A Practitioner's Guide to Continual Multimodal Pretraining

Karsten Roth[1,2,6*] **Vishaal Udandarao**[1,3*] **Sebastian Dziadzio**[1°] **Ameya Prabhu**[1°]
Mehdi Cherti[4] **Oriol Vinyals**[5] **Olivier Hénaff**[5]
Samuel Albanie[†] **Matthias Bethge**[1†] **Zeynep Akata**[2,6,7†]

[1]Tübingen AI Center, University of Tübingen [2]Helmholtz Munich [3]University of Cambridge
[4]LAION, Jülich Supercomputing Center (JSC), Research Center Jülich (FZJ), Helmholtz
[5]Google DeepMind [6]Munich Center for ML [7]Technical University of Munich
[*]equal project lead, order interchangeable [°]core contributors [†]equal supervision.

Abstract

Multimodal foundation models, despite being extensively pretrained, become outdated over time. Research into continual pretraining mainly explores (1) infrequent, indiscriminate updates on large-scale new data, or (2) frequent, sample-level updates. However, practical model deployment often operates in the gap between these limit cases, as real-world applications require continual adaptation to specific subdomains or tasks. In this work, we complement current approaches through a new, continual multimodal pretraining test bed with realistic compute constraints and practical deployment requirements (`FoMo-in-Flux`), and provide *comprehensive practical guidance* for effective continual model updates—investigating different method choices, pipeline design and data-centric deployment scenarios.

1 Introduction

Foundation models are expensive to train [137], but quickly become outdated. Hence, they need *continual pretraining*, mainly of two categories: (1) infrequent, large-scale updates, and (2) frequent, but minimal updates. However, many applications operate in the complex gap between these cases. In this work, we provide a new research framework to emulate these complex practical deployment scenarios, and study how to make continual pretraining succeed. Our contributions are as follows:

Creating `FoMo-in-Flux` (***Fo**undation-**Mo**dels-**in**-**Flux***), building on 63 datasets, enhanced with captions to enable multimodal pretraining. Unlike noisy web-crawl datasets like TiC-DataComp [45, 42], `FoMo-in-Flux` comprises curated, high-quality samples with fine-grained class information and precise control over data streams spanning numerous visual and semantic domains.
Realistic Continual Pretraining. Unlike traditional continual learning research, we avoid the *practically unnecessary restriction of limited storage* [131, 132], and allow unrestricted access to both pretraining and adaptation data— we only impose a restriction on the compute budgets.
A Methods Study. Using `FoMo-in-Flux`, we test various strategies for multiple sequential, *minor* continual pretraining updates—ranging from regularization-based strategies like `EWC` [82] and `SI` [202], simple `finetuning`, parameter-efficient adaptation like `LoRA` [68], to model merging [74].
The Importance of Continual Pretraining Strategies beyond simple method choices, such as learning rate scheduling, and proposed task-dependent meta schedules to facilitate long-term continuous, controlled model updates. Moreover, we study both model and compute scaling, and give an overview of important experimental design choices for a continual multimodal pretraining pipeline.
A Data-centric Perspective on Continual Pretraining. Lastly, the concepts and tasks that a model should improve on often arise in sequence, driven by the use-cases it is deployed for. Retaining fine-grained control over the sequence of semantic and visual concepts allows us to create and study six realistic data-centric streams, including *easy to hard ordering*, *concept frequency ordering*, and *dataset-incremental ordering*. Finally, we provide insights into the impact of data mixtures on the accumulation and retention trade-off as new concepts and subdomains are introduced.

38th Conference on Neural Information Processing Systems (NeurIPS 2024) Workshop on Scalable Continual Learning for Lifelong Foundation Models.

Together, our study aims to assist practitioners in understanding how various factors like deployment scenarios, data limitations, continual learning and finetuning strategies, and constraints on computing power or model capacity affect the ability to carry out *long-term, controlled model updates*.

2 Introducing the `FoMo-in-Flux` Benchmark

`FoMo-in-Flux` (***Fo**undation-**Mo**dels-**in**-**Flux***) extends the study of continual pretraining beyond monolithic pretraining datasets, such as TiC-RedCaps/TiC-DataComp [45], to specialized subdomains with fine-grained control over data streams and adaptation over long task horizons (see §6,7 for an extensive discussion of and comparison to related works). `FoMo-in-Flux` comprises 63 classification and retrieval datasets over $2.53M$ samples and $23,045$ concepts spanning diverse visual and semantic domains s.a. natural images, sketches, synthetic and long-tailed data (section 7). Building concept-first allows experimentation with very precise and controlled streaming orderings. Moreover, by operating on much cleaner data building blocks than web-crawled datasets like TiC-RedCaps or DataComp [45, 42], we ensure cleaner alignment between concepts and images. Crucial caption-data is generated procedurally for datasets with ground truth information, or through a two-stage captioning mechanisms. The 63 datasets are divided into 41 datasets used for *adaptation* only, and 22 hold-out datasets to probe *retention* of initial zero-shot generalization.

Continual Pretraining Updates. We start with a model θ_0 pretrained on a large dataset $\mathcal{P}$, and an empty buffer $\mathcal{B}$. We allow unrestricted access to pretraining data and an unlimited replay buffer $\mathcal{B}$, as data storage is a negligible contributor to real-world cost [131, 132]. At each update step $j \in \{1, 2, \ldots, T\}$ within the allocated update budget, the following happens: **(1)** The stream reveals a task update pool of n_j image-text pairs $\mathcal{D}_j = \{(i_k^j, t_k^j)\}_{k=1}^{n_j}$ spanning $\mathcal{C}_j$ concepts. **(2)** We create the training data mixture $\mathcal{S}_j$ by sampling from $\mathcal{P}$, buffer $\mathcal{B}$, and current task data $\mathcal{D}_j$ with respective ratios $\lambda_\mathcal{P}, \lambda_\mathcal{B}$, and $\lambda_\mathcal{D}$, such that $\lambda_\mathcal{P} + \lambda_\mathcal{B} + \lambda_\mathcal{D} = 1$. If samples in $\mathcal{B}$ are insufficient (particularly at the start of task adaptation), we oversample from $\mathcal{D}_j$, with $\lambda_\mathcal{D}$ fixed. **(3)** We apply a continual update method $\mathcal{M}$ with a fixed compute budget F: $\theta_j = \texttt{train}(\mathcal{M}, \mathcal{D}_j, \theta_{j-1})$. This compute budget F also determines the overall number of update steps conducted. **(4)** We add samples from the update pool $\mathcal{D}_j$ to the unrestricted buffer $\mathcal{B}$. However, while all samples can be stored in buffer $\mathcal{B}$, they cannot all be sampled for training set $\mathcal{S}$, as the compute budget F imposes an implicit memory restriction [132].

Continual Pretraining Computational Cost. To ensure practical and fair comparison, we fix computation cost budget for each time step to account for the efficiency of each method. However, a single compute metric does not paint a complete picture of practically relevant efficiency [35, 114]. To account for this, we introduce *Memory-Adjusted-FLOPs (MAFs)*, a novel metric that highlights two aspects most relevant for practitioners: Total FLOPs per iteration and maximum utilization of device memory. To compute MAFs, we multiply FLOPs count of each method by a *memory multiplier*—the ratio of a method's maximum memory used to that of full base model fine-tuning. The total amount of MAFs derives directly from pretraining budgets in [26] (see supp. for details).

Designing Data-Centric Task-Sequences. We design realistic data deployment scenarios by breaking down datasets into individual concepts, which are then ordered according to a chosen criterion. This is visualized in Fig. 7. For this, having a controlled set of image-caption pairs is critical, as it allows for well-defined and meaningful arrangement of concepts into sequences according to an ordering $\pi(\mathcal{C})$. Each ordering π divides the set of samples $\mathcal{D}$ into T disjoint subsets $\{\mathcal{D}_1, \ldots, \mathcal{D}_T\}$ of concepts $\mathcal{C}$ sampled without replacement, i.e. $\mathcal{C}_i \bigcap \mathcal{C}_j = \phi, \forall i, j$. We define and motivate six different orderings below (for more details, see Supp.): **1. Easy-To-Hard Ordering** (`performance`) motivated by curriculum learning [54, 149, 158, 163, 201] assumes users deploying their model to related concepts and usecases first, with incremental movement towards to harder concepts. **2. Concept Frequency Ordering** (`concept-frequency`) draws motivation from Udandarao et al. [173], with user requests starting from least frequent concepts first (more likely edge cases causing undesired performance drops) to more frequent concepts represented well in the pretraining pool. **3. Concept Similarity Ordering** (`similarity`), inspired by Yıldız et al. [197], assumes that training on conceptually similar tasks allows users to minimize catastrophic forgetting over tasks. **4. Time-incremental Ordering** (`time`) [12, 69, 18, 131, 45] arranges in chronological order. **5. Dataset-Incremental Ordering** (`dataset`) is motivated by [143, 107, 108, 183, 199], but extended to a larger sequence of datasets broken down into the desired number of tasks T. **6. Random Ordering** (`random`), a baseline class-incremental ordering widely used across continual learning setups [144, 193, 66, 132], mimics a scenario with unstructured user requests for model improvement.

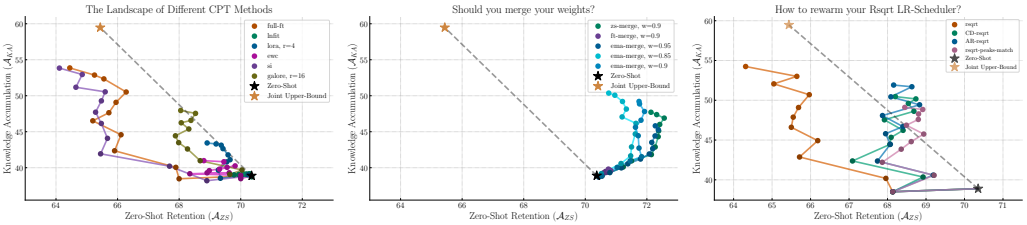


Figure 1: Impact of parameter-efficient and continual learning methods (**left**), model merging (**middle**) and meta learning rate schedules (**right**) on practical continual pretraining.

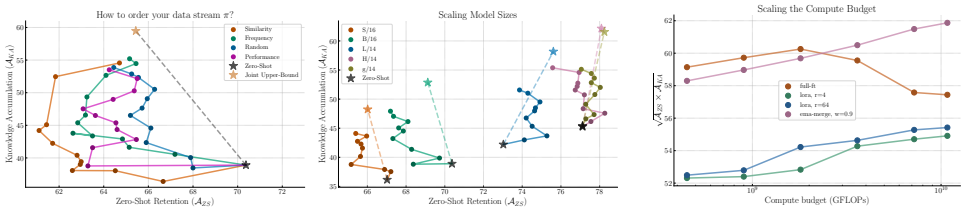


Figure 2: Impact of different data deployment scenarios (**left**), model size (**center**) and compute scaling (**right**) on practical continual pretraining.

3 Experiments

This section summarizes key experimental results conducted over `FoMo-in-Flux`. For more details, and additional experimental studies, we refer to the supplementary.

Which methods should you opt for in continual pretraining over longer update cycles? Using `FoMo-in-Flux`, we conduct an in-depth study across six different method families: naive continual finetuning (`Full-FT` [72]), parameter-selective tuning (`LNFit` [34]), parameter-additive tuning (`LoRA`), continual learning with regularization (`EWC` [82], `SI` [202]), low-rank approximations on the model update gradients (`GaLore` [212]) and model merging (testing exponential moving average, zero-shot and finetune-based merging - see Supp. for details). Results are visualized in fig. 1 (left and center), with full joint finetuning as adaptation upper bound, and initial zero-shot performance as adaptation lower and retention upper bound: Naive continual finetuning and parameter-selective tuning provide the extreme points in knowledge accumulation and retention of original zero-shot performance. Switching from `GaLore` to parameter-efficient tuning with e.g., `LoRA` provides near linear interpolation points between both. However, judiciously merging model weights exhibits unique long-horizon continual pretraining behaviour, allowing for significant and consistent knowledge accumulation across update tasks with minimal forgetting; breaking with the hypothetical linear trade-off line between original zero-shot performance and the joint finetuning upper-bound!

Meta Learning Rate Schedules. By default, each pretraining update is treated independently with a fixed schedule design - each task rewarms, decays and cools down independently to each other. However, as continual pretraining updates appear in succession, forgetting of previously seen tasks beyond initial pretraining knowledge has to also be accounted for. On top of that, with every task update, the model is encouraged to move further away from its pretraining starting point. To account for both factors, we introduce meta learning rate schedules. These derive *naturally and hyperparameter-free* from hypothetical scenarios wherein the previous task schedule is simply extended across incoming tasks in variable fashion (see 10.1 for more details). While we test four different meta schedules, our results in fig. 1 (right) highlight the impact of different meta-schedule variations on top of the standard reciprocal square-root scheduler [204]. Indeed, for longer continual pretraining sequences, meta-schedules deriving from *rsqrt* schedule lend themselves much better to longer-horizon continual pretraining tasks: As can be seen, the *rsqrt* meta-schedules achieve nearly the same gain in knowledge accumulation, while *vastly increasing the amount of retained knowledge* and exceeding the hypothetical linear zero-shot vs joint finetuning trade-off line!

Concept- and Sample-based Deployment Scenarios. Across the noted deployment scenarios (see fig. 2 left), while the `concept-frequency` stream has the marginally best accumulation-retention tradeoff, we find that *convergence end-points are surprisingly similar* - especially w.r.t. the initial zero-shot and the joint finetuning upper bound reference points. However, while endpoints are remarkably similar, different orderings π induce significantly different trajectories in the accumulation-retention

space, with `similarity` the most sample inefficient ordering, while `random` produces the most favorable trajectories. These insights mean that the type and order of continual minor model updates primarily impact intermediate model versions. This is crucial to account for with respect to the model release horizon and the expected time frame before conducting large-scale continual pretraining updates. However, it also means that across long update horizons irrespective of particular task orders, continually pretrained models arrive at similar performance breakpoints.

Scaling Model Size. As can be seen in fig. 2 (center), we find that with a controlled increase of model size, the ability to continually pretrain over longer minor update cycles improves. While the absolute change in knowledge accumulation $\mathcal{A}_{KA}$ remains rather consistent (within the interval of 8% and 10%), zero-shot retention $\mathcal{A}_{ZS}$ improves, in parts even with slight positive backward transfer for ViT-L14 (tripling the parameter count w.r.t ViT-B/16). These results suggest that model scaling can benefit long-term re-use and the opportunity to maintain and consistently improve the base model over longer minor update cycles, suggesting model scaling helps mitigate forgetting [140].

Scaling Compute Budgets. Instead of investing into compute increases through larger model sizes, one can also adjust the directly allocated compute budgets; changing for example the number of update steps and task iterations. For our reference model B/16 and its associated compute budget of 1.8×10^9 FLOPs, we thus conduct $2\times$, $4\times$ and $6\times$ increases, as well as $0.5\times$ and $0.25\times$ reductions to understand how the continual pretraining abilities vary as a function of associated compute budgets and the applied continual pretraining strategies of choice. As seen in fig. 2 (left) which aggregates knowledge accumulation $\mathcal{A}_{KA}$ and zero-shot retention $\mathcal{A}_{ZS}$ through their geometric mean, simple continual finetuning (brown) can not consistently leverage increased compute budgets; having to trade off increased knowledge accumulation with a disproportionate loss in the models initial generalization capabilities. However, coupled with simple model merging, we find that models become much better at effectively utilizing the additional budget increase; exhibit a log-linear budget-performance relation.

4 Conclusion

This work introduces `FoMo-In-Flux` - a novel, large-scale, fine-grained controllable and long horizon continual multimodal pretraining benchmark. Using `FoMo-In-Flux`, we conduct an extensive investigation into how to continually pretrain contrastive multimodal models from a data-centric, method-centric, and training-recipe-centric perspective. Key findings, alongside those provide in the supplementary, are as follows: **(1) Method Choices.** Under practical update scenarios, continual learning and parameter-efficient fine-tuning techniques favor knowledge retention (stability) while simple fine-tuning focuses on adaptation (plasticity). However, in combination with **model merging**, fine-tuning sufficiently addresses this trade-off, allowing for strong knowledge retention **and** adaptation. **(2) Meta Learning Rate Schedules.** Learning rates matter, and can naturally be accounted for in long-horizon continual pretraining via **meta** learning rate schedules, reducing the loss of pretraining knowledge while preserving high adaptation performance. **(3) Model and Compute Scaling.** Simple fine-tuning does not scale well with increased compute resources or more frequent updates, unlike parameter-efficient fine-tuning, and particularly fine-tuning with model merging. On the other hand, **increasing model size** helps it acquire new knowledge while retaining its foundational properties, even within the same compute budget. **(4) Data-centric Stream Orderings.** The **order** in which data updates are applied significantly impacts the model's ability to learn new information and retain its zero-shot capabilities. This is important to account for during deployment. However, when underlying data distributions are the same, models converge to **comparable final performance** across update sequences. **(5) Data mixture ratio.** The ratio between pretraining-, update-, and buffer data affects the model's final performance, and "IID-fying" knowledge accumulation is crucial. Specifically, replaying previous adaptation tasks helps the model adapt better, while replaying pretraining data is less critical. However, the choice of pretraining data can influence how well the model retains knowledge.

References

[1] Dreamlike photoreal v2.0. https://huggingface.co/dreamlike-art/dreamlike-photoreal-2.0. 20, 24

[2] 10 monkey species. URL https://www.kaggle.com/datasets/slothkong/10-monkey-species. 23

[3] Davide Abati, Jakub Tomczak, Tijmen Blankevoort, Simone Calderara, Rita Cucchiara, and Babak Ehteshami Bejnordi. Conditional channel gated networks for task-aware continual learning. In *CVPR*, 2020. 19

[4] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 20

[5] Takuya Akiba, Makoto Shing, Yujin Tang, Qi Sun, and David Ha. Evolutionary optimization of model merging recipes. *arXiv preprint arXiv:2403.13187*, 2024. 27

[6] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/97af07a14cacba681feacf3012730892-Paper.pdf. 22

[7] Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *ACL (Short)*, 2022. 25, 26, 27

[8] Thomas Berg, Jiongxin Liu, Seung Woo Lee, Michelle L. Alexander, David W. Jacobs, and Peter N. Belhumeur. Birdsnap: Large-scale fine-grained visual categorization of birds. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2019–2026, 2014. doi: 10.1109/CVPR.2014.259. 20, 21, 22

[9] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad – a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 22

[10] Dan Biderman, Jose Gonzalez Ortiz, Jacob Portes, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and John P. Cunningham. Lora learns less and forgets less, 2024. URL https://arxiv.org/abs/2405.09673. 27

[11] Dan Biderman, Jose Gonzalez Ortiz, Jacob Portes, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, et al. Lora learns less and forgets less. *arXiv preprint arXiv:2405.09673*, 2024. 20

[12] Jorg Bornschein, Alexandre Galashov, Ross Hemsley, Amal Rannen-Triki, Yutian Chen, Arslan Chaudhry, Xu Owen He, Arthur Douillard, Massimo Caccia, Qixuan Feng, et al. Nevis' 22: A stream of 100 tasks sampled from 30 years of computer vision research. *JMLR*, 2023. 2, 18, 19, 24

[13] Matteo Boschini, Lorenzo Bonicelli, Pietro Buzzega, Angelo Porrello, and Simone Calderara. Class-incremental continual learning into the extended der-verse. *TPAMI*, 2022. 30

[14] Matteo Boschini, Lorenzo Bonicelli, Pietro Buzzega, Angelo Porrello, and Simone Calderara. Class-incremental continual learning into the extended der-verse. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1–16, 2022. ISSN 1939-3539. doi: 10.1109/tpami.2022.3206549. URL http://dx.doi.org/10.1109/TPAMI.2022.3206549. 18

[15] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014. 23

[16] Chris Burgess and Hyunjik Kim. 3d shapes dataset. https://github.com/deepmind/3dshapes-dataset/, 2018. 20, 22, 23

[17] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. In *NeurIPS*, 2020. 19, 30

[18] Zhipeng Cai, Ozan Sener, and Vladlen Koltun. Online continual learning with natural distribution shifts: An empirical study with visual data. In *ICCV*, 2021. 2, 18, 19, 24

[19] Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulo, Elisa Ricci, and Barbara Caputo. Modeling the background for incremental learning in semantic segmentation. In *CVPR*, 2020. 19

[20] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021. 36

[21] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. In *ICLR*, 2019. 19

[22] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019. 19

[23] Cheng Chen, Junchen Zhu, Xu Luo, Hengtao Shen, Lianli Gao, and Jingkuan Song. Coin: A benchmark of continual instruction tuning for multimodel large language model. *arXiv preprint arXiv:2403.08350*, 2024. 20

[24] Jie Chen, Zhipeng Chen, Jiapeng Wang, Kun Zhou, Yutao Zhu, Jinhao Jiang, Yingqian Min, Wayne Xin Zhao, Zhicheng Dou, Jiaxin Mao, et al. Towards effective and efficient continual pre-training of large language models. *arXiv preprint arXiv:2407.18743*, 2024. 19

[25] Qizhou Chen, Taolin Zhang, Dongyang Li, Longtao Huang, Hui Xue, Chengyu Wang, and Xiaofeng He. Lifelong knowledge editing for llms with retrieval-augmented continuous prompt learning. *arXiv preprint arXiv:2405.03279*, 2024. 20

[26] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. *arXiv preprint arXiv:2212.07143*, 2022. 2, 22, 29, 30, 32

[27] Pinaki Nath Chowdhury, Aneeshan Sain, Ayan Kumar Bhunia, Tao Xiang, Yulia Gryaditskaya, and Yi-Zhe Song. Fs-coco: Towards understanding of freehand sketches of common objects in context. In *ECCV*, 2022. 20, 22

[28] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. 22

[29] Adam Coates, Andrew Ng, and Honglak Lee. An Analysis of Single Layer Networks in Unsupervised Feature Learning. In *AISTATS*, 2011. https://cs.stanford.edu/~acoates/papers/coatesleeng_aistats_2011.pdf. 23

[30] Guillaume Couairon, Matthijs Douze, Matthieu Cord, and Holger Schwenk. Embedding arithmetic of multimodal queries for image retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4950–4958, 2022. 33

[31] Nico Daheim, Thomas Möllenhoff, Edoardo Ponti, Iryna Gurevych, and Mohammad Emtiyaz Khan. Model merging by uncertainty-based gradient matching. In *ICLR*, 2024. 27

[32] Matthias De Lange and Tinne Tuytelaars. Continual prototype evolution: Learning online from non-stationary data streams. In *ICCV*, 2021. 19

[33] Matthias De Lange, Gido van de Ven, and Tinne Tuytelaars. Continual evaluation for lifelong learning: Identifying the stability gap. *arXiv preprint arXiv:2205.13452*, 2022. 19, 26

[34] Thomas De Min, Massimiliano Mancini, Karteek Alahari, Xavier Alameda-Pineda, and Elisa Ricci. On the effectiveness of layernorm tuning for continual learning in vision transformers. In *ICCV-W*, 2023. 3, 25, 26, 27

[35] Mostafa Dehghani, Yi Tay, Anurag Arnab, Lucas Beyer, and Ashish Vaswani. The efficiency misnomer. In *ICLR*, 2022. 2

[36] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 23

[37] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. 23

[38] Nick DiSanto. Isic melanoma dataset, 2023. URL https://dx.doi.org/10.21227/9p2y-yq09. 20, 22

[39] Xiaoyi Dong, Jianmin Bao, Ting Zhang, Dongdong Chen, Shuyang Gu, Weiming Zhang, Lu Yuan, Dong Chen, Fang Wen, and Nenghai Yu. Clip itself is a strong fine-tuner: Achieving 85.7% and 88.0% top-1 accuracy with vit-b and vit-l on imagenet. *arXiv preprint arXiv:2212.06138*, 2022. 33

[40] Yanai Elazar, Akshita Bhagia, Ian Magnusson, Abhilasha Ravichander, Dustin Schwenk, Alane Suhr, Pete Walsh, Dirk Groeneveld, Luca Soldaini, Sameer Singh, et al. What's in my big data? *arXiv preprint arXiv:2310.20707*, 2023. 23

[41] Christian Ertler, Jerneja Mislej, Tobias Ollmann, Lorenzo Porzi, Gerhard Neuhold, and Yubin Kuang. The mapillary traffic sign dataset for detection and classification on a global scale. In *European Conference on Computer Vision*, pages 68–84. Springer, 2020. 22

[42] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *NeurIPS*, 2023. 1, 2, 21, 36

[43] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *IJCV*, 2024. 20

[44] Qiankun Gao, Chen Zhao, Yifan Sun, Teng Xi, Gang Zhang, Bernard Ghanem, and Jian Zhang. A unified continual learning framework with general parameter-efficient tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11483–11493, 2023. 25

[45] Saurabh Garg, Mehrdad Farajtabar, Hadi Pouransari, Raviteja Vemulapalli, Sachin Mehta, Oncel Tuzel, Vaishaal Shankar, and Fartash Faghri. Tic-clip: Continual training of clip models. In *ICLR*, 2024. 1, 2, 18, 19, 24, 25

[46] Evangelia Gogoulou, Timothée Lesort, Magnus Boman, and Joakim Nivre. A study of continual learning under language shift. *arXiv preprint arXiv:2311.01200*, 2023. 19

[47] Dipam Goswami, Yuyang Liu, Bartłomiej Twardowski, and Joost van de Weijer. Fecam: Exploiting the heterogeneity of class distributions in exemplar-free continual learning. *NeurIPS*, 2023. 20

[48] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. arxiv 2017. *arXiv preprint arXiv:1706.02677*, 2017. 29

[49] Sachin Goyal, Ananya Kumar, Sankalp Garg, Zico Kolter, and Aditi Raghunathan. Finetune like you pretrain: Improved finetuning of zero-shot vision models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19338–19347, 2023. 21, 26, 33

[50] Gregory Griffin, Alex Holub, and Pietro Perona. *Caltech-256 Object Category Dataset*. Mar 2007. 23

[51] Yanan Gu, Xu Yang, Kun Wei, and Cheng Deng. Not just selection, but exploration: Online class-incremental continual learning via dual view consistency. In *CVPR*, 2022. 20

[52] 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. 32

[53] Zhongrui Gui, Shuyang Sun, Runjia Li, Jianhao Yuan, Zhaochong An, Karsten Roth, Ameya Prabhu, and Philip Torr. knn-clip: Retrieval enables training-free segmentation on continually expanding large vocabularies. *arXiv preprint arXiv:2404.09447*, 2024. 19, 20

[54] Guy Hacohen and Daphna Weinshall. On the power of curriculum learning in training deep networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2535–2544. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/hacohen19a.html. 2, 22

[55] Alexander Hägele, Elie Bakouch, Atli Kosson, Loubna Ben Allal, Leandro Von Werra, and Martin Jaggi. Scaling laws and compute-optimal training beyond fixed training durations. *arXiv preprint arXiv:2405.18392*, 2024. 19

[56] Hasan Abed Al Kader Hammoud, Hani Itani, Fabio Pizzati, Philip Torr, Adel Bibi, and Bernard Ghanem. Synthclip: Are we ready for a fully synthetic clip training? *arXiv preprint arXiv:2402.01832*, 2024. 22

[57] Jinwei Han, Zhiwen Lin, Zhongyisun Sun, Yingguo Gao, Ke Yan, Shouhong Ding, Yuan Gao, and Gui-Song Xia. Anchor-based robust finetuning of vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26919–26928, 2024. 33

[58] Md Yousuf Harun and Christopher Kanan. Overcoming the stability gap in continual learning. *arXiv preprint arXiv:2306.01904*, 2023. 19, 20, 25

[59] Md Yousuf Harun, Jhair Gallardo, Tyler L Hayes, and Christopher Kanan. How efficient are today's continual learning algorithms? *CVPR-W*, 2023. 19

[60] Jinghan He, Haiyun Guo, Ming Tang, and Jinqiao Wang. Continual instruction tuning for large multimodal models. *arXiv preprint arXiv:2311.16206*, 2023. 20

[61] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019. 23

[62] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *ICCV*, 2021. 23

[63] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *CVPR*, 2021. 23

[64] Haikel Hichri. NWPU-RESISC45 Dataset with 12 classes. 9 2021. doi: 10.6084/m9.figshare.16674166.v1. URL https://figshare.com/articles/dataset/NWPU-RESISC45_Dataset_with_12_classes/16674166. 22

[65] Saihui Hou, Yushan Feng, and Zilei Wang. Vegfru: A domain-specific dataset for fine-grained visual categorization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 22

[66] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 831–839, 2019. 2, 24

[67] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In *International Joint Conference on Neural Networks*, number 1288, 2013. 23

[68] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9. 1, 25, 26, 27

[69] Hexiang Hu, Ozan Sener, Fei Sha, and Vladlen Koltun. Drinking from a firehose: Continual learning with web-scale natural language. *IEEE Transactions on Pattern Analysis and Machine Intelligence, to appear*, 2022. 2, 24

[70] Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024. 19

[71] David P. Hughes and Marcel Salathé . An open access repository of images on plant health to enable the development of mobile disease diagnostics through machine learning and crowdsourcing. *CoRR*, abs/1511.08060, 2015. URL http://arxiv.org/abs/1511.08060. 22

[72] Adam Ibrahim, Benjamin Thérien, Kshitij Gupta, Mats L Richter, Quentin Anthony, Timothée Lesort, Eugene Belilovsky, and Irina Rish. Simple and scalable strategies to continually pre-train large language models. *arXiv preprint arXiv:2403.08763*, 2024. 3, 18, 19, 25, 26, 29, 30, 36

[73] Wisdom Oluchi Ikezogwo, Mehmet Saygin Seyfioglu, Fatemeh Ghezloo, Dylan Stefan Chan Geva, Fatwir Sheikh Mohammed, Pavan Kumar Anand, Ranjay Krishna, and Linda Shapiro. Quilt-1m: One million image-text pairs for histopathology. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=OL2JQoO0kq. 20, 22

[74] Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. Patching open-vocabulary models by interpolating weights. In *NeurIPS*, 2022. 1, 19, 20, 25, 27, 28, 33

[75] iNaturalist 2021 competition dataset. iNaturalist 2021 competition dataset. https://github.com/visipedia/inat_comp/tree/master/2021, 2021. 20, 22

[76] Phillip Isola, Joseph J. Lim, and Edward H. Adelson. Discovering states and transformations in image collections. In *CVPR*, 2015. 20, 22

[77] Joel Jang, Seonghyeon Ye, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, Stanley Jungkyu Choi, and Minjoon Seo. Towards continual knowledge learning of language models. *arXiv preprint arXiv:2110.03215*, 2021. 19

[78] Joel Jang, Seonghyeon Ye, Changho Lee, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, and Minjoon Seo. Temporalwiki: A lifelong benchmark for training and evaluating ever-evolving language models. *arXiv preprint arXiv:2204.14211*, 2022. 19

[79] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 22

[80] Aniruddha Kembhavi, Michael Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. *ArXiv*, abs/1603.07396, 2016. URL https://api.semanticscholar.org/CorpusID:2682274. 22

[81] Muhammad Uzair Khattak, Syed Talal Wasim, Muzammal Naseer, Salman Khan, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Self-regulating prompts: Foundational model adaptation without forgetting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15190–15200, 2023. 27

[82] 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. *PNAS*, 2017. 1, 3, 19, 25, 26

[83] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International conference on machine learning*, pages 5637–5664. PMLR, 2021. 20

[84] Dawid Jan Kopiczko, Tijmen Blankevoort, and Yuki M Asano. VeRA: Vector-based random matrix adaptation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=NjNfLdxr3A. 26, 27

[85] Jedrzej Kozal, Jan Wasilewski, Bartosz Krawczyk, and Michal Wozniak. Continual learning with weight interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4187–4195, 2024. 25

[86] Ivan Krasin, Tom Duerig, Neil Alldrin, Andreas Veit, Sami Abu-El-Haija, Serge Belongie, David Cai, Zheyun Feng, Vittorio Ferrari, Victor Gomes, Abhinav Gupta, Dhyanesh Narayanan, Chen Sun, Gal Chechik, and Kevin Murphy. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github.com/openimages*, 2016. 20, 22

[87] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. 23

[88] Alex Krizhevsky. One weird trick for parallelizing convolutional neural networks. *arXiv preprint arXiv:1404.5997*, 2014. 30

[89] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-100 (canadian institute for advanced research). URL http://www.cs.toronto.edu/~kriz/cifar.html. 22

[90] Fei-Fei Li, Marco Andreeto, Marc'Aurelio Ranzato, and Pietro Perona. Caltech 101, Apr 2022. 23

[91] Haoran Li, Jingfeng Wu, and Vladimir Braverman. Fixed design analysis of regularization-based continual learning. In *Conference on Lifelong Learning Agents*, pages 513–533. PMLR, 2023. 25

[92] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023. 20

[93] Victor Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Y Zou. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. *Advances in Neural Information Processing Systems*, 35:17612–17625, 2022. 33

[94] Yan-Shuo Liang and Wu-Jun Li. Inflora: Interference-free low-rank adaptation for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23638–23647, 2024. 25

[95] Peiyuan Liao, Xiuyu Li, Xihui Liu, and Kurt Keutzer. The artbench dataset: Benchmarking generative models with artworks. *arXiv preprint arXiv:2206.11404*, 2022. 20, 22

[96] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll'a r, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. URL http://arxiv.org/abs/1405.0312. 23

[97] Zhiqiu Lin, Jia Shi, Deepak Pathak, and Deva Ramanan. The clear benchmark: Continual learning on real-world imagery. In *NeurIPS*, 2021. 18, 19

[98] Zhiqiu Lin, Deepak Pathak, Yu-Xiong Wang, Deva Ramanan, and Shu Kong. Continual learning with evolving class ontologies. *Advances in Neural Information Processing Systems*, 35:7671–7684, 2022. 18

[99] Adam Liska, Tomas Kocisky, Elena Gribovskaya, Tayfun Terzi, Eren Sezener, Devang Agrawal, D'Autume Cyprien De Masson, Tim Scholtes, Manzil Zaheer, Susannah Young, et al. Streamingqa: A benchmark for adaptation to new knowledge over time in question answering models. In *International Conference on Machine Learning*, pages 13604–13622. PMLR, 2022. 18, 19

[100] Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. *arXiv preprint arXiv:2402.09353*, 2024. 26, 27

[101] Zhuang Liu and Kaiming He. A decade's battle on dataset bias: Are we there yet? *arXiv preprint arXiv:2403.08632*, 2024. 35

[102] Vincenzo Lomonaco and Davide Maltoni. Core50: a new dataset and benchmark for continuous object recognition. In *Conference on robot learning*, pages 17–26. PMLR, 2017. 19

[103] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 21

[104] Yilin Lyu, Liyuan Wang, Xingxing Zhang, Zicheng Sun, Hang Su, Jun Zhu, and Liping Jing. Overcoming recency bias of normalization statistics in continual learning: Balance and adaptation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=Ph65E1bE6A. 30

[105] Simone Magistri, Joost van de Weijer, Andew D Bagdanov, et al. An empirical analysis of forgetting in pre-trained models with incremental low-rank updates. *arXiv preprint arXiv:2405.18069*, 2024. 20, 25

[106] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 22

[107] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 7765–7773. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00810. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Mallya_PackNet_Adding_Multiple_CVPR_2018_paper.html. 2, 24

[108] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 2, 24

[109] Daniel Marczak, Bartłomiej Twardowski, Tomasz Trzciński, and Sebastian Cygert. Magmax: Leveraging model merging for seamless continual learning. *arXiv preprint arXiv:2407.06322*, 2024. 25

[110] Imad Eddine Marouf, Subhankar Roy, Enzo Tartaglione, and Stéphane Lathuilière. Weighted ensemble models are strong continual learners. *arXiv preprint arXiv:2312.08977*, 2023. 19, 25

[111] Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017. 20, 22, 23

[112] Mark D McDonnell, Dong Gong, Amin Parveneh, Ehsan Abbasnejad, and Anton van den Hengel. Ranpac: Random projections and pre-trained models for continual learning. In *NeurIPS*, 2023. 19, 20

[113] Sachit Menon and Carl Vondrick. Visual classification via description from large language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=jlAjNL8z5cs. 20

[114] Otniel-Bogdan Mercea, Alexey Gritsenko, Cordelia Schmid, and Anurag Arnab. Time-, memory-and parameter-efficient visual adaptation. *arXiv preprint arXiv:2402.02887*, 2024. 2

[115] Umberto Michieli and Pietro Zanuttigh. Incremental learning techniques for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*, pages 0–0, 2019. 19

[116] Sudhanshu Mittal, Maxim Tatarchenko, Özgün Çiçek, and Thomas Brox. Parting with illusions about deep active learning. *arXiv preprint arXiv:1912.05361*, 2019. 34

[117] Jishnu Mukhoti, Yarin Gal, Philip HS Torr, and Puneet K Dokania. Fine-tuning can cripple your foundation model; preserving features may be the solution. *arXiv preprint arXiv:2308.13320*, 2023. 19, 25, 27

[118] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 4. Granada, 2011. 23

[119] Thao Nguyen, Gabriel Ilharco, Mitchell Wortsman, Sewoong Oh, and Ludwig Schmidt. Quality not quantity: On the interaction between dataset design and robustness of clip. *Advances in Neural Information Processing Systems*, 35:21455–21469, 2022. 36

[120] Thao Nguyen, Matthew Wallingford, Sebastin Santy, Wei-Chiu Ma, Sewoong Oh, Ludwig Schmidt, Pang Wei Koh, and Ranjay Krishna. Multilingual diversity improves vision-language representations. *arXiv preprint arXiv:2405.16915*, 2024. 36

[121] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics and Image Processing*, pages 722–729, 2008. doi: 10.1109/ICVGIP.2008.47. 20, 22

[122] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. The oxford-iiit pet dataset. URL https://www.robots.ox.ac.uk/~vgg/data/pets/. 23

[123] Jupinder Parmar, Sanjev Satheesh, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Reuse, don't retrain: A recipe for continued pretraining of language models. *arXiv preprint arXiv:2407.07263*, 2024. 19

[124] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 21

[125] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 20, 22, 23

[126] Lukás Picek, Milan Sulc, Jirí Matas, and Jacob Heilmann-Clausen. Overview of fungiclef 2022: Fungi recognition as an open set classification problem. In Guglielmo Faggioli, Nicola Ferro, Allan Hanbury, and Martin Potthast, editors, *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022*, volume 3180 of *CEUR Workshop Proceedings*, pages 1970–1981. CEUR-WS.org, 2022. URL https://ceur-ws.org/Vol-3180/paper-157.pdf. 20, 22

[127] Lukáš Picek, Milan Šulc, Jiří Matas, Thomas S Jeppesen, Jacob Heilmann-Clausen, Thomas Læssøe, and Tobias Frøslev. Danish fungi 2020-not just another image recognition dataset. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1525–1535, 2022. 22

[128] Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015. 23

[129] Angéline Pouget, Lucas Beyer, Emanuele Bugliarello, Xiao Wang, Andreas Peter Steiner, Xiaohua Zhai, and Ibrahim Alabdulmohsin. No filter: Cultural and socioeconomic diversityin contrastive vision-language models. *arXiv preprint arXiv:2405.13777*, 2024. 36

[130] Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. Gdumb: A simple approach that questions our progress in continual learning. In *ECCV*, 2020. 19

[131] Ameya Prabhu, Zhipeng Cai, Puneet Dokania, Philip Torr, Vladlen Koltun, and Ozan Sener. Online continual learning without the storage constraint. *arXiv preprint arXiv:2305.09253*, 2023. 1, 2, 18, 19, 20, 24

[132] Ameya Prabhu, Hasan Abed Al Kader Hammoud, Puneet Dokania, Philip HS Torr, Ser-Nam Lim, Bernard Ghanem, and Adel Bibi. Computationally budgeted continual learning: What does matter? In *CVPR*, 2023. 1, 2, 19, 24, 25, 26, 36

[133] Ameya Prabhu, Hasan Abed Al Kader Hammoud, Ser-Nam Lim, Bernard Ghanem, Philip HS Torr, and Adel Bibi. From categories to classifier: Name-only continual learning by exploring the web. *arXiv preprint arXiv:2311.11293*, 2023. 20

[134] Ameya Prabhu, Shiven Sinha, Ponnurangam Kumaraguru, Philip HS Torr, Ozan Sener, and Puneet K Dokania. Randumb: A simple approach that questions the efficacy of continual representation learning. *arXiv preprint arXiv:2402.08823*, 2024. 19, 26

[135] Sarah Pratt, Ian Covert, Rosanne Liu, and Ali Farhadi. What does a platypus look like? generating customized prompts for zero-shot image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15691–15701, October 2023. 20

[136] Gao Qiankun, Zhao Chen, Sun Yifan, Xi Teng, Zhang Gang, Ghanem Bernard, and Zhang Jian. A unified continual learning framework with general parameter-efficient tuning. *ICCV*, 2023. 19

[137] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 21, 22, 23, 33

[138] Colin Raffel. A call to build models like we build opensource software, 2021. *https://colinraffel.com/blog/a-call-to-build-models-like-we-build-open-source-software.html*. 19

[139] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020. 23

[140] Vinay Venkatesh Ramasesh, Aitor Lewkowycz, and Ethan Dyer. Effect of scale on catastrophic forgetting in neural networks. In *International Conference on Learning Representations*, 2021. 4, 32

[141] Alexandre Ramé, Nino Vieillard, Léonard Hussenot, Robert Dadashi, Geoffrey Cideron, Olivier Bachem, and Johan Ferret. Warm: On the benefits of weight averaged reward models. *arXiv preprint arXiv:2401.12187*, 2024. 25

[142] Anton Razzhigaev, Arseniy Shakhmatov, Anastasia Maltseva, Vladimir Arkhipkin, Igor Pavlov, Ilya Ryabov, Angelina Kuts, Alexander Panchenko, Andrey Kuznetsov, and Denis Dimitrov. Kandinsky: an improved text-to-image synthesis with image prior and latent diffusion. *arXiv preprint arXiv:2310.03502*, 2023. 20, 24

[143] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. *Advances in neural information processing systems*, 30, 2017. 2, 24

[144] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, 2017. 2, 24

[145] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pages 5389–5400, 2019. 23

[146] Jonathan Roberts, Kai Han, and Samuel Albanie. Satin: A multi-task metadataset for classifying satellite imagery using vision-language models. *arXiv preprint arXiv:2304.11619*, 2023. 22

[147] William A Gaviria Rojas, Sudnya Diamos, Keertan Ranjan Kini, David Kanter, Vijay Janapa Reddi, and Cody Coleman. The dollar street dataset: Images representing the geographic and socioeconomic diversity of the world. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL https://openreview.net/forum?id=qnfYsave0U4. 22

[148] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022. 20, 24

[149] Karsten Roth, Timo Milbich, and Bjorn Ommer. Pads: Policy-adapted sampling for visual similarity learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2, 22

[150] Karsten Roth, Jae Myung Kim, A. Sophia Koepke, Oriol Vinyals, Cordelia Schmid, and Zeynep Akata. Waffling around for performance: Visual classification with random words and broad concepts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15746–15757, October 2023. 20

[151] Karsten Roth, Lukas Thede, A. Sophia Koepke, Oriol Vinyals, Olivier J Henaff, and Zeynep Akata. Fantastic gains and where to find them: On the existence and prospect of general knowledge transfer between any pretrained model. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=m50eKHCttz. 27

[152] Simon Schrodi, David T Hoffmann, Max Argus, Volker Fischer, and Thomas Brox. Two effects, one trigger: On the modality gap, object bias, and information imbalance in contrastive vision-language representation learning. *arXiv preprint arXiv:2404.07983*, 2024. 33

[153] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 21

[154] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018. 36

[155] Guangyuan Shi, Jiaxin Chen, Wenlong Zhang, Li-Ming Zhan, and Xiao-Ming Wu. Overcoming catastrophic forgetting in incremental few-shot learning by finding flat minima. *Advances in neural information processing systems*, 34:6747–6761, 2021. 30

[156] Peiyang Shi, Michael C Welle, Mårten Björkman, and Danica Kragic. Towards understanding the modality gap in clip. In *ICLR 2023 Workshop on Multimodal Representation Learning: Perks and Pitfalls*, 2023. 33

[157] Yang Shu, Xingzhuo Guo, Jialong Wu, Ximei Wang, Jianmin Wang, and Mingsheng Long. CLIPood: Generalizing CLIP to out-of-distributions. 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 31716–31731. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/shu23a.html. 20

[158] Samarth Sinha, Animesh Garg, and Hugo Larochelle. Curriculum by smoothing. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21653–21664. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/f6a673f09493afcd8b129a0bcf1cd5bc-Paper.pdf. 2, 22

[159] James Seale Smith, Paola Cascante-Bonilla, Assaf Arbelle, Donghyun Kim, Rameswar Panda, David Cox, Diyi Yang, Zsolt Kira, Rogerio Feris, and Leonid Karlinsky. Construct-vl: Data-free continual structured vl concepts learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14994–15004, 2023. 25

[160] James Seale Smith, Yen-Chang Hsu, Lingyu Zhang, Ting Hua, Zsolt Kira, Yilin Shen, and Hongxia Jin. Continual diffusion: Continual customization of text-to-image diffusion with c-lora. *arXiv preprint arXiv:2304.06027*, 2023. 25

[161] James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *CVPR*, 2023. 19

[162] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Computer Vision and Pattern Recognition (CVPR)*, 2016. 20, 23

[163] Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. Curriculum learning: A survey. *Int. J. Comput. Vision*, 130(6):1526–1565, jun 2022. ISSN 0920-5691. doi: 10.1007/s11263-022-01611-x. URL https://doi.org/10.1007/s11263-022-01611-x. 2, 22

[164] Tejas Srinivasan, Ting-Yun Chang, Leticia Pinto Alva, Georgios Chochlakis, Mohammad Rostami, and Jesse Thomason. Climb: A continual learning benchmark for vision-and-language tasks. *Advances in Neural Information Processing Systems*, 35:29440–29453, 2022. 19

[165] Zafir Stojanovski, Karsten Roth, and Zeynep Akata. Momentum-based weight interpolation of strong zero-shot models for continual learning. *arXiv preprint arXiv:2211.03186*, 2022. 25, 27, 30

[166] Hai-Long Sun, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Pilot: A pre-trained model-based continual learning toolbox. *arXiv preprint arXiv:2309.07117*, 2023. 19

[167] Yi Sun, Xin Xu, Jian Li, Guanglei Xie, Yifei Shi, and Qiang Fang. Continual learning through networks splitting and merging with dreaming-meta-weighted model fusion. *arXiv preprint arXiv:2312.07082*, 2023. 20

[168] Lukas Thede, Karsten Roth, Olivier J Hénaff, Matthias Bethge, and Zeynep Akata. Reflecting on the state of rehearsal-free continual learning with pretrained models. *arXiv preprint arXiv:2406.09384*, 2024. 26

[169] Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528, 2011. doi: 10.1109/CVPR.2011.5995347. 35

[170] Vishaal Udandarao. Understanding and fixing the modality gap in vision-language models. *Master's thesis, University of Cambridge*, 2022. 33

[171] Vishaal Udandarao, Max F Burg, Samuel Albanie, and Matthias Bethge. Visual data-type understanding does not emerge from scaling vision-language models. In *The Twelfth International Conference on Learning Representations*, 2023. 33

[172] Vishaal Udandarao, Ankush Gupta, and Samuel Albanie. Sus-x: Training-free name-only transfer of vision-language models. In *ICCV*, 2023. 20

[173] Vishaal Udandarao, Ameya Prabhu, Adhiraj Ghosh, Yash Sharma, Philip HS Torr, Adel Bibi, Samuel Albanie, and Matthias Bethge. No "zero-shot"' without exponential data: Pretraining concept frequency determines multimodal model performance. *arXiv preprint arXiv:2404.04125*, 2024. 2, 23

[174] Gido M van de Ven and Andreas S Tolias. Three scenarios for continual learning. In *NeurIPS-W*, 2018. 18

[175] Tom Veniat, Ludovic Denoyer, and Marc'Aurelio Ranzato. Efficient continual learning with modular networks and task-driven priors. *arXiv preprint arXiv:2012.12631*, 2020. 19

[176] Eli Verwimp, Shai Ben-David, Matthias Bethge, Andrea Cossu, Alexander Gepperth, Tyler L Hayes, Eyke Hüllermeier, Christopher Kanan, Dhireesha Kudithipudi, Christoph H Lampert, et al. Continual learning: Applications and the road forward. *arXiv preprint arXiv:2311.11908*, 2023. 19

[177] Eli Verwimp, Kuo Yang, Sarah Parisot, Lanqing Hong, Steven McDonagh, Eduardo Pérez-Pellitero, Matthias De Lange, and Tinne Tuytelaars. Clad: A realistic continual learning benchmark for autonomous driving. *Neural Networks*, 161:659–669, 2023. 19

[178] Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc Le, et al. Freshllms: Refreshing large language models with search engine augmentation. *arXiv preprint arXiv:2310.03214*, 2023. 19

[179] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. Caltech-ucsd birds-200-2011 (cub-200-2011). Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 20, 22

[180] Fanqi Wan, Xinting Huang, Deng Cai, Xiaojun Quan, Wei Bi, and Shuming Shi. Knowledge fusion of large language models. *arXiv preprint arXiv:2401.10491*, 2024. 20

[181] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, pages 10506–10518, 2019. 23

[182] Jianren Wang, Xin Wang, Yue Shang-Guan, and Abhinav Gupta. Wanderlust: Online continual object detection in the real world. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10829–10838, 2021. 18, 19

[183] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 2, 24

[184] Peng Wang, Zexi Li, Ningyu Zhang, Ziwen Xu, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. Wise: Rethinking the knowledge memory for lifelong model editing of large language models. *arXiv preprint arXiv:2405.14768*, 2024. 20

[185] Xiao Wang, Yuansen Zhang, Tianze Chen, Songyang Gao, Senjie Jin, Xianjun Yang, Zhiheng Xi, Rui Zheng, Yicheng Zou, Tao Gui, et al. Trace: A comprehensive benchmark for continual learning in large language models. *arXiv preprint arXiv:2310.06762*, 2023. 19

[186] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision (ECCV)*, 2022. 19

[187] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 19

[188] Yeming Wen, Dustin Tran, and Jimmy Ba. Batchensemble: an alternative approach to efficient ensemble and lifelong learning. *arXiv preprint arXiv:2002.06715*, 2020. 19

[189] Martin Wistuba, Prabhu Teja Sivaprasad, Lukas Balles, and Giovanni Zappella. Continual learning with low rank adaptation. *arXiv preprint arXiv:2311.17601*, 2023. 20, 25

[190] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7959–7971, June 2022. 25, 27, 33

[191] Tongtong Wu, Linhao Luo, Yuan-Fang Li, Shirui Pan, Thuy-Trang Vu, and Gholamreza Haffari. Continual learning for large language models: A survey. *arXiv preprint arXiv:2402.01364*, 2024. 19

[192] Xiaoxia Wu, Ethan Dyer, and Behnam Neyshabur. When do curricula work? In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=tW4QEInpni. 34

[193] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *CVPR*, 2019. 2, 24

[194] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 23

[195] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3485–3492, 2010. doi: 10.1109/CVPR.2010.5539970. 21, 22

[196] Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-merging: Resolving interference when merging models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=xtaX3WyCj1. 27

[197] Çağatay Yıldız, Nishaanth Kanna Ravichandran, Prishruit Punia, Matthias Bethge, and Beyza Ermis. Investigating continual pretraining in large language models: Insights and implications. *arXiv preprint arXiv:2402.17400*, 2024. 2, 19, 23

[198] Aron Yu and Kristen Grauman. Fine-grained visual comparisons with local learning. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 192–199, 2014. doi: 10.1109/CVPR.2014.32. 22

[199] Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Ping Hu, Dong Wang, Huchuan Lu, and You He. Boosting continual learning of vision-language models via mixture-of-experts adapters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23219–23230, June 2024. 2, 24

[200] Qiying Yu, Quan Sun, Xiaosong Zhang, Yufeng Cui, Fan Zhang, Yue Cao, Xinlong Wang, and Jingjing Liu. Capsfusion: Rethinking image-text data at scale. *arXiv preprint arXiv:2310.20550*, 2023. 20

[201] Zhenghang Yuan, Lichao Mou, Qi Wang, and Xiao Xiang Zhu. From easy to hard: Learning language-guided curriculum for visual question answering on remote sensing data. *IEEE Transactions on Geoscience and Remote Sensing*, 60, 2022. 2, 22

[202] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *ICML*, 2017. 1, 3, 25, 26

[203] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019. 22

[204] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12104–12113, June 2022. 3, 19, 30

[205] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18123–18133, 2022. 21

[206] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023. 35

[207] Chenshuang Zhang, Fei Pan, Junmo Kim, In So Kweon, and Chengzhi Mao. Imagenet-d: Benchmarking neural network robustness on diffusion synthetic object. *CVPR*, 2024. 23

[208] Gengwei Zhang, Liyuan Wang, Guoliang Kang, Ling Chen, and Yunchao Wei. Slca++: Unleash the power of sequential fine-tuning for continual learning with pre-training. *arXiv preprint arXiv:2408.08295*, 2024. 19

[209] Renrui Zhang, Rongyao Fang, Peng Gao, Wei Zhang, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv preprint arXiv:2111.03930*, 2021. 20

[210] Yuanhang Zhang, Zhidi Lin, Yiyong Sun, Feng Yin, and Carsten Fritsche. Regularization-based efficient continual learning in deep state-space models. *arXiv preprint arXiv:2403.10123*, 2024. 25

[211] Zihan Zhang, Meng Fang, Ling Chen, and Mohammad-Reza Namazi-Rad. Citb: A benchmark for continual instruction tuning. *arXiv preprint arXiv:2310.14510*, 2023. 20

[212] Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. Galore: Memory-efficient llm training by gradient low-rank projection. *arXiv preprint arXiv:2403.03507*, 2024. 3, 25, 26

[213] Zangwei Zheng, Mingyuan Ma, Kai Wang, Ziheng Qin, Xiangyu Yue, and Yang You. Preventing zero-shot transfer degradation in continual learning of vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19125–19136, 2023. 19

[214] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 22

[215] Da-Wei Zhou, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need. *arXiv preprint arXiv:2303.07338*, 2023. 19

[216] Da-Wei Zhou, Hai-Long Sun, Jingyi Ning, Han-Jia Ye, and De-Chuan Zhan. Continual learning with pre-trained models: A survey. *arXiv preprint arXiv:2401.16386*, 2024. 19

[217] Da-Wei Zhou, Hai-Long Sun, Han-Jia Ye, and De-Chuan Zhan. Expandable subspace ensemble for pre-trained model-based class-incremental learning. *arXiv preprint arXiv:2403.12030*, 2024. 19

[218] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. 20

[219] Weixun Zhou, Shawn Newsam, Congmin Li, and Zhenfeng Shao. Patternnet: A benchmark dataset for performance evaluation of remote sensing image retrieval. *ISPRS Journal of Photogrammetry and Remote Sensing*, 145:197–209, November 2018. ISSN 0924-2716. doi: 10.1016/j.isprsjprs.2018.01.004. URL http://dx.doi.org/10.1016/j.isprsjprs.2018.01.004. 22

[220] Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, Quoc V Le, James Laudon, et al. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35:7103–7114, 2022. 19

[221] Zhen Zhu, Weijie Lyu, Yao Xiao, and Derek Hoiem. Continual learning in open-vocabulary classification with complementary memory systems. *arXiv preprint arXiv:2307.01430*, 2023. 20

5 Limitations and Future Work

Limitations. In this work, our aim was to create a meaningful benchmark, provide practical guidelines, and offer insights into various multimodal continual pretraining scenarios. We focused on *continual, controlled, minor* model updates. We developed `FoMo-in-Flux` to include many publicly accessible datasets covering a wide range of potential adaptation sub-domains. However, our findings on knowledge accumulation $\mathcal{A}_{KA}$ and zero-shot retention $\mathcal{A}_{ZS}$ are tied to our chosen adaptation and evaluation datasets. Consequently, though unlikely, various sub-domains relevant for future applications might not be sufficiently covered. Additionally, our methods were based off of default hyperparameter ranges from original publications (`LoRA`, `VeRA`, `DoRA`, `BitFit`, `LNFit`, `FS-Merge`, `EMA-Merge`) or continual learning repositories (`mammoth` [14]). While we tested the validity of each method and the chosen hyperparameters to elicit meaningful finetuning responses on respective single datasets, it overall means that our conclusions rely on the optimality of these provided hyperparameter ranges.

Future Work. Our benchmark and findings provide a crucial starting point reference for further research into continual multimodal pretraining. We sketch a few important and immediate future research directions:

- **(Meta-) Learning Rate Schedules and Beyond:** Our experiments show the importance of learning rate schedules (and meta-variants) designed for longer horizon continual (minor) model updates. We used a default cosine learning rate schedule and one infinite learning rate schedule (rsqrt), along with five meta-schedule variants, but our results showcase that there is a lot of potential in further exploring infinite schedules, as well as extensions into task- and order-conditioned learning rate schedules to allow for continual model pretraining and model updates.
- **Further Scaling Up Compute and Models:** We studied continual learning under realistic constraints (MAFs), with compute budgets derived from `DataComp-small`. Investigating other computational budgets including over-training, and extending budgets to be potentially task-order dependent could have practical relevance. Extending our insights to even larger model scales (ViT-bigG/14 and beyond) can offer further practical guidance. We have investigated the effect of model and compute scaling (see fig. 14) independently and to a first degree, however we believe there is a lot more exciting future work to be done.
- **Text-to-Image Generative Models:** Besides vision-language representation learning, `FoMo-in-Flux` can be used to study continuous minor updates of text-to-image generative models (such as generative diffusion models) on a fine-grained class and concept level, leveraging its diverse set of captions and information about respective image concepts.
- **Optimal Training Mixtures:** Our results indicate that knowledge retention during minor updates depends heavily on replaying data from previous tasks, guided towards "iid"-fying the learning task. This process helps prevent knowledge forgetting related to pretraining. However, there is room to better understand optimal training mixtures within limited compute budgets. Finding the best ways to allocate FLOPs and memory for replay on large pretraining data is crucial.

Broader Impact. Better continual model pretraining and the ability to minimize the need for large-scale model retraining can have significant impact on cost, compute and consequently environmental footprint. By encouraging research into extending the re-usability of large-scale pretrained models before a major continual model update or even full retraining from scratch is needed, we believe our work will lead to more economical and ecological utilization of foundation models. We do not believe that there are any immediate negative societal consequences as a result of this work.

6 Categorizing Continual Pretraining: A Versioning Perspective

Traditional continual learning has been categorized into class-, domain-, and task-incremental settings [174]. However, continual pretraining benchmarks do not fit these categories, as they exhibit high-overlaps in captions as opposed to disjoint classes [72, 12, 98], and time-varying gradual class and domain shifts [45, 97, 18, 131, 99, 182]. Similarly, continual learning strategies are typically

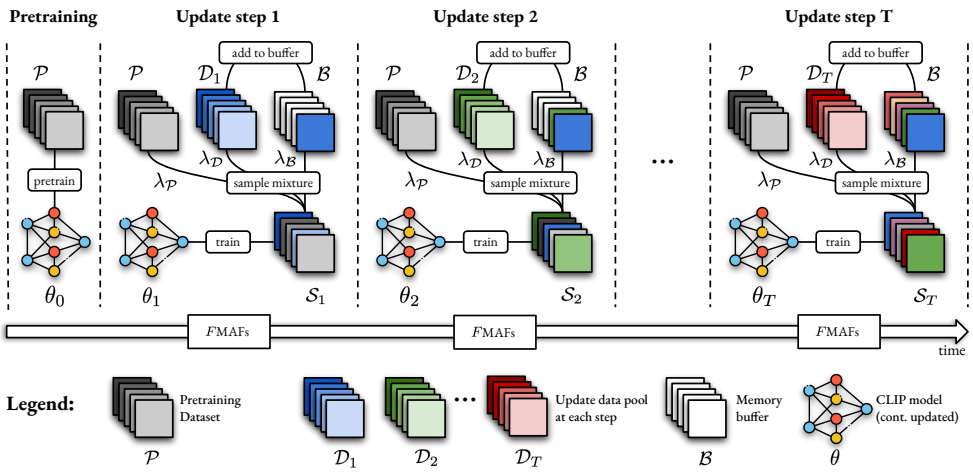


Figure 3: **FoMo-In-Flux pipeline.** *(Pretraining)* We start from pretrained CLIP θ_0 and its pretraining pool $\mathcal{P}$. *(Update steps)* At each step t, we sample training instances $\mathcal{S}_t$ from $\mathcal{P}$, current update pool $\mathcal{D}_t$, and memory buffer $\mathcal{B}$ (containing all past $\mathcal{D}_t$s), and train for a fixed compute budget (F MAFs).

Benchmark	# Samples	# Tasks	Ordering	Domains	Update Style	Multi-modal	Zero-Shot Retention	Compute-Bound	Data-Mixtures	Real World Stream Variants
CORe50 [102]	165K	9	Class-/Data-Inc	Objects	Major	×	×	×	×	×
Split-ImageNet [188]	1.2M	10	Class-Inc	Web Images	Major	×	×	×	×	×
PTM-Adaptation [166]	30K-100K	5-20	Class-Inc	Web Images	Minor	×	×	×	×	×
CLAD [177]	23K	∼2000	Time-Inc	Synthetic	Patch	×	×	×	×	×
OAK [182]	326K	∼2000	Time-Inc	Egocentric	Patch	×	×	×	×	×
Inc-PASCAL [115]	11K	2-6	Class-Inc	Web Images	Major	×	×	×	×	×
Inc-ADE20K [19]	20K	2-6	Class-Inc	Scene Parsing	Major	×	×	×	×	×
StreamingQA [99]	100K	6	Time-Inc	Text	Major	×	×	×	×	×
TemporalWiki [78]	32M	4	Time-Inc	Text	Major	×	✓	×	×	×
CKL [77]	30K	2	Task-Inc	Text	Minor	×	✓	×	×	×
CTrL [175]	300K	100	Task-Inc	Objects	Major	×	×	×	×	×
CLEAR [97]	7.8M	10	Time-Inc	Web Images	Minor	×	×	×	×	×
ImageNet2K [132]	1.2M	20-200	Class-/Data-Inc	Web Images	Major	×	×	✓	×	×
Offline-CGLM [132]	500K	20-200	Time-Inc	Web Images	Major	×	×	✓	×	×
In1K-P365-LT [58]	62K	5	Class-/Data-Inc	Web Images	Minor	×	×	✓	×	×
NEVIS [12]	8M	79	Task-Inc	Mixed	Major	×	×	✓	×	×
CLOC [18]	39M	39M	Time-Inc	Geolocation	Patch	×	×	✓	×	×
CGLM [131]	500K	500K	Time-Inc	Landmarks	Patch	×	×	✓	×	×
CLiMB [164]	1.3M	4	Task-Inc	Mixed	Minor	✓	✓	×	×	×
MTIL [213]	250K	5-20	Class-Inc	Mixed	Minor	✓	×	×	×	×
Ctl-M2D2 [197]	6.6B	160	Domain-Inc	Text	Minor	×	✓	×	×	×
TiC-DataComp [45]	100M/1B/12B	6	Time-Inc	Web Images	Major	✓	✓	✓	×	×
`FoMo-in-Flux` (**Ours**)	2.5M	20+	Data-Centric	Mixed	Minor	✓	✓	✓	✓	✓

Table 1: `FoMo-in-Flux` **comparison to existing benchmarks** used in continual learning/pretraining studies: it features large timesteps, data-centric streams, provides image-text pairs, a minor-update style, measures zero-shot retention, and is compute-constrained.

grouped [32, 130] into replay [22, 17], regularization [117, 82, 21], and parameter-isolation methods [217, 3, 220], with more recent additions like prompt-tuning [186, 187, 161, 136], fixed-representation [112, 215, 134], and model-mixture methods [110, 74] (see [216] for a survey). However, continual foundation model updates are dominated by replay [132, 45], parameter-efficient finetuning [58] and retrieval-augmented methods [178, 131, 53], as traditional methods do not help under computational constraints [59, 176, 131] and do not outperform simple baselines [134, 112, 132, 208]. Hence, we provide a new categorization suitable for continual pretraining literature.

Our categorization for continual pretraining literature is inspired by the semantic software versioning framework [138]. We believe that different scopes of updates require distinct strategies, indicating that no single solution fits all continual pretraining scenarios (see [191] for a survey, and table 1 for an overview of related benchmarks under the semantic versioning umbrella). We believe foundation models require distinct update strategies, similar to major, minor, and patch updates in software versioning:

Major Updates. Large-scale continual pretraining over extensive compute, data, and time resources that substantially alter overall performance. Methods focusing on significant updates [45, 72, 46] consistently employ continual fine-tuning of the model, which has been found to be the primary strategy through extensive comparisons with other works [45, 185, 132, 24]. Currently explored topics include continual LR scheduling [55, 72, 204, 123, 70] to minimize the stability gap [33].

Patch Updates. Frequent but minor, targeted updates in which continual fine-tuning leads to poor zero-shot capability retention with little new knowledge gained. These are best managed by continual knowledge editing [25, 184] or sample-wise updates using a fixed backbone [131, 221, 53, 112, 47].

Minor Updates. Adaptations to whole subdomains and general concepts out of scope for knowledge edits, but without the need for large-scale major updates. Some examples are: updating specific parts of a model with LoRA [58, 11, 105, 189], model merging [74, 167, 180], instruction tuning [60, 211, 23], incorporating expert knowledge on particular subdomains or specialized visual distribution shifts [83, 209, 172, 157, 113, 135, 150, 51, 218, 43, 133]). Real-world situations that might warrant a minor update include incorporating new tasks, such as visual reasoning over fine-grained object categories [8, 179, 75, 126, 121, 162], or new domains like sketches [27, 125], drawings [125, 95], or synthetic [16, 111] and medical imagery [73, 38]. Within our multimodal setup, these minor updates can also jointly involve new or infrequently encountered concepts [16, 111], s.a. aforementioned fine-grained expert knowledge, medical applications or new compositions [76].

7 FoMo-in-Flux

Captioning. As classification datasets lack image-caption pairs necessary for vision-language model pretraining, we provide captions for each image. More precisely, we introduce high-quality class-specific captions through three different methods: **(1)** A scalable two-stage captioning mechanism, which uses BLIP-2 [92] to generate general captions for each image and CapsFusion [200] (T5-XL) to merge and align captions with available information on ground-truth class names (c.f. fig. 4). **(2)** Procedural generation for a few specific datasets (such as Shapes3D [16] and DSprites [111]) using available dataset-specific information, such as image latents or descriptors (c.f. fig. 5), creating captions that for example contain information about the approximate location of the object, its orientation, size or shape. These captions are then adjusted at random based on captions generated by GPT-4 [4], with some being complete, and some only including the basic information. **(3)** Captions already provided alongside class labels as part of the dataset (e.g., OpenImages [86] or our *obscure* datasets, see section 7.1 and fig. 6).

Coverage. tables 2 and 3 highlight the diversity of domains and concepts covered in `FoMo-in-Flux`—ranging from diagrams and paintings, natural high- and low-resolution images, to synthetic and generative images, covering fine-grained and specialized domains, such as remote sensingand medical images. On the language side, concept and classes covered also vary noticeably, with e.g. ArtBench10 built around art-style and artist classification (as reflected in the captions), Quilt-1M introducing medical captions for histopathological image data, or our synthetic *Obscure* datasets introducing rare or fantastical, tail-end concepts with corresponding image captions. Dataset licenses are provided in both tables, all of which permit academic re-use. We provide references to original publications, most of which contain information how to download each dataset. To facilitate reproduction, our codebase comes with automatic download mechanisms for datasets where possible, and manual instructions otherwise.

7.1 Creating our *Obscure* Datasets

To improve diversity and increase the number of synthetic samples in our benchmark, we created the *Obscure Animals* and *Obscure Things* datasets using text-to-image models. An additional motivation for creating these datasets was to include classes that are systematically seen as obscure or not commonly occurring in the wild. The goal was both to mimic tail-ends of image and concept distributions, as well as the issue of more AI-generated content making its way into model training data, potentially misrepresenting some concepts (see *e.g.*, Fig 6). We first query ChatGPT to produce a set of 100 obscure animal names and 100 obscure object names. We then ask ChatGPT again to produce diverse prompts for each class name to be used as text prompts to feed into a text-to-image generation model.

We manually reviewed the quality of the text prompts for veracity and faithfulness to real world contexts. We then used the Kandinsky-2.1 [142], Stable Diffusion-2.1 [148], and Dreamlike-PhotoReal [1] text-to-image models to generate images for each classname using the curated text prompts. Finally, for each class we manually cleaned and filtered the images to ensure faithfulness. To create as clean a test set as possible, we conservatively removed an entire class if more than 30% of its images were ambiguous, unclear or outright unfaithful to the class—we used reference images

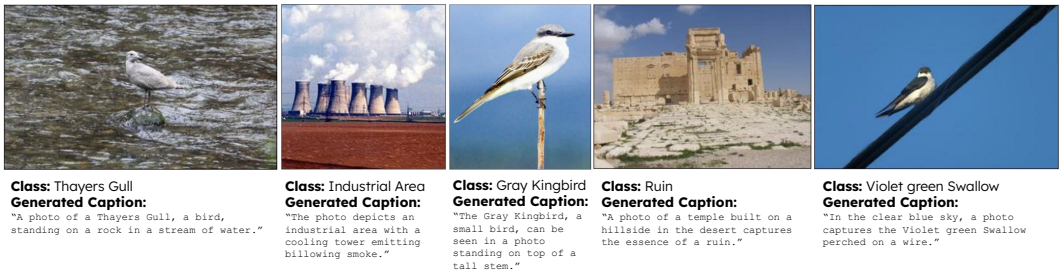


Figure 4: **Visualisation of generated captions.** We showcase some sample captions generated using our two-stage pipeline for fine-grained classes (birds from Birdsnap [8]), and general, coarse classes (taken from SUN397 [195]). The generated captions combine both image descriptions as well as important semantic class information.

from Google Images for this manual verification. Examples are visualized in Fig. 6. We provide download links here for obscure animals and obscure things.

8 Experimental Setup

We detail the default models, compute budgets, metrics, training schedules, and data mixtures used here.

Pretrained Models. We conducted our main experiments using a ViT-B-16 CLIP model pretrained on the LAION-2B dataset [153]. We also conducted some additional ablation experiments with a ViT-B-32 CLIP model (to understand the effects of different patch resolution) and ViT-S/16, ViT-L/14, ViT-H/14 and ViT-g/14 models. All our CLIP models are pretrained on LAION-2B, except for the ViT-S/16 model which is pretrained on the DataComp-1B dataset [42].

Default Continual Pretraining Settings. Unless otherwise specified, we always train each continual pretraining method for 20 update steps, T=20. Each update step comprises of continually training a CLIP model for a fixed number of samples derived by the computational budget outlined above. We fix the compute budgets per update step by taking the `DataComp-Small` total FLOP budget, i.e., 1.8×10^9 GFLOPs and dividing it by the total number of update steps.. By default, we use a random 2M subset of `LAION-400M` as our pretraining data pool $\mathcal{P}$ and operate with uniform mixing ratios $\{\lambda_{\mathcal{P}}=0.33,\lambda_{\mathcal{D}}=0.34,\lambda_{\mathcal{B}}=0.33\}$. For our reference upper bound performance, we train a CLIP model initialized from the same `open_clip` checkpoints jointly on all 41 adaptation datasets (with the samples randomly shuffled). We do this training for a compute budget of $T \times F$ MAFs, equivalent to the overall compute budget available for the entire continual pretraining process.

Training Details. We train all continual pretraining methods with the CLIP contrastive loss [137, 49] and learnable temperature τ, initialized to 0.01 (we provide ablations for the impact of τ initialization in section 10.4). We select the best-reported hyperparameters for each method from previous literature, only tuning the peak learning rate for each method. We use cosine-decay LR-scheduling with linear warmup of 10% (we study more LR-schedules in section 10.1), with an AdamW optimizer [103], a batch-size of 512 [103], and clip gradients with norm higher than 1. We run all experiments using PyTorch [124]. To truly study updates in both vision and language space, we update both encoders jointly (following Zhai et al. [205], we ablate this choice in section 10.3).

Metrics. From a model updating perspective, there are two main quantities of interest: the degree of *adaptation* to new data and the *retention* of pretraining knowledge. For all experiments, we therefore report two main metrics: Knowledge Accumulation ($\mathcal{A}_{KA}$), the average accuracy (or recall@5 for retrieval) over all concepts in the 41 adaptation datasets, and Zero-Shot Retention ($\mathcal{A}_{ZS}$), the zero-shot transfer accuracy (or recall@5 for retrieval) on the held-out set of 22 datasets.

Plotting Style. In most plots showing our main experimental result, we depict the zero-shot baseline as a black star and the joint training upper-bound as a golden star, with a dotted line connecting the two to approximate the joint training trajectory on the $\mathcal{A}_{KA}$-$\mathcal{A}_{ZS}$ plane. Every other trajectory depicts the training progression of individual experimental runs. Note that these trajectories always begin at the zero-shot baseline (black star).

Table 2: **Adaptation-only datasets** over various visual and textual domains like diagrams, paintings, natural, synthetic or generative images, remote sensing, art styles, traffic signs or textural data; with datasets from Radford et al. [137] with lower zero-shot performance, common transfer or aggregation benchmark datasets such as DomainNet [125] or VTAB [203] and specialized datasets like MVTec-AD [9].

Dataset	#Train	#Test	#Classes	Domain	License	Captions
Classification-based						
AI2Diagrams [80]	2720	681	15	diagrams	CC BY-SA	generated
ArtBench10 [95]	47531	11883	1870	paintings	Fair Use	generated
Birdsnap [8]	31905	7977	500	finegrained, natural	Unspecified, but academic usage	generated
Cifar100 [89]	50000	10000	100	natural	Unspecified, but academic usage	generated
CLEVR [79]	55931	13983	217	synthetic	CC BY 4.0	generated
CLRS [146]	13525	1475	25	remote sensing	Academic purposes [146]	generated
Country211 [137]	31650	21100	211	natural	various CC	generated
CUB200-2011 [179]	5994	5794	200	finegrained, natural	custom non-commercial	generated
DF20-mini [127]	32724	3637	179	finegrained, natural	custom non-commercial	generated
Dollarstreet [147]	13555	4103	1701	finegrained, natural	CC BY-SA 4.0	generated
Domainnet-Clipart [125]	33525	14604	345	illustrations	custom non-commercial	generated
Domainnet-Infograph [125]	36023	15582	345	diagrams	custom non-commercial	generated
Domainnet-Painting [125]	50416	21850	344	paintings	custom non-commmerical	generated
Domainnet-Sketch [125]	48212	20916	345	sketch	custom non-commercial	generated
Dsprites [111]	75000	25000	27	synthetic	Apache 2.0	procedural
DTD [28]	1880	1880	47	textural	custom non-commercial	generated
FGVCAircraft [106]	3334	3333	100	finegrained, natural	custom non-commercial	generated
Flowers102 [121]	6149	1020	102	finegrained, natural	Unspecified, but academic usage	generated
FRU92 [65]	55814	9200	92	finegrained, natural	Apache 2.0	generated
iNaturalist2021 [75]	125000	25000	2500	finegrained, natural	custom non-commercial	generated
Isicmelanoma [38]	2245	562	7	medical	CC-BY-NC	generated
Mitstates [76]	43002	10751	1959	finegrained, natural	Unspecified, but academic usage	generated
Mtsd [41]	59978	8737	227	finegrained, traffic signs	CC BY-NC-SA 4.0	generated
MVTec-AD (Base) [9]	2903	726	15	high-resolution, industrial	CC BY-NC-SA 4.0	generated
MVTec-AD (Faults) [9]	1380	345	88	high-resolution, industrial	CC BY-NC-SA 4.0	generated
ObjectNet [6]	40134	10000	313	natural	CC BY 4.0	generated
Obscure Animals	17000	4238	74	generative	MIT	custom
Obscure Things	19128	4758	84	generative	MIT	custom
OpenImages [86]	115333	8593	589	natural	Apache 2.0	available
PatternNet [219]	26600	3800	38	remote sensing	custom non-commercial	generated
Places365 [214]	120231	36499	365	natural	custom non-commercial	generated
Plantvillage [71]	43444	10681	38	finegrained, natural	CC0	generated
Quilt-1M [73]	95862	23966	157	medical	Academic purposes	available
Resisc45 [64]	18900	6300	45	remote sensing	Unspecified, but academic usage	generated
Shapes3D [16]	75000	25000	864	synthetic	Apache 2.0	procedural
SnakeCLEF2023 [126]	151031	14117	1599	finegrained, natural	custom non-commercial	generated
SUN397 [195]	15880	19850	397	natural	custom non-commercial	generated
SynthCLIP106 [56]	84800	13886	106	generative	CC BY-NC 4.0	generated
Veg200 [65]	61117	20000	200	finegrained, natural	Apache 2.0	generated
Zappos50k [198]	37829	9458	1847	finegrained, object	custom non-commerical	generated
Retrieval-based						
FSCOCO [27] (avg T2I/I2T R@5)	7105	1777	115	sketch	CC BY-NC 4.0	Available
Total	**1759782**	**453020**	**18449**			

8.1 Designing Data-Centric Task-Sequences

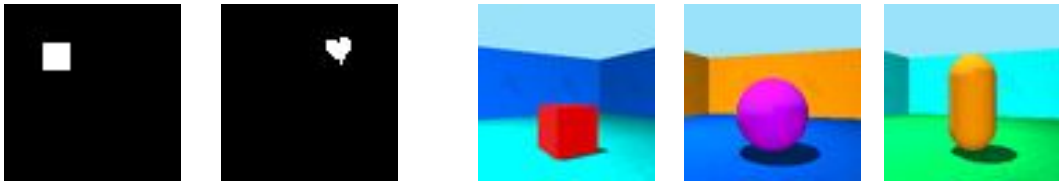
In addition to studying different pretraining sets $\mathcal{P}$ and data mixture ratios $(\lambda_{\mathcal{P}}, \lambda_{\mathcal{B}}, \lambda_{\mathcal{D}})$, we also investigate different realistic orderings by breaking down the `FoMo-in-Flux` datasets into individual concepts, which are then ordered according to a chosen criterion (including the option to study reverse orderings). This is visualized in Fig. 7. In order to do so, having a controlled set of image-caption pairs is critical, as it allows for well-defined and meaningful arrangement of concepts into sequences according to an ordering $\pi(\mathcal{C})$. Each ordering π divides the set of samples $\mathcal{D}$ into T disjoint subsets $\{\mathcal{D}_1, \ldots, \mathcal{D}_T\}$ of concepts $\mathcal{C}$ sampled without replacement, i.e. $\mathcal{C}_i \bigcap \mathcal{C}_j = \phi, \forall i, j$. We define and motivate six different orderings below:

1. Easy-To-Hard Ordering (`performance`) is motivated by curriculum learning [54, 149, 158, 163, 201], assuming users deploying their model to easier concepts and usecases first, with incremental movement towards to harder concepts.

Implementation. We approach the notion of "easy" vs. "hard" samples by ordering them according to base model performance. For each concept, we select 50 random image-text pairs and then randomly sample further 50 image-text pairs from the CC-3M dataset to represent random samples from CLIP's pretraining data pool [26]. For each of the 100 image-text pairs, we compute the sample-wise contrastive loss using a CLIP ViT-L-14 model, and average it over concepts. The lower the mean loss

Table 3: `FoMo-in-Flux` **Evaluation-only Datasets.** We utilize a subset of standard evaluation datasets used in Radford et al. [137], as well as an array of ImageNet-like variations (including the original ImageNet) to probe different aspect of vision-language understanding and alignment. Moreover, datasets like Food101 [15] or OxfordPets [122] were selected due to their high initial zero-shot performance scores.

Dataset	# Train	# Test	# Classes	Domain	License	Captions
Classification-based						
Caltech101 [90]	6026	2651	101	natural	CC BY 4.0	generated
Caltech256 [50]	21307	9300	257	natural	CC BY 4.0	generated
Cars196 [162]	8144	8041	196	finegrained, natural	custom non-commercial	generated
Cifar10 [87]	50000	10000	10	natural, low-res	Unspecified, but academic usage	generated
Domainnet-Quickdraw [125]	60375	25875	345	sketch	custom non-commercial	generated
EuroSAT [61]	18900	8100	10	Remote Sensing	MIT	generated
FashionMNIST [194]	60000	10000	10	b&w, low-res	MIT	generated
Food101 [15]	75750	25250	101	finegrained, natural	Unspecified, but academic usage	generated
GTSRB [67]	18635	8005	43	traffic signs	CC0	generated
ImageNet [36]	0	50000	1000	natural	custom non-commercial	generated
ImageNet-A [63]	0	7500	200	adversarial, natural	MIT	generated
ImageNet-D [207]	0	4835	103	generative	MIT	generated
ImageNet-R [62]	0	30000	200	renditions (e.g. sketch, paintings)	MIT	generated
ImageNet-S [181]	0	50889	1000	sketch	MIT	generated
ImageNet-V2 [145]	0	10000	1000	natural	MIT	generated
MNIST [37]	60000	10000	10	b&w, low-res	CC BY-SA 3.0	generated
Monkeys10 [2]	1097	272	10	natural	CC0	generated
OxfordPets [122]	3680	3669	37	natural	CC BY-SA 4.0	generated
STL10 [29]	5000	8000	10	natural, low-res	custom non-commercial	generated
SVHN [118]	73257	26032	10	natural, low-res	custom non-commercial	generated
Retrieval-based						
MSCOCO [96] (avg T2I/I2T R@5)	0	5000	0	natural	CC BY 4.0	available
Flickr30k [128] (avg T2I/I2T R@5)	0	1000	0	natural	CC0	available
Total	**462171**	**314419**	**4653**			



Caption: "A black-and-white photo of a square located in the top left (exact position: x = 0.097 (0: left, 1: right) and y = 0.097 (0: top, 1: bottom)."

Caption: "A black-and-white photo of a heart located in the top right: Located at x = 0.839 (with 0 as left, 1 as right) and y = 0.032 (0 top, 1 bottom)."

Caption: "Captured from a -30.00 degree angle: a red, small cube against a blue wall on a blue flooring."

Caption: "A pink sphere (larger size), displayed on blue flooring with a orange background, viewed from a 17.14 degree angle."

Caption: "From the 17.14 degree direction, the orange pill makes a striking impression against a simple backdrop."

Figure 5: **Visualisation of programmatically generated captions** for Shapes3D [16] (*right*) and DSprites [111] (*left*, black and white). Chosen at random, some captions are complete with exact details, while some only have more generic descriptors. Caption style leverages templates generated by GPT-4. The default resolution of these images is 64×64, hence the low-resolution appearance.

per concept, the easier it is. We then sort all the concepts by their mean loss in ascending order, and consider that to be the data stream ordering.

2. Concept Frequency Ordering (`concept-frequency`) draws motivation from Udandarao et al. [173], with user requests for model improvement starting from least frequent concepts first (as these constitute edge cases that are most likely to cause undesired performance drops) and incrementally extending to more frequent concepts, which are already represented well in the pretraining pool.

Implementation. We use the *What's In My Big Data* [40] tool's elastic search index to search for the frequency of occurrence of each of the class names in the `C4` [139] dataset. We compute the frequencies of each of the classes, and order them such that the least frequent concepts (long-tail) occur first and the most frequent ones (head-concepts) are at the end.

3. Concept Similarity Ordering (`similarity`), inspired by Yıldız et al. [197], is based on the hypothesis that training on conceptually similar tasks allows users to minimize catastrophic forgetting over tasks.

Figure 6: **Examples of our generated obscure things and animals along with captions**, covering 100 rare and uncommonly occurring things and animals. For each class, images are generated using either Kandinsky-2.1 [142], Stable Diffusion 2.1 [148] or Dreamlike-PhotoReal [1].

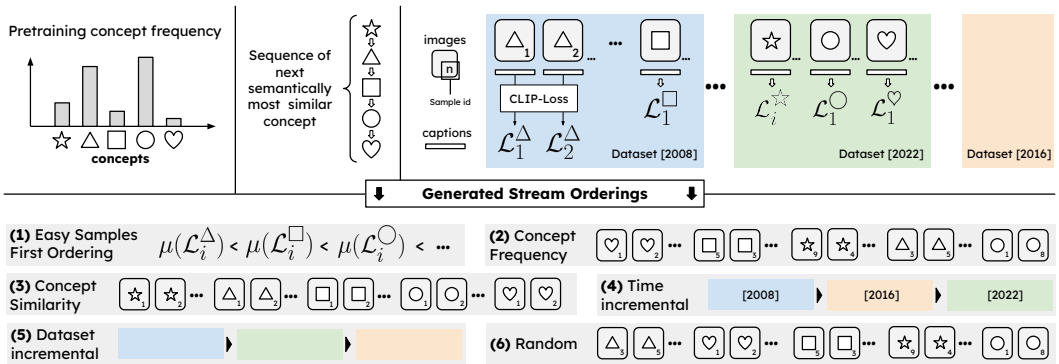


Figure 7: **Pictographic visualization of different data stream orderings** included within the `FoMo-in-Flux` benchmark setup.

Implementation. To find a *trajectory* with the highest semantic similarity between subsequent concepts, we start with a similarity matrix containing the pairwise similarities between all the class names (via CLIP ViT-L-14 text embeddings of templated text captions of the respective classes). Defining each class as a node in a graph, with weights between the classes being their similarity, the problem reduces to finding the minimum spanning path. We use a simple greedy algorithm: pick a starting class, find its closest neighbour from the remaining set of classes, and keep repeating until we exhaust all classes. We repeat this procedure for every class as a starting point and pick the path with the smallest total weight across all starting classes.

4. Time-incremental Ordering (`time`), inspired by [12, 69, 18, 131, 45], arranges in chronological order.

Implementation. As we only have reliable time information about datasets (via release dates of corresponding publications or the official dataset upload date), concepts are ordered on a dataset-level [12]. These year-level groups are arranged from oldest to most recent, assuming that older datasets are more likely to be conceptually integrated within the pretraining data. Within each year, concepts are randomly ordered. Alongside the above orderings, we compare with two baseline methods popular in continual learning, to better understand the trade-offs made by these data-centric orderings:

5. Dataset-Incremental Ordering (`dataset`) is motivated by [143, 107, 108, 183, 199], but extended to a larger sequence of datasets. To set up `dataset`, we simply randomly sample datasets from Tab. 2 to create a dataset-incremental concept sequence. This sequence is then broken down into the desired number of tasks T.

6. Random Ordering (`random`), a baseline class-incremental ordering widely used across continual learning setups [144, 193, 66, 132], mimics a scenario where user requests for model improvement are unstructured. For this ordering, we simply shuffle class names at random.

8.2 Verifying Downstream Datasets: Finetuning must improve Performance

In order to estimate a reference upper bound on adaptation performance, verify the quality of generated captions, and perform a sanity-check on our training pipeline, we fine-tune CLIP-ViT-B/32 and CLIP-

ViT-B/16 individually on each dataset in our training split, as well as all the evaluation-only datasets which come with training samples. We fine-tune the models on each dataset for 10 epochs. For *all datasets*, we find that finetuning a pretrained CLIP model on our generated captions consistently, and in parts very significantly, improves initial zero-shot performance. This showcases the validity of our generated captions, and supports the inclusion of each listed dataset in the `FoMo-in-Flux` benchmark.

9 Continual Pretraining: A Method Perspective

Main Findings

1. **Model Merging** techniques exhibit a unique, promising continual pretraining dynamic (fig. 8)—showing improved base generalization performance for shorter continual pretraining horizons and better retention across full continual pretraining sequence, while also achieving substantial gains in knowledge accumulation beyond that achieved by parameter-efficient tuning techniques or full finetuning.
2. **Parameter-efficient tuning** techniques like `LoRA`, `DoRA` or `VeRA` face significant plasticity issues, meaning they sacrifice the capacity necessary to adapt effectively in a bid to improve knowledge retention (fig. 9 left, right). This behaviour is significantly exacerbated in parameter-selective tuning techniques like `LNFit` and `BitFit`. Low-rank approximations on gradient updates, as done in `GaLore` [212], appear to provide a simple middle ground in knowledge accumulation and retention between full finetuning and parameter-efficient finetuning.
3. **Continual learning regularization** strategies under compute-restricted circumstances show strong plasticity issues when the degree of regularization is high (`EWC`), but have minimal and negative effect (`SI`) when it is low.

[TL;DR] Simple continual finetuning coupled with model merging appears to offer the most promise for continual model pretraining across longer update cycles.

We begin by exploring how different continual learning and finetuning strategies affect knowledge accumulation and zero-shot retention at the model level, with the goal of understanding their trade-offs from a practical perspective. We study several promising directions for continual pretraining of foundation models:

- *Naive continual finetuning* [45, 132, 72], which has emerged as a dominant approach for major updates on realistic large-scale benchmarks, making it a contender for handling minor updates as well.
- *Parameter-efficient tuning methods* like `LoRA` [68], which have become a method of choice for minor updates on a smaller scale or for adapting to new tasks with reduced memory requirements [58, 105, 189, 159, 160, 44, 94] through the use of low-rank weight approximations. In a related fashion, recent work by Zhao et al. [212] has shown promise for model finetuning through low-rank approximations on the optimization gradients (`GaLore`).
- *Parameter-selective tuning methods* such as `BitFit` [7] or `LNFit` [34], which only tune and update particular parameter subsets in the pretrained model such as bias or normalization terms.
- *Traditional regularization strategies* from continual learning literature [82, 202], which have yielded surprisingly strong performance in recent studies both in parameter [91, 210] and feature space [117], despite being developed and tested in small-scale scenarios where the model is trained from scratch.
- *Model merging*, which has gained popularity [190, 74, 141] in non-continual learning scenarios as a means to aggregate models tuned across different tasks, and has been studied in some recent [165, 110] and concurrent works [85, 109] as a method to facilitate continual pretraining over longer adaptation periods.

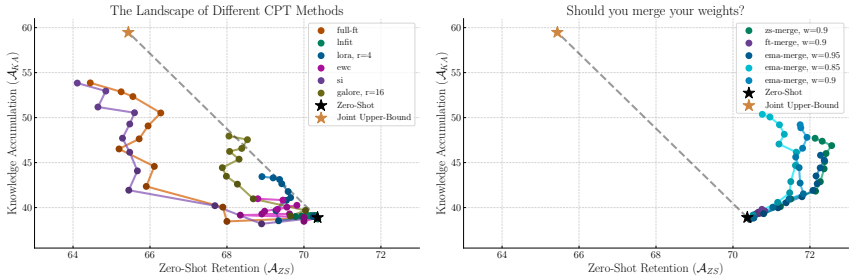


Figure 8: **Which methods to use for continual pretraining over long update cycles?** ***(Left)*** An in-depth study across five different method families: Continual finetuning (`Full-FT` [72]) and parameter-selective tuning (`LNFit` [34]) provide the extreme points in knowledge accumulation and retention. Switching from `GaLore` [212] to parameter-efficient tuning (`LoRA`) and continual learning methods (EWC [82], SI [202]) provides near linear transition points between both extremes. ***(Right)*** Judiciously merging model weights exhibits unique long-horizon continual pretraining behaviour, allowing for significantly consistent accumulation across update tasks with maximal retention.

We excluded certain conceptual approaches from our investigation due to limited capacity and prior evidence strongly suggesting they might not be effective. These include prompt-tuning-based continual learning methods, which often collapse to a single prompt [168] or near-chance performance over a longer time horizon [134]. Similarly, we do not include distillation-based CL methods, as they do not show improvements when memory is unrestricted [132].

9.1 Parameter-efficient Finetuning and Continual Learning

In this section, we leverage `FoMo-in-Flux` to understand the applicability of popular parameter-efficient tuning methods to the continual pretraining setting. In particular, we investigate both *parameter-additive* methods (`LoRA` [68], `VeRA` [84] and `DoRA` [100]) and *parameter-selective* approaches tuning only particular weight subsets (`LNFit` [34] and `BitFit` [7]). Finally, we also study recently proposed low-rank approximations to model gradient updates (`GaLore` [212]). Additionally, we examine the extent to which methods developed under smalls-scale continual learning scenarios such as Elastic Weight Consolidation (`EWC`, [82]) or Synaptic Intelligence (`SI`, [202]) can be utilized to provide a favourable trade-off between accumulation and retention. Figure 9 showcases the comparison of all methods under our default 20-update step setting on the `random` data ordering stream. To begin, we find two extreme points:

1. **Strongest accumulation, weakest retention.** Naive contrastive `finetuning` (in orange, fig. 8 left) which achieves strongest knowledge accumulation $\mathcal{A}_{\text{KA}}$ across a full update cycle, at the cost of a significant drop in zero-shot retention $\mathcal{A}_{\text{ZS}}$ even when leveraging learning rate rewarming as suggested in [72]. Note that for our continual contrastive finetuning, we follow best practices sketched out in [49], which recommends using the same objective for both continual and initial pretraining. Moreover, we update both the image and language branch of the model, and initialize from the pretraining temperature (see section 10.4 for more details).

2. **Weakest accumulation, strongest retention.** On the other hand, parameter-selective update methods such as `LNFit` (green, fig. 9 center) and `BitFit` (blue, fig. 9 center) exhibit good knowledge retention, but minimal capacity for the accumulation of new knowledge across longer and complex data streams.

Importantly, we find that naive continual finetuning strongly falls victim to "longer-horizon" stability gap issues [33], where forgetting is high and achievable knowledge gain is strongly limited across the first number of update steps (with each update step being a whole compute-budgeted training cycle over a data chunk).

All other tested methods operate between these two ends of the spectrum, trading off knowledge accumulation approaching that of simple finetuning, and knowledge retention to the degree of parameter-selective updates:

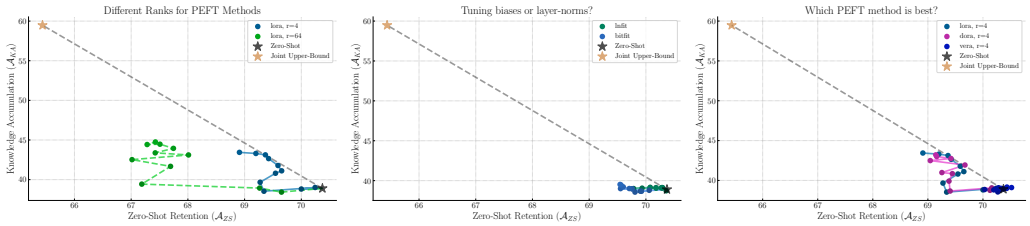


Figure 9: **More Detailed Method Ablations.** (***Left***) Impact of different ranks on continual pretrainability; favouring lower rank values ($r = 4$) over large rank values ($r = 64$) when contrasted against the hypothetical linear tradeoff line between original zero-shot behaviour and performance when finetuned over all data at once. (***Center***) Comparison between parameter-selective `LNFit` [34] and `BitFit` [7]. Both exhibit similar behaviour: strongly limited ability to continuously incorporate new context, with correspondingly minimal deviation in original zero-shot behaviour. (***Right***) Overview of adaptation versus evaluation trajectories for different PEFT methods: `LoRA` [68], `DoRA` [100] and `VeRA` [84]. `LoRA` and `DoRA` behave comparably, with low adaptable parameter counts in `VeRA` heavily limiting the ability to accumulate new knowledge.

1. **Strong accumulation, weak retention.** By retaining the forward pass of the model and only modifying the naturally lower-rank gradient updates during model training, `GaLore` (olive green, fig. 8 left) offers a moderate balance between the ability to effectively incorporate new knowledge within a given compute budget, and retaining original zero-shot generalization behaviour.
2. **Decent accumulation, decent retention.** Parameter-efficient tuning methods such as `LoRA` (blue, fig. 8 left) and `DoRA` (pink, fig. 9 right) provide an effectively linear reduction in both knowledge accumulation and forgetting (particularly with respect to full finetuning) compared to `GaLore`. This conceptually also aligns with recent insights on `LoRA` effectively both learning and forgetting less even in single domain finetuning tasks [10]. However, `VeRA` (dark blue, fig. 9 right), which significantly reduces the number of tunable parameters, behaves closely to parameter-selective tuning methods, offering very little knowledge gain across long and complex data streams.

For parameter-efficient tuning, the scaling between the accumulation-forgetting trade-off and the tunable parameter count is also unsurprisingly reflected when adjusting the rank of `LoRA` (fig. 9 left)—though the loss in original generalization performance outweighs the achievable knowledge accumulation when contrasted against the hypothetical trade-off line between initial zero-shot behaviour and joint finetuning.

Finally, for continual learning regularization methods we find that while `EWC` (pink, fig. 8 left) significantly improves zero-shot retention, it also offers extremely limited $\mathcal{A}_{\text{KA}}$ compared to the initial zero-shot performance. On the other hand, the popular regularisation method `SI` (purple, fig. 8 left) effectively offers no benefits over standard finetuning, either in $\mathcal{A}_{\text{KA}}$ or $\mathcal{A}_{\text{ZS}}$. The poor performance of regularisation-based methods is curious as prior work has hinted at their benefits at scale [117, 81]. However, our fine-grained, and most importantly compute-controlled `FoMo-In-Flux` helps verify these claims, as these regularization mechanisms are both compute- and memory-expensive.

9.2 On the Benefits of Model Merging Techniques

Recently, model merging has emerged as a promising avenue for adapting foundation models [190, 74, 165], enabling efficient aggregation of multiple expert models [196, 151, 31, 5]. Initial work [165] also highlights its potential benefits in small-scale, classification-based continual learning settings. To study their benefits at scale, we investigate three forms of model merging. Denoting the model weights going into task t as θ_{t-1}, the finetuned weights after task t as θ'_t, and the final model-merged output after task t as θ_t, we define (c.f. fig. 10 for details):

1. *Exponential-moving averaging* (`EMA-merge`), as adopted in Stojanovski et al. [165], which tunes the previously merged task weights θ_{t-1} on task t to produce the finetuned weights θ'_t, and then merges θ_{t-1} with θ'_t to produce θ_t.

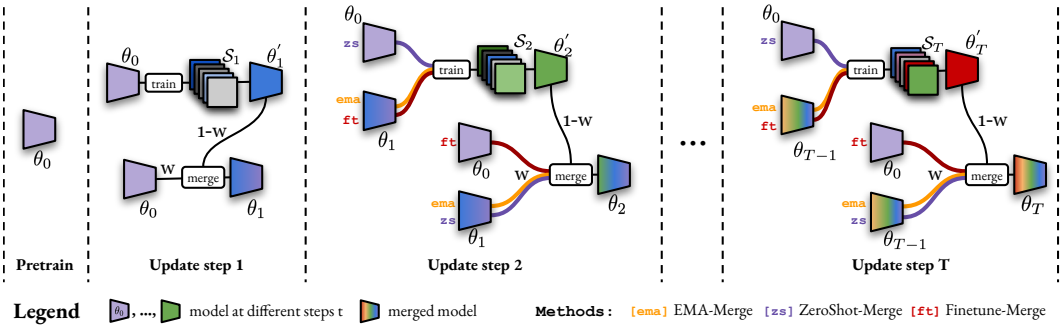


Figure 10: **Different model merging strategies** explored in this work. We use θ' to denote weights θ finetuned after a respective task. Merging θ_{t-1} and θ'_t then results in the merged outputs weights for task t, θ_t. `EMA-Merge`, or exponential moving average merging, merges previously merged weights θ_{t-1} with current task weights θ'_t produced by tuning the same previously merged θ_{t-1} on task t. `ZeroShot-Merge` always tunes the original pretraining weights θ_0 on each task, then weight-interpolates between the finetuned θ'_t and the previously merged θ_{t-1}. `Finetune-Merge` always interpolates between the original pretraining weights θ_0 and the finetuned weights θ'_t. To arrive at θ'_t, the previously merged model θ_{t-1} is trained on task t.

2. *Continual fine-tuning and merging* (`Finetune-merge`) derived from multi-model patching in Ilharco et al. [74]), which produces θ_t by merging the original pretraining weights θ_0 and the finetuned weights θ'_t. To obtain θ'_t, `Finetune-merge` tunes the previously merged model weights θ_{t-1}, same as `EMA-merge`.

3. *Continual zero-shot merge* (`ZeroShot-merge`), a simple ablative merging protocol, which tunes the original pretraining weights θ_0 during each task t and produces θ_t by merging θ_{t-1} and the finetuned θ'_t.

Each `merge` method uses an old-new weight mixing coefficient w, which we ablate over w={0.85,0.9,0.95}. As shown in fig. 8 (right), we surprisingly find that the `EMA-merge` (blue) and `ZS-merge` (green), for the first time, provide impressive boosts in zero-shot retention rates $\mathcal{A}_{\text{ZS}}$ during the first update tasks, and retain slight gains over the entire update cycle.

Moreover, this is coupled with strong knowledge accumulation $\mathcal{A}_{\text{KA}}$, though not yet at the level of standard `finetuning`. As expected, ablating the mixing weight w yields a trade-off between zero-shot retention and knowledge accumulation—higher ws provide better zero-shot retention capabilities while compromising on the accumulation $\mathcal{A}_{\text{KA}}$. However, across both ablated mixing ratios, as well as the merging mechanism, we find that the high-level continual pretraining dynamics remain the same—at worst limited loss (and at best notable gains) in zero-shot retention coupled with strong accumulation capacities, while also breaking favorably with the hypothetical linear trade-off between the initial zero-shot performance and the joint finetuning upper-bound. This strongly contrasts with the method families studied in the previous section, which trade any acquired knowledge accumulation for a strong reduction in zero-shot generalization capabilities.

10 Continual Pretraining: General Training Recipes

Main Findings

1. **Learning rates and schedules matter** in continual pretraining over long update cycles. While the specific choice of schedule for each update task has limited impact, correctly defining meta-schedules modifying each task-specific schedule as a function of the deviation from the initial pretraining weights can significantly break forgetting while allowing for nearly the same degree of knowledge accumulation! Importantly, such meta schedules can be naturally derived without the inclusion of additional hyperparameters.

2. **Model size matters** for continual pretraining. By increasing the model size, retention of generalization performance becomes much less a trade-off with knowledge accumulation. Increased capacity allows the model to acquire high degree of new knowledge without incurring high rates of forgetting; and even allowing for additional positive backward transfer. Consequently, when expecting longer model update cycles, accounting for the higher "future-proofness" of larger models even at higher initial training cost may be crucial.
3. **Compute scaling matters (for some methods)** for continual pretraining. For a fixed model size, increasing the compute budget does not come with a more favorable accumulation-versus-forgetting trade-off when simply finetuning. However, in conjunction with model merging, additional increases in the allocated compute budget actually come with an improved accumulation and forgetting trade-off!
4. **Full model tuning** beats locked image or text encoder training over long update cycles.
5. **Initial stability gap** issues are strongly mitigated by calibration – matching the pretraining and subsequent continual pretraining softmax temperatures.

[TL;DR] Learning rate schedules should account for the update cycle duration. Larger models and compute budgets (particularly alongside model merging) allow for knowledge accumulation with reduced impact on initial knowledge retention, and an overall better accumulation-retention tradeoff.

This section studies the other degrees of freedom orthogonal to particular methodological update strategies that co-occur with the design of a continual pretraining pipeline, particular across our studied longer *minor update* cycles. In particular, this section investigates the following pipeline properties:

1. The importance of the learning rate and its scheduling in section 10.1 as noted already in e.g., [72] - covering the need for matching inital and continual pretraining schedules and the option for meta-learning rate schedules.
2. The impact of both model and compute scaling as independent axes to optimize and account for when planning to deploy a model over longer minor update cycles. More precisely, section 10.2 evaluates the impact on the knowledge accumulation and the zero-shot retention trade-off as a function of both increased model sizes within the same model family, as well as increases in the allocated compute budget within a fixed model size.
3. The relevance of joint image and text encoder tuning in section 10.3 when contrasted against locked image or text encoder training.
4. The importance of aligning initial and continual pretraining softmax temperature in order to minimize stability gap issues highlighted in section 10.4.

10.1 Learning Rates, Schedules and Meta-Schedules

On the Influence of Learning Rate Choices for Continual Pretraining. To define the learning rate of choice for our continual pretraining problem, we derive it directly from the original pretraining values in Cherti et al. [26] (*1e-3*). We note that the exact peak values are corrected for our practical differences in compute availability (operating on a batch-size of $b_{\text{ours}} = 512$ instead of $b_{\text{openclip}} = 88064$); testing both the commonly utilized linear resizing [48]: $\lambda_{\text{scaled}} = {}^{b_{\text{ours}}}/_{b_{\text{openclip}}}$.

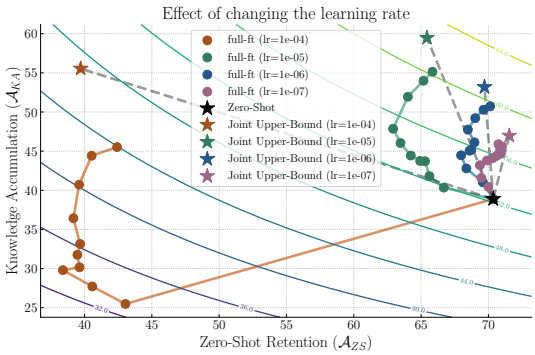


Figure 11: **The effect of the base learning rate on continual pretraining**. The learning trajectory is shown for each value of the learning rate, with the joint training performance as an upper bound. The contour lines show the geometric mean of knowledge accumulation and zero-shot retention ($\sqrt{\mathcal{A}_{KA} \times \mathcal{A}_{ZS}}$). A learning rate of $1e{-5}$ derived from the inital pretraining learning rate achieves the highest

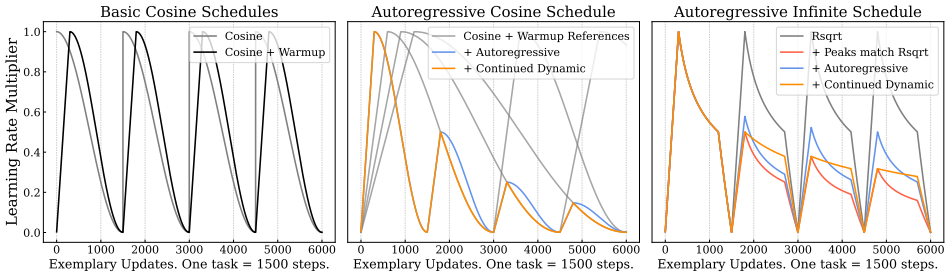


Figure 12: **Visualization of different deployed learning rate schedules**, from task-independent *cosine* and infinite learning rate schedules (*Rsqrt*), to task-dependent meta learning rate schedule.

$\lambda_{\text{openclip}}$ and the respective square-root resizing [88] (giving $5.81e-6$ and $7.625e-5$, respectively). In preliminary experiments, we found that rounding up the linearly resized reference (to $\lambda_{\text{scaled}} = 1e-5$) worked slightly better than both options, and provides a much cleaner entry point. As such, we chose to utilize $1e-5$ as our learning rate reference value. As we find in fig. 11, this (mostly) direct re-use of the maximum learning rate has most importantly the highest degree of knowledge accumulation, but also achieves the highest base joint tradeoff with respect to zero-shot retention. Larger learning rates incur significantly higher rates of particularly early-task forgetting, while smaller learning rates limit the amount of knowledge gained. As such, we set $\lambda_{\text{scaled}} = 1e-5$ as our base learning rate.

Continual Pretraining Learning Rate Schedules. By default, LR schedules are applied on each task individually [17, 155, 13, 165, 104]. As `open_clip` models are trained using cosine schedules, we first study the impact of re-applying the same cosine schedule for each task:

$$\eta_n = \begin{cases} \eta_{\text{min}} + \frac{n}{N_{\text{warm}}}(\eta_{\text{max}} - \eta_{\text{min}}) & n < N_{\text{warm}} \\ \eta_{\text{min}} + \frac{1}{2}(\eta_{\text{max}} - \eta_{\text{min}})\left(1 + \cos\left(\frac{n - N_{\text{warm}}}{N_{\text{task}} - N_{\text{warm}}}\pi\right)\right) \end{cases} \tag{1}$$

with $\eta_n \in [\eta_{\text{min}}, \eta_{\text{max}}]$ the learning rate at step n, and N_{task} the number of update steps for a given task. As recommended in e.g. Ibrahim et al. [72], we utilize linear warmup to the initial pretraining peak learning rate η_{max} used in Cherti et al. [26] for N_{warm} iterations.

To study the impact of a learning rate schedule switch to e.g. infinite learning rate variants for potentially more flexibility down the line, we investigate a switch towards reciprocal square root schedule (*rsqrt*) introduced in Zhai et al. [204]

$$\eta_n = \begin{cases} \eta_{\text{min}} + \frac{n}{N_{\text{warm}}}(\eta_{\text{max}} - \eta_{\text{min}}) & n \geq N_{\text{warm}} \\ \eta_{\text{max}} \cdot \frac{\sqrt{N_{\text{warm}}}}{\sqrt{n + N_{\text{warm}}}} & n \in [N_{\text{warm}}, N_{\text{task}} - N_{\text{cool}}] \\ \eta_{N_{\text{task}} - N_{\text{cool}}} \cdot \frac{N_{\text{task}} - (n + N_{\text{warm}})}{N_{\text{cool}}} & \text{else} \end{cases} \tag{2}$$

Note that *rsqrt* scheduling includes a separate cooldown section, wherein the last N_{cool} steps are used to linear cooldown the previously decayed learning rate.

Both schedules are visualized in fig. 12 (left and right) over multiple tasks, and the result of either application (matching and changing the pretraining learning rate scheduler) to our 20 task update cycle stream is visualized in fig. 13 (center). As can be seen, there is a negligible change in knowledge accumulation $\mathcal{A}_{KA}$ and knowledge retention for either learning rate scheduler; highlight that across longer update cycles, matching the original pretraining scheduler is of lesser importance.

Meta Learning Rate Schedules. In the previous case, by default, each intermediate update is treated independently (see the scheduler visualization in fig. 12 (left)) - meaning each task rewarms and cools down to the same learning rate and with the same decay and cooldown dynamics. However,

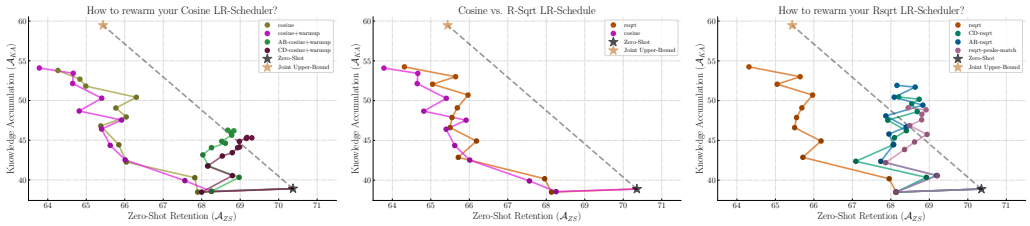


Figure 13: **Meta-scheduling** task-specific LR scheduler has significant impact on the knowledge accumulation and retention trade-off, with meta-schedules derived from infinite LR schedules showing significant transitions across the zeroshot vs finetuning threshold; moving close to accumulation performance of task-independent scheduling, but retaining significantly more pretraining knowledge.

as these continual pretraining updates appear in succession, catastrophic forgetting of previously seen tasks has to also be accounted for, going beyond just the loss in initial foundational knowledge. On top of that, with every task update, the model is encouraged to move further away from its pretraining starting point.

To reduce the impact of task-level forgetting and the increased shift from pretraining, we introduce meta LR scheduling - task-level schedules over each task-specific, iteration-level LR schedule to account for task continuity. These derive *naturally and hyperparameter-free* from hypothetical scenarios wherein the previous task schedule is simply extended across all the new tasks (see gray hypothetical schedules in fig. 12 (*center*)).

In particular, we explore four meta-schedules: **(i)** *autoregressive cosine scheduling*, which selects η_{max} for each task-schedule by building a hypothetical cosine schedule with warmup across the current and all seen tasks and sets it to the intersection point with the warmup process of each respective task (c.f. Fig. 12 center):

$$\eta^T_{\text{max}} = \eta^{\cos}(n' = N^T_{\text{warm}} + \textstyle\sum_t^{T-1} N^t_{\text{task}}, N'_{\text{task}} = \sum_t^T N^t_{\text{task}}) \tag{3}$$

where $\eta^{\cos}(\cdot,\cdot)$ defines the LR returned by the standard cosine LR schedule with warmup at point n' for N'_{task} total iterations. Using the same formulation, we also test **(ii)** *autoregressive continued dynamic* schedule, which warms up to the same η^T_{max}, but continues the schedule following the hypothetical cosine schedule over all total previous steps N_{previous} and the current post-warmup steps N_{warm}. This autoregressive scheduling is naturally extended to the **(iii)** *autoregressive rsqrt schedule*, which simply sets $\eta_{\text{max}} = \eta^{\text{rsqrt}}(n', N'_{\text{task}})$, and **(iv)** which similarly continues the dynamics of a hypothetically extended base schedule ("*Continued Dynamic*"). Finally, we also introduce **(v)** "*Peaks match Rsqrt*", where respective η_{max} matches the continued dynamics while continuing with a standard rsqrt schedule.

The impact of task- and meta-level learning rate schedules for continual model updates are visualized in Fig. 13 on the default 20-task variation of `FoMo-in-Flux` using simple continual finetuning as our reference approach. Indeed, for longer continual pretraining sequences, switching from task-indepedent to meta learning rate schedules notably changes the accumulation versus retention tradeoff behaviour. While within different meta-schedules variations there is limited difference, as shown in fig. 13 (*left* and *right*), meta-learning rate schedules allow for significantly better retention of initial zero-shot transfer performance. In the case of meta-schedules deriving from cosine learning rate schedules however, there is a severe reduction in accumulated new knowledge due to the fast reduction in the learning rate (fig. 12 *left*).

On the opposite end, meta-schedules deriving from infinite learning rate schedules such as the *rsqrt* schedule lend themselves much better to longer-horizon continual pretraining tasks due to the much less aggressive decay in learning rate within tasks: As shown in fig. 12 (*right*), the autoregressive *rsqrt* meta-schedule achieves nearly the same gain in $\mathcal{A}_{KA}$, while *vastly increasing the amount of retained knowledge* and exceeding the hypothetical linear zero-shot vs joint finetuning trade-off line.

10.2 Scaling up Model and Compute Budgets

To understand the impact of both model and compute scaling on the ability to continual pretrain over longer update cycles, we adjust either the underlying vision transformer size (keeping the number

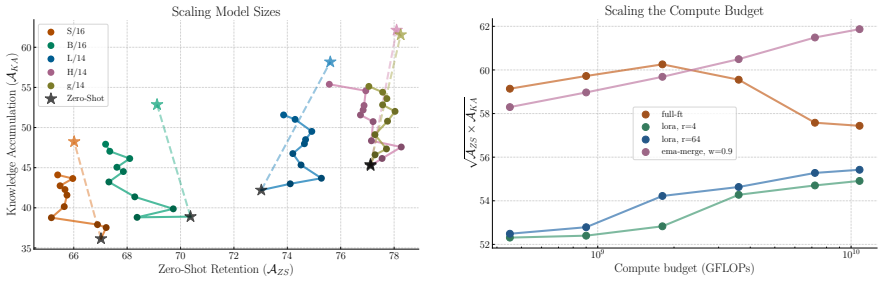


Figure 14: **Model and Compute Scaling for Continual Pretraining.** ***(Left)*** Increasing model size from ViT S/16 to ViT g/14 scales zero-shot performance consistently. In conjunction however, we find that incorporating new context comes with a *reduced* impact on knowledge retention. ***(Right)*** For continual finetuning (with/without model merging), as well as LoRA adapters, we consistently increase the allocated compute budget (for B/16). For normal finetuning, an optimum is reached early. With model merging, we instead see a log-linear scaling in performance with additional compute.

of update steps and task iterations fxied, and covering ViT-S/16 $[62.3M]$, B/16 $[149.6M]$, L/14 $[427.62M]$, H/14 $[986.11M]$ and g/14 $[1366.68M]$ taken from [26]) or the allocated compute budget for a fixed model size (selecting our default ViT-B/16 and the default derived finetuning compute budget of 1.8×10^9 FLOPs as reference, see also section 8). Results for both studies are provided in fig. 14 left and right, respectively.

Scaling Model Size. As can be seen, we find that with a controlled increase of model size, the ability to continually pretrain over longer minor update cycles improves. While the absolute change in knowledge accumulation $\mathcal{A}_{KA}$ remains rather consistent (within the interval of 8% and 10%), zero-shot retention $\mathcal{A}_{ZS}$ improves - where both for the joint finetuning upper bound and continual pretraining, we see improved knowledge retention, and in parts even slight positive backward transfer for ViT-L14 (roughly tripling the parameter count with respect to ViT-B/16).

For ViT-B/16, we see a $\Delta\mathcal{A}_{KA} \approx 9.0\%$ and negative zero-shot retention change $\Delta\mathcal{A}_{ZS} \approx 3.2\%$, while for larger L/14, H/14 and (over a magnitude bigger) g/14 we find $(\Delta_{KA}^{\text{L/14}} \approx 9.4, \Delta_{ZS}^{\text{L/14}} \approx 0.8)$, $(\Delta_{KA}^{\text{H/14}} \approx 10.1\%, \Delta_{ZS}^{\text{H/14}} \approx -1.5\%)$ and $(\Delta_{KA}^{\text{g/14}} \approx 9.8\%, \Delta_{ZS}^{\text{g/14}} \approx -0.05\%)$. Even with higher initial generalization performance, the rate of knowledge accumulation remains roughly the same or even increases, while the ability to maintain its initial generalization capabilities through the longer update cycles in parts notably improves.

These results suggest that model scaling can benefit long-term re-use and the opportunity to maintain and consistently improve the base model over longer minor update cycles, suggesting model scaling helps mitigate forgetting [140]. Our results partly contrasts works in the continual learning domain (though with models trained from scratch) such as [52], which note that at least width alone does not encourage improved knowledge retention. Given our exploratory insights, we believe that our experimental insights warrant further and more controlled inspection into this phenomenon.

Scaling Compute Budgets. Instead of investing into compute increases through larger model sizes, one can also adjust the directly allocated compute budgets; changing for example the number of update steps and task iterations. For our reference model B/16 and its associated compute budget of 1.8×10^9 FLOPs, we thus conduct $2\times$, $4\times$ and $6\times$ increases, as well as $0.5\times$ and $0.25\times$ reductions to understand how the continual pretraining abilities vary as a function of associated compute budgets and the applied continual pretraining strategies of choice.

As seen in fig. 14 (*right*) which aggregates knowledge accumulation $\mathcal{A}_{KA}$ and zero-shot retention $\mathcal{A}_{ZS}$ through their geometric mean, simple continual finetuning (brown) can not consistently leverage increased compute budgets; having to trade off increased knowledge accumulation with a disproportionate loss in the models initial generalization capabilities. However, coupled with simple model merging, we find that models become much better at effectively utilizing the additional budget increase; exhibit a log-linear budget-performance relation. With much lower aggregate accumulation-retention performance, we also find a similar, slightly weaker compute scaling behavior for adapter-based continual pretraining. While the ability to accumulate knowledge, as also indicated in fig. 9, is limited, adapter-based continual pretraining is much more consistent in retaining initial zero-shot performance than simple finetuning.

10.3 Model-specific tuning choices in compute-restricted scenarios

Finally, we highlight the relevance of freezing either image or text encoder in practically compute-restricted continual pretraining in Fig. 15. As freezing either the image or language encoder can allow for significant increases (over a magnitude) in the tuning step budget (as total FLOPs and memory use go down), we find that within the compute-restricted continual multimodal pretraining scenario, tuning both encoders still remains beneficial (aligning with insights provided in Goyal et al. [49] for simple finetuning). While there is negligible difference when freezing each encoder respectively (despite the substantial difference in FLOPs reduction based on tuning the image-encoder alone vs. tuning the text-encoder alone), updating the vision-language model as a joint system incurs a more favorable trade-off between knowledge accumulation and zero-shot retention for each update.

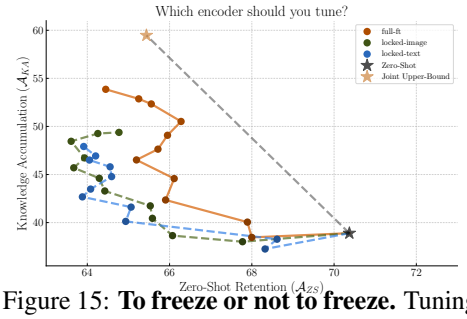


Figure 15: **To freeze or not to freeze.** Tuning both encoders beats single encoder tuning in line with finetuning insights from Goyal et al. [49].

10.4 Softmax Temperatures for Contrastive Losses—*Not Too Hot*!

Recall that CLIP's contrastive loss uses a temperature parameter τ, and it is typically learnable during pretraining. At the beginning of training, it is initialized to 0.07 [137]. Further, to prevent training instabilities, the temperature is clipped to avoid becoming smaller than 0.01. Post training, the learned temperature for all CLIP models considered in this study are found to be exactly 0.01. Moreover, most works that fine-tune a pretrained CLIP model for different downstream tasks, use exactly this learned temperature [49, 170, 171, 190, 39, 74, 57]. Across our main experiments, we follow this standard practice of initializing τ to 0.01 and setting it to be a learnable parameter during continual pretraining. We now explore the impact of different initializations for τ, and sweep over 5 different temperature values, $\{0.01, 0.1, 0.5, 0.75, 1.0\}$. From fig. 16, we observe that τ plays a crucial role for continual pretraining. As we increase the temperature from 0.01 to 0.1, zero-shot retention $\mathcal{A}_{ZS}$ gets impacted by 20% while also noting modest drops on knowledge accumulation $\mathcal{A}_{KA}$, as stability gap issues are excacerbated. Further increasing τ, degrades both A_{ZS} and A_{KA} even more greatly, with the model degenerating to very poor performance. Such drastic changes in model behaviour were also observed in prior work investigating CLIP fine-tuning for downstream tasks [170, 93, 30]—fine-tuning at higher temperatures leads to a decrease in the modality gap between the image and text embedding spaces on the CLIP embedding hypersphere, and hence very quickly degrades the quality of the embedding space for performing downstream tasks [152, 156, 93]. We reproduce and extend the findings of these previous works for the continual pretraining regime, and emphasise the importance of retaining low temperature values for providing optimal $\mathcal{A}_{ZS}$ and $\mathcal{A}_{KA}$.

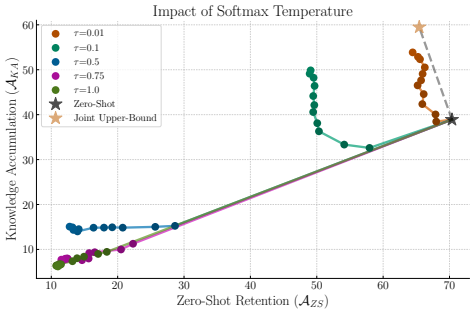


Figure 16: **The softmax temperature for the contrastive loss** is crucial for continual pretraining optimization. The learned temperature after CLIP pretraining is 0.01 (brown trajectory)—higher temperatures than the optimal 0.01 hinder continual pretraining optimization and degrade model weights.

11 Continual Pretraining: A Data-Centric Perspective

Main Findings

1. **Update cycle and deployment scenarios matter.** Within the same overall dataset broken down into continual pretraining updates, trajectors within the accumulation and retention space can significantly differ. If an option, continual updates should be designed as "i.i.d" as possible; ordering based on pretraining concept frequency, concept similarity or loss can result in performance drops particularly during the initial set of updates. However, we find that so long as update cycles operate over the same underlying data distribution that continual pretraining endpoints end up highly similar within the accumulation and retention space.
2. **Retaining a continual pretraining buffer is essential.** Compared to training on currently streamed data and a buffer populated with previously seen streaming data, replaying on pretraining data has much less relative impact. However, the form of subsampling from the pretraining data can notably impact knowledge retention. Together, it is clear that finding ways to "i.i.d"-fy the continual pretraining process is crucial.

[TL;DR] "IID"-fying both the sequence of updates as well as the samples presented at each iteration make the continual pretraining process most effective.

This section provides an important data-centric perspective on continual multimodal pretraining. We study how fine-grained constraints on the sequence of tasks within an update cycle π (section 11.1), specific data-pool choices and mixing ratios between streaming, buffer and pretraining data ($\mathcal{D}/\mathcal{B}/\mathcal{P}$ and $\lambda_{\mathcal{D}}, \lambda_{\mathcal{B}}, \lambda_{\mathcal{P}}$, respectively, in section 11.2), and subsampling over the pretraining data for replay influence favorable trade-offs between between knowledge accumulation $\mathcal{A}_{\text{KA}}$ and zero-shot retention $\mathcal{A}_{\text{ZS}}$ (section 11.3).

11.1 Deployment scenarios impact continual pretrainability

Results on the impact of different deployment scenarios on continual pretrainability over a longer sequence of minor updates are visualized in fig. 17 for the following scenarios (section 8.1): **(1)** `performance` sorted - transition from easy to hard concepts, **(2)** `concept-frequency` sorted - rare pretraining concepts first, **(3)** `concept-similarity` sorted - each update contains concepts semantically related to the preceding update, and **(4)** `random` sorting. `Dataset`-incremental as well as `time`-incremental minor updates are studied separately due to their different structure in section 11.1, and reverse streams are investigated in section 11.1.

Concept- and Sample-based Deployment Scenarios. Across the deployment scenarios in fig. 17 (leftmost), while the `concept-frequency` stream (in green) has the marginally best $\mathcal{A}_{\text{KA}}$–$\mathcal{A}_{\text{ZS}}$ tradeoff with $\mathcal{A}_{\text{KA}}$=55.2, $\mathcal{A}_{\text{ZS}}$=65.6, and `performance` (in pink) performs worst ($\mathcal{A}_{\text{KA}}$=53.8, $\mathcal{A}_{\text{ZS}}$=64.3), we find that *convergence end-points are surprisingly similar* - especially w.r.t. the initial zero-shot and the joint finetuning upper bound reference points. However, while endpoints are remarkably similar, different orderings π induce significantly different trajectories in the accumulation-retention space, with `similarity` the most sample inefficient ordering, while `random` produces the most favorable trajectories. This aligns with prior work from curriculum learning and active learning that have suggested the efficacy of random curriculums [116, 192], which we find extends itself well into the domain of longer-horizon continual pretraining over minor updates. These insights mean that for longer update trajectories and a shared total space of subdomains and tasks of interest, the type and order of continual minor model updates primarily impact initial model versions. This is crucial to account for with respect to the model release horizon and the expected time frame before conducting large-scale continual pretraining updates. However, it also means that across long update horizons irrespective of particular task orders, continually pretrained models arrive at similar performance breakpoints.

Dataset- and Time-based Deployment Scenarios differ from the previous scenarios, in that each update step generally contains much more semantically grouped samples. As we find for both cases

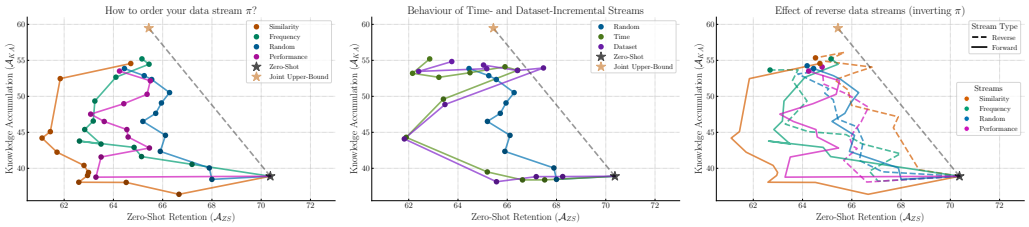


Figure 17: **A Data-centric Perspective on Continual Pretraining.** ***(Left)*** Four concept-level stream orderings π emulating potential update cycles (c.f. section 8.1). Results indicate that deployment scenarios heavily impact intermediate model update stages; however when update cycles operate over shared underlying data distributions, continual pretraining endpoints end up *highly similar*. ***(Center)*** Dataset-level (random or time-incremental) update cycles exhibit less stable deployment trajectories due to high dataset biases [169, 101].***(Right)*** Reversing concept-level datastreams reveals significant trajectory changes. However, the end point similarity still persists.

(randomly ordering datasets in `dataset` or time-ordering in `time`), such an update format induces significantly higher trajectory variance, with much less trajectory coherence when compared to the other four streaming orderings studied above. This is expected given prior work suggesting that visual datasets encode heavy biases [169, 101], and hence tasks that explicitly separate these datasets cause much larger distribution shifts than tasks that (more or less) smoothly mix data samples across the datasets on a concept-level. Still, the degree of accumulation remains comparable, though we find that zero-shot retention is impacted disproportionately higher when orderings π or designed on a dataset-level (down to $\mathcal{A}_{ZS} \approx 62.8\%$, compared e.g. $\mathcal{A}_{ZS}^{\texttt{random}} \approx 64.4\%$ and $\mathcal{A}_{ZS}^{\texttt{frequency}} \approx 65.5\%$ in the best case). This is important to account for when designing minor updates with the goal of retaining original zero-shot performance.

What Happens if We Reverse these Deployment Scenarios? Each sequence introduced in section 8.1 introduces its own particular deployment scenario. Naturally, these scenarios may also either occur or be designed to occur in reverse; updating the model for example with hardest examples first, or choosing highly unrelated concepts before honing in on one specific ordering of similar concepts (by reversing `similarity`). These scenarios do not have to be related to their precursors, and can present their own unique update cycle. Evaluating fig. 17 (*right*), `random` remains consistent. The prevalent difference we find in reversing `similarity`; starting with a stream of unrelated concepts (more so than just random subsampling) and then moving towards a stream of more related concepts. Effectively, early task composition becomes forcibly harder. In doing so, the loss in retention along the trajectory comes with increased knowledge accumulation[1].

This allows the trajectory to remain consistent and close to the hypothetical linear trade-off line between the initial zero-shot behavior and the finetuning upper bound - more so even than `random` streams. Both cases however point towards high variation in the presented concepts during each update step being very beneficial for continual pretraining over longer update cycles, especially when trying to retain consistent model behaviour for each update. Still, even when also accounting for the reversed `performance` ordering, end-points converge to comparable end points! We find the only outlier to this to be the reverse `frequency` stream. As head concepts are encountered early, knowledge accumulation is lower, while the controlled placement of long-tailed, rare concepts towards the end of the update cycle, result in disproportionate forgetting of frequent concepts crucial for achieving and retaining overall accumulation and retention performance.

11.2 Data mixtures inform knowledge accumulation and zero-shot retention

Data control is also reflected in the use of different mixing ratios $\lambda_{\mathcal{P}/\mathcal{D}/\mathcal{B}}$, which we study in Fig. 18. The particular ratios investigated are motivated as follows (note that the baseline reference ratios we use for all our experiments are $\{\lambda_{\mathcal{P}}{=}0.33, \lambda_{\mathcal{D}}{=}0.34, \lambda_{\mathcal{B}}{=}0.33\}$ (in orange)):

No Buffer $\{\lambda_{\mathcal{P}}{=}0.5, \lambda_{\mathcal{D}}{=}0.5, \lambda_{\mathcal{B}}{=}0\}$ **(in pink)** significantly degrades both accumulation and retention, hampering the $\mathcal{A}_{\text{KA}}$–$\mathcal{A}_{\text{ZS}}$ tradeoffs ($-14\%\mathcal{A}_{\text{KA}}$ and $-2.5\%\mathcal{A}_{\text{ZS}}$ compared to the reference).

[1] By composing harder tasks, batch composition becomes also more difficult, which has been aligned with improved vision-language representation learning in *e.g.*, Zhai et al. [206]. Though by reversing `similarity` in our case, the aggregation of similar concepts towards the end of the stream results in diminished knowledge accumulation towards the end of the sequence.

Pretrain-heavy $\{\lambda_{\mathcal{P}}{=}0.8,\lambda_{\mathcal{D}}{=}0.1,\lambda_{\mathcal{B}}{=}0.1\}$ **(in blue)** also does not improve over the reference, since at each update step, we input fewer update samples from $\mathcal{D}$, limiting the accumulation capacity.

Ibrahim et al. [72] $\{\lambda_{\mathcal{P}}{=}0.05,\lambda_{\mathcal{D}}{=}0.48,\lambda_{\mathcal{B}}{=}0.47\}$ **(in green)** defines the mixture ratio used in past CPT work operating on LLMs. We reproduce the findings of [72], finding a 5% pretraining replay suffices to provide a better accumulation tradeoff compared to the reference ($+2.2\%\mathcal{A}_{\text{KA}}$ and $-0.3\%\mathcal{A}_{\text{ZS}}$), suggesting that replaying pretraining data is less essential for optimal performance.

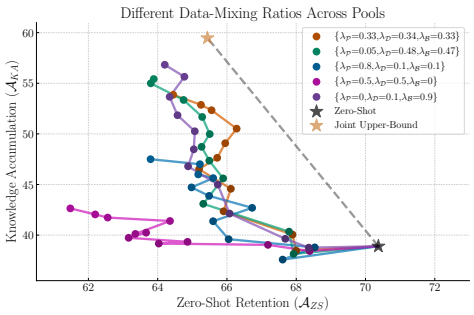


Figure 18: **Different Data Mixture Ratios** $\lambda_{\mathcal{D}/\mathcal{P}/\mathcal{B}}$ between pretraining $\mathcal{P}$, update $\mathcal{D}$ and buffer pool $\mathcal{B}$ yield significantly different adaptation-retention behaviour. "IID-fying" the continual pretraining process through frequent streaming buffer replay is most crucial.

IIDify $\{\lambda_{\mathcal{P}}{=}0,\lambda_{\mathcal{D}}{=}0.1,\lambda_{\mathcal{B}}{=}0.9\}$ **(in violet).** Inspired by the previous result of [72], the question arises on the importance of the overall pretraining pool $\mathcal{P}$. Extending findings in Prabhu et al. [132], we jointly also increase the buffer mixing ratio to encourage more IID training distributions at each update step from the full $\mathcal{D}$ and $\mathcal{B}$ pools. Doing so provides the favored tradeoff compared to all the previous mixtures, corroborating findings in [132].

11.3 Choice of pretraining data pool significantly impacts zero-shot retention

While the overall relevance of replay on pretraining data may be smaller than suitable buffer choices, we complete the previous study by investigating the impact of the pretraining data pool $\mathcal{P}$ on the end model. We experiment with three other pretraining data pools of diverse volumes, caption-sources, curation strategies, and quality measurements—`CC-3M` [154], `CC-12M` [20], `DataComp-Small` [42]—beyond our reference pool `LAION-400M`. For a fair comparison, we randomly subsample each pretraining data pool to a total size of 2M samples, and use this subset as our final pretraining pool P. Here too, we use the reference mixture ratio setting of $\{\lambda_{\mathcal{P}}{=}0.33,\lambda_{\mathcal{D}}{=}0.34,\lambda_{\mathcal{B}}{=}0.33\}$. From fig. 19, it is immediately evident that the choice of the pretraining data pool has a relevant impact on the $\mathcal{A}_{\text{KA}}$–$\mathcal{A}_{\text{ZS}}$ tradeoffs. While adaptation capabilities are barely impacted, using `DataComp-Small` (in pink) yields significantly better zero-shot retention properties, (upto $2.4\%\mathcal{A}_{\text{ZS}}$) gains). We speculate that this could be attributed to the purely English-centric nature of the `CC/LAION` pools compared to the unfiltered `DataComp-Small` which has a significantly higher multilingual and cultural diversity, which has been shown to be beneficial for downstream performance previously [119, 120, 129].

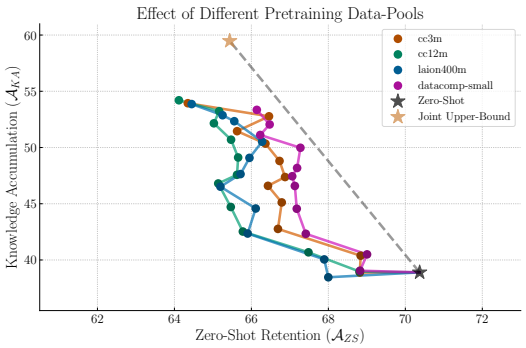


Figure 19: **Quality and Diversity of the Pretraining Pool** $\mathcal{P}$ can matter significantly for retention of initial zero-shot performance, but have limited impact on the ability to accumulate new knowledge.