathcal{D}_i|$ per task. All models train for 300 epochs per task using AdamW with batch size 4096. Details in Appendix E.4.

Continually pre-training LLMs: We use three datasets: DCLM-Baseline (Li et al., 2024) ($\mathcal{D}_0$) for natural language text, Stack (Kocetkov et al., 2022) ($\mathcal{D}_1$) for programming code, and German (Abadji et al., 2022) ($\mathcal{D}_2$) from OSCAR

[1] Some previous works on infinite LR schedules (Ibrahim et al., 2024; Garg et al., 2024) used datasets split from a single original dataset, leading to weaker shifts than those in this work.

Task Completed	Acc. ↑ With ER						Acc. ↑ Without ER					
	ImageNet		Places		FireRisk		ImageNet		Places		FireRisk	
	Cos	Inf	Cos	Inf	Cos	Inf	Cos	Inf	Cos	Inf	Cos	Inf
ImageNet (D_0)	60.34	59.73	30.56	30.61	60.05	60.37	60.34	59.73	30.56	30.61	60.05	60.37
Places (D_1)	58.89	**61.09**	32.35	32.03	60.28	59.68	49.97	**50.77**	32.26	31.95	60.13	60.58
FireRisk (D_2)	54.35	**57.50**	31.12	**31.53**	61.13	61.50	33.39	**36.38**	23.40	**25.19**	62.30	62.11
Metric	**Avg. Acc. ↑**		**FWT ↑**		**BWT ↑**		**Avg. Acc ↑**		**FWT ↑**		**BWT ↑**	
Values	48.87	**50.18**	**15.51**	15.23	-3.61	**-1.37**	39.69	**41.22**	15.43	**15.68**	-17.91	**-15.06**

Table 1. Performance comparison between cosine (Cos) and infinite cosine (Inf) for MAE pre-training across different tasks, with and without a replay buffer. Grey values indicate performance on datasets that were *unseen* during training at that stage. Each row shows model performance after the model has completed training on the task specified in the row label. The infinite schedule generally preserves knowledge better, particularly in the presence of multiple distribution shifts. Note that this is shown by the superior knowledge retention (bolded) on the previous tasks after learning new tasks. The bottom section presents key averaged metrics across all three tasks: Average Accuracy (Avg. Acc.), Forward Transfer (FWT), and Backward Transfer (BWT), (where ↑ indicates that higher is better). Infinite cosine achieves better overall results, especially in reducing forgetting (as shown by less negative BWT values).

corpus for multilingual content. Each contains 100B tokens, creating realistic distribution shifts representative of current CPT applications.

Implementation: We utilize 570M parameter LLaMA-3 architecture with $\eta_{max} = 3e-4$, $\eta_{min} = 3e-5$, varying $\eta_{const} \in [1e-4, 2e-4]$ and cooldown proportions. Batch size is 1024 with sequence length 2048. Complete details in Appendix G

4. Results

4.1. Results for pre-training mae on multiple datasets

We present the results of our experiments on large scale MAE pre-training in Table 1 (left). The infinite schedule achieves accuracy comparable to a cosine schedule after ImageNet ($\mathcal{D}_0$) pre-training. The effectiveness becomes more pronounced after continual training on Places2 ($\mathcal{D}_1$) with a replay buffer (ER), where the infinite schedule outperforms the cosine schedule on the previous task while achieving better performance on the current dataset. Even under the strong distribution shift introduced by Firerisk ($\mathcal{D}_2$), the infinite cosine schedule proves remarkably robust, achieving **57.50%** accuracy on ImageNet. After completing all three tasks, the infinite schedule achieves an average accuracy of **50.18%** across all datasets, $\approx$ **1.3%** higher than the cosine schedule. The Forward Transfer (FWT) metrics are comparable between the two schedules, while the infinite schedule shows better resistance to catastrophic forgetting with a higher Backward Transfer (BWT). We perform extended forgetting analysis in Appendix L.

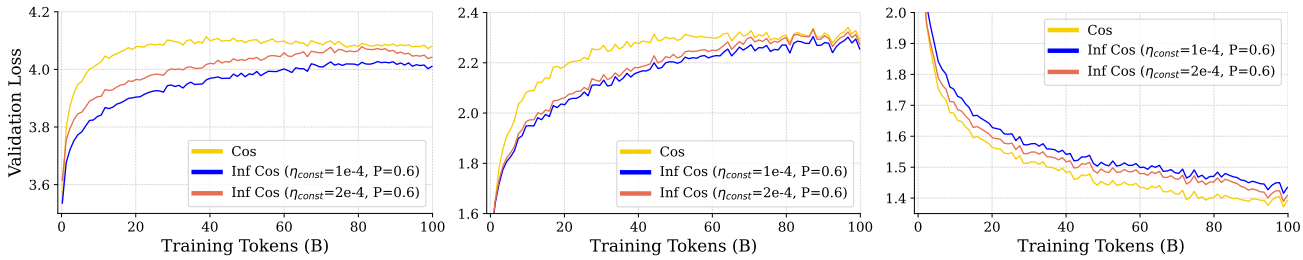
Similarly, when evaluating infinite schedule without experience replay in Table 1 (right), we observe that it maintains its competitive performance even though there is a significant forgetting. After initial pre-training on ImageNet, it shows comparable performance to the cosine schedule. After pre-training on Places2, infinite schedule demonstrates higher accuracy on the previous task i.e ImageNet. Similar to replay experiment, this is more visible after the third distribution shift where the infinite schedule maintains ImageNet accuracy at **36.38%**, outperforming the cosine schedule's **33.39%**. This improvement is particularly significant given the challenging nature of continual learning without a replay buffer. In the overall metrics, the infinite schedule achieves a higher average accuracy and Forward Transfer (FWT). Importantly, even without replay, infinite schedule demonstrates better resistance to catastrophic forgetting, with a high Backward Transfer (BWT).

4.2. Results for Continual Pre-training LLMs

We begin pre-training on the DCLM dataset and observe that even in this pre-training phase, rapid annealing in the case of infinite schedule yields a lower validation loss compared to the cosine schedule, offering a competitive advantage. This trend is evident in Figure 7, with further details provided in Appendix H and performance on LLM benchmarks on Appendix M. After pre-training on DCLM, we continue training on the Stack dataset. Figure 3 shows the validation loss on the DCLM ($\mathcal{D}_0$) and Stack ($\mathcal{D}_1$) dataset for cosine and infinite schedule with varying η_{const} and P. We observe that all the configurations of infinite schedule helps in mitigating catastrophic forgetting with a lower validation loss on DCLM data, as compared to cosine, with a minimum validation loss for $\eta_{const} = 1e-4$ and longer cooldown of $P = 0.6$. This is in concurrence with the observations for MAE large scale pre-training.

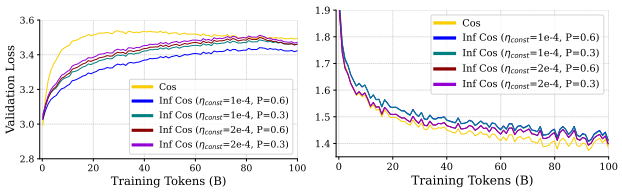
However, we observe that the infinite schedule exhibits slightly lower adaptability to the current task (Stack) compared to the cosine schedule. Specifically, the infinite schedule ($\eta_{const} = 1e-4$, $P = 0.6$), which minimizes forgetting, shows a marginally higher validation loss on Stack. However, with a higher $\eta_{const} = 2e-4$, the infinite schedule achieves performance comparable to cosine on the current task while maintaining a lower validation loss on the upstream task.

To alleviate forgetting, we further introduce a replay mechanism where we sample 50% of the data from the previous task (DCLM) and 50% from the current task (Stack). Figure Figure 4 shows the validation loss on the DCLM ($\mathcal{D}_0$) and Stack ($\mathcal{D}_1$) dataset for cosine and infinite schedule with varying η_{const} and P with replay. We observe that the infinite schedule with $\eta_{const} = 2e-4$ and longer cooldown of $P = 0.6$ helps in mitigating catastrophic forgetting with minimum validation loss, as compared to cosine and other configurations of infinite scheduling. We further



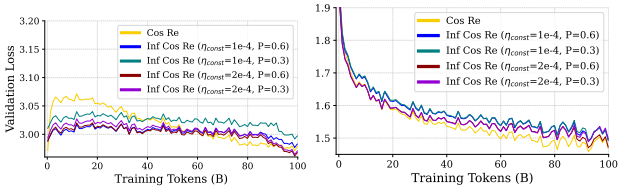
(a) Valid. Loss on DCLM (b) Valid. Loss on Stack (c) Valid. Loss on German

Figure 2. Validation Loss (↓ is better) for different schedules. CPT is on German data ($\mathcal{D}_2$), validating on all German ($\mathcal{D}_2$) DCLM ($\mathcal{D}_0$) and Stack ($\mathcal{D}_1$) datasets. Infinite schedules (both $\eta_{const} \in \{1e-4, 2e-4\}$) gives a lower validation loss on previous tasks as compared to cosine. The downstream performance of infinite schedule on the current task (German) is comparable to cosine.



(a) Valid. Loss on DCLM (b) Valid. Loss on Stack

Figure 3. Validation Loss (↓ is better) for different schedules. CPT is on Stack data ($\mathcal{D}_1$), validating on both DCLM ($\mathcal{D}_0$) and Stack ($\mathcal{D}_1$) datasets. All the configurations of infinite schedules mitigate catastrophic forgetting with a lower validation loss on DCLM data, as compared to cosine. However, the downstream performance of infinite schedule on the current task(Stack) is slightly lower than cosine.



(a) Valid. Loss on DCLM (50% Replay) (b) Valid. Loss on Stack (50% Replay)

Figure 4. Validation Loss (↓ is better) for different schedules accompanied with replay. CPT is on Stack data ($\mathcal{D}_1$), validating on both DCLM ($\mathcal{D}_0$) and Stack ($\mathcal{D}_1$) datasets. Infinite schedule with $\eta_{const} = 2e-4$ and longer cooldown of $P = 0.6$ helps in mitigating catastrophic forgetting with minimum validation loss, as compared to cosine and other configurations of infinite scheduling. Even on the current task (Stack), $\eta_{const} = 2e-4$ and $P = 0.6$ yield a validation loss closely matching that of the cosine schedule.

observe that infinite schedule, irrespective of the P and η_{const} gives a lower validation loss as compared to cosine. A higher η_{const} likely enhances adaptability to the current task, while a lower η_{const} minimizes forgetting on previous tasks. Since replay mitigates forgetting, a higher η_{const} ultimately achieves the best overall performance, balancing adaptability and retention. Furthermore, we present experiments in Appendix J that demonstrate the flexibility and agility of the infinite cosine schedule in preserving knowledge from previous tasks.

To further strengthen our evaluation, we introduce a language shift by continually pre-training on the German dataset (German language). This transition imposes a more pronounced distributional shift, as the model moves from programming language data (Stack) to natural language. As in previous sections, we measure validation loss across all datasets while continually pre-training on German as shown in Figure 2 Given our earlier findings that short cooldown proportions are detrimental, we train models only with $P = 0.6$ under an infinite schedule. Consistent with our previous observations (Figure 3), we find that the infinite schedule with $\eta_{const} = 1e-4$ and $P = 0.6$ yields the best performance in mitigating forgetting. Consequently, we say that optimal constant learning rate should be selected through careful hyperparameter tuning.

5. Conclusion

Our results suggest that infinite cosine schedules offer a flexible and robust framework for continual pre-training (CPT) of foundation models across vision and language domains. They enable seamless training continuation from intermediate checkpoints, support dynamic adaptation strategies such as adjustable replay, and maintain strong performance under distribution shifts without requiring a predefined training budget. On large-scale experiments across multiple vision and language datasets, infinite schedules consistently outperform repeated cosine decay, both alone and with replay mechanisms. While we do not claim universal superiority, our experiments demonstrate that infinite schedules provide competitive retention of prior knowledge and improved stability in non-IID continual learning scenarios, making them a practical alternative to repeated cosine decay in real-world CPT pipelines. Our exploration opens promising avenues for future research, including theoretical analysis of infinite schedules, comparing different cooldown functions across modalities, and extending studies to wider architectures and self-supervised learning frameworks.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Abadji, J., Ortiz Suarez, P., Romary, L., and Sagot, B. Towards a cleaner document-oriented multilingual crawled corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pp. 4344–4355, Marseille, France, June 2022. European Language Resources Association. URL https://aclanthology.org/2022.lrec-1.463.

Achiam, O. J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., and et al., S. A. Gpt-4 technical report. 2023. URL https://api.semanticscholar.org/CorpusID:257532815.

Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., and Tuytelaars, T. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 139–154, 2018.

Balestriero, R., Ibrahim, M., Sobal, V., Morcos, A., Shekhar, S., Goldstein, T., Bordes, F., Bardes, A., Mialon, G., and Tian, Y. Avi schwarzschild, andrew gordon wilson, jonas geiping, quentin garrido, pierre fernandez, amir bar, hamed pirsiavash, yann lecun, and micah goldblum. *A cookbook of self-supervised learning*, 6, 2023.

Bao, H., Dong, L., Piao, S., and Wei, F. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Chen, T., Xu, B., Zhang, C., and Guestrin, C. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016.

Cossu, A., Tuytelaars, T., Carta, A., Passaro, L., Lomonaco, V., and Bacciu, D. Continual pre-training mitigates forgetting in language and vision, 2022. URL https://arxiv.org/abs/2205.09357.

DeepSeek-AI, Zhu, Q., Guo, D., Shao, Z., Yang, D., Wang, P., Xu, R., Wu, Y., Li, Y., Gao, H., Ma, S., Zeng, W., Bi, X., Gu, Z., Xu, H., Dai, D., Dong, K., Zhang, L., Piao, Y., Gou, Z., Xie, Z., Hao, Z., Wang, B., Song, J., Chen, D., Xie, X., Guan, K., You, Y., Liu, A., Du, Q., Gao, W., Lu, X., Chen, Q., Wang, Y., Deng, C., Li, J., Zhao, C., Ruan, C., Luo, F., and Liang, W. Deepseek-coder-v2: Breaking the barrier of closed-source models in code intelligence. *CoRR*, abs/2406.11931, 2024. URL https://arxiv.org/abs/2406.11931.

Defazio, A., Cutkosky, A., Mehta, H., and Mishchenko, K. Optimal linear decay learning rate schedules and further refinements. *arXiv preprint arXiv:2310.07831*, 2023.

Defazio, A., Yang, X. A., Khaled, A., Mishchenko, K., Mehta, H., and Cutkosky, A. The road less scheduled. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=0XeNkkENuI.

Díaz-Rodríguez, N., Lomonaco, V., Filliat, D., and Maltoni, D. Don't forget, there is more than forgetting: new metrics for continual learning. *arXiv preprint arXiv:1810.13166*, 2018.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Douillard, A., Ramé, A., Couairon, G., and Cord, M. DyTox: Transformers for continual learning with dynamic token expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9281–9291, 2022. doi: 10.1109/CVPR52688.2022.00906. URL https://doi.org/10.1109/CVPR52688.2022.00906.

Fang, Y., Wang, W., Xie, B., Sun, Q., Wu, L., Wang, X., Huang, T., Wang, X., and Cao, Y. Eva: Exploring the limits of masked visual representation learning at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19358–19369, 2023.

Garg, S., Farajtabar, M., Pouransari, H., Vemulapalli, R., Mehta, S., Tuzel, O., Shankar, V., and Faghri, F. TiC-CLIP: Continual training of CLIP models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2310.16226.

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. Accurate, large minibatch sgd: Training imagenet in 1 hour, 2018.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., and et. al, A. M. The llama

3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Guo, Y., Fu, J., Zhang, H., Zhao, D., and Shen, Y. Efficient continual pre-training by mitigating the stability gap, 2024. URL https://arxiv.org/abs/2406.14833.

Gupta, K., Thérien, B., Ibrahim, A., Richter, M. L., Anthony, Q., Belilovsky, E., Rish, I., and Lesort, T. Continual pre-training of large language models: How to (re) warm your model? *arXiv preprint arXiv:2308.04014*, 2023.

Hägele, A., Bakouch, E., Kosson, A., allal, L. B., Werra, L. V., and Jaggi, M. Scaling laws and compute-optimal training beyond fixed training durations. In *Workshop on Efficient Systems for Foundation Models II @ ICML2024*, 2024. URL https://openreview.net/forum?id=ompl7supoX.

He, K., Chen, X., Xie, S., Li, Y., Dollar, P., and Girshick, R. Masked Autoencoders Are Scalable Vision Learners. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15979–15988, New Orleans, LA, USA, June 2022. IEEE. ISBN 978-1-66546-946-3. doi: 10.1109/CVPR52688.2022.01553. URL https://ieeexplore.ieee.org/document/9879206/.

Hu, S., Tu, Y., Han, X., He, C., Cui, G., Long, X., Zheng, Z., Fang, Y., Huang, Y., Zhao, W., et al. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024.

Ibrahim, A., Thérien, B., Gupta, K., Richter, M. L., Anthony, Q. G., Belilovsky, E., Lesort, T., and Rish, I. Simple and scalable strategies to continually pre-train large language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=DimPeeCxKO.

Ke, Z., Shao, Y., Lin, H., Konishi, T., Kim, G., and Liu, B. Continual pre-training of language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=m_GDIItaI3o.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., and Girshick, R. Segment anything. *arXiv:2304.02643*, 2023.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. Overcoming catastrophic forgetting in neural networks. *Proc. of the national academy of sciences*, 2017. URL https://www.pnas.org/content/pnas/114/13/3521.full.pdf.

Kocetkov, D., Li, R., Ben Allal, L., Li, J., Mou, C., Muñoz Ferrandis, C., Jernite, Y., Mitchell, M., Hughes, S., Wolf, T., Bahdanau, D., von Werra, L., and de Vries, H. The stack: 3 tb of permissively licensed source code. *Preprint*, 2022.

Krizhevsky, A. et al. Learning multiple layers of features from tiny images. 2009.

Li, J., Fang, A., Ansari, H. P., Faghri, F., Ali, A. M. E., Toshev, A., Shankar, V., Smyrnis, G., Jordan, M., Igvi, M., Dimakis, A., Zhang, H., Bansal, H., Vasiljevic, I., Mercat, J., Jitsev, J., Arora, K., Chen, M., Muenninghoff, N., Soldaini, L., Koh, P. W., Heckel, R., Xin, R., Gadre, S., Shao, R., Pratt, S., Garg, S., Keh, S., Gururangan, S., Sanyal, S., Bitton, Y., Kollar, T., Wortsman, M., Guha, E., Abbas, A., Hsieh, C.-Y., Ghosh, D., Ilharco, G., Daras, G., Marathe, K., Gardner, J., Nezhurina, M., Dave, A., Carmon, Y., and Schmidt, L. Datacomp-lm: In search of the next generation of training sets for language models, 2024. URL https://arxiv.org/abs/2406.11794.

Li, Z. and Hoiem, D. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. URL https://arxiv.org/abs/1606.09282.

Lopez-Paz, D. and Ranzato, M. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

Loshchilov, I. and Hutter, F. SGDR: stochastic gradient descent with warm restarts. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Skq89Scxx.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

McCloskey, M. and Cohen, N. J. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pp. 109–165. Elsevier, 1989.

Mehta, S. V., Patil, D., Chandar, S., and Strubell, E. An empirical investigation of the role of pre-training in lifelong learning. *J. Mach. Learn. Res.*, 24:214:1–214:50, 2023. URL http://jmlr.org/papers/v24/22-0496.html.

Mirzadeh, S., Chaudhry, A., Yin, D., Hu, H., Pascanu, R., Görür, D., and Farajtabar, M. Wide neural networks forget less catastrophically. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., and Sabato, S. (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 15699–15717. PMLR, 2022. URL https://proceedings.mlr.press/v162/mirzadeh22a.html.

Oquab, M., Darcet, T., Moutakanni, T., Vo, H. V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Howes, R., Huang, P.-Y., Xu, H., Sharma, V., Li, S.-W., Galuba, W., Rabbat, M., Assran, M., Ballas, N., Synnaeve, G., Misra, I., Jegou, H., Mairal, J., Labatut, P., Joulin, A., and Bojanowski, P. Dinov2: Learning robust visual features without supervision, 2023.

Parmar, J., Satheesh, S., Patwary, M., Shoeybi, M., and Catanzaro, B. Reuse, don't retrain: A recipe for continued pretraining of language models. *CoRR*, abs/2407.07263, 2024. URL https://doi.org/10.48550/arXiv.2407.07263.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Prabhu, A., Torr, P. H., and Dokania, P. K. Gdumb: A simple approach that questions our progress in continual learning. In *European Conference on Computer Vision*, pp. 524–540. Springer, 2020.

Prabhu, A., Hammoud, H. A. A. K., Lim, S.-N., Ghanem, B., Torr, P. H., and Bibi, A. From categories to classifiers: Name-only continual learning by exploring the web. *arXiv preprint arXiv:2311.11293*, 2023.

Qiao, F. and Mahdavi, M. Learn more, but bother less: parameter efficient continual learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=ZxtaNh5UYB.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021. URL http://proceedings.mlr.press/v139/radford21a.html.

Rajbhandari, S., Rasley, J., Ruwase, O., and He, Y. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–16. IEEE, 2020.

Ramasesh, V. V., Lewkowycz, A., and Dyer, E. Effect of scale on catastrophic forgetting in neural networks. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=GhVS8_yPeEa.

Rolnick, D., Ahuja, A., Schwarz, J., Lillicrap, T., and Wayne, G. Experience replay for continual learning. *Advances in neural information processing systems*, 32, 2019.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge, January 2015. URL http://arxiv.org/abs/1409.0575. arXiv:1409.0575 [cs].

Scialom, T., Chakrabarty, T., and Muresan, S. Fine-tuned Language Models are Continual Learners, October 2022. URL http://arxiv.org/abs/2205.12393. arXiv:2205.12393 [cs].

Seo, M., Cho, S., Lee, M., Misra, D., Choi, H., Kim, S. J., and Choi, J. Just say the name: Online continual learning with category names only via data generation. *arXiv preprint arXiv:2403.10853*, 2024.

Shang, J., Schmeckpeper, K., May, B. B., Minniti, M. V., Kelestemur, T., Watkins, D., and Herlant, L. Theia: Distilling diverse vision foundation models for robot learning. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=ylZHvlwUcI.

Shen, S., Seneviratne, S., Wanyan, X., and Kirley, M. Firerisk: A remote sensing dataset for fire risk assessment with benchmarks using supervised and self-supervised learning. In *2023 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pp. 189–196. IEEE, 2023.

Shen, Y., Stallone, M., Mishra, M., Zhang, G., Tan, S., Prasad, A., Soria, A. M., Cox, D. D., and Panda, R. Power scheduler: A batch size and token number agnostic learning rate scheduler. *arXiv preprint arXiv:2408.13359*, 2024.

Singh, M., Duval, Q., Alwala, K. V., Fan, H., Aggarwal, V., Adcock, A., Joulin, A., Dollár, P., Feichtenhofer,

C., Girshick, R., Girdhar, R., and Misra, I. The effectiveness of MAE pre-pretraining for billion-scale pretraining. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5484–5494, 2023. doi: 10.1109/ICCV51070.2023.00505. URL https://openaccess.thecvf.com/content/ICCV2023/html/Singh_The_Effectiveness_of_MAE_Pre-Pretraining_for_Billion-Scale_Pretraining_ICCV_2023_paper.html.

Singh, V., Aljundi, R., and Belilovsky, E. Controlling forgetting with test-time data in continual learning. *arXiv preprint arXiv:2406.13653*, 2024.

Smith, J. S., Karlinsky, L., Gutta, V., Cascante-Bonilla, P., Kim, D., Arbelle, A., Panda, R., Feris, R., and Kira, Z. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11909–11919, 2023.

Smith, S. L., Kindermans, P., Ying, C., and Le, Q. V. Don't decay the learning rate, increase the batch size. In *6th International Conference on Learning Representations, ICLR 2018*, 2018. URL https://openreview.net/forum?id=B1Yy1BxCZ.

Soldaini, L., Kinney, R., Bhagia, A., Schwenk, D., Atkinson, D., Authur, R., Bogin, B., Chandu, K., Dumas, J., Elazar, Y., Hofmann, V., Jha, A. H., Kumar, S., Lucy, L., Lyu, X., Lambert, N., Magnusson, I., Morrison, J., Muennighoff, N., Naik, A., Nam, C., Peters, M. E., Ravichander, A., Richardson, K., Shen, Z., Strubell, E., Subramani, N., Tafjord, O., Walsh, P., Zettlemoyer, L., Smith, N. A., Hajishirzi, H., Beltagy, I., Groeneveld, D., Dodge, J., and Lo, K. Dolma: an open corpus of three trillion tokens for language model pretraining research. *CoRR*, abs/2402.00159, 2024. URL https://doi.org/10.48550/arXiv.2402.00159.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. URL https://arxiv.org/abs/2302.13971.

Wang, Z., Liu, L., Duan, Y., Kong, Y., and Tao, D. Continual learning with lifelong vision transformer. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 171–181, 2022a. doi: 10.1109/CVPR52688.2022.00027.

Wang, Z., Zhang, Z., Ebrahimi, S., Sun, R., Zhang, H., Lee, C.-Y., Ren, X., Su, G., Perot, V., Dy, J., et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision*, pp. 631–648. Springer, 2022b.

Wang, Z., Zhang, Z., Lee, C.-Y., Zhang, H., Sun, R., Ren, X., Su, G., Perot, V., Dy, J., and Pfister, T. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 139–149, 2022c.

Winata, G. I., Xie, L., Radhakrishnan, K., Wu, S., Jin, X., Cheng, P., Kulkarni, M., and Preotiuc-Pietro, D. Overcoming catastrophic forgetting in massively multilingual continual learning. *arXiv preprint arXiv:2305.16252*, 2023.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. Transformers: State-of-the-Art Natural Language Processing. pp. 38–45. Association for Computational Linguistics, October 2020. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Yan, S., Xie, J., and He, X. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3014–3023, 2021.

Yang, G., Hu, E., Babuschkin, I., Sidor, S., Liu, X., Farhi, D., Ryder, N., Pachocki, J., Chen, W., and Gao, J. Tuning large neural networks via zero-shot hyperparameter transfer. *Advances in Neural Information Processing Systems*, 34:17084–17097, 2021.

Ye, F. and Bors, A. G. Task-free dynamic sparse vision transformer for continual learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 16442–16450, 2024.

Yıldız, Ç., Ravichandran, N. K., Punia, P., Bethge, M., and Ermis, B. Investigating continual pretraining in large language models: Insights and implications. *arXiv preprint arXiv:2402.17400*, 2024.

Zhai, X., Kolesnikov, A., Houlsby, N., and Beyer, L. Scaling Vision Transformers, June 2022. URL http://arxiv.org/abs/2106.04560. arXiv:2106.04560 [cs].

Zhang, W., Janson, P., Aljundi, R., and Elhoseiny, M. Overcoming generic knowledge loss with selective parameter update. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24046–24056, 2024.

Zhang, X. Github, 2021. URL https://github.com/IcarusWizard/MAE.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023. URL https://arxiv.org/abs/2303.18223.

Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., and Torralba, A. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

A. Related Work

Continual Pre-training (CPT) of Vision Foundation Models Continually pre-training Vision Transformers (ViTs) (Dosovitskiy et al., 2020; Bao et al., 2021) adapts them to sequential data while mitigating catastrophic forgetting. Wang et al. (2022a) introduced the Lifelong Vision Transformer (LVT), using inter-task attention to preserve critical weights. Ye & Bors (2024) proposed a task-free dynamic sparse ViT. The rise of large-scale foundation models, particularly Vision-Language Models (VLMs) (Radford et al., 2021; Garg et al., 2024; Zhang et al., 2024; Singh et al., 2024), has reshaped CL, where CPT provides an efficient alternative to full retraining. Unlike parameter-efficient methods (Wang et al., 2022c;b; Smith et al., 2023), our work adapts the entire model.

Continual Pre-training (CPT) of Large Language Models (LLMs) Recent studies (Scialom et al., 2022; Winata et al., 2023; Mehta et al., 2023; Gupta et al., 2023) show that CPT enables LLMs to learn general representations for various tasks. Cossu et al. (2022) demonstrated that CPT mitigates catastrophic forgetting, with self-supervised approaches outperforming supervised ones. Larger pretrained models exhibit less forgetting due to increasingly orthogonal class representations (Ramasesh et al., 2022; Mirzadeh et al., 2022). Additionally, Scialom et al. (2022) found that self-supervised pre-training naturally enables CL.

Alternatives to Cosine Schedule The cosine decay schedule (Loshchilov & Hutter, 2017) is common in vision tasks, where cyclic learning rates help avoid suboptimal minima (Smith et al., 2018). For language models, single-cycle cosine annealing is standard (Gupta et al., 2023; Parmar et al., 2024), but its fixed step count limits continuous training. To address this, the Warmup-Stable-Decay (WSD) scheduler (Hu et al., 2024) enables continuous training. Shen et al. (2024) refined this with the power scheduler, using exponential decay based on token count, while Hägele et al. (2024) proposed constant learning rates with cooldowns, addressing data scaling but not distribution shifts.

B. Discussion

In our large-scale experiments, we have explored different hyperparameters of the infinite cosine schedule across both vision and language tasks. In the case without replay, the choice of η_{const} follows a similar pattern across both modalities, where a lower η_{const} yields optimal performance. However, with replay, an apparent discrepancy emerges: vision tasks still favor a lower η_{const}, while language tasks seem to benefit from a higher η_{const}. In vision tasks, the variation between high and low η_{const} spans an order of magnitude (i.e., a factor of 10x), whereas in language tasks, the difference is narrower. This suggests that the relative comparison of η_{const} across modalities is not directly meaningful, as the scales of sensitivity differ between vision and language models.

C. MAE pretraining on CIFAR10

To validate our hypothesis on infinite learning rate schedules, we conduct an experiment with a small-scale MAE CPT on CIFAR-10 (Krizhevsky et al., 2009), using a controlled setting for rigorous baseline evaluation. The dataset is divided into five sequential tasks, each with two classes introduced in label order (0-9). We employ a ViT-tiny (Dosovitskiy et al., 2020) to match the scale of CIFAR-10, with our implementation based on Zhang (2021). We use a lightweight decoder with learned positional embeddings to reconstruct the masked patches. We train for 400 epochs with a batch size of 512. Hyperparameters for this small scale experiment are provided in Appendix E.2.

Baselines and Adaptations: We compare our approach with the following CL baselines, adapting them for self-supervised pre-training: **Sequential Fine-tuning**: trains sequentially without mitigating forgetting, serving as the primary baseline. **Experience Replay (ER)** (Rolnick et al., 2019): maintains a memory buffer with $\{40\%, 50\%\}$ samples of prior tasks, sampled uniformly. Each batch contains equal proportion of current task data and randomly sampled data from replay buffer. **Memory Aware Synapses (MAS)** (Aljundi et al., 2018): estimates parameter importance by measuring how changes affect the model output, then penalizes updates to important weights. We adapted it for self-supervised learning by computing importance of weights from the L2 norm of the encoder's output, with a regularization $\lambda = 0.75$. **Learning without Forgetting (LwF)** (Li & Hoiem, 2017): preserves knowledge by distilling responses from the previous model version. We modified it for self-supervised learning with feature distillation on the encoder's output, weighted by $\alpha = 0.75$. **GDumb** (Prabhu et al., 2020): Uses stratified sampling to maintain a balanced buffer. The model resets to random initialization for each new task and trains from scratch on buffer data. For evaluation, we use standard CL metrics from Lopez-Paz & Ranzato (2017): Average Accuracy (Acc), Forward Transfer (FWT), and Backward Transfer (BWT), defined in Appendix E.3.

C.1. Results for pre-training MAE on CIFAR10

Replay	FT-seq		MAS		LwF		ER		GDumb		Ours (Inf Cos)	
	Acc ↑	BWT ↑	Acc ↑	BWT ↑	Acc ↑	BWT ↑	Acc ↑	BWT ↑	Acc ↑	BWT ↑	Acc ↑	BWT ↑
0%	58.16	-17.65	50.44	-19.11	50.52	-19.78	-	-	-	-	**60.03**	**-12.61**
40%	-	-	50.36	-18.90	-	-	53.98	-21.55	48.76	-19.51	**61.45**	**-12.76**
50%	-	-	50.91	-18.37	-	-	57.94	-18.53	48.46	-18.76	**62.16**	**-12.61**

Table 2. Average linear probe accuracy (Acc) and Backward Transfer (BWT) (where ↑ indicates that higher is better) for comparing CL baselines utilizing cosine schedule with Infinite Schedule on CIFAR10 with varying replay (ER) strategies. It can be observed that the infinite schedule (Inf Cos) consistently achieves superior performance compared to the cosine schedule across all experimental configurations.

Table 2 demonstrates that the infinite cosine schedule outperforms the standard cosine, achieving higher average linear probe accuracy and BWT across all tasks in small-scale CPT on CIFAR-10. Specifically, in CPT without experience replay (ER), it improves average accuracy by **1.87%** and BWT by approximately **4%** over Finetuning (FT-seq) with a repeated cosine schedule.

Interestingly, in this setup, the combination of the repeated cosine schedule and experience replay (ER) degrades model performance, as seen in the comparison between FT-seq and ER with 40% replay. This decline likely stems from limited data diversity in small datasets, leading the more aggressive re-warming of the repeated cosine schedule to overfitting to the replay buffer. In contrast, the infinite learning rate schedule eliminates rewarming, effectively circumventing these issues. We would like to emphasize that the unexpectedly poor performance of these methods relative to FT-seq stems from a fundamental difference in the continual learning setting. While these baselines were originally designed for incremental supervised classification, our work centers on incremental masked image modeling (MIM) — a self-supervised task with distinct objectives and evaluation protocols. To the best of our knowledge, this is the first systematic evaluation of continual learning baselines in a self-supervised MIM setting, leaving us without alternatives specifically designed for this novel paradigm. Therefore, the results should be interpreted with this context in mind.

While replay behavior in small data scenarios is not our primary focus, it is worth noting that we used a relatively large replay buffer despite the dataset's limited size. The key finding is that the infinite cosine schedule, despite its simplicity, consistently outperforms baselines in both average accuracy and backward transfer (BWT). Notably, the strong performance gains with larger replay buffers suggest that our method scales effectively to large-scale pre-training, where the vast size of modern datasets provides sufficient replay samples to mitigate catastrophic forgetting, even at low buffer sampling rates.

D. Extended Experimental setup

Our experiments span both vision and language domains focusing on significant distribution shifts across a sequence of datasets $\mathcal{D}_0, \mathcal{D}_1, \ldots, \mathcal{D}_{N-1}$. We first evaluate infinite schedule on a small-scale MAE pre-training (He et al., 2022), comparing it to CL baselines (Sec **??**). Next, we scale up to large-scale vision datasets with significant distribution shifts (Sec D). Finally, we demonstrate its generalizability by continually pre-training LLMs across diverse distributions (Sec D.1).

Continual pre-training of MAEs: We use Masked Autoencoders (MAE) (He et al., 2022) for vision pre-training, leveraging their alignment with language models and strong performance in masked image modeling (Fang et al., 2023; Singh et al., 2023). As described by He et al. (2022), MAE pre-training masks a subset of image patches and reconstructs the original image using a Vision Transformer (ViT) (Dosovitskiy et al., 2020) encoder-decoder architecture. After pre-training, the decoder is discarded, and the encoder serves as a feature extractor for downstream vision tasks. Additional details regarding MAE pre-training are provided in Appendix E.1.

Datasets: Our pre-training pipeline utilizes three carefully selected large-scale datasets ($N = 3$). The CPT sequence begins with ImageNet (Russakovsky et al., 2015) ($\mathcal{D}_0$), having 1.28M object-centric images across 1,000 categories, providing a foundation in object recognition. Next, Places2 subset (Zhou et al., 2017) ($\mathcal{D}_1$) introduces a distribution shift with 1M scene-understanding images spanning 365 categories. Finally, FireRisk (Shen et al., 2023) ($\mathcal{D}_3$) presents a substantial shift to remote sensing with 91K satellite images for environmental monitoring. This progression increases distribution shifts, transitioning from object recognition to scene understanding followed by aerial imagery.

Evaluation: Our evaluation strategy measures both task-specific performance and cross-task knowledge transfer using linear probing. After pre-training on each dataset $\mathcal{D}_i$, we freeze the encoder f_θ as a fixed feature extractor and train a linear classifier $h_{\psi_i} : \mathbb{R}^d \to \mathbb{R}^{c_i}$ for each task, where c_i is the number of classes. The classifier is optimized with cross-entropy loss, and evaluated on task-specific validation sets using classification accuracy.

Implementation: We build on the PyTorch (Paszke et al., 2019) MAE framework with a ViT-B/16 backbone. For the infinite schedule, we keep a constant learning rate $\eta_{const} = 3.75e-5$, while the baseline follows a standard cosine decay schedule with SOTA hyperparameters (He et al., 2022). Experiments are conducted with and without a replay buffer of size $B = 0.05 \times |\mathcal{D}_i|$ per task. All models are trained for 300 epochs per task using AdamW (Loshchilov & Hutter, 2019) with a batch size of 4096. Further implementation and hyperparameter details are given in Appendix E.4.

D.1. Continually pre-training LLMs

Language Datasets: We consider three datasets for continually pre-training LLMs: DCLM-Baseline (Li et al., 2024) ($\mathcal{D}_0$), Stack (Kocetkov et al., 2022) ($\mathcal{D}_1$) and German (Abadji et al., 2022) ($\mathcal{D}_2$). DCLM is a large-scale dataset of natural language text, Stack is a specialized dataset of programming code snippets, and German is a subset of the multilingual OSCAR corpus (Abadji et al., 2022). The Stack and German datasets were chosen to represent strong, but realistic distribution shifts that are both representative of current CPT applications (DeepSeek-AI et al., 2024) and allow us to evaluate the model's ability to adapt to new tasks under challenging distribution shifts. We use the standard training splits for both datasets, treating each dataset as locally IID.

All the three datasets are tokenized through LLaMA-3 tokenizer (Grattafiori et al., 2024) owing to its large vocabulary size of 128K tokens (100K from *titktoken*[2] and 28K additional tokens for non-English languages). We sample a small subset of 100B tokens from each of the DCLM-Baseline (total = 3T), Stack (total = 744B), and OSCAR (total = 168B) datasets for our CPT experiments. We would like to emphasize that as the domain shifts farther away from the tokenizer's training corpus, the tokenizer might become the key bottleneck to performance. Such scenarios would be unrealistic without a way to adapt the tokenizer. With this in mind, we were careful to select challenging new domains that are still well represented in the tokenizer's vocabulary. Though we did not perform a formal tokenizer coverage analysis, our use of German alongside English datasets aligns with LLaMA-3's multilingual capabilities (Grattafiori et al., 2024). Stable validation loss across domains indicates no significant tokenizer-data mismatch in practice. We leave the treatment, continual tokenizer adaptation to future work.

Implementation details: We compare Infinite Cosine Schedule with the de-facto Cosine + Warmup Schedule. We fix $\eta_{max} = 3e-4$ and $\eta_{min} = 3e-5$ as described in (Ibrahim et al., 2024) for both schedules while varying the cooldown proportion ($N_{warmup} < n \leq N_{const}$) and the η_{const} for the infinite schedule. We utilize LLaMA-3 architecture (Grattafiori et al., 2024) with 570M parameters, training it as an autoregressive decoder-only transformer with a causal language modeling objective. We use a batch size of 1024 and sequence length 2048. Further details on hyperparameters are provided in the Appendix G.

E. Implementation details and hyperparameters for Vision pre-training

E.1. Formal Definition of MAE Pre-training

Formally the MAE pre-training procedure is described as follows: For each image $\mathbf{x} \in \mathcal{D}$, where $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$ and $\mathbf{x}_i \sim$ IID, we first partition it into a sequence of non-overlapping patches $\{\mathbf{p}_i\}_{i=1}^N$. We use the same masking ratio from the original MAE (He et al., 2022) that randomly masks 75% of these patches, creating two complementary sets: visible patches $\mathcal{V}$ and masked patches $\mathcal{M}$. An encoder $f_\theta(\cdot)$, implemented as a Vision Transformer (Dosovitskiy et al., 2020), processes only the visible patches to obtain latent representations $\mathbf{h}_v = f_\theta(\{\mathbf{p}_i\}_{i\in\mathcal{V}})$. These encoded features, along with mask tokens $\{\mathbf{m}_j\}_{j\in\mathcal{M}}$, are then fed to a decoder $g_\phi(\cdot)$ to reconstruct the original image: $\hat{\mathbf{x}} = g_\phi(\{\mathbf{h}_v\} \cup \{\mathbf{m}_j\})$. The entire framework is trained end-to-end by minimizing the mean squared error loss $\mathcal{L}_{mse} = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2$ between the original and reconstructed images. After pre-training, the decoder is discarded, and the encoder serves as a feature extractor for downstream vision tasks.

[2] https://github.com/openai/tiktoken/tree/main

E.2. Hyperparameters and Implementation details for CIFAR10 MAE

For our architecture, we employ a ViT-tiny encoder (12 layers, 192 hidden dimension, 3 attention heads) to match the scale of CIFAR-10, with our implementation based on the Zhang (2021)'s work. Our model uses a masking ratio of 0.75, consistent with the original MAE, and incorporates a lightweight decoder (4 layers) with learned position embeddings to reconstruct the masked patches. Regarding the learning rate configuration, we selected a maximum learning rate of 7.5e-5 through hyperparameter tuning over the values [7e-5, 1.5e-4, 3e-4] on the first two tasks, with a minimum learning rate of 7.5e-6. For most experiments, we employ a constant learning rate of 1.875e-5 and a cooldown proportion of 0.4, except for experiments without replay where we increase the constant learning rate to 5.625e-5. These optimal values were determined through experiments similar to our large-scale setup, testing cooldown proportions [0.3, 0.4, 0.5] and constant learning rates [1.875e-5, 5.625e-5]. While these findings align with our large-scale experiments, the small dataset size necessitated a slightly larger cooldown proportion to maintain a higher learning rate for a longer duration. For linear probing experiments in our small-scale setup, we utilized the AdamW(Loshchilov & Hutter, 2019) optimizer with a weight decay coefficient of $5e-3$ and momentum parameters β_1 and β_2 set to 0.9 and 0.95 respectively. The linear probing experiments implemented a cosine decay learning rate schedule with a maximum learning rate $\eta_{max} = 1e-3$, running for 100 epochs total, including 10 warmup epochs, with a batch size of 128. Complete hyperparameter details for pre-training and linear probing can be found in the corresponding tables Table 3 and Table 4. For the baseline methods MAS (Aljundi et al., 2018) and LwF (Li & Hoiem, 2017), we conducted hyperparameter tuning using grid search over the first two tasks. For MAS, we explored values of α and λ in [0.25, 0.5, 0.75]. Similarly for LwF, we searched for optimal α values within the same range.

Description	Value
optimizer	AdamW
weight decay	5.00e-03
β_1	0.9
β_2	0.95
batch size	512
warmup epochs	20
Total epochs	400
Max learning rate η_{max}	7.50e-05
Min learning rate η_{min}	1.50e-06
Constant learning rate η_{const}	1.875e-5
ViT-tiny	
Parameters	$7M$
Num Attention Heads	3
Num Layers	12
Hidden Size	192
Hidden Activation	GeLU
Positional Embedding	Learnable
Patch Size	2×2
Image Size	32×32
Dropout Rate	0.1

Table 3. Hyperparameters for pre-training on the small scale setup

Description	Value
optimizer	AdamW
weight decay	5.00e-03
β_1	0.9
β_2	0.95
learning rate schedule	cosine decay
batch size	128
warmup epochs	10
Total epochs	100
η_{max}	1.00e-03

Table 4. Hyperparameters for linear probing on the small scale setup.

E.3. Evaluation Metrics for MAE CPT

For evaluation, we employ three key metrics following (Lopez-Paz & Ranzato, 2017). **Average Accuracy** ($Acc = \frac{1}{T}\sum_{i=1}^{T} R_{T,i}$) provides an overall measure of model performance across all tasks, where T is the total number of tasks and $R_{T,i}$ represents the performance on task i after training on all T tasks. **Forward Transfer** ($FWT = \frac{1}{T-1}\sum_{i=2}^{T}(R_{i-1,i}-b_i)$) measures the model's ability to leverage knowledge from previous tasks, where b_i represents the accuracy of a randomly initialized feature extractor. **Backward Transfer**($BWT = \frac{1}{T-1}\sum_{i=1}^{T-1}(R_{T,i} - R_{i,i})$) quantifies the impact of subsequent task learning on previous task performance.

E.4. Implementation details MAE on Imagenet, Places , Firerisk

Our implementation builds upon the PyTorch (Paszke et al., 2019) implementation of MAE (He et al., 2022) with ViT-B/16 (He et al., 2022) backbone architecture with 12 layers, 768 hidden dimension, and 12 attention heads. For the infinite learning rate schedule, we maintain a constant learning rate η_{const} =3.75e-5 during constant phase, while our baseline employs the standard cosine decay schedule.To ensure fair comparison, both schedules share identical maximum $\eta_{max} = 1.5e-04$ and minimum learning rate, $\eta_{min} = 1.5e-06$, with hyperparameters for the cosine schedule directly adopted from He et al. (2022). We also employ learning rate scaling similar to Goyal et al. (2018). We list all the hyperparameters on Table 5. To mitigate catastrophic forgetting, we implement a replay buffer with a buffer size of $B = 0.05 \times |\mathcal{D}_i|$ per task, utilizing uniform random sampling for buffer updates. All experiments utilize the AdamW optimizer, (Loshchilov & Hutter, 2019) with training conducted over 300 epochs per task. Following, Ibrahim et al. (2024) we reset the optimizer states before each task. For linear probing as shown in Table 6, we utilized the LARS optimizer with no weight decay ($\lambda = 0$). The optimizer's momentum parameter β_1 was set to 0.9. The learning rate followed a cosine decay schedule with a maximum learning rate (η_{max}) of 1.00×10^{-1}. Training was conducted over 90 epochs with a large batch size of 4096 and included RandomResizedCrop augmentation. This configuration leverages the LARS optimizer's efficiency for large-batch training while maintaining training stability across the diverse image datasets.

Description	Value
optimizer	AdamW
weight decay	0.05
β_1	0.9
β_2	0.95
batch size	4096
warmup epochs	40
augmentation	RandomResizedCrop
Total epochs	300
Max learning rate η_{max}	1.50e-04
Min learning rate η_{min}	1.50e-06
Constant learning rate η_{const}	3.75e-05
ViT-B/16	
Parameters	$86M$
Num Attention Heads	12
Num Layers	12
Hidden Size	768
Hidden Activation	GeLU
Weight Decay	0.3
Positional Embedding	Learnable
Patch Size	16×16
Image Size	224×224
Dropout Rate	0.1

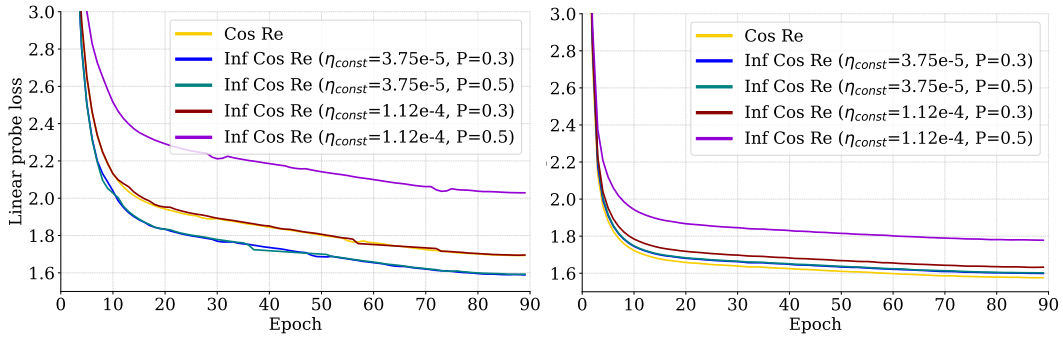
Table 5. Hyperparameters for pre-training MAE on Imagenet, Places and Firerisk

Description	Value
optimizer	LARS
weight decay	0
β_1	0.9
learning rate schedule	cosine decay
batch size	4096
warmup epochs	10
augmentation	RandomResizedCrop
Total epochs	90
η_{max}	1.00e-01

Table 6. Hyperparameters for linear probing on ImageNet, Places and Firerisk

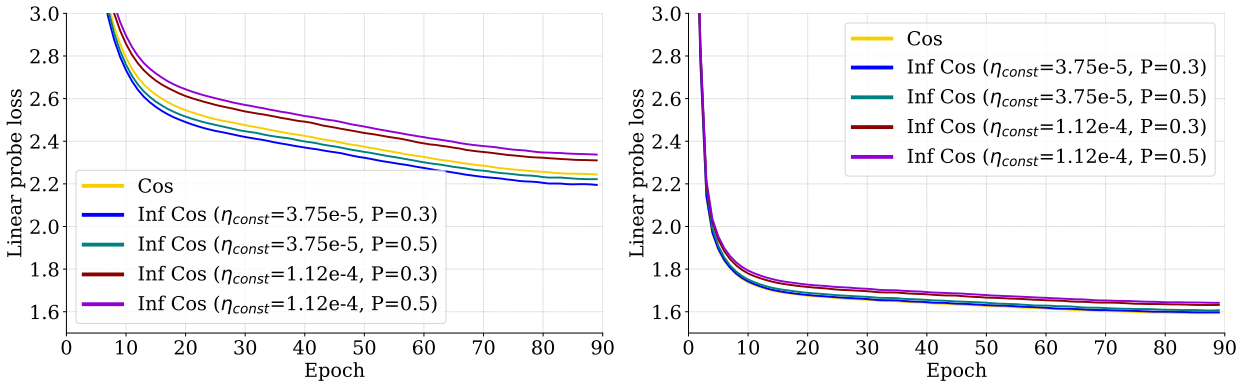
F. Effect of cooldown proportion and constant learning rate

In Figure 5, we analyze how the cooldown proportion and constant learning rate in the infinite schedule affect model performance on past and current tasks in ImageNet and Places2. The graphs compare linear probe validation loss across epochs for the standard cosine schedule and infinite schedules with varying configurations. Our analysis shows that lower constant learning rate ($\eta_{const} = 3.75e-5$) consistently reduces forgetting as compared to higher rate ($\eta_{const} = 1.12e-4$), Further, it can be observed that for $\eta_{const} = 3.75e-5$ cooldown proportion has negligble effect, but for $\eta_{const} = 1.12e-4$ shorter cooldown period ($P = 0.3$) outperform longer period ($P = 0.5$). This is likely because shorter cooldown phase represents quick decay to a stable η_{const} whereas a longer cooldown would mean a high learning rate for longer durations,



(a) Linear probe loss on Imagenet (b) Linear probe loss on Places2

Figure 5. Linear probe loss (↓ is better) for cosine schedule and infinite schedule with different constant learning rate and cooldown proportion with replay buffer. We observe that the infinite schedule with a lower $\eta_{const} = 3.75e-5$ has the lowest forgetting compared to other schedules.



(a) Linear probe loss on Imagenet (b) Linear probe loss on Places2

Figure 6. Linear probe loss (↓ is better) for cosine scheduler and infinite scheduler with different configurations without replay buffer. Infinite learning schedule with lower constant learning rate has lower forgetting compared to cosine schedule

which could cause instability in training, thus increasing forgetting.

Our analysis in Figure 6 (a) and (b) investigates learning dynamics in scenarios without a replay buffer, comparing the standard cosine schedule against infinite schedules through linear probe validation loss across epochs. The results mirror patterns observed with replay mechanisms, albeit with substantially higher catastrophic forgetting. Lower constant learning rates (η_{const}=3.75e-5) exhibit markedly reduced forgetting compared to higher rates (η_{const}=1.12e-4). For the lower constant learning rate, we observe that cooldown proportion has minimal impact on performance. In contrast, with higher constant learning rates, shorter cooldown periods yield better performance than longer ones. The dramatic increase in forgetting without replay underscores the critical importance of replay mechanisms in preserving cross-task performance.

G. Implementation details and hyperparameters for language pre-training

All models are trained with AdamW (Loshchilov & Hutter, 2019) on 100B tokens for each dataset, using a batch size of 1024 and a sequence length of 2048 approximately corresponding to $47,684$ total training steps. Optimizer states get reset between datasets, as this is common when we have to begin from an open weight model (e.g. from Huggingface (Wolf et al., 2020)). We train with data parallelism across 32 nodes, each equipped with 8 GPUs, maintaining a micro-batch size of 4. The training setup includes activation checkpointing (Chen et al., 2016) and ZeRO-1 optimizer sharding (Rajbhandari et al., 2020) to reduce memory overhead.

Table 7. **Hyperparameters of LR schedules.** All models used the same LR schedule hyperparameters. We refer the readers to (Ibrahim et al., 2024) section 7.2 for a more thorough explanation of these schedules.

Description	Value
Pre-training	
Total Iterations	47684
Max learning rate (η_{max})	$3 \cdot 10^{-4}$
Min learning rate (η_{min})	$3 \cdot 10^{-5}$
Constant learning rate (η_{const})	$1 \cdot 10^{-4}$
Warmup percent (N_w)	1
Cooldown iters percent (N_c)	60
Constant iters percent (N_d)	25
Continual Pre-training	
Total Iterations	47684
Max learning rate (η_{max})	$3 \cdot 10^{-4}$
Min learning rate (η_{min})	$3 \cdot 10^{-5}$
Constant learning rate (η_{const})	$1 \cdot 10^{-4}$
Warmup percent (N_w)	1
Cooldown iters percent (N_c)	0
Constant iters percent (N_d)	85

Table 8. **Hyperparameters of the ViT and LM transformers in our study.**

Description	Value
Dense Transformer LM	
Parameters	$571,148,288$
Non-Embedding Parameters	$439,814,144$
Num attention heads	16
Num layers	24
Hidden size	1024
FFN Hidden size	2816
FFN Type	GeGLU
Optimizer	AdamW
β_1,β_2	$0.9, 0.95$
Batch size	1024
Sequence length	2048
Hidden activation	GeLU
Weight decay	0.1
Gradient clipping	1.0
Decay	Cosine
Positional embedding	Rotary
GPT-J-Residual	True
Weight tying	False
Vocab Size	128000
Rotary PCT	0.25
ViT-B/16	
Parameters	$86,567,656$
Num Attention Heads	12
Num Layers	12
Hidden Size	768
FFN Hidden Size	3072
FFN Type	MLP
Optimizer	Adam
β_1, β_2	$0.9, 0.999$
Batch Size	4096
Sequence Length	197
Hidden Activation	GeLU
Weight Decay	0.3
Gradient Clipping	1.0
Positional Embedding	Learnable
Patch Size	16×16
Image Size	224×224
Dropout Rate	0.1
Common	

H. Pretraining with DCLM data

Figure 7 shows the validation loss on the DCLM dataset for cosine and infinite schedule with varying η_{const} and cooldown proportion P. We observe that the infinite schedule with a higher constant learning rate ($\eta_{const} = 2e-4$) and cooldown proportion ($P = 0.6$) performs better than the cosine schedule and the other configurations of the infinite schedule. The final checkpoint, in the case of infinite schedule, is obtained via annealing which we perform for 15% of the total iterations after the constant phase, as shown in Figure 1. It can be inferred that the infinite schedule with $\eta_{const} = 2e-5$ and $P = 0.6$

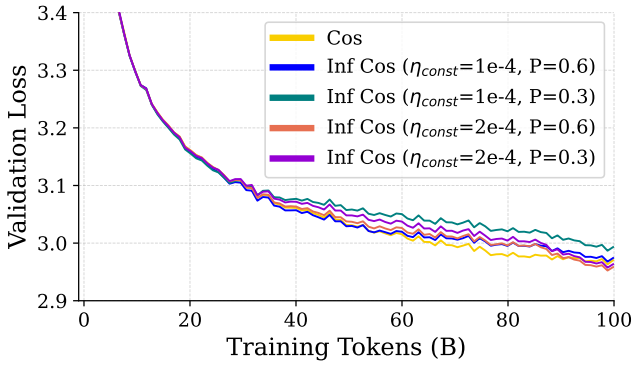


Figure 7. Validation Loss(↓ is better) for Different schedules, Training and Validating on DCLM Dataset

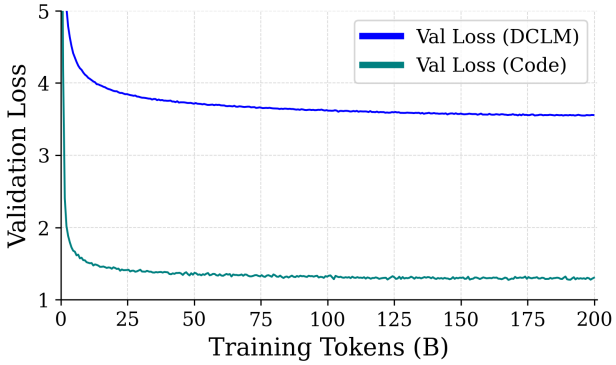


Figure 8. Validation Loss(↓ is better) for Cosine while training on combined DCLM and Code

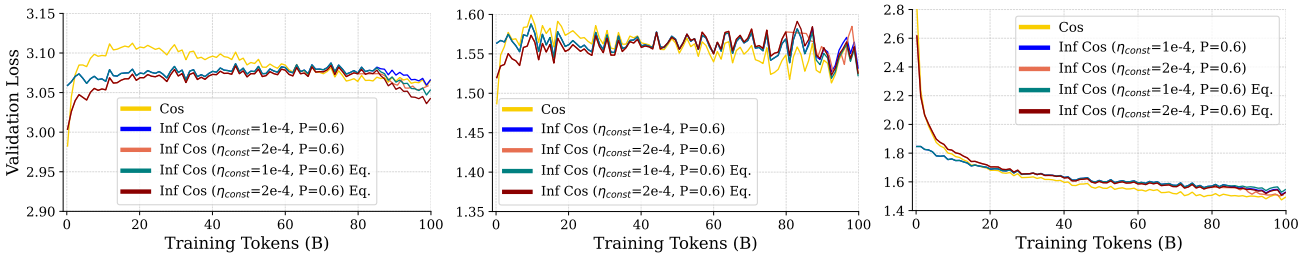
performs the best, with validation loss rapidly decaying in the annealing phase. We also observe that a shorter cooldown phase ($P = 0.3$) results in suboptimal performance with higher validation loss, thus indicating that a longer cooldown phase is beneficial. We note that this corresponds to $28K$ steps. As for the η_{const}, we observe that both $1e-4$ and $2e-4$ perform similarly, with the latter having a slightly lower validation loss, indicating that a higher constant learning rate gives a better exploration possibility during training.

I. Pre-training with combined DCLM and Stack Data

We show the validation loss on the combined DCLM and Stack dataset with cosine scheduling in Figure 8. It can be inferred that both the validation loss on DCLM and Stack is worse as compared to continual pre-training with infinite learning schedule. This indicates that the infinite schedule is able to preserve the knowledge of the previous task as well as improve transferability better as compared to cosine schedule, even with combined training on both tasks.

J. Dynamic Adjustment of Replay Buffer with equal proportion of pre-training data during Annealing Phase

We perform an ablation by annealing on equal proportion of data, i.e. 33.33% each of DCLM ($\mathcal{D}_0$), Stack ($\mathcal{D}_1$) and German ($\mathcal{D}_2$). As shown in Figure 9, it can be inferred that the performance on previous tasks improve as compared to the config where we use 50% of buffer, since now the proportion of previous tasks data has increased (25% for each of DCLM and Stack $\rightarrow$ 33.33%). But this does not deteriorate the downstream performance on German data. Hence in cases, where upstream performance is critical, we can anneal on an equal proportion of data.



(a) Valid. Loss on DCLM (b) Valid. Loss on Stack (c) Valid. Loss on German

Figure 9. Validation Loss (↓ is better) for different schedules accompanied with replay. The total fraction of replay is 50%, with 25% of DCLM and other 25% from Stack Data. CPT is on German data ($\mathcal{D}_2$), validating on all German ($\mathcal{D}_2$) DCLM ($\mathcal{D}_0$) and Stack ($\mathcal{D}_1$) datasets. We further perform an experiment with equal proportion of data during annealing, referred to as Eq. in the above graphs. It can be observed that equal proportions improve the upstream performance (lower validation loss on DCLM and Stack). The downstream performance is quite similar to other Infinite schedules (both $\eta_{const} \in \{1e-4, 2e-4\}$) config. The downstream performance of the infinite schedule on the current task (German) is comparable to cosine.

K. Effect of Checkpoint selection in Past task performance

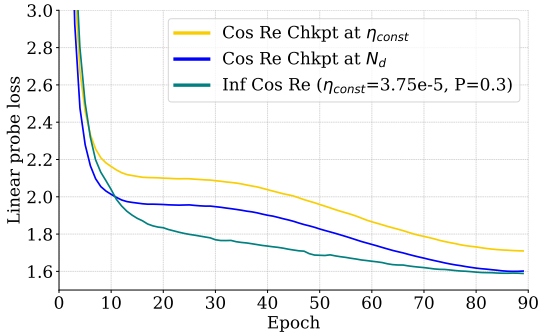


Figure 10. Comparison of linear probe loss on Imagenet (D_0) with checkpoints obtained while training Places 2 (D_1). All experiments were conducted with replay. We compare different checkpoint selection strategies against the infinite scheduler. Infinite Cosine learning rate schedule (green) achieves consistently lower loss compared to cosine schedules restarted from either checkpoint at η_{const} (yellow) or N_d (blue).

To address the relative importance of the learning rate scheduling function versus the choice of checkpoint, we conducted an additional ablation study. Figure 10 shows the performance comparison of three different approaches:

1. Cosine learning rate schedule with checkpoint from η_{const} checkpoint (yellow line): This checkpoint is obtained at the point where the cosine schedule reaches the learning rate corresponding to the value of η_{const} used in our infinite cosine schedule.
2. Cosine learning rate schedule with checkpoint from N_d checkpoint (blue line): This checkpoint is obtained at the time corresponding to when the annealing of the infinite schedule begins.
3. Infinite Cosine learning rate schedule with $\eta_{\text{const}} = 3.75\text{e-}5$ and $P = 0.3$ (green line): Our best performing scheduler from main paper.

As shown in Figure 10, the Infinite Learning Rate schedule consistently achieves lower linear probe loss throughout training compared to both cosine schedule variants. The results demonstrate that this approach provides more effective representation learning compared to cosine schedule with early checkpointing.

Strategic checkpoint selection plays a critical role in preserving previously acquired knowledge. However, this flexibility is not inherently supported by the standard cosine schedule, which is typically designed to run uninterrupted until the

Strategy	Overall Acc	Overall REM	Overall BWT	Overall FWT	CL_score
Cosine + ER (5%)	44.76	97.11	-2.89	12.85	37.96
Infinite cosine + ER (5%)	**50.56**	**99.54**	**-0.46**	12.78	**40.61**
Cosine	43.61	84.61	-15.39	12.80	31.41
Infinite cosine	44.36	86.98	-13.02	**13.08**	32.85
Full baseline	53.40	-	-	-	-

Table 9. Comparison of forgetting metrics (as defined in Díaz-Rodríguez et al. (2018)) across various methods. Infinite cosine with replay shows competitive results with lower forgetting, approaching the performance of the Full baseline.

end of training. In our comparison experiments, we manually extracted checkpoints at arbitrary points with high learning rates which is an approach that deviates from standard usage of cosine schedule. In contrast, the Infinite Learning Rate schedule naturally enables such flexibility through its constant learning rate phase, explicitly designed for this requirement. This built-in mechanism aids in knowledge retention across tasks, contributing to the consistently lower linear probe loss observed compared to other approaches.

L. Additional Forgetting metrics

We present a more thorough analysis of forgetting dynamics in our continual learning framework in Table 9. While our main paper reports standard metrics which measured at the end of the training including Average Accuracy (AA), Forward Transfer (FWT), and Backward Transfer (BWT), this section extends our evaluation with additional metrics proposed by Díaz-Rodríguez et al. (2018) to provide a more complete picture of the retention capabilities of our approach after every task.

Díaz-Rodríguez et al. (2018) provides metrics to analyze the performance of the model at every timestep to incorporate the dynamic nature of CL. Hence we use an overall prefix for these metrics which are taken as an average after every time step. **Overall Accuracy** ($A = \frac{\sum_{i \geq j}^{T} R_{i,j}}{\frac{T(T+1)}{2}}$) gives the average of accuracy on all tasks after every timestep. **Overall Backward transfer** ($BWT = \frac{\sum_{i=2}^{T} \sum_{j=1}^{i-1} (R_{i,j} - R_{j,i})}{\frac{T(T-1)}{2}}$) measures the backward transfer after every timestep, while **Overall Forward transfer** ($FWT = \frac{\sum_{i<j}^{T} R_{i,j}}{\frac{T(T-1)}{2}}$) measures the forward transfer after every time step. **Overall remembering** ($REM = 100 - |min(BWT, 0)|$) quantifies the amount of knowledge remembered by the model across all time steps. Finally, we calculate the CL score to get a weighted average of all the metrics $CL_{score} = \sum_{i=1}^{\#c} w_i c_i$, where $\#c$ denotes the total number of metrics used. We use an equal weighting and report the equal weighted average. We additionally show the performance of a model which is pre-trained on the combination of all datasets, then linear probed on each dataset and obtained accuracy. We report the overall accuracy of it across all tasks as a reference in the table.

M. Evaluation on LLM benchmarks

While the validation loss provides a good measure of performance on the pre-training objective, LLMs abilities are typically judged by their performance on evaluation tasks. With the caveat that we use base models, i.e our models have not been instruction-tuned, fine-tuned, or adapted to human preferences in any way, we present their evaluation on popular benchmarks in this section. Table 10 shows the evaluation results on various benchmarks for different schedules. We observe that with replay, the infinite schedule with $\eta_{const} = 2e-4$ gives the best performance across all the benchmarks with an average accuracy of **46.81%**. For the model after pre-training on German, infinite schedule with $\eta_{const} = 1e-4$ gives the best performance across the German evaluation benchmarks with an average accuracy of **28.10%** as shown in Table 11. These results highlight that infinite schedules not only circumvent catastrophic forgetting but also provide a competitive advantage in downstream evaluations.

Scheduler	Training Tokens	LOAI	HS	OBQA	WG	ARC-e	PIQA	LQA	**Avg.**
Cosine	100B DCLM → 100B Stack	33.17	31.79	25.2	49.17	42.72	62.51	25.49	38.58
	100B DCLM → 100B Stack (50% Replay)	47.56	43.96	32.2	52.33	50.50	**69.53**	**28.57**	46.37
Inf Cos (η_{const} = 1e-4)	100B DCLM → 100B Stack	35.73	33.47	26.0	51.78	43.39	62.19	28.11	40.09
	100B DCLM → 100B Stack (50% Replay)	**49.16**	43.72	**32.6**	52.09	50.59	68.93	27.65	46.39
Inf Cos (η_{const} = 2e-4)	100B DCLM → 100B Stack	33.99	32.44	26.2	51.93	43.10	60.99	26.57	39.31
	100B DCLM → 100B Stack (50% Replay)	48.73	**44.42**	31.6	**54.85**	**51.73**	69.31	27.04	**46.81**

LOAI: LambdaOpenAI, HS: HellaSwag, OBQA: OpenBookQA, WG: WinoGrande,LQA: LogicQA

Table 10. Zero-shot results on popular LM benchmarks. Normalized accuracy is reported. We observe on average, as expected, that the infinite schedule with $\eta_{const} = 2e-4$ with a 50% replay gives the best performance across all the benchmarks. Even without Replay, both the infinite schedules give better performance as compared to cosine. This demonstrates the effectiveness of infinite schedule in mitigating forgetting.

Scheduler	Training Tokens	ARC-de	HS-de	**Avg.**
Cosine	100B DCLM → 100B Stack → 100B German	23.29	32.89	28.09
Inf Cos (η_{const} = 1e-4)	100B DCLM → 100B Stack → 100B German	**23.64**	32.56	**28.10**
Inf Cos (η_{const} = 2e-4)	100B DCLM → 100B Stack → 100B German	23.21	**32.90**	28.06

Table 11. Zero-shot results showing adaptability of the model after completing training on the German data ($\mathcal{D}_2$) on popular LM benchmarks. We observe that the infinite schedule with $\eta_{const} = 1e-4$ achieves the best performance on German evaluation benchmarks, demonstrating that the infinite schedule adapts more effectively than the cosine schedule on the most recent task.

N. Practical Guide for Hyperparameter Selection in Infinite Cosine Scheduling

Selecting learning rate hyperparameters for continual pre-training can be computationally expensive due to the scale of training. This challenge applies to both repeated cosine and infinite cosine schedules and is not unique to the latter. However, when control over the initial pre-training phase is available, recent advances in hyperparameter-transfer techniques (Yang et al., 2021) can help reduce tuning costs substantially.

While hyperparameter stability is not the primary focus of this work, we provide practical guidance based on consistent trends observed in our experiments. The following empirically grounded rule-of-thumb can serve as a starting point for configuring the infinite cosine schedule effectively:

- **Step 1: Selecting** η_{max} - Choose a maximum learning rate (η_{max}) that yields stable validation loss on the initial domain. This follows standard large-scale pre-training practice. If a tuned cosine schedule is already available from prior work, its η_{max} serves as a strong candidate.
- **Step 2: Setting** η_{const} - Define the constant learning rate for Inf-Cos as the midpoint between η_{max} and η_{min}. For example, with $\eta_{max} = 3e-4$ and $\eta_{min} = 3e-5$, a suitable baseline is $\eta_{\text{const}} \approx 1.65 \times 10^{-4}$.

We further find the following trends to be consistent along with takeaways from (Hägele et al., 2024)

- **Without replay:** When no replay buffer is available, a relatively lower η_{const} is preferred to better preserve previously acquired knowledge as shown in Figure 3 and Figure 2.
- **With replay:** When replay is available, a relatively higher η_{const} can be utilized to improve adaptability without severely impacting retention — as shown in Figure 4.

These insights allow practitioners to configure Infinite Schedules without exhaustive search, and makes it a more flexible and robust scheduler as compared to repeated cosine.