Early Data Exposure Improves Robustness to Subsequent Fine-Tuning

Anonymous Authors[1]

Abstract

How can we train models whose post-trained capabilities survive subsequent fine-tuning? Rather than focusing on downstream interventions to mitigate forgetting of upstream capabilities, we study how upstream training choices — that is, the manner in which a capability is acquired — shape how robustly that capability is retained. We investigate this question in a controlled three-stage language-model pipeline: pretraining, post-training to acquire a target capability, and downstream fine-tuning on a new objective. Across 135M and 1B models, two post-training domains, and two downstream fine-tuning tasks, we find that immediate post-training performance does not reliably predict retention after subsequent fine-tuning. In particular, *early exposure* — mixing post-training data into pretraining — consistently improves the frontier between retained upstream performance and downstream performance. Replay and dropout, applied during post-training, provide complementary gains. Our findings suggest that robustness to subsequent fine-tuning should be treated as a first-class objective of upstream training, addressed preventatively rather than reactively during fine-tuning itself.

1. Introduction

When a post-trained language model is released for downstream fine-tuning, its carefully acquired capabilities are at risk. A downstream team that adapts the model — to specialize it for a domain, refresh its knowledge with new events, or repurpose it for an agent or code workflow — routinely sees catastrophic forgetting of the instruction-following, alignment, or safety behaviors that the upstream developer invested heavily to install (Yang et al., 2025; Olmo et al., 2025). Most prior work treats this as a problem for the downstream fine-tuner: replay earlier data, regularize the update, restrict the trainable parameters, or jointly optimize competing objectives (Bethune et al., 2025; Kirkpatrick et al., 2017; Hu et al., 2021; Biderman et al., 2024; Wortsman et al., 2022).

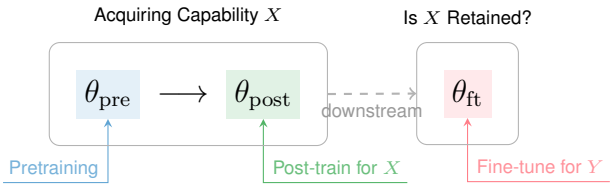


Figure 1. Three-stage pipeline. A first party pretrains then post-trains a model on capability X. A downstream user fine-tunes θ_{post} on a task Y, causing forgetting of X. We study how the *way* X is learned upstream affects whether it is retained.

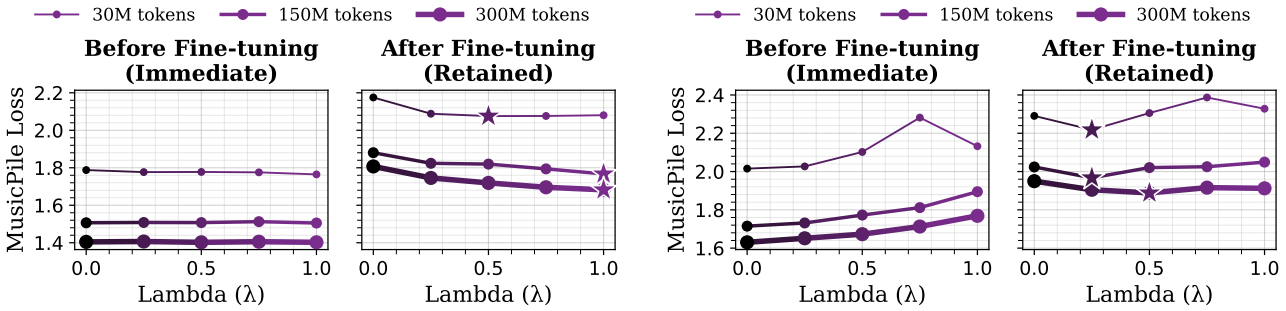
We take a complementary view: robustness to subsequent fine-tuning should be treated as an objective of upstream model development. Our central thesis is that *how a model learns shapes how it forgets* — two models that reach identical post-training performance can differ substantially in how well those capabilities survive later adaptation.

To study this, we use a controlled three-stage pipeline (Figure 1): an upstream developer first pretrains on a broad corpus $\mathcal{D}_{\text{pre}}$, then post-trains on a smaller targeted dataset $\mathcal{D}_{\text{post}}$ to acquire capability X, and finally hands the resulting model θ_{post} to a downstream user who fine-tunes on a new objective $\mathcal{D}_{\text{ft}}$. We hold the downstream method fixed (standard SFT) and sweep its learning rate to characterize how upstream choices shape the tradeoff between downstream performance and retention.

Our main intervention is simple: we expose the model to some of the eventual post-training data earlier by mixing it in during pretraining. Across datasets and model sizes, this *early exposure* improves the tradeoff between retained upstream capability and downstream fine-tuning loss (Figure 3), even when it has little or no visible effect on immediate post-training performance. Our compute-matched experiments show that even under a fixed post-training data budget, the optimum lies between the extremes of all-pretraining and all-post-training allocation. Replay and dropout, two

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the ICML 2026 Workshop "Continual Adaptation at Scale: Towards Sustainable AI". Do not distribute.



(a) Varying mixture fraction λ. *(b)* Compute-matched.

Figure 2. **Left:** As λ increases, immediate MusicPile loss is nearly constant while retained MusicPile loss after fine-tuning on ChemPile improves — mixing benefits are *latent*. **Right:** With total MusicPile exposure held fixed across pretraining and post-training, increasing λ worsens immediate MusicPile loss but improves retained MusicPile loss. Even under a fixed budget, allocating some exposure earlier yields better retention.

classic interventions, provide further complementary gains when applied during post-training. A theoretical analysis (§4) suggests early exposure causes the post-training capability to be represented in *specialized features* that are isolated from directions overwritten by downstream fine-tuning.

2. Setting and Evaluation

A developer (1) pretrains on $\mathcal{D}_{\text{pre}}$ to produce θ_{pre}, optionally mixing a fraction $\lambda \in [0, 1]$ of $\mathcal{D}_{\text{post}}$ into this stage (λ=0: no exposure; λ>0: *early exposure*; we restrict $\lambda \leq 1$, at most one pass over $\mathcal{D}_{\text{post}}$). They then (2) post-train on $\mathcal{D}_{\text{post}}$ to yield θ_{post}, and a downstream user (3) fine-tunes on $\mathcal{D}_{\text{ft}}$ to produce θ_{ft}. Writing $\mathcal{L}(\theta; \mathcal{D})$ for held-out validation loss, we track three central quantities:

$$\mathcal{L}_{\text{im}} := \mathcal{L}(\theta_{\text{post}}; \mathcal{D}_{\text{post}}) \quad \text{(immediate)},$$
$$\mathcal{L}_{\text{ret}} := \mathcal{L}(\theta_{\text{ft}}; \mathcal{D}_{\text{post}}) \quad \text{(retained)},$$
$$\mathcal{L}_{\text{ft}} := \mathcal{L}(\theta_{\text{ft}}; \mathcal{D}_{\text{ft}}) \quad \text{(downstream)}.$$

We additionally track the retained pretraining loss $\mathcal{L}_{\text{pre}} := \mathcal{L}(\theta_{\text{ft}}; \mathcal{D}_{\text{pre}})$. Loss is a reliable scale-invariant proxy for capability (Du et al., 2024; Gadre et al., 2024) and smoother than accuracy at our scales.

Frontiers. Sweeping upstream choices and hyperparameters yields checkpoints with different tradeoffs. We summarize attainable tradeoffs by Pareto frontiers in two projections: $(\mathcal{L}_{\text{ret}}, \mathcal{L}_{\text{ft}})$ for retention–adaptation, and $(\mathcal{L}_{\text{pre}}, \mathcal{L}_{\text{ret}})$ for broader-vs-targeted retention.

Pipelines and models. We fix $\mathcal{D}_{\text{pre}}$=C4 and study four pipelines (Table 1), spanning domain adaptation and instruction tuning, with 135M and 1B SmolLM2-style models (Allal et al., 2025). Stage 2 trains to convergence on $\mathcal{D}_{\text{post}}$ (early stopping, up to 2B tokens). Stage 3 fine-tunes for 200M tokens at varying learning rates. Full hyperparameters are in Appendix B.

Pipeline	$\mathcal{D}_{\text{pre}}$	$\mathcal{D}_{\text{post}}$	$\mathcal{D}_{\text{ft}}$
Music → Chemistry	C4	MusicPile	ChemPile
Music → Instruction	C4	MusicPile	FLAN
Instruction → Chemistry	C4	FLAN	ChemPile
Instruction → Music	C4	FLAN	MusicPile

Table 1. Experimental instantiations of the three-stage pipeline.

3. Experiments and Results

We ask: (i) does pretraining-time mixing affect retention once post-training is run to convergence (§3.1)? (ii) under a fixed $\mathcal{D}_{\text{post}}$ budget, should data be mixed or reserved for post-training (§3.2)? (iii) does the benefit persist across hyperparameter sweeps and pipelines (§3.3)? and (iv) how does mixing compose with replay and dropout (§3.4)?

3.1. Immediate post-training performance does not reflect downstream retention

We fix the Stage 2 post-training procedure and vary only $\lambda \in \{0, 0.25, 0.5, 0.75, 1.0\}$ for $|\mathcal{D}_{\text{post}}| \in \{30\text{M}, 150\text{M}, 300\text{M}\}$ (with $\mathcal{D}_{\text{post}} \subset$ MusicPile). After post-training to convergence, we fine-tune on $\mathcal{D}_{\text{ft}} \subset$ ChemPile and measure $\mathcal{L}_{\text{im}}$ and $\mathcal{L}_{\text{ret}}$ at a fixed Stage 3 LR of 5×10^{-5} (Figure 2a).

Result. Varying λ has little effect on $\mathcal{L}_{\text{im}}$: across dataset sizes, immediate post-training loss remains nearly flat. However, the same checkpoints behave very differently under subsequent fine-tuning: as λ increases, $\mathcal{L}_{\text{ret}}$ consistently decreases.

Takeaway. Early exposure can substantially improve retention under subsequent fine-tuning even when it provides little or no benefit to immediate post-training performance.

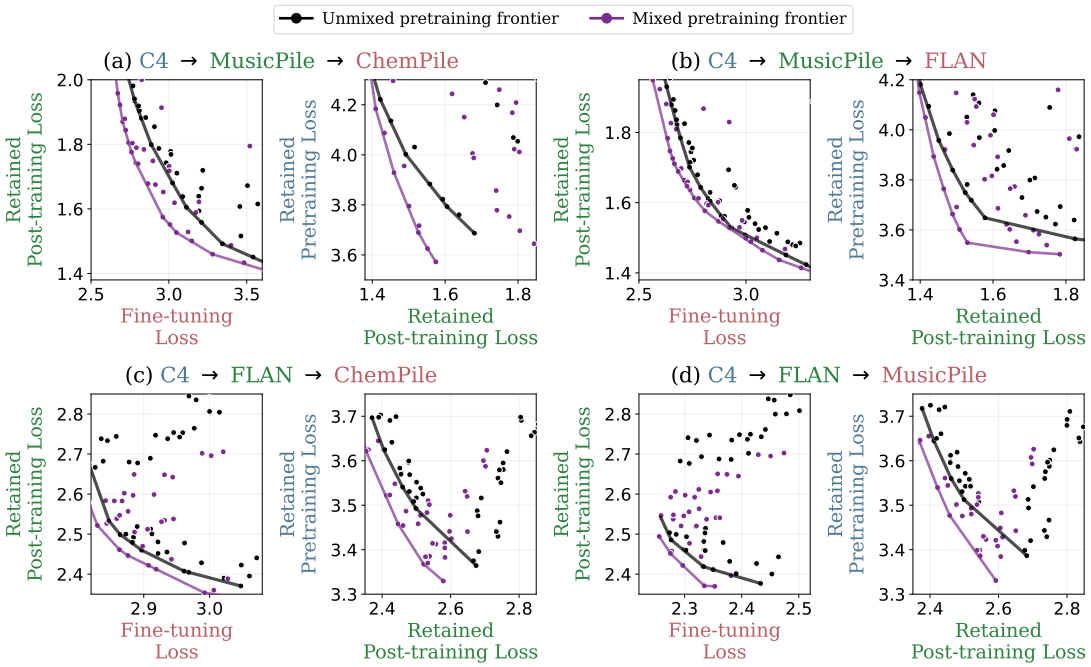


Figure 3. **Mixing during pretraining improves the frontier across four training pipelines (135M).** Each panel corresponds to one 3-stage pipeline. Within each panel, the left plot shows retained post-training loss vs. downstream fine-tuning loss; the right plot shows retained pretraining loss vs. retained post-training loss. **Black** is unmixed pretraining; **purple** is mixed pretraining. Across all four pipelines, mixing shifts the frontier toward lower retained post-training loss, lower retained pretraining loss, and lower downstream fine-tuning loss.

3.2. Early exposure and post-training play different roles under a fixed data budget

To isolate *when* $\mathcal{D}_{\text{post}}$ is introduced from *how much* is seen in total, we fix the total $\mathcal{D}_{\text{post}}$ tokens across Stages 1 and 2 and vary only allocation: a λ-fraction is mixed during pretraining, $(1-\lambda)$ is reserved for post-training. Every model sees exactly one pass over $\mathcal{D}_{\text{post}}$ in total. We sweep $\lambda \in \{0, 0.25, 0.5, 0.75, 1.0\}$ for $|\mathcal{D}_{\text{post}}| \in \{30\text{M}, 150\text{M}, 300\text{M}\}$ on MusicPile → ChemPile (Figure 2b).

Result. As λ increases, $\mathcal{L}_{\text{im}}$ worsens (fewer tokens for the dedicated post-training stage), but $\mathcal{L}_{\text{ret}}$ after Stage 3 fine-tuning consistently improves. The best immediate performance occurs when all of $\mathcal{D}_{\text{post}}$ is reserved for Stage 2; the best retained performance occurs at a nonzero mixture fraction. Concentrating $\mathcal{D}_{\text{post}}$ in Stage 2 yields stronger immediate fitting; exposure during pretraining makes that capability less brittle (Appendix D).

Takeaway. Under a fixed $\mathcal{D}_{\text{post}}$ budget, the best immediate post-training performance occurs when all data is reserved for Stage 2, but the best retained performance occurs at a positive mixture fraction.

3.3. Early exposure improves the loss frontier across hyperparameter sweeps

We now move beyond fixed configurations: for each pipeline (Table 1) we sweep Stage 2 hyperparameters under both unmixed and mixed pretraining, fine-tune every checkpoint over a range of Stage 3 learning rates, and compare frontiers (Figure 3).

Result. Across all four pipelines, early exposure consistently shifts both frontiers. In the $(\mathcal{L}_{\text{ret}}, \mathcal{L}_{\text{ft}})$ view, mixing yields lower retained post-training loss at matched downstream loss. In the $(\mathcal{L}_{\text{pre}}, \mathcal{L}_{\text{ret}})$ view, mixing also improves the tradeoff between preserving broader pretraining capabilities and the post-trained capability. Gains appear across both domain and behavioral post-training, and persist at 1B scale (Appendix C.5).

Takeaway. Across hyperparameter sweeps and training pipelines, early exposure consistently improves the attainable tradeoffs among downstream loss, retained post-training loss, and retained pretraining loss.

3.4. Replay and dropout provide complementary gains

Replay (mixing earlier-stage data into a training run) and dropout (stochastic regularization) are classical tools, typi-

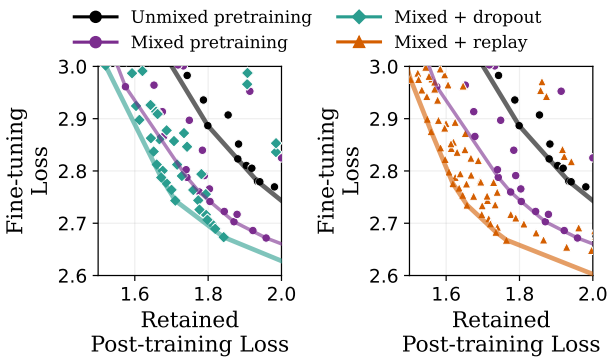


Figure 4. **Replay and dropout provide complementary gains on top of mixed pretraining** (MusicPile → ChemPile, 135M). Adding dropout (left) or replay (right) on top of mixed pretraining further shifts the loss frontier.

cally used by the *downstream* fine-tuner to mitigate forgetting at the moment it occurs (Bethune et al., 2025; Hinton et al., 2012). We repurpose them *upstream* as preventative interventions on *how* $\mathcal{D}_{\text{post}}$ is learned in Stage 2, rather than reactive corrections during Stage 3. Concretely, replay mixes 1% of $\mathcal{D}_{\text{pre}}$ into post-training, and dropout regularizes the post-training update. As before, we sweep Stage 2 hyperparameters and Stage 3 learning rates and report frontiers.

Result. Both replay and dropout further improve the frontier relative to early exposure alone (Figure 4; full panels in Appendix C). Replay encourages the model to acquire $\mathcal{D}_{\text{post}}$ without overwriting broader pretraining features; dropout may promote more distributed representations during Stage 2. Crucially, neither replaces pretraining-time mixing: the strongest frontiers come from combining post-training interventions with mixed pretraining. Pretraining-time mixing changes *when* the model first encounters $\mathcal{D}_{\text{post}}$, while replay and dropout shape *how* that capability is learned during Stage 2. At 1B, replay's effect on $(\mathcal{L}_{\text{ft}}, \mathcal{L}_{\text{ret}})$ is weaker but it continues to preserve broader $\mathcal{D}_{\text{pre}}$ performance (Appendix C.5).

Takeaway. Replay and dropout are upstream alternatives to mixing during pretraining that also improve robustness to subsequent fine-tuning, and their gains are complementary with early exposure.

4. Why does early exposure help?

A two-layer linear analysis (full setup, theorems, and proofs in Appendix D) makes the mechanism precise. Partition the input space into *invariant* features (identical singular values across $\mathcal{D}_{\text{pre}}$, $\mathcal{D}_{\text{post}}$, $\mathcal{D}_{\text{ft}}$), *inconsistent* features (shared dimensions whose values conflict across tasks), and *specialized* features (active only on $\mathcal{D}_{\text{post}}$, with zero covariance under $\mathcal{D}_{\text{pre}}$ and $\mathcal{D}_{\text{ft}}$). Linear networks learn features in descending singular-value order (Gidel et al., 2019; Springer et al., 2025): without exposure to $\mathcal{D}_{\text{post}}$ the specialized features have effective singular value zero and are never learned, so unmixed post-training is forced to lower $\mathcal{D}_{\text{post}}$ loss by perturbing the *inconsistent* features. Early exposure mixes an α-fraction of $\mathcal{D}_{\text{post}}$ during pretraining, boosting the specialized features' effective singular value to $\alpha\beta$ — enough to cross the learning threshold even for small α.

Why this aids retention. Singular values that are zero at the start of post-training stay zero. Mixed post-training therefore routes adaptation through the newly learned *specialized* features, while unmixed post-training continues to use *inconsistent* features. Under subsequent fine-tuning on $\mathcal{D}_{\text{ft}}$, gradient updates have zero projection along specialized directions (since $\mathcal{D}_{\text{ft}}$ has no covariance there) and so leave them untouched, but they overwrite the inconsistent features that lie in $\mathcal{D}_{\text{ft}}$'s span. Adaptation routed through specialized features survives; adaptation routed through inconsistent features does not.

5. Conclusion

We make three claims. First, how well a capability survives later fine-tuning cannot be read off from how well a model performs on that capability immediately after it is acquired. Second, the *manner* in which a capability is learned — when it is introduced, how it is presented, and what else the model is learning at the same time — shapes how durable that capability is. Third, upstream training offers a family of complementary interventions: early exposure during pretraining, replay during post-training, and regularization during post-training each shift the retention–adaptation frontier. Together, these reframe robustness to fine-tuning from a downstream problem into a design objective of upstream training.

Limitations and future work. We cap $\lambda \leq 1$ (at most one pass over $\mathcal{D}_{\text{post}}$ during pretraining). In the two-stage setting, Baek et al. (2026) study much heavier repetition of fine-tuning data during pretraining and find that it can reduce overfitting and forgetting of the pretraining domain; how those dynamics interact with our three-stage setting — where the model is fine-tuned again *after* post-training — remains open. Characterizing the scaling laws of simple methods like early exposure — how their benefits translate to larger models, longer training, and bigger token budgets — is a natural next step. Our downstream stage is supervised fine-tuning only; we leave to future work how early exposure interacts with preference-based and reinforcement-learning schemes such as RLHF and DPO. Finally, we invite work on a complementary algorithmic direction: late-stage objectives or regularizers that reproduce the representational effect of early exposure *without* intervening on pretraining, since pretraining-corpus modifications are often impractical for downstream users of open-weight models. Full proofs of §4 are in Appendix D; related work in Appendix A.

References

Allal, L. B., Lozhkov, A., Bakouch, E., Blázquez, G. M., Penedo, G., Tunstall, L., Marafioti, A., Kydlíček, H., Lajarín, A. P., Srivastav, V., Lochner, J., Fahlgren, C., Nguyen, X.-S., Fourrier, C., Burtenshaw, B., Larcher, H., Zhao, H., Zakka, C., Morlon, M., Raffel, C., von Werra, L., and Wolf, T. Smollm2: When smol goes big – data-centric training of a small language model, 2025. URL https://arxiv.org/abs/2502.02737.

Baek, C., Monti, R. P., Schwab, D., Abbas, A., Adiga, R., Blakeney, C., Böther, M., Burstein, P., Carranza, A. G., Deng, A., Doshi, P., Dorna, V., Fang, A., Jiang, T., Joshi, S., Larsen, B. W., Lee, J. C., Mentzer, K. L., Merrick, L., Mongstad, H., Pan, F., Suri, A., Teh, D., Telanoff, J., Urbanek, J., Wang, Z., Wills, J., Yin, H., Raghunathan, A., Kolter, J. Z., Gaza, B., Morcos, A., Leavitt, M., and Maini, P. The finetuner's fallacy: When to pretrain with your finetuning data, 2026. URL https://arxiv.org/abs/2603.16177.

Bethune, L., Grangier, D., Busbridge, D., Gualdoni, E., Cuturi, M., and Ablin, P. Scaling laws for forgetting during finetuning with pretraining data injection, 2025. URL https://arxiv.org/abs/2502.06042.

Biderman, D., Portes, J., Ortiz, J. J. G., Paul, M., Greengard, P., Jennings, C., King, D., Havens, S., Chiley, V., Frankle, J., Blakeney, C., and Cunningham, J. P. Lora learns less and forgets less, 2024. URL https://arxiv.org/abs/2405.09673.

Du, Z., Zeng, A., Dong, Y., and Tang, J. Understanding emergent abilities of language models from the loss perspective. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Gadre, S. Y., Smyrnis, G., Shankar, V., Gururangan, S., Wortsman, M., Shao, R., Mercat, J., Fang, A., Li, J., Keh, S., Xin, R., Nezhurina, M., Vasiljevic, I., Jitsev, J., Soldaini, L., Dimakis, A. G., Ilharco, G., Koh, P. W., Song, S., Kollar, T., Carmon, Y., Dave, A., Heckel, R., Muennighoff, N., and Schmidt, L. Language models scale reliably with over-training and on downstream tasks, 2024. URL https://arxiv.org/abs/2403.08540.

Gidel, G., Bach, F., and Lacoste-Julien, S. Implicit regularization of discrete gradient dynamics in linear neural networks, 2019. URL https://arxiv.org/abs/1904.13262.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., and Hadsell, R. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114 (13):3521–3526, March 2017. ISSN 1091-6490. doi: 10.1073/pnas.1611835114. URL http://dx.doi.org/10.1073/pnas.1611835114.

Maini, P., Feng, Z., Schwarzschild, A., Lipton, Z. C., and Kolter, J. Z. Tofu: A task of fictitious unlearning for llms, 2024. URL https://arxiv.org/abs/2401.06121.

Maini, P., Goyal, S., Sam, D., Robey, A., Savani, Y., Jiang, Y., Zou, A., Fredrikson, M., Lipton, Z. C., and Kolter, J. Z. Safety pretraining: Toward the next generation of safe ai, 2025. URL https://arxiv.org/abs/2504.16980.

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

Nishi, K., Ramesh, R., Okawa, M., Khona, M., Tanaka, H., and Lubana, E. S. Representation shattering in transformers: A synthetic study with knowledge editing, 2025. URL https://arxiv.org/abs/2410.17194.

O'Brien, K., Casper, S., Anthony, Q., Korbak, T., Kirk, R., Davies, X., Mishra, I., Irving, G., Gal, Y., and Biderman, S. Deep ignorance: Filtering pretraining data builds tamper-resistant safeguards into open-weight llms, 2025. URL https://arxiv.org/abs/2508.06601.

Olmo, T., :, Ettinger, A., Bertsch, A., Kuehl, B., Graham, D., Heineman, D., Groeneveld, D., Brahman, F., Timbers, F., Ivison, H., Morrison, J., Poznanski, J., Lo, K., Soldaini, L., Jordan, M., Chen, M., Noukhovitch, M., Lambert, N., Walsh, P., Dasigi, P., Berry, R., Malik, S., Shah, S., Geng, S., Arora, S., Gupta, S., Anderson, T., Xiao, T., Murray, T., Romero, T., Graf, V., Asai, A., Bhagia, A., Wettig, A., Liu, A., Rangapur, A., Anastasiades, C., Huang, C., Schwenk, D., Trivedi, H., Magnusson, I., Lochner, J., Liu, J., Miranda, L. J. V., Sap, M., Morgan, M., Schmitz, M., Guerquin, M., Wilson, M., Huff, R., Bras, R. L., Xin, R., Shao, R., Skjonsberg, S., Shen, S. Z., Li, S. S., Wilde, T., Pyatkin, V., Merrill, W., Chang, Y., Gu, Y., Zeng, Z., Sabharwal, A., Zettlemoyer, L., Koh, P. W., Farhadi, A.,

Smith, N. A., and Hajishirzi, H. Olmo 3, 2025. URL https://arxiv.org/abs/2512.13961.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and Lowe, R. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.

Qi, X., Zeng, Y., Xie, T., Chen, P.-Y., Jia, R., Mittal, P., and Henderson, P. Fine-tuning aligned language models compromises safety, even when users do not intend to!, 2023. URL https://arxiv.org/abs/2310.03693.

Sam, D., Goyal, S., Maini, P., Robey, A., and Kolter, J. Z. When should we introduce safety interventions during pretraining?, 2026. URL https://arxiv.org/abs/2601.07087.

Springer, J. M., Goyal, S., Wen, K., Kumar, T., Yue, X., Malladi, S., Neubig, G., and Raghunathan, A. Overtrained language models are harder to fine-tune, 2025. URL https://arxiv.org/abs/2503.19206.

Wortsman, M., Ilharco, G., Kim, J. W., Li, M., Kornblith, S., Roelofs, R., Gontijo-Lopes, R., Hajishirzi, H., Farhadi, A., Namkoong, H., and Schmidt, L. Robust fine-tuning of zero-shot models, 2022. URL https://arxiv.org/abs/2109.01903.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu, D., Zhou, F., Huang, F., Hu, F., Ge, H., Wei, H., Lin, H., Tang, J., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Zhou, J., Lin, J., Dang, K., Bao, K., Yang, K., Yu, L., Deng, L., Li, M., Xue, M., Li, M., Zhang, P., Wang, P., Zhu, Q., Men, R., Gao, R., Liu, S., Luo, S., Li, T., Tang, T., Yin, W., Ren, X., Wang, X., Zhang, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Zhang, Y., Wan, Y., Liu, Y., Wang, Z., Cui, Z., Zhang, Z., Zhou, Z., and Qiu, Z. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Yang, X., Wang, X., Zhang, Q., Petzold, L., Wang, W. Y., Zhao, X., and Lin, D. Shadow alignment: The ease of subverting safely-aligned language models, 2023. URL https://arxiv.org/abs/2310.02949.

A. Related Work

Catastrophic Forgetting. A recurring challenge in sequential training is *catastrophic forgetting*: when a model is optimized on new data, its performance can deteriorate on behaviors it previously exhibited (McCloskey & Cohen, 1989). For language models, this shows up in modern training pipelines: instruction tuning and RLHF can trade off against preexisting capabilities, often discussed as an "alignment tax" (Ouyang et al., 2022). Several works show that behaviors introduced during safety finetuning can be quickly weakened or reversed by subsequent training on different objectives or data (Yang et al., 2023; Qi et al., 2023). These tradeoffs also appear in adjacent settings such as knowledge editing (Nishi et al., 2025) and unlearning (Maini et al., 2024). Beyond documenting the effect, recent work has begun to map how training choices shape its severity: LoRA-style adaptation can alter forgetting dynamics (Biderman et al., 2024), and longer pretraining can change how brittle or persistent acquired capabilities are (Springer et al., 2025). We focus on catastrophic forgetting of post-trained capabilities, and study what properties of an intermediate checkpoint determine whether capabilities persist under subsequent training.

Data Placement in Pretraining. A line of recent works examines pre-training interventions for enforcing desired downstream capabilities and properties. Maini et al. (2025); O'Brien et al. (2025) propose filtering and augmenting data during pre-training to improve safety. Similarly, Sam et al. (2026) demonstrate that the impact of such interventions improves as they are introduced earlier in pre-training. While these works incorporate downstream tasks during pre-training, they extensively modify the pre-training corpus by incorporating data-augmentations and filtering. Baek et al. (2026) demonstrate that mixing post-training data during pre-training can immediately improve in-domain performance relative to simply fine-tuning. In our work, we introduce an additional benefit of early exposure to post-training data: robustness to catastrophic forgetting during future training.

B. Training Details

B.1. SmolLM2-1B Model Architecture

Table 2. SmolLM2-1B model architecture (custom config interpolated from SmolLM2 family).

Parameter	Value
Parameters	1.03B
Hidden dimension	1,728
Attention heads	27
Layers	24
Head dimension	64
Query groups	27 (MHA)
MLP intermediate size	4,608
Vocabulary size	49,152
Context length	8,192 (max), 1,024 (training)
Normalization	RMSNorm
Position encoding	RoPE (base=100,000)

B.2. Dataset Statistics

Table 3. 135M Parameter Experiments

Dataset	Split	Tokens	Description
C4	Train	8.7B	General web text pretraining corpus
MusicPile	Train	0.3B	Music-domain text corpus
ChemPile	Train	0.3B	Chemistry-domain text corpus
FLAN	Train	0.3B	Instruction-tuning dataset

Table 4. 1B Parameter Experiments

Dataset	Split	Tokens	Description
C4	Train	19.7B	General web text pretraining corpus
MusicPile	Train	0.3B	Music-domain text corpus
ChemPile	Train	0.3B	Chemistry-domain text corpus
FLAN	Train	0.3B	Instruction-tuning dataset

B.3. Optimizer Configuration

Table 5. Optimizer configuration used across all experiments.

Parameter	Value
Optimizer	AdamW
β_1	0.9
β_2	0.95
Gradient clipping	1.0 (max norm)
Precision	bf16-mixed

B.4. 1B Stage 1 Pretraining

Table 6. Stage 1 pretraining configuration for 1B experiments.

Parameter	Value	Notes
Total tokens	20B	Chinchilla-optimal for 1.03B params
C4 corpus	21.0B tokens	120 shards, tokenized
Learning rate	5e-4	Peak LR
Minimum LR	5e-5	Cosine decay target
Warmup steps	1,000	
Global batch size	512	
Micro batch size	30	Per-GPU
Eval interval	1,000 steps	
Save interval	1,000 steps	
GPUs	8× L40S	Single node
Seed	42	

B.5. Post-training Hyperparameters

Table 7. FFT hyperparameter search space for Stage 2 post-training (135M).

Parameter	Values	Notes
Learning rate	{1e-4, 2e-4, 5e-4, 1e-3, 5e-3}	Peak LR
Minimum LR	5e-5	Cosine decay target
Dropout	{0.0, 0.02, 0.05}	embed/attn/resid/mlp
Weight decay	0.1	Fixed
Warmup steps	500	Fixed
Batch size	{192, 480, 896}	Global batch size
Max tokens	2B	With early stopping

Table 8. Stage 2 FFT hyperparameter search space for 1B-scale post-training.

Parameter	Values	Notes
Learning rate	{1e-5, 2e-5, 5e-5, 1e-4, 2e-4}	Peak LR
Minimum LR	5e-5	Cosine decay target
Dropout	0.0	Fixed (see §B.7 for ablation)
Weight decay	0.1	Fixed
Warmup steps	100	
Global batch size	512	
Micro batch size	30	Per-GPU
Max tokens	2B	With early stopping
CPT budget	300M	Tokens per epoch (subsampled)
Early stopping patience	3	Evaluation intervals
Evaluation interval	100 steps	
GPUs	8× L40S	Single node
Seed	40	

B.6. LoRA Configuration

Table 9. LoRA hyperparameter configuration for Stage 2 post-training.

Parameter	Values	Notes
LoRA rank (r)	64	Fixed
LoRA alpha (α)	128	Fixed, $\alpha/r = 2$
LoRA dropout	{0.0, 0.02, 0.05}	Same as FFT dropout
LoRA targets	projection, mlp, head	Q/K/V excluded
Learning rate	{1e-4, 2e-4, 5e-4, 1e-3, 5e-3}	Same as FFT
Weight decay	0.1	Same as FFT
Other parameters	Same as FFT (Table 7)	

B.7. 1B Stage 2 CPT: Dropout Ablation

Table 10. Dropout ablation configuration for 1B Stage 2 CPT. MusicPile CPT pipeline only. 12 runs total (2 λ × 3 LRs × 2 dropout rates).

Parameter	Values	Notes
Dropout	{0.02, 0.05}	embed/attn/resid/mlp
Learning rate	{1e-5, 2e-5, 5e-5}	Best 3 from baseline
λ	{0.0, 1.0}	MusicPile mixing only
CPT dataset	MusicPile	Priority pipeline
Other parameters	Same as baseline (Table 8)	

C. Additional Plots

C.1. 135M Dropout and Replay Frontiers (Combined)

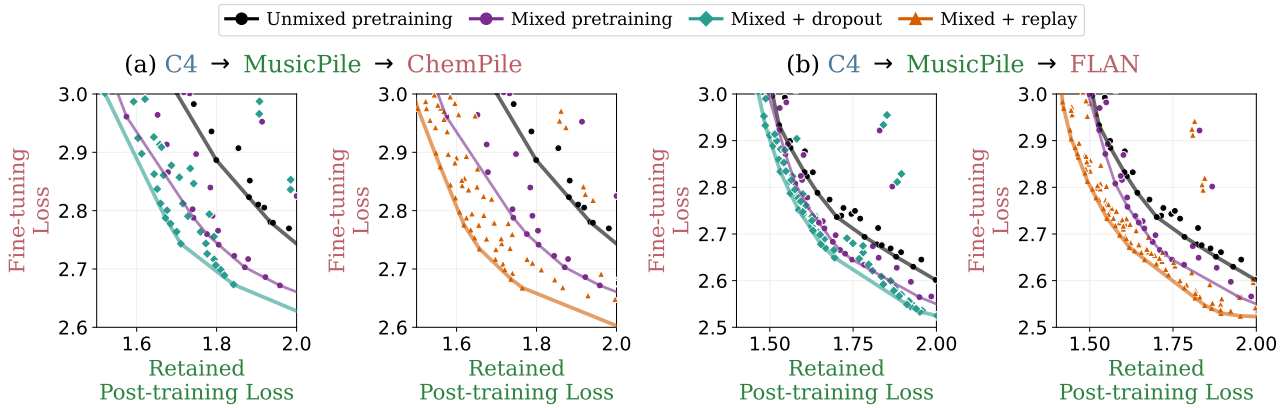


Figure 5. **Replay and dropout provide complementary gains on top of mixed pretraining (135M, full panels).** Each subfigure shows one 3-stage pipeline. The **left** panel compares unmixed pretraining, mixed pretraining, and mixed pretraining + dropout; the **right** panel compares unmixed pretraining, mixed pretraining, and mixed pretraining + replay. Across both downstream settings, adding dropout or replay to mixed pretraining further shifts the loss frontier.

C.2. 135M Dropout and Replay Frontiers with Retained Pretraining Loss (C4)

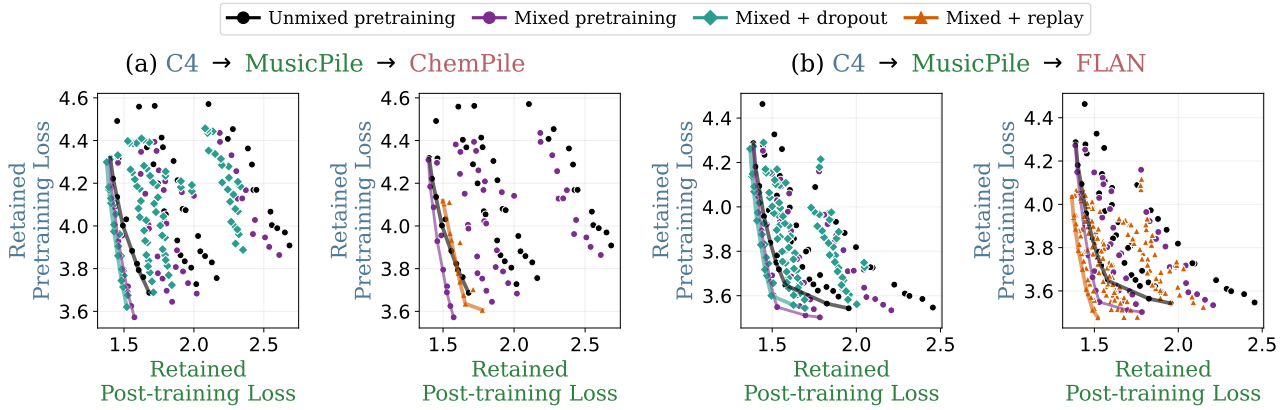


Figure 6. **Dropout and replay preserve broader pretraining capability in addition to the post-training capability (135M).** Companion to Figure 5, plotting the same Stage 2 hyperparameter sweeps against retained pretraining loss on C4 instead of downstream fine-tuning loss. **(a)** C4 → MusicPile → ChemPile. **(b)** C4 → MusicPile → FLAN.

C.3. Additional 135M Dropout and Replay Frontiers (without mixing)

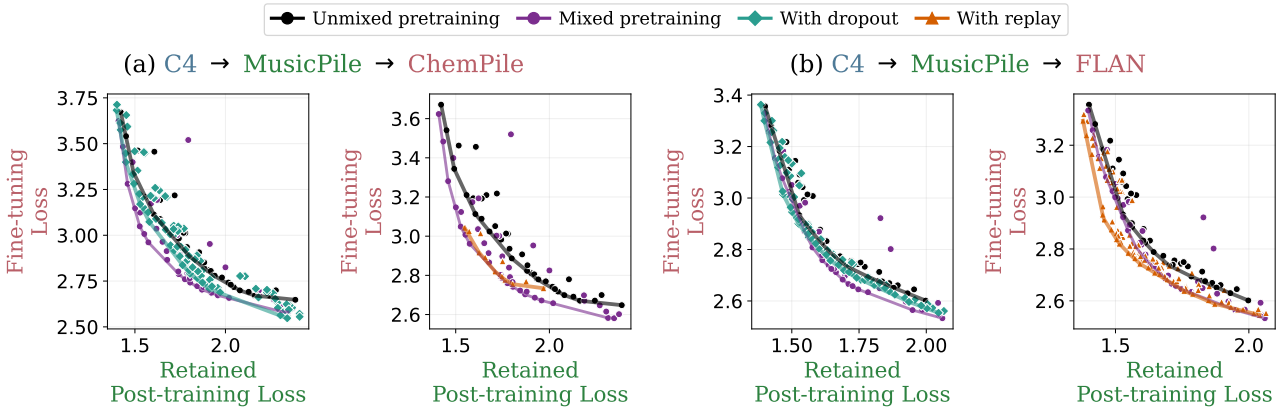


Figure 7. **Dropout and replay applied without pretraining-time mixing (135M).** To isolate the effect of post-training interventions from pretraining-time mixing, each panel applies dropout or replay on top of *unmixed* pretraining (λ=0), with the mixed-pretraining frontier shown for reference. Both interventions shift the fine-tuning–retention frontier, but less than pretraining-time mixing alone.

C.4. 135M LoRA Experiments

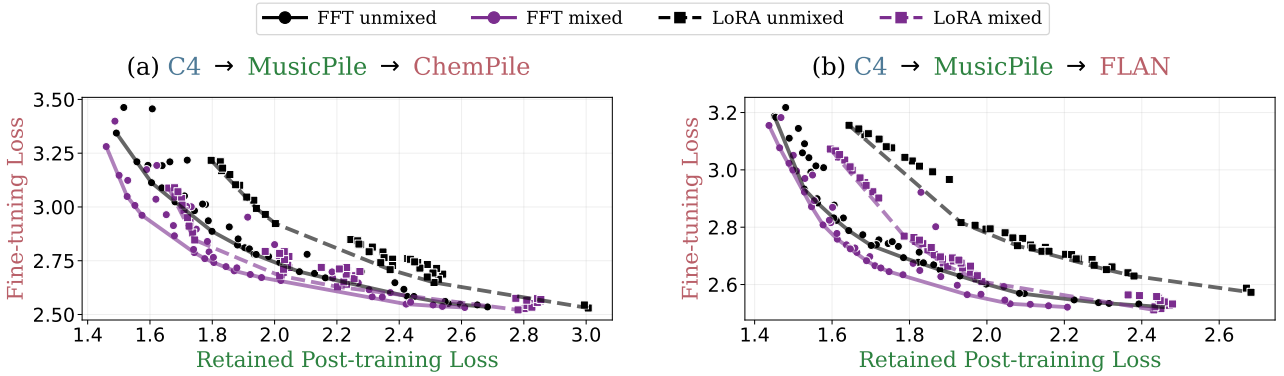


Figure 8. **FFT vs LoRA fine-tuning–retention frontiers (135M).** FFT with unmixed pretraining (black circles, solid), FFT with mixed pretraining (purple circles, solid), LoRA with unmixed pretraining (black squares, dashed), and LoRA with mixed pretraining (purple squares, dashed). Mixed pretraining improves both the FFT and LoRA frontiers in both downstream settings.

C.5. 1B Experiments

Black denotes the frontier obtained from unmixed pretraining; **purple** denotes mixed pretraining.

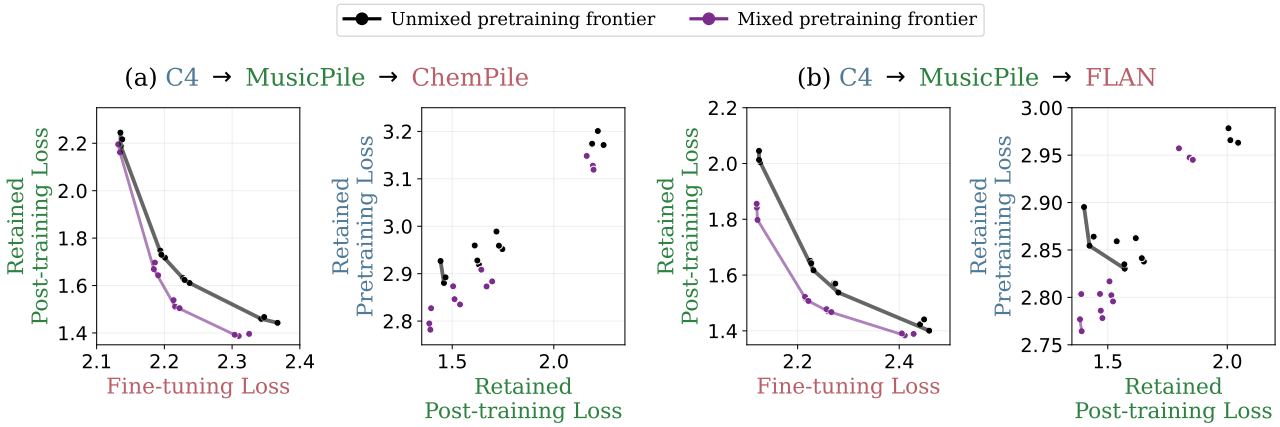


Figure 9. **Mixing frontiers at 1B, MusicPile post-training pipelines.** As at 135M, mixed pretraining consistently shifts the frontier toward lower retained post-training loss, lower retained pretraining loss, and lower downstream fine-tuning loss.

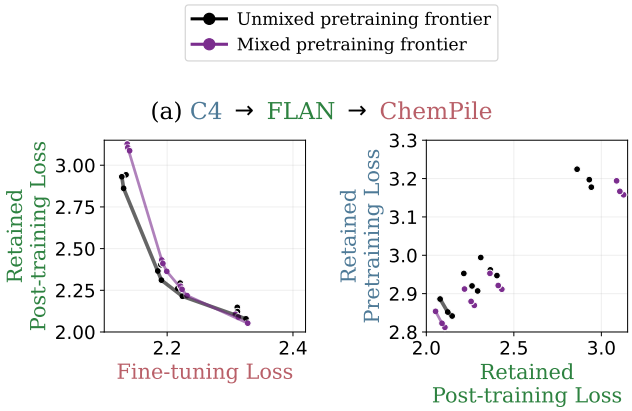


Figure 10. **Mixing frontiers at 1B, FLAN post-training pipeline.** Left: retained post-training loss vs. fine-tuning loss. Right: retained pretraining loss (C4) vs. retained post-training loss.

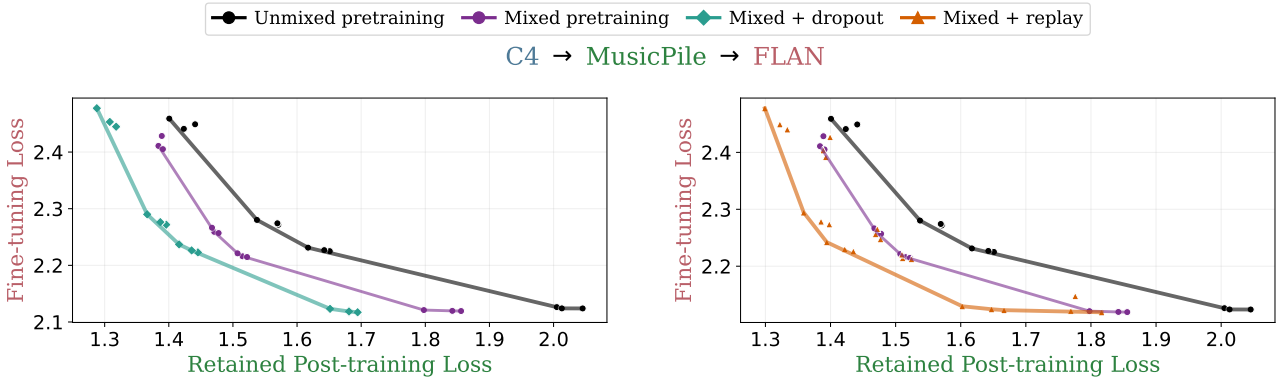


Figure 11. **Replay and dropout provide complementary gains on top of mixed pretraining at 1B.** Both dropout and replay further shift the fine-tuning–retention frontier beyond mixed pretraining alone.

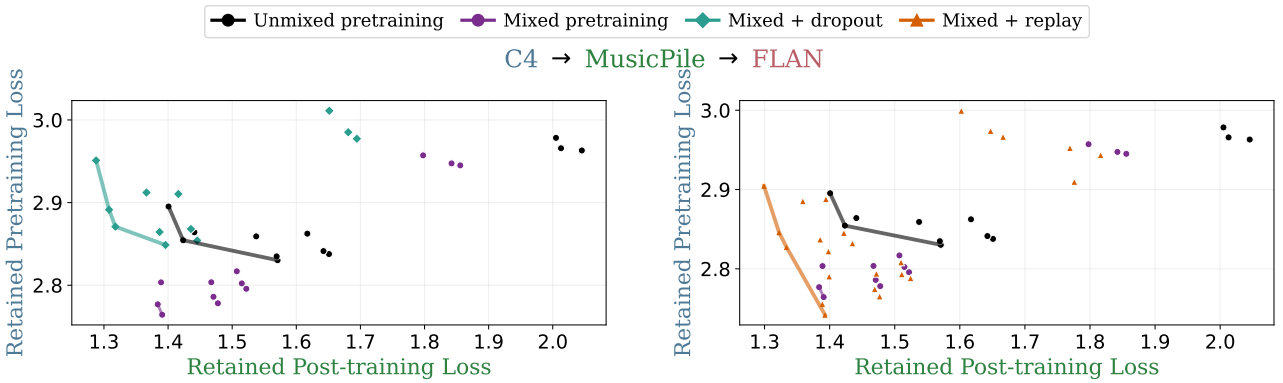


Figure 12. **Dropout and replay on top of mixed pretraining, broader pretraining retention (1B).** At 1B, dropout at the rate we swept degrades C4 retention relative to mixed pretraining alone (we attribute this to insufficient tuning of the dropout rate); replay continues to preserve and often improves C4 retention.

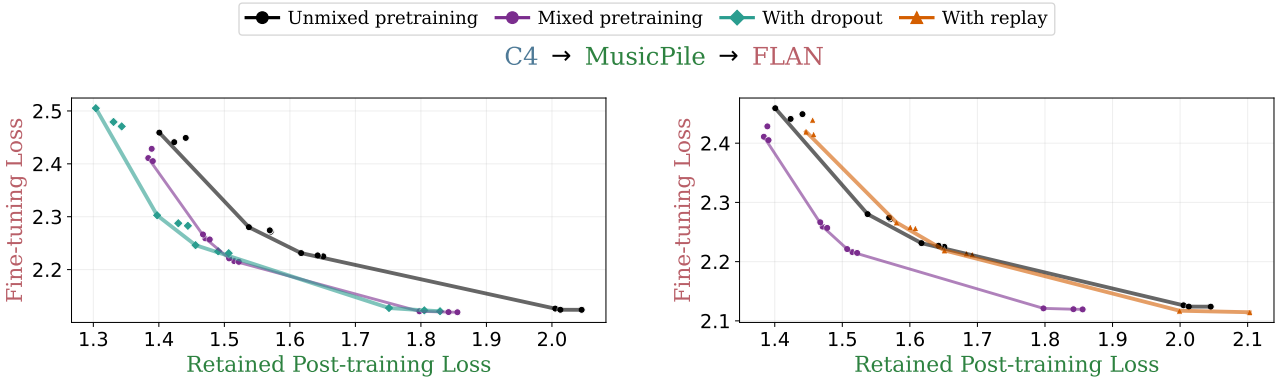


Figure 13. **Dropout and replay applied without pretraining-time mixing (1B).** At 1B, dropout continues to shift the fine-tuning–retention frontier relative to the unmixed baseline, while replay alone has a weaker effect.

D. Theoretical Analysis

D.1. Preliminaries and Setup

Model and data distribution We consider a two-layer linear network $\theta = \mathbf{W_1}\mathbf{W_2}\mathbf{x}$ on a series of regression problems using the squared loss. In particular, the problems take the form of $\mathcal{L}_t(\theta) = \mathbf{E}_{x\sim\mathcal{D}_t}[||\theta x - \mathbf{A}^t x||_2^2]$, where $t \in \{\text{pre}, \text{post}, \text{ft}\}$ index the current stage of training. Here, $\mathcal{D}_t$ denotes the input distribution and the ground-truth outputs are generated as $\mathbf{A}^t\mathbf{X}$, where $\mathbf{X} \sim \mathcal{D}_t$. Following the analysis in Springer et al. (2025), we consider the singular values and vectors of $\mathbf{A}^t$ as the learned features for the training task t.

Assumption D.1 (Simultaneous Diagonalizability)**.** There are orthonormal matrices $\mathbf{U}, \mathbf{V}$ such that for $t \in \{\text{pre}, \text{post}, \text{ft}\}$ we can write $\mathbf{A}^t = \mathbf{U}\mathbf{\Sigma_t}\mathbf{V}^\top$, where all the $\mathbf{\Sigma_t}$ are diagonal matrices.

In order to model transfer and interference between the distributions, we will next specify a structure on the relationships between the different features. We first assume the presence of *invariant features*, capturing common linguistic capabilities that are broadly applicable across domains and tasks. Across these definitions, we assume a consistent indexing of the singular values (although the ordering of the singular values in terms of their magnitude may be different). In the following we will denote the singular values of the the task covariances interchangeably with the notations $\sigma_i^t = (\Sigma_t)_{ii}$ to denote the ground-truth value of the feature.

Definition D.2 (Invariant Features)**.** For $i \in [1, n-2k]$, we have that $(\mathbf{\Sigma}_{\text{pre}})_{ii} = (\mathbf{\Sigma}_{\text{post}})_{ii} = (\mathbf{\Sigma}_{\text{ft}})_{ii}$. For clarity and to emphasize their static nature, we will often denote the values of the invariant features as $\sigma_1^{\text{inv}}, ..., \sigma_{n-2k}^{\text{inv}}$. For conciseness, we will also use $d_{\text{invariant}} = n - 2k$ to refer to the number of invariant features.

In addition to these highly general features, we also consider the features through which the model may learn more domain specific information. We consider that such specialization can be implemented through one of two pathways:

Definition D.3 (Inconsistent Features)**.** We define feature (indexed by i) to be inconsistent if we have that $(\mathbf{\Sigma}_{\text{post}})_{ii} > (\mathbf{\Sigma}_{\text{pre}})_{ii}$ and $(\mathbf{\Sigma}_{\text{post}})_{ii} - (\mathbf{\Sigma}_{\text{pre}})_{ii} > c_{\text{mis}}$

Inconsistent features therefore incur a tradeoff between reducing loss on $\mathcal{D}_{\text{post}}$ and preserving performance on $\mathcal{D}_{\text{pre}}$. Finally, we introduce *specialized features*, which do not incur such a tradeoff.

Definition D.4 (Specialized features)**.** We consider feature i is **specialized** if we have that $(\mathbf{V}^\top)_i\mathbf{x} = 0$ and that $(\mathbf{\Sigma}_{\text{post}})_{ii} > 0$. For simplicity we will assume that all specialized features take the same value of $(\mathbf{\Sigma}_{\text{post}})_{ii} = \beta$. We will also consider that $\beta > \frac{1}{2}c_{\text{mis}}$

Intuitively, $\mathcal{D}_{\text{pre}}$ and $\mathcal{D}_{\text{ft}}$ have no covariance along the specialized feature directions. As we will show, this results in gradient steps taken along them causing no interference along these directions. However, as a result of their zero-covariance, these features are also impossible to learn without explicitly seeing the post-training data.

Downstream Tuning Task We consider that the downstream tuning task is relatively more similar to the pretraining task than the post-training task. As such, we consider that the inputs are sampled according to $x \sim \mathcal{N}(0, \mathbf{I}_{n-k})$ (i.e. it doesn't not activate the specialized features). As previously, we have that the singular values corresponding to the invariant features remain constant. We will also consider that $\mathbf{A}^{\text{post}}$ and $\mathbf{A}^{\text{ft}}$ diverge on the inconsistent feature. Concretely we have that $(\mathbf{\Sigma}_{\text{post}})_{ii} > (\mathbf{\Sigma}_{\text{ft}})_{ii}$ and $(\mathbf{\Sigma}_{\text{post}})_{ii} - (\mathbf{\Sigma}_{\text{ft}})_{ii} > c_{\text{mis}}$.

Mixed Training We parameterize the mixed distribution by a parameter α and train on the distribution $\mathcal{D}_{\alpha,\text{mixed}} = (1-\alpha)\mathcal{D}_{\text{pre}} + \alpha\mathcal{D}_{\text{post}}$. As all distributions in our setting have mean zero, we have that the covariance matrix of this mixture of Gaussian distributions is $(1-\alpha)\mathbf{\Sigma}_{\text{pre}} + \alpha\mathbf{\Sigma}_{\text{post}}$.

Assumption D.5 (Invariant Features are High Magnitude)**.** We consider that the invariant features are higher magnitude than the specialized features and the inconsistent features, concretely:

$$\sigma_i^{\text{pre}} > \sigma_j^{\text{pre}}$$

$\forall i \in [0, d_{\text{invariant}}]$ and $\forall j \in (d_{\text{invariant}}, n)$. We make a similar assumption on the relationship between the invariant features and the specialized features, concretely:

$$\sigma_i^{\text{pre}} > \sigma_j^{\text{post}}$$

$\forall i \in [0, d_{\text{invariant}}]$ and $\forall j \in (d_{\text{invariant}}, n)$. This intuitively encodes that the invariant features correspond to the strongest directions in the data.

Assumption D.6 (Sufficient Specialized Mixing)**.** We assume that there exists $\alpha \in [0, 1]$

$$\alpha\beta > (1-\alpha)\sigma_i^{\text{pre}} + \alpha\sigma_i^{\text{post}} \;\; \forall i \in [d_{\text{invariant}}, d_{\text{invariant}} + k]$$

Intuitively, Assumption D.6 suggests that there exists a mixing ratio such that the mixing specialized features become more salient than the inconsistent features. However, this mixing ratio need not be high if the strength of the specialized feature is high in the covariance of $\mathcal{D}_{\text{post}}$.

D.2. Analysis of Initial Pretraining

Here, we will study the dynamics of the pretraining stage. We first introduce an important result on the sequential learning dynamics of features in two layer linear networks (Gidel et al., 2019). For a given pretraining task where $\mathbf{X}, \mathbf{Y}$ represent the inputs and outputs, respectively. For a given task, we define that $\mathbf{\Sigma}_{xy} = \frac{1}{n}\mathbf{X}^\top\mathbf{Y}$ and $\mathbf{\Sigma}_x = \frac{1}{n}\mathbf{X}^\top\mathbf{X}$. We will write that $\mathbf{\Sigma}_{xy} = \sum_{i=1}^{R_{xy}} \sigma_i \mathbf{u_i}\mathbf{v_i}^T$, where R_{xy} is the rank of $\mathbf{\Sigma_{xy}}$. We will also assume that:

Assumption D.7 (Joint Decomposition)**.** There exist orthogonal matrices $\mathbf{U}, \mathbf{V}$ such that

$$\mathbf{\Sigma}_{xy} = \mathbf{U}\mathbf{D}_{xy}\mathbf{V}^\top, \mathbf{\Sigma}_{xx} = \mathbf{U}\mathbf{D}_{xx}\mathbf{U}^\top \tag{1}$$

and we will denote the singular values of $\mathbf{\Sigma}_{xy}$ as $\sigma_1, ..., \sigma_{R_{xy}}$ and that the diagonal entries of $\mathbf{D}_{xx}$ as $\lambda_1, ..., \lambda_{R_x} = 1$.

Next, we will characterize the initialization scale of model before pretraining. Following (Springer et al., 2025), we have the following initialization:

Assumption D.8 (Pretrained Initialization Scale)**.** Let $(\mathbf{W}_1(0), \mathbf{W}_2(0))$ be the parameters at initialization. Then we have that $\mathbf{W}_1(0) = \mathbf{W}_2(0) = \exp(-\mathcal{T})\mathbf{I_d}$.

Essentially, Assumption D.7 requires that the model parameters are close, but not exactly 0 which yields *sequential feature learning*. We next explicitly re-state the result from (Gidel et al., 2019).

Theorem D.9 (Sequential Learning of Features (Gidel et al., 2019; Springer et al., 2025))**.** *Suppose $\mathbf{W}_1, \mathbf{W}_2$ obey the initialization in Assumption D.8 and the pretraining task obeys Assumption D.7. Then there exist times $t_1, ..., t_r$ such that*

$$||\mathbf{W}_1(t_i) - \mathbf{U}(\Sigma_{:i})^{\frac{1}{2}}||_F \leq \exp(-C\tau)$$

$$||\mathbf{W}_2(t_i) - (\Sigma_{:i})^{\frac{1}{2}}\mathbf{V}^\top||_F \leq \exp(-C\tau)$$

Where $\Sigma_{:i}$ is defined to be diag$(\sigma_1, ...\sigma_i, 0, ...0)$, equivalently the rank i approximation of diag$(\sigma_1, ..., \sigma_{R_{xy}})$.

Conceptually, Theorem D.9 demonstrates that during the pretraining process, $\mathbf{W}_1\mathbf{W}_2$ learn features in decreasing order of their of the singular value of Σ_{xy}. Next, we will apply this result in order to compare the features learned during mixed and non-mixed pretraining.

Theorem D.10 (Only Mixing Learns Specialized Features)**.** *Let $\theta^{gen}(t) = \mathbf{W}_1^{gen}(t)\mathbf{W}_2^{gen}(t)$ be the parameters learned when pretraining only on $\mathcal{D}_{gen}$ and $\theta^{mixed}(t) = \mathbf{W}_1^{mixed}(t)\mathbf{W}_2^{mixed}(t)$. Denote $\mathbf{u}_{spec}, \mathbf{v}_{spec}$ to be the right and left singular values corresponding to the specialized feature. Then, there exists a time t such that*

$$\begin{aligned}
||\mathbf{W}_1^{(unmixed)} - \mathbf{U}(\mathbf{\Sigma}^{(unmixed)})^{\frac{1}{2}}||_F &\leq \exp(-C\tau) \\
||\mathbf{W}_2^{(unmixed)} - (\mathbf{\Sigma}^{(unmixed)})^{\frac{1}{2}}\mathbf{V}^\top||_F &\leq \exp(-C\tau) \\
||\mathbf{W}_1^{(unmixed)} - \mathbf{U}(\mathbf{\Sigma}^{(unmixed)})^{\frac{1}{2}}||_F &\leq \exp(-C\tau) \\
||\mathbf{W}_2^{(unmixed)} - (\mathbf{\Sigma}^{(unmixed)})^{\frac{1}{2}}\mathbf{V}^\top||_F &\leq \exp(-C\tau)
\end{aligned}$$

where $\mathbf{\Sigma}_{mixed} = diag(\sigma_1^{inv}, ...\sigma_{n-2k}^{inv}, \mathbf{0}_k, \alpha\beta, ..., \alpha\beta)$ and $\mathbf{\Sigma}_{unmixed} = diag(\sigma_1^{inv}, ..., \sigma_{n-2k}^{inv}, \sigma_1^{post}, ..., \sigma_k^{post}, \mathbf{0}_k)$

Proof. This is a relatively straightforward application of Theorem D.9. Denote $(\mathbf{X}^{(\text{mixed})}, \mathbf{Y}^{(\text{mixed})})$ as the data used for mixed pretraining and $\mathbf{\Sigma}_{xy}^{(\text{mixed})} = \frac{1}{n}(\mathbf{X}^{(\text{mixed})})^\top \mathbf{Y}^{(\text{mixed})}$. We have that $\mathbf{\Sigma}_{xy}^{(\text{mixed})} = (1-\alpha)\mathbf{\Sigma}_{xy}^{\text{pre}} + \alpha \mathbf{\Sigma}_{xy}^{\text{spec}}$ which follows from the fact that the Σ_{xy} are submatrices of the covariance matrix of Gaussian random vectors. By Theorem D.9, we have that the features are learned in order of the singular values of $\mathbf{\Sigma_{xy}}$. By Assumptions D.5 and D.6, we have that the top $n-k$ singular values of $\mathbf{\Sigma}_{xy}^{(\text{mixed})}$ are the $n-2k$ shared features and the k specialized features. Define $\mathbf{\Sigma}_{:n-k}^{(\text{mixed})} = \text{diag}(\sigma_1^{\text{inv}}, ...\sigma_k^{\text{inv}}, \mathbf{0}_k, \alpha\beta, ..., \alpha\beta)$. Applying Theorem D.9, we have that

$$||\mathbf{W}_1^{(\text{mixed})} - \mathbf{U}(\mathbf{\Sigma}_{:n-k}^{(\text{mixed})})^{\frac{1}{2}}||_F \leq \exp(-C\tau)$$
$$||\mathbf{W}_2^{(\text{mixed})} - (\mathbf{\Sigma}_{:n-k}^{(\text{mixed})})^{\frac{1}{2}}\mathbf{V}^\top||_F \leq \exp(-C\tau)$$

Repeating this analysis for unmixed training, we have that the top $n-k$ singular values of $\mathbf{\Sigma}_{xy}^{(\text{gen})}$ are the $n-2k$ shared features are the k inconsistent features. We can define $\mathbf{\Sigma}_{:n-k}^{(\text{unmixed})} = \text{diag}(\sigma_1^{\text{inv}}, ...\sigma_k^{\text{inv}}, \sigma_1^{\text{post}}, ..., \sigma_1^{\text{post}}, \mathbf{0}_k)$. Similarly, by applying Theorem D.9, we have that

$$||\mathbf{W}_1^{(\text{unmixed})} - \mathbf{U}(\mathbf{\Sigma}_{:n-k}^{(\text{unmixed})})^{\frac{1}{2}}||_F \leq \exp(-C\tau)$$
$$||\mathbf{W}_2^{(\text{unmixed})} - (\mathbf{\Sigma}_{:n-k}^{(\text{unmixed})})^{\frac{1}{2}}\mathbf{V}^\top||_F \leq \exp(-C\tau)$$

□

Intuitively, our result in Theorem D.9 demonstrates that mixing $\mathcal{D}_{\text{post}}$ during pretraining results in a pretrained initialization that has different features. Mixing learns the specialized features, while not mixing learns only the inconsistent features. In the following, we will examine the impact that these different features have on the retention of $\mathcal{D}_{\text{post}}$ during subsequent training.

D.3. Analysis of Post-Training

We now study the dynamics of the post-training process. To formalize the post-training process, we first examine the dynamics beginning from the idealized pretraining initialization (as performed by (Springer et al., 2025)). We perform the post-training stage on the regularized loss $\mathbf{E}[||\theta x - \mathbf{A}^{\text{sp}} x||_F^2] + \lambda||\theta - \theta_0||_F^2$. Observe that because $x \sim \mathcal{N}(0, \mathbf{I}_d)$, this is equivalent to $||\theta - \mathbf{A}^{\text{sp}}||_F$. We follow the assumptions on the regularity of fine-tuning established in (Springer et al., 2025).

Assumption D.11 (Bound on Parameters Throughout Training)**.**

$$||\hat{\mathbf{W}}_1^{(\text{mixed})}||_{\text{op}} \leq \sqrt{\Gamma}$$
$$||\hat{\mathbf{W}}_1^{(\text{mixed})}||_{\text{op}} \leq \sqrt{\Gamma}$$
$$||\hat{\mathbf{W}}_1^{(\text{unmixed})}||_{\text{op}} \leq \sqrt{\Gamma}$$
$$||\hat{\mathbf{W}}_1^{(\text{unmixed})}||_{\text{op}} \leq \sqrt{\Gamma}$$

Moreover, we assume that the regularization strength and the learning rates are likewise bounded.

Assumption D.12 (Bound on Learning Rate)**.**

$$4\eta(\lambda + 2)\Gamma < 1$$

Idealized Pretraining Initialization We denote the ideal initialization parameters for the mixed and unmixed cases $(\hat{\mathbf{W}}_1(0), \hat{\mathbf{W}}_2(0))$.

$$\hat{\mathbf{W}_1^{(\text{mixed})}}(0) = \mathbf{U}(\Sigma_{:n-k}^{\text{mixed}})^{\frac{1}{2}}$$
$$\hat{\mathbf{W}_2^{(\text{mixed})}}(0) = (\Sigma_{:n-k}^{\text{mixed}})^{\frac{1}{2}}\mathbf{V}^\top$$

Similarly, we have the following idealized initialization for the unmixed initialization:

$$\hat{\mathbf{W}_1^{(\text{unmixed})}}(0) = \mathbf{U}(\Sigma_{:n-k}^{\text{unmixed}})^{\frac{1}{2}}$$
$$\hat{\mathbf{W}_2^{(\text{unmixed})}}(0) = (\Sigma_{:n-k}^{\text{unmixed}})^{\frac{1}{2}}\mathbf{V}^\top$$

In the idealized setting, we can track the evolution of each singular value independently. In particular, we have the following update rules as derived in (Springer et al., 2025) (where we denote σ_i^{spec} as the i-th singular value of $\mathbf{A}^{\text{spec}}$ and likewise for $\sigma_i^{(\text{un})\text{mixed}}(t)$ as the i-th singular value at step t). In what follows, we will suppress the superscript for compactness:

$$\sigma_i(t+1) = \sigma_i(t) + 2\eta\sigma_i(t)(\eta\sigma_i(t)^2 - (\sigma_{\text{spec},i})^2) + 2\eta\lambda(\sigma_i(t)^2 - \sigma_i(0)^2) \tag{2}$$

As a result, note that when $\sigma_i^{(\text{un})\text{mixed}}(0) = 0$, $\sigma_t^{(\text{un})\text{mixed}}(t) = 0$ for all t.

Next,we will study the dynamics of the non-zero singular values (Lemma A.11 (Springer et al., 2025). We will assume that post-training is performed for a sufficient number of steps.

Assumption D.13 (Sufficient Post-Training Steps)**.** We have that the number of post-training steps (denoted by K) satisfies $K \geq \frac{1}{\lambda c_{\text{min}}} \log \frac{100\Gamma}{\epsilon}$, for a constant ϵ and where $c_{\text{min}} = \min\{(\mathbf{\Sigma}_{\text{post}})_{ii} | \mathbf{\Sigma}_{\text{post}})_{ii} \neq 0\}$ – that is the minimum, non-zero singular value.

Given these technical conditions, we now state a general result (adapted for our setting from (Springer et al., 2025)).

Lemma D.14. *When training on $\mathcal{D}_{post}$ with infinite batch size from the ideal pretraining initialization and taking sufficient number of steps K, for all $i \in rank(\theta_n(0))$, we have that*

$$|(\mathbf{U}^\top \hat{\theta}^{(un)mixed}(t)\mathbf{V})_{ii} - (\mathbf{\Sigma}_{post})_{ii}| \leq \epsilon \tag{3}$$

where $\mathbf{\Sigma}_{post}$ is such that $\mathbf{A}_{post} = \mathbf{U}\mathbf{\Sigma}_{post}\mathbf{V}^\top$.

Now, we will define the matrices $\mathbf{\Sigma}^{\text{shared,post}} = \text{diag}(\sigma_1^{\text{inv}}, ...\sigma_{n-2k}^{\text{inv}}, \sigma_{d_{\text{invariant}}+1}^{\text{post}}, ..\sigma_{d_{\text{invariant}}+k}^{\text{post}}, \mathbf{0}_k)$ and $\mathbf{\Sigma}^{\text{spec,post}} = \text{diag}(\sigma_1^{\text{inv}}, ...\sigma_{n-2k}^{\text{inv}}, \mathbf{0}_k, \sigma_{d_{\text{invariant}}+k+1}^{\text{post}}, ..\sigma_{d_{\text{invariant}}+2k}^{\text{post}})$, Intuitively, $\mathbf{\Sigma}^{\text{shared,post}}$ lowers loss on $\mathcal{D}_{\text{post}}$ by shifting the values on the shared features, while $\mathbf{\Sigma}^{\text{spec,post}}$ accomplishes this by modifying the unique features. We are now ready to state the main theorem.

Theorem D.15 (Post-training on θ^{mixed} versus θ^{unmixed})**.** *Let $\theta^{mixed}(K)$ denote the parameters after training the idealized unmixed initialization starting for K steps and let $\theta^{mixed}(K)$ be the same starting from the idealized mixed checkpoint. Then we have*

$$||\mathbf{U}^\top \theta^{mixed}\mathbf{V} - \mathbf{\Sigma}^{spec,post}||_{op} \leq \epsilon$$
$$||\mathbf{U}^\top \theta^{unmixed}\mathbf{V} - \mathbf{\Sigma}^{shared,post}||_{op} \leq \epsilon$$

Proof. This theorem follows by noting that under the idealized pretraining initialization any singular value that is 0 at initialization remains that way during the entire optimization trajectory. Note that the 0 singular values of $\mathbf{U}^\top\theta^{\text{mixed}}\mathbf{V}$ coincide with $\mathbf{\Sigma}^{\text{spec, post}}$ (the specialized features) and likewise $\mathbf{U}^\top\theta^{\text{unmixed}}\mathbf{V}$ coincide with $\mathbf{\Sigma}^{\text{shared, post}}$ (the inconsistent features).

This implies that we have

$$\max_{i\in[1,n]} |(\mathbf{U}^\top\theta^{\text{mixed}}\mathbf{V})_{ii} - (\mathbf{\Sigma}^{\text{spec,post}})_{ii}| \leq \max_{i\in\text{rank}(\theta)} |(\mathbf{U}^\top\theta^{\text{mixed}}\mathbf{V})_{ii} - (\mathbf{\Sigma}^{\text{spec,post}})_{ii}|$$
$$\max_{i\in[1,n]} |(\mathbf{U}^\top\theta^{\text{unmixed}}\mathbf{V})_{ii} - (\mathbf{\Sigma}^{\text{shared,post}})_{ii}| \leq \max_{i\in\text{rank}(\theta)} |(\mathbf{U}^\top\theta^{\text{mixed}}\mathbf{V})_{ii} - (\mathbf{\Sigma}^{\text{shared,post}})_{ii}|$$

Now, applying the result from Lemma D.14 yields the desired claim. □

D.4. Analysis of Downstream Adaptation

In the previous section, we characterized the impact of post-training from a mixed versus an unmixed initialization, demonstrating that different features are used to minimize the loss on $\mathcal{D}_{\text{post}}$. In this section, we study how these different features impact the ultimate retention of $\mathcal{D}_{\text{post}}$. We first establish that the singular values corresponding to directions in which there is no covariance remain unchanged throughout the downstream fine-tuning stage.

Lemma D.16. *Consider performing downstream unregularized fine-tuning on $\mathcal{D}_{ft}$. If $x \sim \mathcal{D}_{ft}$ has 0 covariance along a singular direction, the corresponding singular value remains unchanged throughout downstream adaptation.*

Proof. To see this, note that the gradient updates for $\mathbf{W}_1$ and $\mathbf{W}_2$ take the following form:

$$\mathbf{W}_1(k+1) = \mathbf{W}_1(k) - 2\eta(\mathbf{W}_1(k)\mathbf{W}_2(k) - \mathbf{A}^{\text{spec}})\Sigma_x\mathbf{W}_2^\top$$
$$\mathbf{W}_2(k+1) = \mathbf{W}_2(k) - 2\eta\mathbf{W}_1(k)\Sigma_x(\mathbf{W}_1(k)\mathbf{W}_2(k) - \mathbf{A}^{\text{spec}})$$

Here, we have that $\Sigma_x$denotes the covariance of the input data x. Thus, along any singular direction in which the data has 0 variance, the $\mathbf{\Sigma}_x$ term will project the gradient to 0. Therefore, the singular values on such directions must also remain unchanged. □

We consider performing downstream adaptation by taking steps using unregularized gradient descent on $\mathcal{D}_{\text{ft}}$ and show the following result.

Theorem D.17. *Consider performing K steps of gradient descent on the downstream finetuning dataset beginning from the intializations $\theta^{\text{post, mixed}}(K)$ and let $\theta^{\text{FT,mixed}}(K)$, $\theta^{\text{FT,unmixed}}(K)$ $\theta^{\text{post, unmixed}}(K)$ denote the final parameters. Let $\Delta_{\text{unmixed}} = \mathcal{L}(\theta^{\text{FT, mixed}}; \mathcal{D}_{\text{post}}) - \mathcal{L}(\theta^{\text{post, mixed}}(K); \mathcal{D}_{\text{post}})$ and likewise $\Delta_{\text{mixed}} = \mathcal{L}(\theta^{\text{FT, unmixed}}; \mathcal{D}_{\text{post}}) - \mathcal{L}(\theta^{\text{post,unmixed}}(K); \mathcal{D}_{\text{post}})$. Then we have that $\Delta_{\text{unmixed}} > \Delta_{\text{mixed}}$.*

Proof. As the invariant features take the same values, they will not move during the downstream adaptation. Moreover, due to the Lemma D.16, we also have that the specialized features will not change during the the downstream fine-tuning. This implies that $\Delta_{\text{unmixed}} = 0$. Next we will examine the changes induced by downstream training on the unmixed models. Observe that we have that the $\mathcal{L}(\theta; \mathcal{D}_{\text{post}}) = ||\theta - \mathbf{A}^{\text{spec}}||_F^2$. We will define the following matrices $\Sigma_{\text{FT}}^{(\text{unmixed})} = \text{diag}(\sigma_1^{\text{inv}}, ..., \sigma_{d_{\text{invariant}}}^{\text{inv}}, \sigma_{d_{\text{invariant}}+1}^{\text{ft}}, ...\sigma_{\text{d}_{\text{invariant}}+k}^{\text{ft}}\mathbf{0}_k)$ and note that Lemma D.14 gives us that

$$||\mathbf{U}^\top\theta^{(\text{unmixed})_{\text{FT}}}\mathbf{V} - \Sigma_{\text{FT}}^{(\text{unmixed})}||_{\text{op}} \leq \epsilon$$

The loss function we use here is simply the squared difference of the singular values. Thus,we can upper bound:

$$\mathcal{L}(\theta_{\text{post}}^{\text{unmixed}}; \mathcal{D}_{\text{spec}}) \leq k(\beta+\epsilon)^2 + k\epsilon^2$$

and likewise lower bound

$$\mathcal{L}(\theta_{\text{ft}}^{\text{unmixed}}; \mathcal{D}_{\text{spec}}) \geq k(\beta-\epsilon)^2 + k(c_{\text{mis}}-\epsilon)^2$$

Then,we can lower bound $\Delta_{\text{unmixed}} \geq k[(\beta+\epsilon)^2 - (\beta-\epsilon)^2] + k[(c_{\text{mis}}-\epsilon)^2 - \epsilon^2] = k[4\beta\epsilon - 2c_{\text{mis}}\epsilon + c_{mis}^2]$. From the condition that $\beta > \frac{1}{2}c_{\text{mis}}$ and the positivity of $(c_{\text{mis}})^2$, we thus have that $\Delta_{\text{unmixed}} > 0$, which is what we wanted to show.

□