# FINE-TUNING WITHOUT FORGETTING VIA LOSS-ADAPTIVE LEARNING RATES

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Fine-tuning large language models on new data improves task performance but degrades capabilities learned during pretraining, a phenomenon known as catastrophic forgetting. Existing methods mitigate this by modifying the fine-tuning objective to suppress high-loss tokens or sequences, but these tokens are essential for learning new tasks, especially those with poor pretraining coverage. In such settings, hard tokens should still contribute to learning, so forgetting must be controlled without suppressing them. We identify a simple mechanism for doing so: per-step forgetting is bounded by the product of the learning rate and the square root of the current training loss. This suggests that high-loss batches are especially prone to inducing forgetting. Motivated by this observation, we introduce FINCH, a loss-adaptive learning-rate schedule that reduces the learning rate on high-loss batches and increases it as the model converges, while leaving the fine-tuning objective unchanged. Across knowledge acquisition, science, and low-resource language adaptation benchmarks on four models from three families, FINCH reduces forgetting by 94% on average while matching the task performance of standard fine-tuning. On Qwen3-4B knowledge acquisition, FINCH cuts TruthfulQA degradation by $5\times$ and reverses HaluEval degradation. Overall, our results show that learning-rate schedules are an effective tool to shape model behavior during fine-tuning, beyond just target-task optimization.

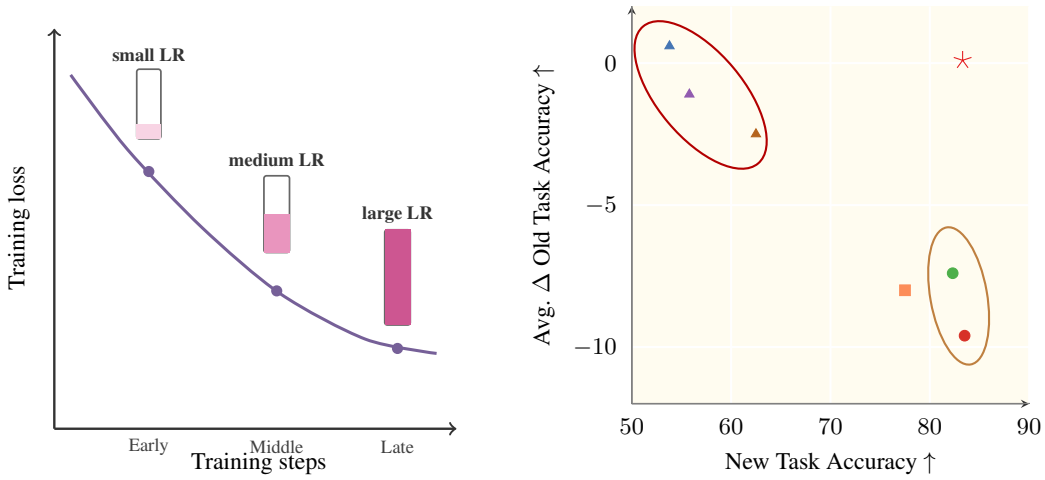
## 1 INTRODUCTION

Large language models (LLMs) are increasingly specialized through fine-tuning on narrow, task-specific corpora (Singhal et al., 2025; Roziere et al., 2023; Mecklenburg et al., 2024). While such adaptation can substantially improve target-task performance, it often degrades broader capabilities acquired during pretraining, a phenomenon known as *catastrophic forgetting* (McCloskey & Cohen, 1989; French, 1999). Such degradation leads to increased hallucination, weakened safety alignment, and degraded instruction-following and reasoning (Qi et al., 2023; Jiang et al., 2024; Luo et al., 2025; Kalajdzievski, 2024; Gekhman et al., 2024). A common mitigation is to replay pretraining data during fine-tuning (Rolnick et al., 2019; de Masson D'Autume et al., 2019; Scialom et al., 2022), but this is rarely possible in practice because the pretraining data for most modern LLMs is partially or fully proprietary (Grattafiori et al., 2024; Yang et al., 2025; Li et al., 2024).

This has motivated work in the *data-oblivious* setting, where only the task corpus is available (Sanyal et al., 2025). A common observation across these methods is that tokens or sequences assigned high loss by the pretrained model are the primary drivers of forgetting. This motivates suppressing their influence during fine-tuning: Sanyal et al. (2025) downweight high-loss sequences, Lin et al. (2025) reweight losses at the token level, and Wu et al. (2025a) mask tokens whose perplexity exceeds a threshold. However, these approaches struggle when the target task requires learning new content, because the pretrained model assigns low probability and therefore high loss to many task-relevant tokens that must be learned. For example, knowledge acquisition requires learning new names and facts the model has never seen, while low-resource language adaptation requires learning new vocabulary and grammar. Suppressing hard tokens in these settings hurts new-task performance without reliably reducing forgetting, and existing methods often fail to match vanilla SFT on new-task accuracy in this regime (Figure 1b).



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.




(a) (b)

Figure 1: **Overview of FINCH. (a)** Conceptual schematic of FINCH: early in fine-tuning, training loss is high and FINCH uses a smaller learning rate; as the model improves, the learning rate increases. **(b)** New-task accuracy versus average old-task accuracy change on knowledge acquisition with Qwen3-4B. Methods are SFT (●), FLOW (■), DFT (▲), TALR (▲), STM (▲), LoRA (●), and FINCH (⋆). The ellipses highlight two failure modes: data-reweighting methods underperform on the new task, while SFT and LoRA learn the new task but forget substantially. FINCH avoids both.

We control forgetting on a different axis: rather than changing the contribution of different tokens, we change the learning rate used for each training step. Our analysis shows that per-step forgetting is tied to the distributional mismatch between the current model and the target data, which is reflected in the current mini-batch loss. We show that per-step forgetting is bounded by the learning rate times the square root of the mini-batch loss (Section 3.2). With a constant learning rate, this bound is largest when the loss is high, typically early in fine-tuning when the model is still far from the target and updates are most likely to damage retained capabilities. To control this bound, we set the learning rate inversely proportional to the square root of the mini-batch loss. We call this schedule FINCH (**F**orgetting-aware **In**verse loss s**ch**edule), which uses smaller learning rates on high-loss batches and larger learning rates as the model moves closer to the target. This keeps the per-step forgetting bound uniform across training and implies a cumulative forgetting bound (Section 3.3). Unlike token-reweighting methods, FINCH leaves the training objective unchanged within each batch, so high-loss tokens still contribute to learning. This allows FINCH to reduce forgetting without sacrificing target-task accuracy. In contrast, simply lowering the learning rate under a standard schedule reduces forgetting but fails to reach competitive new-task accuracy (Section 4).

We evaluate on three settings where the pretrained model has limited coverage: knowledge acquisition, low-resource language adaptation, and science reasoning. Across four models from three families spanning 4B to 32B parameters, FINCH lies on the observed Pareto frontier. It reaches target-task performance comparable to standard SFT while **reducing net average forgetting by** 94% (Section 4; Appendix B). On knowledge acquisition, it cuts TruthfulQA degradation by 5× relative to SFT while better preserving hallucination detection (Section 5). Overall, our results show that learning-rate schedules shape model behavior during fine-tuning, beyond target-task accuracy.

## 2 Related Work

### 2.1 Catastrophic Forgetting

Catastrophic forgetting refers to the degradation of previously acquired knowledge when a model is trained on new data (McCloskey & Cohen, 1989; French, 1999). Replay, which mixes old training data into fine-tuning, is a common mitigation strategy (Rolnick et al., 2019; de Masson D'Autume et al., 2019; Scialom et al., 2022), but pretraining data is rarely available for modern

LLMs (Grattafiori et al., 2024; Yang et al., 2025). Other methods constrain updates using old-task representations (Lin et al., 2022a; Singh et al., 2025) or protect important parameters (Song et al., 2025), but similarly require old data and are mostly evaluated on small image benchmarks such as MNIST (LeCun et al., 2002) and CIFAR-10 (Krizhevsky et al., 2009). Like other data-oblivious methods, FINCH requires no access to pretraining data.

In the data-oblivious setting, several methods reduce forgetting by constraining updates or modifying the fine-tuning objective. Kirkpatrick et al. (2017) regularize weights to stay close to their pretrained values. LoRA (Hu et al., 2022) constrains updates to a low-rank subspace, reducing forgetting but hurting target-domain performance (Biderman et al., 2024). Distillation-based methods (Agarwal et al., 2024; Lu & Lab, 2025) replace SFT with student rollouts scored by a stronger teacher; this is computationally expensive and struggles when the base model assigns low probability to relevant sequences (Shao et al., 2024). A separate line of work reduces forgetting by downweighting high-loss tokens or sequences: Sanyal et al. (2025) upweight low-loss sequences, Wu et al. (2025a) mask tokens above a perplexity threshold, Lin et al. (2025) scale each token's loss by $p^{\frac{1}{\tau}}$ where $p$ is the token probability and $\tau$ is a constant, and Wu et al. (2025b) rescale gradients by token probability. However, for many tasks learning hard tokens is essential, so suppressing them hurts target performance. FINCH leaves the training objective unchanged and controls forgetting solely through the learning rate.

### 2.2 Learning Rate

The learning rate affects both optimization speed and the final solution: larger rates can bias training toward wider, better-generalizing minima (Smith & Topin, 2019), but may cause instability beyond a critical threshold (Lewkowycz et al., 2020). In practice, transformer fine-tuning commonly uses linear warmup followed by cosine decay (Goyal et al., 2017; Loshchilov & Hutter, 2016; Vaswani et al., 2017). Prior work studies warmup as a way to stabilize early training, either by moving the model toward flatter, better-conditioned regions (Kalra & Barkeshli, 2024) or by controlling Adam-related instability, large angular updates, and high gradient signal-to-noise ratio (Kosson et al., 2024).

Learning rates have also been studied for catastrophic forgetting. Kenneweg et al. (2022) assign separate learning rates to BERT layers and tune them jointly with Bayesian optimization, but this becomes expensive for modern LLMs because the number of hyperparameters grows with model depth and each trial requires a full fine-tuning run (Kandasamy et al., 2015). Lin et al. (2025) prescribe a fixed small learning rate to reduce forgetting, but in our experiments this comes at severe cost to new task performance. FINCH instead adapts the learning rate to the current mini-batch loss, reducing update size when forgetting risk is high while allowing larger steps as the model approaches the target distribution.

Due to space constraints, we defer a more detailed discussion of related work to Appendix C.

## 3 Method

In this section, we formalize catastrophic forgetting during fine-tuning and derive a simple adaptive learning rate schedule to mitigate it. We begin by setting up notation and defining our forgetting metric (Section 3.1). We then derive a theoretical bound showing that per-step forgetting is controlled by the product of the learning rate and the square root of the current training loss (Section 3.2). This motivates an adaptive schedule that reduces the learning rate when training loss is large and increases it as training converges, keeping the per-step forgetting bound roughly constant (Section 3.3).

### 3.1 Setup and Forgetting Metric

Let $p_\theta$ denote the model distribution parameterized by $\theta$, initialized at pretrained weights $\theta_0$, and let $q$ denote the fine-tuning target distribution. Given a training dataset $D_{\text{train}} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ with $(x_i, y_i) \sim q$, fine-tuning minimizes the cross-entropy

$$\mathcal{L}(\theta) = -\mathbb{E}_{(x,y)\sim D_{\text{train}}}[\log p_\theta(y \mid x)]$$

via SGD[1] over $T$ steps. At each step $i$, a mini-batch $\mathcal{B}_i \subset D_{\text{train}}$ is sampled and parameters are updated as

$$\theta_{i+1} = \theta_i - \eta_i \nabla_\theta \mathcal{L}_{\mathcal{B}_i}(\theta_i),$$

where $\eta_i > 0$ is the learning rate at step $i$, and we write $p_i := p_{\theta_i}$ for the model at step $i$.

Fine-tuning may degrade capabilities acquired during pretraining. To measure this, let $q_{\text{old}}$ denote the distribution over pretraining tasks, and let $D_{\text{old}} \sim q_{\text{old}}$ be a held-out evaluation dataset used only for measuring forgetting, not for training.

**Definition 1** (Forgetting)**.** *The old-task loss is $\mathcal{L}_{\text{old}}(p_i) = -\mathbb{E}_{(x,y)\sim q_{\text{old}}}[\log p_i(y \mid x)]$, estimated on $D_{\text{old}}$. Per-step forgetting is $\Delta\mathcal{L}_{\text{old}}(p_i) := \mathcal{L}_{\text{old}}(p_{i+1}) - \mathcal{L}_{\text{old}}(p_i)$.*

Bounding per-step forgetting at each step directly controls cumulative forgetting $\mathcal{L}_{\text{old}}(p_T) - \mathcal{L}_{\text{old}}(p_0) = \sum_{i=0}^{T-1} \Delta\mathcal{L}_{\text{old}}(p_i)$.

### 3.2 Forgetting Analysis

We now bound the per-step forgetting $\Delta\mathcal{L}_{\text{old}}(p_i)$ defined in Section 3.1. We state the theorem and provide a proof sketch here. Appendix A gives the full proof for sequence losses and mini-batches.

By definition,

$$\Delta\mathcal{L}_{\text{old}}(p_i) = -\mathbb{E}_{(x,y)\sim q_{\text{old}}}\left[\log \frac{p_{i+1}(y \mid x)}{p_i(y \mid x)}\right] \le \max_{(x,y)} \left|\log \frac{p_{i+1}(y \mid x)}{p_i(y \mid x)}\right|.$$

Thus, controlling the worst-case pointwise log-ratio between consecutive models controls per-step forgetting. We now bound this log-ratio in terms of the learning rate and the current training loss. The argument uses standard boundedness conditions on the inputs, the fine-tuning trajectory, and the activation derivatives.

**Assumption 1** (Bounded network and smoothness)**.** *The model $p_\theta$ is a softmax network. The input domain is bounded, the fine-tuning trajectory remains in a bounded parameter region, and the activation functions have bounded first and second derivatives on the relevant range. Concretely, there exist constants $B_x, B_\theta, L_1, L_2 > 0$ such that*

$$\|x\|_2 \le B_x, \qquad \|\theta_i\|_2 \le B_\theta, \qquad \sup_u |\sigma'(u)| \le L_1, \qquad \sup_u |\sigma''(u)| \le L_2$$

*for all inputs $x$, training steps $i$, and activation functions $\sigma$ appearing in the network.*

Bounded-input and bounded-parameter assumptions are standard in learning-theoretic analyses (Shalev-Shwartz & Ben-David, 2014). The smoothness condition is also mild for modern LLM architectures: SwiGLU-style activations (Dauphin et al., 2017; Shazeer, 2020), which are widely used in LLMs, have bounded first and second derivatives on bounded input ranges. We can now state the main per-step bound.

**Theorem 1** (Per-step forgetting bound)**.** *Under Assumption 1, there exist constants $C_1, C_2 > 0$ such that, at every training step $i$,*

$$\Delta\mathcal{L}_{\text{old}}(p_i) \le C_1 \eta_i \sqrt{\mathcal{L}_{\mathcal{B}_i}(\theta_i)} + C_2 \eta_i^2 \mathcal{L}_{\mathcal{B}_i}(\theta_i),$$

*where*

$$\mathcal{L}_{\mathcal{B}_i}(\theta_i) := \frac{1}{|\mathcal{B}_i|} \sum_{(x,y)\in\mathcal{B}_i} \text{CE}(q(\cdot \mid x) \| p_i(\cdot \mid x))$$

*is the average cross-entropy loss on the mini-batch $\mathcal{B}_i$ used at step $i$.*

**Proof sketch.** The proof has three steps (Full proof in Appendix A).

*Step 1: Uniform bounds.* By repeated application of the chain rule, Assumption 1 implies that the logit Jacobian, the score-function norm, and the log-probability Hessian are uniformly bounded along the fine-tuning trajectory.

[1]We analyze SGD for simplicity; in practice, FINCH works directly with AdamW, as we confirm in Section 4.

*Step 2: Taylor expansion of the old-task log-ratio.* Fix an old-task example $(x_{\text{old}}, y_{\text{old}})$. Taylor's theorem applied to $\log p_\theta(y_{\text{old}} \mid x_{\text{old}})$ with the SGD update $\theta_{i+1} = \theta_i - \eta_i \nabla_\theta \mathcal{L}_{\mathcal{B}_i}(\theta_i)$ gives

$$\left| \log \frac{p_{i+1}(y_{\text{old}} \mid x_{\text{old}})}{p_i(y_{\text{old}} \mid x_{\text{old}})} \right| \leq C_1 \eta_i \|\nabla_\theta \mathcal{L}_{\mathcal{B}_i}(\theta_i)\|_2 + C_2 \eta_i^2 \|\nabla_\theta \mathcal{L}_{\mathcal{B}_i}(\theta_i)\|_2^2.$$

Thus, the absolute old-task log-ratio is controlled by the norm of the fine-tuning gradient.

*Step 3: Bounding the fine-tuning gradient by the loss.* For a training example $(x_{\text{train}}, y_{\text{train}})$ with target distribution $q(\cdot \mid x_{\text{train}})$, the cross-entropy gradient has the form

$$\nabla_\theta \text{CE}(q\|p_i) = J_{\theta_i}(x_{\text{train}})^\top \big(p_i(\cdot \mid x_{\text{train}}) - q(\cdot \mid x_{\text{train}})\big).$$

Using the Jacobian bound, Pinsker's inequality, and $\text{KL}(q\|p_i) \leq \text{CE}(q\|p_i)$ gives $\|\nabla_\theta \text{CE}(q\|p_i)\|_2 \leq C_3 \sqrt{\text{CE}(q\|p_i)}$. Averaging over the mini-batch gives $\|\nabla_\theta \mathcal{L}_{\mathcal{B}_i}(\theta_i)\|_2 \leq C_3 \sqrt{\mathcal{L}_{\mathcal{B}_i}(\theta_i)}$. Substituting this into Step 2 gives the theorem (full proof in Appendix A).

The theorem shows that the leading term in the forgetting bound scales as $\eta_i \sqrt{\mathcal{L}_{\mathcal{B}_i}(\theta_i)}$. Thus, per-step forgetting is controlled by the product of the learning rate and the square root of the current mini-batch loss. This motivates choosing a smaller learning rate on high-loss batches and relaxing it as the loss decreases, which leads to the adaptive schedule in the next section.

### 3.3 Adaptive Learning Rate

We now translate Theorem 1 into a learning-rate schedule. The leading term in the per-step bound depends on the product $\eta_i \sqrt{\mathcal{L}_{\mathcal{B}_i}(\theta_i)}$. To keep this quantity fixed across training, we set

$$\eta_i = \frac{\kappa}{\sqrt{\mathcal{L}_{\mathcal{B}_i}(\theta_i)}},$$

where $\kappa > 0$ controls the target scale of the per-step change. This gives smaller learning rates on high-loss batches and larger learning rates as the model approaches the fine-tuning target. Substituting this choice into Theorem 1 makes the leading term constant across steps, which gives the following cumulative bound.

**Corollary 1.** *Under the conditions of Theorem 1, suppose the learning rates are sufficiently small and*

$$\eta_i = \frac{\kappa}{\sqrt{\mathcal{L}_{\mathcal{B}_i}(\theta_i)}}.$$

*Then the cumulative forgetting satisfies*

$$\mathcal{L}_{\text{old}}(p_T) - \mathcal{L}_{\text{old}}(p_0) = O(T\kappa).$$

In contrast, a fixed learning rate $\eta$ gives a per-step forgetting upper bound of $O(\eta \sqrt{\mathcal{L}_{\mathcal{B}_i}(\theta_i)})$, which is large early in training when the loss is high and shrinks only as the model converges. The adaptive schedule keeps this upper bound uniformly constant at $O(\kappa)$, suggesting more controlled forgetting throughout training compared to a fixed learning rate. At the same time, this analysis is not intended to identify the optimal schedule or establish tightness of the bound. Rather, it motivates a simple loss-adaptive schedule, whose effectiveness we validate empirically in Sections 4 and 5.

In practice, we implement the schedule as

$$\eta_i = \min\left( \frac{\eta_{\text{base}}}{\sqrt{\bar{\mathcal{L}}_i + \varepsilon}},\ \eta_{\max} \right),$$

where $\varepsilon > 0$ ensures numerical stability and $\eta_{\max}$ caps the learning rate when the training loss becomes small. To maintain learning-rate stability, $\bar{\mathcal{L}}_i$ is an exponential moving average of mini-batch losses with coefficient $\alpha = 0.9$ fixed throughout: $\bar{\mathcal{L}}_i = \alpha \bar{\mathcal{L}}_{i-1} + (1-\alpha)\mathcal{L}_{\mathcal{B}_i}(\theta_i)$. We fix $\eta_{\max} = 5 \times 10^{-5}$ across all experiments and treat $\eta_{\text{base}}$ as a hyperparameter selected via grid search.

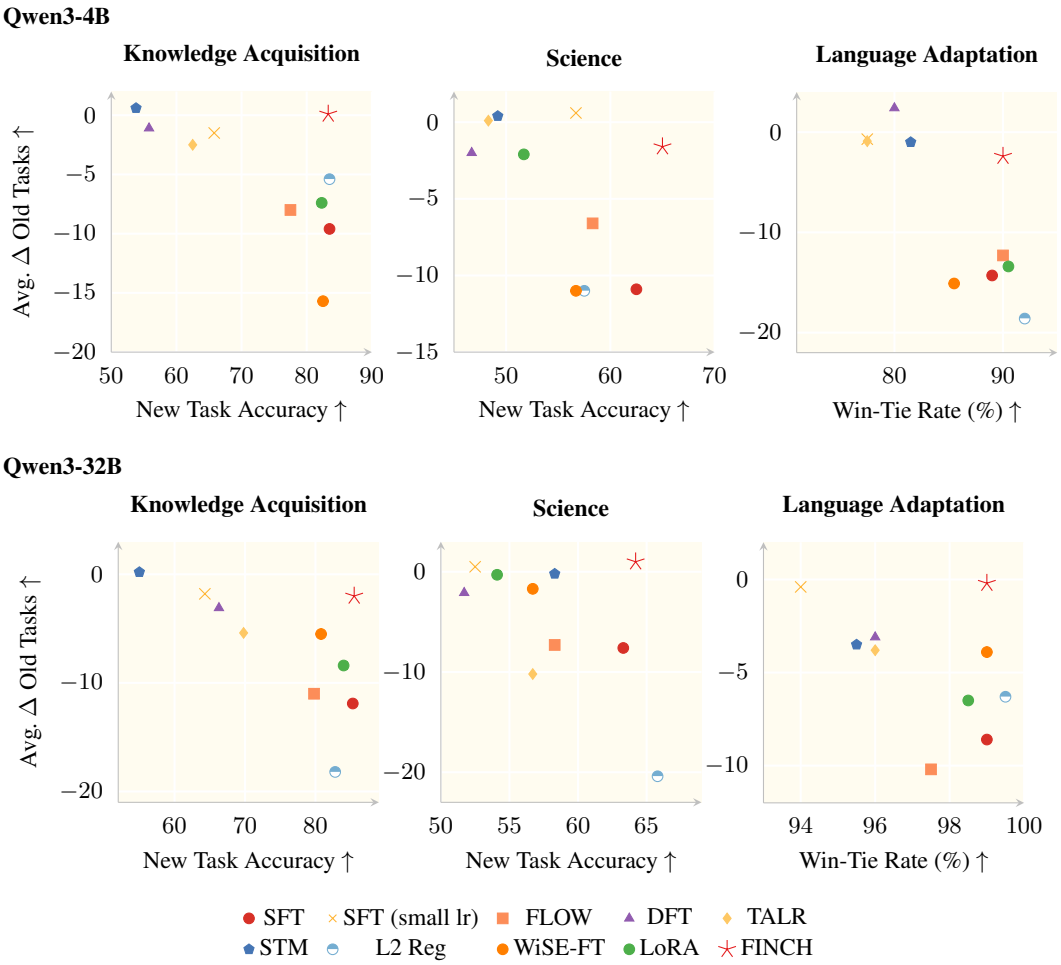



Figure 2: Target-task performance versus average old-task change across fine-tuning methods on Qwen3-4B (top row) and Qwen3-32B (bottom row). Each point represents one method in one model–task setting; the upper right indicates stronger adaptation and better retention.

# 4 FINCH Improves the Learning–Forgetting Trade-Off

## 4.1 Experimental Setting

**Tasks.** We evaluate FINCH on three settings where learning the target task requires adapting to examples the pretrained model initially finds difficult, including low-probability task-relevant tokens:

- *Language adaptation (LA):* Instruction following in Galician, a low-resource language with limited pretraining coverage. We use Galician Alpaca, a translated version of the Stanford Alpaca instruction-following dataset (Taori et al., 2023; Chen et al., 2024b).
- *Science:* Undergraduate-level scientific reasoning using the Chemistry L-3 subset of SciKnowEval (Feng et al., 2024), a multiple-choice benchmark covering chemistry concepts.
- *Knowledge acquisition (KA):* Acquiring novel factual information using TOFU (Maini et al., 2024), a dataset of 200 synthetic author profiles each consisting of 20 question-answer pairs probing biographical facts.

**Evaluation.** We evaluate performance on held-out examples from each target task. For LA, we report the win-tie rate of the fine-tuned model against the pretrained model, using an LLM judge (Zheng et al., 2023). For Science and KA, we report multiple-choice accuracy. To measure forgetting, we report the change in performance relative to the pretrained model across four general

Table 1: Target-task performance and old-task change on Qwen3-4B and Qwen3-32B. KA and Science targets are held-out accuracy; LA target is the Galician win-tie rate against the pretrained model. Avg. $\Delta$ is the signed mean of fine-tuned minus pretrained performance, in percentage points, over HellaSwag, WinoGrande, IFEval, and MMLU. Higher values are better. Full per-benchmark results for all four models are in Appendix B.

| Model | Method | KA | | Science | | LA | |
|---|---|---|---|---|---|---|---|
| | | Target ↑ | Avg. $\Delta$ ↑ | Target ↑ | Avg. $\Delta$ ↑ | Win-Tie ↑ | Avg. $\Delta$ ↑ |
| Qwen3-4B | Base | 53.5 | +0.0 | 29.2 | +0.0 | 50.0% | +0.0 |
| | SFT | 83.5 | −9.6 | 62.5 | −10.9 | 89.0% | −14.3 |
| | SFT (small lr) (Lin et al., 2025) | 65.8 | −1.5 | 56.7 | +0.6 | 77.5% | −0.7 |
| | FLOW (Sanyal et al., 2025) | 77.5 | −8.0 | 58.3 | −6.6 | 90.0% | −12.3 |
| | DFT (Wu et al., 2025b) | 55.8 | −1.1 | 46.7 | −2.0 | 80.0% | +2.4 |
| | TALR (Lin et al., 2025) | 62.5 | −2.5 | 48.3 | +0.1 | 77.5% | −0.9 |
| | STM (Wu et al., 2025a) | 53.8 | +0.6 | 49.2 | +0.4 | 81.5% | −1.0 |
| | L2 Reg (Kirkpatrick et al., 2017) | 83.5 | −5.4 | 57.5 | −11.0 | 92.0% | −18.6 |
| | WiSE-FT (Wortsman et al., 2022) | 82.5 | −15.7 | 56.7 | −11.0 | 85.5% | −15.1 |
| | LoRA (Hu et al., 2022) | 82.3 | −7.4 | 51.7 | −2.1 | 90.5% | −13.4 |
| | FINCH | 83.3 | +0.1 | 65.0 | −1.6 | 90.0% | −2.4 |
| Qwen3-32B | Base | 57.8 | +0.0 | 50.8 | +0.0 | 50.0% | +0.0 |
| | SFT | 85.3 | −11.9 | 63.3 | −7.6 | 99.0% | −8.6 |
| | SFT (small lr) (Lin et al., 2025) | 64.3 | −1.8 | 52.5 | +0.5 | 94.0% | −0.4 |
| | FLOW (Sanyal et al., 2025) | 79.8 | −11.0 | 58.3 | −7.3 | 97.5% | −10.2 |
| | DFT (Wu et al., 2025b) | 66.3 | −3.1 | 51.7 | −2.1 | 96.0% | −3.1 |
| | TALR (Lin et al., 2025) | 69.8 | −5.4 | 56.7 | −10.2 | 96.0% | −3.8 |
| | STM (Wu et al., 2025a) | 55.0 | +0.2 | 58.3 | −0.2 | 95.5% | −3.5 |
| | L2 Reg (Kirkpatrick et al., 2017) | 82.8 | −18.2 | 65.8 | −20.4 | 99.5% | −6.3 |
| | WiSE-FT (Wortsman et al., 2022) | 80.8 | −5.5 | 56.7 | −1.7 | 99.0% | −3.9 |
| | LoRA (Hu et al., 2022) | 84.0 | −8.4 | 54.1 | −0.3 | 98.5% | −6.5 |
| | FINCH | 85.5 | −2.0 | 64.2 | +1.0 | 99.0% | −0.2 |

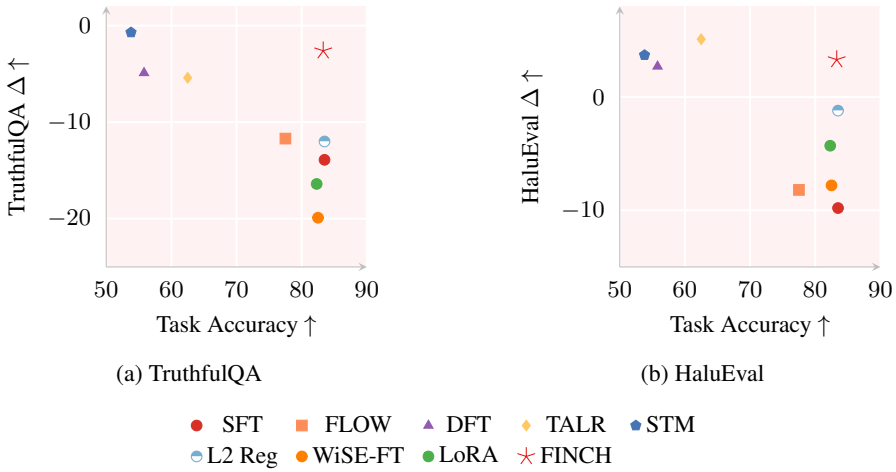
benchmarks: HellaSwag (Zellers et al., 2019) and WinoGrande (Sakaguchi et al., 2021) for commonsense reasoning, MMLU (Hendrycks et al., 2020) for general knowledge and reasoning, and IFEval (Zhou et al., 2023) for instruction following.

**Baselines.** We compare against three classes of mitigation baselines: objective-modifying methods that downweight low-probability tokens or sequences, including FLOW (Sanyal et al., 2025), DFT (Wu et al., 2025b), TALR (Lin et al., 2025), and STM (Wu et al., 2025a); deviation-constraining methods, including L2 regularization (Kirkpatrick et al., 2017), SFT (small lr) (Lin et al., 2025), and WiSE-FT (Wortsman et al., 2022); and parameter-efficient fine-tuning via LoRA (Hu et al., 2022). We evaluate four models: Qwen3-4B-Instruct and Qwen3-32B (Yang et al., 2025), Llama-3-8B (Grattafiori et al., 2024), and Gemma 4 26B A4B. We report the Qwen3-4B and Qwen3-32B results here and the Llama-3-8B and Gemma 4 26B A4B results in Appendix B (Figure 4). For each baseline method, we use a standard warmup-cosine schedule, sweep hyperparameters, and report the checkpoint with the highest validation target-task performance; details are in Appendix B.

### 4.2 Results

Figure 2 summarizes the trade-off between target-task adaptation and preservation of general capabilities on Qwen3-4B and Qwen3-32B. The corresponding Llama-3-8B and Gemma 4 26B A4B plots are in Appendix B (Figure 4). Across the twelve model–task settings, standard SFT achieves strong target-task performance but substantially degrades performance on the general benchmarks. Several baselines reduce forgetting, but often by sacrificing adaptation performance: for example, STM, DFT, and TALR preserve old capabilities better in some settings but obtain much lower target-task accuracy. In contrast, FINCH consistently lies near the upper-right region of the trade-off plots, achieving competitive target-task performance while incurring substantially smaller average benchmark degradation. On the observed point estimates, it lies on the Pareto frontier in all twelve settings. Overall, FINCH reduces net average forgetting by $94\%$ relative to standard SFT (Appendix B).

Table 1 shows this pattern in detail across KA, Science, and LA on the two Qwen models. On Qwen3-4B KA, FINCH reaches $83.3\%$ task accuracy, comparable to SFT's $83.5\%$, while reducing




(a) TruthfulQA (b) HaluEval

Figure 3: Truthfulness and hallucination detection vs. task accuracy on knowledge acquisition (Qwen3-4B). Higher TruthfulQA and HaluEval Δ indicate less degradation. FINCH achieves competitive task accuracy with substantially less degradation on both benchmarks.

average forgetting from $-9.6$ to $+0.1$. On Qwen3-32B, FINCH matches or exceeds SFT's target performance on KA (85.5 versus 85.3), Science (64.2 versus 63.3), and LA (99.0% versus 99.0%), while improving Avg. Δ from $-11.9$ to $-2.0$, from $-7.6$ to $+1.0$, and from $-8.6$ to $-0.2$, respectively. Detailed results for Llama-3-8B and Gemma 4 26B A4B, as well as full per-benchmark results for all four models, are provided in Appendix B (Tables 3–5), showing a similar pattern. These results show that FINCH learns the new tasks without substantially sacrificing general capabilities on average, although some individual benchmark declines remain.

**Sequential fine-tuning.** To test whether FINCH retains capabilities learned in earlier stages, we train Qwen3-4B successively on Science, Galician language adaptation, and KA, carrying each stage's model into the next. Both methods learn the final KA task, reaching 80.8% accuracy for SFT and 81.3% for FINCH. By the end, Science accuracy and Galician win-tie rate have fallen from their respective peaks by 10.0 and 2.5 points under SFT, versus only 0.8 and 0.5 points under FINCH (Table 6 in Appendix B).

## 5 FINCH Better Preserves Reliability Under Fine-Tuning

Beyond raw capability, factuality and robustness to hallucination are important properties (Wang et al., 2024; Huang et al., 2025) that can degrade after fine-tuning (Gekhman et al., 2024; Zucchet et al., 2025). We evaluate truthfulness and hallucination detection on Qwen3-4B.

**Factuality and Truthfulness.** We use multiple-choice TruthfulQA (Lin et al., 2022b; Gao et al., 2024) and report the change in accuracy relative to the pretrained model. Figure 3 shows the trade-off between target-task performance and TruthfulQA degradation for KA. Corresponding Science and LA results and the detailed KA results (Table 14) are in Appendix B.8. Across the three tasks, FINCH achieves competitive target-task performance while substantially reducing TruthfulQA degradation. In KA, standard SFT reaches 83.5% task accuracy with a 13.9-point TruthfulQA drop, whereas FINCH reaches 83.3% accuracy with only a 2.6-point drop.

**Hallucination Detection.** LLMs can often recognize hallucinated responses when explicitly asked to compare them against factual alternatives (Li et al., 2023). We ask whether FINCH preserves this ability after fine-tuning, measuring the change in HaluEval accuracy relative to the pretrained model. Figure 3b reports the HaluEval results; detailed tables for the three tasks are provided in Appendix B.8 (Tables 14, 15, and 16). In KA, standard SFT reaches 83.5% task accuracy but reduces HaluEval by 9.8 points, whereas FINCH reaches 83.3% accuracy and improves HaluEval by 3.3 points. Unlike the general benchmark results, several baselines also improve HaluEval in KA.

This is most visible for the token- or sequence-reweighting methods, such as DFT, TALR, and STM, which suppress high-loss examples and therefore reduce forgetting more directly. However, these methods achieve substantially lower KA accuracy, while FINCH preserves hallucination detection without sacrificing target-task performance. The same trend appears only partially in Science and LA, suggesting that hallucination-detection changes depend on both the fine-tuning method and the target dataset. Understanding this interaction is an interesting direction for future work.

## 6 Discussion

In this section, we discuss additional experiments that examine FINCH across different settings. We also perform ablations to assess its sensitivity to training choices. Full results are in Appendix B.7.

**Math.** Our main experiments study tasks where the pretrained model performs poorly and learning new facts, vocabulary, or language involves many low-probability tokens. Math is more familiar to most pretrained models: in our measurements, only $3.1\%$ of MATH-500 tokens have log probability below $-5$, versus $23.8\%$ of KA tokens. We fine-tune Llama-3-8B-Instruct on 10K OpenR1-Math-220k examples and evaluate on MATH-500. FINCH reaches $33.3\%$ accuracy with Avg. $\Delta = -0.1$, comparable to FLOW's $33.8\%$ and $-0.4$; both largely preserve prior performance (Table 7 in Appendix B.7). On math, where very low-probability tokens are less common, existing methods already perform well, and FINCH does not clearly outperform them.

**Learning difficult examples.** We test whether FINCH learns examples that the pretrained model initially finds difficult. For Qwen3-4B KA, we select the 100 validation examples on which the pretrained model most strongly favors an incorrect answer. FINCH answers $59\%$ correctly, compared with $58\%$ for SFT and $44\%$ for FLOW (Table 8 in Appendix B.7). FINCH scales the update for an entire batch, so those hard examples retain their relative contribution to learning. Across 100 random batches at fixed checkpoints, loss differences between batches are small compared with the drop in average loss over training (Table 9 in the appendix), suggesting that the learning rate mainly follows the model's progress on the task.

**Warmup and gradient clipping.** We vary SFT's warmup duration from $1\%$ to $20\%$ on Qwen3-4B KA, but forgetting remains substantial (Table 10 in Appendix B.7). Conceptually, warmup follows a predetermined schedule $\eta_t = f(t)$, whereas FINCH adapts to the current loss, $\eta_t = f(L(\theta_t))$. Gradient clipping is another way to limit update size. We use a maximum gradient norm of $1.0$ in the main experiments and test smaller thresholds on Qwen3-4B KA (Appendix B.5). Moderate clipping has little effect; thresholds that substantially reduce forgetting also lower target-task accuracy.

**Ablations on $\alpha$ and batch size.** We fix the EMA coefficient at $\alpha = 0.9$ for every model and task. To test sensitivity, we vary $\alpha$ on Qwen3-4B KA and find little change in target accuracy or average old-task performance (Table 11 in Appendix B.7). We also test batch sizes from 32 to 256 in the same setting. Across these sizes, FINCH maintains $83.3$–$84.3\%$ target accuracy and Avg. $\Delta$ between $-1.8$ and $+1.0$ (Table 12 in the appendix).

## 7 Conclusion, Limitations, and Future Work

We introduce FINCH, a loss-adaptive learning-rate schedule to mitigate catastrophic forgetting during fine-tuning. Our per-step bound motivates smaller learning rates on high-loss batches and larger ones as the loss falls. Across four models and three tasks, FINCH reduces net average old-task degradation by $94\%$ while reaching comparable target-task performance. On Qwen3-4B KA, it also lies on the observed Pareto frontier for target accuracy versus both TruthfulQA and HaluEval.

**Limitations and Future Work.** Our theoretical bound is stated for SGD, whereas our experiments use AdamW, so it does not formally characterize the optimizer used in practice. FINCH can converge slightly more slowly than SFT: on Qwen3-4B KA, it reaches peak task performance after $490$ steps, compared with $430$ for SFT. Addressing both limitations is a direction for future work.

### AI USE STATEMENT

In this work, we used GPT-5.2 to translate Stanford Alpaca instructions and outputs from English to Galician for the language-adaptation dataset, as described in Appendix B.1. We did not use Generative AI to generate other synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, assist with translation, clean and reformat dataset, support qualitative and thematic data analysis, interpret results. Additionally, we used generative AI to aid and polish the writing and to retrieve and discover related work. We have reviewed all AI-assisted work and we take responsibility for the final content of this work.

### REPRODUCIBILITY STATEMENT

Section 3 explains the method and its analysis. Section 4 describes the tasks, baselines, evaluation, and main results. Appendix B provides the experimental details and additional results. We submit the code as supplementary material.

## REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The twelfth international conference on learning representations*, 2024.

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

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, et al. Lora learns less and forgets less. *arXiv preprint arXiv:2405.09673*, 2024.

Howard Chen, Jiayi Geng, Adithya Bhaskar, Dan Friedman, and Danqi Chen. Continual memorization of factoids in language models. *arXiv preprint arXiv:2411.07175*, 2024a.

Pinzhen Chen, Shaoxiong Ji, Nikolay Bogoychev, Andrey Kutuzov, Barry Haddow, and Kenneth Heafield. Monolingual or multilingual instruction tuning: Which makes a better alpaca. In *Findings of the Association for Computational Linguistics: EACL 2024*, pp. 1347–1356, 2024b.

Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *International conference on machine learning*, pp. 933–941. PMLR, 2017.

Cyprien de Masson D'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. Episodic memory in lifelong language learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Itay Evron, Edward Moroshko, Rachel Ward, Nathan Srebro, and Daniel Soudry. How catastrophic can catastrophic forgetting be in linear regression? In *Conference on Learning Theory*, pp. 4028–4079. PMLR, 2022.

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. *arXiv preprint arXiv:2406.09098*, 2024.

Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

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`.

Zorik Gekhman, Gal Yona, Roee Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. Does fine-tuning llms on new knowledge encourage hallucinations? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 7765–7784, 2024.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Calypso Herrera, Florian Krach, and Josef Teichmann. Local lipschitz bounds of deep neural networks. *arXiv preprint arXiv:2004.13135*, 2020.

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. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Jianheng Huang, Leyang Cui, Ante Wang, Chengyi Yang, Xinting Liao, Linfeng Song, Junfeng Yao, and Jinsong Su. Mitigating catastrophic forgetting in large language models with self-synthesized rehearsal. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1416–1428, 2024.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55, 2025.

Gangwei Jiang, Zhaoyi Li, Defu Lian, and Ying Wei. Refine large language model fine-tuning via instruction vector. *arXiv preprint arXiv:2406.12227*, 2024.

Damjan Kalajdzievski. Scaling laws for forgetting when fine-tuning large language models. *arXiv preprint arXiv:2401.05605*, 2024.

Dayal Singh Kalra and Maissam Barkeshli. Why warmup the learning rate? underlying mechanisms and improvements. *Advances in Neural Information Processing Systems*, 37:111760–111801, 2024.

Kirthevasan Kandasamy, Jeff Schneider, and Barnabás Póczos. High dimensional bayesian optimisation and bandits via additive models. In *International conference on machine learning*, pp. 295–304. PMLR, 2015.

Philip Kenneweg, Alexander Schulz, Sarah Schröder, and Barbara Hammer. Intelligent learning rate distribution to reduce catastrophic forgetting in transformers. In *International Conference on Intelligent Data Engineering and Automated Learning*, pp. 252–261. Springer, 2022.

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.

Atli Kosson, Bettina Messmer, and Martin Jaggi. Analyzing & reducing the need for learning rate warmup in gpt training. *Advances in Neural Information Processing Systems*, 37:2914–2942, 2024.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 2002.

Chungpa Lee, Jy-yong Sohn, and Kangwook Lee. Fine-tuning without forgetting in-context learning: A theoretical analysis of linear attention models. *arXiv preprint arXiv:2602.23197*, 2026.

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.

Boqi Li, Youjun Wang, and Weiwei Liu. Towards understanding catastrophic forgetting in two-layer convolutional neural networks. In *Forty-second International Conference on Machine Learning*, 2025.

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, 2024.

Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. Halueval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 6449–6464, 2023.

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. *arXiv preprint arXiv:2509.20758*, 2025.

Sen Lin, Li Yang, Deliang Fan, and Junshan Zhang. Trgp: Trust region gradient projection for continual learning. *arXiv preprint arXiv:2202.02931*, 2022a.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 3214–3252, 2022b.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Kevin Lu and Thinking Machines Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20251026. https://thinkingmachines.ai/blog/on-policy-distillation.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*, 2025.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C Lipton, and J Zico Kolter. Tofu: A task of fictitious unlearning for llms. *arXiv preprint arXiv:2401.06121*, 2024.

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

Nick Mecklenburg, Yiyou Lin, Xiaoxiao Li, Daniel Holstein, Leonardo Nunes, Sara Malvar, Bruno Silva, Ranveer Chandra, Vijay Aski, Pavan Kumar Reddy Yannam, et al. Injecting new knowledge into large language models via supervised fine-tuning. *arXiv preprint arXiv:2404.00213*, 2024.

Open R1 Team. Openr1-math-220k, 2025. URL `https://huggingface.co/datasets/open-r1/OpenR1-Math-220k`.

OpenAI. Math-500, 2023. URL `https://github.com/openai/prm800k/tree/main#math-splits`.

Penghui Qi, Xiangxin Zhou, Zichen Liu, Tianyu Pang, Chao Du, Min Lin, and Wee Sun Lee. Rethinking the trust region in llm reinforcement learning. *arXiv preprint arXiv:2602.04879*, 2026.

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

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in neural information processing systems*, 32, 2019.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, et al. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*, 2023.

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. *arXiv preprint arXiv:2502.02797*, 2025.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Thomas Scialom, Tuhin Chakrabarty, and Smaranda Muresan. Fine-tuned language models are continual learners. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 6107–6122, 2022.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*, 2026.

Anika Singh, David Martinez, Aayush Dhaulakhandi, Varun Chopade, Likhith Malipati, Vasu Sharma, Kevin Zhu, Sunishchal Dev, and Ryan Lagasse. Mitigating forgetting in continual learning with selective gradient projection. In *The 14th International Joint Conference on Natural Language Processing and The 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pp. 299–313, 2025.

Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Mohamed Amin, Le Hou, Kevin Clark, Stephen R Pfohl, Heather Cole-Lewis, et al. Toward expert-level medical question answering with large language models. *Nature medicine*, 31(3):943–950, 2025.

Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pp. 369–386. SPIE, 2019.

Shezheng Song, Hao Xu, Jun Ma, Shasha Li, Long Peng, Qian Wan, Xiaodong Liu, and Jie Yu. How to alleviate catastrophic forgetting in llms finetuning? hierarchical layer-wise and element-wise regularization. *arXiv preprint arXiv:2501.13669*, 2025.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Aladin Virmaux and Kevin Scaman. Lipschitz regularity of deep neural networks: analysis and efficient estimation. *Advances in neural information processing systems*, 31, 2018.

Yuxia Wang, Minghan Wang, Muhammad Arslan Manzoor, Fei Liu, Georgi Nenkov Georgiev, Rocktim Jyoti Das, and Preslav Nakov. Factuality of large language models: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 19519–19529, 2024.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7959–7971, 2022.

Chao-Chung Wu, Zhi Rui Tam, Chieh-Yen Lin, Yun-Nung Chen, Shao-Hua Sun, and Hung-yi Lee. Mitigating forgetting in llm fine-tuning via low-perplexity token learning. *arXiv preprint arXiv:2501.14315*, 2025a.

Yongliang Wu, Yizhou Zhou, Zhou Ziheng, Yingzhe Peng, Xinyu Ye, Xinting Hu, Wenbo Zhu, Lu Qi, Ming-Hsuan Yang, and Xu Yang. On the generalization of sft: A reinforcement learning perspective with reward rectification. *arXiv preprint arXiv:2508.05629*, 2025b.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

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.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

Wenhong Zhu, Ruobing Xie, Rui Wang, Xingwu Sun, Di Wang, and Pengfei Liu. Proximal supervised fine-tuning. *arXiv preprint arXiv:2508.17784*, 2025.

Nicolas Zucchet, Jörg Bornschein, Stephanie Chan, Andrew Lampinen, Razvan Pascanu, and Soham De. How do language models learn facts? dynamics, curricula and hallucinations. *arXiv preprint arXiv:2503.21676*, 2025.

# A PROOFS

## A.1 AUXILIARY BOUNDS IMPLIED BY ASSUMPTION 1

We first record standard consequences of Assumption 1. Since the input domain is bounded, the fine-tuning trajectory remains in a bounded parameter region, and the activation functions have bounded first and second derivatives on the relevant range, the network computation is uniformly controlled along training. In particular, the logits, the logit Jacobian, the score-function gradient, and the log-probability Hessian are uniformly bounded for all contexts and labels, uniformly over the bounded parameter region traversed by the fine-tuning trajectory. Similar boundedness and smoothness assumptions are standard in learning-theoretic analyses of gradient methods (Shalev-Shwartz & Ben-David, 2014), and related work derives explicit Lipschitz or derivative bounds for neural networks (Virmaux & Scaman, 2018; Herrera et al., 2020).

**Lemma 1.** *Under Assumption 1, there exist constants $M, G, H > 0$ such that for all parameters $\theta$ in the bounded region traversed during fine-tuning, contexts $x$, and labels $y$,*

$$\|J_\theta(x)\|_{\mathrm{op}} \le M, \qquad \|\nabla_\theta \log p_\theta(y \mid x)\|_2 \le G, \qquad \|\nabla_\theta^2 \log p_\theta(y \mid x)\|_{\mathrm{op}} \le H,$$

*where $J_\theta(x)$ denotes the Jacobian of the logits with respect to $\theta$.*

*Proof.* We give the argument for a finite-depth differentiable network; the same reasoning applies to transformer architectures under the corresponding bounded-computation assumption, since they are finite computational graphs composed of differentiable operations.

For notational simplicity, consider a layerwise representation

$$h_0 = x, \qquad a_\ell = W_\ell h_{\ell-1} + b_\ell, \qquad h_\ell = \sigma_\ell(a_\ell), \qquad \ell = 1, \dots, L,$$

with final logits $z_\theta(x) = h_L$. The argument below only uses finiteness of the computation graph and boundedness of the quantities appearing in the chain rule, not the specific feedforward form above.

We first show that all hidden states are uniformly bounded. Since the input domain is bounded, $\|h_0\|_2 = \|x\|_2 \le B_x$. Assume inductively that $h_{\ell-1}$ is uniformly bounded. Since $\theta$ lies in the bounded parameter region considered in Assumption 1, the weights $W_\ell$ and biases $b_\ell$ are uniformly bounded. Therefore the pre-activation

$$a_\ell = W_\ell h_{\ell-1} + b_\ell$$

is uniformly bounded. Since $\sigma_\ell$ has bounded first derivative, it is Lipschitz; because $\sigma_\ell$ is finite at the origin, it maps bounded sets to bounded sets. Hence $h_\ell = \sigma_\ell(a_\ell)$ is uniformly bounded. By induction, every hidden state, and therefore the final logits $z_\theta(x)$, are uniformly bounded.

Next consider the logit Jacobian $J_\theta(x) = \nabla_\theta z_\theta(x)$. Each entry of this Jacobian is obtained by repeated application of the chain rule through the network. Every factor that appears is either an input coordinate, a hidden activation, a parameter, or a first derivative of an activation. All such factors are uniformly bounded by the previous paragraph and Assumption 1. Since the network has finite depth and finite parameter dimension, there exists a constant $M > 0$ such that

$$\|J_\theta(x)\|_{\mathrm{op}} \le M.$$

For the score-function gradient,

$$\nabla_\theta \log p_\theta(y \mid x) = J_\theta(x)^\top \nabla_z \log p_\theta(y \mid x).$$

For a softmax output layer,

$$\nabla_z \log p_\theta(y \mid x) = e_y - p_\theta(\cdot \mid x),$$

so

$$\|\nabla_z \log p_\theta(y \mid x)\|_2 \le \sqrt{2}.$$

Hence

$$\|\nabla_\theta \log p_\theta(y \mid x)\|_2 \le \|J_\theta(x)\|_{\mathrm{op}} \, \|\nabla_z \log p_\theta(y \mid x)\|_2 \le \sqrt{2}\, M.$$

Absorbing $\sqrt{2}$ into the constant gives the desired bound with some $G > 0$.

Finally, consider the Hessian $\nabla^2_\theta \log p_\theta(y \mid x)$. Differentiating once more introduces finitely many terms produced by the product and chain rules. These terms involve bounded inputs, bounded hidden activations, bounded parameters, bounded first derivatives of activations, bounded second derivatives of activations, and first- and second-order derivatives of the log-softmax map. Since the logits are uniformly bounded and the log-softmax map is smooth, these log-softmax derivatives are also uniformly bounded on the relevant range. Therefore, since the architecture has finite depth and finite parameter dimension, all entries of the Hessian are uniformly bounded. Since the parameter dimension is finite, this implies a uniform operator-norm bound. Hence

$$\|\nabla^2_\theta \log p_\theta(y \mid x)\|_{\mathrm{op}} \le H$$

for some constant $H > 0$. □

### A.2 Proof of Theorem 1

We now prove the per-step forgetting bound for the sequence-level mini-batch loss used during training.

**Theorem 1** (Per-step forgetting bound)**.** *Suppose each training example in the mini-batch $\mathcal{B}_i$ is a sequence $s = (x_{1:T_s}, y_{1:T_s})$, and define its average token-level cross-entropy as*

$$\ell_s(\theta) := \frac{1}{T_s} \sum_{t=1}^{T_s} \mathrm{CE}\big(q_t(\cdot \mid x_{<t}) \| p_\theta(\cdot \mid x_{<t})\big),$$

*where $q_t(\cdot \mid x_{<t})$ is the target distribution at position $t$ conditioned on the prefix $x_{<t}$. Let*

$$\mathcal{L}_{\mathcal{B}_i}(\theta) := \frac{1}{|\mathcal{B}_i|} \sum_{s \in \mathcal{B}_i} \ell_s(\theta)$$

*denote the average sequence loss on the mini-batch used at training step $i$. Under Assumption 1, there exist constants $C_1, C_2 > 0$ such that*

$$\Delta\mathcal{L}_{\mathrm{old}}(p_i) \le C_1 \eta_i \sqrt{\mathcal{L}_{\mathcal{B}_i}(\theta_i)} + C_2 \eta_i^2 \mathcal{L}_{\mathcal{B}_i}(\theta_i).$$

*Consequently, for sufficiently small $\eta_i$,*

$$\Delta\mathcal{L}_{\mathrm{old}}(p_i) = O\Big(\eta_i \sqrt{\mathcal{L}_{\mathcal{B}_i}(\theta_i)}\Big).$$

*Proof.* Let

$$g_i^{\mathcal{B}} := \nabla_\theta \mathcal{L}_{\mathcal{B}_i}(\theta_i)$$

be the mini-batch gradient, so the SGD update is

$$\theta_{i+1} - \theta_i = -\eta_i g_i^{\mathcal{B}}.$$

We begin from the old-task loss:

$$\mathcal{L}_{\mathrm{old}}(p_{i+1}) - \mathcal{L}_{\mathrm{old}}(p_i) = -\mathbb{E}_{(x_{\mathrm{old}}, y_{\mathrm{old}}) \sim q_{\mathrm{old}}} \left[ \log \frac{p_{i+1}(y_{\mathrm{old}} \mid x_{\mathrm{old}})}{p_i(y_{\mathrm{old}} \mid x_{\mathrm{old}})} \right].$$

Therefore,

$$\Delta\mathcal{L}_{\mathrm{old}}(p_i) \le \sup_{(x_{\mathrm{old}}, y_{\mathrm{old}})} \left| \log \frac{p_{i+1}(y_{\mathrm{old}} \mid x_{\mathrm{old}})}{p_i(y_{\mathrm{old}} \mid x_{\mathrm{old}})} \right|.$$

It thus suffices to bound the pointwise log-ratio on the right-hand side.

Fix any $(x_{\mathrm{old}}, y_{\mathrm{old}})$ and define

$$f(\theta) = \log p_\theta(y_{\mathrm{old}} \mid x_{\mathrm{old}}).$$

By Taylor's theorem, for some point $\tilde{\theta}$ on the line segment joining $\theta_i$ and $\theta_{i+1}$,

$$f(\theta_{i+1}) - f(\theta_i) = \nabla_\theta f(\theta_i)^\top (\theta_{i+1} - \theta_i) + \frac{1}{2} (\theta_{i+1} - \theta_i)^\top \nabla^2_\theta f(\tilde{\theta}) (\theta_{i+1} - \theta_i).$$

We take the bounded parameter region in Assumption 1 to contain the line segments between consecutive iterates, so Lemma 1 also applies at $\tilde{\theta}$. Substituting the update rule,

$$\log \frac{p_{i+1}(y_{\text{old}} \mid x_{\text{old}})}{p_i(y_{\text{old}} \mid x_{\text{old}})} = -\eta_i \nabla_\theta \log p_{\theta_i}(y_{\text{old}} \mid x_{\text{old}})^\top g_i^{\mathcal{B}} + \frac{\eta_i^2}{2} (g_i^{\mathcal{B}})^\top \nabla_{\tilde{\theta}}^2 \log p_{\tilde{\theta}}(y_{\text{old}} \mid x_{\text{old}}) g_i^{\mathcal{B}}.$$

Taking absolute values and using Cauchy–Schwarz gives

$$\left| \log \frac{p_{i+1}(y_{\text{old}} \mid x_{\text{old}})}{p_i(y_{\text{old}} \mid x_{\text{old}})} \right| \leq \eta_i \|\nabla_\theta \log p_{\theta_i}(y_{\text{old}} \mid x_{\text{old}})\|_2 \|g_i^{\mathcal{B}}\|_2 + \frac{\eta_i^2}{2} \|\nabla_{\tilde{\theta}}^2 \log p_{\tilde{\theta}}(y_{\text{old}} \mid x_{\text{old}})\|_{\text{op}} \|g_i^{\mathcal{B}}\|_2^2.$$

By Lemma 1, it remains to bound $\|g_i^{\mathcal{B}}\|_2$ in terms of the mini-batch loss. We first bound the gradient of the loss for a single sequence $s = (x_{1:T_s}, y_{1:T_s})$. By definition,

$$\ell_s(\theta) = \frac{1}{T_s} \sum_{t=1}^{T_s} \mathrm{CE}\big(q_t(\cdot \mid x_{<t}) \| p_\theta(\cdot \mid x_{<t})\big),$$

hence

$$\nabla_\theta \ell_s(\theta_i) = \frac{1}{T_s} \sum_{t=1}^{T_s} \nabla_\theta \mathrm{CE}\big(q_t(\cdot \mid x_{<t}) \| p_{\theta_i}(\cdot \mid x_{<t})\big).$$

For each token position $t$, the gradient of the token-level cross-entropy is

$$\nabla_\theta \mathrm{CE}\big(q_t(\cdot \mid x_{<t}) \| p_{\theta_i}(\cdot \mid x_{<t})\big) = J_{\theta_i}(x_{<t})^\top \big(p_{\theta_i}(\cdot \mid x_{<t}) - q_t(\cdot \mid x_{<t})\big),$$

where $J_{\theta_i}(x_{<t})$ is the Jacobian of the logits at context $x_{<t}$ with respect to $\theta$. Therefore,

$$\big\|\nabla_\theta \mathrm{CE}\big(q_t(\cdot \mid x_{<t}) \| p_{\theta_i}(\cdot \mid x_{<t})\big)\big\|_2 \leq \|J_{\theta_i}(x_{<t})\|_{\text{op}} \, \|p_{\theta_i}(\cdot \mid x_{<t}) - q_t(\cdot \mid x_{<t})\|_2.$$

Using $\|v\|_2 \leq \|v\|_1$, Pinsker's inequality, and

$$\mathrm{KL}\big(q_t(\cdot \mid x_{<t}) \| p_{\theta_i}(\cdot \mid x_{<t})\big) \leq \mathrm{CE}\big(q_t(\cdot \mid x_{<t}) \| p_{\theta_i}(\cdot \mid x_{<t})\big),$$

we get

$$\big\|\nabla_\theta \mathrm{CE}\big(q_t(\cdot \mid x_{<t}) \| p_{\theta_i}(\cdot \mid x_{<t})\big)\big\|_2 \leq \|J_{\theta_i}(x_{<t})\|_{\text{op}} \sqrt{2 \, \mathrm{CE}\big(q_t(\cdot \mid x_{<t}) \| p_{\theta_i}(\cdot \mid x_{<t})\big)}.$$

By Lemma 1, $\|J_{\theta_i}(x_{<t})\|_{\text{op}} \leq M$, so

$$\big\|\nabla_\theta \mathrm{CE}\big(q_t(\cdot \mid x_{<t}) \| p_{\theta_i}(\cdot \mid x_{<t})\big)\big\|_2 \leq M \sqrt{2 \, \mathrm{CE}\big(q_t(\cdot \mid x_{<t}) \| p_{\theta_i}(\cdot \mid x_{<t})\big)}.$$

Now apply the triangle inequality across token positions:

$$\|\nabla_\theta \ell_s(\theta_i)\|_2 \leq \frac{1}{T_s} \sum_{t=1}^{T_s} \big\|\nabla_\theta \mathrm{CE}\big(q_t(\cdot \mid x_{<t}) \| p_{\theta_i}(\cdot \mid x_{<t})\big)\big\|_2 .$$

Combining with the previous bound,

$$\|\nabla_\theta \ell_s(\theta_i)\|_2 \leq \frac{M}{T_s} \sum_{t=1}^{T_s} \sqrt{2 \, \mathrm{CE}\big(q_t(\cdot \mid x_{<t}) \| p_{\theta_i}(\cdot \mid x_{<t})\big)}.$$

Using Jensen's inequality for the concave function $u \mapsto \sqrt{u}$,

$$\frac{1}{T_s} \sum_{t=1}^{T_s} \sqrt{2 \, \mathrm{CE}\big(q_t(\cdot \mid x_{<t}) \| p_{\theta_i}(\cdot \mid x_{<t})\big)} \leq \sqrt{\frac{2}{T_s} \sum_{t=1}^{T_s} \mathrm{CE}\big(q_t(\cdot \mid x_{<t}) \| p_{\theta_i}(\cdot \mid x_{<t})\big)}.$$

Thus

$$\|\nabla_\theta \ell_s(\theta_i)\|_2 \leq M \sqrt{2 \, \ell_s(\theta_i)}.$$

We now lift this bound from a single sequence to the whole mini-batch. Since

$$\mathcal{L}_{\mathcal{B}_i}(\theta) = \frac{1}{|\mathcal{B}_i|} \sum_{s \in \mathcal{B}_i} \ell_s(\theta),$$

its gradient is

$$g_i^{\mathcal{B}} = \nabla_\theta \mathcal{L}_{\mathcal{B}_i}(\theta_i) = \frac{1}{|\mathcal{B}_i|} \sum_{s \in \mathcal{B}_i} \nabla_\theta \ell_s(\theta_i).$$

Applying the triangle inequality across sequences,

$$\|g_i^{\mathcal{B}}\|_2 \le \frac{1}{|\mathcal{B}_i|} \sum_{s \in \mathcal{B}_i} \|\nabla_\theta \ell_s(\theta_i)\|_2 \le \frac{M}{|\mathcal{B}_i|} \sum_{s \in \mathcal{B}_i} \sqrt{2\,\ell_s(\theta_i)}.$$

Applying Jensen's inequality once more,

$$\frac{1}{|\mathcal{B}_i|} \sum_{s \in \mathcal{B}_i} \sqrt{2\,\ell_s(\theta_i)} \le \sqrt{\frac{2}{|\mathcal{B}_i|} \sum_{s \in \mathcal{B}_i} \ell_s(\theta_i)} = \sqrt{2\,\mathcal{L}_{\mathcal{B}_i}(\theta_i)}.$$

Therefore,

$$\|g_i^{\mathcal{B}}\|_2 \le M \sqrt{2\,\mathcal{L}_{\mathcal{B}_i}(\theta_i)}.$$

Substituting this bound into the Taylor estimate and using Lemma 1 again, we obtain

$$\left| \log \frac{p_{i+1}(y_{\text{old}} \mid x_{\text{old}})}{p_i(y_{\text{old}} \mid x_{\text{old}})} \right| \le C_1\, \eta_i \sqrt{\mathcal{L}_{\mathcal{B}_i}(\theta_i)} + C_2\, \eta_i^2\, \mathcal{L}_{\mathcal{B}_i}(\theta_i),$$

for constants $C_1, C_2 > 0$ independent of $i$, $(x_{\text{old}}, y_{\text{old}})$, and the mini-batch.

Taking the supremum over $(x_{\text{old}}, y_{\text{old}})$ gives

$$\Delta\mathcal{L}_{\text{old}}(p_i) \le C_1\, \eta_i \sqrt{\mathcal{L}_{\mathcal{B}_i}(\theta_i)} + C_2\, \eta_i^2\, \mathcal{L}_{\mathcal{B}_i}(\theta_i).$$

It remains only to justify the simplified big-$O$ form. Assumption 1 implies that the logits are uniformly bounded, hence the softmax probabilities are bounded away from zero. Therefore the token-level cross-entropies, sequence losses, and mini-batch losses are uniformly bounded above: there exists $L_{\max} < \infty$ such that

$$\mathcal{L}_{\mathcal{B}_i}(\theta_i) \le L_{\max}$$

for all $i$. Hence

$$\eta_i^2 \mathcal{L}_{\mathcal{B}_i}(\theta_i) = \eta_i \sqrt{\mathcal{L}_{\mathcal{B}_i}(\theta_i)} \cdot \eta_i \sqrt{\mathcal{L}_{\mathcal{B}_i}(\theta_i)} \le \eta_i \sqrt{\mathcal{L}_{\mathcal{B}_i}(\theta_i)} \cdot \eta_i \sqrt{L_{\max}}.$$

Thus, for sufficiently small $\eta_i$, the second-order term is absorbed into the first-order term, and

$$\Delta\mathcal{L}_{\text{old}}(p_i) = O\Big(\eta_i \sqrt{\mathcal{L}_{\mathcal{B}_i}(\theta_i)}\Big).$$

□

### A.3 Proof of Corollary 1

**Corollary 1.** *Under the conditions of Theorem 1, suppose the learning rates are sufficiently small and*

$$\eta_i = \frac{\kappa}{\sqrt{\mathcal{L}_{\mathcal{B}_i}(\theta_i)}}.$$

*Then the cumulative forgetting satisfies*

$$\mathcal{L}_{\text{old}}(p_T) - \mathcal{L}_{\text{old}}(p_0) = O(T\kappa).$$

*Proof.* Since the learning rates are sufficiently small, Theorem 1 gives

$$\Delta\mathcal{L}_{\text{old}}(p_i) = O\Big(\eta_i \sqrt{\mathcal{L}_{\mathcal{B}_i}(\theta_i)}\Big).$$

Substituting

$$\eta_i = \frac{\kappa}{\sqrt{\mathcal{L}_{\mathcal{B}_i}(\theta_i)}}$$

gives

$$\Delta\mathcal{L}_{\text{old}}(p_i) = O(\kappa)$$

for each step $i$. Summing over $i = 0, \ldots, T-1$,

$$\mathcal{L}_{\text{old}}(p_T) - \mathcal{L}_{\text{old}}(p_0) = \sum_{i=0}^{T-1} \Delta\mathcal{L}_{\text{old}}(p_i) = O(T\kappa).$$

□

# B Experimental Details

## B.1 Tasks

We evaluate FINCH on three settings where the pretrained model does not have good performance and a large fraction of task-relevant tokens are hard:

- *Language adaptation (LA):* Instruction following in Galician, a low-resource language with limited pretraining coverage. We use Galician Alpaca, a translated version of the Stanford Alpaca instruction-following dataset (Taori et al., 2023; Chen et al., 2024b). We translate instructions and outputs separately using GPT-5.2 with the following prompt:

  "Translate the following text from English to Galician.
  Requirements:
  – Keep the exact meaning.
  – Keep formatting and punctuation where possible.
  – Return only the translated text.
  Text: {text}"

  We use a 70/5/25 train/validation/test split.
- *Science.* Undergraduate-level scientific reasoning using the Chemistry L-3 subset of SciKnowEval (Feng et al., 2024), a multiple-choice benchmark covering chemistry concepts. Solutions were obtained from Shenfeld et al. (2026) that used gpt-4o to generate solutions. We use a 70/5/25 train/validation/test split.
- *Knowledge acquisition (KA).* Acquiring novel factual information using TOFU (Maini et al., 2024), a dataset of 200 synthetic author profiles each consisting of 20 question–answer pairs probing biographical facts. We train and validate on a 90/10 split of the full set of 4,000 question–answer pairs. For evaluation, we use a subset of 400 paraphrased questions paired with both a correct answer and several perturbed (incorrect) answers, which allows us to compute multiple-choice accuracy.

## B.2 Hyperparameters

All models are optimized with AdamW. For all baselines, we use a cosine learning rate schedule with a linear warmup over the first 5% of training steps, train for up to 5 epochs, and select the checkpoint with the best validation performance on the target task. We clip gradients to a maximum norm of 1.0 in all main experiments. We test smaller clipping thresholds in Section B.5.

We sweep over learning rates in $\{5 \times 10^{-6},\ 1 \times 10^{-5},\ 2 \times 10^{-5},\ 3 \times 10^{-5},\ 5 \times 10^{-5},\ 1 \times 10^{-4}\}$ and batch sizes in $\{16, 32\}$ for all methods except LoRA, for which we follow Sanyal et al. (2025) and sweep over $\{2 \times 10^{-4},\ 2 \times 10^{-3},\ 2 \times 10^{-2},\ 2 \times 10^{-1}\}$.

For the method-specific hyperparameters, we follow Sanyal et al. (2025) when applicable. For WiSE-FT (Wortsman et al., 2022), we set the interpolation coefficient to $\alpha = 0.5$, and for L2 regularization (Kirkpatrick et al., 2017), we set $\lambda = 10^{-3}$. For STM (Wu et al., 2025a), we use a perplexity threshold of 2.5; larger thresholds improve target-task accuracy but lead to forgetting comparable to standard SFT. For SFT (small lr) (Lin et al., 2025), the original paper does not specify a general rule for choosing the small learning rate, so we choose the largest learning rate whose average forgetting is less than 2 percentage points on the validation benchmarks, giving the strongest target-task performance under a low-forgetting constraint.

Experiments were run on H200 GPUs. One node with eight H200s was available; individual runs used up to four H200s, commonly one or four.

## B.3 Evaluation Metrics

**Average old-task change.** For each method, Avg. $\Delta$ is the mean change in performance relative to the pretrained model across HellaSwag, WinoGrande, IFEval, and MMLU:

$$\text{Avg. } \Delta = \frac{1}{4} \sum_{b \in \{\text{HS,WG,IFEval,MMLU}\}} \left( \text{Acc}_b^{\text{ft}} - \text{Acc}_b^{\text{base}} \right).$$

Higher values indicate better preservation of general capabilities.

### B.4 Evaluation Prompts

**Galician win-tie judge prompt.** For low-resource language adaptation, we evaluate Galician language quality using pairwise head-to-head judging. For each example, we randomly swap the order of the two model responses to reduce positional bias. The judge is instructed to compare only the Galician language quality of the two responses, ignoring factual correctness, helpfulness, and content quality. The exact judge instruction is:

**Galician Language Quality Judge Prompt**

Please act as an impartial judge and evaluate the Galician language quality of the responses provided by two AI assistants to the user prompt below. You will be given assistant A's answer and assistant B's answer.
Evaluate each response on the following criteria:

1. Is the response written in Galician (not Spanish, Portuguese, English, or other languages)?
2. How natural and fluent is the Galician? Does it sound like a native speaker, or does it have Spanish/Portuguese interference?
3. How consistent is the Galician throughout the response—does it code-switch mid-response?

After your brief explanation, you must output only one of the following choices as your final verdict with a label:

1. Assistant A is significantly better: `[[A>>B]]`
2. Assistant A is slightly better: `[[A>B]]`
3. Tie, relatively the same: `[[A=B]]`
4. Assistant B is slightly better: `[[B>A]]`
5. Assistant B is significantly better: `[[B>>A]]`

Example output: "My final verdict is tie: `[[A=B]]`".

### B.5 What if we have a small maximum gradient norm?

We also test whether forgetting can be mitigated by simply reducing the maximum gradient norm during fine-tuning. Table 2 reports this control on knowledge acquisition. Reducing the clipping threshold gives only a weak trade-off. Moderate clipping thresholds, such as $0.1$ or $0.01$, preserve target-task accuracy but barely improve forgetting; stronger clipping improves Avg. $\Delta$ mainly by substantially hurting target-task accuracy. Thus, small gradient clipping does not help.

Table 2: Effect of reducing the maximum gradient norm on knowledge acquisition. Task Acc. reports held-out multiple-choice accuracy on author-profile questions, and Avg. $\Delta$ is the average change on old-task benchmarks.

| Model | Max grad norm | Task Acc. ↑ | Avg. Δ ↑ |
|---|---|---|---|
| Qwen3-4B | 1 | 83.5 | −9.6 |
| | 0.1 | 82.3 | −8.4 |
| | 0.01 | 81.8 | −8.3 |
| | 0.001 | 71.8 | −7.1 |
| | 0.0001 | 58.5 | −4.6 |
| Llama-3-8B | 1 | 83.3 | −9.0 |
| | 0.1 | 83.0 | −10.1 |
| | 0.01 | 82.5 | −8.5 |
| | 0.001 | 78.0 | −3.6 |
| | 0.0001 | 62.8 | −0.4 |

### B.6 Additional Results for Section 4

Table 4 and Table 5 provide the results for Science and LA respectively. The following appendix tables give the per-benchmark results for all four models on KA, Science, and LA.

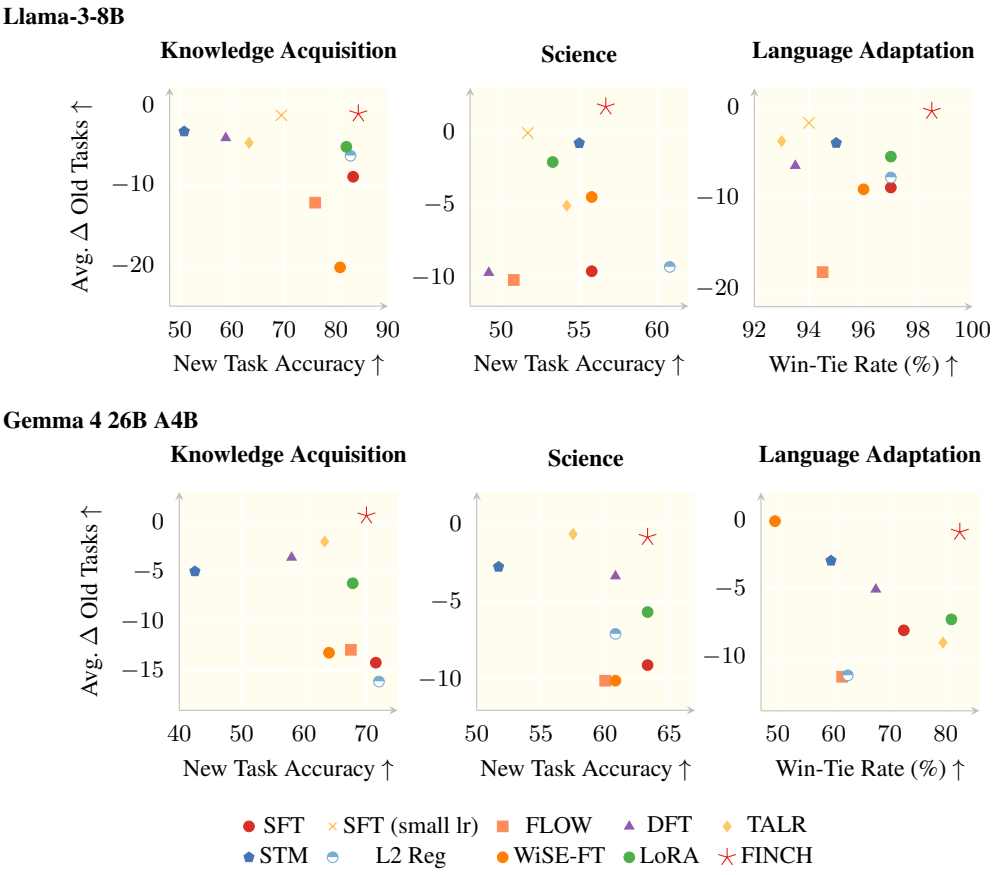



Figure 4: Target-task performance versus average old-task change on Llama-3-8B (top row) and Gemma 4 26B A4B (bottom row). The point colors and markers follow Figure 2.

Table 3: Target-task adaptation and forgetting for KA across four models. Task Acc. reports held-out multiple-choice accuracy on author-profile questions. HellaSwag (HS), WinoGrande (WG), IFEval, and MMLU report absolute performance changes relative to the pretrained model, and Avg. $\Delta$ is their mean. Higher values indicate better preservation of general capabilities.

| Model | Method | Task Acc. ↑ | HS ↑ | WG ↑ | IFEval ↑ | MMLU ↑ | Avg. $\Delta$ ↑ |
|---|---|---|---|---|---|---|---|
| Qwen3-4B | Base | 53.5 | +0.0 | +0.0 | +0.0 | +0.0 | +0.0 |
| | SFT | 83.5 | −7.3 | −2.0 | −9.4 | −19.6 | −9.6 |
| | SFT (small lr) (Lin et al., 2025) | 65.8 | −3.8 | −2.2 | +3.1 | −3.2 | −1.5 |
| | FLOW (Sanyal et al., 2025) | 77.5 | −6.4 | +0.7 | −13.7 | −12.7 | −8.0 |
| | DFT (Wu et al., 2025b) | 55.8 | −2.7 | −2.7 | +3.0 | −1.9 | −1.1 |
| | TALR (Lin et al., 2025) | 62.5 | −1.3 | +1.0 | −4.5 | −5.2 | −2.5 |
| | STM (Wu et al., 2025a) | 53.8 | −1.2 | +1.1 | +1.4 | +0.9 | +0.6 |
| | L2 Reg (Kirkpatrick et al., 2017) | 83.5 | −5.5 | +0.2 | −1.9 | −14.3 | −5.4 |
| | WiSE-FT (Wortsman et al., 2022) | 82.5 | −14.7 | −3.2 | −37.8 | −7.2 | −15.7 |
| | LoRA (Hu et al., 2022) | 82.3 | −4.4 | −6.9 | −9.1 | −9.2 | −7.4 |
| | FINCH | 83.3 | −0.6 | +1.4 | +0.5 | −0.8 | +0.1 |
| Qwen3-32B | Base | 57.8 | +0.0 | +0.0 | +0.0 | +0.0 | +0.0 |
| | SFT | 85.3 | −12.6 | −6.0 | −21.9 | −7.2 | −11.9 |
| | SFT (small lr) (Lin et al., 2025) | 64.3 | −1.6 | −1.4 | −7.1 | +2.7 | −1.8 |
| | FLOW (Sanyal et al., 2025) | 79.8 | −9.9 | −9.0 | −16.2 | −8.9 | −11.0 |
| | DFT (Wu et al., 2025b) | 66.3 | −2.0 | −1.9 | −10.3 | +1.6 | −3.1 |
| | TALR (Lin et al., 2025) | 69.8 | −1.5 | −5.0 | −14.2 | −1.1 | −5.4 |
| | STM (Wu et al., 2025a) | 55.0 | +0.2 | +0.4 | −1.6 | +1.9 | +0.2 |
| | L2 Reg (Kirkpatrick et al., 2017) | 82.8 | −10.3 | −9.8 | −37.5 | −15.2 | −18.2 |
| | WiSE-FT (Wortsman et al., 2022) | 80.8 | −4.7 | −7.5 | −7.1 | −2.6 | −5.5 |
| | LoRA (Hu et al., 2022) | 84.0 | −8.3 | −7.2 | −11.8 | −6.4 | −8.4 |
| | FINCH | 85.5 | −1.9 | −1.4 | −4.7 | +0.0 | −2.0 |
| Llama-3-8B | Base | 52.8 | +0.0 | +0.0 | +0.0 | +0.0 | +0.0 |
| | SFT | 83.3 | −10.4 | −0.3 | −21.7 | −3.6 | −9.0 |
| | SFT (small lr) (Lin et al., 2025) | 69.5 | −0.7 | +0.5 | −6.2 | +0.9 | −1.4 |
| | FLOW (Sanyal et al., 2025) | 76.0 | −8.4 | +2.5 | −36.9 | −6.0 | −12.2 |
| | DFT (Wu et al., 2025b) | 58.8 | −7.8 | +2.4 | −9.0 | −2.2 | −4.2 |
| | TALR (Lin et al., 2025) | 63.3 | −5.5 | +0.6 | −12.3 | −2.0 | −4.8 |
| | STM (Wu et al., 2025a) | 50.8 | −0.4 | +0.0 | −12.3 | −0.7 | −3.4 |
| | L2 Reg (Kirkpatrick et al., 2017) | 82.8 | −10.8 | +0.8 | −11.8 | −3.9 | −6.4 |
| | WiSE-FT (Wortsman et al., 2022) | 80.8 | −20.4 | −3.4 | −50.0 | −7.0 | −20.2 |
| | LoRA (Hu et al., 2022) | 82.0 | −0.9 | −0.6 | −15.7 | −3.8 | −5.3 |
| | FINCH | 84.3 | −4.2 | +2.2 | +0.0 | −2.6 | −1.2 |
| Gemma 4 26B A4B | Base | 41.5 | +0.0 | +0.0 | +0.0 | +0.0 | +0.0 |
| | SFT | 71.5 | −12.5 | −11.7 | −21.3 | −11.5 | −14.2 |
| | FLOW (Sanyal et al., 2025) | 67.5 | −9.0 | −5.8 | −28.1 | −8.7 | −12.9 |
| | DFT (Wu et al., 2025b) | 58.0 | −0.8 | +1.4 | −15.8 | +0.7 | −3.6 |
| | TALR (Lin et al., 2025) | 63.3 | −0.8 | −1.0 | −5.1 | −1.0 | −2.0 |
| | STM (Wu et al., 2025a) | 42.5 | −3.0 | −0.2 | −16.4 | −0.5 | −5.0 |
| | L2 Reg (Kirkpatrick et al., 2017) | 72.0 | −12.0 | −10.2 | −31.9 | −10.4 | −16.1 |
| | WiSE-FT (Wortsman et al., 2022) | 64.0 | −7.2 | −7.4 | −30.6 | −7.6 | −13.2 |
| | LoRA (Hu et al., 2022) | 67.8 | −2.2 | −6.1 | −13.3 | −3.3 | −6.2 |
| | FINCH | 70.0 | +1.9 | +0.9 | −1.8 | +1.3 | +0.6 |

Table 4: Target-task adaptation and forgetting on Science QA across four models. Task Acc. reports held-out multiple-choice accuracy on science questions. HellaSwag (HS), WinoGrande (WG), IFEval, and MMLU report absolute performance changes relative to the pretrained model, and Avg. $\Delta$ is their mean. Higher values indicate better preservation of general capabilities.

| Model | Method | Task Acc. ↑ | HS ↑ | WG ↑ | IFEval ↑ | MMLU ↑ | Avg. $\Delta$ ↑ |
|---|---|---|---|---|---|---|---|
| Qwen3-4B | Base | 29.2 | +0.0 | +0.0 | +0.0 | +0.0 | +0.0 |
| | SFT | 62.5 | −11.2 | +2.5 | −17.6 | −17.3 | −10.9 |
| | SFT (small lr) | 56.7 | −5.3 | −0.4 | +4.9 | +3.1 | +0.6 |
| | FLOW (Sanyal et al., 2025) | 58.3 | −7.0 | −3.0 | −6.4 | −10.0 | −6.6 |
| | DFT (Wu et al., 2025b) | 46.7 | −3.4 | −0.3 | −5.7 | +1.4 | −2.0 |
| | TALR (Lin et al., 2025) | 48.3 | −0.8 | −0.3 | −2.5 | +3.8 | +0.1 |
| | STM (Wu et al., 2025a) | 49.2 | −2.4 | +2.9 | −1.7 | +2.6 | +0.4 |
| | L2 Reg (Kirkpatrick et al., 2017) | 57.5 | −17.7 | −0.9 | −23.9 | −1.3 | −11.0 |
| | WiSE-FT (Wortsman et al., 2022) | 56.7 | −11.9 | +0.5 | −23.7 | −8.9 | −11.0 |
| | LoRA (Hu et al., 2022) | 51.7 | −7.5 | +6.7 | −2.1 | −5.6 | −2.1 |
| | FINCH | 65.0 | −2.9 | +0.3 | −4.4 | +0.6 | −1.6 |
| Qwen3-32B | Base | 50.8 | +0.0 | +0.0 | +0.0 | +0.0 | +0.0 |
| | SFT | 63.3 | −7.2 | −11.4 | −8.5 | −3.4 | −7.6 |
| | SFT (small lr) | 52.5 | −0.2 | −0.2 | +0.0 | +2.5 | +0.5 |
| | FLOW (Sanyal et al., 2025) | 58.3 | −3.5 | −8.2 | −14.1 | −3.5 | −7.3 |
| | DFT (Wu et al., 2025b) | 51.7 | −3.1 | −1.6 | −4.7 | +1.0 | −2.1 |
| | TALR (Lin et al., 2025) | 56.7 | −12.0 | −9.5 | −10.6 | −8.7 | −10.2 |
| | STM (Wu et al., 2025a) | 58.3 | −0.8 | +0.4 | −2.4 | +2.1 | −0.2 |
| | L2 Reg (Kirkpatrick et al., 2017) | 65.8 | −24.8 | −24.8 | −14.9 | −17.2 | −20.4 |
| | WiSE-FT (Wortsman et al., 2022) | 56.7 | −0.9 | −9.3 | +1.4 | +1.8 | −1.7 |
| | LoRA (Hu et al., 2022) | 54.1 | −0.9 | −3.5 | +1.2 | +2.0 | −0.3 |
| | FINCH | 64.2 | −0.4 | −0.2 | +1.5 | +3.1 | +1.0 |
| Llama-3-8B | Base | 38.3 | +0.0 | +0.0 | +0.0 | +0.0 | +0.0 |
| | SFT | 55.8 | −10.8 | +0.9 | −23.3 | −5.2 | −9.6 |
| | SFT (small lr) | 51.7 | +1.7 | +1.9 | −5.2 | +1.1 | −0.1 |
| | FLOW (Sanyal et al., 2025) | 50.8 | −13.5 | +4.7 | −22.8 | −9.1 | −10.2 |
| | DFT (Wu et al., 2025b) | 49.2 | −8.6 | −0.2 | −18.9 | −11.0 | −9.7 |
| | TALR (Lin et al., 2025) | 54.2 | −7.9 | −0.3 | −7.4 | −4.9 | −5.1 |
| | STM (Wu et al., 2025a) | 55.0 | −5.3 | +6.9 | −4.0 | −0.8 | −0.8 |
| | L2 Reg (Kirkpatrick et al., 2017) | 60.8 | −14.6 | +1.8 | −20.6 | −3.6 | −9.3 |
| | WiSE-FT (Wortsman et al., 2022) | 55.8 | −7.7 | −0.9 | −8.1 | −1.3 | −4.5 |
| | LoRA (Hu et al., 2022) | 53.3 | −5.8 | +6.6 | −2.9 | −6.3 | −2.1 |
| | FINCH | 56.7 | +1.2 | +2.3 | +1.6 | +1.6 | +1.7 |
| Gemma 4 26B A4B | Base | 53.3 | +0.0 | +0.0 | +0.0 | +0.0 | +0.0 |
| | SFT | 63.3 | −9.0 | −8.6 | −10.2 | −8.6 | −9.1 |
| | FLOW (Sanyal et al., 2025) | 60.0 | −12.5 | −0.1 | −4.3 | −23.5 | −10.1 |
| | DFT (Wu et al., 2025b) | 60.8 | −6.3 | −2.9 | −3.9 | −0.5 | −3.4 |
| | TALR (Lin et al., 2025) | 57.5 | −1.0 | −1.3 | −0.8 | +0.2 | −0.7 |
| | STM (Wu et al., 2025a) | 51.7 | −4.5 | −1.5 | −3.9 | −1.3 | −2.8 |
| | L2 Reg (Kirkpatrick et al., 2017) | 60.8 | −10.0 | −3.1 | −6.1 | −9.1 | −7.1 |
| | WiSE-FT (Wortsman et al., 2022) | 60.8 | −14.3 | −3.5 | −4.7 | −17.9 | −10.1 |
| | LoRA (Hu et al., 2022) | 63.3 | −7.4 | −5.4 | −6.5 | −3.5 | −5.7 |
| | FINCH | 63.3 | −0.8 | −1.1 | −1.6 | +0.0 | −0.9 |

Table 5: Target-task adaptation and forgetting on Galician across four models. Win-Tie reports the tie-adjusted head-to-head rate against the pretrained model. HellaSwag (HS), WinoGrande (WG), IFEval, and MMLU report absolute performance changes relative to the pretrained model, and Avg. $\Delta$ is their mean. Higher values indicate better preservation of general capabilities.

| Model | Method | Win-Tie ↑ | HS ↑ | WG ↑ | IFEval ↑ | MMLU ↑ | Avg. $\Delta$ ↑ |
|---|---|---|---|---|---|---|---|
| Qwen3-4B | Base | 50.0% | +0.0 | +0.0 | +0.0 | +0.0 | +0.0 |
| | SFT | 89.0% | −10.0 | −11.6 | −25.1 | −10.4 | −14.3 |
| | SFT (small lr) | 77.5% | +0.1 | −2.4 | −0.1 | −0.4 | −0.7 |
| | FLOW (Sanyal et al., 2025) | 90.0% | −15.5 | −8.7 | −15.6 | −9.4 | −12.3 |
| | DFT (Wu et al., 2025b) | 80.0% | +2.6 | −0.6 | +5.2 | +2.3 | +2.4 |
| | TALR (Lin et al., 2025) | 77.5% | +0.3 | +0.4 | −3.5 | −0.8 | −0.9 |
| | STM (Wu et al., 2025a) | 81.5% | −0.8 | +0.2 | −4.1 | +0.8 | −1.0 |
| | L2 Reg (Kirkpatrick et al., 2017) | 92.0% | −12.4 | −11.7 | −39.2 | −11.2 | −18.6 |
| | WiSE-FT (Wortsman et al., 2022) | 85.5% | −16.3 | −9.6 | −26.6 | −7.8 | −15.1 |
| | LoRA (Hu et al., 2022) | 90.5% | −10.8 | +5.1 | −45.6 | −2.1 | −13.4 |
| | FINCH | 90.0% | −1.6 | −2.7 | +0.0 | −5.1 | −2.4 |
| Qwen3-32B | Base | 50.0% | +0.0 | +0.0 | +0.0 | +0.0 | +0.0 |
| | SFT | 99.0% | −7.4 | −7.3 | −17.2 | −2.6 | −8.6 |
| | SFT (small lr) | 94.0% | +0.4 | +0.2 | −3.9 | +1.7 | −0.4 |
| | FLOW (Sanyal et al., 2025) | 97.5% | −8.4 | −16.4 | −12.5 | −3.5 | −10.2 |
| | DFT (Wu et al., 2025b) | 96.0% | −0.2 | −3.9 | −8.6 | +0.4 | −3.1 |
| | TALR (Lin et al., 2025) | 96.0% | −0.2 | −1.6 | −14.9 | +1.5 | −3.8 |
| | STM (Wu et al., 2025a) | 95.5% | −2.9 | −1.4 | −11.7 | +1.9 | −3.5 |
| | L2 Reg (Kirkpatrick et al., 2017) | 99.5% | −0.8 | −5.7 | −18.8 | +0.2 | −6.3 |
| | WiSE-FT (Wortsman et al., 2022) | 99.0% | −0.6 | −3.2 | −10.2 | −1.5 | −3.9 |
| | LoRA (Hu et al., 2022) | 98.5% | −3.9 | −5.9 | −13.5 | −2.6 | −6.5 |
| | FINCH | 99.0% | +0.2 | −0.8 | −2.4 | +2.1 | −0.2 |
| Llama-3-8B | Base | 50.0% | +0.0 | +0.0 | +0.0 | +0.0 | +0.0 |
| | SFT | 97.0% | −12.2 | −0.8 | −18.3 | −4.2 | −8.9 |
| | SFT (small lr) | 94.0% | −3.6 | +0.2 | −4.1 | +0.4 | −1.8 |
| | FLOW (Sanyal et al., 2025) | 94.5% | −31.8 | −5.2 | −22.5 | −13.2 | −18.2 |
| | DFT (Wu et al., 2025b) | 93.5% | −5.1 | −3.4 | −12.5 | −5.0 | −6.5 |
| | TALR (Lin et al., 2025) | 93.0% | −6.4 | −0.4 | −8.3 | +0.0 | −3.8 |
| | STM (Wu et al., 2025a) | 95.0% | −4.2 | −1.4 | −8.3 | −2.0 | −4.0 |
| | L2 Reg (Kirkpatrick et al., 2017) | 97.0% | −12.2 | −1.8 | −13.3 | −4.0 | −7.8 |
| | WiSE-FT (Wortsman et al., 2022) | 96.0% | −12.6 | −2.0 | −16.7 | −5.0 | −9.1 |
| | LoRA (Hu et al., 2022) | 97.0% | −4.6 | −2.2 | −14.2 | −1.0 | −5.5 |
| | FINCH | 98.5% | −1.2 | −0.6 | +0.0 | +0.0 | −0.5 |
| Gemma 4 26B A4B | Base | 50.0% | +0.0 | +0.0 | +0.0 | +0.0 | +0.0 |
| | SFT | 72.5% | −3.0 | −2.2 | −23.9 | −3.4 | −8.1 |
| | FLOW (Sanyal et al., 2025) | 61.5% | +1.0 | −2.9 | −41.4 | −2.7 | −11.5 |
| | DFT (Wu et al., 2025b) | 67.5% | +2.4 | −1.0 | −21.3 | −0.4 | −5.1 |
| | TALR (Lin et al., 2025) | 79.5% | −4.3 | −1.2 | −28.9 | −1.6 | −9.0 |
| | STM (Wu et al., 2025a) | 59.5% | +0.4 | −1.4 | −10.9 | −0.2 | −3.0 |
| | L2 Reg (Kirkpatrick et al., 2017) | 62.5% | −0.2 | −2.0 | −39.8 | −3.5 | −11.4 |
| | WiSE-FT (Wortsman et al., 2022) | 49.5% | +0.2 | −0.2 | −0.8 | +0.2 | −0.1 |
| | LoRA (Hu et al., 2022) | 81.0% | −2.7 | −2.3 | −21.9 | −2.3 | −7.3 |
| | FINCH | 82.5% | +0.6 | −0.4 | −3.9 | +0.0 | −0.9 |

**Computing the average forgetting reduction.** We compute the reported forgetting reduction by aggregating the signed Avg. $\Delta$ values across all twelve model–task settings (four models and three tasks) in Tables 3–5. Let $\Delta_{m,t}$ denote Avg. $\Delta$ for method $m$ in setting $t$. Since negative values indicate degradation and positive values indicate improvement, we define net degradation as the negative signed sum:

$$\mathrm{Deg}(m) = -\sum_{t=1}^{12} \Delta_{m,t}.$$

For standard SFT, this gives

$$\mathrm{Deg}(\mathrm{SFT}) = 121.8,$$

whereas for FINCH,

$$\mathrm{Deg}(\mathrm{FINCH}) = 6.3.$$

Dividing the corresponding signed sums by twelve gives mean Avg. $\Delta$ values of approximately $-10.2$ and $-0.5$ points, respectively. Thus, the relative reduction in net degradation is

$$1 - \frac{\mathrm{Deg}(\mathrm{FINCH})}{\mathrm{Deg}(\mathrm{SFT})} = 1 - \frac{6.3}{121.8} \approx 94.83\%.$$

We conservatively report this as $94\%$ in the main text by rounding down to a whole percentage point.

**IFEval drops.** We observe that many fine-tuning baselines incur relatively large drops on IFEval. One possible reason is that IFEval differs from the other old-task benchmarks: unlike HellaSwag, WinoGrande, and MMLU, which are multiple-choice evaluations, IFEval requires long-form instruction following and is therefore sensitive to changes in generation behavior. In particular, degradation may shorten model responses and reduce compliance with formatting or constraint-following requirements. For example, on Qwen3-4B after KA fine-tuning with standard SFT, the selected epoch produces much shorter IFEval outputs than the pretrained model: average output length drops from $1720.6$ to $572.4$ characters, and average completion length drops from $1066.4$ to $421.8$ tokens. This suggests that part of the IFEval degradation may reflect degraded long-form instruction following rather than only loss of benchmark knowledge.

### B.6.1 Sequential Fine-Tuning

Starting from pretrained Qwen3-4B, we fine-tune successively on Science, Galician language adaptation (LA), and KA, carrying the model from one stage into the next. We run this sequence separately with SFT and FINCH. After each stage, we evaluate Science accuracy, LA win-tie rate, and KA accuracy (Table 6), so the table shows both learning of the current task and retention of tasks learned at earlier stages.

Table 6: Sequential fine-tuning on Qwen3-4B (Science $\rightarrow$ LA $\rightarrow$ KA). Each row follows one target capability through the stages.

| Method | Evaluation | Pretrained | After Science | After LA | After KA |
|---|---|---|---|---|---|
| SFT | Science Acc. | 29.2 | 62.5 | 50.8 | 52.5 |
| | LA Win-Tie | 50.0 | 52.5 | 92.0 | 89.5 |
| | KA Acc. | 53.5 | 53.3 | 56.3 | 80.8 |
| FINCH | Science Acc. | 29.2 | 65.0 | 63.8 | 64.2 |
| | LA Win-Tie | 50.0 | 51.5 | 92.5 | 92.0 |
| | KA Acc. | 53.5 | 53.5 | 54.8 | 81.3 |

### B.7 Experiments for Section 6

#### B.7.1 Mathematical Reasoning

We start with a 10K subset of OpenR1-Math-220k (Open R1 Team, 2025), remove training examples that share an 8-gram with MATH-500 (OpenAI, 2023), and fine-tune Llama-3-8B-Instruct on the remaining examples. We evaluate MATH-500 accuracy and average old-task change over the four benchmarks in Section 4. Table 7 reports means and standard deviations over three seeds. To compare initial token difficulty on the same pretrained model, we measure the fraction of tokens with log probability below $-5$: $23.8\%$ on KA and $3.1\%$ on MATH-500.

Table 7: Math fine-tuning on Llama-3-8B-Instruct: MATH-500 target accuracy and signed average old-task change. Values are mean (standard deviation) over three seeds, except for the pretrained base.

| **Method** | **Target Acc.** $\uparrow$ | **Avg.** $\Delta \uparrow$ |
|---|---|---|
| Base | 28.6 | 0.0 |
| SFT | 33.6 (1.0) | $-4.8$ (3.4) |
| SFT (small lr) | 30.1 (0.3) | $-0.7$ (2.5) |
| FLOW | 33.8 (0.3) | $-0.4$ (1.4) |
| DFT | 31.2 (1.4) | $+0.4$ (2.2) |
| TALR | 30.9 (1.9) | $+1.3$ (3.3) |
| STM | 31.7 (1.8) | $+0.7$ (1.5) |
| L2-Reg | 30.9 (1.9) | $-5.2$ (2.0) |
| WiSE-FT | 32.3 (2.9) | $-4.2$ (3.0) |
| FINCH | 33.3 (0.5) | $-0.1$ (2.0) |

#### B.7.2 Initially Difficult Knowledge

For Qwen3-4B KA, we first keep validation examples answered incorrectly by the pretrained model. We then rank them by the gap between the pretrained model's perplexity for the correct answer and for its preferred incorrect answer, selecting the 100 largest gaps. We evaluate every fine-tuned method on this same fixed subset (Table 8); the base model's accuracy is zero by construction.

Table 8: Accuracy on 100 Qwen3-4B KA validation examples selected by the pretrained model's perplexity gap between correct and preferred incorrect answers.

| **Method** | **Accuracy (%)** $\uparrow$ |
|---|---|
| Base | 0 |
| SFT | 58 |
| FLOW | 44 |
| FINCH | 59 |

### B.7.3 Variation in Batch Loss

During a Qwen3-4B KA run of FINCH, we hold the model fixed at each checkpoint and evaluate the loss on 100 random batches of size 32. Table 9 gives the mean and standard deviation across those batches at steps $0, 100, \ldots, 1000$. At every checkpoint, the standard deviation is less than $19\%$ of the mean, while the mean falls from $3.223$ to $0.137$ over training. In this experiment, the change as the model learns is much larger than variation between batches at a fixed model state.

Table 9: Loss over 100 random batches of size 32 at fixed checkpoints during a Qwen3-4B KA run of FINCH.

| Step | Mean loss | Std. dev. |
|---|---|---|
| 0 | 3.223 | 0.144 |
| 100 | 1.667 | 0.087 |
| 200 | 1.467 | 0.082 |
| 300 | 1.288 | 0.070 |
| 400 | 1.087 | 0.085 |
| 500 | 0.853 | 0.079 |
| 600 | 0.595 | 0.050 |
| 700 | 0.450 | 0.042 |
| 800 | 0.290 | 0.038 |
| 900 | 0.215 | 0.028 |
| 1000 | 0.137 | 0.025 |

### B.7.4 Warmup Duration

On Qwen3-4B KA, we train SFT with linear warmup over $1\%$, $2\%$, $5\%$, $10\%$, or $20\%$ of steps, followed in each case by cosine decay. The $5\%$ schedule is the default used for the baselines. Table 10 compares target accuracy and Avg. $\Delta$ for these schedules with FINCH.

Table 10: Warmup duration on Qwen3-4B KA. The baseline uses a cosine schedule after warmup; FINCH adapts the learning rate to the loss.

| Method | Warmup | Task Acc. ↑ | Avg. $\Delta$ ↑ |
|---|---|---|---|
| Base | — | 53.5 | 0.0 |
| SFT | 1% | 83.0 | −8.9 |
| SFT | 2% | 81.8 | −6.5 |
| SFT | 5% | 83.5 | −9.6 |
| SFT | 10% | 82.5 | −8.8 |
| SFT | 20% | 82.5 | −6.7 |
| FINCH | Adaptive | 83.3 | +0.1 |

### B.7.5 EMA Coefficient

We use $\alpha = 0.9$ for all four models and all three tasks, without tuning it per setting. On Qwen3-4B KA, we rerun FINCH with $\alpha \in \{0.8, 0.9, 0.95\}$ and compare target accuracy and Avg. $\Delta$ (Table 11). The tested settings yield $83.0$–$83.3\%$ target accuracy and Avg. $\Delta$ between $-0.2$ and $+0.1$. The EMA requires one scalar and no additional forward or backward pass.

Table 11: Sensitivity to the EMA coefficient $\alpha$ on Qwen3-4B KA.

| Method / $\alpha$ | Task Acc. ↑ | Avg. $\Delta$ ↑ |
|---|---|---|
| Base | 53.5 | +0.0 |
| SFT | 83.5 | −9.6 |
| FINCH ($\alpha = 0.8$) | 83.0 | +0.1 |
| FINCH ($\alpha = 0.9$) | 83.3 | +0.1 |
| FINCH ($\alpha = 0.95$) | 83.3 | −0.2 |

### B.7.6 Batch Size

We train FINCH on Qwen3-4B KA with batch sizes of 32, 64, 120, and 256, then measure target accuracy and Avg. $\Delta$ using the same evaluations as in Section 4. Table 12 shows target accuracy of 83.3–84.3% and Avg. $\Delta$ from $-1.8$ to $+1.0$. The trade-off remains similar as larger batches average out more example-level variation in the loss.

Table 12: Sensitivity to batch size on Qwen3-4B KA.

| Batch size | Task Acc. ↑ | Avg. Δ ↑ |
|---|---|---|
| 32 | 83.3 | +0.1 |
| 64 | 84.3 | −1.8 |
| 120 | 83.3 | +1.0 |
| 256 | 83.8 | +0.2 |

### B.7.7 Qwen3-32B Seed Variation

For each of KA, Science, and LA on Qwen3-32B, we rerun SFT and FINCH with three random seeds. Table 13 reports the mean and standard deviation of target-task performance and Avg. $\Delta$ across those reruns; the main tables give the selected point estimates. Target-task differences between the two methods are within seed variation, while their differences in old-task change are larger.

Table 13: Three-seed reruns on Qwen3-32B. Values are mean (standard deviation); LA target is win-tie rate.

| Task | Method | Target ↑ | Avg. Δ ↑ |
|---|---|---|---|
| KA | SFT | 85.4 (0.1) | −11.8 (1.1) |
| KA | FINCH | 85.5 (0.3) | −1.9 (0.4) |
| Science | SFT | 63.2 (1.9) | −9.0 (1.3) |
| Science | FINCH | 64.2 (0.8) | −0.1 (1.2) |
| LA | SFT | 99.3 (0.3) | −7.6 (1.2) |
| LA | FINCH | 99.2 (0.8) | +0.1 (0.8) |

## B.8 Additional Results for Section 5

Table 14 reports the knowledge acquisition results; Figure 5 and Table 15 report Science results; Figure 6 and Table 16 report Galician results.

Table 14: TruthfulQA and HaluEval accuracy changes relative to the pretrained model on knowledge acquisition (Qwen3-4B); higher is better.

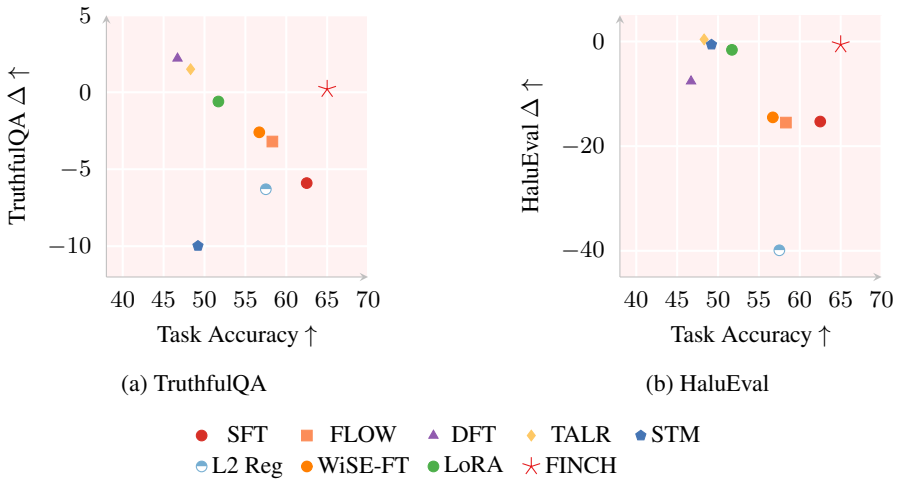
| Model | Method | Task Acc. ↑ | TruthfulQA Δ ↑ | HaluEval Δ ↑ |
|---|---|---|---|---|
| Qwen3-4B | Base | 53.5 | +0.0 | +0.0 |
| | SFT | 83.5 | −13.9 | −9.8 |
| | FLOW (Sanyal et al., 2025) | 77.5 | −11.7 | −8.2 |
| | DFT (Wu et al., 2025b) | 55.8 | −4.9 | +2.7 |
| | TALR (Lin et al., 2025) | 62.5 | −5.4 | +5.1 |
| | STM (Wu et al., 2025a) | 53.8 | −0.7 | +3.7 |
| | L2 Reg (Kirkpatrick et al., 2017) | 83.5 | −12.0 | −1.2 |
| | WiSE-FT (Wortsman et al., 2022) | 82.5 | −19.9 | −7.8 |
| | LoRA (Hu et al., 2022) | 82.3 | −16.4 | −4.3 |
| | FINCH | 83.3 | −2.6 | +3.3 |

Table 15: TruthfulQA and HaluEval accuracy changes relative to the pretrained model on Science (Qwen3-4B); higher is better.

| Model | Method | Task Acc. ↑ | HaluEval Δ ↑ | TruthfulQA Δ ↑ |
|---|---|---|---|---|
| Qwen3-4B | Base | 29.2 | +0.0 | +0.0 |
| | SFT | 62.5 | −15.3 | −5.9 |
| | FLOW (Sanyal et al., 2025) | 58.3 | −15.5 | −3.2 |
| | DFT (Wu et al., 2025b) | 46.7 | −7.6 | +2.2 |
| | TALR (Lin et al., 2025) | 48.3 | +0.4 | +1.5 |
| | STM (Wu et al., 2025a) | 49.2 | −0.6 | −10.0 |
| | L2 Reg (Kirkpatrick et al., 2017) | 57.5 | −39.9 | −6.3 |
| | WiSE-FT (Wortsman et al., 2022) | 56.7 | −14.5 | −2.6 |
| | LoRA (Hu et al., 2022) | 51.7 | −1.6 | −0.6 |
| | FINCH | 65.0 | −0.6 | +0.2 |

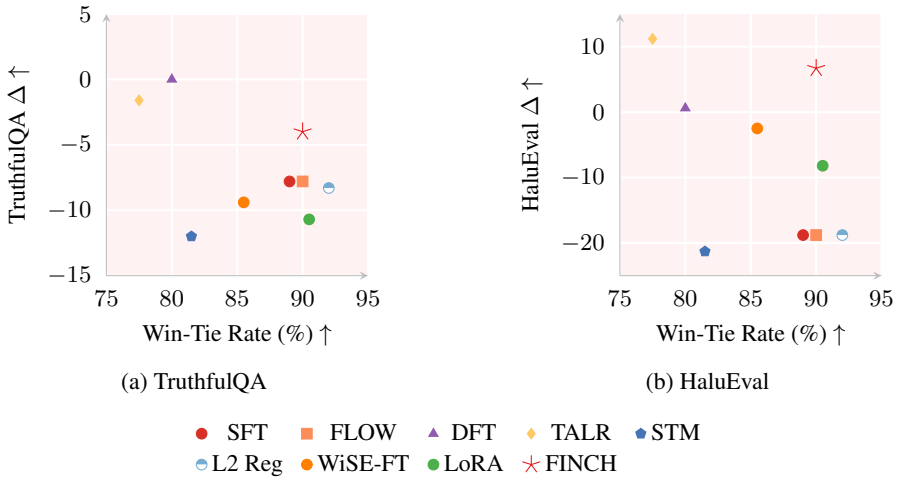
Table 16: TruthfulQA and HaluEval accuracy changes relative to the pretrained model on Galician adaptation (Qwen3-4B); higher is better. Win-Tie is the tie-adjusted head-to-head rate.

| Model | Method | Win-Tie ↑ | HaluEval Δ ↑ | TruthfulQA Δ ↑ |
|---|---|---|---|---|
| Qwen3-4B | Base | — | +0.0 | +0.0 |
| | SFT | 89.0% | −18.8 | −7.8 |
| | FLOW (Sanyal et al., 2025) | 90.0% | −18.8 | −7.8 |
| | DFT (Wu et al., 2025b) | 80.0% | +0.6 | +0.0 |
| | TALR (Lin et al., 2025) | 77.5% | +11.2 | −1.6 |
| | STM (Wu et al., 2025a) | 81.5% | −21.3 | −12.0 |
| | L2 Reg (Kirkpatrick et al., 2017) | 92.0% | −18.8 | −8.3 |
| | WiSE-FT (Wortsman et al., 2022) | 85.5% | −2.5 | −9.4 |
| | LoRA (Hu et al., 2022) | 90.5% | −8.2 | −10.7 |
| | FINCH | 90.0% | +6.7 | −4.0 |




(a) TruthfulQA (b) HaluEval

Figure 5: Truthfulness and hallucination detection vs. task accuracy on Science (Qwen3-4B).




(a) TruthfulQA (b) HaluEval

Figure 6: Truthfulness and hallucination detection vs. win-tie rate on Galician (Qwen3-4B).

# C Extended Related Work

## C.1 Catastrophic Forgetting

Catastrophic forgetting refers to the degradation of previously acquired knowledge when a model is trained on new data (McCloskey & Cohen, 1989; French, 1999). We organize existing mitigation strategies into three groups.

**Methods with Access to Pretraining Data** Replay mixes old training data into fine-tuning and is the most effective mitigation strategy (Rolnick et al., 2019; de Masson D'Autume et al., 2019; Scialom et al., 2022), but pretraining data is rarely available for modern LLMs (Grattafiori et al., 2024; Yang et al., 2025). Some replay methods also use synthetic data when real old data is unavailable (Huang et al., 2024; Chen et al., 2024a; Bansal & Sanghavi, 2025). However, synthetic replay introduces additional generation and filtering cost, depends on the quality and coverage of the generated data, and changes the fine-tuning distribution. Other methods constrain updates to the orthogonal subspace of old task representations (Lin et al., 2022a; Singh et al., 2025) or identify and protect important parameters (Song et al., 2025), but similarly require old data and are evaluated only on small image benchmarks such as MNIST (LeCun et al., 2002) and CIFAR-10 (Krizhevsky et al., 2009). Like all data-oblivious methods, FINCH requires no access to pretraining data.

**Data-Oblivious Methods** Kirkpatrick et al. (2017) regularize weights to stay close to their pretrained values. LoRA (Hu et al., 2022) constrains updates to a low-rank subspace, reducing forgetting but hurting target-domain performance (Biderman et al., 2024). Distillation-based methods (Agarwal et al., 2024; Lu & Lab, 2025) replace SFT with student rollouts scored by a stronger teacher; this is computationally expensive and struggles when the base model assigns low probability to relevant sequences (Shao et al., 2024). A separate line of work reduces forgetting by down-weighting high-loss tokens or sequences: Sanyal et al. (2025) upweight low-loss sequences, Wu et al. (2025a) mask tokens above a perplexity threshold, Lin et al. (2025) scale each token's loss by $p^{1/\tau}$ where $p$ is the token probability and $\tau$ is a constant, and Wu et al. (2025b) rescale gradients by token probability. However, for many tasks learning hard tokens is essential, so suppressing them hurts target performance. FINCH leaves the training objective unchanged and controls forgetting solely through the learning rate.

**Theoretical Analysis** Forgetting has been analyzed theoretically for linear regression (Evron et al., 2022), two-layer CNNs (Li et al., 2025), and linear attention (Lee et al., 2026), but these results are specific to the simple models considered and do not transfer to general LLMs. Li et al. (2024) propose a sharpness-based optimization method to mitigate forgetting. FINCH instead keeps the standard SFT objective and AdamW setup fixed, and controls forgetting through a loss-adaptive learning-rate schedule.

## C.2 Learning Rate

The learning rate determines both the speed and outcome of training, with larger rates tending to find wider, better-generalizing minima (Smith & Topin, 2019) but risking divergence beyond a critical threshold (Lewkowycz et al., 2020). In practice, schedules typically combine linear warmup (Goyal et al., 2017) with cosine decay (Loshchilov & Hutter, 2016), and this pattern is especially common for transformers (Vaswani et al., 2017). The role of warmup has been studied from two complementary angles. Kalra & Barkeshli (2024) argue that small initial learning rates allow the model to move into flatter, better-conditioned regions of the loss landscape before the rate is increased. Kosson et al. (2024) attribute early instability to three compounding factors (bias correction inflation in Adam, large angular parameter updates, and high gradient signal-to-noise ratio) and show that warmup implicitly controls all three.

In the context of catastrophic forgetting, Kenneweg et al. (2022) assign a separate learning rate to each layer in BERT and tune all rates jointly via Bayesian optimization. Scaling this to modern LLMs is infeasible: a 36-layer model such as Qwen3 requires 25 to 30 hyperparameters, the number of required trials scales with dimensionality (Kandasamy et al., 2015), and each trial requires a full fine-tuning run. Lin et al. (2025) prescribe a fixed small learning rate and provide a theoretical justification. They assume each gradient step changes the model by at most $\kappa$ in KL divergence, and

under this assumption approximate the updated model via an exponential tilt; they then bound the total change in old-task loss. The KL constraint is assumed rather than derived, and the exponential-tilt approximation incurs an $O(\kappa)$ error per step, accumulating to $O(T\kappa)$ over $T$ steps. Since a smaller learning rate reduces $\kappa$ but increases $T$, the bound does not improve in the regime they advocate. Our approach derives the learning rate for which the KL constraint holds, and our bound applies without the growing error (Theorem 1). As shown in Section 4, a fixed small learning rate also underperforms empirically, failing to reach competitive target-task accuracy.

Trust-region methods, generally developed for reinforcement learning, also limit update size during training (Schulman et al., 2015; 2017; Zhu et al., 2025). However, these methods are not designed specifically for catastrophic-forgetting mitigation and modify or explicitly constrain the training objective. Moreover, probability-ratio constraints can be conservative for low-probability tokens, which is undesirable in our setting where rare vocabulary, new facts, and low-resource-language tokens must still be learned Qi et al. (2026). FINCH instead keeps the SFT objective unchanged and studies whether the learning-rate schedule alone can control forgetting.