LP-SFT: KEEPING PLAUSIBLE ALTERNATIVES ALIVE IN SUPERVISED FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Supervised fine-tuning (SFT) is widely used to adapt pretrained language models to downstream domains, but can over-specialize the model and degrade its pre-existing capabilities. Standard cross-entropy drives probability mass onto the observed target token and can make plausible alternatives that the pretrained model itself endorsed vanish. Using Shannon and Rényi entropies, we show that pretrained next-token distributions exhibit a regular multimodal structure, with entropy peaks corresponding to different numbers of plausible alternatives. Motivated by this observation, we propose **LP-SFT**, a **L**ocal-**P**reserving **S**upervised **F**ine-**T**uning objective with two key design principles: it removes the supervised token from a local top-K candidate set to avoid conflict with cross-entropy, and applies a locally normalized KL loss to preserve relative preferences among the remaining non-label alternatives. Across mixed-domain and domain-specific fine-tuning experiments, LP-SFT consistently outperforms standard SFT and recent baselines in aggregate performance while keeping base-endorsed alternatives from vanishing, achieving a favorable balance between single-sample accuracy and finite-budget solution accessibility, as measured by pass@1 and pass@k, respectively, with only a modest increase in training cost.

1 INTRODUCTION

Supervised fine-tuning (SFT) has become the dominant paradigm for adapting pretrained language models to instruction following, reasoning, coding, and domain-specific tasks (Ouyang et al., 2022; Wang et al., 2023). Despite its simplicity and effectiveness, SFT can degrade pre-existing capabilities in two ways: over-specialization to the fine-tuning distribution can cause catastrophic forgetting of out-of-domain capabilities, while gains in single-sample accuracy can make correct solutions less accessible on challenging problems (Welleck et al., 2019; Luo et al., 2025).

We argue that a significant portion of this degradation stems from the mismatch between the standard cross-entropy objective and the rich distributional knowledge encoded in pretrained models. Cross-entropy supervises each position with a single observed token and drives probability mass onto it, even when the pretrained next-token distribution holds several equally plausible continuations. Its effect is therefore not only to promote the target, but to abandon alternatives the model itself endorsed. We take this abandonment to be a principal source of forgetting: what SFT destroys is the set of continuations the pretrained model still considered reasonable. Using Shannon and Rényi entropies, we observe a multimodal structure with peaks near $\ln k$, corresponding to near-uniform plateau states over k plausible candidates. This structure persists across model families and datasets, showing that such alternatives are widespread rather than incidental, and that the knowledge at stake lies in the relative preferences among non-label candidates rather than in the target alone.

Motivated by this structure, we propose **LP-SFT** (**L**ocal-**P**reserving **S**upervised **F**ine-**T**uning). We first select a local top-K candidate set from the base distribution to capture plausible next-token alternatives at each position, with K motivated by the observed entropy modes. LP-SFT then follows two design principles. First, it removes the supervised token from this candidate set so that the preservation objective does not oppose cross-entropy. Second, it locally normalizes the base and trainable distributions over the resulting non-label set, preserving relative preferences without constraining its total probability mass, which prevents plausible candidates from vanishing (Figure 1).

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

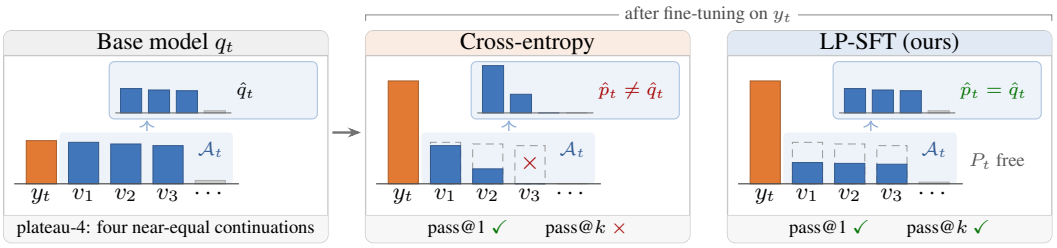


Figure 1: **Illustration of LP-SFT at a plateau position.** $\mathcal{A}_t$: top-K base candidates excluding y_t; insets: normalized profile on $\mathcal{A}_t$; dashed: base heights. LP-SFT constrains only the shape of the alternatives and leaves their mass P_t free, so y_t gains as much as under cross-entropy (pass@1) while no alternative vanishes (pass@k).

Gradient analysis and ablations validate these design choices, while the fixed top-K reference can be cached offline, avoiding the need for a frozen reference model during training.

We evaluate LP-SFT on three base models and three fine-tuning datasets covering general instruction following, code, and mathematics. LP-SFT consistently improves aggregate performance over standard SFT and recent SFT objectives and achieves the best overall pass@k performance. An additional Qwen2.5-72B scaling study confirms that these gains persist at substantially larger scale: LP-SFT has fine-tuning cost comparable to standard SFT, plus a one-time, reusable cache-construction cost.

Our contributions are summarized as follows:

- We provide a systematic analysis of pretrained next-token distributions, revealing multimodal entropy structure and heterogeneous uncertainty regimes across token positions.
- We propose **LP-SFT**, an efficient objective with two complementary design principles: target-token removal avoids conflict with cross-entropy, while local normalization preserves relative preferences among plausible non-label alternatives.
- Extensive experiments across model families and scales up to 72B show that LP-SFT consistently improves aggregate performance over standard SFT and recent baselines, achieving a favorable balance among adaptation, capability retention, and finite-budget solution accessibility.

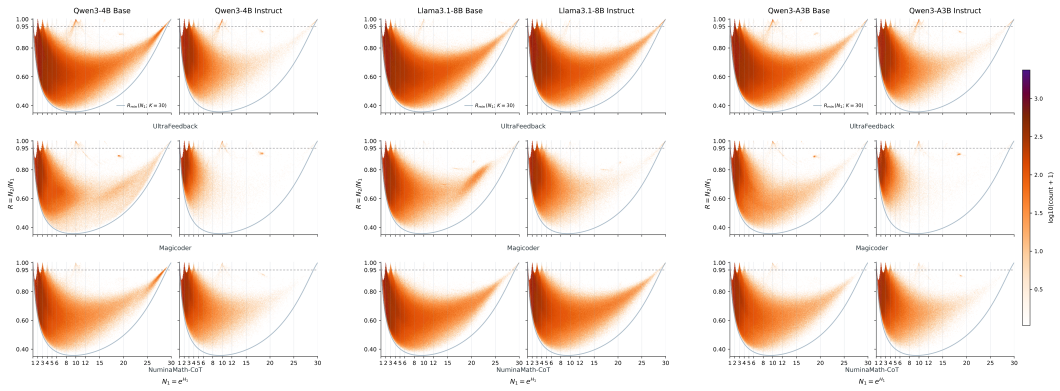
2 Related Work

Improved SFT objectives. Recent work has sought to improve SFT beyond standard cross-entropy. Dynamic Fine-Tuning (DFT) (Wu et al., 2025) and Entropy-Adaptive Fine-Tuning (EAFT) (Diao et al., 2026) modulate the target-token learning signal based on predicted probability or entropy. GEM (Li et al., 2025) reduces overfitting via entropy-regularized distribution matching. Most closely related is Anchored Supervised Fine-Tuning (ASFT) (Zhu et al., 2025), which anchors the full predictive distribution to the base model via a KL penalty; this also constrains the target's mass and requires an online frozen reference model. Decoupled Knowledge Distillation (Zhao et al., 2022) likewise separates target and non-target terms, but for teacher–student transfer. LP-SFT instead removes the target from a cached top-K base set and matches only the normalized profile of the remaining alternatives, leaving target supervision unaffected.

Entropy structure in next-token prediction. Recent work (Wang et al., 2026b) observes that the Shannon entropy of next-token distributions exhibits a multimodal structure, with peaks near $\ln k$ for small integers $k = 1, 2, 3$. Since a uniform distribution over k options has entropy $\ln k$, these peaks indicate states in which uncertainty is spread over k similarly plausible next-token candidates, known as *entropy-compressed states*. Related work also shows that token-level entropy is an informative training signal: high-entropy tokens often correspond to branching points in reasoning, where multiple continuations are plausible and small changes can substantially affect the final outcome (Wang et al., 2026a).

3 Preliminaries

Given an input prompt $\mathbf{x}$ and a target response sequence $\mathbf{y} = (y_1, \ldots, y_T)$, the model estimates the conditional probability of the next response token y_t given the prompt and the preceding response



(a) Qwen3-4B (b) Llama-3.1-8B (c) Qwen3-30B-A3B

Figure 2: Joint distributions of N_1 and $R = N_2/N_1$. High-R ridges at multiple integer support sizes reveal higher-order entropy peaks across datasets and model architectures.

tokens $\mathbf{y}_{<t} = (y_1, \ldots, y_{t-1})$. Here, y_t denotes the ground-truth response token at position t. Let $\mathcal{V}$ denote the vocabulary. At each position t, the model outputs a next-token probability distribution $p_t(\cdot) = p_\theta(\cdot \mid \mathbf{x}, \mathbf{y}_{<t})$ over $\mathcal{V}$.

Top-k Candidate Set. We define the top-k candidate set as

$$C_k(p_t) = \{v_t^{(1)}, v_t^{(2)}, \ldots, v_t^{(k)}\}, \tag{1}$$

where $v_t^{(i)}$ denotes the token with the i-th highest probability under p_t at position t. These tokens are sorted in descending order: $p_t(v_t^{(1)}) \geq p_t(v_t^{(2)}) \geq \cdots \geq p_t(v_t^{(k)})$.

Top-K Normalized Distribution. For entropy analysis, we use a fixed top-K truncation window and normalize the probability mass within $C_K(p_t)$. Unless otherwise specified, we set $K = 30$. The normalized probability of the i-th ranked token is $\hat{p}_{t,i} = p_t(v_t^{(i)}) / \sum_{j=1}^{K} p_t(v_t^{(j)})$ for $i \in \{1, \ldots, K\}$.

Top-K Shannon Entropy. We define the Shannon entropy H_1 and its effective support size N_1 over the normalized top-K distribution (Hill, 1973; Jost, 2006):

$$H_1(p_t) = -\sum_{i=1}^{K} \hat{p}_{t,i} \ln \hat{p}_{t,i}, \qquad N_1(p_t) = \exp(H_1(p_t)). \tag{2}$$

The effective support N_1 is the equivalent number of uniformly likely candidates, while Shannon entropy measures probability spread within the normalized top-K set.

Top-K Rényi-2 Entropy. We also use the Rényi-2 entropy, or collision entropy, to characterize concentration among high-probability candidates. Its effective support size N_2 is defined as

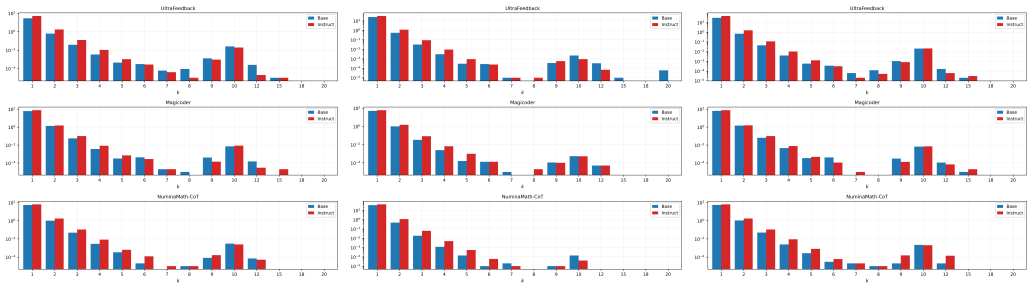
$$H_2(p_t) = -\ln \sum_{i=1}^{K} \hat{p}_{t,i}^2, \qquad N_2(p_t) = \exp(H_2(p_t)) = \frac{1}{\sum_{i=1}^{K} \hat{p}_{t,i}^2}. \tag{3}$$

Compared with Shannon entropy, Rényi-2 entropy places more emphasis on high-probability tokens and is therefore more sensitive to concentration around dominant candidates. Throughout the paper, we use N_1 and N_2 to summarize the effective number of plausible next-token candidates under the normalized top-K distribution.

4 Multimodal Entropy Structure in Next-Token Distributions

4.1 Effective Support Ratio for Identifying Entropy Peaks

While prior work has identified entropy-compressed states as evidence of multimodal structure in next-token prediction, existing analyses mainly characterize low-order peaks up to $\ln 3$, leaving



(a) Qwen3-4B (b) Llama-3.1-8B (c) Qwen3-30B-A3B

Figure 3: Occurrence of plateau-k states, defined by $N_1 \in [k-0.3, k+0.3]$ and $R > 0.95$. Instruction tuning generally increases low-order plateaus ($k = 1$–5), while the trend weakens or reverses at higher orders ($k > 5$).

higher-order entropy-compressed states insufficiently studied. To systematically identify such peaks, we introduce the *effective support ratio*:

$$R(p_t) = \frac{N_2(p_t)}{N_1(p_t)} = \exp\left(H_2(p_t) - H_1(p_t)\right) \in (0, 1]. \tag{4}$$

Here p_t denotes the next-token distribution of either a base or instruction-tuned model, and the range follows from the inequality $H_2(p_t) \leq H_1(p_t)$. The inequality becomes an equality if and only if the normalized distribution is uniform over its nonzero support. If this support contains m tokens, then

$$H_1(p_t) = H_2(p_t) = \ln m, \qquad N_1(p_t) = N_2(p_t) = m, \qquad R(p_t) = 1. \tag{5}$$

Thus, (N_1, R) jointly characterizes a plateau: N_1 identifies its order, while R measures its proximity to uniformity. In particular, $(N_1, R) \approx (m, 1)$ indicates an entropy-compressed state with roughly m plausible tokens.

To examine whether the effective support ratio R consistently reveals the multimodal structure of next-token distributions, we compute the joint distribution of R and N_1 over assistant-token positions for both base and instruction-tuned models on UltraFeedback (Cui et al., 2023), Magicoder-OSS-Instruct-75K (Wei et al., 2023), and NuminaMath-CoT (Li et al., 2024), which represent general-domain instruction following, code generation, and mathematical reasoning, respectively. We evaluate three model variants, Qwen3-4B (Yang et al., 2025), Llama-3.1-8B (Grattafiori et al., 2024), and Qwen3-30B-A3B (Yang et al., 2025), to assess whether the observed structure generalizes across model families, architectures, and scales. We compute N_1 and N_2 over the normalized top-K distribution with $K = 30$, from which R is derived.

As shown in Figure 2, the joint distribution of N_1 and R exhibits clear vertical ridges near integer values of $N_1 = 1, 2, 3, 4, 5, 10, \ldots$, especially in high-$R$ regions. These ridges indicate that the corresponding tokens lie in near-uniform entropy-compressed states with $H_1 \approx \ln m$, where m denotes the effective number of plausible next-token alternatives. Importantly, these ridges persist beyond the previously emphasized low-order regime, suggesting that higher-order entropy-compressed states are widespread across architectures and datasets. The curved lower envelope corresponds to distributions dominated by a single token, with the remaining mass spread over a low-probability tail under finite top-K normalization; Appendix A derives this boundary.

4.2 Plateau-k States: From Low-Order Branching to High-Order Ambiguity

We define entropy-compressed states of order k as plateau-k states, captured by

$$N_1 \in [k-0.3, k+0.3] \quad \text{and} \quad R > 0.95. \tag{6}$$

Figure 3 compares the proportion of plateau-k states between base and instruction-tuned models. Instruction tuning generally increases low-order plateaus ($k = 1$–5), showing that post-training concentrates probability not only on a single token ($k = 1$) but also onto small sets of plausible alternatives. In contrast, this trend weakens or reverses at higher orders ($k > 5$), where a visible concentration appears around $k = 10$. We use $k = 5$ as a descriptive cutoff rather than a strict boundary, since semantic and structural ambiguity can occur at both low and high orders; Appendix B provides qualitative examples.

Low-order plateau-k states: logical branching. Low-order plateau states typically correspond to semantic or logical branching. In these cases, the model assigns comparable probabilities to a small set of locally plausible tokens, which may include the observed target token and other valid alternatives. Such states indicate that the next-token distribution preserves multiple plausible continuation paths rather than collapsing to one token.

High-order plateau-k states: structural ambiguity. High-order plateau states exhibit qualitatively different behavior. They often arise from structural ambiguity, where the local context does not contain enough information to determine a unique next token. Typical cases include random IDs, dates, timestamps, and URLs. In such positions, the model distributes probability across alternative digits or characters; in particular, the concentration around $k = 10$ largely comes from nearly uniform alternatives among the digits 0–9. This suggests that higher-order plateaus capture diffuse structural uncertainty, rather than the semantic branching observed in the low-order regime.

This observation highlights a key limitation of cross-entropy: it supervises only the target token, leaving preferences among plausible alternatives unconstrained and distorting uncertainty in both regimes. Plateau states clarify what is at stake: their non-label candidates are not tail noise, but continuations the pretrained model judges roughly as plausible as the target. Preserving them prevents fine-tuning from discarding endorsed alternatives, while still allowing the target to absorb probability mass. What deserves protection is therefore this organization rather than individual probabilities: the integer ridges in Figure 2 show that the base model allocates uncertainty into discrete, near-uniform regimes matched to the number of admissible continuations. Preserving it does not mean freezing the entropy profile. Figure 3 shows that instruction tuning legitimately sharpens low-order plateaus; the target may absorb arbitrary mass provided the relative arrangement of the remaining candidates survives. Only their indiscriminate destruction must be ruled out. This suggests a direct route to forgetting: by driving mass onto y_t, cross-entropy suppresses endorsed alternatives, making previously held continuations unreachable even as single-sample accuracy improves.

5 Method: Preserving Non-Label Relative Structure

Motivated by the observation that pretrained models encode structured relative preferences across both branching and diffuse uncertainty regimes, we propose **LP-SFT** (**L**ocal-**P**reserving **S**upervised **F**ine-**T**uning) with two design principles: **target-token removal** and **local normalization** of relative preferences (Figure 1).

5.1 A General Non-Label Preservation Objective

Given an instruction-response pair with prompt $\mathbf{x}$ and target sequence $\mathbf{y} = (y_1, \ldots, y_T)$, we denote the frozen base model distribution and the trainable model distribution at position t by $q_t(\cdot) = p_{\theta_0}(\cdot \mid \mathbf{x}, \mathbf{y}_{<t})$ and $p_t(\cdot) = p_\theta(\cdot \mid \mathbf{x}, \mathbf{y}_{<t})$, respectively.

We first select a candidate set S_t from the frozen base-model distribution and remove the target token y_t to form the preservation set $\mathcal{A}_t$:

$$S_t = C_{K_t}(q_t), \qquad \mathcal{A}_t = S_t \setminus \{y_t\}. \tag{7}$$

Section 4.2 only imposes a lower bound on K_t: because preservation is probability-weighted, adding tokens beyond the plateau carries negligible weight, while setting K_t too small truncates endorsed alternatives. We therefore use one generous budget, $K_t = K_{\max} = 10$, covering the low-order plateaus and the regime around $k = 10$, with no per-token schedule. Appendix G.3 confirms both directions.

We then apply **local normalization**, restricting the base and trainable distributions to $\mathcal{A}_t$ and renormalizing within this non-label subset:

$$\hat{q}_t(v) = \frac{q_t(v)}{\sum_{u \in \mathcal{A}_t} q_t(u)} = \frac{q_t(v)}{Q_t}, \qquad \hat{p}_t(v) = \frac{p_t(v)}{\sum_{u \in \mathcal{A}_t} p_t(u)} = \frac{p_t(v)}{P_t}, \qquad \forall v \in \mathcal{A}_t, \tag{8}$$

where $Q_t = q_t(\mathcal{A}_t)$ and $P_t = p_t(\mathcal{A}_t)$ denote the total probability masses that the two models assign to the preservation set. Here, $\hat{q}_t$ and $\hat{p}_t$ capture relative preferences over the non-target alternatives. Local normalization aligns naturally with this plateau structure: it preserves the relative probability

profile among candidates where multimodal knowledge resides, while leaving their total mass P_t unconstrained so that the supervised target y_t can absorb probability freely. In contrast, matching unnormalized probabilities directly constrains total non-target mass and opposes cross-entropy, as Section 5.2 demonstrates. The preservation loss is defined as

$$\mathcal{L}_{\mathrm{LP}}^t = D_{\mathrm{KL}}(\hat{q}_t \,\|\, \hat{p}_t) = \sum_{v \in \mathcal{A}_t} \hat{q}_t(v) \ln \frac{\hat{q}_t(v)}{\hat{p}_t(v)}, \tag{9}$$

and the full training objective is

$$\mathcal{L}(\theta) = \sum_{t=1}^{T} \left[-\ln p_t(y_t) + \mu \, \mathcal{L}_{\mathrm{LP}}^t \right], \tag{10}$$

where μ controls the preservation strength. Since $\hat{q}_t$ is derived from the frozen base model, it is a fixed target, so minimizing Eq. (9) is equivalent, up to a constant independent of θ, to minimizing the local cross-entropy $-\sum_{v \in \mathcal{A}_t} \hat{q}_t(v) \ln \hat{p}_t(v)$.

Unlike standard distillation (Hinton et al., 2015; Gu et al., 2024; Agarwal et al., 2024), which transfers knowledge by matching a teacher distribution, LP-SFT uses the frozen base model only to preserve the local structure of non-label alternatives.

5.2 A Gradient View: Why Both Principles Matter

5.2.1 Local Normalization

The locally normalized loss $\mathcal{L}_{\mathrm{LP}}^t$ has the gradient (Appendix C.1)

$$\frac{\partial \mathcal{L}_{\mathrm{LP}}^t}{\partial z_t(w)} = \begin{cases} \dfrac{p_t(w)}{P_t} - \dfrac{q_t(w)}{Q_t}, & w \in \mathcal{A}_t, \\ 0, & w \notin \mathcal{A}_t. \end{cases} \tag{11}$$

Thus, it only adjusts the relative logits within $\mathcal{A}_t$, leaving those outside unaffected. In contrast, the unnormalized truncation acts fundamentally differently, as seen from its exact decomposition:

$$\widetilde{\mathcal{L}}^t = \sum_{v \in \mathcal{A}_t} q_t(v) \ln \frac{q_t(v)}{p_t(v)} = Q_t \mathcal{L}_{\mathrm{LP}}^t + Q_t \ln \frac{Q_t}{P_t}. \tag{12}$$

The principal difference is the mass term $Q_t \ln(Q_t/P_t)$, which is strictly decreasing in P_t and therefore exerts a one-sided pressure toward $P_t \to 1$, rather than matching the preservation-set mass to its base-model value. This pressure is also visible from the full gradient:

$$\frac{\partial \widetilde{\mathcal{L}}^t}{\partial z_t(w)} = \begin{cases} Q_t p_t(w) - q_t(w), & w \in \mathcal{A}_t, \\ Q_t p_t(w), & w \notin \mathcal{A}_t. \end{cases} \tag{13}$$

Every logit outside $\mathcal{A}_t$ therefore has a positive gradient, while the gradients inside satisfy $\sum_{w \in \mathcal{A}_t} \partial \widetilde{\mathcal{L}}^t / \partial z_t(w) = Q_t(P_t - 1) < 0$. Gradient descent thus suppresses logits outside the preservation set and raises those inside as a group, shifting probability mass into $\mathcal{A}_t$. In addition, the factor Q_t scales the relative-preservation term in Eq. (12), making the effective preservation weight μQ_t vary across token positions. Local normalization removes both this mass-shifting pressure and the position-dependent reweighting, leaving only the KL constraint on the relative distributions within $\mathcal{A}_t$.

5.2.2 Target-Token Removal

Target-token removal eliminates a direct conflict at the supervised logit. If y_t is retained in a locally normalized preservation set, let $\mathcal{L}_{\mathrm{pres}}^t$ denote the resulting preservation term. The target-logit gradients of cross-entropy and the preservation term are

$$\frac{\partial \mathcal{L}_{\mathrm{CE}}^t}{\partial z_t(y_t)} = p_t(y_t) - 1, \qquad \frac{\partial \mathcal{L}_{\mathrm{pres}}^t}{\partial z_t(y_t)} = \hat{p}_t(y_t) - \hat{q}_t(y_t).$$

Cross-entropy increases the target probability, whereas the preservation term pulls the target logit back once $\hat{p}_t(y_t)$ exceeds its base-model value $\hat{q}_t(y_t)$. Denoting the stationary target probability

Table 1: **Fine-tuning results on the mixed-domain dataset.** LP-SFT achieves the strongest result among fine-tuned methods on every reported metric.

Model	Method	Math				Code				GEN.	Avg.
		MATH-500		AIME 21–26		MBPP+		HE+		MMLU	
		pass@1	pass@16	pass@1	pass@32	pass@1	pass@10	pass@1	pass@10	5-shot Acc.	All
Qwen3-14B	Base	69.50	89.40	7.00	26.11	69.62	87.88	77.74	93.32	80.51	66.79
	CE	51.60	85.80	4.62	21.67	57.12	81.49	58.26	88.49	77.92	58.55
	DFT	40.12	62.20	3.25	10.56	68.16	70.84	65.95	71.48	77.90	52.27
	EAFT	51.00	85.60	4.77	26.11	61.15	82.92	64.88	89.24	78.44	60.46
	GEM	47.89	85.80	3.99	22.22	54.29	82.63	55.79	86.26	77.53	57.38
	ASFT	54.57	82.80	4.10	21.11	68.07	78.18	72.10	83.20	78.61	60.30
	LP-SFT	**64.20**	**89.20**	**6.79**	**27.78**	**69.17**	**85.74**	**74.66**	**90.25**	**80.05**	**65.32**

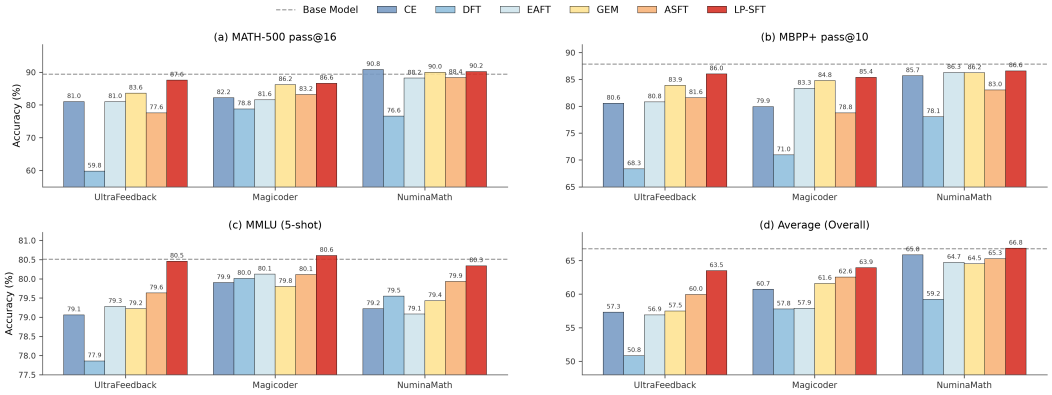


Figure 4: **Qwen3-14B fine-tuning results across three training datasets.** LP-SFT achieves the highest overall average on UltraFeedback, Magicoder, and NuminaMath, while consistently preserving strong pass@k performance and general knowledge.

under cross-entropy plus preservation by $p^{\star}_{t,\mathrm{pres}}(y_t)$, we obtain, for any non-degenerate preservation set,

$$p^{\star}_{t,\mathrm{pres}}(y_t) \le \frac{1+\mu\hat{q}_t(y_t)}{1+\mu} < 1. \tag{14}$$

By combining local normalization with target-token removal, LP-SFT guarantees that the target-logit fixed point remains at the cross-entropy optimum, $p^{\star}_t(y_t) = 1$. This does not imply that the trained model assigns probability one to every supervised token; rather, it means that LP-SFT preserves non-label structure without attenuating SFT's direct supervision on y_t. Appendix C.2 derives the exact stationary points for all four combinations, and Appendix G.2 verifies their empirical ordering.

5.3 Efficient Implementation

A naive implementation of LP-SFT would require keeping both the frozen base model and the trainable model in memory and running an additional base-model forward pass at each training step. Since the base model θ_0 is frozen and the training data is fixed, we instead precompute the required base-distribution information offline.

As shown in Algorithm 1 in Appendix D, LP-SFT is implemented in two stages. In Stage 1, we perform a single offline pass over the training dataset with the base model and cache the top-$K_{\max}$ candidate logits required to construct the local preservation set. In Stage 2, fine-tuning proceeds with only the standard forward pass of the trainable model. The local preservation loss $\mathcal{L}^t_{\mathrm{LP}}$ is computed at each training step using the cached base logits and the current model logits restricted to $\mathcal{A}_t$.

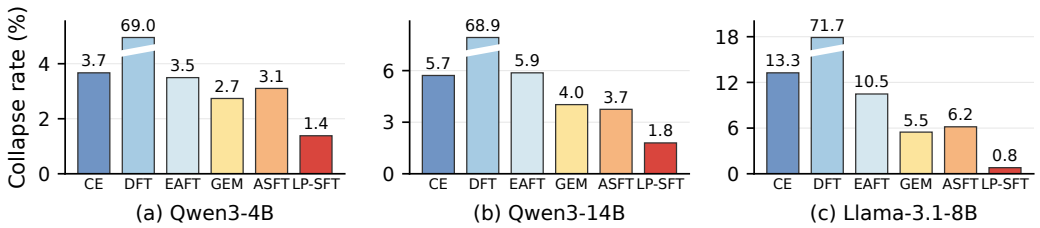


Figure 5: **Collapse of base-endorsed alternatives after fine-tuning on UltraFeedback.** Collapse rate (Eq. 15; lower is better) on base-generated MATH-500 solutions (App. H); DFT clipped.

6 Experiments

We evaluate **LP-SFT** across multiple model scales and domains, focusing on two questions: (1) how it balances single-sample reliability, measured by pass@1, with finite-budget solution accessibility, measured by pass@k, and (2) whether it mitigates catastrophic forgetting while improving overall performance over standard SFT and recent alternatives.

6.1 Setup

Models and datasets. Our main experiments evaluate three base models spanning different scales: **Qwen3-4B**, **Llama-3.1-8B**, and **Qwen3-14B**. To test domain learning and capability preservation, we use **Magicoder-OSS-Instruct-75K** for code, a 100K subset of **NuminaMath-CoT** for mathematics, and **UltraFeedback** (61K) for general alignment. We additionally conduct a rank-64 LoRA scaling study on **Qwen2.5-72B** with UltraFeedback-30K in Appendix F.2.

Baselines. We compare LP-SFT with standard supervised fine-tuning using cross-entropy loss (**CE**), Dynamic Fine-Tuning (**DFT**) (Wu et al., 2025), Entropy-Adaptive Fine-Tuning (**EAFT**) (Diao et al., 2026), **GEM** (Li et al., 2025), and Anchored Supervised Fine-Tuning (**ASFT**) (Zhu et al., 2025). For LP-SFT, we set the preservation weight to $\mu = 1$ and use a fixed top-$K_{\max}$ preservation set with $K_{\max} = 10$ by default.

Evaluation benchmarks and metrics. We evaluate fine-tuned models across mathematics (**MATH-500**, **AIME 2021–2026**), coding (**MBPP+**, **HumanEval+**), and general knowledge (**MMLU**, 5-shot) (Hendrycks et al., 2020; 2021; Liu et al., 2023); detailed task setups are in Appendix E. For generation tasks, we report both **pass@1** (single-sample reliability) and **pass@k** (finite-budget solution accessibility) (Chen et al., 2021). This pairing mirrors the mechanism in Section 4.2: cross-entropy can maintain pass@1 on the supervised target, but by suppressing endorsed alternatives it makes other reasoning paths harder to reach, which surfaces as a decline in pass@k. Avg. denotes the mean of all displayed metrics.

6.2 Mixed-Domain Fine-Tuning

We first evaluate LP-SFT in a mixed-domain setting, in which Qwen3-14B is fine-tuned on a mixture of three datasets covering mathematical reasoning, code generation, and general instruction following, so that competing optimization signals can exacerbate forgetting. As shown in Table 1, LP-SFT achieves the best results across all displayed metrics among the fine-tuned models, including both pass@1 and pass@k.

The distinction is especially clear in general-knowledge retention. Every competing fine-tuning objective substantially reduces MMLU accuracy from the base model's 80.51 to around 78, whereas LP-SFT retains 80.05, a drop of only 0.46 points. The same pattern holds in aggregate performance: competing methods fall 6.33–14.52 average points below the base model, whereas LP-SFT reduces this gap to only 1.47 points.

6.3 Domain-Specific Fine-Tuning

We further evaluate LP-SFT under domain-specific fine-tuning on three datasets: UltraFeedback, Magicoder-OSS-Instruct-75K, and NuminaMath-CoT. Figure 4 summarizes the Qwen3-14B results

Table 2: **Fine-tuning results on UltraFeedback.** LP-SFT obtains the highest overall average for all three model families, with consistent gains in MBPP+ pass@10 and MMLU across all models, alongside substantial mathematical improvements on Qwen3.

Model	Method	Math				Code				GEN.	Avg.
		MATH-500		AIME 21–26		MBPP+		HE+		MMLU	
		pass@1 / pass@16		pass@1 / pass@32		pass@1 / pass@10		pass@1 / pass@10		5-shot Acc.	All
Qwen3-4B	CE	38.66	79.20	1.27	16.67	51.28	72.57	62.26	86.33	71.56	53.31
	DFT	37.75	59.60	0.75	5.00	54.62	61.67	64.42	68.62	70.64	47.01
	EAFT	26.90	75.60	0.90	13.33	52.18	76.85	62.29	86.54	72.23	51.87
	GEM	36.39	79.00	1.09	15.56	46.97	75.85	56.07	87.35	71.38	52.18
	ASFT	40.86	74.00	1.74	12.78	57.46	71.37	68.08	83.72	71.92	53.55
	LP-SFT	**50.15**	**83.40**	**3.39**	**19.44**	**57.79**	**80.69**	**69.30**	**92.29**	**72.62**	**58.79**
Qwen3-14B	CE	45.16	81.00	1.79	16.11	60.32	80.56	64.94	86.84	79.06	57.31
	DFT	40.27	59.80	0.75	4.44	63.81	68.35	68.08	74.17	77.86	50.84
	EAFT	43.00	81.00	1.96	15.56	59.38	80.81	63.87	87.39	79.28	56.92
	GEM	43.00	83.60	2.14	17.78	56.75	83.90	61.37	89.62	79.22	57.49
	ASFT	46.06	77.60	2.14	16.67	**71.01**	81.59	**77.74**	87.23	79.63	59.96
	LP-SFT	**54.71**	**87.60**	**4.70**	**23.33**	68.00	**86.03**	74.30	**92.31**	**80.46**	**63.49**
Llama-3.1-8B	CE	**10.76**	**45.80**	**0.23**	**6.11**	31.61	51.04	33.05	59.12	61.11	33.20
	DFT	5.16	16.60	0.21	5.00	26.59	32.55	25.27	32.18	55.11	22.07
	EAFT	9.31	43.40	0.19	5.56	27.08	49.91	**34.15**	59.50	61.14	32.25
	GEM	8.82	44.00	0.21	**6.11**	30.34	54.33	32.44	**62.84**	61.00	33.34
	ASFT	8.45	35.60	0.05	1.67	36.59	55.49	31.80	51.07	60.58	31.26
	LP-SFT	9.41	42.00	0.12	3.33	**37.00**	**63.04**	29.79	57.75	**62.78**	**33.91**

across these datasets, while Table 2 reports detailed UltraFeedback results for Qwen3-4B, Qwen3-14B, and Llama-3.1-8B. Additional results on Magicoder and NuminaMath are provided in Appendix F.

Across all combinations of the three base models and training datasets, LP-SFT achieves the highest overall average. The gains are most pronounced on UltraFeedback, where general-domain fine-tuning can substantially degrade mathematical reasoning and code generation. On Magicoder and NuminaMath, LP-SFT also improves the overall average and MMLU accuracy while maintaining competitive task-specific pass@k.

Comparison with baselines. Figure 4 and Table 2 show that DFT, ASFT, and GEM each sacrifice pass@1 or pass@k, whereas LP-SFT balances both and preserves general knowledge. Figure 5 traces these trade-offs to base-endorsed alternatives $\mathcal{E}_t$ via the collapse rate

$$\mathrm{CR} = \frac{\sum_t \sum_{v \in \mathcal{E}_t} \mathbf{1}[p_t(v) < 10^{-3}]}{\sum_t |\mathcal{E}_t|}, \qquad \mathcal{E}_t = \{v \in \mathcal{A}_t : q_t(v) \geq 0.05\}. \tag{15}$$

CE collapses 3.7–13.3% of them and DFT about 70%, whereas LP-SFT reduces CR to 0.8–1.8%. Since LP-SFT constrains neither y_t nor the alternative mass P_t, this suggests that keeping endorsed alternatives from degenerating is a key source of its gain. Gains are largest on Qwen3, whereas CE remains stronger on Llama-3.1-8B math, suggesting that stronger bases offer more reliable structure.

Scaling and ablations. On Qwen2.5-72B, LP-SFT achieves the best aggregate score among all fine-tuned methods and slightly exceeds the base model, at CE-level fine-tuning cost plus a one-time cache (Tables 18 and 19). Dropping either target-token removal or local normalization substantially weakens the gains (Appendix G.2); local top-K preservation matches the full-vocabulary variant without a frozen reference model (Appendix G.3); and LP-SFT is robust to μ (Figure 6).

7 Conclusion

Our entropy analysis shows that base-model predictions exhibit structured plateau states, indicating that useful knowledge resides in relative preferences among non-label alternatives rather than their absolute probability mass. This motivates two core design principles: target-token removal and local normalization, which jointly safeguard non-label structure without restricting the supervised target. Crucially, local top-K preservation matches full-vocabulary retention without reference-model overhead. Across diverse domains and model scales up to Qwen2.5-72B, LP-SFT consistently achieves the strongest balance between single-sample reliability and finite-budget solution accessibility.

AI USE STATEMENT

Generative AI tools were used throughout this research. Tasks for which disclosure is required include proposing and refining hypotheses, developing the conceptual framework, designing the methodology and experiments, formulating mathematical claims and assisting in writing the proofs in the appendices, implementing the training, evaluation, and entropy-analysis code, preparing and subsampling the datasets, analyzing the qualitative token-level examples, interpreting experimental results, and translating author-written text into English. We did not use generative AI tools to generate synthetic datasets: all training and evaluation data come from existing public datasets and benchmarks. Transcribing recordings of research material and formulating survey or interview questions are not applicable to this work.

We additionally used generative AI tools for tasks in the recommended disclosure category, including drafting and editing the manuscript, creating figures and LaTeX tables, brainstorming candidate titles, and searching for related literature.

We have reviewed all AI-assisted work. The authors take responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The method and its optimization objective are specified in Section 5, with implementation pseudocode provided in Appendix D. Appendix E documents the training and evaluation protocols, and Appendix G reports hyperparameter sensitivity, component ablations, and computational costs. Complete derivations supporting the theoretical analysis are also included in the appendix.

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 *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, et al. Ultrafeedback: Boosting language models with scaled ai feedback. *arXiv preprint arXiv:2310.01377*, 2023.

Muxi Diao, Lele Yang, Wuxuan Gong, Yutong Zhang, Zhonghao Yan, Yufei Han, Kongming Liang, Weiran Xu, and Zhanyu Ma. Entropy-adaptive fine-tuning: Resolving confident conflicts to mitigate forgetting. *arXiv preprint arXiv:2601.02151*, 2026.

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.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*, volume 2024, pp. 32694–32717, 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.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Mark O Hill. Diversity and evenness: a unifying notation and its consequences. *Ecology*, 54(2): 427–432, 1973.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Lou Jost. Entropy and diversity. *Oikos*, 113(2):363–375, 2006.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, et al. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13(9):9, 2024.

Ziniu Li, Congliang Chen, Tian Xu, Zeyu Qin, Jiancong Xiao, Zhi-Quan Luo, and Ruoyu Sun. Preserving diversity in supervised fine-tuning of large language models. In *International Conference on Learning Representations*, volume 2025, pp. 66127–66154, 2025.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *Advances in neural information processing systems*, 36:21558–21572, 2023.

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.

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

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *Advances in Neural Information Processing Systems*, 38:115452–115486, 2026a.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 13484–13508, 2023.

Yueyang Wang, Jiawei Fu, Baolong Bi, Xili Wang, and Xiaoqing Liu. He-snr: Uncovering latent logic via entropy for guiding mid-training on swe-bench. *arXiv preprint arXiv:2601.20255*, 2026b.

Yuxiang Wei, Zhe Wang, Jiawei Liu, Yifeng Ding, and Lingming Zhang. Magicoder: Empowering code generation with oss-instruct. *arXiv preprint arXiv:2312.02120*, 2023.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319*, 2019.

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*, 2025.

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.

Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11943–11952. IEEE, 2022.

He Zhu, Junyou Su, Peng Lai, Ren Ma, Wenjia Zhang, Linyi Yang, and Guanhua Chen. Anchored supervised fine-tuning. *arXiv preprint arXiv:2509.23753*, 2025.

A Derivation of the Finite-top-K Lower Envelope

We derive the finite-top-K lower-envelope branch of R for a given Shannon effective support size N_1. Let $\hat{p} = (\hat{p}_1, \ldots, \hat{p}_K)$ denote the normalized top-K next-token distribution. For a fixed N_1, minimizing R is equivalent to minimizing N_2, or equivalently maximizing the collision probability $\sum_{i=1}^{K} \hat{p}_i^2$, under a fixed Shannon entropy constraint:

$$\max_{\hat{p}} \quad \sum_{i=1}^{K} \hat{p}_i^2$$
$$\text{s.t.} \quad \begin{cases} \sum_{i=1}^{K} \hat{p}_i = 1, \\ -\sum_{i=1}^{K} \hat{p}_i \ln \hat{p}_i = \ln N_1, \\ \hat{p}_i \geq 0, \qquad i = 1, \ldots, K. \end{cases} \tag{16}$$

The Lagrangian of this constrained optimization problem is

$$\mathcal{L} = \sum_{i=1}^{K} \hat{p}_i^2 - \lambda \left(\sum_{i=1}^{K} \hat{p}_i - 1 \right) - \mu \left(-\sum_{i=1}^{K} \hat{p}_i \ln \hat{p}_i - \ln N_1 \right). \tag{17}$$

For any positive component $\hat{p}_i > 0$, the stationarity condition gives

$$\frac{\partial \mathcal{L}}{\partial \hat{p}_i} = 2\hat{p}_i - \lambda + \mu(\ln \hat{p}_i + 1) = 0, \tag{18}$$

or equivalently

$$2\hat{p}_i + \mu \ln \hat{p}_i = C, \tag{19}$$

where $C = \lambda - \mu$ is a constant independent of i. Since the scalar equation $2x + \mu \ln x = C$ has at most two positive solutions, any interior stationary extremum has at most two distinct positive probability values. Boundary extrema are covered by allowing some probabilities to be zero.

Therefore, the general two-level extremal family can be written as

$$\hat{p}^{\dagger}(j, m, a, b) = \Big(\underbrace{a, \ldots, a}_{j \text{ entries}}, \ \underbrace{b, \ldots, b}_{m-j \text{ entries}}, \ 0, \ldots, 0 \Big), \tag{20}$$

where

$$a > b \geq 0, \qquad 1 \leq j \leq m-1, \qquad 2 \leq m \leq K. \tag{21}$$

The values (a, b) are determined by the normalization and entropy constraints

$$ja + (m-j)b = 1, \qquad -ja \ln a - (m-j)b \ln b = \ln N_1. \tag{22}$$

Among these two-level candidates, the lower-envelope branch observed in the (N_1, R) plane is attained by the most asymmetric case, where the larger probability value appears once. Intuitively, concentrating the high-probability mass on a single coordinate increases the collision probability, while the remaining mass forms a uniform residual tail. Enumeration over the finite range used in our analysis, with $m \in \{2, \ldots, 30\}$ and $N_1 \in (1, m]$, confirms that the lower branch is attained by $j = 1$.

Thus, the relevant extremal family reduces to the one-dominant / uniform-tail form

$$\hat{p}^{\star}(q, m) = \Big(q, \ \underbrace{\tfrac{1-q}{m-1}, \ldots, \tfrac{1-q}{m-1}}_{m-1 \text{ entries}}, \ 0, \ldots, 0 \Big), \qquad q \in [1/m, 1], \quad m \in \{2, \ldots, K\}. \tag{23}$$

For this family, the Shannon entropy is

$$H_1(q, m) = -q \ln q - (1-q) \ln \frac{1-q}{m-1}, \tag{24}$$

and the collision probability is

$$\sum_{i=1}^{K} \hat{p}_i^2 = q^2 + \frac{(1-q)^2}{m-1}. \tag{25}$$

The finite-top-K lower-envelope branch can then be written as

$$R_{\min}(N_1; K) = \min_{2 \le m \le K,\ q \in [1/m, 1]} \frac{1}{N_1 \left[q^2 + \frac{(1-q)^2}{m-1} \right]} \quad \text{s.t.} \quad H_1(q, m) = \ln N_1. \tag{26}$$

Infeasible values of m are automatically excluded by the entropy constraint $H_1(q, m) = \ln N_1$. In our entropy analyses, $K = 30$, since both N_1 and N_2 are computed from the normalized top-K distribution with $K = 30$.

B Representative Plateau Examples

We provide representative token-level examples for different plateau orders. Each example shows a truncated context window around the ground-truth target token, with the target highlighted, together with the corresponding prediction distribution. For $k \le 10$, we show the top-10 candidates; for $k > 10$, we show the top-20 candidates.

Instance 1: $k = 2$ **Target Token:** `of` N_1**:** 2.000 R**:** 1.0000

Context Window (Truncated)

```
To solve for \( x \), we first combine like terms:

\[ \frac{7}{2}x = 14 \]

Now, we multiply both sides of the equation by \( \frac{2}{7} \) to
    isolate \( x \):

\[ x = 14 \times \frac{2}{7} \]
```

Table 3: Top-10 Prediction Distribution for Instance 1 ($k = 2$).

Rank	Token	Prob.	Rank	Token	Prob.
1	of	0.500	6	by	0.000
2	by	0.500	7	By	0.000
3	with	0.000	8	of	0.000
4	cua	0.000	9	the	0.000
5	to	0.000	10	Thai-of	0.000

Instance 2: $k = 3$ **Target Token:** `divide` N_1**:** 3.000 R**:** 0.9999

Context Window (Truncated)

```
7000(1 + r/1)^(1*2)
19828.80 = 17000(1 + r)^2

Now, divide both sides by 17000:

19828.80 / 17000 = (1 + r)^2
```

Table 4: Top-10 Prediction Distribution for Instance 2 ($k = 3$).

Rank	Token	Prob.	Rank	Token	Prob.
1	divide	0.333	6	dividing	0.000
2	let	0.333	7	simplify	0.000
3	we	0.333	8	to	0.000
4	solve	0.000	9	divided	0.000
5	divide	0.000	10	square	0.000

Instance 3: $k = 4$ **Target Token:** + N_1**:** 4.029 R**:** 0.9945

Context Window (Truncated)

```
+ 1 - \frac{3}{ax} \\
&= \frac{2a^2x^2 + ax - 3}{ax} \\
&= \frac{(2ax + 3)(ax-1)}{ax}.
\end{aligned}
\]

For \(x=1\) to be a critical point, \(f'(1)=0\). Plugging in \(
```

Table 5: Top-10 Prediction Distribution for Instance 3 ($k = 4$).

Rank	Token	Prob.	Rank	Token	Prob.
1	␣−	0.250	6	^	0.000
2	−	0.250	7	+\	0.000
3	+	0.250	8	−\	0.000
4	␣+	0.250	9	+a	0.000
5	)^	0.000	10	_	0.000

Instance 4: $k = 5$ **Target Token:** 7 N_1**:** 5.277 R**:** 0.9647

Context Window (Truncated)

```
Keim, SC (2004) "Risk perceptions and safety compliance of workers
    employed in agriculture, forestry, and fishing." Journal of Safety
    Research 35(2), 23 7 -244.
```

Table 6: Top-10 Prediction Distribution for Instance 4 ($k = 5$).

Rank	Token	Prob.	Rank	Token	Prob.
1	5	0.198	6	–	0.003
2	3	0.198	7	4	0.001
3	7	0.198	8	6	0.001
4	1	0.198	9	0	0.001
5	9	0.198	10	2	0.001

Instance 5: $k = 6$ **Target Token:** 4 N_1**:** 6.021 R**:** 0.9972

Context Window (Truncated)

```
A. G., & Tuuli, M. G. (2016). Maternal marijuana use and adverse
    neonatal outcomes. Obstetrics & Gynecology, 128( 4 ), 713-723.
```

Table 7: Top-10 Prediction Distribution for Instance 5 ($k = 6$).

Rank	Token	Prob.	Rank	Token	Prob.
1	2	0.167	6	3	0.167
2	1	0.167	7	7	0.000
3	6	0.167	8	9	0.000
4	5	0.167	9	8	0.000
5	4	0.167	10	0	0.000

Instance 6: $k = 7$ **Target Token:** 3 N_1**:** 7.073 R**:** 0.9722

Context Window (Truncated)

```
According to a 2021 Gallup poll, 54% of Americans support the death
    penalty for murder, while 4 3 % oppose it.
```

Table 8: Top-10 Prediction Distribution for Instance 6 ($k = 7$).

Rank	Token	Prob.	Rank	Token	Prob.
1	2	0.167	6	4	0.131
2	1	0.167	7	5	0.102
3	6	0.167	8	7	0.003
4	3	0.131	9	8	0.001
5	0	0.131	10	9	0.000

Instance 7: $k = 8$ **Target Token:** 4 N_1**:** 8.285 R**:** 0.9733

Context Window (Truncated)

```
'Authorization' => 'Basic NTZmZGJmN2EtODUxYy00M2RiLTk
    4 YWUtYTBhZmEzYzFjZGRi'
```

Table 9: Top-10 Prediction Distribution for Instance 7 ($k = 8$).

Rank	Token	Prob.	Rank	Token	Prob.
1	x	0.159	6	2	0.110
2	1	0.125	7	4	0.110
3	z	0.125	8	5	0.110
4	0	0.125	9	zM	0.011
5	3	0.125	10	...	0.000

Instance 8: $k = 9$ **Target Token:** 6 N_1**:** 9.033 R**:** 0.9974

Context Window (Truncated)

```
Link: <https://www.inc.com/guides/2010/0 6 /8-entrepreneurial-myths.
    html>
```

Table 10: Top-10 Prediction Distribution for Instance 8 ($k = 9$).

Rank	Token	Prob.	Rank	Token	Prob.
1	4	0.111	6	7	0.111
2	3	0.111	7	5	0.111
3	1	0.111	8	2	0.111
4	6	0.111	9	9	0.111
5	8	0.111	10	0	0.000

Instance 9: $k = 10$ **Target Token:** 0 N_1**:** 10.001 R**:** 0.9999

Context Window (Truncated)

```
| Household Type | Number of Households | Percentage |
| --- | --- | --- |
| Single-person households | 1,598,80 6 | 54.0% |
```

Table 11: Top-10 Prediction Distribution for Instance 9 ($k = 10$).

Rank	Token	Prob.	Rank	Token	Prob.
1	3	0.100	6	6	0.100
2	2	0.100	7	0	0.100
3	4	0.100	8	1	0.100
4	5	0.100	9	9	0.100
5	7	0.100	10	8	0.100

Instance 10: $k = 12$ **Target Token:** 9 N_1**:** 11.998 R**:** 0.9981

Context Window (Truncated)

```
i.pinimg.com/originals/54/ba/82/54ba8283ca60486b3acd27528d7faec9.jpg)
     | Set up a dedicated space with simple building blocks ...
```

Table 12: Top-20 Prediction Distribution for Instance 10 ($k = 12$).

Rank	Token	Prob.	Rank	Token	Prob.
1	b	0.092	11	2	0.082
2	f	0.092	12	3	0.082
3	7	0.082	13	d	0.000
4	6	0.082	14	<eot>	0.000
5	1	0.082	15	ac	0.000
6	0	0.082	16	g	0.000
7	4	0.082	17	ab	0.000
8	5	0.082	18	fa	0.000
9	8	0.082	19	=	0.000
10	9	0.082	20	ad	0.000

Instance 11: $k = 15$ **Target Token:** a N_1**:** 14.966 R**:** 0.9969

Context Window (Truncated)

```
| STEM Corner | [Image](https://i.pinimg.com/originals/54/ba/82/54
    ba8283ca60486b3acd27528d7faec9.jpg) | Set up a dedicated space ...
```

Table 13: Top-20 Prediction Distribution for Instance 11 ($k = 15$).

Rank	Token	Prob.	Rank	Token	Prob.
1	4	0.070	11	8	0.062
2	5	0.070	12	3	0.062
3	1	0.070	13	7	0.062
4	9	0.070	14	6	0.062
5	b	0.070	15	f	0.055
6	a	0.070	16	e	0.000
7	d	0.070	17	ad	0.000
8	0	0.070	18	ab	0.000
9	c	0.070	19	fa	0.000
10	2	0.062	20	da	0.000

C Gradient of the Locally Normalized Preservation Loss

This appendix supplies the derivations behind Section 5.2. Each subsection establishes one claim made there, in the order the main text uses them:

- Appendix C.1 derives Eqs. (11)–(13).
- Appendix C.2 derives Eq. (14) and Table 14, showing that only the complete design leaves the target logit unconstrained.

C.1 Normalized and Unnormalized Preservation Gradients

Fix a position t and drop the subscript for readability. Let p and q denote the original next-token distributions of the trainable and base models, respectively, and let z denote the trainable logits. Define

$$P = \sum_{v\in\mathcal{A}} p(v), \qquad Q = \sum_{v\in\mathcal{A}} q(v), \qquad \hat{p}(v) = \frac{p(v)}{P}, \qquad \hat{q}(v) = \frac{q(v)}{Q}.$$

We refer to $p(v)$ and $q(v)$ as the *original probabilities*, and to $\hat{p}(v)$ and $\hat{q}(v)$ as the *locally normalized probabilities*. The preservation set $\mathcal{A}$ and the base distribution q are fixed during training; for nonempty $\mathcal{A}$, both P and Q are strictly positive.

Normalized gradient. For $v \in \mathcal{A}$, local normalization cancels the global softmax normalizer:

$$\hat{p}(v) = \frac{\exp z(v)}{\sum_{u\in\mathcal{A}} \exp z(u)}, \qquad \frac{\partial \ln \hat{p}(v)}{\partial z(w)} = \mathbf{1}[v = w] - \hat{p}(w)\mathbf{1}[w \in \mathcal{A}].$$

Since $\mathcal{L}_{\mathrm{LP}} = -\sum_{v\in\mathcal{A}} \hat{q}(v) \ln \hat{p}(v) + \text{const}$ and $\sum_{v\in\mathcal{A}} \hat{q}(v) = 1$,

$$\frac{\partial \mathcal{L}_{\mathrm{LP}}}{\partial z(w)} = \begin{cases} \hat{p}(w) - \hat{q}(w) = \dfrac{p(w)}{P} - \dfrac{q(w)}{Q}, & w \in \mathcal{A}, \\ 0, & w \notin \mathcal{A}. \end{cases} \tag{27}$$

Unnormalized decomposition. For $\widetilde{\mathcal{L}} = \sum_{v\in\mathcal{A}} q(v) \ln \frac{q(v)}{p(v)}$, substituting $q(v) = Q\hat{q}(v)$ and $p(v) = P\hat{p}(v)$ gives

$$\begin{aligned} \widetilde{\mathcal{L}} &= \sum_{v\in\mathcal{A}} Q\hat{q}(v) \ln \frac{Q\hat{q}(v)}{P\hat{p}(v)} \\ &= Q\, D_{\mathrm{KL}}(\hat{q} \,\|\, \hat{p}) + Q \ln \frac{Q}{P} = Q\mathcal{L}_{\mathrm{LP}} + Q \ln \frac{Q}{P}. \end{aligned} \tag{28}$$

Unnormalized gradient. Using $\partial \ln p(v)/\partial z(w) = \mathbf{1}[v = w] - p(w)$,

$$\begin{aligned}\frac{\partial \widetilde{\mathcal{L}}}{\partial z(w)} &= -\sum_{v\in\mathcal{A}} q(v)\left(\mathbf{1}[v = w] - p(w)\right) \\ &= -q(w)\mathbf{1}[w \in \mathcal{A}] + Qp(w) \\ &= \begin{cases} Qp(w) - q(w), & w \in \mathcal{A}, \\ Qp(w), & w \notin \mathcal{A}. \end{cases}\end{aligned} \tag{29}$$

C.2 Stationary Target Probability Under the Four Designs

This section compares the four combinations of target-token removal and local normalization by deriving the stationary target probability under each design. We use $\mathcal{L}_{\text{pres}}$ to denote the preservation term of the design under consideration. In every case,

$$\frac{\partial \mathcal{L}_{\text{CE}}}{\partial z(y_t)} = p(y_t) - 1.$$

We add the corresponding preservation gradient and solve the logit-level stationarity condition for $p^\star(y_t)$.

(i) Remove, normalized. Here $y_t \notin \mathcal{A}$, so Eq. (27) gives $\partial \mathcal{L}_{\text{pres}}/\partial z(y_t) = 0$. The stationarity condition reduces to $p(y_t) - 1 = 0$, and hence $p^\star(y_t) = 1$ for every μ.

(ii) Retain, normalized. Now $\mathcal{A} = S$ and $y_t \in S$. Writing $\rho = p(S)$ gives $\hat{p}(y_t) = p(y_t)/\rho$, so stationarity requires

$$p(y_t) - 1 + \mu\left(\frac{p(y_t)}{\rho} - \hat{q}(y_t)\right) = 0.$$

Solving for the target probability yields

$$p^\star_{\text{pres}}(y_t) = \frac{1 + \mu\hat{q}(y_t)}{1 + \mu/\rho}$$

Since $\rho \le 1$,

$$p^\star_{\text{pres}}(y_t) \le \frac{1 + \mu\hat{q}(y_t)}{1 + \mu} \le 1.$$

This proves Eq. (14). The final inequality is strict whenever $\hat{q}(y_t) < 1$; the exact stationary value equals 1 only in the degenerate case $\rho = \hat{q}(y_t) = 1$. At a plateau-k position with $k > 1$, $\hat{q}(y_t) \approx 1/k < 1$, so retaining the target strictly caps its stationary probability.

(iii) Remove, unnormalized. Although $y_t \notin \mathcal{A}$, Eq. (29) gives $\partial \widetilde{\mathcal{L}}/\partial z(y_t) = Qp(y_t) > 0$, where $Q = q(\mathcal{A})$. Therefore

$$p(y_t) - 1 + \mu Q p(y_t) = 0, \qquad p^\star(y_t) = \frac{1}{1 + \mu Q} < 1.$$

Removing the target is insufficient: the unnormalized objective suppresses all tokens outside $\mathcal{A}$.

(iv) Retain, unnormalized. With $\mathcal{A} = S$, Eq. (29) gives

$$\frac{\partial \widetilde{\mathcal{L}}}{\partial z(y_t)} = -q(y_t) + q(S)p(y_t), \qquad p^\star(y_t) = \frac{1 + \mu q(y_t)}{1 + \mu q(S)}.$$

This is strictly below 1 whenever S contains nonzero base-model mass beyond the target.

Only case (i) guarantees the unmodified cross-entropy fixed point for every nonempty preservation set. Each incomplete design caps the target probability whenever the set contains nonzero non-target mass. The value $p^\star(y_t) = 1$ expresses compatibility with the cross-entropy objective, not a guarantee of deterministic predictions after training. It means that the preservation term does not attenuate the direct supervised signal at the target logit. Because both losses propagate through shared parameters, indirect interactions may remain.

Table 14: **Stationary target probability under each design.** Only the complete design—target-token removal with local normalization—preserves the cross-entropy fixed point $p_t^\star(y_t) = 1$; each incomplete variant caps the target probability below one. Exemplary values are computed with $\mu = 1$ under an idealized plateau-3 state ($q_t(y_t) = 1/3$, $\rho_t = 1$, and $q_t(\mathcal{A}_t) = 2/3$) to illustrate the relative capping effect.

Remove y_t	**Normalized KL**	$\partial \mathcal{L}^t_{\text{pres}} / \partial z_t(y_t)$	$p_t^\star(y_t)$	**Value**
Yes	Yes	0	1	1.00
No	Yes	$\hat{p}_t(y_t) - \hat{q}_t(y_t)$	$\dfrac{1 + \mu\, \hat{q}_t(y_t)}{1 + \mu/\rho_t}$	0.67
Yes	No	$q_t(\mathcal{A}_t)\, p_t(y_t)$	$\dfrac{1}{1 + \mu\, q_t(\mathcal{A}_t)}$	0.60
No	No	$-q_t(y_t) + q_t(S_t)\, p_t(y_t)$	$\dfrac{1 + \mu\, q_t(y_t)}{1 + \mu\, q_t(S_t)}$	0.67

D LP-SFT Algorithm Pseudocode

Algorithm 1 presents the complete pseudocode for LP-SFT, detailing the two-stage implementation: an offline stage to cache base model predictions, and an online stage for efficient fine-tuning using only the trainable model.

Algorithm 1 LP-SFT Training

Require: Base model θ_0, training dataset $\mathcal{D}$, support size $K_{\max}$, weight μ
// Stage 1: Offline Base Precomputation
for each sample $(\mathbf{x}, \mathbf{y}) \in \mathcal{D}$ **do**
  Compute $q_t = p_{\theta_0}(\cdot \mid \mathbf{x}, \mathbf{y}_{<t})$ and cache top-$K_{\max}$ candidate logits for all valid t
end for
// Stage 2: Local-Preserving Supervised Fine-Tuning
for each mini-batch from $\mathcal{D}$ **do**
  $\mathcal{L} \leftarrow 0$
  for each target position t **do**
    Load cached base logits, set $S_t = C_{K_{\max}}(q_t)$, and construct $\mathcal{A}_t$ via Eq. (7)
    $\mathcal{L}^t_{\text{LP}} \leftarrow 0$
    if $\mathcal{A}_t \neq \emptyset$ **then**
      $\hat{q}_t, \hat{p}_t \leftarrow \text{Normalize}(q_t|_{\mathcal{A}_t}), \text{Normalize}(p_t|_{\mathcal{A}_t})$
      $\mathcal{L}^t_{\text{LP}} \leftarrow \sum_{v \in \mathcal{A}_t} \hat{q}_t(v) \ln \frac{\hat{q}_t(v)}{\hat{p}_t(v)}$
    end if
    $\mathcal{L} \leftarrow \mathcal{L} - \ln p_t(y_t) + \mu \cdot \mathcal{L}^t_{\text{LP}}$
  end for
  Update parameters θ using $\nabla_\theta \mathcal{L}$
end for

E Training and Evaluation Details

E.1 Training Details

All models are fine-tuned using the HuggingFace Transformers library with DeepSpeed for memory-efficient distributed training. We use ZeRO Stage-2 by default, and ZeRO Stage-3 for ASFT, which keeps a frozen reference model in memory during training. We use the AdamW optimizer with a learning rate of 2×10^{-5}, a cosine learning-rate schedule, and a warmup ratio of 0.03. The weight decay is set to 0.0, and we use $\beta_2 = 0.95$ for AdamW. All models are trained for 3 epochs in bfloat16 precision with gradient checkpointing enabled. The maximum sequence length is set to 2048 tokens, and the random seed is fixed to 1234 for reproducibility. We use an effective batch size of 128 across all experiments, adjusting the per-device micro-batch size and gradient accumulation steps according to model scale and method-specific memory usage (e.g., a smaller micro-batch for Qwen3-14B and for ASFT). All experiments are conducted on nodes equipped with 8× NVIDIA A100 (80GB) GPUs.

Table 15: **Base model reference scores.** Qwen3-14B provides the strongest pretrained reference across the evaluation suite, followed by Qwen3-4B and Llama-3.1-8B.

Model	Math				Code				GEN.	Avg.
	MATH-500		AIME 21–26		MBPP+		HE+		MMLU	
	pass@1	pass@16	pass@1	pass@32	pass@1	pass@10	pass@1	pass@10	5-shot Acc.	All
Qwen3-4B-Base	34.58	80.80	5.71	23.89	59.95	82.29	67.74	90.83	73.02	57.65
Qwen3-14B-Base	69.50	89.40	7.00	26.11	69.62	87.88	77.74	93.32	80.51	66.79
Llama-3.1-8B	8.64	44.20	0.03	1.11	42.74	72.12	27.41	66.21	65.28	36.42

Baselines. For the GEM baseline, we set the temperature parameter to $\beta = 0.7$. For EAFT, we use $\alpha = 1.0$ and top-$k = 20$. For ASFT, we set the weight of the full-vocabulary KL loss to 0.05. These hyperparameter choices follow the recommended settings in the respective original papers.

E.2 Evaluation Details

General evaluation protocol. All SFT models are evaluated using the chat prompt template of their respective model family (ChatML for Qwen3 and the Llama-3.1 instruct format for Llama-3.1-8B), with thinking mode disabled for Qwen3 models (`enable_thinking=False`) to ensure a fair comparison under standard non-thinking generation. For base models, we use the corresponding base-model prompting format when evaluating generation tasks, without applying instruction-tuned chat-specific behavior. MMLU is the sole exception, where all models are evaluated with the base non-chat prompt format to avoid formatting artifacts in log-likelihood scoring. We employ vLLM as the inference backend across all benchmarks to accelerate evaluation.

Mathematical reasoning. For MATH-500, we use temperature $T = 0.6$, $n = 16$ samples, `top_p` $= 1.0$, and a maximum generation length of 4096 tokens (`max_model_len` $= 8192$). We use a boxed-answer prompt template, extract final answers from `\boxed{}`, and report pass@1 and pass@16. For AIME, we use temperature $T = 0.6$, $n = 32$ samples, `top_p` $= 1.0$, a maximum generation length of 8192 tokens, and `max_model_len` $= 12288$. The AIME evaluation is conducted on 180 problems collected from AIME competitions from 2021 to 2026, and we report pass@1 and pass@32.

Code generation. HumanEval+ and MBPP+ are evaluated using the EvalPlus framework. We use temperature $T = 0.8$ and generate $n = 20$ samples per problem. Unless otherwise noted, we report unbiased pass@1 and pass@10 on the plus tests with additional hidden test cases.

General knowledge and reasoning. For general knowledge and reasoning, we evaluate MMLU in the 5-shot setting using log-likelihood scoring with the lm-evaluation-harness framework.

F Additional Results

F.1 Overview

We first report the pretrained reference performance of Qwen3-4B, Qwen3-14B, and Llama-3.1-8B in Table 15, providing the baseline needed to assess capability gains and forgetting after fine-tuning. Tables 16 and 17 then present the complete fine-tuning results on Magicoder-OSS-Instruct-75K and NuminaMath-CoT, respectively, complementing the UltraFeedback results in the main text. Finally, Table 18 evaluates whether LP-SFT continues to provide a favorable accuracy–preservation trade-off when scaled to Qwen2.5-72B, while Table 19 compares the corresponding LoRA fine-tuning and cache-construction costs. All evaluations follow the main-text protocol.

F.2 Scaling to Qwen2.5-72B

To examine whether LP-SFT remains effective at a substantially larger scale, we fine-tune Qwen2.5-72B-Base on a 30K subset of UltraFeedback. For a controlled comparison, all methods use rank-64 LoRA on the attention and MLP projections and share the same three-epoch optimization setup with

Table 16: **Fine-tuning results on Magicoder-OSS-Instruct-75K.** LP-SFT achieves the highest overall average for every model family, driven by strong finite-budget performance and consistent preservation of MMLU accuracy.

Model	Method	Math				Code				GEN.	Avg.
		MATH-500		AIME 21–26		MBPP+		HE+		MMLU	
		pass@1	pass@16	pass@1	pass@32	pass@1	pass@10	pass@1	pass@10	5-shot Acc.	All
Qwen3-4B	CE	34.59	78.80	2.12	13.89	63.16	75.73	71.13	82.41	71.87	54.86
	DFT	54.55	70.80	2.48	13.89	61.69	65.77	64.27	68.53	71.61	52.62
	EAFT	44.51	81.40	2.76	20.00	63.36	77.26	**71.25**	86.68	72.10	57.70
	GEM	32.31	77.80	2.31	16.67	62.71	78.93	68.87	87.42	71.76	55.42
	ASFT	53.94	81.40	4.24	19.44	**64.25**	73.17	68.78	81.54	72.30	57.67
	LP-SFT	**55.44**	**84.80**	**5.19**	**21.67**	62.01	**81.74**	70.09	**89.97**	**72.82**	**60.41**
Qwen3-14B	CE	47.35	82.20	4.01	20.00	69.58	79.94	**76.59**	86.87	79.90	60.72
	DFT	**65.34**	78.80	4.81	12.22	66.92	70.95	68.93	72.04	80.01	57.78
	EAFT	22.86	81.60	1.77	16.67	70.98	83.35	75.37	88.29	80.12	57.89
	GEM	52.16	86.20	2.92	16.67	68.89	84.77	74.76	88.25	79.80	61.60
	ASFT	63.54	83.20	5.14	21.11	70.05	78.78	76.25	84.89	80.11	62.56
	LP-SFT	55.07	**86.60**	**5.54**	**23.33**	**71.31**	**85.40**	76.22	**91.39**	**80.61**	**63.94**
Llama-3.1-8B	CE	7.34	33.40	0.05	1.67	**47.63**	63.11	**41.92**	62.05	61.69	35.43
	DFT	6.01	15.80	0.00	0.00	30.70	35.98	22.74	25.81	59.44	21.83
	EAFT	6.94	33.60	0.02	0.56	42.01	60.73	37.41	62.01	60.60	33.76
	GEM	6.20	34.20	**0.14**	**3.89**	46.49	65.39	39.91	**64.67**	61.27	35.80
	ASFT	6.53	27.60	0.07	2.22	43.36	57.21	31.83	48.05	60.77	30.85
	LP-SFT	**8.28**	**38.60**	0.07	2.22	45.70	**66.78**	37.62	62.67	**62.06**	**36.00**

Table 17: **Fine-tuning results on NuminaMath-CoT.** LP-SFT achieves the highest overall average for all three model families, combining strong mathematical reasoning with better retention on code and general knowledge.

Model	Method	Math				Code				GEN.	Avg.
		MATH-500		AIME 21–26		MBPP+		HE+		MMLU	
		pass@1	pass@16	pass@1	pass@32	pass@1	pass@10	pass@1	pass@10	5-shot Acc.	All
Qwen3-4B	CE	57.30	86.00	4.11	21.11	60.71	78.90	**73.11**	90.05	71.89	60.35
	DFT	40.36	68.60	2.20	11.67	**64.72**	70.44	71.01	81.23	71.36	53.51
	EAFT	47.61	84.60	3.92	17.78	61.18	80.74	72.71	**91.46**	71.72	59.08
	GEM	51.92	86.00	3.00	20.56	56.87	79.00	70.03	89.93	72.30	58.85
	ASFT	**61.54**	83.80	4.06	16.67	63.73	77.98	71.31	86.75	72.59	59.83
	LP-SFT	60.71	**86.20**	**5.31**	**22.78**	59.76	**81.02**	71.34	91.42	**72.67**	**61.25**
Qwen3-14B	CE	64.50	**90.80**	5.85	26.11	68.78	85.67	78.60	93.11	79.22	65.85
	DFT	48.30	76.60	3.07	10.56	**72.80**	78.06	**79.27**	84.89	79.55	59.23
	EAFT	58.80	88.20	4.06	25.00	69.55	86.30	79.15	92.44	79.08	64.73
	GEM	61.48	90.00	4.95	26.11	63.65	86.22	75.95	92.91	79.43	64.52
	ASFT	66.71	88.40	5.45	21.67	72.71	83.05	78.87	90.85	79.93	65.29
	LP-SFT	**66.92**	90.20	**7.19**	**29.44**	69.58	**86.60**	77.23	**94.03**	**80.34**	**66.84**
Llama-3.1-8B	CE	**31.67**	70.00	**0.61**	10.00	30.09	60.80	24.09	58.85	60.61	38.52
	DFT	13.90	27.60	0.10	2.22	21.75	32.17	8.41	15.89	55.14	19.69
	EAFT	29.43	67.60	0.57	**12.22**	31.94	63.47	22.96	57.43	60.00	38.40
	GEM	27.12	**70.60**	0.52	10.56	25.50	59.78	20.40	52.43	60.55	36.38
	ASFT	26.38	60.40	0.54	7.78	31.87	57.37	22.29	48.43	61.07	35.13
	LP-SFT	25.42	66.20	0.31	8.33	**36.10**	**65.92**	**27.10**	**61.91**	**62.49**	**39.31**

AdamW, a learning rate of 2×10^{-4}, and an effective batch size of 128. For ASFT, we use the LoRA-ASFT variant proposed in the original paper (Zhu et al., 2025). LP-SFT uses $\mu = 1$ and a cached top-10 non-label reference set, avoiding a frozen reference model during training. As shown in Table 18, LP-SFT achieves the best aggregate performance among the fine-tuned methods, leading on MATH-500 and AIME while retaining high pass@k on both code benchmarks. Although ASFT is stronger on code pass@1, LP-SFT provides a better overall balance and slightly exceeds the base model in average performance.

Training time. Table 19 reports cost on 8× A100 (80GB) GPUs, excluding evaluation. LP-SFT matches the fine-tuning cost of CE. Constructing its top-10 cache for the 30K training examples takes

Table 18: **Fine-tuning results on Qwen2.5-72B-Base with UltraFeedback-30K.** LP-SFT attains the best aggregate score among fine-tuned methods, leading on both mathematical benchmarks and on pass@k for both code benchmarks.

Model	Method	Math				Code				GEN.	Avg.
		MATH-500 pass@1 / pass@16		AIME 21–26 pass@1 / pass@32		MBPP+ pass@1 / pass@10		HE+ pass@1 / pass@10		MMLU 5-shot Acc.	All
Qwen2.5-72B	Base	47.94	85.60	4.60	22.22	70.60	87.50	74.60	92.00	86.01	63.45
	CE	41.81	79.60	1.72	17.22	48.10	78.70	50.50	85.00	83.68	54.04
	DFT	52.89	70.20	2.71	10.00	63.70	67.20	70.60	76.40	83.21	55.21
	EAFT	42.74	79.80	1.79	16.67	49.50	78.30	51.90	86.00	84.15	54.54
	GEM	36.21	76.40	1.25	15.00	36.20	75.00	36.90	80.90	83.34	49.02
	ASFT	55.55	81.40	3.23	17.22	**69.40**	78.90	**73.10**	85.30	85.54	61.07
	LP-SFT	**55.93**	**88.40**	**5.64**	**25.00**	66.10	**86.10**	69.60	**91.40**	**85.66**	**63.76**

11,232 seconds on the same eight GPUs, or 25.0 A100 GPU-hours. This is a one-time preprocessing cost: the cache is reused across epochs, fine-tuning runs, and ablations with the same base model and data.

Table 19: **Qwen2.5-72B LoRA cost on UltraFeedback-30K.** LP-SFT incurs negligible fine-tuning overhead over CE. Its first-run total includes the one-time reusable top-10 cache construction.

Method	One-time cache GPU-hours	Fine-tuning GPU-hours	First-run total GPU-hours
CE	—	120.0	120.0
DFT	—	120.7	120.7
EAFT	—	119.8	119.8
GEM	—	118.2	118.2
ASFT	—	238.4	238.4
LP-SFT	25.0	120.2	145.2

G Additional Ablation and Efficiency Results

This section analyzes sensitivity to the preservation weight μ, the design principles, and support size K_t. All experiments use Qwen3-4B-Base fine-tuned on UltraFeedback.

G.1 Sensitivity to the Preservation Weight

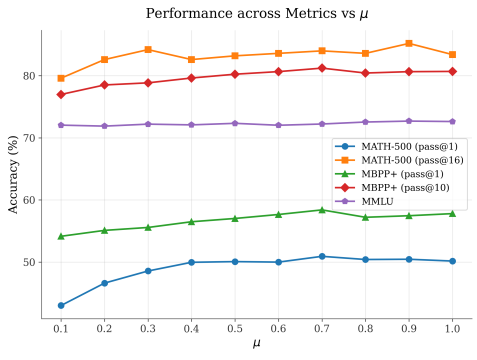


Figure 6: **Sensitivity to the preservation weight μ.** Results are obtained with Qwen3-4B fine-tuned on UltraFeedback and evaluated on MATH-500, MBPP+, and MMLU. Performance varies only slightly across $\mu \in [0.1, 1.0]$, suggesting that LP-SFT is relatively insensitive to the choice of μ.

Figure 6 studies the effect of the preservation weight μ. Across the tested range, LP-SFT exhibits only mild performance variation on MATH-500, MBPP+, and MMLU, with some metrics showing a

Table 20: **Ablation of LP-SFT design principles on Qwen3-4B trained with UltraFeedback.** The complete design achieves the highest aggregate score; any incomplete variant (omitting target-token removal, local normalization, or both) underperforms the full objective (56.77–57.49 vs. 58.79).

Remove y_t	Normalized KL	Math				Code				GEN.	Avg.
		MATH-500 pass@1	MATH-500 pass@16	AIME 21–26 pass@1	AIME 21–26 pass@32	MBPP+ pass@1	MBPP+ pass@10	HE+ pass@1	HE+ pass@10	MMLU 5-shot Acc.	All
No	Yes	45.99	84.20	3.52	20.00	54.30	80.04	64.88	89.22	72.35	57.17
Yes	No	45.71	**84.80**	**3.61**	20.00	51.57	80.08	64.21	88.80	72.14	56.77
No	No	48.25	84.00	**3.61**	**20.56**	54.17	80.28	64.73	89.59	72.20	57.49
Yes	Yes	**50.15**	83.40	3.39	19.44	**57.79**	**80.69**	**69.30**	**92.29**	**72.62**	**58.79**

Table 21: **Ablation on support size K_t (Qwen3-4B-Base, UltraFeedback, $\mu = 1.0$).** Fixed top-10 preservation comes within 0.22 average points of full-vocabulary preservation while using only an offline cache rather than an in-memory reference model.

Support schedule	Base access	Math				Code				GEN.	Avg.
		MATH-500 pass@1	MATH-500 pass@16	AIME 21–26 pass@1	AIME 21–26 pass@32	MBPP+ pass@1	MBPP+ pass@10	HE+ pass@1	HE+ pass@10	MMLU 5-shot Acc.	All
N_1-adaptive $K_t = \lceil N_1(q_t) \rceil$	Cache	48.77	83.20	2.74	**19.44**	55.69	79.17	68.93	89.41	72.28	57.74
Fixed top-10 (default)	Cache	50.15	83.40	**3.39**	**19.44**	57.79	80.69	**69.30**	**92.29**	72.62	58.79
Full non-label vocabulary	Ref. model	**52.35**	**85.00**	**3.39**	**19.44**	**58.57**	**81.60**	67.96	90.10	**72.67**	**59.01**

slight improvement as μ increases. This indicates limited sensitivity to μ within the tested range. We therefore use $\mu = 1$ as the default setting in our main experiments.

G.2 Ablation of the Design Principles

Table 20 evaluates the two design principles on Qwen3-4B-Base trained with UltraFeedback by varying whether the supervised token y_t is removed from the preservation set $\mathcal{A}_t$ and whether locally normalized probabilities are used in the KL loss. Any incomplete configuration falls notably short of the complete objective (56.77–57.49 vs. 58.79), confirming that both design principles are necessary to realize the full preservation benefit. This empirical observation aligns with the gradient analysis in Section 5.2 and the target-logit stationary points in Table 14. Overall, target-token removal and local normalization determine *whether* preservation helps; we study *how broadly* the constraint is applied next.

G.3 Ablation of the Support Size K_t

Having fixed the two design principles, we next vary only the breadth of the candidate set $S_t = C_{K_t}(q_t)$ in Eq. (7). Table 21 compares three choices of K_t under an otherwise identical setup (Qwen3-4B-Base on UltraFeedback, $\mu = 1.0$, both principles applied, matched decoding):

- **N_1-adaptive:** $K_t = \lceil N_1(q_t) \rceil$, a per-token support matched to the estimated plateau order;
- **Fixed top-10 (default):** $K_t = K_{\max} = 10$, covering the low-order plateaus and the prominent higher-order regime around $k = 10$ identified in Section 4.2;
- **Full vocabulary:** $S_t = \mathcal{V}$ and $\mathcal{A}_t = \mathcal{V} \setminus \{y_t\}$, covering all non-label tokens.

The three rows are ordered by how much of the base distribution they constrain, and they probe the two ends of the range separately.

Widening from fixed top-10 to the full vocabulary changes the average by only 0.22 points (58.79 vs. 59.01), far less than the degradation caused by dropping either design principle in Table 20. This is what the probability weighting in Eq. (9) predicts: the additional tokens lie well outside the plateau, so the base model assigns them negligible weight and they can neither help nor hurt much. The costs differ sharply, however, since full-vocabulary preservation requires an in-memory frozen reference

model throughout training (similar to ASFT (Zhu et al., 2025)), whereas fixed top-10 preservation relies entirely on a reusable offline cache (Table 19).

Narrowing behaves differently. The N_1-adaptive schedule ties the budget to the estimated plateau order at each position and is the weakest of the three (57.74). For confident predictions with $N_1(q_t) \leq 2$, which constitute the majority of token positions, it sets $K_t \leq 2$; whenever the supervised token y_t falls within this small candidate set, the preservation set $\mathcal{A}_t = S_t \setminus \{y_t\}$ contains at most one token and the locally normalized KL loss vanishes identically. An entropy-indexed budget thus switches preservation off exactly where the model is confident. This is consistent with how we read Section 4.2: plateau order indicates how many alternatives are clearly plausible, not how many are worth preserving, and the loss already down-weights implausible candidates on its own. Per-token adaptation therefore adds a moving part without adding selectivity, and we adopt the fixed $K_t = K_{\max} = 10$ as the default in LP-SFT.

H Setup of the Collapse-Rate Diagnostic

We compute the collapse rate (Eq. (15)) for the UltraFeedback-trained checkpoints in Table 2, each compared against its own base model. For each of the first 300 MATH-500 problems, the base model generates one solution (same prompt as in Appendix E, temperature 0.6, at most 1024 new tokens). The base model and each fine-tuned checkpoint then score the same problem–solution sequences, so q_t and p_t always share the same context. Special tokens are excluded, $\mathcal{A}_t$ follows Eq. (7), and all probabilities are full-vocabulary without local normalization. The threshold $q_t(v) \geq 0.05$ keeps only alternatives the base model finds plausible. The threshold $p_t(v) < 10^{-3}$ marks a token that would be sampled fewer than once in 1,000 draws, far beyond our pass@k budgets ($k \leq 32$). Table 22 reports the number of positions and endorsed alternatives.

Table 22: Number of scored response positions and base-endorsed alternatives ($q_t(v) \geq 0.05$, $v \in \mathcal{A}_t$) in the collapse-rate diagnostic, computed on 300 base-generated MATH-500 solutions per model family.

Base model	Positions	Endorsed alternatives
Qwen3-4B	87,151	40,227
Qwen3-14B	135,365	39,001
Llama-3.1-8B	111,039	53,935