LEARNING-FORGETTING OPTIMALITY IN SUPERVISED FINETUNING: A CLIFF PERSPECTIVE

Anonymous authors
Paper under double-blind review

ABSTRACT

Supervised finetuning (SFT) of pretrained language models trades off the acquisition of new domain capabilities against retention of prior knowledge. In this work, we extensively characterize this Pareto frontier of learning versus forgetting across nine instruction-tuned models from the Qwen2.5, Llama-3, and OLMo-2 families with full learning-rate sweeps for a number of classical and modern methods for continual learning and domain adaptation. Doing so, we find that across models, this frontier almost universally develops into a sharp *cliff*: a sharp phase transition where training recipes flip from learning without forgetting into catastrophic forgetting without improvements in learning. Across all methods we test, EWC, MAS, MIGU, NanoAdam, KL regularization, LoRA, and two geometry-motivated baselines we propose, we find that, while many methods can improve retention if the SFT baseline does not yet optimally retain, improvements in learnability remain elusive in practice for modern LLMs. This delineates two regimes. Either baseline SFT performance appears as a gradual trade-off between learning and forgetting, in which case many methods can be applied to "sharpen" the cliff and reduce forgetting, or, baseline SFT is already not forgetting, in which case no method we test can substantially intervene.

1 INTRODUCTION

Modern large language models are the result of a sequence of stages: pre-training, mid-training, supervised finetuning, and increasingly some form of reinforcement learning from human or verifiable feedback (Yang et al., 2025; Olmo et al., 2026; Apertus et al., 2025). Each stage targets a different capability, and each new release of an open-weight model effectively adds another post-training phase on top. In this regime, supervised finetuning (SFT) has become the main tool for injecting new behavior into a pre-trained model, whether the goal is domain specialization, instruction tuning, or to shape its safety profile.

This sequential nature of post-training naturally induces a continual learning setting, wherein each successive stage has to balance learning a new domain while retaining previously learned behaviors. The central problem that arises when continually learning in neural networks however, is catastrophic forgetting (McCloskey & Cohen, 1989), which remains a core challenge in modern post-training despite extensive work on the stability–plasticity trade-off. In practice, standard recipes rely on conservative updates, including small learning rates, KL regularization toward a reference policy, and short training horizons (Olmo et al., 2026; Apertus et al., 2025; Rofin et al., 2026; Lin et al., 2026). The dominant assumption is that retention can be controlled primarily through limiting update magnitude. As a result, although continual learning offers more structured approaches, these classical methods are rarely incorporated into contemporary post-training pipelines.

Why is this trade-off so hard to navigate in LLMs? Recent work points to the local geometry of the pre-trained loss landscape as a key factor. Watts et al. (2026) link flatness to forgetting under finetuning, and Catalan-Tatjer et al. (2026) show that robustness to weight perturbations governs both quantization and post-training degradation. From this perspective, a model is the joint product of its data, architecture, and training recipe, where the resulting weight-space geometry shapes how gracefully it can absorb a new optimization stage.

Motivated by this geometric perspective, we also contribute two learning methods that target the local sharpness of the retention loss directly. **Scale-Invariant Balancing (SIB)** uses the scale symmetries

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

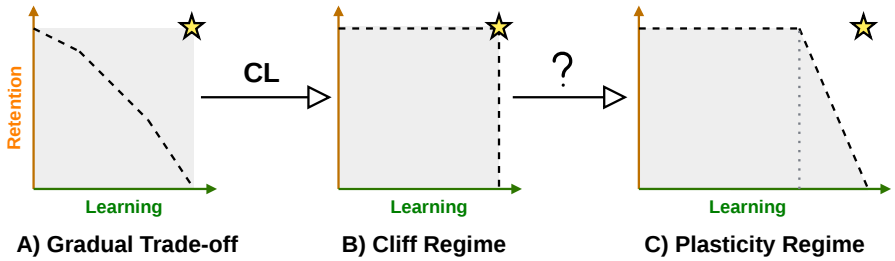


Figure 1: Learning-forgetting optimality. Learning tasks sequentially induces a learning-forgetting trade-off. We observe that, in supervised finetuning, there exist continual learning techniques that reshape gradual learning-forgetting trade-off to the cliff regime, that is, minimal forgetting with maximal learning. However, we have found that when plain SFT is already at the cliff, these technologies do not improve learning, providing little to no benefit over SFT.

of standard transformer architectures to reparameterize the model, once and before any SFT step, into a flatter region of the retention loss landscape. **Pre-Train Gradient Inverse (PTGI)** sets per-parameter learning-rate scales inversely proportional to retention sensitivity, slowing updates along the directions to which prior capabilities are most fragile.

In this paper we treat domain adaptation via SFT explicitly as a multi-objective problem and exhaustively characterize the resulting **learning-forgetting Pareto frontier** across nine open-weight models from three families and across a representative set of continual learning methods, regularizers, and training variations. Therefore, we can analyze how a given method reshapes the trade-off between domain adaptation and performance retention, by sweeping the learning rates.

What emerges is a clear picture, where there is a recurring pattern, the Pareto frontier appears to be determined largely by the model rather than by the method applied on top. In particular, we find two regimes. On one hand, there are settings for which there is a progressive trade-off modulated by the magnitude of the SFT learning rates, wherein a wide variety of interventions, including classical continual learning methods, KL regularization, smaller learning rates, and our geometry-based proposals, manage to Pareto-dominate plain SFT, approaching the optimal corner. On the other hand, well-tuned SFT already traces a cliff, with a sharp phase transition from full retention to catastrophic forgetting, and no method we investigate meaningfully improves on it.

Our contributions are as follows:

1. We map the learning-forgetting Pareto frontier across nine instruction-tuned open-weight models from three families (Qwen2.5, Llama-3, OLMo-2) under a unified SFT protocol with full learning-rate sweeps.
2. We benchmark a representative set of continual learning methods (EWC, MAS, MIGU, NanoAdam) alongside widely used interventions (LoRA, KL penalties, and epoch sweeps) alongside two geometry-motivated baselines we propose (SIB and PTGI), grounded in the sharpness perspective on forgetting.
3. Our central empirical finding is that the learning-forgetting frontier almost universally develops into a sharp *cliff*: a phase transition where training recipes flip from learning without forgetting to catastrophic forgetting without improvements in learning. This cliff appears to be a property of the underlying model rather than of any specific method.
4. We delineate two practical regimes. When baseline SFT traces a gradual trade-off, many methods can ”sharpen” the cliff and reduce forgetting, reframing the central question to *at what learning rate and hyperparameter setting can the model learn while not forgetting?* When baseline SFT already traces a cliff, no method we test substantially intervenes on either axis.
5. Across all methods evaluated, we find that improvements in *learnability* remain elusive: while retention can often be improved, no method substantially outperforms a well-tuned

SFT baseline in acquiring the new domain, highlighting plasticity in LLMs as an open problem for future work.

2 Related work

Sequential finetuning of a pre-trained model forces a choice between absorbing a new task and preserving what was learned before, a tension that has been studied in many forms since McCloskey & Cohen (1989) first articulated catastrophic forgetting. We focus on three threads of prior work that bear directly on this trade-off in modern LLMs: the structure of contemporary post-training pipelines, recent evidence that the severity of forgetting is governed by the local loss geometry around the pre-trained weights, and the continual learning methods that attempt to mitigate or redirect that loss, many of which can be read as implicit approximations of this geometry.

2.1 Post-training of language models

Modern LLMs are the result of a sequence of stages: pre-training, mid-training, instruction tuning, preference alignment and, increasingly, reasoning-oriented post-training, each targeting a different capability (Yang et al., 2025; Olmo et al., 2026; Apertus et al., 2025). SFT is the principal tool for injecting domain expertise, instruction following, and safety behaviors, and is typically followed by RL stages that further refine the model. As a consequence, post-training is inherently a sequential learning problem in which earlier capabilities should not be inadvertently overwritten. RL stages typically address forgetting through KL penalties between the updated and reference policies and very small learning rates (OLMo Team, 2025; Olmo et al., 2026; Apertus et al., 2025), treating the magnitude of post-training updates as a central lever for retention.

2.2 Forgetting in LLMs

Recent work is concerned with the relationship between pre-training and post-training, with growing evidence that the severity of forgetting in LLMs is tied to the local geometry of the loss of the pre-trained weights. Watts et al. (2026) establish a direct connection between the flatness of the loss landscape and catastrophic forgetting during finetuning, showing that sharpness-aware minimization (Foret et al., 2021) can mitigate it. Closely related, Catalan-Tatjer et al. (2026) show that post-training quantization robustness is governed by the same training dynamics suggesting that **robustness to weight perturbations** is a unifying lens for both quantization and post-training forgetting. Post-training quantization has been successfully exploiting weight-space geometry: Liu et al. (2025) show that, although the network is invariant to certain rotations of its weights, different parameterizations within these symmetries differ substantially in quantization error, an observation we build on to shape sensitivity to SFT updates. Empirically, forgetting has been observed under naive finetuning (Kalajdzievski, 2024), where the most consistent practical lever in current recipes remains conservative learning rates, which have been shown to limit feature drift (Rofin et al., 2026; Lin et al., 2026).

2.3 Continual learning

Continual learning offers a long-standing toolbox for the stability-plasticity trade-off that re-emerges in LLM post-training. **Replay-based** methods mix examples from earlier stages into the new training mixture, directly addressing forgetting when prior data is available. **Regularization-based** methods penalize movement away from the pre-trained weights, weighted by an estimate of parameter importance: EWC (Kirkpatrick et al., 2017) uses the diagonal Fisher information, MAS (Aljundi et al., 2018) uses the sensitivity of the model output, and KL penalties on logits effectively regularize functional distance from the base model. From a sharpness perspective, these methods can be seen as proxies for the curvature of the retention loss around the pre-trained weights, slowing updates along directions to which the retained capabilities are most sensitive. *Gradient-projection* methods restrict updates to subspaces orthogonal to those used by previous tasks, while more recent rehearsal-free, task-label-free approaches such as MIGU (Du et al., 2024) mask updates by the magnitude of layer outputs, and FGGM (Tan et al., 2026) replaces this heuristic with a Fisher-based parameter-importance estimate. In LLM-specific settings, parameter-efficient methods such as LoRA (Hu et al., 2021; Biderman et al., 2024) have been proposed as a way to bound the magnitude of weight changes. Our methods are most directly related to the data-aware regularization line: rather than penalizing

motion of individual weights, we use the gradient of the retention loss to (i) reparameterize the model within its scale symmetries to flatten the retention landscape and (ii) allocate per-parameter learning-rate scales inversely proportional to retention sensitivity, both of which can be viewed as data-aware, geometry-based mechanisms for guiding SFT updates.

3 Evaluating the Pareto frontier

In this section, we describe the experimental setup used to study the learning-forgetting Pareto frontier that emerges during supervised fine-tuning (SFT). We conduct our experiments on **ChemL3** (Feng et al., 2025) because existing open-weight language models perform poorly on it prior to fine-tuning, and even relatively small models such as Qwen2.5-0.5B can achieve strong performance after SFT with optimal hyperparameters. This makes ChemL3 a representative benchmark for studying sequential domain-specific fine-tuning.

3.1 Supervised finetuning

Backbones. Generally, open-weights models provide a base model and an instruction-tuned version. We choose instruction-tuned models for two principal reasons. Firstly, these models are trained for more post-training stages, with more domain-specific capabilities already acquired, making them relevant for studying forgetting under sequential learning. Secondly, open-weight models are highly optimized for performance on math benchmarks (Gulati et al., 2025), as a consequence, the forgetting signal will be less noisy. We work with Qwen2.5-Instruct at 0.5B, 1.5B, 3B and 7B (Qwen et al., 2025), Llama-3.2-Instruct at 1B and 3B, Llama-3.1-Instruct at 8B (Grattafiori et al., 2024), OLMo-2-Instruct at 1B and 7B (OLMo Team, 2025). We additionally replicate our main findings on the Qwen3 family at 0.6B, 1.7B, 4B and 8B (Yang et al., 2025), with thinking disabled for both finetuning and evaluation, and compare base and instruction-tuned Qwen2.5-1.5B and -3B; instruction-tuned models turn out to be the more brittle setting (Appendix J).

Optimization process. Following common practice (Taori et al., 2023), we implement a standard instruction tuning training pipeline, including prompt loss masking, where the loss is computed only on the response tokens and not on the prompt and standard instruction message formats for each model. All finetuning runs use AdamW (Loshchilov & Hutter, 2019) with weight decay 0.01. We use a cosine learning-rate schedule (Loshchilov & Hutter, 2017) that warms up for the first 20 steps and then decays the peak learning rate to $0.1\times$ its maximum value, with an additional 50 cool-down steps at the end of training. To map out the retention-learning Pareto frontier per model, we exhaustively sweep the peak learning rate and the number of epochs for all models and methods. Unless indicated otherwise, we default to 3 epochs of SFT.

Learning. As described, we choose **ChemL3** as the domain-specific task for the backbones to learn. This dataset is comprised of the chemistry level-3 split of Feng et al. (2025), a multiple-choice question answer dataset containing relatively niche chemistry questions, that is generally unknown to all of the backbones that we test, and will serve as our learning measure. ChemL3 has four answer options, so a model that follows the answer format but guesses at random scores 0.25, and a model that does not follow the format scores below that. Since base accuracy differs across models, we always compare against each model's own base accuracy (the star in each figure). To test whether our findings are specific to ChemL3, we additionally finetune on two further tasks: LogiQA (Liu et al., 2023), a multiple-choice logical reasoning dataset, and Astro2026, a free-form question answering dataset that we built from 500 astrophysics discoveries reported on Wikipedia during 2026, i.e. after the knowledge cut-off of all models we study, so that base models score near zero on it (Table 2). Astro2026 answers are graded by an LLM judge that has access to the reference answer (Appendix B).

Forgetting. To measure retention of already learned tasks we use GSM8K (Cobbe et al., 2021) to measure the performance drop characterizing forgetting. As mathematical reasoning is learned through substantial effort in pretraining for the open weight models we consider, it represents a sufficiently complex proxy of task retention. Given that GSM8k is also evaluated through verification of model generations, it is further preferable over task retention measurements solely through multiple choice benchmarks, which do not measure whether generative capabilities have been retained. In all of our experiments and baseline models, we observe that GSM8K performance is uncorrelated to ChemL3. We further measure retention on IFEval (Zhou et al., 2023), for instruction following,

	Qwen2.5				OLMo2	Llama-3.2		Llama-3.1
	0.5B	1.5B	3B	7B	1B	1B	3B	8B
Base	0.42	0.45	0.44	0.50	0.39	0.41	0.44	0.49
SFT	0.69	0.67	0.80	0.91	0.90	0.93	0.97	0.98
Best method	0.93	0.85	0.95	0.99	0.96	0.97	1.00	1.00
	EWC	EWC	EWC	EWC/SIB	PTGI	PTGI	EWC	EWC

Table 1: Cliffness (Equation (1)) of the base model, SFT, and the best method on ChemL3 (learning) and GSM8K (retention), in the setting of Figure 3. A method cliffs if its cliffness is at least 0.9.

and on a random 10% subset of MMLU (Hendrycks et al., 2021), for knowledge, and report each retention benchmark separately to capture worst-case forgetting. GSM8K yields the lowest or tied-lowest cliffness (Section 3.2) in 7 of 8 models (Table 2), which makes it a conservative proxy for forgetting. We include concrete examples and further details for all datasets in Appendix A.

3.2 Quantifying Cliffs

Each method f traces a learning–retention trajectory on model m as we sweep its learning rate, and we call the *Pareto frontier* of m the envelope of these trajectories over all methods. To quantify how closely a trajectory approaches the optimal corner of maximal learning and full retention, we define

$$\text{cliffness}(f, m) = \max_{\eta \in \mathcal{H}} \min\left(\frac{\ell_{f,\eta}}{\ell_m}, \frac{r_{f,\eta}}{r_m}\right), \qquad \ell_m = \max_{f',\eta'} \ell_{f',\eta'}, \tag{1}$$

where $\mathcal{H}$ is the set of swept learning rates, $\ell_{f,\eta}$ and $r_{f,\eta}$ are the learning (ChemL3) and retention (GSM8K) accuracies of method f at learning rate η, ℓ_m is the best learning accuracy reached by any method on model m, and r_m is the retention accuracy of the base model before SFT. Geometrically, cliffness is the side length of the largest square anchored at the origin that is dominated by some point of the normalized trajectory. We say that f *cliffs* on m if $\text{cliffness}(f, m) \geq 0.9$. Table 1 reports cliffness in the setting of Figure 3; a high cliffness of the base model itself indicates that finetuning barely improves on the learning task.

4 Methods Aimed at Improving Retention in Supervised Finetuning

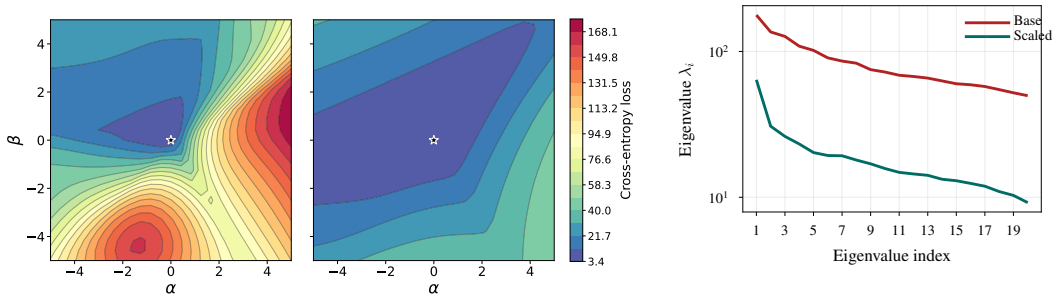
We briefly summarize the continual learning strategies we will later evaluate in our experimental setup. We compare established conventional strategies such as Elastic Weight Consolidation, and only introduce two simple baselines that minimize changes to the gradient of the model before finetuning as comparison point.

4.1 Classical Methods for Continual Learning

We compare against representative continual learning methods that modulate SFT updates (details in Appendix E). **EWC** (Kirkpatrick et al., 2017) and **MAS** (Aljundi et al., 2018) add a quadratic penalty that anchors each parameter to its pre-trained value, weighted by an importance estimate computed on a retain dataset $\mathcal{D}_R$: the diagonal of the empirical Fisher information and the sensitivity of the model output, respectively. Both can be read as proxies for the curvature of the retention loss around the pre-trained weights. **MIGU** (Du et al., 2024) and **NanoAdam** (Zhou et al., 2025) instead reshape the update itself, masking the gradients of parameters with small output activations or updating only the smallest-magnitude weights, and need no retain data.

4.2 Simple Gradient-Based Strategies for Retention

There has been recent work indicating that the sharpness of the model is related to performance degradation to finetuning (Watts et al., 2026; Rofin et al., 2026), or post-training quantization (Catalan-Tatjer et al., 2026), indicating that a model with flatter pre-train loss may be more robust to post-training weight changes. With this motivation, we propose two different methods that



(a) Loss landscapes **(b)** Top 20 Hessian eigenvalues

Figure 2: Landscape of the loss and hessian eigenvalues. In Figure 2a we show a slice of the same size of the landscape of the loss of the base model (left) and that of the reparameterization following the scale-symmetries. Figure 2a demonstrates the effect of stretching that the scaled parametrization has, where the sensitive sharp directions are flattened. In Figure 2b we visualize the top 20 eigenvalues of the hessian of the loss, which demonstrates that the reparameterization has a flatter loss.

minimize the norm of the gradient of the model on the pre-train loss. First, we define a retain dataset w.r.t. which we will minimize its loss' gradient norm: $\mathcal{D}_R$

Weight-space symmetries. Weight-space symmetries have been used for more successful model merging (Ainsworth et al., 2023), accelerated optimization (Zhao et al., 2023), and reduced quantization degradation (Liu et al., 2025). We focus on the specific sub-set of scale-symmetries: given a symmetric pair of tensors $(W_{\text{div}}, W_{\text{mul}})$ within the scale symmetry, we rescale by α to minimize

$$\|\alpha\, g_{\text{div}}\|^2 + \|g_{\text{mul}}/\alpha\|^2, \qquad g_{\text{div}} = \nabla_{W_{\text{div}}}\mathcal{L}(\mathcal{D}_R),\ g_{\text{mul}} = \nabla_{W_{\text{mul}}}\mathcal{L}(\mathcal{D}_R), \tag{2}$$

i.e. the squared gradient norms of the rescaled pair $(W_{\text{div}}/\alpha,\ \alpha W_{\text{mul}})$, which admits the closed-form minimizer

$$\alpha^* = \left(\|g_{\text{mul}}\|^2/\|g_{\text{div}}\|^2\right)^{1/4}, \qquad \alpha^* \in [1/R,\ R], \tag{3}$$

clamped to R (we use R=100 unless stated otherwise). The solution at the "balance" between $\|g_{\text{mul}}\|^2$ and $\|g_{\text{div}}\|^2$ gives name to this method, which we refer to as *scale invariant balancing* (SIB).

Conceptually, this stretches the directions of higher weight sensitivity to $\mathcal{L}(\mathcal{D}_R)$ and compresses the more robust weights, thereby protecting $\mathcal{L}(\mathcal{D}_R)$ from SFT weight updates. Figure 2 shows the loss landscape of a model (left) and its reparameterization following 2, it can be seen that the top eigenvalues of the hessian of $\mathcal{L}(\mathcal{D}_R)$ are much lower, indicating that the reparameterization has effectively flattened the landscape. We generally limit the symmetry pairs that we optimize for to the value/output projection pairs (V/O, per head) and the SwiGLU MLP pairs (W_{up} and W_{gate} rows with W_{down} columns, per neuron), because they are common to all the architectures that we study, and each tensor within the symmetry has a similar amount of elements (as opposed to RMSNorm and its consumer tensors), requiring no normalization. See Appendix F for the scale symmetries pairs in the backbones that we are using, and Algorithm 1 for the step-by-step algorithm of SFT pipeline with SIB.

Per-weight LR scales. The gradient $\nabla_W\mathcal{L}(\mathcal{D}_R)$ indicates the sensitivity of $\mathcal{L}(\mathcal{D}_R)$ to perturbations in W locally. We propose allocating SFT weight updates inversely proportional to $\nabla_W\mathcal{L}(\mathcal{D}_R)$. Concretely, we estimate each parameter's p retain-gradient magnitude once before SFT by averaging across K=50 retain micro-batches

$$g_p^R \;=\; \mathbb{E}_{B\sim\mathcal{D}_R}\big[\,|\nabla_p\,\mathcal{L}(B)|\,\big], \qquad s_p \;=\; \frac{1}{\max\!\big(|g_p^R|,\,\varepsilon\big)}, \tag{4}$$

and use s_p as a post-Adam multiplier on every optimizer step during SFT. Moreover, we recenter the scales tensor-wise so that the per-tensor mean of $\log s_p$ is zero, which attempts to preserve the overall step magnitude within each tensor (so that the global LR schedule retains its meaning) and keeps the subsequent clamp $s_p \in [1/R,\ R]$ symmetric around 1 in log-space via $s_p \;\leftarrow\; s_p \Big/ \exp\!\Big(\frac{1}{|p|}\sum_{p'}\log s_{p'}\Big)$ with a default value of R=100. We refer to this method as pre-train gradient inverse (PTGI), which is described step-by-step in Algorithm 2.

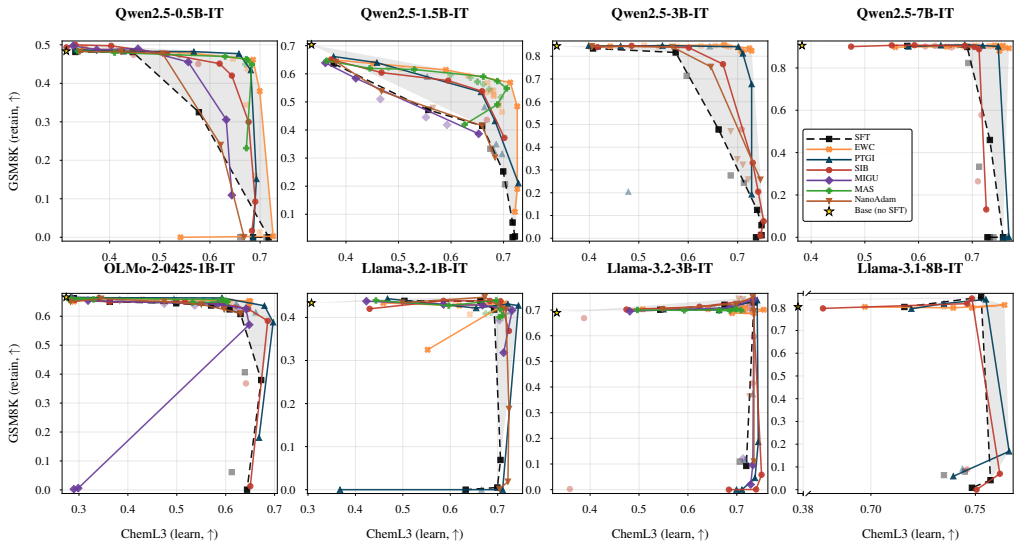


Figure 3: Pareto optimality of continual learning methods. In the settings where there is gradual balance between learning and forgetting (e.g., Qwen2.5 and OLMo2 settings), continual learning methods dominate the Pareto frontier, but notably only move it upward, 'sharpening' the cliff, i.e. the sudden phase transition from complete retention to complete forgetting. When SFT already forms a cliff (e.g., Llama3.2 models), CL methods barely perform better than SFT, indicating their role as performance retention rather than learning enhancement.

5 Experiments

As introduced in Section 3, sequential learning is inherently multi-objective optimization, where the quality of a solution is determined by how it balances different objectives, a balance that is regulated by the magnitude of SFT updates. We begin this section by demonstrating that classical continual learning techniques find Pareto-optimal solutions, that is those with minimal forgetting and maximal learning. We follow this result by exploring how general techniques such as Low-Rank Adapters (LoRA) (Hu et al., 2021; Biderman et al., 2024), Kullback-Leibler (KL) penalty (Kullback & Leibler, 1951), model scale and training for more epochs at low learning rates shape the Pareto frontier. Find complete results, implementation details and hyper-parameter tuning at H. Appendices C and D show that the same pattern holds on two further learning tasks and three retention benchmarks, and that it is not an artifact of output-format drift.

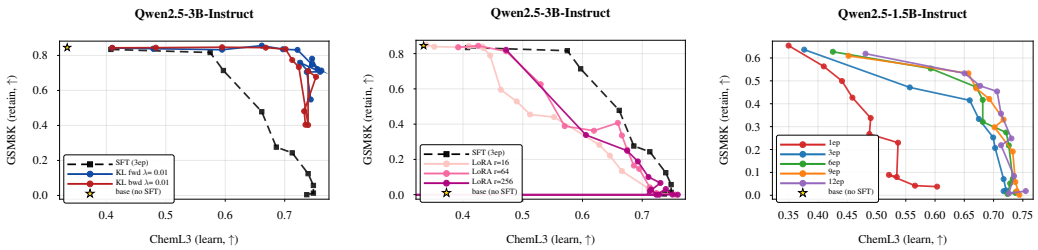
5.1 Effect of continual learning methods on the Pareto frontier.

Figure 3 exposes the different learning-forgetting regimes that emerge during SFT, and how EWC, MIGU, MAS, and NanoAdam (Kirkpatrick et al., 2017; Du et al., 2024; Aljundi et al., 2018; Zhou et al., 2025) modulate this trade-off. Starting off analyzing vanilla SFT, which is the black dashed line, we see a very interesting phenomenon. For the settings (Qwen2.5-0.5B, -1.5B, -3B, -7B and OLMo2-1B) where learning and forgetting are gradually traded-off, continual learning methods approximate the optimal corner at which there is minimal forgetting and maximal learning. Surprisingly, in there is another scenario (Llama3.2-1B, Llama3.2-3B, Llama3.1-8B) where SFT already approximates the optimal corner. In this case, CL methods fail to offer better learning, indicating that these modulate forgetting rather than learning.

This phenomenon is surprisingly robust across models, finetuning strategies and almost all other types of interventions to the task setup we investigate. Owing to the appearance of these charts we dub this phase transition a cliff:

Definition: "Cliffs" in Supervised Finetuning

We use the term *cliff* to refer to the regime of the learning–forgetting Pareto frontier in which increasing the SFT learning rate (or epoch count) sharply degrades retention. Even optimal SFT



(a) KL regularization **(b)** LoRA **(c)** Training epochs

Figure 4: KL regularization, LoRA and training epochs. (a) Similarly to classical continual learning methods, KL regularization mitigates forgetting but does not improve learning beyond the cliff. (b) Full finetuning Pareto-dominates LoRA, which gives a worse learning-forgetting trade-off for most learning rates. (c) Increasing training epochs creates cliffs: once learning saturates, larger learning rates mainly increase forgetting with little gain in learning. More models in Figures 11, 13 and 16.

strategies all observe a sharp phase transition between learning without forgetting and learning while almost completely forgetting the original task, without a smooth trade-off between learning and forgetting. Section 3.2 makes this quantitative.

The appearance of a cliff, i.e. the existence of hyperparameter setting and learning rate where even a modification, i.e. a slight increase in learning rate will completely shift the model into a mode of of forgetting the original task (McCloskey & Cohen, 1989), is surprisingly extreme for large language models. It is further especially noticeable in Figure 3 that the exact method used, whether expensive classical CL method, such as EWC, or a cheaper substitute based on weight symmetries is of limited importance for most models, and based on the data we observe over 8 models we find the following pattern: *if standard finetuning shows a gradual trade-off between learning and forgetting, then most continual learning methods can reduce forgetting, but not improve learning. If standard finetuning already shows a sharp phase transition, i.e. a cliff, then no method we test can substantially improve.*

Model size and family. Across families, larger models generally show a steeper trade-off, with SFT closer to the optimal corner, and Qwen2.5 models show the most gradual trade-off (Appendix H.2).

5.2 Do LoRA, KL regularization, and more epochs form a cliff?

Reinforcement learning is commonly regularized with a KL penalty between the base and updated policy logits to encourage solutions that remain close to the base model and thereby mitigate forgetting (Apertus et al., 2025; OLMo Team, 2025; Olmo et al., 2026). While not conventionally considered a continual learning technique, the method clearly compares. Similarly, LoRA has been observed to induce reduced learning and forgetting (Biderman et al., 2024). Finally, motivated by common observation that small learning rates limit forgetting, the simplest conventional mitigation of learning instabilities is simply to module the number of training epochs to affect the learning-forgetting balance.

In Figure 4a we observe that, similarly to continual learning methods, KL regularization mitigates forgetting but does not aid in learning beyond the cliff. Moreover, in Figure 4b we implement that full finetuning Pareto-dominates LoRA, where it learns less and forgets more, contrary to the results reported in (Biderman et al., 2024). Finally, we observe in Figure 4c that increasing the number of training epochs promotes learning in small learning rates, which maintain GSM8K performance. However, a recurrent pattern arises, when performance has saturated for a model, training for more epochs does not incur in better learning.

5.3 Why do cliffs emerge?

SIB and PTGI are informative probes of finetuning dynamics: both optimize the unmodified SFT loss and only redirect updates away from parameters with a large retention gradient. That this redirection improves retention (Tables 1 and 3) suggests that forgetting is driven not by the SFT objective itself, but by updates along a small set of retention-sensitive directions. This offers an intuition for

the cliff: it emerges whenever updates are steered away from these directions, whether by explicit regularization (KL, EWC, MAS), by routing (SIB, PTGI), or, we hypothesize, by scale, where overparameterization leaves more directions in which to learn the new task without interference. This is a hypothesis consistent with our interventions, not an established mechanism.

6 LIMITATIONS

While our results are consistent across the evaluated setting, it remains important to validate them on datasets spanning a broader range of token budgets and task difficulties to assess the robustness and generality of the observed behavior. In addition, our analysis is restricted to autoregressive transformers with quadratic attention. It is therefore not guaranteed whether the same dynamics extend to other model classes, including mixture-of-experts (MoE) architectures, larger-scale models beyond 8B parameters, sub-quadratic attention mechanisms, or diffusion-based language models.

7 CONCLUSION

We study domain adaptation through supervised fine-tuning (SFT) across a range of open-weight language models, exhaustively cataloging the Pareto trade-off between improving domain performance and preserving pre-training capabilities for each investigated model and method, ranging from classical continual learning methods, to baselines and first-principles approaches.

Our main finding is that this frontier between learning and forgetting appears to depend more on the underlying model than on any concrete method, when sweeping learning rates over the full frontier. We find two main cases exist: In the first, standard supervised finetuning traces an apparently gradual trade-off between learning and forgetting. In this scenario, most continual learning methods, training variations, and even approaches such as KL regularization all manage to reduce forgetting, but none improve learnability, ultimately forming the sharp *cliff* visible in each of our figures, where finetuning recipes transition from complete retention to catastrophic forgetting. In the second case, SFT, if tuned exhaustively, already traces a cliff, with a sharp phase transition into forgetting. In this second case, no method we investigate is able to significantly improve either forgetting or learning.

Overall, these findings point to two practical takeaways. First, if appropriately tuned continual-learning techniques in the broad sense, including smaller learning rates, KL penalties, and explicit regularization, are employed, then forgetting can be almost entirely avoided for all large language models we investigate. The question of a trade-off between learning and forgetting is reframed to the question of *to what learning rate and hyperparameter setting can the model learn while not forgetting?* Second, across all methods we find a surprisingly limited evidence for a successful increase in plasticity in LLMs, with no method substantially outperforming the baseline sweep of supervised finetuning in learning the new domain. Future work may show whether methods that truly increase learnability can be developed.

AI USE STATEMENT

We used generative AI tools for code assistance for implementing standard methods, data processing and filtering, and visualizing results; and, as part of our evaluation protocol, grading free-form Astro2026 answers with an LLM judge that has access to the reference answer (Appendix B). Occasionally, for editing the manuscript which the authors reviewed and edited. The problem framing, experimental design, and interpretation of results are the authors' own. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Samuel K. Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries, 2023. URL https://arxiv.org/abs/2209.04836.

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget, 2018. URL https://arxiv.org/abs/1711.09601.

Project Apertus et al. Apertus: Democratizing open and compliant llms for global language environments, 2025. URL https://arxiv.org/abs/2509.14233.

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. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=aloEru2qCG. Featured Certification.

Albert Catalan-Tatjer, Niccolò Ajroldi, and Jonas Geiping. Training dynamics impact post-training quantization robustness, 2026. URL https://arxiv.org/abs/2510.06213.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Wenyu Du, Shuang Cheng, Tongxu Luo, Zihan Qiu, Zeyu Huang, Ka Chun Cheung, Reynold Cheng, and Jie Fu. Unlocking continual learning abilities in language models, 2024. URL https://arxiv.org/abs/2406.17245.

Kehua Feng, Xinyi Shen, Weijie Wang, Xiang Zhuang, Yuqi Tang, Qiang Zhang, and Keyan Ding. Sciknoweval: Evaluating multi-level scientific knowledge of large language models, 2025. URL https://arxiv.org/abs/2406.09098.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization, 2021. URL https://arxiv.org/abs/2010.01412.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Aryan Gulati, Brando Miranda, Eric Chen, Emily Xia, Kai Fronsdal, Bruno Dumont, Elyas Obbad, and Sanmi Koyejo. Putnam-axiom: A functional and static benchmark for measuring higher level mathematical reasoning in llms, 2025. URL https://arxiv.org/abs/2508.08292.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://arxiv.org/abs/2009.03300.

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. *arXiv preprint arXiv:2106.09685*, 2021.

Damjan Kalajdzievski. Scaling laws for forgetting when fine-tuning large language models, 2024. URL https://arxiv.org/abs/2401.05605.

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.

Solomon Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951. URL https://api.semanticscholar.org/CorpusID:120349231.

Jiacheng Lin, Zhongruo Wang, Kun Qian, Tian Wang, Arvind Srinivasan, Hansi Zeng, Ruochen Jiao, Xie Zhou, Jiri Gesi, Dakuo Wang, Yufan Guo, Kai Zhong, Weiqi Zhang, Sujay Sanghavi, Changyou Chen, Hyokun Yun, and Lihong Li. Sft doesn't always hurt general capabilities: Revisiting domain-specific fine-tuning in llms, 2026. URL https://arxiv.org/abs/2509.20758.

Hanmeng Liu, Jian Liu, Leyang Cui, Zhiyang Teng, Nan Duan, Ming Zhou, and Yue Zhang. LogiQA 2.0—an improved dataset for logical reasoning in natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2947–2962, 2023. doi: 10.1109/TASLP.2023.3293046.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. Spinquant: Llm quantization with learned rotations, 2025. URL https://arxiv.org/abs/2405.16406.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts, 2017. URL https://arxiv.org/abs/1608.03983.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, Benjamin Bossan, and Marian Tietz. PEFT: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft, 2022.

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

Team Olmo et al. Olmo 3, 2026. URL https://arxiv.org/abs/2512.13961.

OLMo Team. 2 OLMo 2 furious. 2025. URL https://arxiv.org/abs/2501.00656.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Mark Rofin, Aditya Varre, and Nicolas Flammarion. (how) learning rates regulate catastrophic overtraining, 2026. URL https://arxiv.org/abs/2604.13627.

Noam Shazeer. Glu variants improve transformer, 2020. URL https://arxiv.org/abs/2002.05202.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning, 2026. URL https://arxiv.org/abs/2601.19897.

Chao-Hong Tan, Qian Chen, Wen Wang, Yukun Ma, Chong Zhang, Chong Deng, Qinglin Zhang, Xiangang Li, and Jieping Ye. Fggm: Fisher-guided gradient masking for continual learning, 2026. URL https://arxiv.org/abs/2601.18261.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Ishaan Watts, Catherine Li, Sachin Goyal, Jacob Mitchell Springer, and Aditi Raghunathan. Sharpness-aware pretraining mitigates catastrophic forgetting, 2026. URL https://arxiv.org/abs/2605.02105.

An Yang et al. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Bo Zhao, Nima Dehmamy, Robin Walters, and Rose Yu. Symmetry teleportation for accelerated optimization, 2023. URL https://arxiv.org/abs/2205.10637.

Chao Zhou, Tom Jacobs, Advait Gadhikar, and Rebekka Burkholz. Pay attention to small weights. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2506.21374.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models, 2023. URL https://arxiv.org/abs/2311.07911.

A CHEMISTRY LEVEL-3 EXAMPLES

Example 1: ESOL Aqueous Solubility Prediction (MoleculeNet)

Prompt: *Given a question and four options, please select the correct answer. Your answer should be "A", "B", "C" or "D". Please directly provide the answer without explanation.*

Question: What is the correct logarithmic solubility value of the molecule `OC(c1ccccc1)C(O)c1ccccc1` in aqueous solutions?

A) -5.19

B) -1.93

C) -3.76

D) -0.55

Answer: B **Subtask:** `ESOL_value_prediction`

Example 2: Molar Weight Prediction (PubChem)

Prompt: *Given a question and four options, please select the correct answer. Your answer should be "A", "B", "C" or "D". Please directly provide the answer without explanation.*

Question: What is the molar weight (g/mol) of the molecule with the canonical SMILES representation

`C1CC(N(C1)S(=O)(=O)C2=CC=C(C=C2)Cl)C(=O)OCC(=O)NC3=CC=C(C=C3)C#N`?

A) 492.600

B) 490.500

C) 493.600

D) 447.900

Answer: D **Subtask:** `s2w` (SMILES-to-weight)

Figure 5: Example multiple-choice samples from the chemistry evaluation datasets used in our experiments.

B ADDITIONAL LEARNING TASKS

LogiQA. We finetune on LogiQA (Liu et al., 2023), a multiple-choice logical reasoning dataset (https://github.com/csitfun/LogiQA2.0). An example:

> *Context:* One seminar had 18 participants. It is known that: (1) At least 5 young teachers are female; (2) At least 6 female teachers are over middle age; (3) At least seven young women are teachers.
>
> *Question:* According to the above information, which of the following can be concluded about the participants?

A) Some young teachers are not women.

B) Some young women are not teachers.

C) There are at least 11 young teachers.

D) There are at least 13 female teachers.

Answer: D

Astro2026. We built Astro2026 from 500 astrophysics discoveries reported on Wikipedia during 2026, after the knowledge cut-off of all models we study. Answers are free-form and graded by an LLM judge that has access to the reference answer. An example:

> *Question:* When did NASA launch the Artemis II lunar flyby mission?
>
> *Answer:* NASA launched the Artemis II lunar flyby mission on 1 April 2026.

	Qwen2.5				OLMo2	Llama-3.2		Llama-3.1
	0.5B	1.5B	3B	7B	1B	1B	3B	8B
Learning: LogiQA, retention: GSM8K								
Base	0.72	0.75	0.80	0.79	0.93	0.82	0.68	0.74
SFT	0.86	0.86	0.98	0.88	1.00	1.00	0.95	0.92
Best method	0.93	0.94	0.99	0.92	0.99	0.99	1.00	0.96
	KL	SIB	KL	NanoAdam	SIB	SIB	NanoAdam/LoRA	EWC/NanoAdam
Learning: Astro2026, retention: GSM8K								
Base	0.01	0.04	0.04	0.09	0.03	0.03	0.02	0.03
SFT	0.34	0.33	0.97	1.00	0.55	0.89	0.87	0.91
Best method	0.94	0.89	0.97	1.00	0.94	0.92	0.93	0.91
	EWC	EWC	SFT	SFT	PTGI	SIB	PTGI	SFT
Learning: ChemL3, SFT; retention benchmark varies								
GSM8K	0.69	0.67	0.80	0.91	0.90	0.93	0.97	0.98
IFEval	0.80	0.77	0.93	0.96	0.87	0.94	0.97	0.98
MMLU	0.80	0.77	0.93	0.97	0.91	0.94	0.97	0.98

Table 2: Cliffness across learning tasks and retention benchmarks. Top: cliffness on LogiQA and Astro2026 with GSM8K retention; on Astro2026 we ran SFT, EWC, SIB and PTGI only. Bottom: SFT cliffness on ChemL3 measured against each retention benchmark separately.

ChemL3 with reasoning traces. For the format-drift experiment of Appendix D we use the ChemL3 variant of Shenfeld et al. (2026), which augments each question with a GPT-4o reasoning trace.

C Do cliffs generalize across learning tasks and retention benchmarks?

The top of Table 2 repeats the analysis of Table 1 for two further learning tasks, keeping GSM8K as the retention benchmark. The two-regime pattern replicates: either SFT already cliffs, or the best continual learning method approaches the optimal corner without learning substantially more than SFT; on both tasks the best method reaches a cliffness of at least 0.89 on every model. On LogiQA, base models already have high cliffness, i.e. no method learns substantially beyond the base model, which again points to plasticity rather than forgetting as the binding constraint. The bottom of Table 2 replaces GSM8K by IFEval and MMLU for SFT on ChemL3: the same cliff appears for all three capabilities, at capability-dependent learning rates, and GSM8K is the capability that degrades fastest.

D Is the cliff an artifact of output-format drift?

ChemL3 answers are single letters, whereas GSM8K requires free-form generation, so part of the collapse could reflect drift in output format rather than loss of mathematical capability. Three experiments suggest that the cliff is not a format artifact. First, on Astro2026, where the model generates free-form answers, SFT or a continual learning method retains at least 89% of the GSM8K accuracy while reaching at least 89% of the best learning accuracy on every model (Table 2). Second, SFT of Qwen2.5-7B on ChemL3 augmented with GPT-4o reasoning traces (Shenfeld et al., 2026) still draws a cliff, with a cliffness of 0.94. Third, we probe elicitation directly. Qwen2.5-7B consistently begins its GSM8K answers with "To", so we prefill the assistant turn with "To". At the first learning rate past the cliff, prefilling recovers GSM8K accuracy from 10% to 86%, but the cliff does not disappear: ChemL3 accuracy does not improve further, and GSM8K accuracy drops to 0% at the next learning rate. Prefilling moves cliffness only from 0.92 to 0.96.

E CLASSICAL CONTINUAL LEARNING METHODS

We compare against representative continual-learning methods that modulate SFT updates with respect to a retain dataset $\mathcal{D}_R$. These fall into two groups: methods that add a regularizer anchoring weights to their pre-trained values, and methods that reshape the update itself.

Importance-weighted regularization. A first family augments the SFT loss with a quadratic penalty that anchors each parameter to its pre-trained value $\theta^\star_p$, weighted by a per-parameter importance estimate Ω_p computed once on $\mathcal{D}_R$:

$$\mathcal{L}(\theta) \;=\; \mathcal{L}_{\mathrm{SFT}}(\theta) \;+\; \tfrac{\lambda}{2}\sum_p \Omega_p\,(\theta_p - \theta^\star_p)^2. \tag{5}$$

EWC (Kirkpatrick et al., 2017) sets Ω_p to the diagonal of the empirical Fisher information, $\Omega_p = \mathbb{E}_{B\sim\mathcal{D}_R}\big[(\nabla_p \mathcal{L}(B))^2\big]$, while **MAS** (Aljundi et al., 2018) replaces it with the sensitivity of the model output, $\Omega_p = \mathbb{E}_{B\sim\mathcal{D}_R}\big[\,|\nabla_p \|f_\theta(B)\|^2|\,\big]$, which can be estimated without labels. Both can be read as second-order proxies for the curvature of the retention loss around $\theta^\star$, slowing motion along directions to which the retained capabilities are most fragile, and connecting naturally to the sharpness perspective discussed in Section 2.2.

Update-shaping methods. A second family does not modify the loss but reshapes the SFT update itself. **MIGU** (Du et al., 2024) is a data-free approach that, at each step, computes the magnitude of the forward-pass activations and masks the gradient on parameters whose corresponding output activations fall below a per-layer percentile. **NanoAdam** (Zhou et al., 2025) similarly keeps the standard Adam step but applies it only to the smallest-magnitude weights at each iteration, freezing large-magnitude weights on the assumption that they encode features acquired during pre-training. Unlike EWC and MAS, neither method makes use of $\mathcal{D}_R$, but modify the training using finetuning-time signals only, which makes them attractive when retain data is unavailable.

F WEIGHT-SPACE SYMMETRIES

Proposition F.1. *Let $\hat{W} := \alpha W$ and consider Adam updates applied to $\hat{W}$ with learning rate η. Then, up to the additive stabilizer ϵ, the induced dynamics on W are equivalent to Adam updates with learning rate η/α applied directly to W.*

Proof. The Adam update on the scaled parameters is given by

$$\hat{W}^{t+1} = \hat{W}^t - \eta \frac{\hat{m}^t}{\sqrt{\hat{v}^t} + \epsilon}, \tag{6}$$

where $\hat{m}^t$ and $\hat{v}^t$ denote the bias-corrected first and second moment estimates of

$$\hat{g}^t := \nabla_{\hat{W}} \mathcal{L}(\hat{W}^t).$$

Since $\hat{W} = \alpha W$ and $\mathcal{L}(\hat{W}) = \mathcal{L}(\alpha W)$, the chain rule yields

$$\hat{g}^t = \frac{1}{\alpha} g^t, \qquad g^t := \nabla_W \mathcal{L}(W^t).$$

By linearity of the Adam moment updates, this scaling propagates to the first and second moments:

$$\hat{m}^t = \frac{1}{\alpha} m^t, \qquad \hat{v}^t = \frac{1}{\alpha^2} v^t.$$

Substituting into the normalized update term gives

$$\frac{\hat{m}^t}{\sqrt{\hat{v}^t} + \epsilon} = \frac{(1/\alpha) m^t}{(1/\alpha)\sqrt{v^t} + \epsilon} \tag{7}$$

$$= \frac{m^t}{\sqrt{v^t} + \alpha\epsilon}. \tag{8}$$

Algorithm 1 SFT with Scale-Invariant Balancing (SIB), V/O + MLP symmetry pairs

Require: Pre-trained weights $\theta^\star$; fine-tuning data $\mathcal{D}_{\mathrm{FT}}$; retain (PT) data $\mathcal{D}_{\mathrm{PT}}$; the set of scale-invariant pairs we optimize across all architectures, $\mathcal{P} = \mathcal{P}_{\mathrm{V/O}} \cup \mathcal{P}_{\mathrm{MLP}}$, with

- $\mathcal{P}_{\mathrm{V/O}} = \{(W_V^{(\ell)}, W_O^{(\ell)})\}_{\ell=1}^{L}$ (per-layer value $\to$ output projection pair);
- $\mathcal{P}_{\mathrm{MLP}} = \{(W_{\mathrm{up}}^{(\ell)}, W_{\mathrm{down}}^{(\ell)}), (W_{\mathrm{gate}}^{(\ell)}, W_{\mathrm{down}}^{(\ell)})\}_{\ell=1}^{L}$ (gate/up rows paired with the down-projection's columns);

clamp ratio R; FT optimizer $\mathcal{O}$ (AdamW), base LR η, epochs E.

$\theta \leftarrow \theta^\star$

Phase 1 — Geometry teleportation (one-shot, gradient pass on PT data)

for $(W_{\mathrm{div}}, W_{\mathrm{mul}}) \in \mathcal{P}$ **do**

  Compute gradient norms $g_{\mathrm{div}} = \left\|\nabla_{W_{\mathrm{div}}} \mathcal{L}_{\mathrm{PT}}(\theta)\right\|$, $g_{\mathrm{mul}} = \left\|\nabla_{W_{\mathrm{mul}}} \mathcal{L}_{\mathrm{PT}}(\theta)\right\|$ from a single PT minibatch

  $\alpha \leftarrow \mathrm{clip}\Big(\sqrt{g_{\mathrm{mul}}/g_{\mathrm{div}}},\ 1/R,\ R\Big)$ ▷ equalises gradient magnitudes across the pair

  $W_{\mathrm{div}} \leftarrow W_{\mathrm{div}}/\alpha, \quad W_{\mathrm{mul}} \leftarrow W_{\mathrm{mul}} \cdot \alpha$ ▷ output-preserving reparametrisation

end for

Phase 2 — Standard SFT from the rebalanced weights

for epoch $= 1 \dots E$ **do**

  for minibatch $b \sim \mathcal{D}_{\mathrm{FT}}$ **do**

    $\theta \leftarrow \mathcal{O}(\theta,\ \nabla_\theta \mathcal{L}_{\mathrm{FT}}(\theta; b),\ \eta)$

  end for

end for

return θ

Hence,

$$\hat{W}^{t+1} = \hat{W}^t - \eta \frac{m^t}{\sqrt{v^t} + \alpha\epsilon}. \tag{9}$$

Dividing by α and defining $W^t := \hat{W}^t/\alpha$ yields

$$W^{t+1} = W^t - \frac{\eta}{\alpha} \frac{m^t}{\sqrt{v^t} + \alpha\epsilon}. \tag{10}$$

In regimes where $\alpha\epsilon \ll \sqrt{v^t}$, the stabilizer term can be neglected, yielding an effective learning-rate rescaling by $1/\alpha$. □

F.1 Algorithm

F.2 Weight-space symmetries

For each of the following weight pairs, there exists a positive vector α such that replacing $W \mapsto W/\alpha$ on one side and $W \mapsto W\alpha$ on the other leaves the network's input–output map exactly unchanged:

- **V/O**: V rows $\leftrightarrow$ O columns (per head).
- **SwiGLU (Shazeer, 2020)**: $(W_{\mathrm{up}}, W_{\mathrm{gate}})$ rows $\leftrightarrow$ W_{down} columns (per neuron).
- **RMSNorm / linear**: γ entries $\leftrightarrow$ columns of downstream linear layers sharing that norm.
- **Q/K** (with RoPE): per-kv-head scale, constrained to be shared within each kv-head to preserve RoPE.

We limit the symmetries to V/O pair and the SwiGLU MLPs.

Algorithm 2 SFT with PT-Gradient Inverse LR (PTGI)

Require: Pre-trained weights $\theta^\star$; fine-tuning data $\mathcal{D}_{\mathrm{FT}}$; retain (PT) data $\mathcal{D}_{\mathrm{PT}}$; retain budget B (number of PT minibatches, default 50); clamp ratio R; FT optimizer $\mathcal{O}$, base LR η, epochs E; numerical floor ε.

$\theta \leftarrow \theta^\star$

Phase 1 — Per-parameter LR-scale estimation on PT data

Initialise $|g|_i \leftarrow 0$ for every trainable coordinate i

for $b = 1 \dots B$ **do**

  $g \leftarrow \nabla_\theta \mathcal{L}_{\mathrm{PT}}(\theta;\, \mathrm{batch}_b)$

  $|g|_i \leftarrow |g|_i + |g_i|/B$ for all i ▷ accumulator; equiv. $\mathbb{E}_b[\,|g_i|\,]$

end for

$s_i \leftarrow 1 \,/\, (|g|_i + \varepsilon)$ for all i ▷ important-for-PT params get small s_i

for each parameter tensor T **do**

  $s_i \leftarrow s_i \,/\, \mathrm{geomean}_{i \in T}(s_i)$ for $i \in T$ ▷ per-tensor centring; preserves η on average

  $s_i \leftarrow \mathrm{clip}(s_i,\, 1/R,\, R)$ for $i \in T$

end for

Phase 2 — SFT with the per-parameter LR multipliers

for epoch $= 1 \dots E$ **do**

  for minibatch $b \sim \mathcal{D}_{\mathrm{FT}}$ **do**

    $g^{\mathrm{FT}} \leftarrow \nabla_\theta \mathcal{L}_{\mathrm{FT}}(\theta; b)$

    $\theta_i \leftarrow \mathcal{O}(\theta_i,\, g_i^{\mathrm{FT}},\, \eta \cdot s_i)$ for all i ▷ coordinate-wise effective LR $\eta \cdot s_i$

  end for

end for

return θ

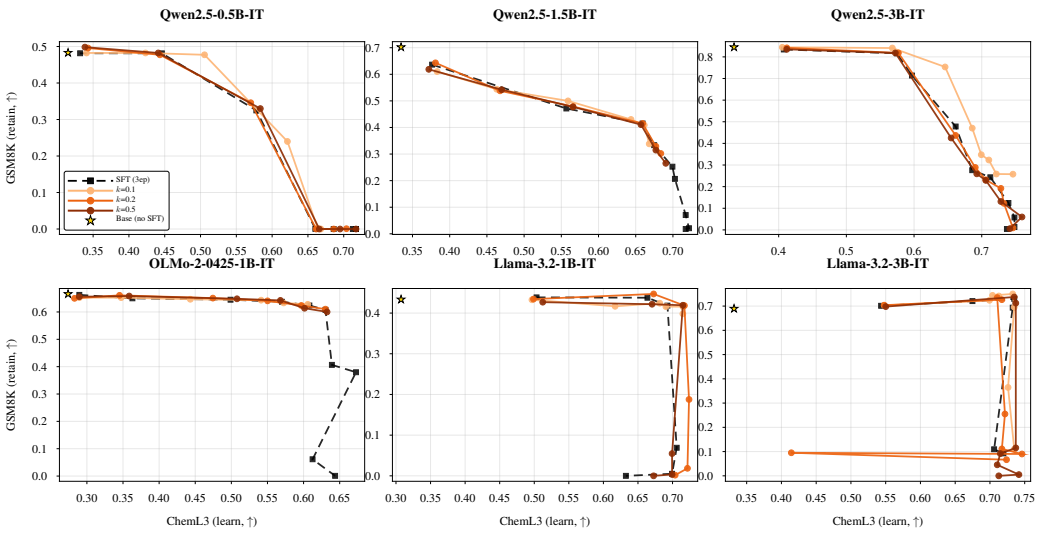


Figure 6: NanoAdam hyperparameter sweeps (Zhou et al., 2025).

G PTGI ALGORITHM

H MORE CLIFFS

H.1 CONTINUAL LEARNING METHODS

We generally tuned the hyper-parameters on Qwen2.5-0.5B, which is the cheapest model. We not always sweep the learning rates for each hyperparameter value, as it is exorbitantly expensive. For every method and model we sweep up to 16 learning rates; Figures 6 to 9, 12 and 14 show the resulting hyperparameter sweeps.

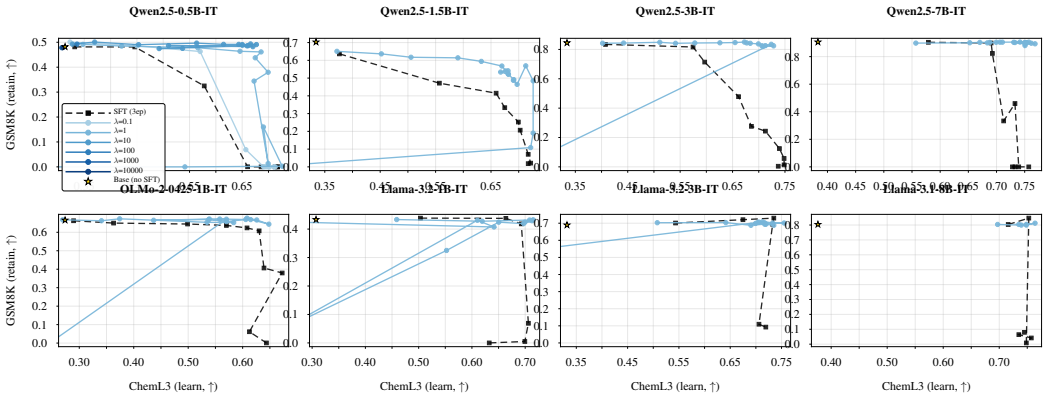


Figure 7: EWC hyperparameter sweeps (Kirkpatrick et al., 2017).

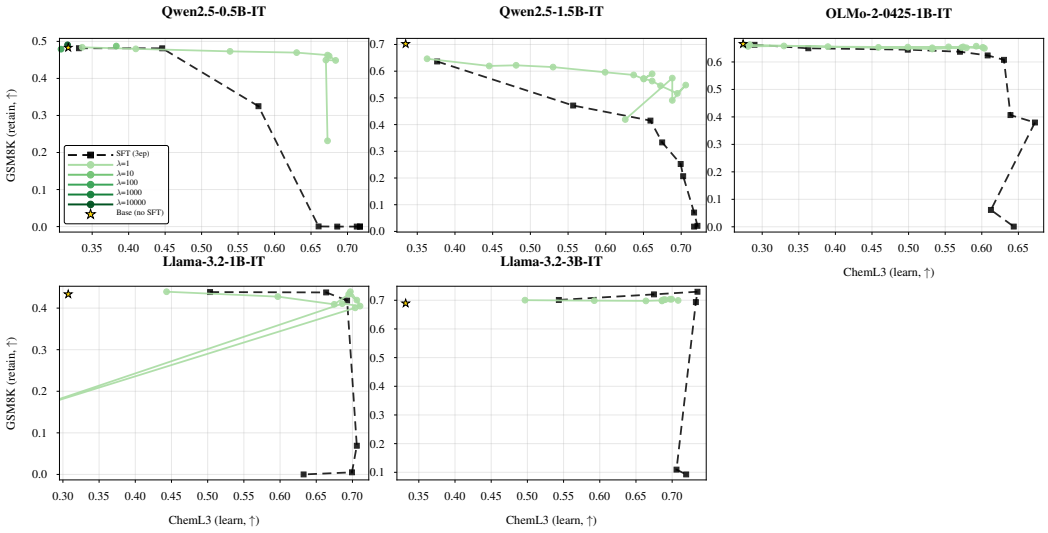


Figure 8: MAS hyperparameter sweeps (Aljundi et al., 2018).

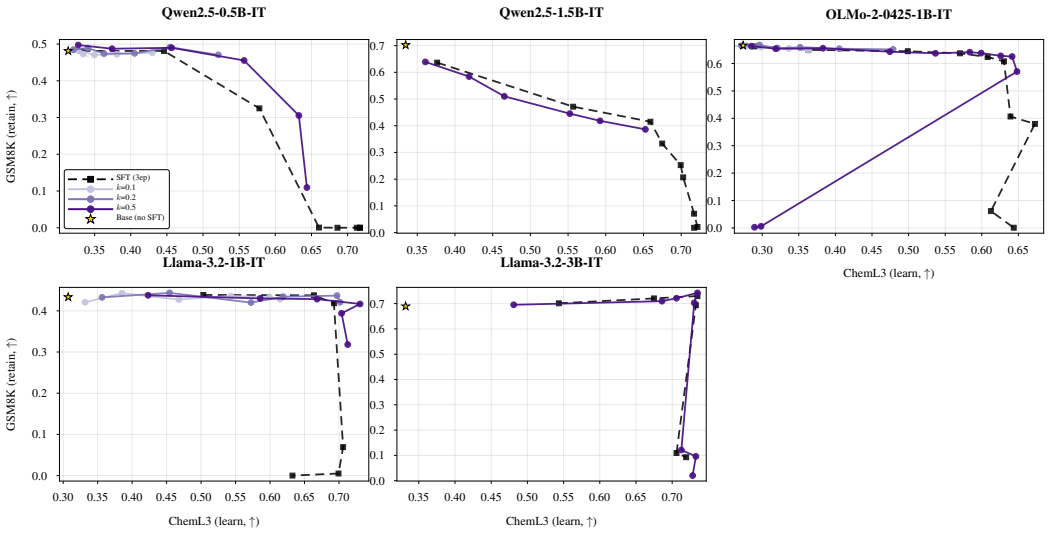


Figure 9: MIGU hyperparameter sweeps (Du et al., 2024).

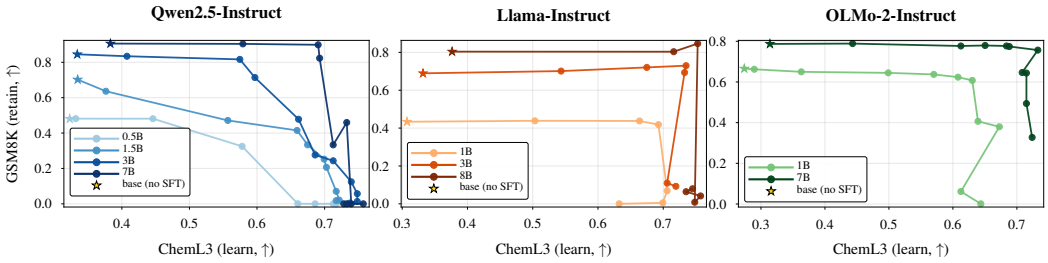


Figure 10: Effect of model size and family on learning-forgetting trade-off. Across model families and sizes, larger models generally show steeper trade-offs with SFT closer to the optimal regime. Qwen2.5 exhibits the most gradual trade-off with similar learning across sizes (< 4 pp difference), while Llama3.X and OLMo2 show stronger size dependence with early ChemL3 saturation and sharp GSM8K collapse.

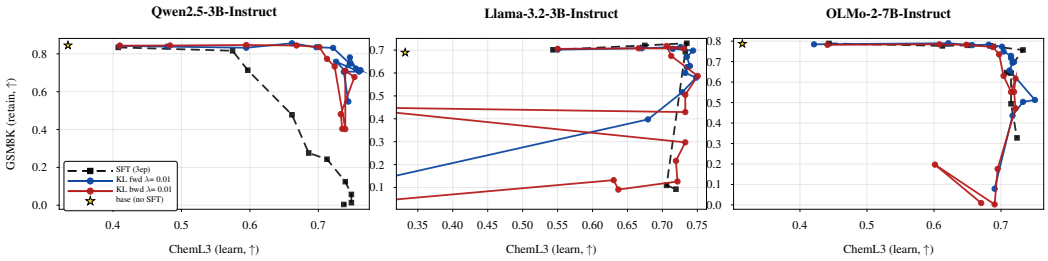


Figure 11: Pareto optimality of KL regularization Similarly to classical continual learning methods, KL regularization mitigates forgetting but does not improve learning beyond the cliff.

H.2 Pareto frontier across model sizes and families

The analysis of the learning–forgetting trade-off in Figure 10 enables a convenient comparison both across model sizes within a family and across families at comparable scales. In general, larger models exhibit a steeper trade-off, in which SFT often approaches the optimal top-right corner. Among the evaluated families, Qwen2.5 models display the most gradual trade-off during standard finetuning: all model sizes attain similar learning levels (0.5B and 7B models are within 4pp), where smaller variants sacrifice more GSM8K performance to achieve comparable learning. In contrast, for Llama3.X and OLMo2, ChemL3 accuracy is substantially more dependent on model size. In these families, ChemL3 performance mostly saturates before falling off the cliff where GSM8K accuracy collapses for no better learning. This is yet another data-point of the exceptional fine-tunability of Qwen models.

H.3 KL divergence

H.4 LoRA

We follow the canonical recipe from Biderman et al. (2024):

- **Target modules:** all transformer linear projections (`q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, `down_proj`). Attention-only adapters underperformed in their study.
- **Scaling:** $\alpha = 2r$, reported to be crucial at high rank.
- **Other hyperparameters:** dropout $= 0$, bias not adapted, task type causal LM.
- **Rank sweep:** $r \in \{16, 64, 256\}$.
- **Learning-rate sweep:** $\eta \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} \times 10^{-5} \cup \{1.2, 1.5, 1.7, 2, 3\} \times 10^{-4}$, the high head matching the $[10^{-5}, 5 \times 10^{-4}]$ range recommended by the original paper.

We use `peft` (Mangrulkar et al., 2022). All other training hyperparameters match the full finetuning baselines.

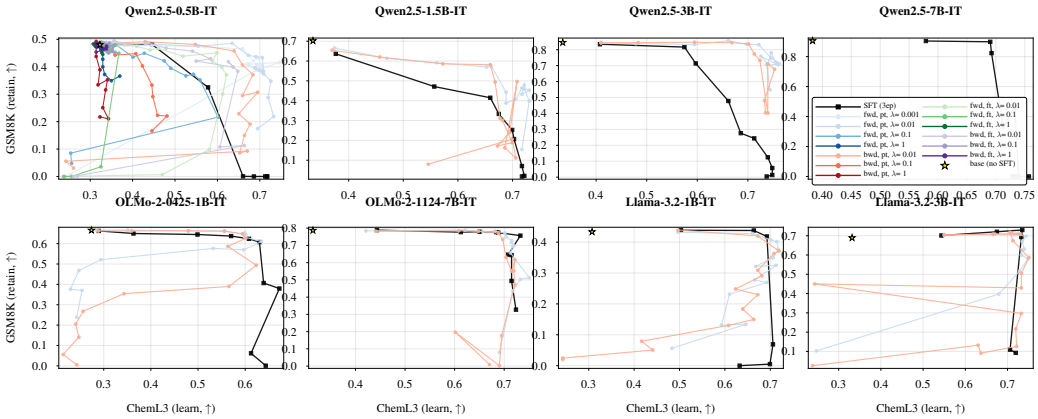


Figure 12: Cliffs created by KL regularization.

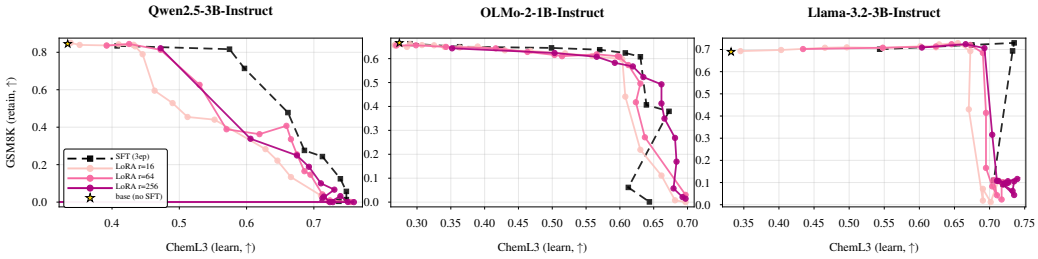


Figure 13: Full finetuning Pareto-dominates LoRA. We have found that LoRA provides a worse learning-forgetting trade-off than full finetuning across all model families and sizes, for most of the learning rates.

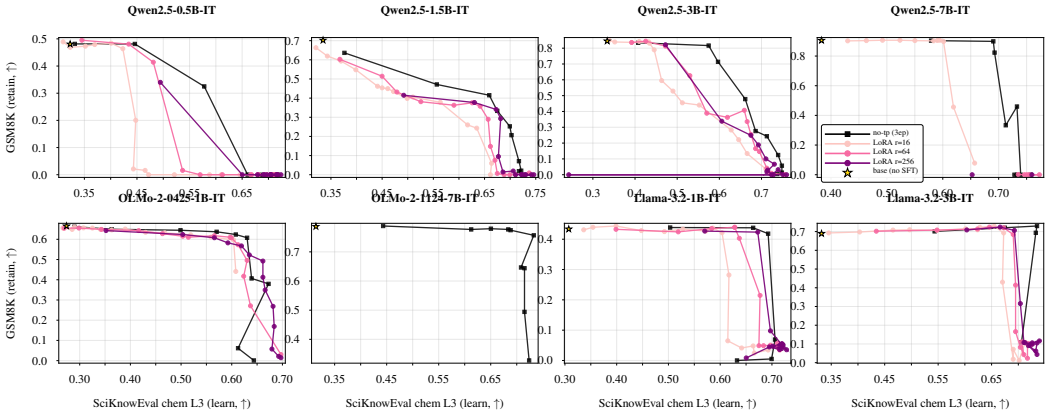


Figure 14: Full finetuning Pareto-dominates LoRA.

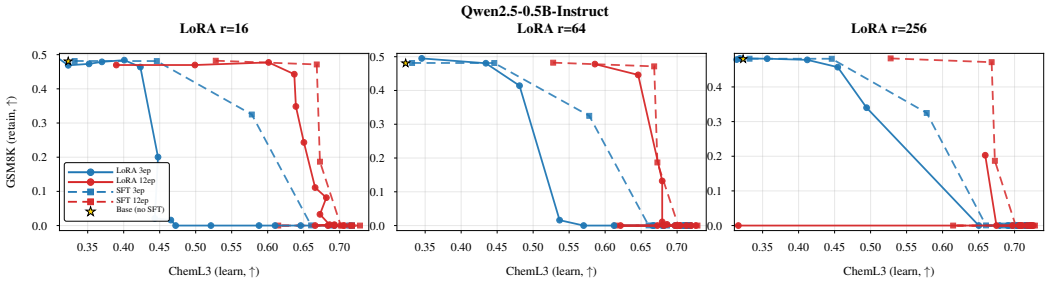


Figure 15: Epochs sweep for LoRA. Full finetuning still dominates LoRA

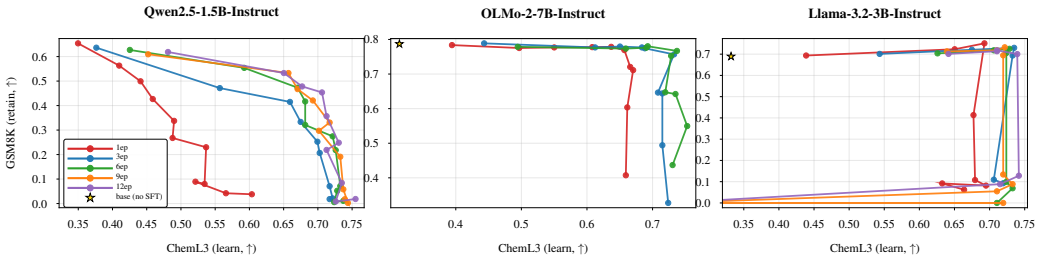


Figure 16: Increasing training epochs creates cliffs. For some models, learning is not yet saturated at low epoch counts, so higher learning rates trade retention for improved learning. As epochs increase, learning saturates and the Pareto frontier develops a cliff, where larger learning rates mainly increase forgetting with little gain in learning.

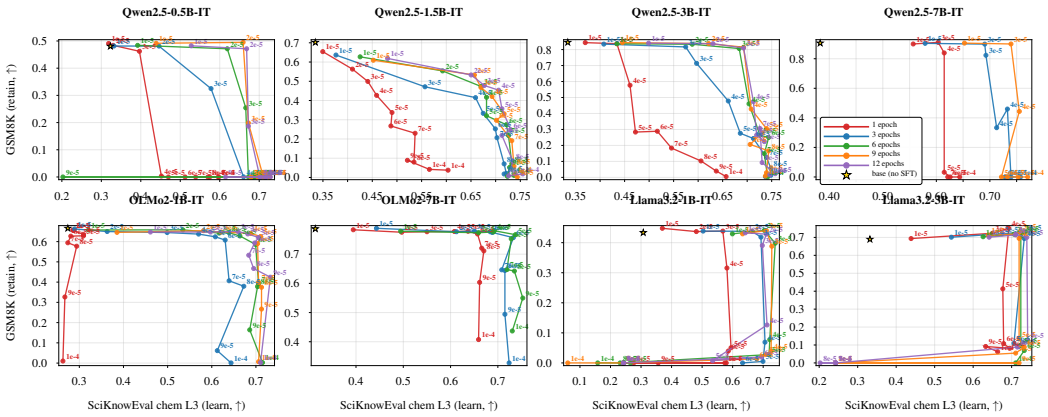


Figure 17: Cliffs created by different epochs.

H.5 Number of epochs per each model

H.6 Composition of CL methods

Interestingly some of the aforementioned CL methods use different mechanisms to achieve similar results. Particularly interesting is the case of SIB, for two reasons: 1) SIB is applied once before SFT, and naturally composes with any downstream intervention; 2) a flatter reparameterization should aid in mitigating forgetting also when combined with other CL methods. We show in 18 the effect of composing SIB with PTGI or EWC for some settings.

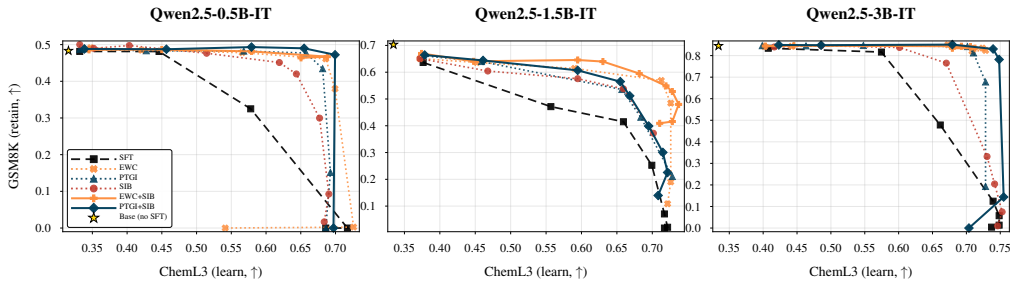


Figure 18: Composition of of scale symmetries with other continual learning. We show that the scale invariant reparameterization SIB can be combined with other continual learning methods to dominate the Pareto frontier.

I UNCERTAINTY ESTIMATES

We run two additional seeds for SFT on every model of Figure 3, and for every method on Qwen2.5-1.5B, each at the learning rate of highest cliffness. Table 3 reports the mean and 95% t-interval over the three seeds. Although there is some stochasticity, the conclusions are unchanged.

SFT across models Model	ChemL3	GSM8K
Qwen2.5-0.5B	0.580 ± 0.021	0.322 ± 0.028
Qwen2.5-1.5B	0.656 ± 0.008	0.405 ± 0.036
Qwen2.5-3B	0.604 ± 0.017	0.695 ± 0.042
Qwen2.5-7B	0.696 ± 0.012	0.829 ± 0.013
OLMo-2-1B	0.627 ± 0.008	0.607 ± 0.005
Llama-3.2-1B	0.698 ± 0.012	0.384 ± 0.090
Llama-3.2-3B	0.732 ± 0.013	0.738 ± 0.020
Llama-3.1-8B	0.751 ± 0.006	0.848 ± 0.017

Methods on Qwen2.5-1.5B Method	ChemL3	GSM8K
SFT	0.656 ± 0.008	0.405 ± 0.036
EWC ($\lambda = 1$)	0.703 ± 0.021	0.572 ± 0.009
MAS ($\lambda = 1$)	0.699 ± 0.029	0.533 ± 0.034
SIB (V/O+MLP)	0.657 ± 0.006	0.545 ± 0.015
PTGI (all)	0.652 ± 0.012	0.531 ± 0.010
NanoAdam ($k = 0.2$)	0.650 ± 0.025	0.402 ± 0.039
MIGU ($k = 0.5$)	0.650 ± 0.006	0.393 ± 0.023

Table 3: Uncertainty estimates. Accuracy (mean $\pm$ 95% t-interval over 3 seeds) at the learning rate of highest cliffness. Left: SFT across models. Right: all methods on Qwen2.5-1.5B.

J ADDITIONAL MODELS

Qwen3. We run the same protocol on the Qwen3 family, with thinking disabled for both finetuning and evaluation, for SFT, EWC, SIB and PTGI (Table 4). The same two-regime pattern appears: either SFT already cliffs, or continual learning methods reach the cliff, again improving retention but not learning.

Pretrained versus instruction-tuned models. We focus on instruction-tuned models because they carry more accumulated post-training capabilities, which makes forgetting more measurable and practically relevant. To compare, we run plain SFT sweeps over four learning rates on the pretrained and the instruction-tuned versions of Qwen2.5-1.5B and Qwen2.5-3B (Table 5). Instruction-tuned models are the harder, more brittle setting.

	Qwen3-0.6B	Qwen3-1.7B	Qwen3-4B	Qwen3-8B
Base	0.38	0.50	0.56	0.59
SFT	0.86	0.91	0.96	0.95
Best method	0.96 (EWC)	0.99 (PTGI)	0.97 (SIB)	0.97 (PTGI)

Table 4: Cliffness on Qwen3 (ChemL3 learning, GSM8K retention), with thinking disabled.

SFT cliffness	Qwen2.5-1.5B	Qwen2.5-3B
Pretrained model	0.92	0.94
Instruction-tuned model	0.67	0.80

Table 5: Pretrained versus instruction-tuned models. SFT cliffness (ChemL3 learning, GSM8K retention).

BROADER IMPACT

This work studies the trade-off between domain adaptation and retention of pre-training capabilities during supervised finetuning. By framing post-training as a Pareto problem, our results help practitioners see what is being lost during finetuning and reduce silent regressions in capabilities, including safety-relevant behaviors such as refusal and instruction following. Our experiments use only publicly available models and datasets in the 0.5B to 8B parameter range, and quantitative conclusions may not transfer directly to larger models or multi-stage RL pipelines.

K COMPUTE RESOURCES

All of the experiments have been performed using a single A100 80GB, H100, or B200 GPU.