Which Tokens Should SFT Actually Learn? A Token-Trimming Perspective on Mathematical Reasoning

Anonymous ACL submission

Abstract

Supervised fine-tuning (SFT) applies a uniform cross-entropy loss to all target tokens, even though different tokens provide unequal learning signals for mathematical reasoning. This uniform treatment can over-sharpen already mastered tokens while amplifying learning pressure on uncertain, low-confidence tokens, leading to suboptimal training dynamics. We propose **Trimmed Logit-Gap SFT (TrimSFT)**, a simple token-level reweighting method that scales the SFT loss according to the logit gap between the gold token and its strongest competitor. TrimSFT trims supervision away from both extremes: tokens already mastered (large logit gap) and tokens weakly supported by the current model (small or negative logit gap), concentrating learning on the middle region between them. We instantiate this principle with a Gaussian weight centered at margin m with bandwidth τ, requiring no reference model or additional forward pass. We evaluate TrimSFT on six base models from the Llama, Qwen, and DeepMath families across five mathematical reasoning benchmarks. TrimSFT consistently improves over standard SFT, achieving the best average performance on five out of six models, with gains of up to $+26.9$ points over SFT on MATH500. Further analyses show that the bandwidth τ matters more than the exact margin location, and that half-trim variants that remove supervision pressure from only one side yield inferior trade-offs. A token-level logit-gap distribution analysis suggests that TrimSFT reshapes model confidence in a more balanced way than uniform SFT or monotonic reweighting methods. These results indicate that effective reasoning SFT should trim both extremes rather than treating all tokens uniformly.

1 Introduction

Large language models (LLMs) have demonstrated strong capabilities on complex reasoning tasks (Wei et al., 2022b; Wang et al., 2024; Fan et al., 2024), with mathematical reasoning serving as a key testbed for studying multi-step inference (Jia et al., 2025; Liu et al., 2025). Supervised fine-tuning (SFT) is widely used to adapt pretrained models to mathematical reasoning data (Yu et al., 2024; Yue et al., 2024), and often serves as an initialization stage for downstream reinforcement learning or preference optimization (Ouyang et al., 2022; Bai et al., 2022).

Despite its effectiveness, standard SFT applies token-level cross-entropy with uniform weighting across all target tokens (Lin et al., 2026), ignoring that different tokens can provide very different learning signals (Wu et al., 2026; Gong et al., 2025). In reasoning trajectories, some tokens may already be well mastered and continue to receive unnecessary sharpening pressure, which can contribute to over-confidence (Pereyra et al., 2017; Chen et al., 2025; Wei et al., 2022a). Other tokens may be highly uncertain, noisy, or beyond the model's current capability, yet still induce large losses and dominate the optimization signal. These two extremes suggest that treating all tokens uniformly can pull learning away from the most informative region (Wu et al., 2026; Lin et al., 2017; Han et al., 2018). Effective reasoning SFT should instead trim both extremes and focus supervision on tokens near the boundary of the model's current capability.

Motivated by this intuition, we propose **Trimmed Logit-Gap SFT (TrimSFT)**, a token-level reweighting method that trims supervision away from both extremes: tokens already mastered (large logit gap) and tokens weakly supported by the current model (small or negative logit gap). For each target token, we compute its *logit gap*, defined as the margin between the gold token's logit and that of its strongest competitor, and use a Gaussian weight centered at margin m with bandwidth τ to scale the cross-entropy loss (Figure 1, left). Tokens whose logit gaps lie near m receive stronger supervision, while tokens with much smaller or larger

gaps are softly down-weighted so that learning concentrates on the middle region between them. The weight is computed from the model's own logits in the same forward pass, requiring no reference model or additional forward pass. To examine the contribution of each side of this two-sided trimming profile, we further introduce two half-trim variants (Figure 1, right): **Trim-Easy SFT (TrimSFT-E)**, which trims the hard side only and preserves full weight on high-gap (easy) tokens, and **Trim-Hard SFT (TrimSFT-H)**, which trims the easy side only and preserves full weight on low-gap (hard) tokens.

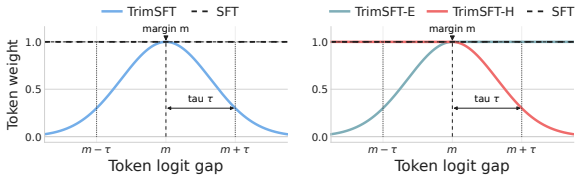

Figure 1: Token weighting functions for SFT, TrimSFT, and its half-trim variants TrimSFT-E and TrimSFT-H. The margin m determines the center of the weighting function, and τ controls the width of the weighted region.

We evaluate TrimSFT on six base models from the Llama, Qwen, and DeepSeekMath families across five mathematical reasoning benchmarks. Our results show that TrimSFT consistently improves over standard SFT, and we further analyze the mechanism behind these gains. Our contributions are summarized as follows:

- We introduce **TrimSFT**, a token-level reweighting method that scales the SFT loss by the logit gap between the gold token and its strongest competitor. It requires no reference model or additional forward pass.
- TrimSFT achieves the best average performance on five out of six models, with gains of up to $+26.9$ points over SFT on MATH500. It also improves capability coverage and self-consistency under repeated sampling, as measured by pass@8 and best-of-8, respectively.
- Ablations show that performance is more sensitive to the bandwidth τ than to the margin m, suggesting that the width of the selected logit-gap region is more important than its center.
- Half-trim variants show that trimming only one side yields inferior trade-offs: TrimSFT-H collapses below SFT, and TrimSFT-E degrades on the hardest problems. Token-level logit-gap distribution analysis further shows that TrimSFT reshapes model confidence in a more balanced way than uniform SFT or monotonic reweighting methods.

2 Related Work

Supervised Fine-Tuning for Mathematical Reasoning. Mathematical reasoning has emerged as a central testbed for evaluating the multi-step inference capabilities of large language models (Cobbe et al., 2021; Hendrycks et al., 2021; He et al., 2024; Saxton et al., 2019; Lewkowycz et al., 2022). A common recipe for adapting pretrained models to mathematical tasks is supervised fine-tuning (SFT) on curated reasoning trajectories (Yu et al., 2024; Yue et al., 2024; Toshniwal et al., 2024; Li et al., 2024). Much of the progress in this line has come from data-centric improvements, including synthesizing high-quality chain-of-thought solutions (Yu et al., 2024; Luo et al., 2025), distilling from stronger teachers (Shao et al., 2024; Yang et al., 2024), incorporating recent long chain-of-thought trajectories from reasoning-specialized models (Guo et al., 2025; Hugging Face, 2025; Ye et al., 2025), and constructing curated datasets through difficulty- or correctness-based filtering (Toshniwal et al., 2024; Li et al., 2024). Despite these advances, the underlying objective typically remains the standard token-level cross-entropy loss applied uniformly to every token. In contrast, our work targets the per-token objective rather than the training data, making it complementary to existing data-centric approaches.

Token-Level Reweighting and Selection in SFT. While vanilla SFT applies a uniform cross-entropy loss across all tokens, recent work has explored token-level reweighting or selection to account for differences in training value across tokens (Lin et al., 2024; Ruan et al., 2025; Wu et al., 2026). These methods differ mainly in the signals used to score tokens and in the form of loss modulation. Some approaches rely on auxiliary signals or pre-computed token masks, such as reference-model-based scoring in Rho-1 (Lin et al., 2024) and counterfactual selection in CFT (Ruan et al., 2025). DFT (Wu et al., 2026) and Focal Loss (Lin et al., 2017) both rescale token-level cross-entropy using the predicted probability of the gold token, though in opposite monotonic directions: DFT up-weights tokens already assigned high probability, while Focal Loss down-weights them. Our method departs

from these probability-based schemes in two ways: it uses the *logit gap* as a more scale-sensitive signal, and applies a Gaussian band-pass weighting that targets a bounded intermediate region rather than following a monotonic trend. We elaborate on this comparison in Section 3.3.

Logit Gaps, Margins, and Confidence Shaping. Margin-related quantities have been widely used to shape confidence and compare competing predictions, but they are rarely used directly as token-level supervision weights. In classification calibration, label smoothing (Müller et al., 2019; Pereyra et al., 2017) and margin-based label smoothing (Liu et al., 2022) shape confidence and margin behavior to mitigate overconfidence, while logit normalization (Wei et al., 2022a) and calibration-oriented training objectives (Guo et al., 2017) regulate confidence and logit magnitude during training. Related concerns have also been studied in large language models, where recent work examines confidence calibration and overconfidence in generated outputs (Zhang et al., 2024; Leng et al., 2024). In preference optimization, methods such as SimPO (Meng et al., 2024) use sequence-level logit margins between preferred and rejected outputs. These works use margin-related quantities either to regulate confidence or to define sequence-level preference signals. In contrast, TrimSFT uses the per-token logit gap as a supervision weight during SFT, trimming both extremes of the token-confidence spectrum rather than constraining or maximizing margins globally.

3 Method

We begin from the standard supervised fine-tuning (SFT) objective and then introduce TrimSFT, a token-level reweighting scheme driven by the model's own logit gaps. The key idea is to allocate stronger supervision to tokens whose logits place them near a bounded decision region, while reducing learning pressure on tokens that are already well mastered or currently beyond the model's effective reach. TrimSFT deliberately reduces supervision pressure at both ends of the logit-gap spectrum: tokens with very large gaps (already mastered) receive lower weight, as do tokens with very small or negative gaps (weakly supported by the current model).

Given an input prompt x and a target token sequence $y_1, \ldots, y_T$, standard SFT trains a model π_θ by minimizing the token-level cross-entropy loss

$$\mathcal{L}_{\text{SFT}} = -\sum_{t=1}^{T} \log \pi_\theta(y_t \mid x, y_{<t}), \tag{1}$$

which assigns uniform weight to every target token. Let $z_t \in \mathbb{R}^V$ denote the pre-softmax logits at position t, and let z_{t,y_t} denote the logit assigned to the gold token y_t. We define the *logit gap* at position t as

$$\Delta_t = z_{t,y_t} - \max_{v \neq y_t} z_{t,v}, \tag{2}$$

namely, the margin between the gold token's logit and that of its strongest competing token. A positive Δ_t indicates that the gold token is currently the top-1 prediction, while a larger magnitude reflects greater separation from the closest alternative. This quantity provides a local, token-level measure of how decisively the model distinguishes the correct token from competing candidates.

3.1 Trimmed Logit-Gap SFT

Our goal is to trim supervision away from both extremes of the logit-gap spectrum and concentrate learning on tokens whose gaps fall within an intermediate region. Intuitively, tokens with very large positive gaps are already well separated and may benefit little from continued sharpening, whereas tokens with very small or negative gaps may be highly uncertain, unstable, or beyond the model's current capability. We therefore seek a weighting function that peaks around a moderate logit-gap region and decays on both sides.

We instantiate this with a Gaussian weight on the logit gap:

$$w_t = \exp\left(-\frac{(\Delta_t - m)^2}{2\tau^2}\right), \tag{3}$$

where m is the center of the weighting function and τ controls the width of the high-weight region. Tokens whose logit gaps lie near m receive the largest weights, while tokens whose gaps are substantially smaller or larger are smoothly down-weighted. In this sense, m determines *where* learning should be concentrated, and τ determines *how broadly* that concentration should spread.

The resulting TrimSFT objective is

$$\mathcal{L}_{\text{TrimSFT}} = -\sum_{t=1}^{T} \text{sg}(w_t)\ \log \pi_\theta(y_t \mid x, y_{<t}), \tag{4}$$

where $\mathrm{sg}(\cdot)$ denotes stop-gradient. The weights w_t are computed from the same forward pass as the logits and then detached, so they act purely as a rescaling per-token of the cross-entropy loss without contributing a gradient through θ. This does not require any auxiliary model or extra forward pass, and adds only minimal computational overhead.

Compared with probability-based reweighting, the logit gap is particularly suitable in our setting for two reasons. First, it directly measures the separation between the gold token and its strongest competitor, making it naturally aligned with how close a token is to the model's decision boundary. Second, unlike the gold-token probability, which tends to saturate in high-confidence regimes, the logit gap remains discriminative even when probabilities would otherwise appear nearly indistinguishable. The Gaussian weighting then realizes a smooth two-sided trimming profile over token states: supervision is concentrated on tokens whose logit gaps lie near the chosen margin m, and decays on both sides. Unlike monotonic reweighting schemes that progressively emphasize either harder or easier tokens across the full confidence range, TrimSFT explicitly trims both extremes and focuses learning on a bounded intermediate region.

3.2 Half-Trim Variants

The weighting function in TrimSFT is symmetric around the margin m, assigning lower weights to tokens on both sides as their logit gaps move away from m. To isolate the contribution of each side, we define two half-trim variants that retain Gaussian decay on only one side while keeping full weight on the other (Figure 1, right).

TrimSFT-E. TrimSFT-E trims the hard side only: it keeps full weight on tokens whose logit gaps exceed the margin m, and applies Gaussian decay only on the lower-gap side:

$$w_t^{\mathrm{E}} = \begin{cases} \exp\left(-\dfrac{(\Delta_t - m)^2}{2\tau^2}\right), & \Delta_t < m, \\ 1, & \Delta_t \geq m. \end{cases} \tag{5}$$

This preserves full supervision for tokens already beyond the chosen margin, while down-weighting tokens with smaller gaps.

TrimSFT-H. TrimSFT-H trims the easy side only: it keeps full weight on tokens whose logit gaps fall below the margin m, and applies Gaussian decay only on the higher-gap side:

$$w_t^{\mathrm{H}} = \begin{cases} 1, & \Delta_t \leq m, \\ \exp\left(-\dfrac{(\Delta_t - m)^2}{2\tau^2}\right), & \Delta_t > m. \end{cases} \tag{6}$$

This preserves full supervision for tokens with smaller gaps, while down-weighting tokens that are already well separated.

Together, these variants retain the same margin m and bandwidth τ as TrimSFT, but isolate the contribution of each side of the weighting profile. Comparing them with full TrimSFT allows us to determine whether its gains arise primarily from trimming high-gap tokens, low-gap tokens, or both.

3.3 A Unified View of Token Reweighting

Standard SFT, TrimSFT, and several representative reweighting methods, including DFT (Wu et al., 2026) and a focal-loss-based SFT baseline (FSFT) (Lin et al., 2017), can all be written in a unified token-reweighted form:

$$\mathcal{L} = \sum_{t=1}^{T} \mathrm{sg}(w_t)\, \ell_t, \tag{7}$$

where $\ell_t = -\log \pi_\theta(y_t \mid x, y_{<t})$ is the standard token-level cross-entropy loss, $\mathrm{sg}(\cdot)$ denotes stop-gradient, and the methods differ only in the choice of w_t. Representative objectives then correspond to different weighting functions:

$$w_t = \begin{cases} 1, & \text{SFT}, \\ (1 - p_{t,y_t})^{\gamma}, & \text{FSFT}, \\ p_{t,y_t}, & \text{DFT}, \\ \exp\left(-\dfrac{(\Delta_t - m)^2}{2\tau^2}\right), & \text{TrimSFT}. \end{cases} \tag{8}$$

In this framework, standard SFT assigns equal training weight to every token. FSFT and DFT instead reshape the SFT objective using the gold-token probability: FSFT places larger weights on lower-confidence tokens, thereby emphasizing harder or less well-mastered positions, whereas DFT increases the relative contribution of higher-probability tokens to compensate for the implicit inverse-probability bias of standard cross-entropy. In contrast, TrimSFT uses the logit gap Δ_t and applies a non-monotonic weighting profile that trims both extremes while concentrating learning on tokens whose gaps lie near a prescribed margin m.

This unified view makes two differences especially clear. First, TrimSFT operates on the logit gap rather than the probability, so it directly tracks the separation between the gold token and its strongest competitor and avoids the saturation that compresses high-confidence tokens to nearly identical weights under probability-based schemes. Second, TrimSFT targets the middle region between mastered and currently unreachable tokens, whereas existing probability-based reweighting methods follow a single monotonic trend over the full confidence range.

4 Experiments

4.1 Setup

Dataset. `NuminaMath-CoT` (Li et al., 2024) is a large-scale math reasoning dataset containing about 860K problem-solution pairs with chain-of-thought annotations. For all experiments, we randomly sample 20,000 problems for supervised fine-tuning.

Models. We evaluate TrimSFT and the baseline methods on six base models from three families, spanning parameter scales from 1.5B to 8B: Llama3.2-3B (AI, 2024), Llama3.1-8B (Grattafiori et al., 2024), DeepSeekMath-7B (Shao et al., 2024), Qwen2.5-Math-1.5B, Qwen2.5-Math-7B (Yang et al., 2024), and Qwen3-4B-Base (Yang et al., 2025). We use only base models, rather than instruction-tuned variants, to reduce the influence of prior instruction tuning and enable a cleaner comparison of different supervised fine-tuning objectives.

Baselines. We compare **TrimSFT** with four representative baselines: (i) **Base**, the original pretrained model without supervised fine-tuning; (ii) **SFT**, standard supervised fine-tuning with uniform token-level weighting; (iii) **FSFT**, a focal-loss-based variant that upweights lower-confidence tokens during training (Lin et al., 2017); and (iv) **DFT** (Wu et al., 2026), a probability-based reweighting approach that assigns larger weights to higher-probability tokens.

Evaluation benchmarks. We evaluate all methods on five mathematical reasoning benchmarks spanning a range of difficulty levels: MATH500 (Hendrycks et al., 2021), OlympiadBench (He et al., 2024), Minerva (Lewkowycz et al., 2022), AMC (AI-MO, 2024b), and AIME24 (AI-MO, 2024a). For each benchmark, we sample $N = 8$ generations per problem and report **average@8** as the primary metric, which reflects the model's average generation quality. In Section 4.3, we additionally report **pass@8** and **best-of-8**, where majority voting is used as the selector, to characterize the model's exploration ability and self-consistency, respectively. Detailed evaluation settings are provided in Appendix A.

Implementation details. For TrimSFT, we use a single hyperparameter setting, $(m, \tau) = (1.5, 0.8)$, for all models and benchmarks in the main comparison (Table 1), without any model- or benchmark-specific tuning. All results are reported using the checkpoint obtained after one training epoch. We study the effects of varying m and τ in Sections 4.4 and 4.5, and provide additional training and evaluation details in Appendix A. Code is available at `https://anonymous.4open.science/r/TrimSFT-F727`.

4.2 Main Results

Table 1 reports the main results under the **average@8** accuracy across six base models and five mathematical reasoning benchmarks, with TrimSFT evaluated using $(m, \tau) = (1.5, 0.8)$.

Overall, TrimSFT consistently improves over standard SFT across all six models and achieves the best average performance on five out of six models. The gains are especially large on math-oriented models: for example, TrimSFT improves the average score from 15.47 to 31.77 on Qwen2.5-Math-1.5B and from 20.95 to 35.98 on Qwen2.5-Math-7B. On MATH500, TrimSFT is the top-performing method across all six models, with the largest gain over SFT reaching $+26.93$ points on Qwen2.5-Math-1.5B ($40.02 \rightarrow 66.95$). Compared with DFT, TrimSFT achieves stronger overall average performance on five of the six models, suggesting that logit-gap-based reweighting provides a useful alternative to probability-based token weighting. In contrast, FSFT performs poorly in most settings, indicating that simply emphasizing harder tokens is insufficient for reasoning SFT. These results provide initial evidence for our hypothesis that effective supervised fine-tuning for mathematical reasoning benefits from trimming both extremes of the logit-gap spectrum rather than treating all tokens uniformly.

Table 1: Average@8 accuracy (%) on math reasoning benchmarks. **Bold** marks the best result and underlined marks the second-best within each model block. Avg. denotes the mean across benchmarks.

Model	Method	MATH500	OlyBench	Minerva	AMC	AIME24	Avg.
Llama3.2-3B	Base	1.50	0.86	0.66	1.48	0.00	0.90
	SFT	4.80	1.56	1.65	1.48	0.00	1.90
	FSFT	1.90	0.96	0.83	1.19	0.00	0.98
	DFT	8.78	2.35	**3.60**	1.05	0.00	3.16
	TrimSFT	**9.45**	**2.81**	2.92	**3.53**	**2.00**	**4.14**
Llama3.1-8B	Base	2.10	0.93	1.65	1.20	0.00	1.18
	SFT	12.05	2.59	3.26	3.40	0.41	4.34
	FSFT	2.95	0.89	0.96	1.64	0.00	1.29
	DFT	19.77	4.89	**5.81**	4.73	0.83	7.21
	TrimSFT	**20.95**	**5.13**	5.52	**6.17**	**1.41**	**7.84**
DeepSeekMath-7B	Base	5.45	1.63	2.15	1.94	0.00	2.23
	SFT	23.40	5.52	7.36	7.08	0.41	8.75
	FSFT	6.80	1.70	2.26	2.80	0.00	2.71
	DFT	38.42	12.16	**13.30**	**13.23**	**1.24**	**15.67**
	TrimSFT	**38.80**	**12.94**	12.26	12.35	**1.24**	15.52
Qwen2.5-Math-1.5B	Base	31.45	16.23	8.45	13.68	3.32	14.63
	SFT	40.02	12.11	10.08	12.64	2.50	15.47
	FSFT	10.30	1.93	2.34	1.76	0.00	3.27
	DFT	63.17	26.90	20.55	**27.95**	7.93	29.30
	TrimSFT	**66.95**	**29.41**	**24.96**	**27.95**	**9.58**	**31.77**
Qwen2.5-Math-7B	Base	40.07	18.45	13.43	18.50	**11.66**	20.42
	SFT	50.65	16.91	17.46	18.09	1.66	20.95
	FSFT	18.12	3.06	4.02	4.12	0.41	5.95
	DFT	69.75	33.08	26.31	34.41	**11.66**	35.04
	TrimSFT	**71.15**	**33.61**	**28.88**	**37.48**	8.76	**35.98**
Qwen3-4B-Base	Base	34.35	17.96	11.76	17.06	4.15	17.06
	SFT	47.45	15.90	15.10	16.16	1.66	19.25
	FSFT	12.40	1.97	3.16	3.39	0.41	4.27
	DFT	61.65	**27.37**	22.88	24.74	**5.41**	28.41
	TrimSFT	**61.80**	26.08	**25.03**	**27.93**	4.15	**29.00**

4.3 Capability Ceiling and Self-Consistency

Beyond **average@8**, we further evaluate **pass@8** and **best-of-8** with majority voting to characterize two complementary aspects of model behavior. Pass@8 measures whether the model can produce at least one correct solution among multiple samples, reflecting its capability ceiling or exploration coverage. Best-of-8 with majority voting measures whether the model's sampled solutions consistently support the correct answer, reflecting the self-consistency of its generation distribution.

We report results on Qwen2.5-Math-1.5B and Llama3.1-8B in Figure 2. For both models, TrimSFT uses the same fixed hyperparameter setting $(m, \tau) = (1.5, 0.8)$ as in the main comparison in Table 1.

As shown in Figure 2, TrimSFT consistently improves pass@8 over standard SFT on both models across most benchmarks, indicating that two-sided logit-gap trimming expands the model's ability to discover correct solutions under repeated sampling. This suggests that TrimSFT does not merely optimize a single deterministic output, but improves the broader solution space explored by the model. The improvement also extends to best-of-8 with majority voting, where TrimSFT remains stronger than or competitive with the baselines across the two models. Since majority voting requires multiple samples to converge toward the correct answer, these gains indicate that the sampled solutions become more reliably aligned rather than only occasionally correct. Together with the average@8 results in Table 1, these findings show that TrimSFT improves average generation quality, capability coverage, and self-consistency under repeated sampling.

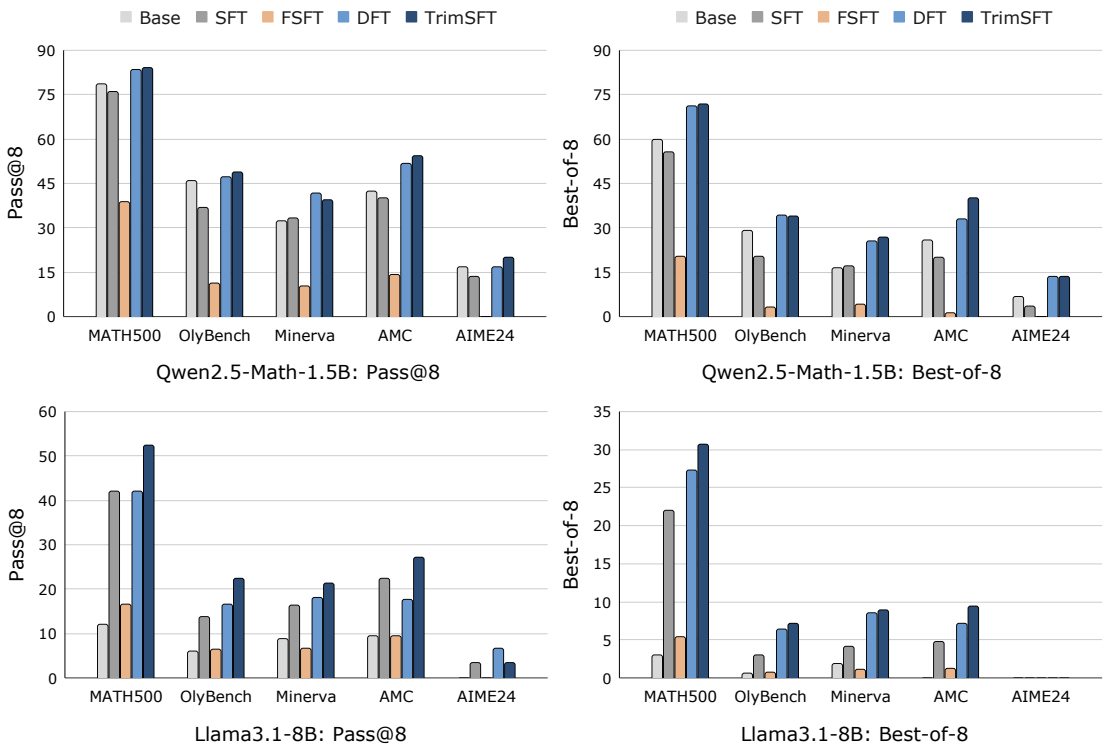

Figure 2: Pass@8 and best-of-8 with majority voting on Qwen2.5-Math-1.5B and Llama3.1-8B across five mathematical reasoning benchmarks.

4.4 Ablation: Margin and Bandwidth

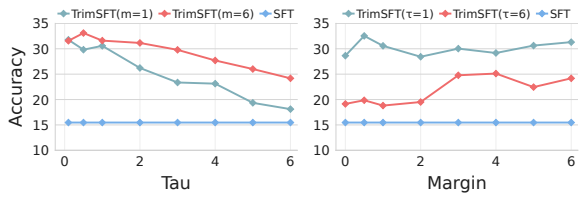

Figure 3: Ablation of margin m and bandwidth τ on Qwen2.5-Math-1.5B. Left: varying τ with fixed $m \in \{1, 6\}$; right: varying m with fixed $\tau \in \{1, 6\}$. The dashed line denotes the average SFT baseline.

We further study the effect of the two key hyperparameters in TrimSFT: the margin m, which determines the center of the weighted logit-gap region, and the bandwidth τ, which controls how broadly tokens around this margin are emphasized. Figure 3 reports average@8 accuracy averaged over the five benchmarks on Qwen2.5-Math-1.5B under different choices of m and τ, with the full numerical results provided in Appendix B.

Figure 3 shows that TrimSFT is more sensitive to the bandwidth τ than to the margin m. When m is fixed, increasing τ leads to a clear performance drop, especially for $m = 1$, indicating that an overly broad weighting function weakens the intended focus on boundary-region tokens. In contrast, when $\tau = 1$ is fixed, TrimSFT remains consistently strong across different values of m and stays well above the SFT baseline, with only moderate variation as the margin changes. This relative insensitivity to m holds primarily when the bandwidth is small: with $\tau = 6$, performance varies more visibly with m, whereas with $\tau = 1$ the curve remains comparatively stable. These results suggest that the width of the selected region is more important than its exact center: TrimSFT benefits most when supervision is concentrated within a relatively narrow band of logit gaps, under which the choice of m becomes less critical.

4.5 Half-Trim Variants

We compare TrimSFT with its two half-trim variants, TrimSFT-E and TrimSFT-H, to examine whether the full two-sided trimming profile is necessary. TrimSFT-E trims the hard side only and keeps easy tokens at full weight, whereas TrimSFT-H trims the easy side only and keeps hard tokens at full weight. To avoid drawing conclusions from a single margin choice, we evaluate the three variants under representative margins $m \in \{1, 3, 5\}$ while fixing $\tau = 1$. Table 2 reports results on three representative benchmarks.

Table 2: Comparison between TrimSFT and its half-trim variants on Qwen2.5-Math-1.5B with fixed $\tau = 1$ across representative margins $m \in \{1, 3, 5\}$. **Bold** marks the best result among the three variants.

m	Method	MATH500	AMC	AIME24
1	TrimSFT	62.80	31.74	**12.07**
	TrimSFT-E	**65.98**	**33.01**	7.51
	TrimSFT-H	28.75	8.84	0.41
3	TrimSFT	**65.98**	**30.29**	**7.50**
	TrimSFT-E	64.30	26.18	4.56
	TrimSFT-H	40.92	13.39	2.06
5	TrimSFT	65.96	31.03	**7.50**
	TrimSFT-E	**66.67**	**32.19**	6.25
	TrimSFT-H	42.50	16.03	1.65

Table 2 shows a consistent asymmetry between the two half-trim variants. TrimSFT-E is often competitive with or slightly stronger than TrimSFT on MATH500 and AMC, suggesting that preserving full supervision for higher-gap tokens can benefit relatively easier or medium-difficulty benchmarks. However, this advantage does not consistently transfer to the harder AIME24 benchmark, where TrimSFT achieves the best performance across all three reported margins. In contrast, TrimSFT-H performs substantially worse across all settings, mirroring the poor performance of FSFT in Table 1. Both half-trim variants leave one extreme at full weight, and TrimSFT provides a better trade-off by trimming both high-gap and low-gap extremes rather than relying on either half-trim alone.

5 Mechanism: Logit Gap Distribution Analysis

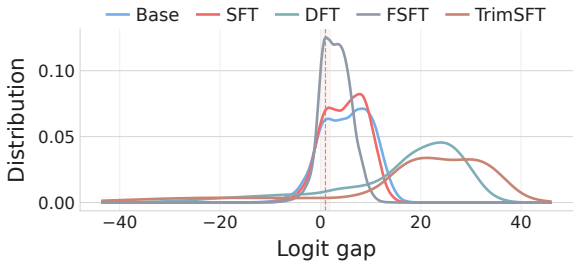

Figure 4: Logit-gap distributions on 100 randomly sampled training examples for Qwen2.5-Math-1.5B, aggregated over all response tokens.

To better understand how different fine-tuning objectives reshape token-level confidence, i.e., the model's margin-based confidence in the gold response tokens, we analyze the distribution of logit gaps on the training data. We randomly sample 100 examples from the training set and compute teacher-forced logit gaps for every token in the response part of each example. We then aggregate all token-level gaps across the sampled examples and plot their kernel density estimates for Base, SFT, DFT, FSFT, and TrimSFT in Figure 4.

The resulting distributions reveal clear differences in how each objective reshapes token-level confidence. Standard SFT shifts the distribution to the right relative to the base model, indicating stronger separation between gold tokens and their competitors after fine-tuning. FSFT, in contrast, leaves more mass near small logit gaps, consistent with its emphasis on low-confidence tokens and its poor performance in Table 1. DFT also moves the distribution toward larger gaps and appears closer to TrimSFT than to standard SFT, which may help explain why DFT often achieves competitive performance. However, DFT and TrimSFT reach this behavior through different mechanisms: DFT monotonically emphasizes high-probability tokens, whereas TrimSFT trims supervision away from both extremes of the logit-gap spectrum. As a result, TrimSFT shifts the distribution toward larger gaps while preserving a broad shape and reducing mass in the small-gap region. This supports our interpretation that TrimSFT improves reasoning SFT by reshaping token-level confidence in a more balanced way, rather than simply over-emphasizing uncertain tokens or uniformly sharpening already confident ones.

6 Conclusion

We presented TrimSFT, a token-level reweighting method that uses logit gaps to concentrate supervision on an intermediate confidence region. By trimming both already well-separated tokens and tokens with weak current support, TrimSFT provides a non-monotonic alternative to uniform SFT and monotonic reweighting methods. Across six base models and five mathematical reasoning benchmarks, TrimSFT consistently improves over standard SFT, while further analyses show gains in capability coverage, self-consistency, and better trade-offs than half-trim variants. Mechanistically, TrimSFT reshapes token-level confidence by moving tokens toward more confident regions while avoiding uniform over-sharpening. Overall, reasoning SFT should allocate supervision selectively near the model's current decision boundary rather than applying uniform pressure across all tokens.

Limitations

TrimSFT is evaluated primarily on mathematical reasoning benchmarks with supervised fine-tuning from base models, so its effectiveness on broader domains such as code generation, open-ended instruction following, or long-form reasoning remains to be further studied. In addition, TrimSFT uses two hyperparameters, the margin m and bandwidth τ, to define the emphasized logit-gap region. Our ablations show that the method is relatively robust to the exact margin choice and generally improves over standard SFT across a range of settings, suggesting that careful tuning is not necessary to obtain gains. Still, different model scales or data distributions may benefit from adaptive strategies that adjust the weighting region automatically during training.

Ethics

TrimSFT is a token-level reweighting method for supervised fine-tuning and does not involve new data collection or human annotation. All experiments are conducted using publicly available pretrained models and mathematical reasoning datasets intended for research use. As with other supervised fine-tuning methods, improving reasoning capabilities may also strengthen broader generation abilities of language models, which could potentially be misused in downstream applications.

References

Meta AI. 2024. Llama 3.2: Revolutionizing edge AI and vision with open, customizable models. https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/. Meta AI Blog, accessed 2026.

AI-MO. 2024a. AIMO validation AIME. https://huggingface.co/datasets/AI-MO/aimo-validation-aime.

AI-MO. 2024b. AIMO validation AMC. https://huggingface.co/datasets/AI-MO/aimo-validation-amc.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, and 1 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*. Available: https://arxiv.org/pdf/2204.05862.

Feng Chen, Allan Raventos, Nan Cheng, Surya Ganguli, and Shaul Druckmann. 2025. Rethinking fine-tuning when scaling test-time compute: Limiting confidence improves mathematical reasoning. In *Workshop on Reasoning and Planning for Large Language Models*. Available: https://arxiv.org/pdf/2502.07154.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*. Available: https://arxiv.org/pdf/2110.14168.

Lizhou Fan, Wenyue Li, Sheng Ma, Kuan-Chieh Lee, Yixin Yu, and Libby Hemphill. 2024. Nphardeval: Dynamic benchmark on reasoning ability of large language models via complexity classes. In *Annual Meeting of the Association for Computational Linguistics*. Available: https://aclanthology.org/2024.acl-long.225.pdf.

Xuan Gong, Senmiao Wang, Hanbo Huang, Ruoyu Sun, and Shiyu Liang. 2025. Vcore: Variance-controlled optimization-based reweighting for chain-of-thought supervision. *arXiv preprint arXiv:2510.27462*. Available: https://arxiv.org/pdf/2510.27462.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*. Available: https://arxiv.org/pdf/2407.21783.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*. Available: https://arxiv.org/pdf/1706.04599.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*. Available: https://arxiv.org/pdf/2501.12948.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in Neural Information Processing Systems*. Available: https://arxiv.org/pdf/1804.06872.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, and 1 others. 2024. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Annual Meeting of the Association for Computational Linguistics*. Available: https://aclanthology.org/2024.acl-long.211.pdf.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. Available: https://arxiv.org/pdf/2103.03874.

Hugging Face. 2025. Open r1: A fully open reproduction of deepseek-r1.

Yaning Jia, Chunhui Zhang, Xingjian Diao, Xiangchi Yuan, Zhongyu Ouyang, Chiyu Ma, and Soroush Vosoughi. 2025. What makes a good curriculum? disentangling the effects of data ordering on llm mathematical reasoning. *arXiv preprint arXiv:2510.19099*. Available: https://arxiv.org/pdf/2510.19099.

Jixuan Leng, Chengsong Huang, Banghua Zhu, and Jiaxin Huang. 2024. Taming overconfidence in llms: Reward calibration in rlhf. *arXiv preprint arXiv:2410.09724*. Available: https://arxiv.org/pdf/2410.09724.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, and 1 others. 2022. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems*. Available: https://arxiv.org/pdf/2206.14858.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, and 1 others. 2024. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*. Available: https://huggingface.co/collections/AI-MO/numinamath.

Jiacheng Lin, Zhongruo Wang, Kun Qian, Tian Wang, Arvind Srinivasan, Hansi Zeng, Ruochen Jiao, Xie Zhou, Jiri Gesi, Dakuo Wang, Yufan Guo, Kai Zhong, Weiqi Zhang, sujay sanghavi, Changyou Chen, Hyokun Yun, and Lihong Li. 2026. SFT doesn't always hurt general capabilities: Revisiting domain-specific fine-tuning in LLMs. In *International Conference on Learning Representations*. Available: https://arxiv.org/pdf/2509.20758.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *International Conference on Computer Vision*. Available: https://arxiv.org/pdf/1708.02002.

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Yelong Shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, and 1 others. 2024. Rho-1: Not all tokens are what you need. *arXiv preprint arXiv:2404.07965*. Available: https://arxiv.org/pdf/2404.07965.

Bingyuan Liu, Ismail Ben Ayed, Adrian Galdran, and Jose Dolz. 2022. The devil is in the margin: Margin-based label smoothing for network calibration. In *Conference on Computer Vision and Pattern Recognition*. Available: https://arxiv.org/pdf/2111.15430.

Chengwu Liu, Ye Yuan, Yichun Yin, Yan Xu, Xin Xu, Zaoyu Chen, Yasheng Wang, Lifeng Shang, Qun Liu, and Ming Zhang. 2025. Safe: Enhancing mathematical reasoning in large language models via retrospective step-aware formal verification. In *Annual Meeting of the Association for Computational Linguistics*. Available: https://aclanthology.org/2025.acl-long.594.pdf.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, Yansong Tang, and Dongmei Zhang. 2025. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. In *International Conference on Learning Representations*. Available: https://arxiv.org/pdf/2308.09583.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. In *Advances in Neural Information Processing Systems*. Available: https://arxiv.org/pdf/2405.14734.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. 2019. When does label smoothing help? In *Advances in Neural Information Processing Systems*. Available: https://arxiv.org/pdf/1906.02629.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*. Available: https://arxiv.org/pdf/2203.02155.

Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*. Available: https://arxiv.org/pdf/1701.06548.

Zhiwen Ruan, Yixia Li, He Zhu, Yun Chen, Peng Li, Yang Liu, and Guanhua Chen. 2025. Enhancing large language model reasoning via selective critical token fine-tuning. *arXiv preprint arXiv:2510.10974*. Available: https://arxiv.org/pdf/2510.10974.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. 2019. Analysing mathematical reasoning abilities of neural models. *arXiv preprint arXiv:1904.01557*. Available: https://arxiv.org/pdf/1904.01557.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical

reasoning in open language models. *arXiv preprint arXiv:2402.03300*. Available: https://arxiv.org/pdf/2402.03300.

Shubham Toshniwal, Ivan Moshkov, Sean Narenthiran, Daria Gitman, Fei Jia, and Igor Gitman. 2024. Openmathinstruct-1: A 1.8 million math instruction tuning dataset. In *Advances in Neural Information Processing Systems*. Available: https://arxiv.org/pdf/2402.10176.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, and 1 others. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems*. Available: https://arxiv.org/pdf/2406.01574.

Hongxin Wei, Renchunzi Xie, Hao Cheng, Lei Feng, Bo An, and Yixuan Li. 2022a. Mitigating neural network overconfidence with logit normalization. In *International Conference on Machine Learning*. Available: https://arxiv.org/pdf/2205.09310.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022b. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*. Available: https://arxiv.org/pdf/2201.11903.

Yongliang Wu, Yizhou Zhou, Zhou Ziheng, Yingzhe Peng, Xinyu Ye, Xinting Hu, Wenbo Zhu, Lu Qi, Ming-Hsuan Yang, and Xu Yang. 2026. On the generalization of SFT: A reinforcement learning perspective with reward rectification. In *International Conference on Learning Representations*. Available: https://arxiv.org/pdf/2508.05629.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*. Available: https://arxiv.org/pdf/2505.09388.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, and 1 others. 2024. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*. Available: https://arxiv.org/pdf/22409.12122.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. Limo: Less is more for reasoning. *arXiv preprint arXiv:2502.03387*. Available: https://arxiv.org/pdf/2502.03387.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng YU, Zhengying Liu, Yu Zhang, James Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2024. Metamath: Bootstrap your own mathematical questions for large language models. In *International Conference on Learning Representations*. Available: https://arxiv.org/pdf/2309.12284.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. 2024. MAmmoTH: Building math generalist models through hybrid instruction tuning. In *International Conference on Learning Representations*. Available: https://arxiv.org/pdf/2309.05653.

Mozhi Zhang, Mianqiu Huang, Rundong Shi, Linsen Guo, Chong Peng, Peng Yan, Yaqian Zhou, and Xipeng Qiu. 2024. Calibrating the confidence of large language models by eliciting fidelity. In *Conference on Empirical Methods in Natural Language Processing*. Available: https://aclanthology.org/2024.emnlp-main.173.pdf.

Contents of Appendix

A Experiment Setting

Parameter setting. We train all models for one epoch using the AdamW optimizer with a learning rate of 5×10^{-5}, cosine learning rate decay with warmup, and gradient clipping. Training is conducted with bfloat16 mixed precision and FlashAttention-2 acceleration under the FSDP framework. Unless otherwise specified, the global batch size is 24 with a per-device micro-batch size of 4, and the maximum sequence length is set to 2048. For TrimSFT, we set the margin to $m = 1.5$ and the bandwidth to $\tau = 0.8$.

Baseline setting. All four methods, SFT, FSFT, DFT, and TrimSFT, are trained under an identical configuration, including optimizer, learning rate, schedule, batch size, sequence length, precision, and number of epochs. They differ only in the per-token weight w_t applied to the cross-entropy loss, as summarized in Section 3.3. The only method-specific hyperparameters beyond this shared recipe are the margin m and bandwidth τ in TrimSFT, and the focusing parameter γ in FSFT, for which we use $\gamma = 2$ following common practice in the focal-loss literature (Lin et al., 2017). DFT introduces no additional hyperparameters.

Evaluation protocol. For evaluation, we sample eight generations per problem using temperature 1.0, top-p sampling with $p = 1.0$, and a maximum generation length of 2048 tokens. We use

temperature 1.0 to encourage diverse sampled solutions, which is necessary for evaluating average@8, pass@8, and best-of-8 under repeated sampling. Average@8 is computed as the mean correctness over the eight sampled generations, pass@8 checks whether at least one generation is correct, and best-of-8 uses majority voting over the extracted final answers. We extract final answers from the generated solutions and judge correctness using the same evaluation protocol across all methods and benchmarks.

Computing infrastructure. Experiments are conducted on NVIDIA RTX A6000 GPUs, each with 48GB memory, using CUDA 12.8 and NVIDIA driver 570.207. Each training run uses two GPUs with FSDP unless otherwise specified.

B Detailed Results for Ablations

Table 3 and Table 4 report the full per-benchmark results for the margin and bandwidth ablation study shown in Figure 3. Each setting is denoted by (m, τ), where m is the margin and τ is the bandwidth. While the main text summarizes the averaged trends across the evaluated benchmarks, this appendix provides the complete per-benchmark results for both varying the bandwidth under fixed margins and varying the margin under fixed bandwidths.

Table 3: Ablation results with fixed m and varying τ. Each setting is denoted as (m, τ).

(m, τ)	MATH500	OlyBench	Minerva	AMC	AIME24	Avg.
Fixed $m = 1$						
$(1, 0.1)$	66.83	29.53	23.86	29.02	9.61	31.77
$(1, 0.5)$	63.72	27.17	23.16	25.90	9.16	29.82
$(1, 1)$	62.80	27.66	18.65	31.74	12.07	30.58
$(1, 2)$	60.38	24.68	18.46	25.00	4.58	26.22
$(1, 3)$	53.63	21.05	16.30	22.50	3.32	23.36
$(1, 4)$	54.05	21.03	16.88	20.59	3.12	23.13
$(1, 5)$	46.83	16.38	14.15	17.34	2.07	19.35
$(1, 6)$	46.93	15.30	12.96	16.78	2.07	18.81
Fixed $m = 6$						
$(6, 0.1)$	65.12	30.76	23.31	30.75	7.93	31.57
$(6, 0.5)$	68.60	30.69	26.83	31.45	7.94	33.10
$(6, 1)$	66.83	29.28	22.65	27.65	11.65	31.61
$(6, 2)$	66.39	28.60	21.11	33.39	8.76	31.65
$(6, 3)$	63.95	27.44	20.50	29.55	7.50	29.79
$(6, 4)$	60.70	26.40	19.21	26.77	5.42	27.70
$(6, 5)$	56.90	24.88	18.65	25.45	4.15	26.01
$(6, 6)$	54.40	22.26	16.86	22.81	4.58	24.18
SFT	40.02	12.11	10.08	12.64	2.50	15.47

The per-benchmark results are consistent with the averaged trends in the main text. When the margin m is fixed, increasing the bandwidth τ generally reduces performance, especially when τ becomes large. In contrast, when $\tau = 1$ is fixed, the method remains stronger than standard SFT across a wide range of margin values, indicating that it does not require careful tuning of the exact margin location to obtain improvements.

Table 4: Ablation results with fixed τ and varying m. Each setting is denoted as (m, τ).

(m, τ)	MATH500	OlyBench	Minerva	AMC	AIME24	Avg.
Fixed $\tau = 1$						
$(0, 1)$	61.10	26.64	20.96	26.60	7.91	28.64
$(0.5, 1)$	66.75	30.73	23.45	31.76	9.99	32.54
$(1, 1)$	62.80	27.66	18.65	31.74	12.07	30.58
$(2, 1)$	62.25	26.06	19.39	27.36	7.09	28.43
$(3, 1)$	65.98	26.76	19.64	30.29	7.50	30.03
$(4, 1)$	63.60	26.90	22.03	27.51	5.83	29.17
$(5, 1)$	65.96	28.55	20.13	31.03	7.50	30.63
$(6, 1)$	66.83	29.27	22.65	27.65	11.65	31.61
Fixed $\tau = 6$						
$(0, 6)$	46.97	15.98	13.83	16.46	2.47	19.14
$(0.5, 6)$	48.13	17.14	14.23	16.03	3.75	19.86
$(1, 6)$	46.93	15.30	12.96	16.78	2.07	18.81
$(2, 6)$	46.30	17.74	13.88	16.32	3.32	19.51
$(3, 6)$	54.58	21.48	17.74	24.26	5.84	24.78
$(4, 6)$	55.88	22.75	17.19	23.96	5.82	25.12
$(5, 6)$	51.50	20.25	16.53	21.46	2.49	22.45
$(6, 6)$	54.40	22.26	16.86	22.81	4.57	24.18
SFT	40.02	12.11	10.08	12.64	2.50	15.47

C Half-Trim Variants

Table 5 reports additional results for TrimSFT and its two half-trim variants, TrimSFT-E and TrimSFT-H, on Qwen2.5-Math-1.5B. The main text reports representative margins $m \in \{1, 3, 5\}$, while this appendix includes the full set of margins $m \in \{1, 2, 3, 4, 5\}$. All variants are evaluated with fixed $\tau = 1$.

Table 5: Comparison between TrimSFT and its half-trim variants with fixed $\tau = 1$. **Bold** marks the best result among the three variants.

m	Method	MATH500	AMC	AIME24
1	TrimSFT	62.80	31.74	**12.07**
	TrimSFT-E	**65.98**	**33.01**	7.51
	TrimSFT-H	28.75	8.84	0.41
2	TrimSFT	62.25	27.36	7.09
	TrimSFT-E	**65.32**	**32.06**	**7.10**
	TrimSFT-H	37.05	11.18	0.41
3	TrimSFT	**65.98**	**30.29**	**7.50**
	TrimSFT-E	64.30	26.18	4.56
	TrimSFT-H	40.92	13.39	2.06
4	TrimSFT	63.60	**27.51**	**8.83**
	TrimSFT-E	**64.58**	27.49	7.51
	TrimSFT-H	42.43	13.98	2.47
5	TrimSFT	65.96	31.03	**7.50**
	TrimSFT-E	**66.67**	**32.19**	6.25
	TrimSFT-H	42.50	16.03	1.65

The results are consistent with the trends dis-

cussed in the main text. TrimSFT-E is often competitive with TrimSFT on relatively easier or medium-difficulty benchmarks such as MATH500 and AMC, suggesting that preserving full supervision on high-gap tokens can sometimes be useful. However, TrimSFT achieves better results on the harder AIME24 benchmark across the reported margins. In contrast, TrimSFT-H consistently performs much worse than both TrimSFT and TrimSFT-E, indicating that preserving full supervision on low-gap tokens while trimming high-gap tokens leads to an inferior trade-off. Overall, these results support the benefit of two-sided trimming over relying on either half-trim variant alone.