Resistance to Forgetting

Lawrence Feng advised by Aditi Raghunathan[*]
[*]Carnegie Mellon University

1. Overview of the Problem

Large language models (LLMs) are trained through multi-stage pipelines consisting of (1) large-scale pretraining on broad corpora, (2) continual pretraining or domain-specific finetuning (CPT/FT), and (3) post-training on unrelated data for alignment or generalization. A recurring challenge in this pipeline is *catastrophic forgetting*: after learning a specialized domain D, subsequent training on unrelated data D' significantly degrades performance on D.

A key empirical observation motivating this work is the following:

> **Mixing a specialized domain into pretraining—even at very small percentages—substantially increases resistance to forgetting after later training.**

This phenomenon appears consistently across dataset sizes and mixture ratios. It suggests that *when* a model first encounters a domain may be as important as *how much* it sees. Early exposure during pretraining likely interacts with global representational development (e.g., circuit formation, decreased knowledge entropy) in ways that make later overwriting more difficult.

However, practitioners rarely have access to pretraining data and cannot rerun mixture-based pretraining. This raises the central question of the thesis:

> **Can we induce the benefits of pretraining-domain mixing *using only finetuning-stage methods*, such as LoRA or adapter-reset LoRA?**

Understanding this dynamic is important for domain adaptation, safety and alignment, and model editing. This thesis investigates both the empirical behavior and the underlying training dynamics that govern resistance to forgetting.

2. Work Completed During the First Semester

2.1. Stagewise Mixture Experiments

The experiments conducted this semester fall into two distinct formulations. Both study how the placement and repetition of domain data P affect retention, but they differ in how P is allocated and reused.

Corresponding author(s): lawrencf@andrew.cmu.edu

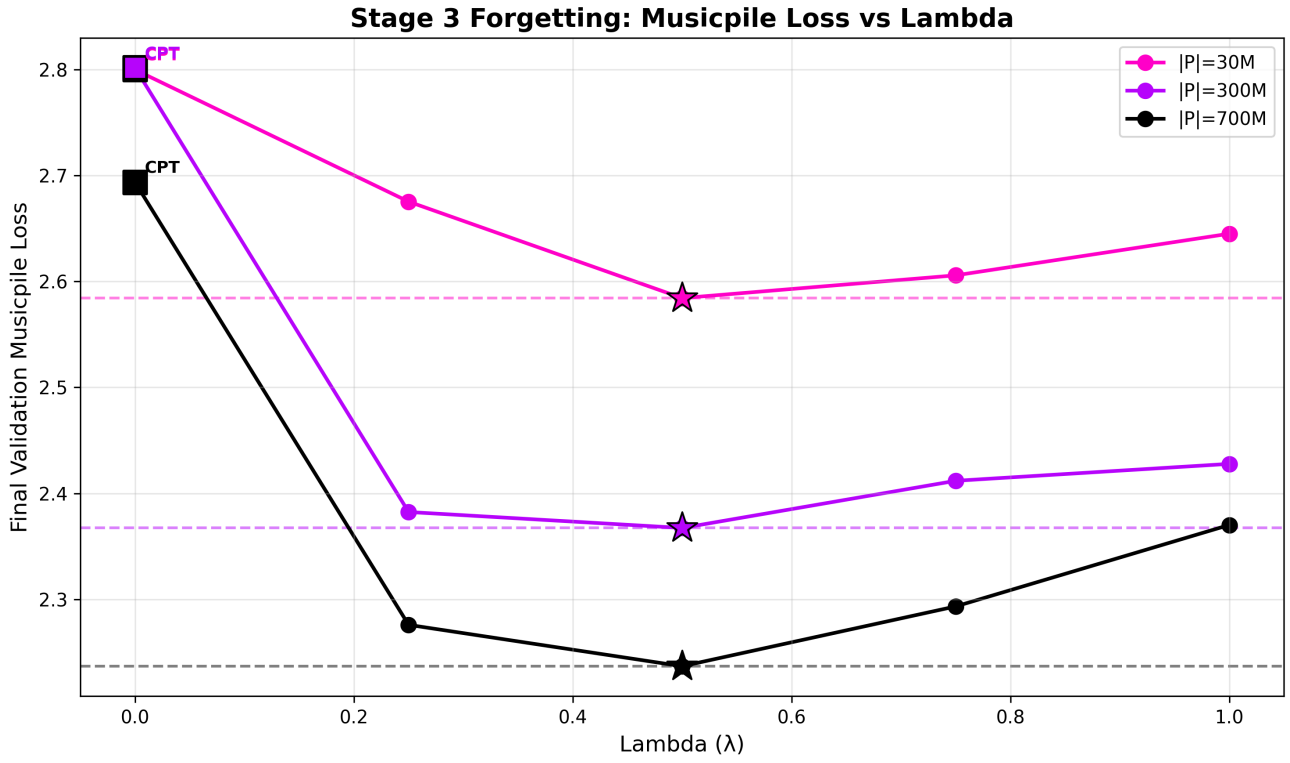

Figure 1: Compute-matched mixture experiments. Each curve corresponds to models trained with identical FLOPs but different mixture ratios λ.

2.1.1. Compute-Matched Formulation

In the compute-matched setting, the total number of tokens processed and the total FLOPs are held constant across all runs. The domain dataset P is *partitioned* between pretraining and continual pretraining:

$$\textbf{Compute-Matched:} \quad \text{Stage 1: C4} + \lambda P \quad \rightarrow \quad \text{Stage 2: } (1-\lambda)P \quad \rightarrow \quad \text{Stage 3: C4 (forgetting)}.$$

Here, P is consumed exactly once: λP in Stage 1 and $(1-\lambda)P$ in Stage 2, so every setting uses the same amount of compute.

This formulation isolates the effect of *when* the model sees domain data, since the total amount of supervision is identical across λ.

Key findings.

- Even small mixture ratios (3–4%) strongly reduce Stage 3 forgetting.
- $\lambda = 0.5$ typically achieves the best Stage 2 performance, balancing early mixture with concentrated, recent exposure during CPT.
- $\lambda = 1.0$ eliminates Stage 2 exposure entirely, leading to poorer specialization despite strong forgetting resistance.

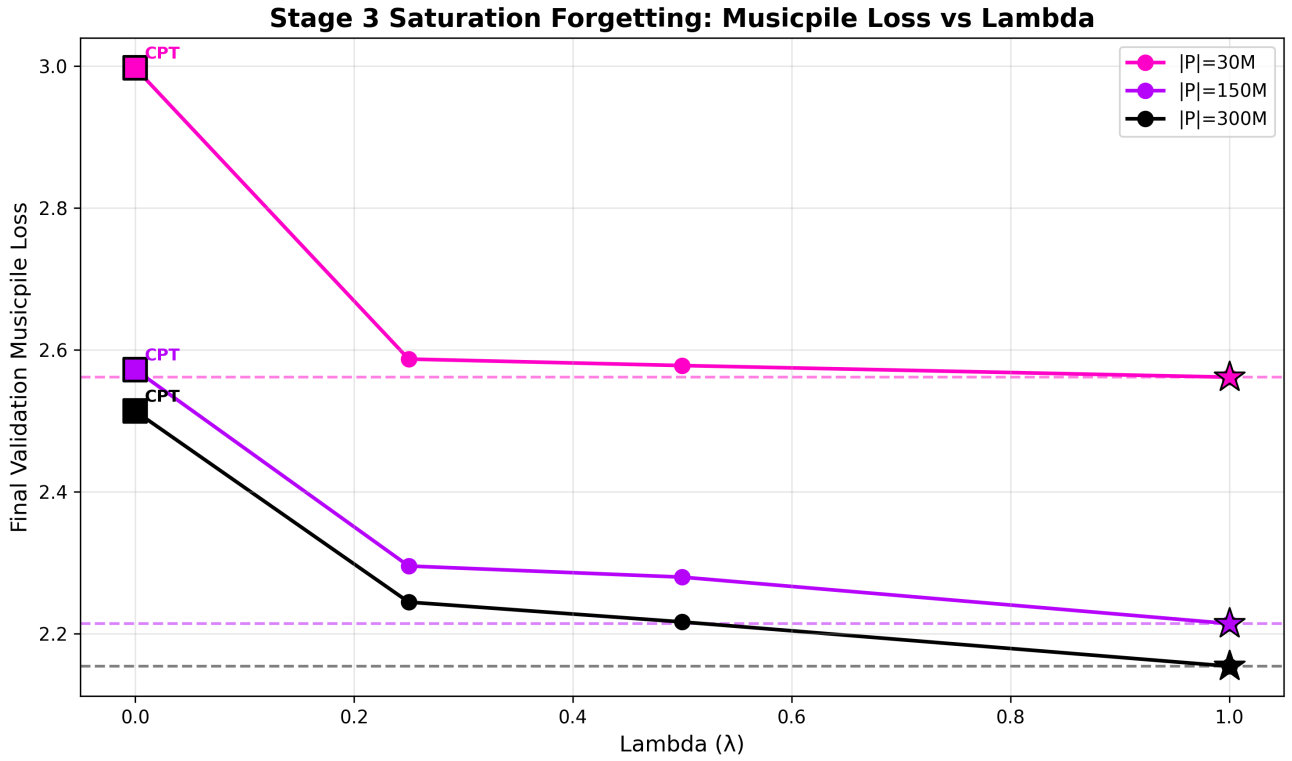

Figure 2: Data-matched experiments. Each point corresponds to a fixed corpus size $|P|$ allocated differently between Stage 1 and Stage 2 (with multi-epoch saturation in Stage 2).

2.1.2. Data-Matched Formulation

In the data-matched setting, the practitioner has a fixed corpus of domain data P (e.g., 30M–300M MusicPile tokens). Unlike the compute-matched setup, this corpus may be *reused* across stages. Stage 1 consumes $\lambda|P|$ tokens, while Stage 2 repeatedly cycles over the *entire* $|P|$ until convergence:

Data-Matched: Stage 1: C4 + $\lambda|P|$ $\rightarrow$ Stage 2: $|P|$ (multi-epoch saturation) $\rightarrow$ Stage 3: C4 (forgetting).

Thus, the total exposure to domain data depends on the number of epochs in Stage 2:

$$\text{Total domain tokens seen} = \lambda|P| + E \cdot |P|, \quad \text{where } E \text{ is the number of Stage 2 epochs.}$$

During Stage 2, training continues until the model *saturates* on the domain—that is, until additional epochs produce negligible improvements. We employ early stopping to prevent overfitting.

Key findings.

- Mixing in a small amount of a downstream task into pretraining may result in neglible performance gains after pretraining.
- However, the benefit of mixing emerges after further fine-tuning and training of unrelated domains. Mixing in even a tiny bit of the specialized domain into pretraining causes resistance to forgetting.

3. Justification for Deviations from the Original Proposal

Two small adjustments were made to the original plan.

Adopting a staged training framework. As the project progressed, it became useful to organize experiments into a multi-stage training setup (pretraining, domain exposure, forgetting). This provided a clearer way to study how and when models retain or overwrite information.

Adding finetuning baselines. The project now includes LoRA and adapter-reset LoRA as additional methods for analyzing how optimization structure affects factual retention. These methods complement the core research questions and support more controlled comparisons.

These refinements better structure the investigation without altering the overall direction of the project.

4. Plan for Next Semester

Next semester's work falls into three areas:

Completing the LoRA/ReLoRA comparison. The training infrastructure for both standard LoRA and adapter-reset LoRA (ReLoRA) is implemented and validated, with initial runs underway. The primary goal is to evaluate these methods within the same three-stage framework used for the mixture experiments. I will measure how each method retains domain behavior after exposure to unrelated data, and whether their update structure approximates the robustness benefits of pretraining-stage mixture.

Analyzing representations. Beyond behavioral metrics, I will examine whether ReLoRA produces more distributed or stable internal representations than standard LoRA. This includes studying activation patterns, attention distributions, and other diagnostics that indicate how and where domain knowledge is stored—and whether these correlate with forgetting resistance.

Extending to additional domains. The current experiments use MusicPile as the specialized domain. To determine whether the observed effects generalize, I will apply the same methodology to at least one additional domain. This will help distinguish domain-specific artifacts from general properties of data ordering and optimization.