(How) Learning Rates Regulate Catastrophic Overtraining

Mark Rofin, Aditya Varre, Nicolas Flammarion
EPFL

Abstract

Supervised finetuning (SFT) is a common first stage of LLM post-training, teaching the model to follow instructions and shaping its behavior as a helpful assistant. At the same time, SFT may harm the fundamental capabilities of an LLM, particularly after long pretraining: a phenomenon known as *catastrophic overtraining* (Springer et al., 2025). To understand overtraining, we first investigate catastrophic forgetting in finetuning through the lens of implicit regularization of the learning rate. For models trained to *the same SFT loss*, we identify how the learning rate mediates optimization: finetuning with large and small steps converges to qualitatively different models. Next, we link forgetting to overtraining: learning rate decay increases the sharpness of the pretrained model, which in turn exacerbates catastrophic forgetting during SFT, leading to overtraining. Our findings paint a picture of the overtraining mechanism in LLMs and broadly contribute to the understanding of the interplay between optimization dynamics during pretraining and finetuning.[1]

1 Introduction

Modern Large Language Models are built in the pretraining/post-training paradigm: a large-scale pretraining on unsupervised text corpora produces a generally capable base model, which is finetuned on instruction-following data and later optimized for sound reasoning or human preferences with reinforcement learning. The second stage, known as Supervised Finetuning, or SFT, is necessary to obtain a good-enough initialization for the RL phase. Without SFT, the model may struggle to get reliable training signal during RL or produce unreadable chains-of-thought (Ouyang et al., 2022; Guo et al., 2025).

SFT is successful if it results in the model acquiring the instruction-following capability without sacrificing any of its general skills. This ideal case is, however, almost unachievable in practice, due to the problem of *catastrophic forgetting*: machine learning models trained to perform one task often lose their competence when sequentially finetuned on another (Goodfellow et al., 2013). Catastrophic forgetting is known to apply to LLMs (Kleiman et al., 2023), for whom the problem can be formalized as follows.

A model θ is pretrained for T steps, yielding a base model $\theta_{\text{base}} := \theta_T$. That model is later finetuned on the SFT dataset $\mathcal{D}_{\text{FT}}$, yielding a finetuned model $\theta_{\text{FT}} := \text{FT}(\theta_{\text{base}})$. Model performance is evaluated on two datasets: SFT dataset $\mathcal{D}_{\text{FT}}$ and out-of-distribution (with respect to SFT) dataset $\mathcal{D}_{\text{OOD}}$. If finetuning was successful, the model improved on SFT dataset and $\mathcal{L}(\theta_{\text{FT}}, \mathcal{D}_{\text{FT}}) < \mathcal{L}(\theta_{\text{base}}, \mathcal{D}_{\text{FT}})$. However, its performance on OOD tasks might have worsened: if $\mathcal{L}(\theta_{\text{FT}}, \mathcal{D}_{\text{OOD}}) \gg \mathcal{L}(\theta_{\text{base}}, \mathcal{D}_{\text{OOD}})$, we state that the model has suffered catastrophic forgetting.

In a recent work, Springer et al. (2025) have uncovered a surprising new dimension in the phenomenon of forgetting: *catastrophic overtraining*. Specifically, it is observed that prolonged pretraining increases forgetting. That is, if θ_{t_1} and $\theta_{t_2}, t_1 < t_2$ are two checkpoints taken during pretraining, then $\mathcal{L}(\text{FT}(\theta_{t_1}), \mathcal{D}_{\text{OOD}}) < \mathcal{L}(\text{FT}(\theta_{t_2}), \mathcal{D}_{\text{OOD}})$ even though $\mathcal{L}(\theta_{t_1}, \mathcal{D}_{\text{OOD}}) > \mathcal{L}(\theta_{t_2}, \mathcal{D}_{\text{OOD}})$: a better pretrained model leads to a worse finetuned model for standard training. Understanding this phenomenon has some urgency for the future

[1]The code is available at `https://github.com/tml-epfl/eos-overtraining`.

of LLMs: if it is indeed true that increasing pretraining budget results in stronger forgetting, the paradigm of scaling pretraining data needs to be adjusted. Springer et al. (2025) attribute this effect to *progressive sensitivity to noise*, hypothesizing that adding more tokens in pretraining leads to higher plasticity of models to downstream modifications, but the cause for progressive sensitivity itself has so far remained unclear.

One potential explanation is that catastrophic forgetting/overtraining are statistical artifacts arising from the fact that any model is limited in capacity by its parameter count and cannot fit both pretraining and finetuning data. For a fixed capacity, the better an LLM is in modeling the pretraining distribution, the more it has to sacrifice to fit the finetuning one, yielding progressive sensitivity. However, recently Lin et al. (2026) found that using low learning rates (LR) during SFT can partially mitigate forgetting, suggesting that there is more to the story than just model capacity: optimization dynamics play a role as well.

In this work, we take on an optimization view on catastrophic forgetting. We investigate how finetuning LR controls the trajectory of a model in parameter space, particularly its deviation from regions where pretrained features are preserved. Furthermore, we connect our findings to *progressive sharpening* (Cohen et al., 2021), a well-known fact in optimization literature: overtrained models are sharper and shifted towards plasticity on the "stability-plasticity tradeoff", being more prone to learning new and forgetting old features. Concretely, our contributions are as follows:

- We identify an implicit regularization of low finetuning LRs toward preserving base model features and capabilities (Section 3).
- We explain this regularization from the perspective of loss landscapes and show that it is connected to the notion of model sharpness, having our intuition supported in a diagonal-network model of finetuning (Section 4).
- We place our results in context of progressive sharpening and edge of stability, ultimately attributing overtraining to the increase in sharpness of a base model caused by LR decay in pretraining (Section 5).

2 Background

Catastrophic forgetting in LLM finetuning. *Catastrophic forgetting* is a well-known problem in machine learning, referring to the phenomenon that models trained to perform one task may lose their capabilities when finetuned to perform another (Goodfellow et al., 2013; Kirkpatrick et al., 2017). LLMs are known to suffer from this problem as well (Kleiman et al., 2023; Wang et al., 2024; Sun & Dredze, 2025; Jin et al., 2025; Shenfeld et al., 2026). Due to its importance there is a broad stream of works aiming to understand and improve SFT in general (Pareja et al., 2025; Kotha & Liang, 2026; Han et al., 2026) and in particular make it forget less (Biderman et al., 2024; Sanyal et al., 2025; Bansal & Sanghavi, 2025; Patil et al., 2026).

Progressive sharpening. A related finding coming from the optimization literature is *progressive sharpening* of neural networks (Cohen et al., 2021). *Sharpness* of the model θ is defined as the top eigenvalue of its loss Hessian (although other definitions also exist, see Andriushchenko et al. (2023a) for the relationship between them). It has been shown that optimization dynamics of gradient descent lead to the gradual increase of sharpness during training, until it reaches a threshold $2/\eta$, where η is the LR. After reaching this threshold, the model enters *the edge of stability regime*, where sharpness oscillates around $2/\eta$.

There is no definitive proof that LLMs follow the same dynamic since they use stochastic optimization, (typically) single-pass training, and adaptive optimizers that account for sharpness in their preconditioner (Cohen et al., 2023). However, some evidence consistent with this hypothesis exists: Belloni et al. (2025); Wen et al. (2025); Yano et al. (2026) showed that decreasing LR in pretraining lets the model enter the regions of higher sharpness. Catalan-Tatjer et al. (2025) found that quantization error increases when LR is decayed, which points to higher sharpness as well.

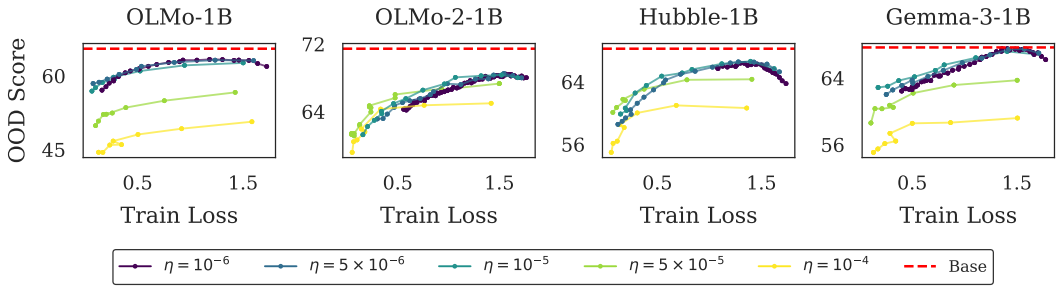


Figure 1: SFT training loss vs OOD performance for different models and LRs. Checkpoints taken throughout finetuning are shown by dots and connected within each run. The horizontal red line represents the performance of a base model pre-finetuning (without the chat template). All models lose a part of capabilities due to catastrophic forgetting. The curves for higher LRs lie below the ones for low LRs, indicating stronger forgetting.

Sharpness is relevant as it is often linked to neural networks' ability to generalize (Keskar et al., 2017; Liu et al., 2023), although the relationship may not be straightforward (Jiang et al., 2020; Andriushchenko et al., 2023a). Li et al. (2024a) and Yano et al. (2026) found that model sharpness is negatively correlated with both in- and out-of-distribution performance after LLM finetuning. Concurrently with our work, Watts et al. (2026) discovered that controlling sharpness in pretraining mitigates catastrophic forgetting. Closest to our work, Lin et al. (2026) observed that smaller LRs in finetuning yield a better ID-OOD tradeoff, which may be attributed to easier navigation of sharp loss regions. Lin et al. (2026), however, focus on behavioral metrics of training with variable LRs and do not investigate the mechanistic differences between them, providing a contribution distinct from our work.

Implicit regularization. Implicit regularization in overparameterized networks (Neyshabur et al., 2014) refers to the property of an optimization algorithm whereby, through its choice of hyperparameters, it biases the training trajectory toward a particular minimum among many possible solutions (Soudry et al., 2018). Prior work has studied the implicit regularization effects of factors such as initialization (Woodworth et al., 2020), stochastic noise (Blanc et al., 2020; Varre et al., 2024), and the LR (Andriushchenko et al., 2023a), showing that, depending on these choices, training can operate either in a lazy regime (Chizat et al., 2019)—retaining features at initialization—or in a rich (feature-learning) regime (Yang & Hu, 2021).

3 Low and High Learning Rates Lead to Qualitatively Different Models

Before discussing overtraining and the properties of different pretraining checkpoints, we first establish a foundational understanding of catastrophic forgetting in standalone models. Springer et al. (2025) observed that catastrophic overtraining manifests stronger with high finetuning LRs. This is also corroborated by the findings of Lin et al. (2026). We aim to understand the reason behind it in this section, but first, we validate this premise.

3.1 Do Lower Learning Rates Forget Less?

Experimental setup. We follow the setup of Springer et al. (2025) with a few modifications. We consider 4 models: OLMo 1 and 2 (Groeneveld et al., 2024; OLMo Team et al., 2024), Hubble (Wei et al., 2025), Gemma 3 (Gemma Team et al., 2025) (each with 1B parameters). We finetune each of these models with a range of LRs until the model overfits. Anthropic-HH (Bai et al., 2022) is the dataset we use throughout the paper, we also present the results for Tülu 3 (Lambert et al., 2025) in the Appendix. To eliminate the LR schedule as a confounding factor, we adopt a constant schedule with warmup, motivated by the observation of Pareja et al. (2025) that, for SFT, it achieves performance comparable to the typically used cosine schedule. We track two main metrics: *train SFT loss*, measured on the finetuning datasets and showing how well the model fits the SFT dataset, as well as the *OOD score*, measured on 6 tasks evaluating the general capabilities of the model. See Appendix A for experimental details.

Note that, as in Springer et al. (2025), we use "OOD" to refer to all tasks that are out of distribution *with respect to SFT*, that is, all tasks not directly related to instruction following. Thus, the OOD score can be interpreted as a measure of base LLM capabilities. This contrasts with some prior work where "OOD" refers to tasks or examples that are similar to the training distribution but differ from it in certain properties.

Results. Figure 1 presents the results of the experiment. First, SFT always degrades the OOD performance: as expected, finetuning on instruction-following data is unlikely to introduce new fundamental capabilities into the model.

Second, we observe a striking effect: when finetuned to a fixed SFT loss value, models trained with higher LRs during SFT exhibit lower OOD performance. This gap cannot be trivially attributed to differences in the magnitude of parameter updates from the pretrained model, as all variants reach the same training loss and therefore move equivalent distances with respect to the SFT objective. However, the choice of LR alone steers the model toward regions of the parameter space associated with differing OOD performance, highlighting an implicit regularization effect. Specifically, lower LRs better preserve the base model's capabilities, whereas higher LRs lead to more pronounced forgetting.

Models finetuned with low learning rate suffer less from catastrophic forgetting.

3.2 Measuring the Feature Drift with Different Learning Rates

The implicit regularization effect of the LR has been widely studied in supervised learning (Keskar et al., 2017; Xing et al., 2018). Larger LRs operate in a rich regime, favoring flatter minima and enabling feature learning (Li et al., 2019; Andriushchenko et al., 2023b), whereas smaller LRs remain in the lazy regime. Building on these insights, we investigate the role of implicit regularization in finetuning. We hypothesize that finetuning with a low LR operates in a lazy regime, thereby preventing the features from drifting from pretraining, whereas higher LRs promote the learning of task-specific features and, in turn, lead to overwriting of some pretrained representations.

We now introduce a metric to quantify similarity to the pretrained model. Specifically, we measure the mean principal angle (MPA) between the residual stream representations of different models. Let X^L_{base} and X^L_{FT} be (centered) activation matrices of the base and finetuned models at residual stream of layer L, and $X^L_{\text{base}} = U_{\text{base}} S_{\text{base}} \begin{bmatrix} Q_{\text{base}} \\ \tilde{V}^\top_{\text{base}} \end{bmatrix}$, $X^L_{\text{FT}} = U_{\text{FT}} S_{\text{FT}} \begin{bmatrix} Q_{\text{FT}} \\ \tilde{V}^\top_{\text{FT}} \end{bmatrix}$, their SVD decompositions with $Q_{\text{base}}, Q_{\text{FT}} \in \mathbb{R}^{k \times d}$ being the top-k right singular vectors. We then compute the metric

$$\text{MPA}(X^L_{\text{base}}, X^L_{\text{FT}}) = \frac{1}{k} \sum_{i}^{k} \arccos\big(\sigma_i(Q_{\text{base}} Q^\top_{\text{FT}})\big),$$

where $\sigma_i(\cdot)$ is the i-th singular value. For identical models, MPA is 0, and increases as the finetuned model diverges from its initialization, with its value interpretable as an angle in radians between subspaces formed by the models' residual streams. Similar metrics are commonly used to estimate subspace alignment (Chi et al., 2020; Zhou et al., 2025)[2]. We also study feature drift in Sparse Autoencoders for Gemma 3 (DeepMind, 2025) in Appendix B.1 and find the same pattern.

The results for OLMo are shown in Figure 2 and for other models in Appendix B.3.1. Across models, finetuning with high LRs exhibits larger principal angles than models finetuned with low LRs – again, *for any fixed level of training loss*. Thus, attaining the same optimization objective corresponds to greater deviation from the base model under higher LR. In contrast, lower LRs operate in a lazy regime, leading to smaller principal angles and better preservation of the base model's features. This makes the difference in forgetting

[2]We have also experimented with CKA (Kornblith et al., 2019) but found its dynamics during finetuning to be too noisy to get meaningful signal.

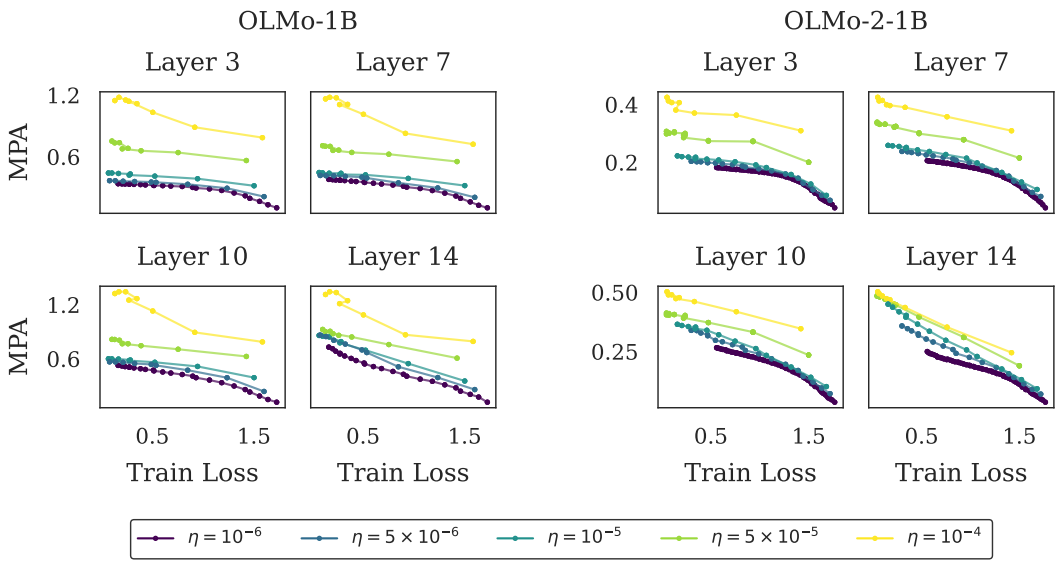


Figure 2: SFT training loss vs MPA between the representations of the base and finetuned models throughout SFT (defined in Section 3.2). The angles increase as SFT progresses and the model moves from its initialization; but for any fixed level of training loss, higher LRs cause more rotation.

severity natural: the more the *features drift* after SFT, the less knowledge of the base model is preserved.

> Representations of models finetuned with low LR diverge less from the base model compared to the ones of models finetuned with high LR.

4 Feature Drift is Mediated by Learning Rate and Sharpness

In this section, we first take a closer look at the optimization landscape to provide insights into how the learning rate induces feature drift during finetuning. We then consider simplified settings to study how such feature drift manifests in models where feature learning is well understood. Finally, we connect these observations to the sharpness of the loss landscape and demonstrate how it influences feature drift.

4.1 Learning Rate Controls Feature Drift

To understand why smaller learning rates lead to reduced feature drift, we plot the dynamics of SFT loss and MPA for OLMo-1 1B under different learning rates in Figure 3a,b. We observe that MPA increases rapidly in the initial training steps, with the ordering across learning rates emerging early and then continuing to grow more gradually, while the SFT loss remains similar across learning rates. Overall, MPA is highly sensitive to the choice of learning rate, with most of this effect arising early in training.

To shed light on this sensitivity, we focus on the first 100 steps of finetuning and analyze the loss landscape the optimizer traverses during this period. Specifically, for each update from θ_t to θ_{t+1}, we compute the MPA and SFT loss along the interpolated trajectory $(1-\alpha)\theta_t + \alpha\theta_{t+1}$, for $0 \leq \alpha \leq 1$. This construction allows us to quantify how model properties evolve along each optimization step[3]. The results are shown in Figure 3c,d. First, we observe that the loss is, on average, convex with respect to LR. Second, the increase in MPA is approximately linear in the LR.

These two observations account for the differing behavior of MPA and loss. Specifically, increasing the learning rate amplifies MPA more than it reduces the loss per training step. To formalize this, consider the ratio $\alpha/\ell(0)-\ell(\alpha)$, where ℓ is convex in α and $\ell(\alpha) \leq \ell(0)$ on

[3]A similar technique was used by Belloni et al. (2025) to analyze Weight-Stable-Decay LR schedule.

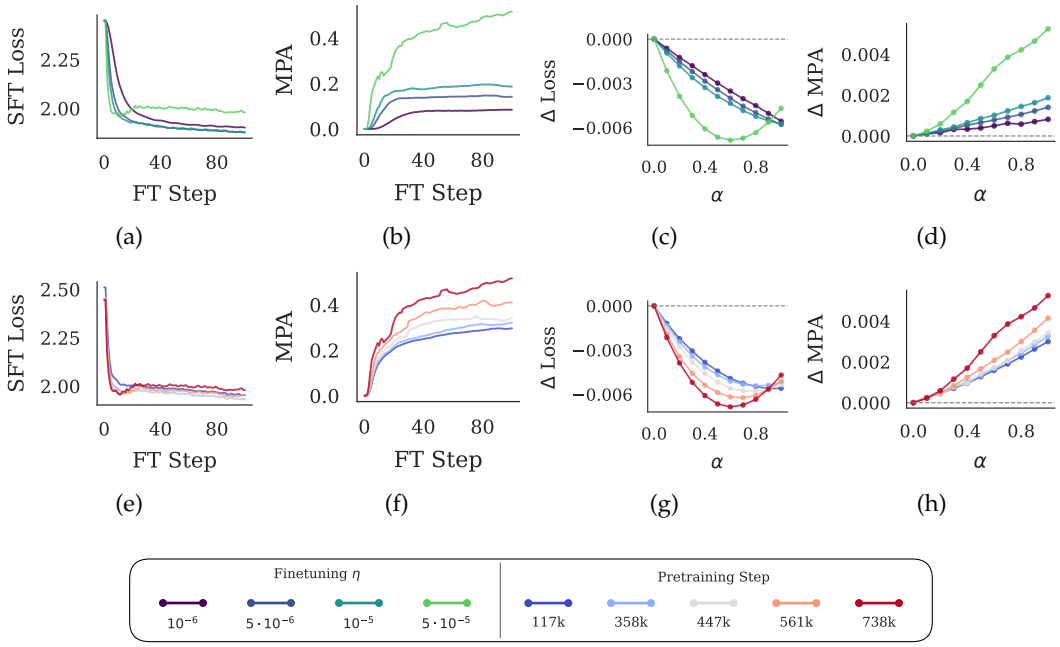


Figure 3: Loss landscape analysis for OLMo 1 1B. Panels a,b,e,f: the dynamics of SFT loss and MPA (layer 10) during the start of finetuning. Panels c,d,g,h: ΔLoss and ΔMPA along each gradient step. Top row: fixed checkpoint, each line is a different LR. Bottom row: fixed LR, each line is a different checkpoint.

the interval of interest. This ratio increases with α, implying that for $\alpha_1 < \alpha_2$, the ratio is larger at α_2. Consequently, over multiple steps, achieving the same total decrease in loss results in a larger accumulated increase in MPA for higher learning rates due to the ratio.

When scaling the LR, similarity between finetuned and base model decreases faster than train loss.

4.2 Feature Drift in a Two-Layer Diagonal Network

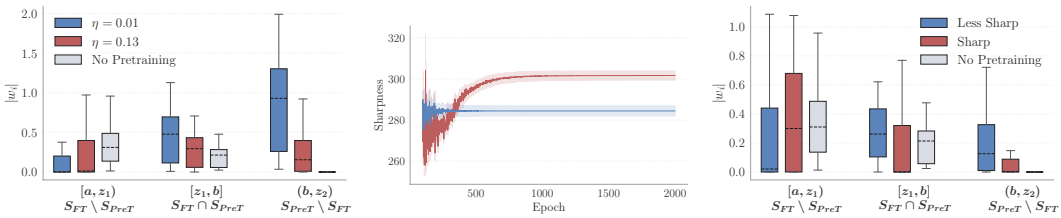
To ground and test the generality of this feature drift and its dependence on the LR, we consider a simple and well-studied feature learning setting: sparse regression with diagonal linear networks (Woodworth et al., 2020). We modify this setup to capture the trade-off between pretrained and finetuned features, mediated by the learning rate.

Features. For integers $a < b$, let $[\![a, b]\!] := \{a, a+1, \ldots, b\}$. Each coordinate of the regressor corresponds to a feature. We assume that the pretrained model and the finetuning task have feature sets $\mathcal{S}_{\text{FT}} = [\![a, b]\!]$ and $\mathcal{S}_{\text{PreT}} = [\![z_1, z_2]\!]$, where $0 \leq a < z_1 < b < z_2$. The two sets overlap on $[\![z_1, b]\!]$; features in $[\![a, z_1 - 1]\!]$ are specific to finetuning, while those in $[\![b-1, z_2]\!]$ are specific to pretraining. Let w_{FT} denote the ground-truth weights for the finetuning task, with $\text{support}(w_{\text{FT}}) = \mathcal{S}_{\text{FT}}$, and let w_{PreT} denote the pretrained model with $\text{support}(w_{\text{PreT}}) = \mathcal{S}_{\text{PreT}}$.

Finetuning. Let $X \in \mathbb{R}^{n \times d}$ denote the finetuning data and $Y = X w_{\text{FT}} \in \mathbb{R}^n$ the corresponding labels. We assume $n < d$, i.e., the model is overparameterized, reflecting the data-limited regime of finetuning. We consider a two-layer diagonal network parameterized by u, v, with objective $1/2 \|X(u \odot v) - Y\|^2$, where $\odot$ denotes the Hadamard product. We initialize the model with $u_0 \odot v_0 = w_{\text{PreT}}$ and perform mini-batch SGD with different step sizes $\eta_s < \eta_l$, training all models to near-zero training loss. The resulting solutions are shown in Figure 4a. A randomly initialized model at small scale recovers the finetuning features $\mathcal{S}_{\text{FT}}$ (Woodworth et al., 2020). In contrast, when initialized with pretrained features, large learning rates lead to substantial forgetting of features in $\mathcal{S}_{\text{PreT}} \setminus \mathcal{S}_{\text{FT}}$, i.e., features present in pretraining but not required for finetuning. For small learning rates, the model remains

closer to the pretrained features while still fitting the finetuning task, consistent with a lazy regime. This mirrors the phenomenon observed in the previous section, suggesting that it is more general and can already be observed in a simple toy model.

This observation is consistent with the theoretical results of Even et al. (2023), which show that SGD with large LRs operates near the stability threshold given by sharpness, promoting feature learning and driving training into the *rich regime* by effectively dampening the initialization α_{eff}. In contrast, small LRs remain in the *lazy regime*. This connection to sharpness and stability raises a natural question: how does the sharpness of the pretrained model influence this behavior? For a fixed LR, models in sharper regions operate closer to the stability threshold and, by spending more time in such regions, are more likely to enter the rich regime and exhibit greater feature learning. Figures 4b, 4c support this picture, showing that more time spent in sharper regions correlates with greater forgetting. This perspective also provides insight into overtraining: if sharpness induced during pretraining carries over to finetuning, then smaller pretraining learning rates may increase sharpness, which in turn can lead to increased forgetting during finetuning.



(a) (b) (c)

Figure 4: Finetuning with Diagonal Networks. We train a diagonal network with $d = 100$ using mini-batch SGD (10 runs) and plot the final distribution of absolute value of coordinates of weights. Here $(a, b, c, d) = (0, 23, 45, 66)$. In (a), when initialized without pretraining features, the model recovers the finetuning features, while larger step sizes lead to greater forgetting of pretraining features, as seen in the region $\mathcal{S}\text{FT} \triangle \mathcal{S}\text{PreT}$. In (b,c), two runs with different initial sharpness and with same LR. Sharpness is measured as $\text{Tr}(\nabla^2 L)$. We observe that the more time the model spends in sharper regions, the greater the forgetting.

4.3 Sharpness Amplifies Feature Drift

Above, we established a connection between feature drift and sharpness. In particular, sharpness also provides a natural reference scale for distinguishing between "small" and "large" LRs. It is well known that the behavior of a given LR is tightly coupled to the sharpness of the loss landscape (see Section 2). In particular, stable optimization typically requires the LR to scale inversely with sharpness; when the LR is too large relative to the local sharpness, the optimizer becomes unstable and is effectively "catapulted" out of the region (Lewkowycz et al., 2020).

We hypothesize that pretraining sharpness plays a key role during finetuning. Intuitively, sharper pretrained models correspond to initialization in sharper regions of the finetuning loss landscape, where feature learning is more sensitive to optimization steps. Consequently, we expect sharper models to exhibit stronger forgetting under high learning rates.

To test this hypothesis, we repeat our experiment in Section 4.1 fixing the finetuning LR and controlling the sharpness of the initialization. To control sharpness, we pick the snapshots of the base model at different pretraining steps (as we show in Section 5.1, later checkpoints are sharper). We plot the results in Figure 3 (bottom row).

A striking observation is that the effect of controlling the sharpness of the initialized model is *exactly the same* as the effect of controlling the LR: later (and sharper) checkpoints are more prone to overshooting the loss minima per step, and exhibit higher slope of MPA increase by stepsize. These results support that *LR and sharpness of the base model are coupled:* both are the toggles controlling the rate of loss decrease/MPA increase per step in a similar fashion.

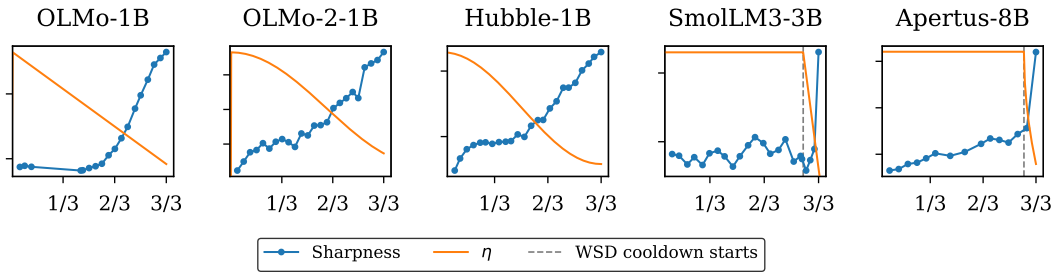


Figure 5: Evolution of model sharpness throughout pretraining (x-axis reflects pretraining progress). Sharpness grows as LR is decayed; for two models trained with WSD schedule, a rapid increase in sharpness coincides with the LR cooldown phase.

The similarity/train loss gap is widened by sharper base models.

5 Catastrophic Overtraining is Caused by Sharpening

In Section 4, we have seen that feature drift depends on finetuning LR and sharpness of the pretrained model. Here, we will make the connections with these observations and the phenomenon of overtraining concrete.

5.1 Learning rate decay during pretraining causes sharpness to increase

It is well-known that sharpness of neural nets tends to increase during pretraining until reaching a plateau whose level is dictated by pretraining LR: an effect that is known as *Edge of Stability* (EoS, Cohen et al. (2021)). For a constant LR η, due to EoS, the top eigenvalue of the loss Hessian reaches the point of $2/\eta$ and then oscillates. During LLM pretraining, however, LR is typically progressively decreased. Thus, the threshold $2/\eta$ increases throughout training, and hence one can expect sharpness to grow when LR is decayed.

We test this by measuring sharpness of the pretraining checkpoints of 5 LLMs with available pretraining checkpoints. Since measuring true sharpness defined through Hessian eigenvalues is computationally infeasible for large models, we measure a proxy evaluating how much models' predictions change after a Gaussian perturbation of parameters:

$$\text{est.sharpness}(\theta) = \mathbb{E}_{x\sim\mathcal{D}_{\text{pretraining}},\,\varepsilon\sim\mathcal{N}(0,\sigma^2)}\Big[D_{\text{KL}}\big(\text{LM}_{\theta+\varepsilon}(x), \text{LM}_{\theta}(x)\big)\Big].$$

We verify that this proxy reflects the true sharpness for a 14m model in Appendix B.2.

Figure 5 demonstrates that it is, indeed, true for a range of pretrained models: sharpness increases during training. Moreover, for two models we test that use the Warmup-Stable-

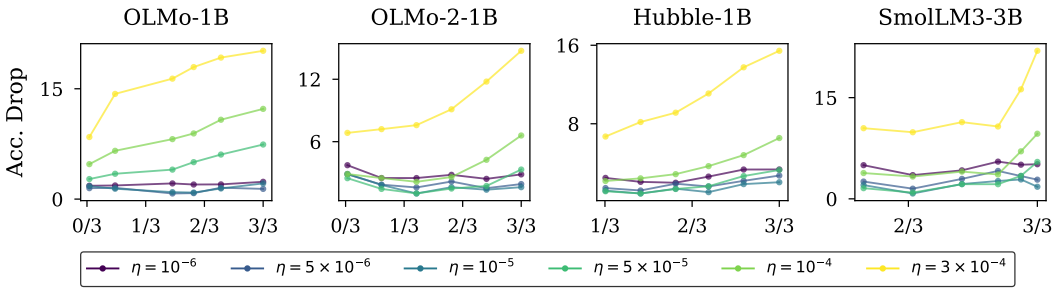


Figure 6: The dynamic of accuracy drop (the difference between the OOD score between the pretrained and finetuned models) throughout pretraining (x-axis shows the pretraining progress). Forgetting increases in later checkpoints.

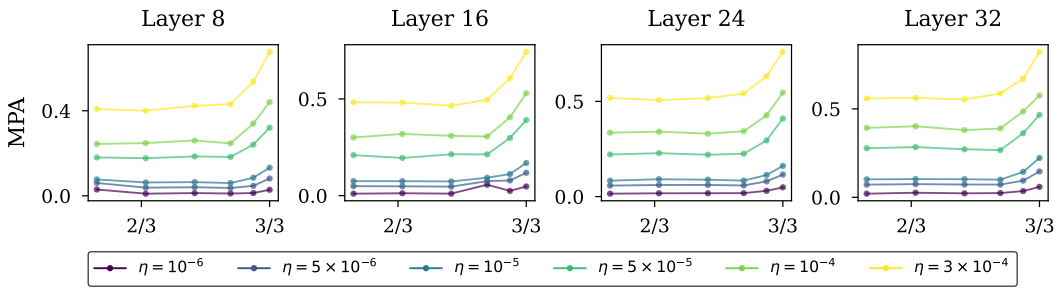


Figure 7: The principal angles between representations of base and finetuned SmolLM3-3B as a function of pretraining steps. The representational dissimilarity stays constant until the pretraining LR is decreased, when it surges.

Decay (WSD, Hu et al. (2024)) LR schedule (Apertus 8B (Project Apertus et al., 2025) and SmolLM 3B (Bakouch et al., 2025)), we observe that sharpness is flat when LR is stable, and increases dramatically once LR is decreased. Since the timing of LR decrease is a controlled hyperparameter, this points to a causal connection between LR decay and sharpness.

LR decay during pretraining increases the base model sharpness.

5.2 Increased sharpness induces stronger forgetting

In Section 4, we have shown that for a fixed finetuning LR, the features of sharper checkpoints diverge more from their initializations. In Section 5.1, we have shown that sharpness increases as pretraining LR is decayed. Combined, these findings imply that forgetting and feature drift must get stronger with LR decay in pretraining. This prediction is tested here.

We largely follow the setup of Springer et al. (2025) and train for a fixed number of finetuning steps, starting from different checkpoints during pretraining and measuring forgetting together with feature drift. The results for forgetting are shown in Figure 6, and indeed, forgetting for high finetuning LRs escalates throughout training. Most strikingly, SmolLM shows the sudden increase in forgetting exactly where its pretraining LR is decayed and sharpness grows. We also plot the principal angles between the representations of SmolLM at 4 different layers in Figure 7 and observe a similar pattern: feature drift increases during the cooldown phase of WSD, simultaneously with forgetting.

Base checkpoints obtained after LR decay are sharper, and therefore forget more when finetuned with a fixed LR.

6 Conclusion

In this paper, we studied catastrophic forgetting in LLMs and its amplification over pretraining, known as catastrophic overtraining. We first showed that, for *a fixed base checkpoint and varying finetuning LRs*, lower finetuning LRs implicitly regularize toward preserving pretrained features, thereby mitigating forgetting. We then explained this effect through the different per-step scaling of loss and feature drift with LR. We reproduced the same phenomenon in diagonal-network models and found that increasing model sharpness has an effect analogous to increasing the finetuning learning rate. This connection led us to examine *fixed finetuning LR across different base checkpoints*. Since lower pretraining LRs increase base-model sharpness, our results suggest that overtraining is driven by LR decay rather than token budget alone. Our experiments support this view.

Our findings tell a dual story of the influence of LRs on catastrophic forgetting and overtraining. Lower *finetuning* LRs preserve base-model features better and reduce forgetting, whereas lower *pretraining* LRs increase sharpness and make forgetting worse. This suggests two practical recommendations:

Use the smallest finetuning LR that doesn't compromise in-distribution performance – as also suggested by Lin et al. (2026).

Control sharpness of the pretrained model. The latter can be achieved in different ways studied in recent work: using sharpness-aware optimization algorithms (Foret et al., 2021), pretraining with higher LR or shortening the LR decay period (Watts et al., 2026), or even discarding pretraining LR decay completely (Yano et al., 2026).

Limitations

Focus on small models. Our experiments are mostly done with the models in 1-3B parameter range. Although such models are considered small by modern standards, we find them suitable for the purposes of this paper. Both catastrophic forgetting and overtraining have been observed for the models of this size and are not emergent with scale. Therefore, we believe it is justified to study the effects manifesting at 1B scale with models of 1B scale, leaving investigations with larger models for future work.

Agnosticism to SFT performance. An important detail of our work is that we do not discuss test SFT performance, and in all figures in the main paper we report SFT *training* loss. That is done intentionally, as our goal is to study the implicit bias of the learning rate choice in finetuning, that is, for solutions found with different LRs that solve the optimization objective equally well, what are the differences in their properties. The way to measure how well the finetuning optimization objective is solved is the SFT training loss. However, this means that we cannot give recommendations for balancing forgetting and test SFT performance.

Causal connection between LR decay and sharpness. The connection we draw between the pretraining LR decay and model sharpness relies on prior work and correlational evidence, particularly the results with SmolLM. We believe these results are convincing but technically cannot claim causality.

Acknowledgements

MR thanks Raghav Singhal, Kaustubh Ponkshe, Julian Minder, and Aleksandra Bakalova for useful discussions and feedback on the early drafts of the paper. AV is supported by Swiss data science fellowship.

References

Maksym Andriushchenko, Francesco Croce, Maximilian Müller, Matthias Hein, and Nicolas Flammarion. A modern look at the relationship between sharpness and generalization. In *International Conference on Machine Learning*, pp. 840–902. PMLR, 2023a.

Maksym Andriushchenko, Aditya Vardhan Varre, Loucas Pillaud-Vivien, and Nicolas Flammarion. SGD with large step sizes learns sparse features, 2023b. URL https://openreview.net/forum?id=ipRGZ91NvG4.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Elie Bakouch, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Lewis Tunstall, Carlos Miguel Patiño, Edward Beeching, Aymeric Roucher, Aksel Joonas Reedi, Quentin Gallouédec, Kashif Rasul, Nathan Habib, Clémentine Fourrier, Hynek Kydlicek, Guilherme Penedo, Hugo Larcher, Mathieu Morlon, Vaibhav Srivastav, Joshua Lochner, Xuan-Son Nguyen, Colin Raffel, Leandro von Werra, and Thomas Wolf. SmolLM3: smol, multilingual, long-context reasoner. https://huggingface.co/blog/smollm3, 2025.

Parikshit Bansal and Sujay Sanghavi. Context-free synthetic data mitigates forgetting. *arXiv preprint arXiv:2505.13811*, 2025.

Annalisa Belloni, Lorenzo Noci, and Antonio Orvieto. Universal dynamics of warmup stable decay: understanding wsd beyond transformers. In *High-dimensional Learning Dynamics 2025*, 2025.

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 Patrick Cunningham. LoRA learns less and forgets less. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=aloEru2qCG. Featured Certification.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International conference on machine learning*, pp. 2397–2430. PMLR, 2023.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Guy Blanc, Neha Gupta, Gregory Valiant, and Paul Valiant. Implicit regularization for deep neural networks driven by an ornstein-uhlenbeck like process. In *Conference on Learning Theory, COLT 2020*, Proceedings of Machine Learning Research. PMLR, 2020.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Albert Catalan-Tatjer, Niccolò Ajroldi, and Jonas Geiping. Training dynamics impact post-training quantization robustness. *arXiv preprint arXiv:2510.06213*, 2025.

Ethan A Chi, John Hewitt, and Christopher D Manning. Finding universal grammatical relations in multilingual bert. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5564–5577, 2020.

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *Advances in Neural Information Processing Systems*, 32, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Jeremy Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. In *The Ninth International Conference on Learning Representations*, 2021.

Jeremy Cohen, Behrooz Ghorbani, Shankar Krishnan, Naman Agarwal, Sourabh Medapati, Michal Badura, Daniel Suo, Zachary Nado, George E Dahl, and Justin Gilmer. Adaptive gradient methods at the edge of stability. In *NeurIPS 2023 Workshop Heavy Tails in Machine Learning*, 2023.

Jeremy Cohen, Alex Damian, Ameet Talwalkar, J Zico Kolter, and Jason D Lee. Understanding optimization in deep learning with central flows. In *The Thirteenth International Conference on Learning Representations*, 2025.

DeepMind. Gemma scope 2: helping the ai safety community deepen understanding of complex language model behavior, December 2025. URL https://deepmind.google/blog/gemma-scope-2-helping-the-ai-safety-community-deepen-understanding-of-complex-language-model-behavior/.

Mathieu Even, Scott Pesme, Suriya Gunasekar, and Nicolas Flammarion. (S)GD over diagonal linear networks: Implicit regularisation, large stepsizes and edge of stability. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=6Tm1mposlrM.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The Pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024. URL https://zenodo.org/records/12608602.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ram'e, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean-Bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gael Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, et al. Gemma 3 technical report. *ArXiv*, abs/2503.19786, 2025.

Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013.

Dirk Groeneveld, Iz Beltagy, Evan Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. Olmo: Accelerating the science of language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15789–15809, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Tessa Han, Sebastian Bordt, Hanlin Zhang, and Sham M Kakade. Weight decay improves language model plasticity. In *Workshop on Scientific Methods for Understanding Deep Learning*, 2026.

Shengding Hu, Yuge Tu, Xu Han, Ganqu Cui, Chaoqun He, Weilin Zhao, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, et al. Minicpm: Unveiling the potential of small language models with scalable training strategies. In *First Conference on Language Modeling*, 2024.

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *The Eighth International Conference on Learning Representations*, 2020.

Hangzhan Jin, Sitao Luan, Sicheng Lyu, Guillaume Rabusseau, Doina Precup, and Mohammad Hamdaqa. RL fine-tuning heals the OOD forgetting in SFT. In *First Workshop on Foundations of Reasoning in Language Models*, 2025. URL https://openreview.net/forum?id=SN1PCQ0ApV.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*, 2017.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Anat Kleiman, Jonathan Frankle, Sham M Kakade, and Mansheej Paul. Predicting task forgetting in large language models. 2023. URL https://openreview.net/pdf?id=0BMg0OgNTP.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pp. 3519–3529. PMlR, 2019.

Suhas Kotha and Percy Liang. Replaying pre-training data improves fine-tuning. *arXiv preprint arXiv:2603.04964*, 2026.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James Validad Miranda, Alisa Liu, Nouha Dziri, Xinxi Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. In *Second Conference on Language Modeling*, 2025.

Aitor Lewkowycz, Yasaman Bahri, Ethan Dyer, Jascha Sohl-Dickstein, and Guy Gur-Ari. The large learning rate phase of deep learning: the catapult mechanism. *arXiv preprint arXiv:2003.02218*, 2020.

Hongyu Li, Liang Ding, Meng Fang, and Dacheng Tao. Revisiting catastrophic forgetting in large language model tuning. In *Findings of the association for computational linguistics: EMNLP 2024*, pp. 4297–4308, 2024a.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, et al. Datacomp-lm: In search of the next generation of training sets for language models. *Advances in Neural Information Processing Systems*, 37:14200–14282, 2024b.

Yuanzhi Li, Colin Wei, and Tengyu Ma. Towards explaining the regularization effect of initial large learning rate in training neural networks. In *NeurIPS*, 2019.

Jiacheng Lin, Zhongruo Wang, Kun Qian, Tian Wang, Arvind Srinivasan, Hansi Zeng, Ruochen Jiao, Xie Zhou, Jiri Gesi, Dakuo Wang, et al. Sft doesn't always hurt general capabilities: Revisiting domain-specific fine-tuning in llms. *The Fourteenth International Conference on Learning Representations*, 2026.

Hong Liu, Sang Michael Xie, Zhiyuan Li, and Tengyu Ma. Same pre-training loss, better downstream: Implicit bias matters for language models. In *International Conference on Machine Learning*, pp. 22188–22214. PMLR, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.

OLMo Team, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2 olmo 2 furious. *arXiv preprint arXiv:2501.00656*, 2024.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Aldo Pareja, Nikhil Shivakumar Nayak, Hao Wang, Krishnateja Killamsetty, Shivchander Sudalairaj, Wenlong Zhao, Seungwook Han, Abhishek Bhandwaldar, Guangxuan Xu, Kai Xu, et al. Unveiling the secret recipe: A guide for supervised fine-tuning small llms. In *The Thirteenth International Conference on Learning Representations*, 2025.

Darshan Patil, Ekaterina Lobacheva, Razvan Pascanu, and Sarath Chandar. Loss smoothing for stable adaptation under distribution shift. *arXiv preprint arXiv:2607.00634*, 2026.

Project Apertus, Alejandro Hernández-Cano, Alexander Hägele, Allen Hao Huang, Angelika Romanou, Antoni-Joan Solergibert, Barna Pasztor, Bettina Messmer, Dhia Garbaya, Eduard Frank Ďurech, et al. Apertus: Democratizing open and compliant llms for global language environments. *arXiv preprint arXiv:2509.14233*, 2025.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Sunny Sanyal, Hayden Prairie, Rudrajit Das, Ali Kavis, and Sujay Sanghavi. Upweighting easy samples in fine-tuning mitigates forgetting. In *International Conference on Machine Learning*, pp. 52922–52957. PMLR, 2025.

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. Rl's razor: Why online reinforcement learning forgets less. *The Fourteenth International Conference on Learning Representations*, 2026.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research*, 19(70):1–57, 2018.

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. In *International Conference on Machine Learning*, pp. 56719–56789. PMLR, 2025.

Kaiser Sun and Mark Dredze. Amuro & char: Analyzing the relationship between pre-training and fine-tuning of large language models. In *Proceedings of the 10th Workshop on Representation Learning for NLP (RepL4NLP-2025)*, pp. 131–151, 2025.

Aditya Varre, Margarita Sagitova, and Nicolas Flammarion. SGD vs GD: Rank deficiency in linear networks. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5362–5383, 2024. doi: 10.1109/TPAMI.2024.3367329.

Ishaan Watts, Catherine Li, Sachin Goyal, Jacob Mitchell Springer, and Aditi Raghunathan. Sharpness-aware pretraining mitigates catastrophic forgetting. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=CHvRfubYke.

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. *arXiv preprint arXiv:2510.19811*, 2025.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In Leon Derczynski, Wei Xu, Alan Ritter, and Tim Baldwin (eds.), *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pp. 94–106, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4413. URL https://aclanthology.org/W17-4413/.

Kaiyue Wen, Zhiyuan Li, Jason S Wang, David Leo Wright Hall, Percy Liang, and Tengyu Ma. Understanding warmup-stable-decay learning rates: A river valley loss landscape view. In *The Thirteenth International Conference on Learning Representations*, 2025.

Blake Woodworth, Suriya Gunasekar, Jason D. Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. Kernel and rich regimes in overparametrized models. In *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pp. 3635–3673, 2020.

Chen Xing, Devansh Arpit, Christos Tsirigotis, and Yoshua Bengio. A walk with sgd. *arXiv preprint arXiv:1802.08770*, 2018.

Greg Yang and Edward J. Hu. Tensor programs iv: Feature learning in infinite-width neural networks. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 11727–11737. PMLR, 18–24 Jul 2021.

Kazuki Yano, Shun Kiyono, Sosuke Kobayashi, Sho Takase, and Jun Suzuki. Pre-training llm without learning rate decay enhances supervised fine-tuning. In *The Fourteenth International Conference on Learning Representations*, 2026.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 4791–4800, 2019.

Zhenhong Zhou, Haiyang Yu, Xinghua Zhang, Rongwu Xu, Fei Huang, Kun Wang, Yang Liu, Junfeng Fang, and Yongbin Li. On the role of attention heads in large language model safety. In *The Thirteenth International Conference on Learning Representations*, 2025.

A Additional Experimental Details

A.1 Supervised Finetuning

We mostly follow Springer et al. (2025) for the finetuning/evaluation design.

Data. We use two SFT datasets: Anthropic HH (Bai et al., 2022) and Tülu 3 (Lambert et al., 2025). Since Anthropic HH is originally a preference-tuning dataset, we use it for SFT by finetuning on "chosen" responses. We use the maximum context length of our models to 512 tokens. Following standard practice, we wrap the SFT examples into a chat template with special tokens `<|user|>` and `<|assistant|>` added to the tokenizer before finetuning.

All results in the main paper are for Anthropic HH. The results for Tülu 3, which are qualitatively similar across experiments, are deferred to Appendix B.

Evaluation. To estimate the OOD score of our models, we use the LM Evaluation Harness framework (Gao et al., 2024). We estimate the 5-shot accuracy of models on 6 tasks: ARC-easy and ARC-challenge (Clark et al., 2018), PIQA (Bisk et al., 2020), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021), and SciQ (Welbl et al., 2017). All tasks are formatted as multiple-choice question benchmarks, so that we can use a base model to solve them by treating the option with the lowest (length-normalized) perplexity as the prediction. We use the default LM-Eval-Harness prompts for all tasks (wrapping them into the chat template for the finetuned models). In the figures in the main paper, we report the averaged accuracy over the tasks as OOD score.

Training. We train with AdamW (Loshchilov & Hutter, 2019), no weight decay, and batch size 256. We use a constant learning rate schedule following 20 steps of linear warmup. We chose to use the constant learning rate to eliminate the necessity to tune the total number of finetuning steps, allowing us to train until convergence. In the experiments in Section 3, we evaluate the validation loss on the SFT dataset every 500 steps and interrupt training after 3 evaluations without improvement or after reaching 30k steps. In Section 5, we train for 500 steps, which is already enough to elicit overtraining.

A.2 Sharpness estimation

To estimate model sharpness, we use the proxy introduced in Section 5.1. We sample a 4k subset of the Pile dataset (Gao et al., 2020) and calculate the average KL-divergence between the next-token predictions of the original model and the predictions of the noised model. We obtain the noised model by adding Gaussian noise $\varepsilon \sim \mathcal{N}(0, \sigma^2)$, $\sigma = 10^{-5}$ to the parameters. For each batch, we average the results over 10 random samples of ε.

B Additional Results

B.1 Verifying the Metric for Feature Drift with Gemma 3

In Section 3, we introduced the feature drift measure based on principal angles between subspaces. Here, we verify that this measure reflects the feature drift as measured by Sparse Autoencoders – a popular tool for analyzing latent features in LLMs (Bricken et al., 2023). To do this, we use Gemma Scope 2 (DeepMind, 2025), the open-source suite of Sparse Autoencoders available for various layers and sizes of the Gemma 3 model (Gemma Team et al., 2025).

To estimate feature drift with SAEs, we track the L2 distance between the features restored by the encoder from the activations of the base and finetuned models (after nonlinearity). We use the SAEs for 4 different layers and hidden dimensionality of 16k. While finetuning the model as discussed in Section 3, we track both SAE L2 norm and MPA jointly. In Figure 8, we observe that they almost perfectly correlate with each other. In Figure 9, we reproduce the implicit regularization effect of low LRs with the SAE-based metric.

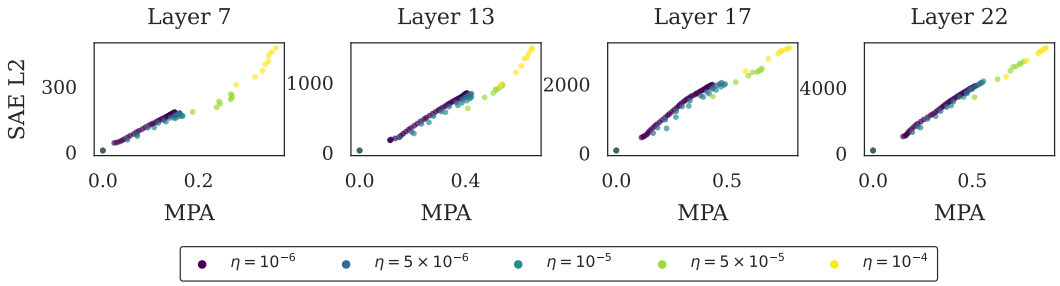


Figure 8: L2 on SAE representations vs MPA for Gemma 3 1B.

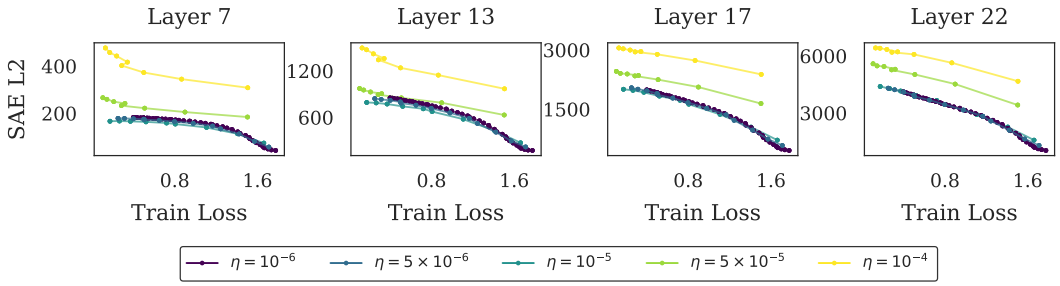


Figure 9: L2 on SAE representations vs SFT train loss for Gemma 3 1B.

B.2 Verifying the Metric for Sharpness with Pythia-14m.

In Section 5.1, we introduce a perturbation-based proxy for estimating sharpness of LLMs since computing the true Hessian eigenspectrum for billion-parameter models is computationally infeasible. However, one can compute true sharpness, defined as the top Hessian eigenvalue, for a small model. Here, we do it for the pretraining checkpoints of Pythia-14m (Biderman et al., 2023), and confirm that our estimate indeed reflects the true sharpness.

To compute true sharpness, we partially reuse the code of Cohen et al. (2025).

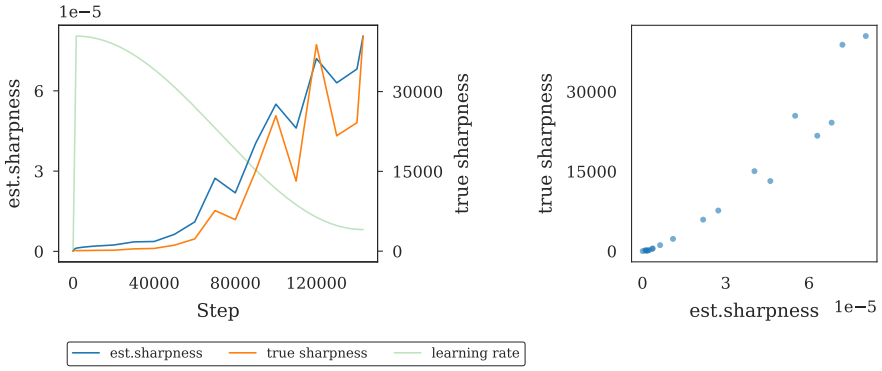


Figure 10: Left: dynamics of estimated sharpness and true sharpness closely follow each other through pretraining. Right: same data, true vs estimated sharpness.

B.3 Additional Plots for Section 3

B.3.1 Results for Other Models and Datasets

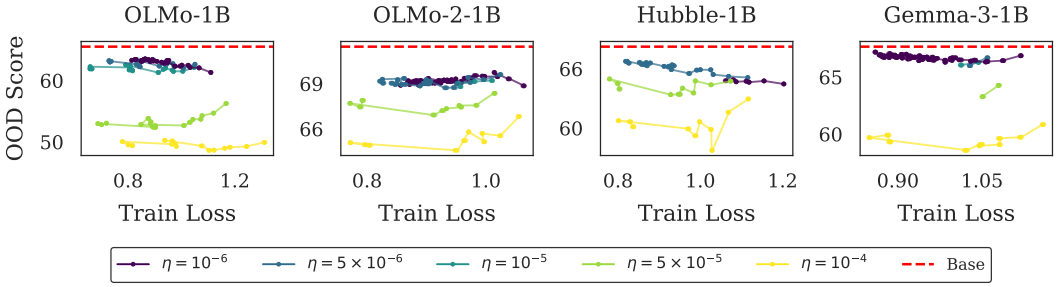
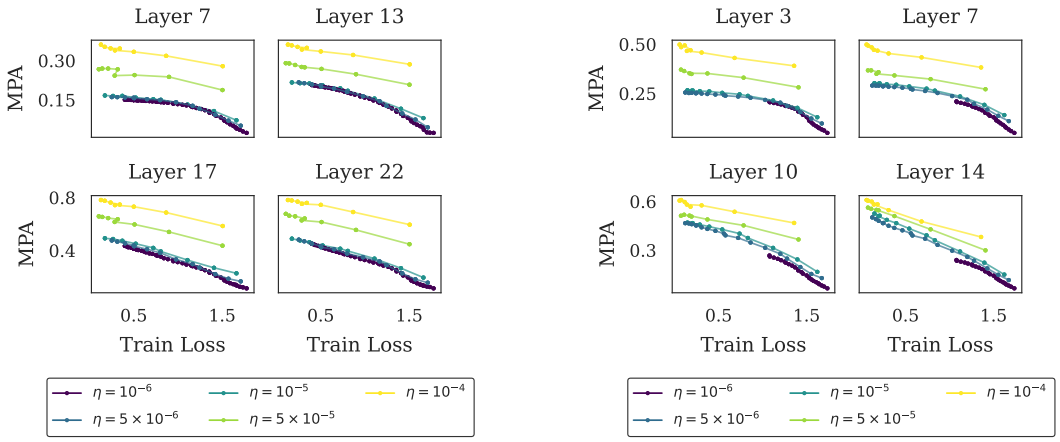
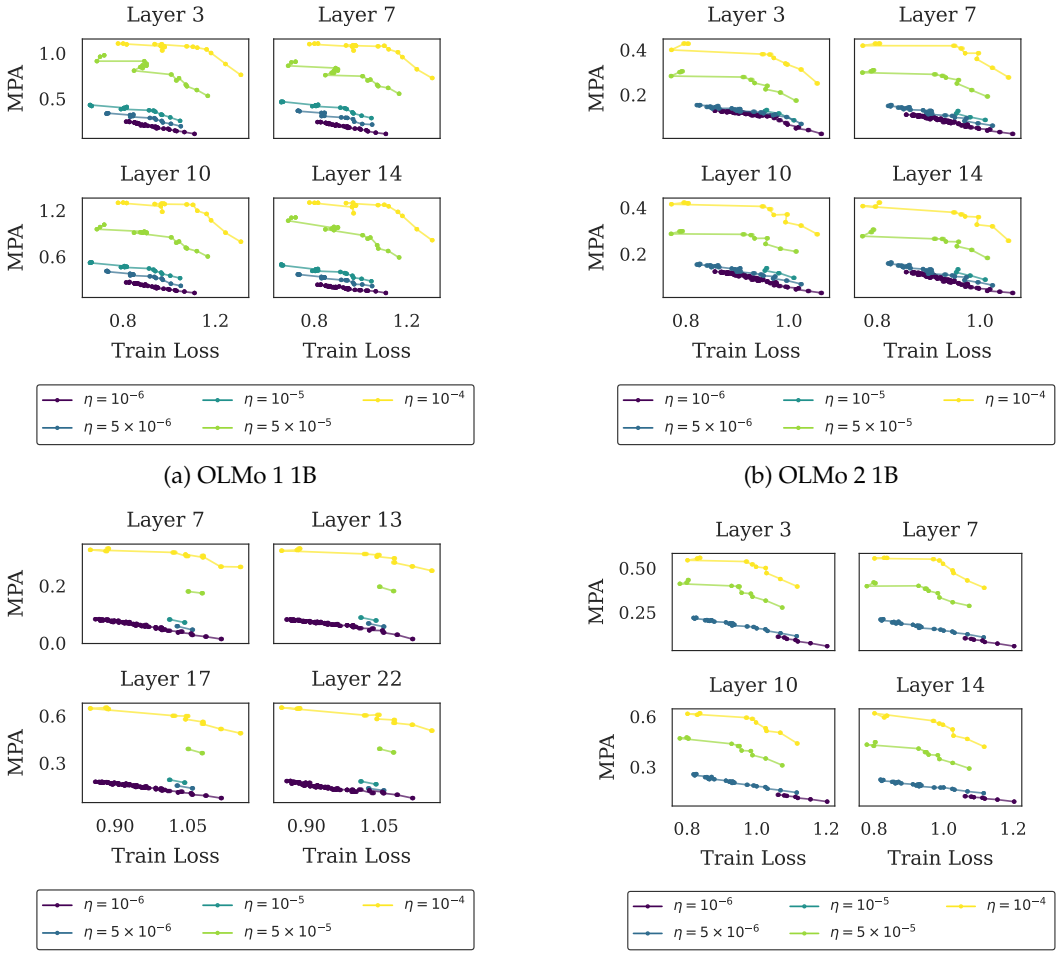


Figure 11: SFT train loss vs OOD score during supervised finetuning (same as Figure 1) for finetuning on the Tülu 3 dataset. The plots appear mostly horizontal, indicating that models go through a rapid forgetting phase in the beginning of finetuning and stay stable afterwards.



(a) Gemma 3 1B (b) Hubble 1B

Figure 12: MPA vs SFT train loss during supervised finetuning (same as Figure 2) for Gemma 3 and Hubble.



(a) OLMo 1 1B (b) OLMo 2 1B

(c) Gemma 3 1B (d) Hubble 1B

Figure 13: MPA vs SFT train loss during supervised finetuning (same as Figures 2 and 12) but for finetuning on the Tülu 3 dataset.

B.3.2 Results for Individual OOD Tasks

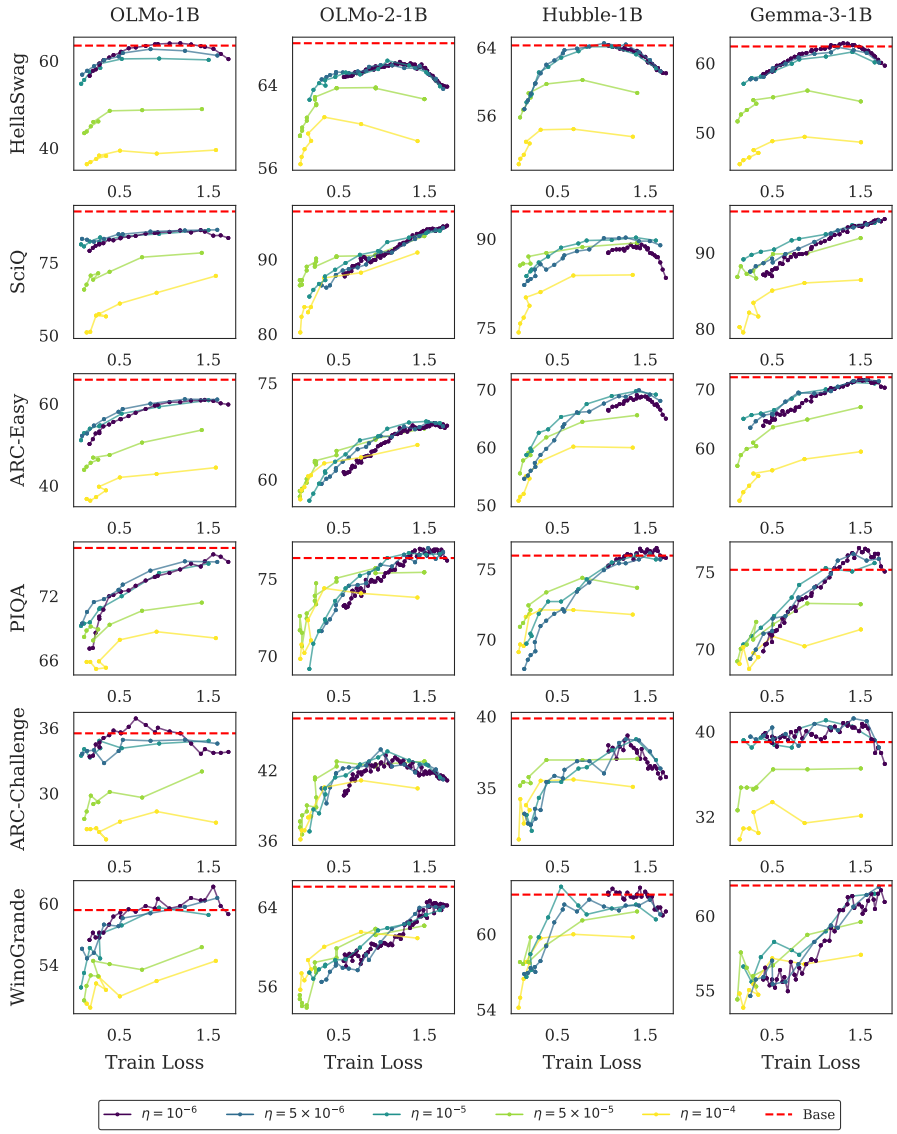


Figure 14: SFT training loss vs OOD performance for each OOD task separately. Same setup as Figure 1 in the paper, Anthropic HH dataset.

B.3.3 SFT validation loss

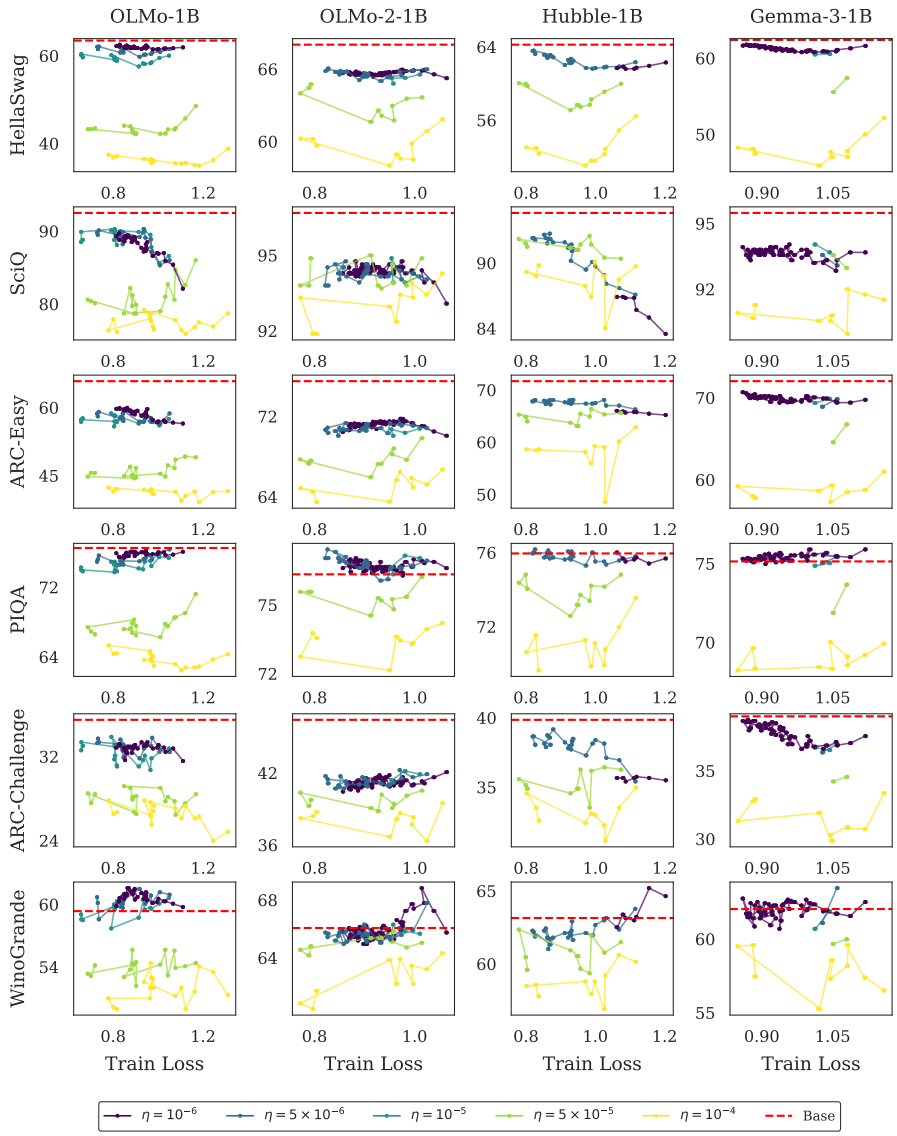


Figure 15: SFT training loss vs OOD performance for each OOD task separately. Same setup as Figure 1 in the paper, Tülu 3 dataset.

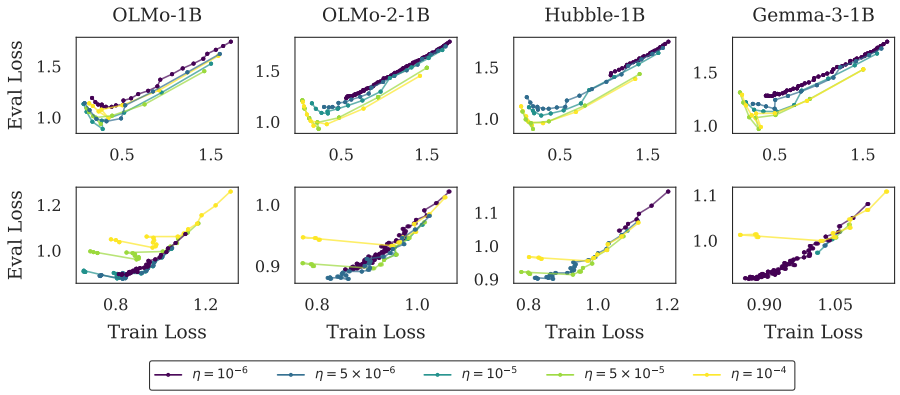


Figure 16: Validation loss on SFT datasets when finetuning with different learning rates (for the same setup as Figure 1 in the paper). Top row: finetuning on Anthropic HH, bottom row: finetuning on Tülu 3.

B.4 Additional Plots for Section 4

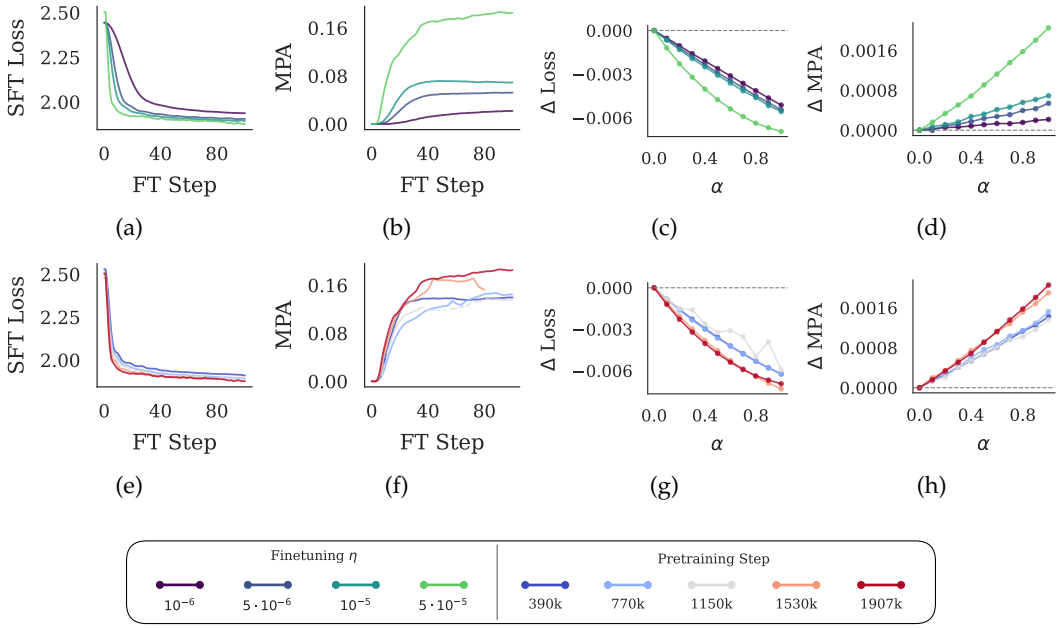


Figure 17: Loss landscape analysis (same as Figure 3) for OLMo 2 1B.

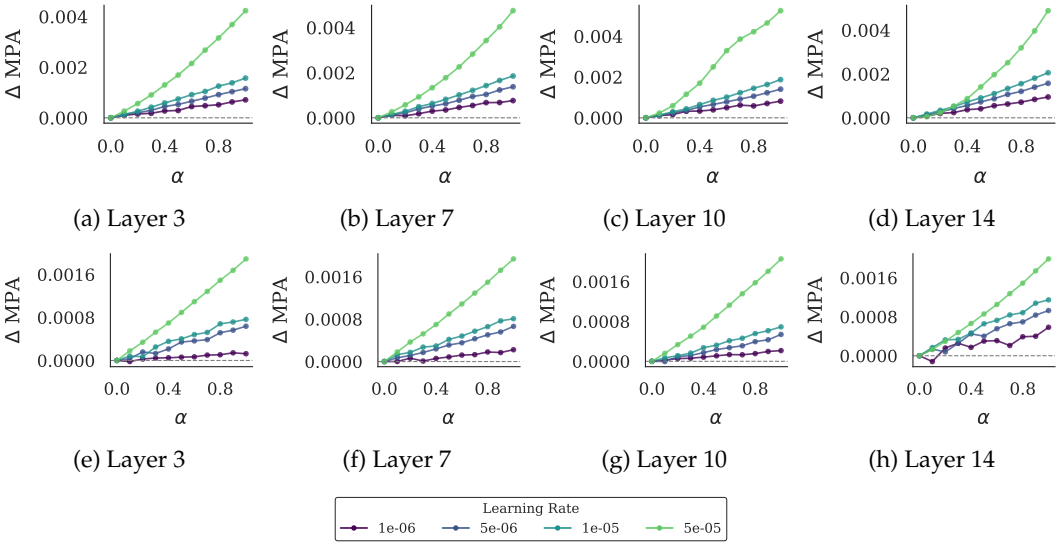


Figure 18: MPA as a function of stepsize with variable learning rates (analogous to Figures 3d, 17d) for different layers of OLMo 1 (top row) and OLMo 2 (bottom row).

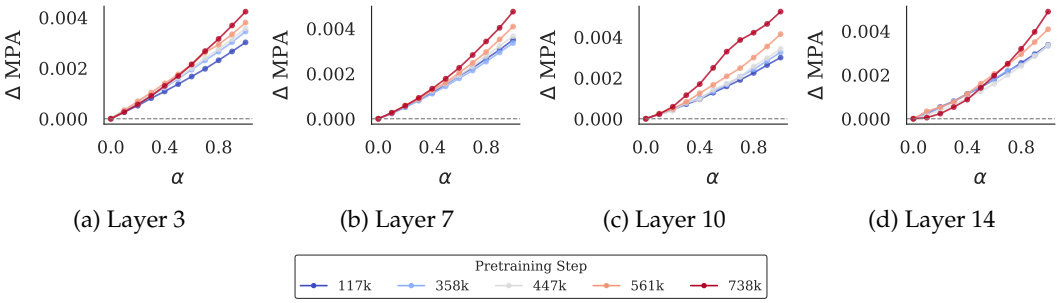


Figure 19: MPA as a function of stepsize with variable checkpoints (analogous to Figure 3h) for different layers of OLMo 1.

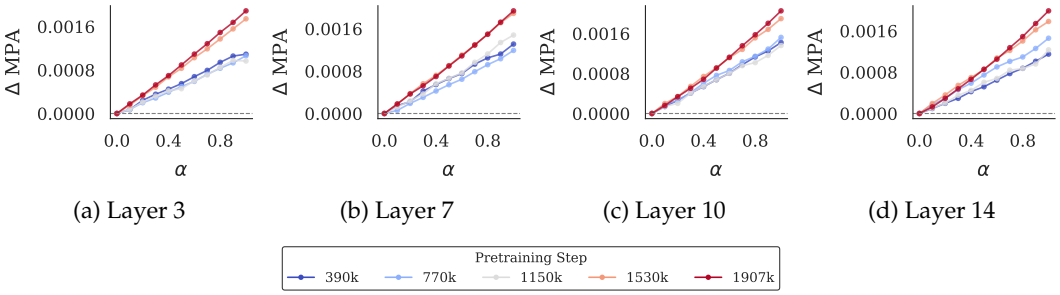


Figure 20: MPA as a function of stepsize with variable checkpoints (analogous to Figure 17h) for different layers of OLMo 2.

B.5 Additional Plots for Section 5

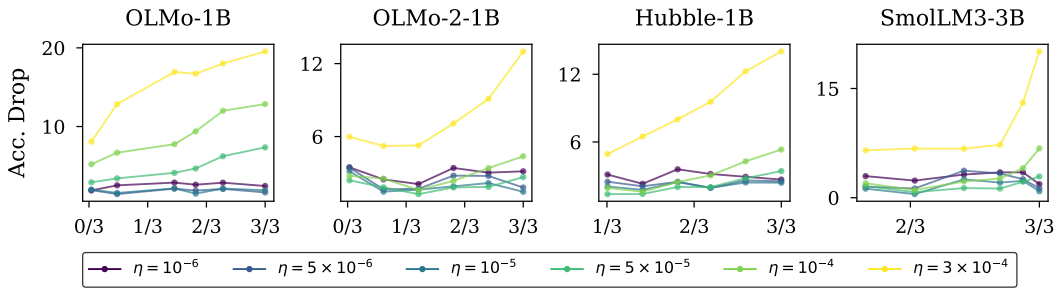
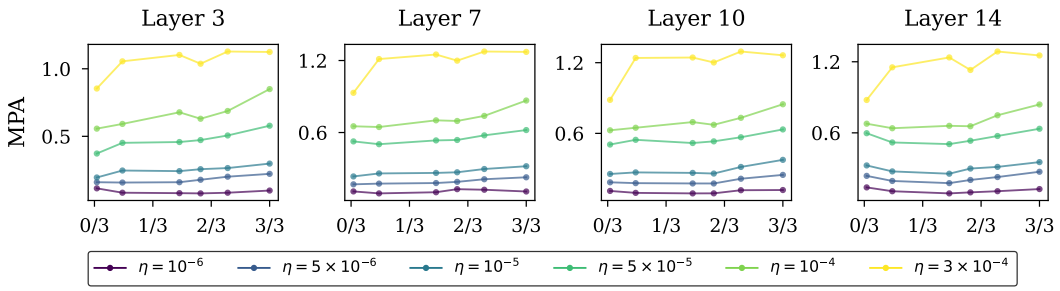
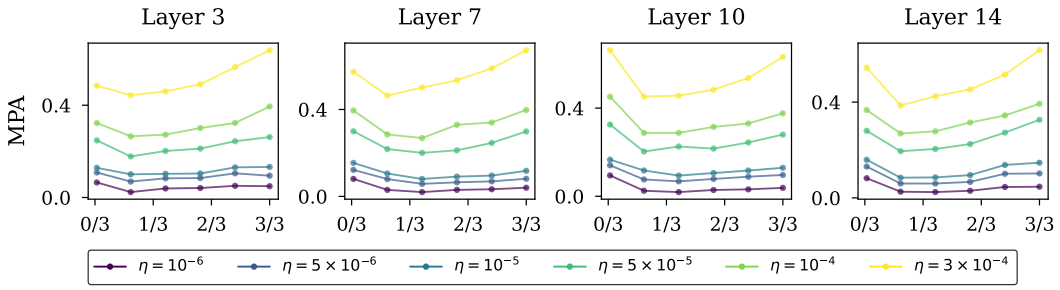


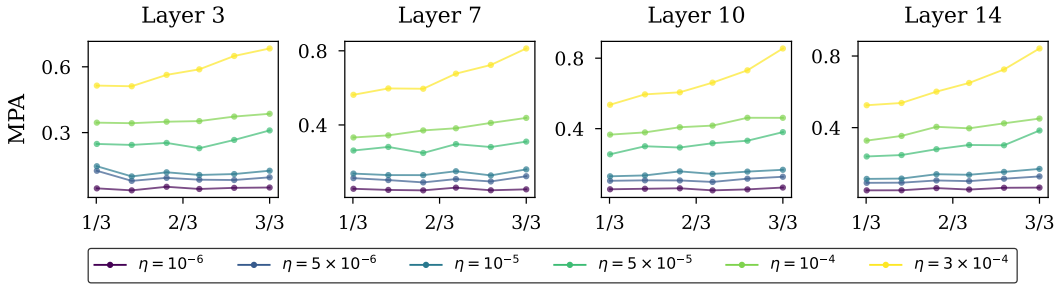
Figure 21: Forgetting (the drop in OOD score between the base and finetuned models) increases throughout pretraining. Same as Figure 6, but for the models finetuned on the Tülu 3 dataset.



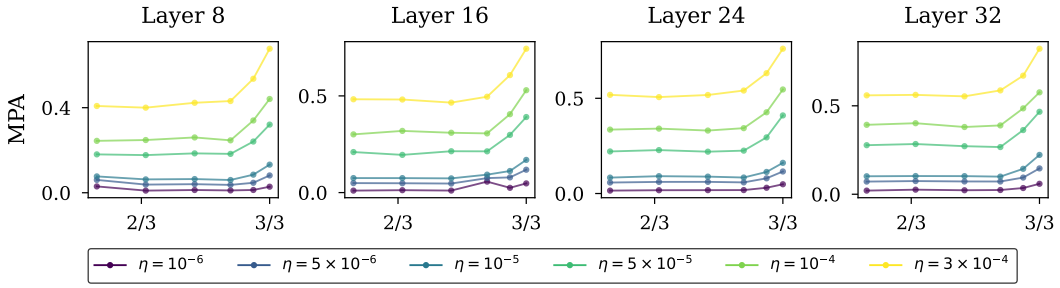
(a) OLMo 1 1B



(b) OLMo 2 1B

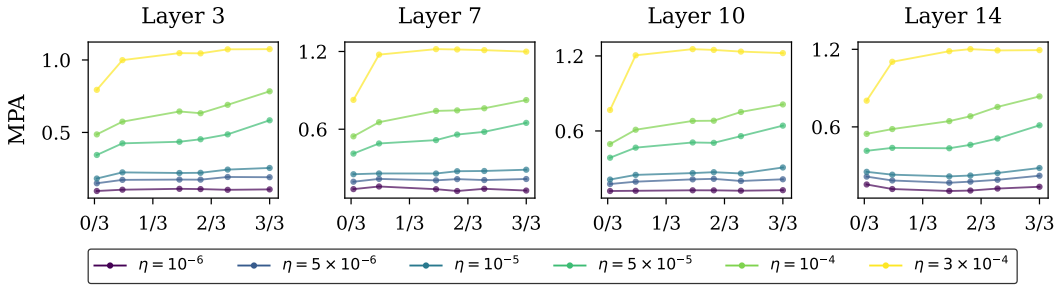


(c) Hubble 1B

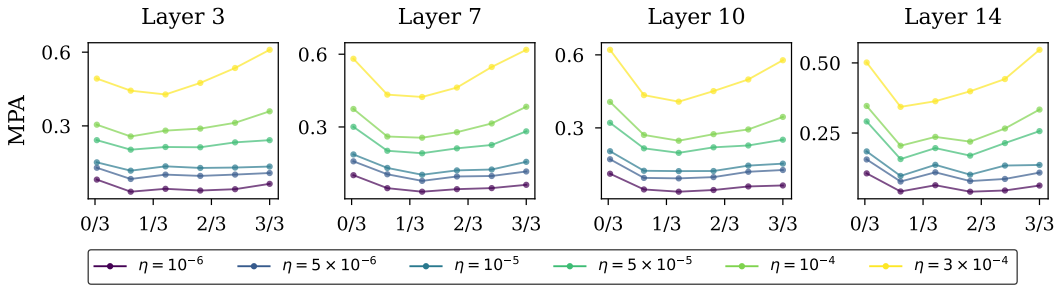


(d) SmolLM 3 3B

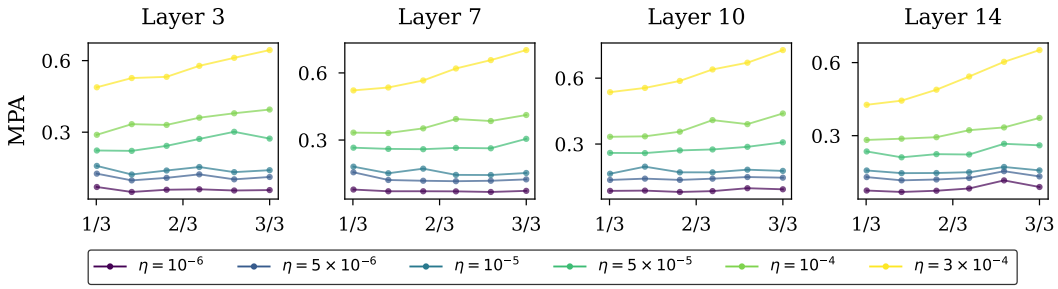
Figure 22: The principal angles between representations of base and finetuned models as a function of pretraining step, Same as Figure 7, but for all models.



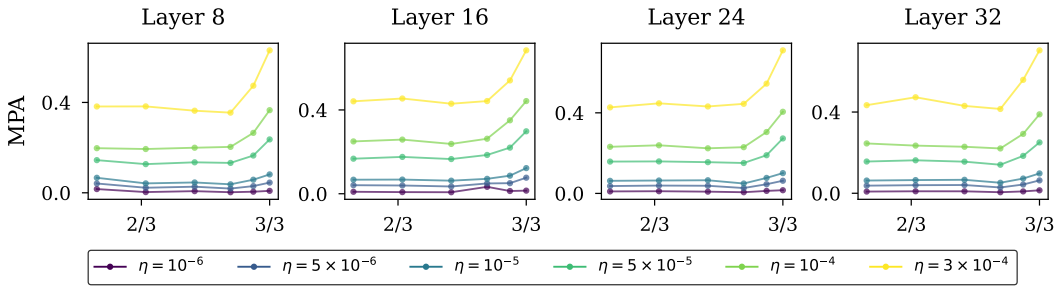
(a) OLMo 1 1B



(b) OLMo 2 1B



(c) Hubble 1B



(d) SmolLM 3 3B

Figure 23: Same as Figures 7 and 22, but for the models finetuned on Tülu 3.

B.6 Partial Pretraining Experiment

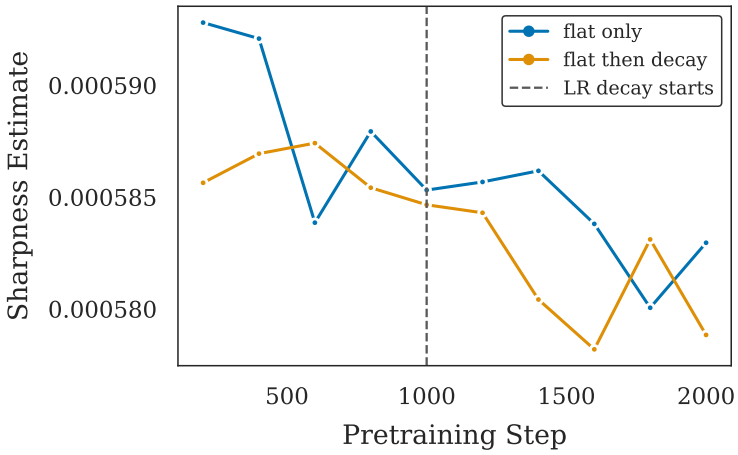


Figure 24: The dynamic of sharpness during a partial pretraining run of SmolLM 3 3B with and without LR decay at step 1000.

To the end of testing the causal relationship between LR decay and sharpness, we performed a *partial* pretraining run of SmolLM3 3B. We start from a checkpoint after 3800k steps of pretraining and continue training for 2k more steps with a global batch size of 2M tokens and a sequence length 4096. We use the DCLM dataset (Li et al., 2024b). During training, we track the KL-based sharpness estimate described in Section 5.1.

We compare two runs:

1. With a flat learning rate of 2e-4 (same as in the original SmolLM pretraining run),
2. With a flat learning rate of 2e-4 until step 1000 and linear LR decay to 0 after that.

The results are shown in Figure 24. We expected to see the effect similar to the one in Figure 5, where sharpness stays roughly constant while LR is constant, and rises when LR is decayed. However, that was not what we observed: sharpness curves for both runs are similar, in fact, sharpness even seems to slightly decrease with more steps, contradicting the results in Section 5.

We suspect two primary explanations for this behavior. First, the substitution of pretraining data from SmolLM mixture to DCLM might have caused an anomalous dynamic. Second, 2000 pretraining steps are probably not enough for reliably tracking sharpness, and the decrease is local. However, even 2000 steps of partial pretraining took more than 50 hours on 4 GPUs, and we chose not to continue with this experiment.

Statement on LLM Usage

LLMs were used for assistance with writing experimental code, search for related work, and minor textual edits. The authors take responsibility for all content in the paper.