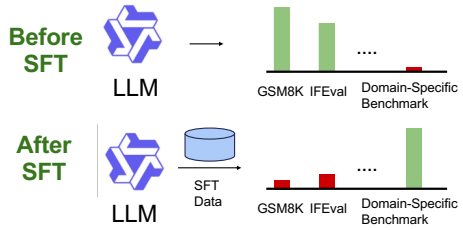
SFT Doesn't Always Hurt General Capabilities: Revisiting Domain-Specific Fine-Tuning in LLMs

Jiacheng Lin[1], †, Zhongruo Wang[2], †, Kun Qian[2], Tian Wang[2], Arvind Srinivasan[2], Hansi Zeng[3], Ruochen Jiao[2], Xie Zhou[2], Jiri Gesi[2], Dakuo Wang[6], Yufan Guo[2], Kai Zhong[2], Weiqi Zhang[2], Sujay Sanghavi[4], Changyou Chen[5], Hyokun Yun[2], Lihong Li[2]

[1] University of Illinois Urbana-Champaign [2] Amazon [3] University of Massachusetts Amherst
[4] University of Texas at Austin [5] University at Buffalo [6] Northeastern University

Recent Papers Claim about Continual Learning: Domain-Specific SFT Harms General Capabilities



But!! Surprisingly, we found that

Using a smaller learning rate allows domain-specific SFT to achieve a favorable trade-off:

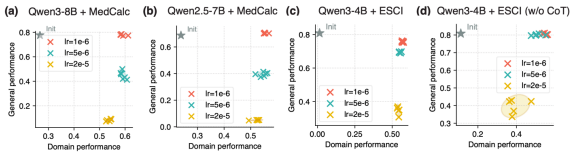
- General-purpose performance degradation is largely mitigated
- Target domain performance is comparable to that with larger learning rates

Revisiting the Role of Learning Rate

Datasets: two domain-specific datasets, **MedCalc and ESCI**

Why choose the two datasets?

- Existing open-source LLMs perform poorly on them
- Thus, domain-specific SFT is motivated to enhance specialized capabilities in domains



Findings:

- Smaller learning rates achieve a more favorable trade-off.
- Label-only supervision loosens learning rate constraints for Pareto-optimal trade-offs.

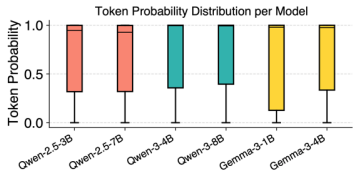
Why do milder updates preserve general abilities while still enabling strong domain gains?

Token-level Analysis

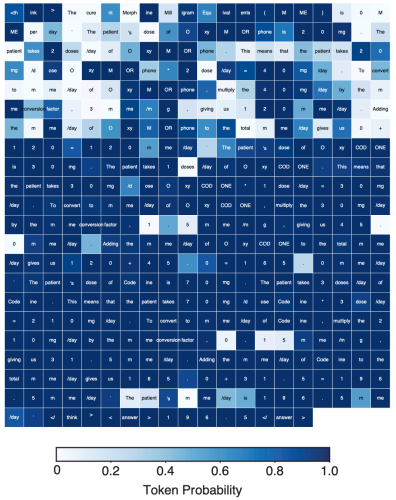
$$p(x_t \mid x_{\text{prompt}}, x_{<t})$$

We compute

- x_t: each target token during SFT
- x_{prompt}: prompt tokens
- $x_{<t}$: previous target tokens



Most tokens in SFT training data pose low learning difficulty.



- The majority of tokens in the target sequence are **confidently predicted by the model**
- The upper quartiles are tightly clustered **near 1.0**, and the medians are consistently high

Theoretical Analysis

We provide a theoretical analysis from the perspective of information theory

Theorem 3.1. *(Informal) Under certain assumptions, consider fine-tuning on a domain-specific dataset $\mathcal{D}_2$ with a fixed target domain improvement $\Delta_* > 0$ (i.e., $\Delta L(\mathcal{D}_2) \le -\Delta_*$). The general-performance degradation on $\mathcal{D}_1$, which is already well modeled by the LLM, admits an upper bound*

$$\Delta L(P_1) \le k_1 \Delta_* + k_2 \Delta_*^2 \lambda$$

where λ is the effective per-step size and k_1, k_2 are constants determined by the model and data. Thus, using smaller steps (smaller λ) leads to strictly tighter guarantees on general-performance preservation.

Theorem 3.2. *(Informal) Under certain assumptions, fix a tolerance on general-performance degradation on $\mathcal{D}_1$ (i.e., $\Delta L(P_1) \le \varepsilon_{\text{tg}}$). Then the maximal safe per-step size satisfies $\lambda_{\max} \propto \frac{\varepsilon_{\text{tg}}}{\sqrt{s}}$, where s is the expected number of low-probability tokens per example on $\mathcal{D}_2$, defined as tokens whose probabilities under the LLM are below a threshold.*

Insights from the Analysis

Directly down-weight hard-token losses to curb their potential disproportionate influence to general performance degradation

Token-Adaptive Loss Reweighting (TALR) for Domain Specific SFT

$$\min_{\mathbf{w} \in \Delta_n} \sum_{i=1}^{n} w_i \cdot \ell_i(\theta) + \tau \sum_{i=1}^{n} w_i \log w_i$$

- assign smaller weights to harder tokens
- it avoid collapsing all weight onto a small subset of tokens, ensuring broader coverage across the sequence.

Input: Domain dataset $\mathcal{D}$, parameters θ, learning rate η, temperature $\tau > 0$, weight floor $w_{\min}$
Output: Updated parameters θ

foreach *training step* **do**

Sample a mini-batch $\{(x_{\text{prompt}}^{(b)}, y^{(b)})\}_{b=1}^{B}$ from $\mathcal{D}$;
Forward pass to obtain token probabilities $\{p_t\}$ for all supervised tokens in the batch;
Token NLLs: $\ell_t \leftarrow -\log p_t$;
Adaptive weights with lower-bound clipping:

$$\tilde{w}_t \leftarrow \exp(-\ell_t/\tau), \qquad w_t \leftarrow \max\big(\text{sg}(\tilde{w}_t), w_{\min}\big)$$

Let N be the number of supervised tokens in the batch;
Mean (averaged) reweighted loss:

$$\mathcal{L}_{\text{TALR}} = \frac{1}{N} \sum_{t=1}^{N} w_t (-\log p_t)$$

Parameter update:

$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\text{TALR}}$$

end

Algorithm 1: Token-Adaptive Loss Reweighting (TALR) for Domain-Specific SFT. The $\text{sg}(\cdot)$ operator denotes *stop gradient*, meaning that w_t is treated as a constant during backpropagation to prevent gradients from flowing through the weight computation.

Experimental Results

Table 1: Comparison of domain and general performance on the MedCalc Benchmark under learning rate 1e−6. Both **Standard SFT** (with smaller learning rate) and **TALR** are our contributions, and together they achieve the best overall trade-offs compared with the other baselines.

Method	Qwen2.5-3B Domain	Qwen2.5-3B General	Qwen3-4B Domain	Qwen3-4B General	Gemma3-4B Domain	Gemma3-4B General	Average Domain	Average General
Standard (Ours)	0.4947	0.6202	0.5484	0.7837	0.5587	0.6734	0.5339	0.6924
L2-Reg	0.4904	0.6205	0.4692	0.7964	0.5595	0.6750	0.5064	0.6973
LoRA	0.1261	0.5831	0.1945	0.7640	0.2233	0.1241	0.1813	0.4904
Wise-FT	0.1948	0.6285	0.1428	0.7884	0.2573	0.7635	0.1983	0.7268
FLOW	0.3641	0.5974	0.4768	0.7870	0.5673	0.6914	0.4694	0.6920
TALR (Ours)	0.4806	0.6478	0.4889	0.7880	0.5338	0.7150	0.5011	0.7169

Table 2: Comparison of domain and general performance on the MedCalc Benchmark under learning rate 5e−6. At this larger learning rate, TALR achieves the best overall trade-off by substantially improving general performance while maintaining comparable domain performance.

Method	Qwen2.5-3B Domain	Qwen2.5-3B General	Qwen3-4B Domain	Qwen3-4B General	Gemma3-4B Domain	Gemma3-4B General	Average Domain	Average General
Standard	0.5459	0.3337	0.5782	0.5425	0.5507	0.2655	0.5583	0.3805
L2-Reg	0.5406	0.3470	0.5782	0.5591	0.5471	0.2796	0.5553	0.3952
LoRA	0.1734	0.5670	0.2367	0.7571	0.3864	0.1241	0.2655	0.4827
Wise-FT	0.3584	0.5869	0.3815	0.7531	0.4638	0.5929	0.4012	0.6443
FLOW	0.5266	0.4419	0.5819	0.5599	0.5500	0.3476	0.5528	0.4498
TALR (Ours)	0.5066	0.5490	0.5834	0.6138	0.5351	0.3427	0.5417	0.5018

Training Dynamics