MIX EARLY, FORGET LESS: DATA MIXING DURING PRETRAINING BUILDS RESISTANCE TO FORGETTING

Anonymous authors
Paper under double-blind review

ABSTRACT

After web-scale pretraining, language models are often further trained to add domain skills and behaviors, and later fine-tuned to ingest new data or meet specific downstream requirements. A persistent challenge in such sequential pipelines is catastrophic forgetting: later training can degrade previously learned capabilities. Prior mitigation strategies largely focus on fine-tuning-time interventions and treat the upstream training procedure as fixed. We show that upstream data placement matters: mixing a small amount (a few % of the overall pretraining mixture) of capability-relevant data into pretraining builds resistance to forgetting, yielding substantially better learning–retention tradeoffs under subsequent training than introducing the domain only after pretraining. We demonstrate this effect across multiple settings, including specialized domain adaptation and instruction tuning. We also study algorithmic choices during continual pretraining and find that dropout and data replay provide additional gains that are consistently complementary to pretraining-time mixing.

1 INTRODUCTION

Modern language models are products of long training lifecycles. After broad pretraining on web-scale data, developers commonly run additional stages of continual pretraining to specialize models for domains like coding (Rozière et al., 2024) and mathematics (Shao et al., 2024), as well as behavioral tuning for instruction following and safety. After pretraining, these models are often further adapted for end tasks—whether to incorporate new knowledge or tailor the model to a particular application. In this setting, *catastrophic forgetting* (McCloskey & Cohen, 1989) is a central obstacle: subsequent training can cause sharp and unpredictable degradation of previously learned capabilities. For example, prior work finds that safety-aligned models can lose safety behaviors after subsequent training on seemingly benign objectives (such as mathematics) (Qi et al., 2023; Alssum et al., 2025; Ponkshe et al., 2025). This raises a basic question: which aspects of a model's path to specialization determine the robustness of its capabilities—whether they are retained or forgotten under subsequent training?

We study a canonical three-stage lifecycle: a base model is first pretrained on general web data, then specialized to a target domain via continual pre-training (CPT), and finally subjected to downstream fine-tuning (FT) on a different task. Our focus is the learning–retention tradeoff: how much specialized capability is retained after later training, at a given level of performance on the downstream task. While failures in retention are well-documented, most mitigation strategies aim to reduce interference during fine-tuning—such as replay (Bethune et al., 2025), elastic weight consolidation (Jhajj & Lin, 2025), or parameter-efficient isolation (Lin et al., 2025)—often treating the upstream pretraining and specialization procedure as fixed. In doing so, they primarily target fine-tuning dynamics rather than the upstream training choices that determine how specialized knowledge is represented. In this work, we turn the focus of our investigation to data mixing during pre-training, and provide empirical evidence and theoretical support that long-term retention depends on both the base model and the model's path to specialization.

To understand what makes a specialized checkpoint more or less robust to later adaptation, we study how the placement of capability-relevant data along the path to specialization affects forgetting. Across MusicPile and FLAN specialization followed by downstream fine-tuning on ChemPile, we find that introducing domain data during pretraining—rather than only after pretraining via

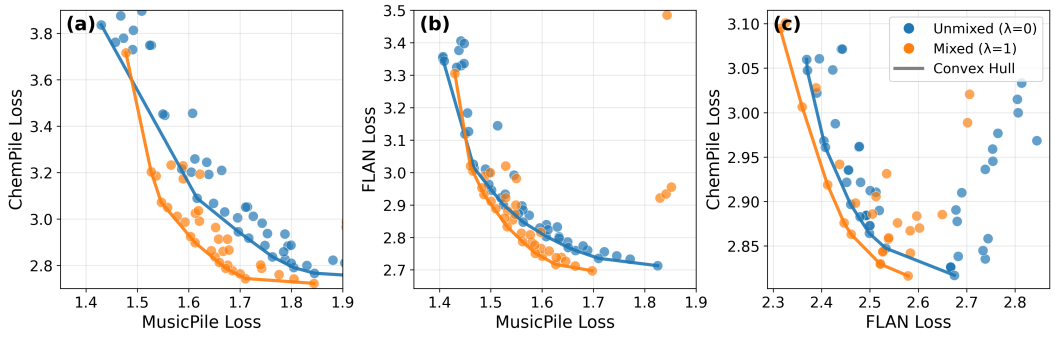

Figure 1: **Data mixing improves long-term retention across post-training pipelines.** Each point is a separate Stage-2 CPT run (different hyperparameters); solid lines trace the Pareto frontier for each initialization. **(a,b)** Starting from either an *unmixed* base model (λ=0, pretrained on C4) or a *mixed* base model (λ=1, pretrained on C4 with MusicPile mixed in), we perform CPT on MusicPile and then apply Stage-3 fine-tuning on **ChemPile** (a) or **FLAN** (b), reusing the *same* CPT checkpoints across panels. **(c)** A separate pipeline where the specialized domain is **FLAN**: we pretrain on C4 with/without FLAN mixing, CPT on FLAN, then fine-tune on **ChemPile** (task-induced forgetting). Across all setups, mixed pretraining shifts the frontier down/left.

CPT—strongly improves long-term retention. Mixing a small fraction of domain data into pretraining consistently improves the attainable learning–retention tradeoff relative to end-loaded specialization. This challenges the intuition that models forget early training data (recency bias) and should therefore encounter specialized knowledge only at the end (Wei et al., 2025). Moreover, the advantage of mixing can be latent: it may not improve immediate post-CPT performance, yet it yields substantially better retention after subsequent training. In Section 4.2, we examine how much of the specialized domain to mix into pretraining and if mixing remains optimal under a fixed compute budget.

We then turn our attention to studying the impact of *algorithmic choices during CPT* on the learning–retention tradeoff. One natural hypothesis is that parameter-efficient methods like LoRA (Hu et al., 2021), which restrict updates to a low-dimensional subspace, might reduce interference and improve retention under subsequent fine-tuning (Biderman et al., 2024). However, we find that LoRA during CPT not only learns less but also exhibits greater degradation during subsequent fine-tuning (see Section 4.3.3). Instead, we find that simpler techniques—dropout and pretraining data replay—offer a superior learning–retention tradeoff (see Sections 4.3.1 and 4.3.2). Finally, while one might expect CPT-time regularization to substitute for pretraining mixing, we find the benefits are complementary: combining these interventions with mixing consistently improves the attainable tradeoff relative to using them without. Thus, CPT-time interventions can help, but they do not eliminate the advantage conferred by mixing data during pretraining.

Finally, we theoretically investigate a simplified two-layer linear network to characterize the representational properties that modulate forgetting under subsequent training. We show that when domain knowledge is stored in features orthogonal to other tasks, it experiences fewer interfering updates in later training stages. Moreover, we prove that mixing specialized data during pretraining encourages such orthogonal features to be learned. In contrast, when specialized data is incorporated only during CPT, domain knowledge is stored by distorting pre-existing features shared by other tasks, and therefore experiences greater interference under later training.

Overall, our results show that training-time decisions can substantially influence the attainable learning–retention tradeoff under subsequent training. In particular, mixing specialized data into pretraining dramatically increases robustness to future forgetting. While regularization and replay during continual pretraining can partially improve retention, they do not eliminate the fragility of capabilities introduced only after pretraining. These findings offer concrete guidance for how model developers can preemptively mitigate catastrophic forgetting—an axis that remains relatively underexplored compared to fine-tuning-time interventions. Finally, our results raise the question of whether novel training algorithms or regularization strategies can replicate the benefits of pretraining data mixing without requiring modifications to pretraining.

2 Related Works.

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 phenomenon shows up in modern training pipelines. For example, instruction tuning and RLHF can trade off against preexisting capabilities, an effect often discussed as an "alignment tax" Ouyang et al. (2022). Relatedly, 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). Forgetting-like tradeoffs also appear in adjacent settings such as knowledge editing Nishi et al. (2025) and unlearning Maini et al. (2024a). Beyond documenting the effect, recent work has started to map how training choices shape its severity: for instance, 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). In this paper, we focus on catastrophic forgetting in *sequential domain adaptation*, and ask a finer-grained question: what properties of a domain-adapted checkpoint determine whether domain knowledge persists under subsequent training?

Training Dynamics of LLMs A significant amount of prior works aim to characterize the learning dynamics of pretraining. Leybzon & Kervadec (2024) find that pretraining-time memorization undergoes cycles of learning and forgetting. Similarly, Wei et al. (2025) that memorized content seen earlier in pretraining can be diluted throughout training, while content seen later in pretraining remains more easily accessible. Other works have studied the dynamics of multiple stages of training. Qi et al. (2025); Liu et al. (2025) study the role of mid-training in LLM pipelines, showing that it helps bridge the distributional differences between pre and post-training. In this work, we study the problem of retaining specialized domain knowledge and demonstrate that mixing data in pre-training can have important benefits.

Pretraining Interventions Prior works examine pre-training time interventions for enforicng desired downstream model 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 earilier in pre-training. Beyond safety, Maini et al. (2024b) shows that rephrasing web data can improve loss and zero-shot capabilities. While these works incorporate downstream tasks during pre-training, they extensively modify the pre-training corpus by incorporating data-augmentations and filtering of the dataset. We show that *merely* mixing domain-specific data during the initial pre-training phase can mitigate catastrophic forgetting.

3 Preliminaries and Setting

Our focus is to train models that are resistant to forgetting under *subsequent adaptation*. Concretely, a model developer may (1) pretrain a base model on a broad corpus, (2) adapt it to a specialized domain, and then (3) ship the model to downstream users who further fine-tune it for their own purposes. We model these stages with three datasets: $\mathcal{D}_{\text{gen}}$ (general pretraining), $\mathcal{D}_{\text{spec}}$ (specialized domain), and $\mathcal{D}_{\text{ft}}$ (downstream adaptation). The specialized corpus is assumed to be much smaller than the general corpus, reflecting the practical regime where domain data is relatively scarce (e.g., $|\mathcal{D}_{\text{spec}}|/|\mathcal{D}_{\text{gen}}|$ is on the order of a few percent or less).

We compare three strategies for incorporating $\mathcal{D}_{\text{spec}}$: pretraining-time mixing (including $\mathcal{D}_{\text{spec}}$ during training on $\mathcal{D}_{\text{gen}}$), post-hoc continual pretraining (CPT) on $\mathcal{D}_{\text{spec}}$ after pretraining, and hybrids that do both. The central question is whether these choices change the tradeoff between downstream performance on $\mathcal{D}_{\text{ft}}$ and retention of specialized competence on $\mathcal{D}_{\text{spec}}$.

We write $\mathcal{L}(\theta; \mathcal{D})$ for the loss of parameters θ evaluated on dataset $\mathcal{D}$. All losses are computed on held-out splits of the corresponding datasets. Retention is measured by the specialized loss after subsequent adaptation, and forgetting can be summarized by the increase in $\mathcal{L}(\cdot; \mathcal{D}_{\text{spec}})$ induced by training on $\mathcal{D}_{\text{ft}}$.

Stage 1: Mixed pretraining. In mixed pretraining, the model is trained on a mixture of a general corpus $\mathcal{D}_{\text{gen}}$ and a fixed amount of specialized data from $\mathcal{D}_{\text{spec}}$. Concretely, we pretrain on all tokens in $\mathcal{D}_{\text{gen}}$ together with a λ-fraction of $\mathcal{D}_{\text{spec}}$ (measured in tokens), which we denote by $\lambda\mathcal{D}_{\text{spec}}$. We

refer to the resulting mixed pretraining corpus as

$$\mathcal{D}_{\text{mix}}(\lambda) \;=\; \mathcal{D}_{\text{gen}} \;+\; (\lambda \mathcal{D}_{\text{spec}}),$$

where $\lambda \in [0, 1]$ controls how much specialized data is included. In practice, training batches are formed by interleaving examples from $\mathcal{D}_{\text{gen}}$ and $\mathcal{D}_{\text{spec}}$ so that, over the full run, the total number of specialized tokens equals a λ fraction of the available specialized corpus. We denote the parameters after mixed pretraining by θ_{pt}.

Stage 2: Continual Pretraining (CPT). Starting from the pretrained parameters θ_{pt}, we continue training exclusively on the specialized corpus $\mathcal{D}_{\text{spec}}$; this adaptation can induce regression on previously learned general capabilities. In the default setting, we apply early stopping and train on $\mathcal{D}_{\text{spec}}$ until the validation loss stops improving (with a maximum budget of 2B tokens), yielding

$$\theta_{\text{pt}} \xrightarrow{\text{CPT on } \mathcal{D}_{\text{spec}}} \theta_{\text{cpt}}.$$

We measure specialized performance immediately afterward by the *immediate loss* $\mathcal{L}_{\text{im}} := \mathcal{L}(\theta_{\text{cpt}}; \mathcal{D}_{\text{spec}})$.

Stage 3: Downstream Fine-tuning. In practice, models are rarely "done" after CPT: downstream users often further adapt them to new objectives, such as instruction tuning or continued domain adaptation on a user-specific corpus. We model this as downstream fine-tuning on a dataset $\mathcal{D}_{\text{ft}}$,

$$\theta_{\text{cpt}} \xrightarrow{\text{FT on } \mathcal{D}_{\text{ft}}} \theta_{\text{ft}},$$

and define the downstream loss $\mathcal{L}_{\text{ft}} := \mathcal{L}(\theta_{\text{ft}}; \mathcal{D}_{\text{ft}})$.

Further training can degrade previously acquired capabilities, so the specialized loss on $\mathcal{D}_{\text{spec}}$ may increase relative to $\mathcal{L}_{\text{im}}$. We therefore measure specialized retention after downstream adaptation by the *retained loss* $\mathcal{L}_{\text{ret}} := \mathcal{L}(\theta_{\text{ft}}; \mathcal{D}_{\text{spec}})$. Our main goal is to understand how choices in how $\mathcal{D}_{\text{spec}}$ is incorporated in Stages 1–2 affect $\mathcal{L}_{\text{ret}}$ under subsequent downstream fine-tuning; we say a model is more *resistant to forgetting* if it achieves lower $\mathcal{L}_{\text{ret}}$ at comparable downstream performance.

3.1 Concrete Experimental Details

Model and token budgets. Across all experiments, we use SmolLM2-135M and perform Stage 1 pretraining on a fixed 10B-token stream from $\mathcal{D}_{\text{gen}}$, together with an additional λ-fraction of the available specialized corpus $\mathcal{D}_{\text{spec}}$. In Stage 2, we apply early stopping on $\mathcal{D}_{\text{spec}}$ with a maximum budget of 2B tokens, except in compute-matched ablations where the $\mathcal{D}_{\text{spec}}$ budget is fixed by construction. In Stage 3, we fine-tune on $\mathcal{D}_{\text{ft}}$ for a fixed token budget specific to each setting (200M tokens in our main ChemPile fine-tuning runs). See Appendix A.4 for more details.

Dataset instantiations. We study two instantiations of the three-stage pipeline. In all cases, the general pretraining corpus is $\mathcal{D}_{\text{gen}}$ (10B tokens subsampled from C4), and we vary how $(\mathcal{D}_{\text{spec}}, \mathcal{D}_{\text{ft}})$ are instantiated.

Specialized-domain setting. We set $\mathcal{D}_{\text{spec}}$ to MusicPile and evaluate retention under two downstream adaptation regimes by setting $\mathcal{D}_{\text{ft}}$ to either more C4, FLAN (instruction tuning), or ChemPile (continued domain training). For MusicPile, we consider varying-sized subsets to study data scarcity (30M–300M tokens).

Instruction-tuning setting. We set $\mathcal{D}_{\text{spec}}$ to FLAN (Stage 2 instruction tuning) and set $\mathcal{D}_{\text{ft}}$ to ChemPile (Stage 3 domain fine-tuning). This setting models benign downstream fine-tuning for domain capability that can nevertheless erode instruction-following or behavioral tuning acquired during the prior stage.

3.2 Evaluation Methodology

Loss frontiers. Sweeping CPT methods and hyperparameters yields a set of runs with different tradeoffs between downstream performance and specialized retention, summarized by points $(\mathcal{L}_{\text{ft}}, \mathcal{L}_{\text{ret}})$. We summarize the best attainable tradeoffs using the *loss frontier*: the subset of non-dominated runs for which no other run achieves lower loss on both axes. We say one method (or base model) *dominates* another if its frontier achieves equal or lower $\mathcal{L}_{\text{ret}}$ at matched $\mathcal{L}_{\text{ft}}$ over the range covered by our sweeps.

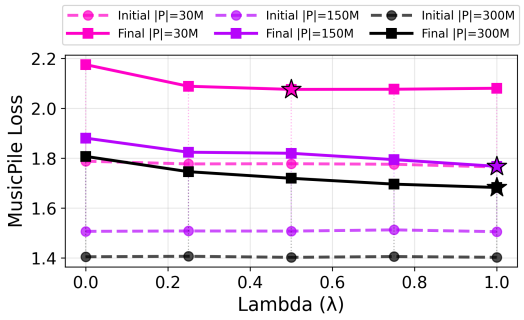

Figure 2: **Impact of the Mixture Fraction** λ. We plot the immediate (dashed) and retained (solid) MusicPile losses as a function of the mixing fraction λ. We find data mixing induces minimal differences immediately after continual pre-training, as evidenced by the flat dashed lines. On the other hand, we find that λ significantly affects the retention of MusicPile performance *post-finetuning*. Notably, even a small proportion of the available MusicPile data significantly improves retention.

Optimization and hyperparameters. Unless otherwise stated, we train with AdamW using linear warmup followed by cosine decay and fix weight decay to the LitGPT default AI (2023) of 0.1. Our base CPT sweep varies learning rate, batch size, and dropout. See Appendix A for details.

4 Experiments and Results

4.1 Mixed Pretraining Mitigates Forgetting

Setup. We compare mixed and unmixed pretraining for the same specialized corpus $\mathcal{D}_{\text{spec}}$ = MusicPile, and evaluate retention under two downstream adaptation regimes: $\mathcal{D}_{\text{ft}}$ = ChemPile and $\mathcal{D}_{\text{ft}}$ = FLAN, as well as a setting where $\mathcal{D}_{\text{spec}}$ = FLAN and $\mathcal{D}_{\text{ft}}$ = ChemPile. For each base model, we sweep CPT hyperparameters as described in Section 3.2, and evaluate the tradeoff between downstream loss $\mathcal{L}_{\text{ft}}$ and retained specialized loss $\mathcal{L}_{\text{ret}}$.

Main result. Figure 1 shows the resulting loss frontiers. In both downstream regimes, mixed pretraining improves the attainable tradeoff: at matched downstream loss, models derived from the mixed base achieve lower retained loss on MusicPile than those derived from the unmixed base. Mixing does not eliminate the learning–retention tradeoff, but shifts the frontier outward, enlarging the set of achievable operating points. Notably, the gap between mixed and unmixed frontiers is larger when $\mathcal{D}_{\text{ft}}$ = ChemPile than when $\mathcal{D}_{\text{ft}}$ = FLAN, consistent with ChemPile inducing a stronger distribution shift during Stage 3. We additionally find that the benefits of mixing hold for preserving instruction tuning (Figure 1 (c)) when considering $\mathcal{D}_{\text{spec}}$ = FLAN: mixing FLAN data during pretraining results in a better learning-retention tradeoff after downstream training on specialized knowledge (ChemPile).

4.2 Fine-grained Investigation of Mixing in Pretraining

Having established that mixed pretraining shifts the downstream–retention frontier, we next ask: (i) how much early exposure to $\mathcal{D}_{\text{spec}}$ is needed, and (ii) does early mixing help when the total $\mathcal{D}_{\text{spec}}$ compute budget is held fixed?

4.2.1 How much $\mathcal{D}_{\text{SPEC}}$ should we mix?

Experimental setup. We fix the Stage 2 CPT procedure and vary only the Stage 1 allocation of MusicPile into pretraining. Concretely, we pretrain on $\mathcal{D}_{\text{mix}}(\lambda) = \mathcal{D}_{\text{gen}} + \lambda\mathcal{D}_{\text{spec}}$ with $\mathcal{D}_{\text{spec}}$ = MusicPile and $\lambda \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$, then run CPT on MusicPile till saturation to obtain θ_{cpt}. To measure retention under subsequent adaptation, we set $\mathcal{D}_{\text{ft}} = \mathcal{D}_{\text{gen}}$ (continued pretraining on C4) and report both $\mathcal{L}_{\text{im}}$ and $\mathcal{L}_{\text{ret}}$ on MusicPile.

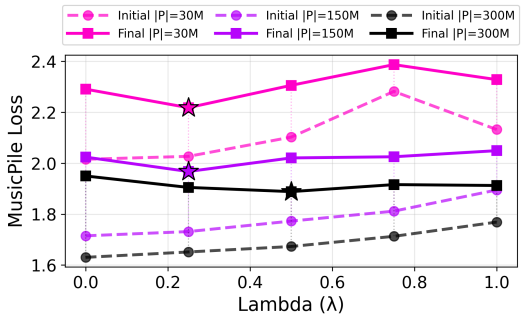

Figure 3: **Mixing remains optimal for retention under a fixed MusicPile compute budget.** We pretrain models with a λ fraction of MusicPile mixed into C4, while holding total MusicPile exposure fixed: across pretraining+CPT, each model sees the same number of MusicPile tokens (one pass). Dashed curves show *immediate* MusicPile loss after CPT; solid curves show *retained* MusicPile loss after subsequent fine-tuning. Across budgets ($|\mathcal{P}| \in \{30\text{M}, 150\text{M}, 300\text{M}\}$), increasing λ worsens immediate post-CPT loss (best at λ=0), but improves retained loss (best at λ>0), indicating that gains from end-loaded CPT are brittle to later training.

Impact of Mixture Fraction In Figure 2, we show the retained and immediate MusicPile losses as a function of the pretraining mixture fraction λ. We observe retained MusicPile loss λ decreases as λ increases, suggesting that increasing the ratio of specialized data during pretraining is generally helpful. Furthermore, we see that the retained loss decreases very sharply from $\lambda = 0$ to the lowest non-zero λ value (0.25) and decreases relatively slowly as λ is further increased. This suggests that even a relatively small (0.75% of $\mathcal{D}_{\text{gen}}$) early exposure to the specialized domain can have a significant impact on its retention.

Benefits of Mixing Can Be Latent. In Figure 2, we additionally plot the *immediate* MusicPile loss $\mathcal{L}_{\text{im}}$. Strikingly, the choice of mixture fraction λ has little effect on $\mathcal{L}_{\text{im}}$. Thus, CPT can drive models to comparable specialized performance immediately after CPT even when they differ in how much $\mathcal{D}_{\text{spec}}$ was seen during initial pretraining. However, these checkpoints behave very differently under subsequent adaptation, as reflected by the spread in retained loss $\mathcal{L}_{\text{ret}}$. More broadly, resistance to forgetting may not be apparent from the immediate specialized loss alone.

Takeaway: Mixing specialized domains into pretraining can have a latent benefit of improving the retention of those specialized domains under further training.

4.2.2 Is early mixing still beneficial under a fixed $\mathcal{D}_{\text{spec}}$ compute budget?

Mixing $\mathcal{D}_{\text{spec}}$ into pretraining and reserving $\mathcal{D}_{\text{spec}}$ for a dedicated CPT stage represent two qualitatively different ways of incorporating scarce domain data: early exposure interleaves $\mathcal{D}_{\text{spec}}$ with broad pretraining, while CPT concentrates optimization on $\mathcal{D}_{\text{spec}}$ after general capabilities have been learned. In the saturation setting, these strategies also differ in how many optimization steps they allocate to $\mathcal{D}_{\text{spec}}$, making it difficult to disentangle *where* $\mathcal{D}_{\text{spec}}$ is placed in the training pipeline from *how much* training on $\mathcal{D}_{\text{spec}}$ occurs. To study the relationship between pretraining-time mixing and dedicated CPT independent of total $\mathcal{D}_{\text{spec}}$ compute, we adopt a compute-matched variant that fixes the total number of $\mathcal{D}_{\text{spec}}$ tokens seen across pretraining (Stage 1) and CPT (Stage 2) and varies only their allocation between the stages.

Experimental setup. We pretrain on $\mathcal{D}_{\text{mix}}(\lambda) = \mathcal{D}_{\text{gen}} + \lambda\mathcal{D}_{\text{spec}}$ and then run CPT on the remaining $(1-\lambda)\mathcal{D}_{\text{spec}}$, so that every model sees exactly one pass over $\mathcal{D}_{\text{spec}}$ across Stages 1–2. Thus, λ controls how a fixed $\mathcal{D}_{\text{spec}}$ budget is allocated between early mixing (Stage 1) and dedicated CPT (Stage 2), with $\lambda = 1$ corresponding to mixing-only (no CPT). We repeat this ablation across three $\mathcal{D}_{\text{spec}}$ sizes.

Mixing worsens immediate performance in the compute-matched setting. Figure 3 shows that the immediate specialized loss $\mathcal{L}_{\text{im}}$ increases with λ in the compute-matched setting. When the total $\mathcal{D}_{\text{spec}}$ budget is fixed, allocating more of $\mathcal{D}_{\text{spec}}$ to Stage 1 mixing yields worse post-CPT MusicPile

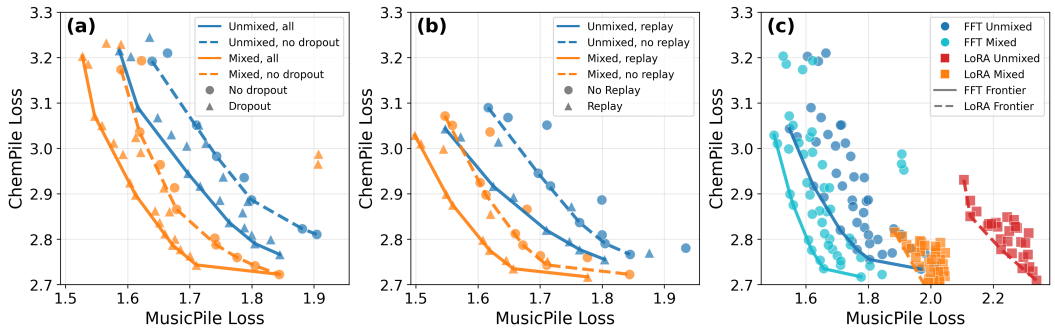

Figure 4: **CPT-time interventions (MusicPile → ChemPile).** We compare loss frontiers between downstream ChemPile loss $\mathcal{L}_{\text{ft}}$ (x-axis) and retained MusicPile loss $\mathcal{L}_{\text{ret}}$ (y-axis) for mixed vs. unmixed Stage 1 checkpoints under different Stage 2 CPT procedures. **(Left)** Dropout during CPT ($p \in \{0, 0.02, 0.05\}$) improves the frontier for both base models. **(Middle)** Replay during CPT (0 vs. 1% $\mathcal{D}_{\text{gen}}$ tokens interleaved) improves retention at matched downstream loss. **(Right)** LoRA-based CPT shows a markedly larger mixed–unmixed gap than full-parameter CPT, though the resulting LoRA frontiers are overall dominated by full-parameter CPT.

performance; concentrating the same $\mathcal{D}_{\text{spec}}$ tokens into a dedicated Stage 2 CPT phase achieves the lowest $\mathcal{L}_{\text{im}}$. In other words, under a fixed $\mathcal{D}_{\text{spec}}$ token budget, CPT is a more direct way to drive down *immediate* specialized loss—though it may do so by moving along a broader tradeoff surface (e.g., with general capabilities).

Mixing improves retention under subsequent adaptation. Despite harming $\mathcal{L}_{\text{im}}$, early mixing improves retention after downstream training. As λ increases, the retained loss $\mathcal{L}_{\text{ret}}$ decreases and the degradation gap $\mathcal{L}_{\text{ret}} - \mathcal{L}_{\text{im}}$ shrinks, indicating that specialized gains from CPT are more brittle under further training than gains supported by early exposure. Consequently, the best retained specialized performance is achieved at a non-zero mixture fraction, revealing a tradeoff between optimizing immediate performance and improving robustness to forgetting.

Takeaway: While CPT can quickly improve performance on rare domains, these improvements can degrade quickly under further training.

Our results in this section highlight an important consideration for incorporating specialized domains into LLMs. Across both saturation and compute-matched settings, we find that selecting methods based solely on *immediate* specialized performance (low $\mathcal{L}_{\text{im}}$) can mask substantial differences in how well that competence is preserved under subsequent adaptation. In particular, approaches that achieve similar or even better short-term improvements may be more brittle, exhibiting larger degradation from $\mathcal{L}_{\text{im}}$ to $\mathcal{L}_{\text{ret}}$ as training continues. These findings suggest that evaluating adaptation methods only at convergence on $\mathcal{D}_{\text{spec}}$ can be misleading when the model undergoes additional training.

In the next section, we study if *algorithmic choices* during continual pretraining could achieve the same gains from data mixing during pretraining.

4.3 Studying the Impact of Continual Pretraining Methodology

Previously, we showed that incorporating $\mathcal{D}_{\text{spec}}$ during Stage 1 pretraining improves its retention under subsequent adaptation, compared to relying on a dedicated Stage 2 CPT phase alone. In practice, however, pretraining-time mixing is not always feasible: specialized data may arrive after pretraining has finished, be subject to licensing or privacy constraints, or be too costly to retroactively integrate into the pretraining pipeline. This motivates a practical question: can we recover some of the retention benefits of early mixing using only interventions at CPT time? In this section, we systematically study how common CPT-time choices affect the downstream–retention tradeoff.

4.3.1 Dropout and Pretraining Mixing are Complementary

Dropout is a standard regularizer, but it is often disabled in modern decoder-only LM training, making it a relatively underexplored knob for continual pretraining. We hypothesize that introducing small amounts of dropout during CPT encourages more robust representations of $\mathcal{D}_{\text{spec}}$, improving retention under subsequent adaptation.

Setup. We study dropout as a CPT-time intervention by enabling dropout during Stage 2 and varying the dropout rate $p \in \{0.0, 0.02, 0.05\}$. Initial experiments indicated any larger p led to worse post-CPT loss. For each p, we sweep the base CPT optimization grid described in Section 3.2 (learning rate and batch size, with weight decay and warmup fixed), starting from both mixed and unmixed Stage 1 checkpoints, and compare the resulting loss frontiers (Figure 4 and 9, left).

Results. Dropout improves the downstream–retention tradeoff for both mixed and unmixed base models, shifting their frontiers outward; however, it does not fully close the gap between them. This suggests that CPT-time regularization and pretraining-time mixing provide complementary benefits: dropout improves robustness of $\mathcal{D}_{\text{spec}}$ acquisition within Stage 2, while mixing changes the base model in a way that remains advantageous even with the same CPT intervention.

4.3.2 Replay Data and Pretraining Mixing are Complementary

Setup. We study rehearsal during CPT by injecting a small amount of pretraining data into Stage 2, following Bethune et al. (2025). In this ablation (run for the 300M-token MusicPile setting), during CPT on $\mathcal{D}_{\text{spec}}$ we either use no replay or interleave an additional 1% of $\mathcal{D}_{\text{gen}}$ tokens relative to the $\mathcal{D}_{\text{spec}}$ CPT budget. We sweep the same base CPT optimization grid (Section 3.2) from both mixed and unmixed Stage 1 checkpoints and compare loss frontiers (Figure 4, middle).

Results. Replay improves the downstream–retention tradeoff for both mixed and unmixed base models (Figure 4, middle). At matched downstream loss $\mathcal{L}_{\text{ft}}$, replay yields lower retained loss $\mathcal{L}_{\text{ret}}$, indicating improved resistance to forgetting under subsequent adaptation. However, replay does not substitute for pretraining-time mixing: over most of the frontier, unmixed models with replay are still dominated by mixed models without replay. Combining mixing and replay yields the best tradeoffs, suggesting these interventions are complementary.

4.3.3 LoRA Produces a Worse Frontier, While Amplifying Mixing Effects

Setup. We study parameter-efficient CPT by replacing full-parameter updates in Stage 2 with LoRA updates. We fix the LoRA configuration (rank r=64, scaling α=128) and sweep the base CPT optimization grid (Section 3.2) starting from both mixed and unmixed Stage 1 checkpoints, comparing the resulting loss frontiers to full-parameter CPT (Figure 4, right).

Result. LoRA traces a qualitatively different downstream–retention tradeoff from full-parameter CPT. Relative to full-parameter CPT, LoRA tends to achieve lower downstream ChemPile loss $\mathcal{L}_{\text{ft}}$ but worse retained MusicPile loss $\mathcal{L}_{\text{ret}}$. Consequently, the LoRA frontiers are overall dominated by the full-parameter (FFT) frontiers for both mixed and unmixed base models. At the same time, the effect of pretraining-time mixing is amplified under LoRA: the separation between mixed and unmixed LoRA frontiers is substantially larger than in the full-parameter setting, suggesting that early exposure to $\mathcal{D}_{\text{spec}}$ becomes more important when Stage 2 adaptation is constrained to a low-rank subspace. Interestingly, we also find that the change in loss on $\mathcal{D}_{\text{spec}}$ before and after finetuning is much greater for LoRA compared to FFT (see Figure 6).

Takeaway: CPT methods that improve training without mixing during pretraining are complementary with mixed pretraining.

5 Analysis of Data Mixing

In the previous section, we showed how mixing domain-specific data during pretraining can improve its retention during further training stages. In this section, we study this phenomena in a simplified

two-layer linear network. We consider a stylized version of the three-stage training setup that we study empirically in Section 4.

Setup We study a series of regression problems to simulate general pre-training ($\mathcal{D}_{\text{gen}}$), the specialized domain ($\mathcal{D}_{\text{spec}}$), and downstream finetuning ($\mathcal{D}_{\text{ft}}$), as we describe below. We train a two layer linear network $\theta = \mathbf{W}_1\mathbf{W}_2$ on these tasks sequentially using gradient descent on the squared loss: $\mathcal{L}(\theta; \mathcal{D}) = \mathbf{E}[||\theta\mathbf{x} - \mathbf{y}||^2]$.

Each task that we consider is defined by an input distribution $\mathcal{D}_x$ and outputs are generated as $\mathbf{y} = \mathbf{A}^t\mathbf{x}$. Following the setup in Springer et al. (2025), we consider that all tasks in the sequence can be simultaneously diagonalized by matrices $\mathbf{U}, \mathbf{V}$, such that the singular value decomposition of $\mathbf{A}^t = \mathbf{U}\mathbf{\Sigma}^{(t)}\mathbf{V}^\top$. We consider the singular values of $\mathbf{A}^{(t)}$ as *features* for the task task t. We discuss the structure and relationship between the stages in the following.

Shared v.s. Specialized Dimensions We consider that the input space can be partitioned into a set of shared $n-k$ shared and k specialized dimensions. Concretely, the first $n-k$ coordinates correspond to shared dimensions (which are active across $\mathcal{D}_{\text{gen}}$, $\mathcal{D}_{\text{spec}}$,and $\mathcal{D}_{\text{ft}}$. On the other hand, the last k dimensions take non-zero values only on the specialized domain $\mathcal{D}_{\text{spec}}$. Similarly, we consider that the singular vectors can be partitioned into $\mathbf{V} = [\mathbf{V}_{\text{gen}}|\mathbf{V}_{\text{spec}}]$ where $\mathbf{V}_{\text{gen}}$ spans the first $n-k$ dimensions.

Pretraining Task We consider that $\mathcal{D}_{\text{gen}}$ has inputs $x \sim \mathcal{N}(0, \mathbf{I}_{n-k})$– only taking non-zero values on the shared features– and that targets as $y = \mathbf{A}^{\text{gen}}x$. We define the pre-training features (singular values) as $\sigma_1^{\text{gen}}, ..., \sigma_{n-k}^{\text{gen}}$ (i.e. such that $\Sigma^{\text{gen}} = \text{diag}(\sigma_1^{\text{gen}}, ..., \sigma_{n-k}^{\text{gen}}, \mathbf{0}_k)$ and $\mathbf{A}^{\text{gen}} = \mathbf{U}\Sigma^{\text{gen}}\mathbf{V}^\top$.

Specialized Task The specialized task $\mathcal{D}_{\text{spec}}$ has inputs sampled $x \sim \mathcal{N}(0, \mathbf{I}_n)$ and has outputs generated $y = \mathbf{A}^{\text{spec}}x$. We define the specialized features as $\sigma_1^{\text{spec}}, ..., \sigma_n^{\text{spec}}$. We can characterize the relationship between the general and specialized domain by categorizing the first $n-2k$ features as *invariant* (taking the same value between $\mathbf{A}^{\text{gen}}$and $\mathbf{A}^{\text{spec}}$). We characterize the remaining k shared features as *shared-misaligned* features $\mathbf{A}^{\text{gen}}$ and $\mathbf{A}^{\text{spec}}$. In particular, we have $\forall i \in [n-2k, n-k]$, $\sigma_i^{\text{spec}} = C_{\text{spec}}\sigma_i^{\text{gen}}$.

Downstream Tuning Task We consider that the downstream tuning task is relatively more similar to the pretraining task than the specialized 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. However, the we have that the *shared-misaligned* features are misaligned between $\mathbf{A}^{\text{spec}}$ and $\mathbf{A}^{\text{ft}}$, in particular that $\sigma_i^{\text{ft}} = C_{\text{ft}}\sigma_i^{\text{spec}}$.

5.1 Mixed and Unmixed Pretraining Learn Different Features

We first characterize the difference between mixed and unmixed pretraining in terms of the features learned. We prove that given a sufficient mixing ratio α, θ^{mixed} learns the specialized feature, whereas θ^{unmixed} does not.

Theorem 1 (*Informal:* Only Mixing Learns Specialized Features)**.** *Let $\theta^{mixed} = \mathbf{W}_1^{(mixed)}\mathbf{W}_2^{(mixed)}$ be the the parameters learned by pretraining only on $\mathcal{D}_{mixed}$ and $\theta^{unmixed} = \mathbf{W}_1^{(unmixed)}\mathbf{W}_2^{(unmixed)}$ until a timestep t_{n-k}. Then θ^{mixed} learns the specialized features, while $\theta^{unmixed}$ does not.*

We provide a formal statement in Appendix C. Our result leverages the fact that linear networks learn features in descending order of their singular value Gidel et al. (2019); Springer et al. (2025), while the remaining features stay close to 0. Intuitively, without mixing $\mathcal{D}_{\text{spec}}$, the $n-k$ specialized features have the lowest singular values and hence are not learned. On the other hand, a sufficient amount of mixing increases the importance of specialized features such that they are learned.

5.2 Continual Pretraining Primarily Reuses Existing Features

In the previous section, we identified that mixed-pretraining learns domain-unique, specialized features while unmixed pre-training does not. In this section, we study the impact of this initialization on the updates during the continual pre-training stage. In particular, we show a crucial difference between how continual pretraining reduces $\mathcal{L}(.; \mathcal{D}_{\text{spec}})$ in the unmixed and unmixed settings. Starting

from θ^{mixed}, CPT minimizes the loss by leveraging the specialized features, whereas in the setting of θ^{unmixed}, CPT primarily modifies the misaligned shared features.

Theorem 2 (*Informal*: CPT on θ^{mixed} versus θ^{umixed})**.** *Consider performing CPT on $\mathcal{D}_{spec}$ starting from the θ^{mixed} and $\theta^{unmixed}$ for K steps. The learned parameters after CPT $\theta^{unmixed}_{cpt}(K) \approx \mathbf{U}\Sigma^{shared}_{CPT}\mathbf{V}^\top$ while $\theta^{mixed}_{cpt}(K) \approx \mathbf{U}\Sigma^{spec}_{cpt}\mathbf{V}^\top$.*

Here, $\mathbf{U}\Sigma^{\text{shared}}_{\text{cpt}}\mathbf{V}^\top$ represent the optimal parameters under the constraint that only the *invariant* and *shared-misaligned* features can be used, while $\mathbf{U}\Sigma^{\text{spec}}_{\text{cpt}}\mathbf{V}^\top$represents the optimal parameters when *specialized* features can be used. The crux of the argument relies on the fact that continual pre-training primarily leverages features that are learned in the pre-trained initialization, rather than learning new features. Intuitively, if a learned feature $\Sigma_{ii} = 0$ during the beginning of CPT, then we show that it remains 0 throughout the CPT process. Thus, while CPT on θ^{mixed} is able to update the singular values of the specialized features (having learned them in pretraining), CPT on θ^{unmixed} only has access to the shared mis-aligned features and thus only updates those. In the next subsection, we examine the implication of this difference for the downstream forgetting behaviors.

5.3 Characterizing Downstream Forgetting Behavior

Previously, we showed that $\theta^{\text{mixed}}_{\text{cpt}}$ and $\theta^{\text{unmixed}}_{\text{cpt}}$ leverage different features to lower the loss on $\mathcal{D}_{\text{spec}}$. In this section, we demonstrate this difference has crucial implications relative to the amount of catastrophic forgetting experienced by the models when undergoing further fine-tuning.

Theorem 3 (*Informal:* $\theta^{\text{unmixed}}_{\text{cpt}}$ experiences more forgetting than $\theta^{\text{mixed}}_{\text{cpt}}$)**.** *Let $\Delta_{mixed}, \Delta_{ummixed}$ denote the difference in loss on $\mathcal{D}_{spec}$ before and after training on $\mathcal{D}_{ft}$, for the mixed and unmixed checkpoints, respectively. Then we have that $\Delta_{unmixed} \geq \Delta_{mixed}$.*

We defer the proof of this theorem to Appendix C, but describe the intuition here. We show that the downstream fine-tuning does not perturb the *specialized features* due to their orthogonality with the downstream fine-tuning task. On the other hand, the shared-misaligned features are not orthogonal and therefore get perturbed during the finetuning process on $\mathcal{D}_{\text{ft}}$. In the unmixed setting, the loss reduction on $\mathcal{D}_{\text{spec}}$ achieved during CPT is concentrated in the shared-misaligned features, making them susceptible to downstream erasure. On the other hand, in θ^{mixed}, the preservation of the specialized features mitigates forgetting.

6 Discussion

In this work, we study how upstream training choices impact the retention of specialized domain knowledge during downstream training. We demonstrate that mixing specialized data into the initial pre-training stage substantially improves its retention versus incorporating it in a subsequent training stage. Future works can identify additional levers by which model-designers can pre-emptively mitigate forgetting, as well as identifying new regularization techniques that enhance retention *without* requiring data-mixing.

7 Impact Statement

This paper presents work whose goal is to advance the field of machine learning. We hope that this enables models to be finetuned more reliably without losing previously acquired capabilities. Our findings on retention dynamics may also inform efforts to make safety training more robust to subsequent finetuning.

REFERENCES

Lightning AI. Litgpt. https://github.com/Lightning-AI/litgpt, 2023.

Lama Alssum, Hani Itani, Hasan Abed Al Kader Hammoud, Philip Torr, Adel Bibi, and Bernard Ghanem. Unforgotten safety: Preserving safety alignment of large language models with continual learning, 2025. URL https://arxiv.org/abs/2512.10150.

Louis Bethune, David Grangier, Dan Busbridge, Eleonora Gualdoni, Marco Cuturi, and Pierre Ablin. Scaling laws for forgetting during finetuning with pretraining data injection, 2025. URL https://arxiv.org/abs/2502.06042.

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, 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.

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

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, 2021. URL https://arxiv.org/abs/2106.09685.

Gaganpreet Jhajj and Fuhua Lin. Elastic weight consolidation for knowledge graph continual learning: An empirical evaluation, 2025. URL https://arxiv.org/abs/2512.01890.

Danny D. Leybzon and Corentin Kervadec. Learning, forgetting, remembering: Insights from tracking LLM memorization during training. In Yonatan Belinkov, Najoung Kim, Jaap Jumelet, Hosein Mohebbi, Aaron Mueller, and Hanjie Chen (eds.), *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 43–57, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.blackboxnlp-1.4. URL https://aclanthology.org/2024.blackboxnlp-1.4/.

Jessy Lin, Luke Zettlemoyer, Gargi Ghosh, Wen-Tau Yih, Aram Markosyan, Vincent-Pierre Berges, and Barlas Oğuz. Continual learning via sparse memory finetuning, 2025. URL https://arxiv.org/abs/2510.15103.

Emmy Liu, Graham Neubig, and Chenyan Xiong. Midtraining bridges pretraining and posttraining distributions, 2025. URL https://arxiv.org/abs/2510.14865.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. Tofu: A task of fictitious unlearning for llms, 2024a. URL https://arxiv.org/abs/2401.06121.

Pratyush Maini, Skyler Seto, He Bai, David Grangier, Yizhe Zhang, and Navdeep Jaitly. Rephrasing the web: A recipe for compute and data-efficient language modeling, 2024b. URL https://arxiv.org/abs/2401.16380.

Pratyush Maini, Sachin Goyal, Dylan Sam, Alex Robey, Yash Savani, Yiding Jiang, Andy Zou, Matt Fredrikson, Zacharcy C. Lipton, and J. Zico Kolter. Safety pretraining: Toward the next generation of safe ai, 2025. URL https://arxiv.org/abs/2504.16980.

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

Kento Nishi, Rahul Ramesh, Maya Okawa, Mikail Khona, Hidenori Tanaka, and Ekdeep Singh Lubana. Representation shattering in transformers: A synthetic study with knowledge editing, 2025. URL https://arxiv.org/abs/2410.17194.

Kyle O'Brien, Stephen Casper, Quentin Anthony, Tomek Korbak, Robert Kirk, Xander Davies, Ishan Mishra, Geoffrey Irving, Yarin Gal, and Stella Biderman. Deep ignorance: Filtering pretraining data builds tamper-resistant safeguards into open-weight llms, 2025. URL https://arxiv.org/abs/2508.06601.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.

Kaustubh Ponkshe, Shaan Shah, Raghav Singhal, and Praneeth Vepakomma. Safety subspaces are not linearly distinct: A fine-tuning case study, 2025. URL https://arxiv.org/abs/2505.14185.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to!, 2023. URL https://arxiv.org/abs/2310.03693.

Zhenting Qi, Fan Nie, Alexandre Alahi, James Zou, Himabindu Lakkaraju, Yilun Du, Eric Xing, Sham Kakade, and Hanlin Zhang. Evolm: In search of lost language model training dynamics, 2025. URL https://arxiv.org/abs/2506.16029.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. Code llama: Open foundation models for code, 2024. URL https://arxiv.org/abs/2308.12950.

Dylan Sam, Sachin Goyal, Pratyush Maini, Alexander Robey, and J. Zico Kolter. When should we introduce safety interventions during pretraining?, 2026. URL https://arxiv.org/abs/2601.07087.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Jacob Mitchell Springer, Sachin Goyal, Kaiyue Wen, Tanishq Kumar, Xiang Yue, Sadhika Malladi, Graham Neubig, and Aditi Raghunathan. Overtrained language models are harder to fine-tune, 2025. URL https://arxiv.org/abs/2503.19206.

Johnny Tian-Zheng Wei, Ameya Godbole, Mohammad Aflah Khan, Ryan Wang, Xiaoyuan Zhu, James Flemings, Nitya Kashyap, Krishna P. Gummadi, Willie Neiswanger, and Robin Jia. Hubble: a model suite to advance the study of llm memorization, 2025. URL https://arxiv.org/abs/2510.19811.

Xianjun Yang, Xiao Wang, Qi Zhang, Linda Petzold, William Yang Wang, Xun Zhao, and Dahua Lin. Shadow alignment: The ease of subverting safely-aligned language models, 2023. URL https://arxiv.org/abs/2310.02949.

A Training Details

A.1 Dataset Statistics

Table 1: Dataset statistics. All token counts measured with SmolLM2 tokenizer. Subsampled from HuggingFace.

Dataset	Split	Tokens	Description
C4	Train	8.7B	General web text pretraining corpus
MusicPile	Train	300M	Music-domain text corpus
ChemPile	Train	300M	Chemistry-domain text corpus
FLAN	Train	300M	Instruction-tuning dataset

A.2 Optimizer Configuration

Table 2: 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

A.3 FFT (Full Fine-Tuning) Configuration

Table 3: FFT hyperparameter search space for Stage 2 continual pretraining.

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

A.4 LoRA Configuration

Table 4: LoRA hyperparameter configuration for Stage 2 continual pretraining.

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 3)	

B ADDITIONAL PLOTS

B.1 MIXING

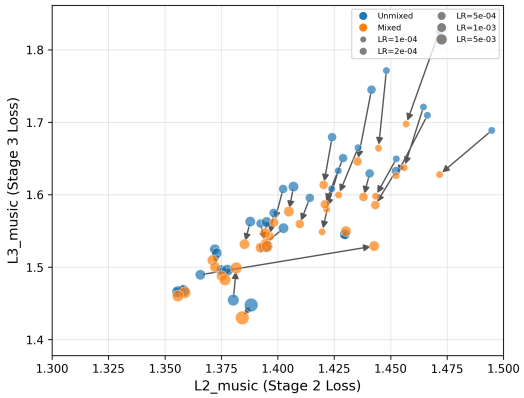

Figure 5: Pairs of runs identical hyperparameter except for whether or not they started from a mixed $\lambda = 1.0$ checkpoint.

B.2 LORA VS FFT DURING CPT

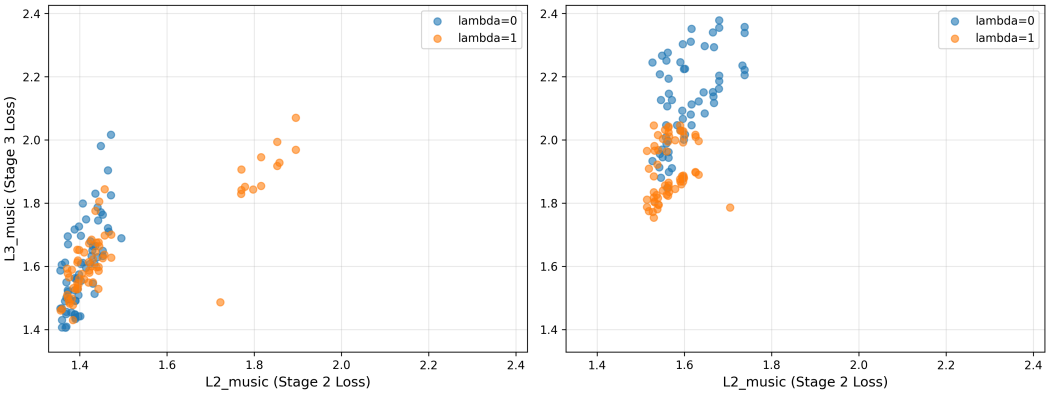

Figure 6: Stage 3 MusicPile loss vs Stage 2 Musicpile loss. We see a greater downward shift between blue and orange points in the LoRA plotting, suggesting a greater resistance to forgetting from mixing. Notably, the point clouds are mostly vertically stacked, indicating similar stage 2 perforamance across hyperparameters.

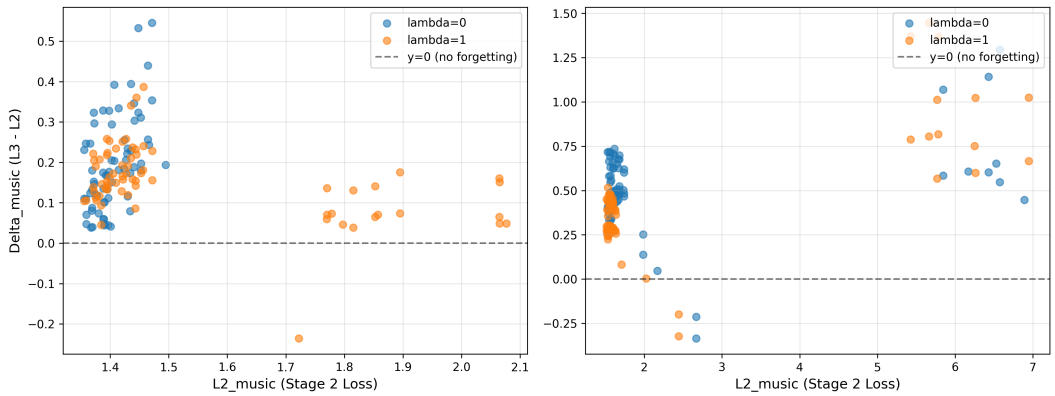

Figure 7: Similar to the figure above, but with delta loss on the y-axis

B.3 Replay and Dropout

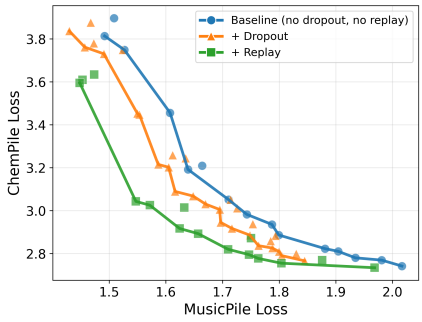

Figure 8: Comparison in frontier shift from baseline when using dropout vs replay.

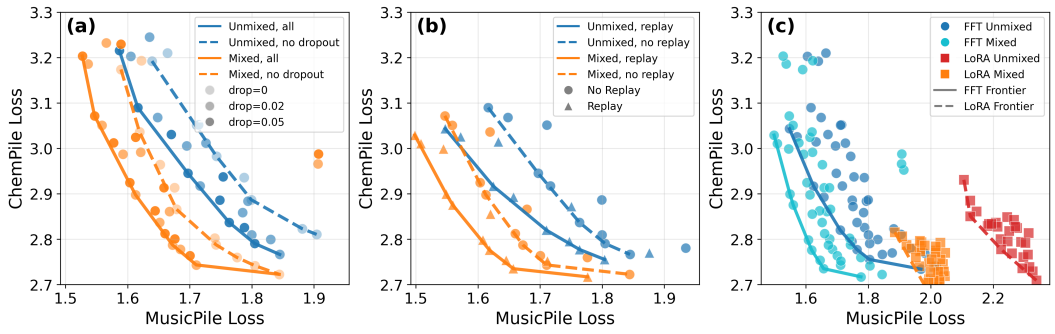

Figure 9: Same as figure 4, but dropout % is displayed using opacity in (a).

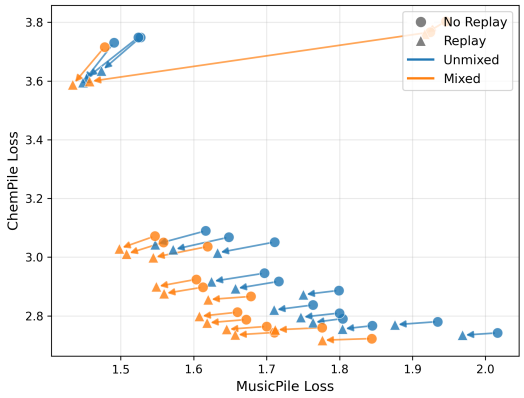

Figure 10: Pairs of hyperparameters, differing only by if they used replay.

C Theoretical Analysis

C.1 Preliminaries and Setup

Model 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 $\mathcal{L}_t(\theta) = \mathbf{E}_{x\sim\mathcal{D}^t}[||\theta x - \mathbf{A}^t x||_2^2]$. We precisely define the problems encountered in the different stages below. Here, $\mathcal{D}^t$ denotes the input distribution for the task t and the ground-truth outputs are generated as $\mathbf{A}^t\mathbf{X}$.

General Pretraining Domain We define the input distribution as $x \sim \mathcal{N}(0, \mathbf{I}_{d-k})$. Intuitively, the inputs for the general pre-training domain are supported on all but k of the features. Those remaining features are "specialized features" and used only by the specialized domain task (as defined below). We define the general task as $Y = \mathbf{A}^{\text{gen}}x$.

Specialized Domain We consider the specialized data as given by $y = \mathbf{A}^{\text{sp}}x$. Here the input distribution $\mathcal{D}^{\text{spec}}$ is defined to be $x \sim \mathcal{N}(0, \mathbf{I}_d)$. Intuitively, the isotropic nature of the specialized domain means that the specialized task leverages both the general capabilities of the model, as well as some "domain-specific" features contained specialized to the domain.

Partially Shared Structure We consider the setting in which $\mathbf{A}^{\text{pre}}$ and $\mathbf{A}^{\text{sp}}$ can be simultaneously diagonalized by matrices. In particular, $\mathbf{A}^{\text{pre}} = \mathbf{U}\mathbf{\Sigma}^{\text{pre}}\mathbf{V}^T$ and $\mathbf{A}^{\text{sp}} = \mathbf{U}\mathbf{\Sigma}^{\text{sp}}\mathbf{V}^T$. We will additionally consider that $\mathbf{V}^\top = [\mathbf{V}_{\text{gen}}|\mathbf{V}_{\text{sp}}]$, where the $\mathbf{V}_{\text{sp}}$ intuitively forms a basis for the final $d-k$ features. That is, . Intuitively, we can partition the features into the following structure. For some $d_{\text{invariant}}$ we have that for $1 \le i \le d_{\text{invariant}}$, $\mathbf{\Sigma}^{\text{pre}}_{ii} = \mathbf{\Sigma}^{\text{sp}}_{ii}$. The remaining $d - 1 - d_{\text{invariant}}$ are overlapping features and we have that $C_{\text{spec}}\mathbf{\Sigma}^{\text{pre}}_{ii} = \mathbf{\Sigma}^{\text{spec}}_{ii}$, where C_{spec} is a constant that determines the degree of misalignment on these features between the general and specialized tasks. Finally, the final feature is d, we have that $\mathbf{\Sigma}^{\text{pre}}_{dd} = 0$ while $\mathbf{\Sigma}^{\text{sp}}_{dd} = \beta$.

Downstream Tuning Task We consider that the downstream tuning task is relatively more similar to the pretraining task than the specialized 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. However, the we have that the *shared-misaligned* features are misaligned between $\mathbf{A}^{\text{spec}}$ and $\mathbf{A}^{\text{ft}}$, in particular that $\sigma_i^{\text{ft}} = C_{\text{ft}}\sigma_i^{\text{spec}}$.

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

[Invariant Features are High Magnitude]. We consider that the invariant features are higher magnitude than the specialized features and the shared-misaligned 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:

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

$\forall i \in [0, d_{\text{invariant}}]$ and $\forall j \in (d_{\text{invariant}}, n)$. This intuitively encodes that the invariant features have the highest salience in the data.

[Sufficient Specialized Mixing] We assume that there exists $\alpha > 0$

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

Intuitively, Assumption C.1 suggests that there exists a mixing ratio such that the mixing specialized features become more salient than the shared-misaligned features.

C.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

[Joint Decomposition] We assume that 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 $\lambda_1, ..., \lambda_{r_x} = 1$. We will further assume

Initialization We define that $\mathbf{W}_2(0) = \mathbf{W}_1(0) = \exp(-\mathcal{T})\mathbf{I}_d$.

Theorem 4 (Sequential Learning of Features Gidel et al. (2019); Springer et al. (2025))**.** *Consider $\mathbf{W}_1, \mathbf{W}_2$ be intialized as above. 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 we define $\Sigma_{:i}$ to be diag$(\sigma_1, ...\sigma_i, 0, ...0)$.

The main intuition of Theorem 4 is that, during the pre-training process, $\mathbf{W}_1\mathbf{W}_2$ learn features in the order 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 5 (Only Mixing Learns Specialized Features)**.** *Let $\theta^{pre}(t) = \mathbf{W}_1^{pre}(t)\mathbf{W}_2^{pre}(t)$ be the parameters learned when pre-training 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*

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

Proof. This is a relatively straightforward application of 4. 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 $\mathbf{\Sigma}_{xy}^{\text{(mixed)}} = (1-\alpha)\mathbf{\Sigma}_{xy}^{\text{pre}} + \alpha\mathbf{\Sigma}_{xy}^{\text{spec}}$. By Theorem 4, we have that the features are learned in order of the singular values of $\mathbf{\Sigma_{xy}}$. By Assumptions C.1 and C.1, 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 4, 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 $\mathbf{\Sigma}_{xy}^{\text{(gen)}}$ ae the $n-2k$ shared features are the k shared-misaligned features. We can define $\mathbf{\Sigma}_{:n-k}^{\text{(unmixed)}} = \text{diag}(\sigma_1^{\text{inv}}, ...\sigma_k^{\text{inv}}, \sigma_1^{\text{shared-mis}}, ..., \sigma_1^{\text{shared-mis}}, \mathbf{0}_k)$. Similarly, by applying 4, 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 4 demonstrates that mixing $\mathcal{D}_{\text{spec}}$ during pretraining results in a pre-trained initialization that has different features. Mixing learns the specialized features, while not mixing learns only the shared-misaligned features. In the following, we will examine the impact that these different features have on the retention of $\mathcal{D}_{\text{spec}}$ during subsequent training.

C.3 Analysis of Continual Pre-training

Next, we study the dynamics of the continual pretraining process. To formalize the continual pretraining process, we first examine the dynamics beginning from the idealized pretraining initialization (as performed by Springer et al. (2025)). We perform the continual-pretraining 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 $x \sim \mathcal{N}(0, \mathbf{I}_d)$, thisis equivalent to $||\theta - \mathbf{A}^{\text{sp}}||_F$. We follow the assumptions on the regularity of fine-tuning established in Springer et al. (2025)

[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

[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 $(\hat{\mathbf{W}}_1(0), \hat{\mathbf{W}}_2(0))$. For the mixed ca

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

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 $\sigma_i^{\text{(un)mixed}}(t)$ as the i-th singular value at step t:

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

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

Next,we observe the dynamics of the non-zero singular values (Lemma A.11 Springer et al. (2025)

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

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

Across the settings, we assume that K is sufficiently large such that $\epsilon < \frac{(n-k)(\alpha-1)\min_i \sigma^{\text{spec}_i}}{4k\beta+2(\alpha-1)}$
Now, we will define the matrices $\mathbf{\Sigma}^{\text{shared,CPT}} = \text{diag}(\sigma_1^{\text{inv}}, ...\sigma_{n-2k}^{\text{inv}}, \sigma_{d_{\text{invariant}}+1}^{\text{spec}}, ..\sigma_{d_{\text{invariant}}+k}^{\text{spec}}, \mathbf{0}_k)$ and $\mathbf{\Sigma}^{\text{spec,CPT}} = \text{diag}(\sigma_1^{\text{inv}}, ...\sigma_{n-2k}^{\text{inv}}, \mathbf{0}_k, \sigma_{d_{\text{invariant}}+k+1}^{\text{spec}}, ..\sigma_{d_{\text{invariant}}+2k}^{\text{spec}})$, Intuitively, $\mathbf{\Sigma}^{\text{shared,CPT}}$ lowers loss on $\mathcal{D}_{\text{spec}}$ by shifting the values on the shared features, while $\mathbf{\Sigma}^{\text{spec,CPT}}$ accomplishes this by modifying the unique features. We are now ready to state the main theorem.

Theorem 6 (CPT 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*

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

Proof. This theorem follows by noting that under the idealized pre-training initialization any singular value that is 0 at initialization remains that wsy during the entire optimization trajectory. Note that the 0 singular values of $\mathbf{U}^\top\theta^{\text{mixed}}\mathbf{V}$ coincide with $\mathbf{\Sigma}^{\text{spec, CPT}}$ (the shared misaligned features) and likwise $\mathbf{U}^\top\theta^{\text{unmixed}}\mathbf{V}$ coincide with $\mathbf{\Sigma}^{\text{shared, CPT}}$ (the shared misaligned features).

This implies that we have

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

Now, applying the result from Lemma B.8 yields the desired claim. □

C.4 Analysis of Downstream Adaptation

We first establish that the singular values corresponding to directions in which there is no covariance remain unchanged throughout the downstream fine-tuning stage. Consider performing downstream unregularized fine-tuning on $\mathcal{D}_{\text{ft}}$. If $x \sim \mathcal{D}_{\text{ft}}$ 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. Thus, 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 7. *Consider performing K steps of gradient descent on the downstream finetuning dataset beginning from the intializations $\theta^{CPT,\ mixed}(K)$ and let $\theta^{FT,mixed}(K)$, $\theta^{FT,unmixed}(K)$ $\theta^{CPT,\ unmixed}(K)$ denote the final parameters. Let $\Delta_{unmixed} = \mathcal{L}(\theta^{FT,\ mixed}; \mathcal{D}_{spec}) - \mathcal{L}(\theta^{CPT,\ mixed}(K); \mathcal{D}_{spec})$ and likewise $\Delta_{mixed} = \mathcal{L}(\theta^{FT,\ unmixed}; \mathcal{D}_{spec}) - \mathcal{L}(\theta^{CPT,unmixed}(K); \mathcal{D}_{spec})$. Then we have that $\Delta_{unmixed} > \Delta_{mixed}$*

Proof. As the invariant features take the same values, they will not move during the downstream adaptation. Moreover, due to the Lemma C.4, 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 due to the full-rankedness of covariance of $\mathcal{D}_{\text{spec}}$, we have that the $\mathcal{L}(\theta; \mathcal{D}_{\text{spec}}) = ||\theta - \mathbf{A}^{\text{spec}}||_F^2$–which is the . By applying Lemma C.3 We define the following matrices $\Sigma_{\text{FT}}^{(\text{unmixed})} = \text{diag}(\sigma_1^{\text{invariant}}, ..., \sigma_{d_{\text{invariant}}}^{\text{invariant}}, \sigma_{d_{\text{invariant}}+1}^{\text{ft}}, ...\sigma_{\text{d}_{\text{invariant}}+k}^{\text{ft}} \mathbf{0}_k)$ and note that Lemma C.3 gives us that

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

Likewise, we have that

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

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

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

and likewise lower bound

$$\mathcal{L}(\theta_{\text{ft}}^{\text{unmixed}}; \mathcal{D}_{\text{spec}}) \geq k(\beta - \epsilon)^2 + (n-k)((\alpha - 1) - \epsilon)^2$$

Then,we can lower bound $\Delta_{\text{unmixed}} \geq k[(\beta + e)^2 - (\beta - \epsilon)^2] + (n-k)[(\alpha - 1 - \epsilon)^2 - \epsilon^2]$. From the upper bound of ϵ, we have that this implies $\Delta_{\text{mixed}} \geq 0$. Hence, we have $\Delta_{\text{unmixed}} \geq \Delta_{\text{mixed}}$ □