NOT ALL CONFLICTS ARE EQUAL: TRAJECTORY-AWARE TOKEN SELECTION FOR SUPERVISED FINE TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Supervised fine-tuning adapts pretrained language models to new domains, but often degrade their general capabilities. We study *conflicting tokens*, where the supervised target disagrees with a confident pretrained prediction. Recent token-level methods treat these tokens differently depending on the goal: suppressing them better preserves general capabilities but often believed to limit adaptation, while emphasizing them improves target-domain adaptation at the cost of retention. However, achieving strong target-domain adaptation while preserving general capabilities remains challenging. This motivates us to look more closely at conflicting tokens. We find that they do not behave uniformly during SFT: some resolve quickly but often revert, some are learned gradually, and others remain difficult to learn. Retaining gradually learned conflicts while masking other two groups gives the best adaptation–retention balance among the strategies we study. Based on this finding, we introduce FORETOKEN, an efficient pre-SFT forecasting method that estimates each conflict's resolution tendency from the initial optimization direction. This avoids an extra SFT run to observe token trajectories before constructing the mask. Across mathematical, medical, agentic, and code domains and three model families, FORETOKEN improves target-domain performance while better preserving general capabilities. On Qwen3-8B, it outperforms the strongest token-level baseline by 7.6 points on mathematical reasoning.

1 INTRODUCTION

Supervised fine-tuning (SFT) is the standard approach for adapting pretrained large language models (LLMs) to specialized domains, but such adaptation often comes at the cost of capabilities acquired during pretraining (Huan et al., 2025; Chen et al., 2025; Shenfeld et al., 2026; Chu et al., 2025; Sanyal et al., 2025a; Li et al., 2025). One reason is that standard SFT optimizes every target token under the same maximum-likelihood objective, even though supervised tokens differ substantially in difficulty, uncertainty, and compatibility with the model's pretrained knowledge (Zhang et al., 2026; Wang et al., 2026; Jung et al., 2025). The hardest case is *conflicting tokens*, whose supervision disagrees with the pretrained model's own prediction (Ren et al., 2024; Gekhman et al., 2024a; Zhong et al., 2025; Diao et al., 2026). How such tokens should be treated remains unclear. One line of work reduces their influence (Wu et al., 2026; Lin et al., 2026; Diao et al., 2026), while another emphasizes difficult or under-learned tokens (Yu et al., 2026; Ruan et al., 2025). Both establish the need for selective token supervision, while suggesting opposite strategies for the same tokens.

Despite these advances, the adaptation–retention trade-off remains unresolved. Recent token-level methods highlight this challenge, reducing the influence of difficult or conflicting tokens can better preserve pretrained capabilities but limit target-domain learning, while stronger optimization can improve specialization at the cost of greater general-capability degradation (Diao et al., 2026; Lin et al., 2026; Sanyal et al., 2025b). Recent evidence further indicates that no single treatment of difficult, low-probability tokens is universally effective (Li et al., 2026). This growing evidence raises a natural question: *must all conflicting tokens be learned to achieve the full target-domain gain, or must they all be suppressed to preserve general capabilities?* We address this question by conducting controlled masking experiments over different subsets of conflicting tokens (Table 1). Surprisingly, the choice of which conflicts to suppress matters substantially, some subsets improve

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.

both target-domain and general performance, while others weaken adaptation. Conflicting tokens therefore do not play a uniform role during adaptation, some provide useful supervision for specialization, whereas others can be suppressed without cost.

This finding motivates us to distinguish conflicts that provide useful supervision for specialization from those that should be suppressed. To understand this difference, we track the same initially conflicting tokens throughout standard SFT. A clear pattern appears, tokens that look similar before training can behave very differently during SFT. Our analysis reveals three distinct learning dynamics: some conflicts resolve rapidly and later re-enter the conflict region, others resolve progressively under sustained optimization, while a third group remains unresolved for most of training (Figure 3). We refer to these as *Early*, *Intermediate*, and *Late* trajectories, respectively. Importantly, these differences also matter for training, masking Early and Late conflicts while retaining Intermediate ones yields the strongest adaptation–retention behavior among the trajectory-based choices we examine. This suggests that a conflict's usefulness is closely tied to how it evolves during optimization.

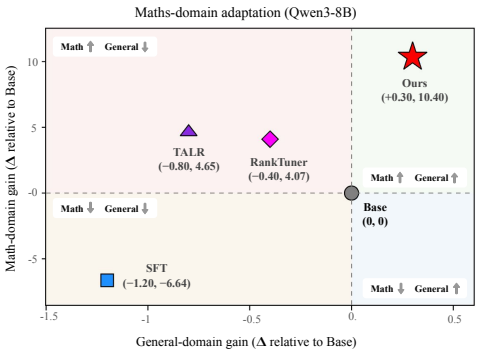


Figure 1: Math-domain adaptation and general-capability retention on Qwen3-8B. Both axes show performance changes after SFT relative to the pretrained model. FORETOKEN lies in the upper-right quadrant, achieving strong math-domain (Pass@1) adaptation while maintaining comparable general-domain performance.

However, using these trajectories directly is not practical. A post-hoc strategy can first run standard SFT to determine each conflict's trajectory and then restart training with the resulting mask, but this requires performing SFT twice. An online alternative avoids the extra run, yet identifies persistent conflicts only after they have already influenced many optimization updates (see Table C.1). A practical solution therefore requires identifying each conflict's trajectory before it has influenced training. We therefore propose FORETOKEN, a pre-SFT forecasting framework that estimates each conflict's resolution tendency from its response to the initial SFT optimization direction, before the first parameter update. FORETOKEN provides these estimates with a one-time pre-SFT forecasting step, *avoiding the additional full SFT run needed to observe trajectories post hoc*. Using these predictions, we retain Intermediate conflicts while masking Early and Late conflicts throughout SFT, enabling selective adaptation. Across four adaptation domains and three model families, our method consistently improves target-domain performance while retaining general capabilities. On mathematical reasoning with Qwen3-8B, it outperforms the strongest token-level baseline by 7.6 points while maintaining general-domain average over the pretrained model (Figure 1).

Our main contributions are summarized below.

- We show that conflicting tokens play fundamentally different roles during SFT, exhibiting distinct Early, Intermediate, and Late learning trajectories with different effects on adaptation and retention.
- We introduce FORETOKEN, which estimates the resolution tendency of individual conflicting tokens before the first SFT update and uses these estimates to guide selective masking throughout training.
- We extensively evaluate our approach across math, medical, agent, and code domains using Qwen3-8B, Llama-3.1-8B-Instruct, and DeepSeek-R1-0528-Qwen3-8B, and further demonstrate its effectiveness under sequential continual fine-tuning.

2 RELATED WORK

Domain adaptation and capability retention. Supervised fine-tuning improves target-domain performance, but the resulting specialization often weakens capabilities acquired during pretraining (Chu et al., 2025; Huan et al., 2025; Chen et al., 2025). Several causes have been identified: supervision that introduces knowledge the pretrained model does not already have encourages hallucination (Gekhman et al., 2024b), and maximum-likelihood training on narrow data reduces output diversity

(Li et al., 2025). Proposed remedies work mostly at the level of examples or objectives, for instance by upweighting easy examples to limit drift (Sanyal et al., 2025b) or by replacing the log-likelihood objective itself (Li et al., 2026). Recent evidence indicates that this degradation is not unavoidable and depends strongly on which training signals are emphasized during fine-tuning (Lin et al., 2026).

Token-level selective supervision. Recent work adjusts how much each supervised token contributes to SFT, but the two main lines pull in opposite directions. One reduces the influence of difficult or conflicting tokens: DFT (Wu et al., 2026) rescales updates on low-probability tokens to avoid the very large gradients produced by standard cross-entropy, TALR (Lin et al., 2026) down-weights hard, low-probability tokens, and EAFT (Diao et al., 2026) suppresses tokens that are both low-probability and low-entropy in order to protect pretrained capabilities. The other line does the opposite and emphasizes tokens judged difficult, under-learned, or important to the final answer: RankTuner (Yu et al., 2026) upweights under-learned tokens using probability and uncertainty, and CFT (Ruan et al., 2025) finds tokens that are critical to the reasoning process and focuses training on them. A separate line selects tokens to reduce activation memory rather than to change what is learned (Simoulin et al., 2024; Zeng et al., 2026). In all of these, a token's treatment follows a fixed rule applied before training, and every selected token is handled the same way.

Learning dynamics and predicting them in advance. Our approach connects two lines of work. The first groups training data by how it behaves during training: Toneva et al. (2018) track examples that are learned and then forgotten again over the course of training, and Swayamdipta et al. (2020) separate examples into easy, ambiguous, and hard-to-learn regions using the model's confidence and its variability across epochs. These categories describe real differences in how data is learned, but they can only be assigned after a full training run has finished, and they are defined over examples rather than tokens. The second line estimates the effect of training data without carrying out the training, using influence functions (Koh & Liang, 2017), gradient similarity to a target objective (Pruthi et al., 2020; Xia et al., 2024), or directional derivatives obtained from a small number of perturbed evaluations (Nesterov & Spokoiny, 2017; Malladi et al., 2023). These methods score whole examples for selection and usually need per-example gradients.

3 PRELIMINARIES

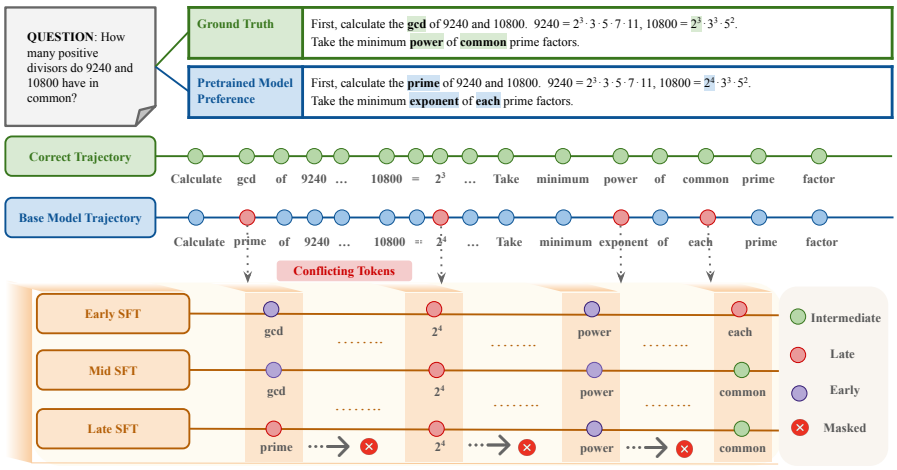


Figure 2: **Conflicting tokens follow different learning trajectories, motivating selective supervision in FORETOKEN. (Top)** We illustrate a NuminaMath-CoT example. Supervised target (green) and pretrained model's own preferred prediction (blue) agree at most positions but disagree at few: *gcd*/*prime*, $2^3/2^4$, *power*/*exponent*, and *common*/*each*. These positions form conflict set $\mathcal{C}$, identified at θ_0 before any parameter update. **(Bottom)** Tracking same four tokens at early, middle, and late stages of standard SFT shows that they are learned in different ways. *power* is corrected immediately and stays correct (Early); *gcd* is also corrected early but later reverts to the pretrained prediction *prime* (Early, unstable); 2^4 resists the objective and is never corrected (Late); and *each* changes gradually, becoming *common* under sustained optimization (Intermediate). FORETOKEN estimates these resolution tendencies of conflicting tokens before SFT, then masks Early and Late conflicts ($\otimes$) while retaining Intermediate one; resulting mask remains fixed throughout training.

3.1 Problem Setup

Supervised fine-tuning. Let $\mathcal{D} = \{(x^{(n)}, y^{(n)})\}_{n=1}^{N}$ denote the SFT dataset, where $x^{(n)}$ is the input and $y^{(n)} = (y_1^{(n)}, \ldots, y_{T_n}^{(n)})$ is the target response. Let $\mathcal{T} = \{(n, i) : 1 \le n \le N,\ 1 \le i \le T_n\}$ be the set of all target-token positions, and we write $i \in \mathcal{T}$ and drop the sample index n when it is clear from context. Given a language model p_θ initialized from pretrained parameters θ_0, let $p_i(\theta) = p_\theta(y_i \mid x, y_{<i})$ denote the probability assigned to the target token at position i. Standard SFT minimizes the token-averaged negative log-likelihood $\mathcal{L}_{\mathrm{SFT}}(\theta) = -\frac{1}{|\mathcal{T}|}\sum_{i\in\mathcal{T}} \log p_i(\theta)$.

Conflicting tokens. We characterize each target token by how the *pretrained* model p_{θ_0} treats it, using two statistics: the probability it assigns to the supervised target, and how confident it is in its own prediction. For the former, let $p_i^0 = p_i(\theta_0)$. For the latter, let $\mathcal{V}_i^K \subset \mathcal{V}$ denote the K tokens with the highest probability under $p_{\theta_0}(\cdot \mid x, y_{<i})$, and define the renormalized top-K distribution

$$\widetilde{p}_i(v) = \frac{p_{\theta_0}(v \mid x, y_{<i})}{\sum_{u\in\mathcal{V}_i^K} p_{\theta_0}(u \mid x, y_{<i})}$$

for $v \in \mathcal{V}_i^K$. Its entropy $H_i = -\sum_{v\in\mathcal{V}_i^K} \widetilde{p}_i(v) \log \widetilde{p}_i(v)$ is small when the pretrained model concentrates its mass on a single alternative and large when it is uncertain among several candidates, and restricting to the top-K tokens makes H_i insensitive to the long tail of the vocabulary. We call a target token *conflicting* when the pretrained model assigns it low probability while being confident in a different prediction. Given thresholds τ_p and τ_H, the set of conflicting positions is $\mathcal{C} = \{i \in \mathcal{T} : p_i^0 \le \tau_p \text{ and } H_i \le \tau_H\}$. Thus $\mathcal{C}$ isolates supervision that directly contradicts a confident pretrained belief, as opposed to supervision the model is merely unsure about.

3.2 Do All Conflicts Matter Equally?

Prior token-level methods treat the conflicting set $\mathcal{C}$ as a single population, suppressing or emphasizing all of its members under one rule. To test whether this is justified, we mask different subsets of conflicting tokens, identified at different stages of SFT, and compare the results in Table 1. Two findings stand out. First, every masking variant outperforms No Mask, which retains all conflicts, so some conflicts actively harm adaptation. Second, Mask C outperforms Mask All, which removes every conflict, so the conflicts that Mask C keeps are useful for adaptation. Conflicting tokens therefore differ in how they affect SFT, which raises the question of which conflicts help and which hinder.

Table 1: Effect of masking conflicting tokens on adaptation and retention selected from different stages of SFT (Qwen3-8B, math, Pass@1).

	Target Domain		General Domain	
Setting	**AMC ↑**	**AIME ↑**	**MMLU ↑**	**IFEval ↑**
Base	34.5	6.0	**73.0**	**80.6**
No Mask	29.4	2.5	71.1	79.1
Mask A	35.2	6.7	72.4	78.0
Mask B	42.0	**10.6**	72.3	78.0
Mask C	**53.8**	9.2	72.4	79.1
Mask D	46.1	9.6	72.4	78.7
Masking all	49.4	7.4	73.2	79.0

3.3 What Makes Conflicts Different?

The subsets in Table 1 differ only in *when* their conflicts are resolved during SFT, suggesting that resolution timing may separate helpful conflicts from harmful ones. We therefore track the conflicting tokens throughout SFT and examine how their status changes over training. Most conflicts resolve early: 72.4% within 200 steps, although 49% later re-enter the conflict region. Others require sustained optimization, while 17.5% never resolve during SFT (Figure 3). We refer to these three behaviors as *Early*, *Intermediate*, and *Late*, respectively.

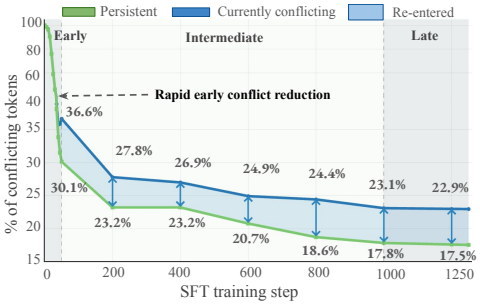


Figure 3: Evolution of initially conflicting tokens cross different stages of SFT (Qwen3-8B, math). The widening gap (light blue) between the two curves further shows that many early-resolved conflicts later become conflicting again.

Early conflicts are resolved quickly but often unstably, nearly half of them later fall back into the conflict region, suggesting that the model does not reliably retain what it learns from them. Late conflicts show the opposite problem, they stay at low

probability throughout training, repeatedly pulling the model away from its pretrained predictions without ever being learned. Intermediate conflicts, in contrast, are learned gradually under sustained optimization, suggesting they carry new target-domain information that SFT can absorb. We therefore hypothesize that retaining Intermediate conflicts while masking Early and Late ones gives the best trade-off, which our ablation confirms (Table 5). However, using these trajectories raises a practical challenge: they can only be identified after observing how conflicts evolve during SFT. A post-hoc strategy would therefore require an additional SFT run to first discover the trajectories, while an online strategy identifies persistent conflicts only after they have already influenced many updates. Table 10 illustrates how late online identification limits its effectiveness. This motivates estimating each conflict's resolution tendency *before the first SFT update*.

4 METHOD

We now turn the trajectory patterns of Section 3.3 into a decision rule that can be applied before training begins Figure 2. The idea is straightforward: a conflict leaves the conflict region once its target probability rises above τ_p or its uncertainty rises above τ_H, so how early it resolves depends on how quickly SFT pushes it toward one of these two boundaries. That speed can be estimated before any parameter update, by measuring how each token responds to the direction in which SFT is about to move the model. We use this response to construct a continuous score that captures each conflict's tendency toward early resolution and use it to define the fixed token mask before SFT.

4.1 PRE-SFT TRAJECTORY FORECASTING

Measuring how a token responds to the first SFT step. Standard SFT begins by moving the pretrained parameters along the descent direction $D = -G$, where $G = \nabla_\theta \mathcal{L}_{\mathrm{SFT}}(\theta_0)$. This direction is shared by all tokens and captures where training is initially headed. Individual tokens, however, respond to it very differently: a token whose target probability rises sharply along D is being pulled out of the conflict region, whereas one whose probability barely moves is being left behind. We therefore measure, for each conflict $i \in \mathcal{C}$, the directional derivatives $v_i^p = \nabla_\theta \log p_i(\theta_0)^\top D$ and $v_i^H = \nabla_\theta H_i(\theta_0)^\top D$, which describe how the token's probability and uncertainty change as the model takes its first step. Computing these quantities exactly is not practical, since obtaining $\nabla_\theta \log p_i$ for every conflict separately would require one backward pass per token. We avoid this by noting that only the projection onto D is needed, which a central finite difference recovers from two additional forward passes (Nesterov & Spokoiny, 2017), for any token statistic $f_i(\theta)$,

$$\nabla_\theta f_i(\theta_0)^\top D \approx \frac{f_i(\theta_0 + hD) - f_i(\theta_0 - hD)}{2h},$$

which we apply to $f_i(\theta) = \log p_i(\theta)$ and $f_i(\theta) = H_i(\theta)$. Because D is shared, a single pair of perturbed passes yields the response of every conflict at once. The direction itself is computed from the exact SFT gradient, only the token-level responses are approximated. The perturbation size is set relative to the scale of the model as $h = \epsilon \, \|\theta_{\mathrm{tr}}\| / \|D\|$, where θ_{tr} are the trainable parameters and ϵ controls the relative step.

From response to trajectory score. A large response does not by itself mean that a token will resolve early, because conflicts start at different distances from the boundaries of the conflict region. A token already close to τ_p needs only a small increase in probability to escape, while one deep inside the region needs a much larger one. We therefore compare each response against the distance it has to cover, $d_i^p = \log \tau_p - \log p_i$ and $d_i^H = \tau_H - H_i$, and, since crossing either boundary is enough to leave the region, keep whichever of the two routes is faster:

$$s_i = \max\left(\frac{v_i^p}{d_i^p}, \, \frac{v_i^H}{d_i^H} \right).$$

The score s_i measures how strongly the initial SFT direction moves a conflict toward the nearest boundary relative to its remaining distance. Larger scores indicate a stronger tendency toward early resolution, while smaller scores indicate greater persistence. To construct the groups used for masking, we let the continuous score distribution determine the boundaries rather than fixing absolute thresholds, since the scale of s_i can vary across models and domains. Because s_i is heavy-tailed, we

first apply the monotone transform $\tilde{s}_i = \mathrm{sign}(s_i)\log(1+|s_i|)$, which reduces skew without changing the ordering of the conflicts, and then fit a three-component Gaussian mixture model to $\{\tilde{s}_i\}_{i\in\mathcal{C}}$. Sorting the fitted components by their means associates the lowest-, middle-, and highest-scoring components with Late, Intermediate, and Early *resolution tendencies*, respectively. Let $\hat{z}_i$ denote the resulting score-derived group:

$$\hat{z}_i = \begin{cases} \text{Late}, & \tilde{s}_i \le \gamma_L, \\ \text{Intermediate}, & \gamma_L < \tilde{s}_i < \gamma_E, \\ \text{Early}, & \tilde{s}_i \ge \gamma_E. \end{cases}$$

$\gamma_L < \gamma_E$ are the two boundaries between adjacent GMM components. Thus, the groups are determined entirely from the pre-SFT score distribution, without observing any actual training dynamics during SFT.

4.2 ForeToken-Guided Supervised Fine-Tuning

Using the score-derived groups, ForeToken determines which tokens contribute to SFT. We retain conflicts assigned to the Intermediate group, mask those assigned to Early and Late groups, and keep all non-conflicting tokens unchanged.

Let $\hat{z}_i \in \{\text{Early}, \text{Intermediate}, \text{Late}\}$ denote the score-derived group of token i. Using the binary mask m_i, the training objective is

$$\mathcal{L}_{\text{ForeToken}}(\theta) = -\frac{\sum_{i\in\mathcal{T}} m_i \log p_i(\theta)}{\sum_{i\in\mathcal{T}} m_i}, \qquad m_i = \mathbb{I}[i \notin \mathcal{C} \ \vee \ \hat{z}_i = \text{Intermediate}]\,.$$

Because masking is applied at individual token positions rather than token types, the same token may be retained in one context and masked in another. Masked positions still remain in the sequence as context, but their contribution to the training loss is removed. This fixed mask is constructed before the first SFT update and used throughout training.

5 Analysis of Conflict Dynamics

How reliable is pre-SFT trajectory forecasting? We first examine whether the pre-SFT score reflects how conflicts subsequently evolve under standard SFT.We assess this by ranking conflicts by their pre-SFT scores. Figure 4 shows a clear consistent pattern: higher-scoring subsets contain more conflicts that resolve Early and fewer that remain Late or unresolved. Among the top 10% of conflicts ranked by the pre-SFT score, 87.5% follow an Early trajectory, compared with 72.4% overall, while the fraction of Late conflicts decreases from 20.7% to 7.5%. The unresolved fraction similarly decreases from 17.5% to 5.9%. At the highest-score region, 91.1% of the top 1% follow an Early trajectory, while only 4.7% are Late. Consistent with this ordering, the median score decreases from Early (0.153) to Intermediate (0.081) and Late (0.038) conflicts (Appendix D). Overall, the pre-SFT score provides a reliable ranking signal, higher scores consistently correspond to conflicts that resolve earlier under subsequent SFT, without observing actual trajectories.

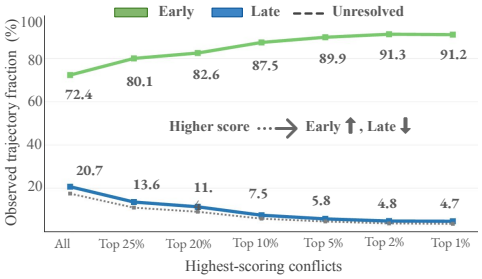


Figure 4: Trajectories of conflicts ranked by pre-SFT score (Qwen3-8B, math). Each point shows the fraction of Early, Late, and never-resolved conflicts among the top-$k\%$ highest-scoring conflicts; "All" is the full tokens conflict population.

Probability and entropy dynamics. Figure 5 examines the probability and entropy dynamics of Early, Intermediate, and Late trajectories during standard SFT. Early conflicts exhibit a sharp initial increase in target-token probability, reaching 0.49 within 200 steps, but their correction is not entirely stable, with many re-entering the conflict region. In contrast, Intermediate conflicts show a gradual increase in probability, crossing τ_p near step 600, while their entropy remains relatively stable. Late conflicts exhibit limited changes in both quantities, with target-token probability reaching only 0.11 and entropy remaining low (0.17) at the final checkpoint.

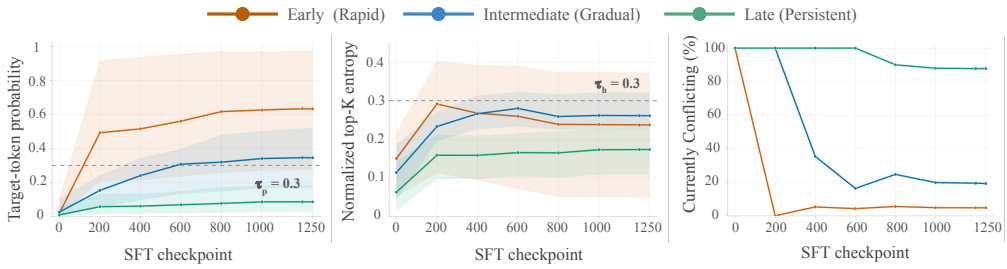


Figure 5: Learning dynamics of initially conflicting tokens grouped by their observed Early, Intermediate, and Late trajectories during standard SFT. From left to right, the panels show the evolution of target-token probability, normalized top-K entropy, and the percentage of tokens that remain in the conflict region. Dashed horizontal lines indicate the probability and entropy thresholds ($\tau_p = \tau_H = 0.3$).

6 EXPERIMENTS

We evaluate FORETOKEN across multiple model families, adaptation domains, and training settings. Our experiments assess its ability to improve task adaptation while preserving general capabilities, its effectiveness under sequential adaptation and different model scales, and the relationship between the pre-SFT forecasting score and the subsequent learning behavior of conflicting tokens.

6.1 EXPERIMENTAL SETUP

Models and Training Data. We evaluate our approach on Qwen3-8B, Llama-3.1-8B-Instruct, and DeepSeek-R1-0528-Qwen3-8B across four adaptation domains: mathematical reasoning, medical reasoning, agent/tool use, and code generation. For a controlled comparison, we use a fixed 10K-example training subset for each domain identical across all compared methods, drawn from NuminaMath (Li et al., 2024), medical-o1-reasoning-SFT (Chen et al., 2024), Nemotron Agentic Tool Use (NVIDIA, 2025), and Magicoder-OSS-Instruct-75K (Wei et al., 2023), respectively. Additional dataset preprocessing and formatting details are provided in Appendix B.

Baselines. We compare against the pretrained BASE model, standard SFT, and representative token-level fine-tuning approaches: DFT (Wu et al., 2026), TALR (Lin et al., 2026), EAFT (Diao et al., 2026), and RANKTUNER (Yu et al., 2026). These methods cover probability-based reweighting, confident-conflict suppression, and joint probability–entropy based token selection, providing direct comparisons to our trajectory-guided treatment of conflicting tokens.

Evaluation Benchmarks and Metrics. For mathematical reasoning, we evaluate on AMC, AIME, Minerva Math (Lewkowycz et al., 2022), MATH-OAI (Lightman et al., 2024), and OlympiadBench (He et al., 2024), and report Pass@1 and Pass@16. For medical adaptation, we evaluate on MedMCQA (Pal et al., 2022), MedQA (Jin et al., 2021), and PubMedQA (Jin et al., 2019); for agent adaptation, on BFCL (Patil et al., 2025) and MetaTool (Huang et al., 2024); and for code generation, on HumanEval+ (Liu et al., 2023) and MBPP+ (Austin et al., 2021). We measure preservation of general capabilities using MMLU (Hendrycks et al., 2020), IFEval (Zhou et al., 2023) and CLUEWSC (Xu et al., 2020). We additionally report domain-wise averages for compact comparison. Full evaluation and decoding protocols are provided in Appendix B.

6.2 MAIN RESULTS

Mathematical Reasoning. Table 2 shows that our method consistently improves the adaptation–retention trade-off across model families. On Qwen3-8B, our method achieves a Math Avg. of 58.3, outperforming the strongest baseline, RankTuner, by 7.6 points, while also slightly improving the general-domain (MMLU and IFEval) average over the base model (77.1 vs. 76.8). On DeepSeek-R1-0528-Qwen3-8B, our method reaches 50.7 Math Avg., a 2.9-point improvement over RankTuner, while substantially reducing the general-capability degradation observed with standard SFT (66.8 vs. 61.7). Additional results for Llama-3.1-8B-Instruct and CLUEWSC are provided in Appendix C.

Table 2: Performance comparison on mathematical reasoning and general-domain benchmarks across two model families (Qwen3-8B and DeepSeek-R1-0528 Qwen3-8B). We report Pass@1 (P@1) and Pass@16 (P@16) for the math-domain benchmarks, and accuracy (Acc.) for MMLU and IFEval. Math Avg. is the mean of all ten displayed math scores, while Gen. Avg. is the mean of MMLU and IFEval. Best results among fine-tuned methods, excluding the base model, are shown in **bold**; ties at the reported precision are all highlighted. The blue row reports the difference between our method and the best competing fine-tuning baseline.

Model	Method	Math Domain											General Domain		
		AMC		AIME		Minerva		MATH-OAI		Olympiad		Math Avg.	MMLU	IFEval	Gen. Avg.
		P@1	P@16	P@1	P@16	P@1	P@16	P@1	P@16	P@1	P@16	Avg.	Acc.	Acc.	Avg.
Qwen3-8B	Base	34.5	70.0	6.0	26.7	31.4	48.5	65.1	87.4	27.2	51.1	44.8	73.0	80.6	76.8
	SFT	29.4	77.5	2.5	16.7	21.8	55.1	56.4	89.2	21.0	56.3	42.6	72.1	79.1	75.6
	EAFT	37.0	75.0	4.6	20.0	19.8	51.8	55.7	88.4	21.5	56.4	43.0	71.8	78.0	74.9
	DFT	41.1	72.5	4.2	16.7	26.1	40.1	69.2	85.0	32.6	54.8	44.2	72.6	**80.6**	76.6
	TALR	43.4	77.5	6.7	16.7	35.7	60.3	68.8	87.4	32.9	57.8	48.7	73.1	78.9	76.0
	RankTuner	46.6	**90.0**	6.7	30.0	28.0	52.9	69.5	89.0	34.1	60.3	50.7	73.1	79.7	76.4
	ForeToken	**55.2**	**90.0**	**12.5**	**46.7**	**37.2**	**65.4**	**73.7**	**94.6**	**37.6**	**69.9**	**58.3**	**73.5**	**80.6**	**77.1**
DeepSeek-R1-0528 Qwen3-8B	Base	25.6	65.0	3.5	23.3	19.6	44.1	38.4	73.6	11.5	35.4	34.0	69.7	68.6	69.1
	SFT	23.4	67.5	3.1	16.7	19.3	52.2	51.2	86.8	17.1	52.0	38.9	70.6	52.9	61.7
	EAFT	22.7	72.5	4.8	23.3	17.0	49.3	51.0	87.2	17.4	50.8	39.6	70.8	51.8	61.3
	DFT	33.4	65.0	**10.2**	23.3	25.4	51.1	65.3	86.6	28.5	54.2	44.3	**71.2**	60.4	65.8
	TALR	33.3	67.5	10.0	23.3	25.6	51.1	64.8	86.4	28.0	51.6	44.2	70.5	53.1	61.8
	RankTuner	38.4	82.5	9.2	**30.0**	24.5	53.3	64.2	90.0	28.0	57.6	47.8	68.2	**65.6**	**66.9**
	ForeToken	**44.5**	**87.5**	9.2	26.7	**27.8**	**60.7**	**66.1**	**91.8**	**30.6**	**62.1**	**50.7**	70.9	62.7	66.8

Generalization Across Domains. Table 4 shows that the benefits of FORETOKEN extend beyond mathematical reasoning to code, medical, and agent adaptation. For code generation, our method improves the domain average from 67.5 under standard SFT to 70.9, exceeding the base model's 69.7. In the medical domain, it improves over SFT by 4.1 points (69.7 vs. 65.6) while preserving the general-domain average at 77.9. For agent domain, our method nearly retains the base model's domain performance (70.0 vs. 70.4) while improving the general-domain average to 78.1. Overall, these results show that the benefits of trajectory-guided conflict selection generalize across heterogeneous adaptation domains while better preserving general capabilities. Detailed benchmark-level results for each domain are provided in Appendix C.

Table 3: Final continual learning performance after sequential adaptation on Math → Agent → Medical → Code. All results are reported as percentages. **Avg.** denotes the mean across benchmarks within each domain.

Method	Math			Agent			Medical			Code			General			
	AMC	MATH-OAI	Avg.	BFCL	MetaTool	Avg.	MedMCQA	MedQA	Avg.	HumanEval+	MBPP+	Avg.	MMLU	IFEval	CLUEWSC	Avg.
Base	34.5	65.1	49.8	64.8	76.0	70.4	66.0	76.9	71.5	78.0	61.4	69.7	73.0	80.6	79.6	77.7
CL SFT	22.8	72.2	47.5	59.4	71.5	65.4	57.4	67.8	62.6	63.5	63.5	63.5	72.6	75.2	73.4	73.7
CL ForeToken	**49.7**	**72.0**	**60.8**	**59.1**	**74.1**	**66.6**	**60.1**	**70.6**	**65.4**	**73.2**	**62.4**	**67.8**	**72.2**	**75.6**	**75.7**	**74.5**

Sequential Continual Fine-Tuning. We further evaluate our method under sequential adaptation across Math → Agent → Medical → Code. Figure 6, Table 3 reports the final performance after completing the full training sequence. Across all four domains, our method consistently outperforms sequential SFT, with the largest gain on Math (60.8 vs. 47.5), Code (67.8 vs. 63.5), Medical (65.4 vs. 62.6), and Agent (66.6 vs. 65.4). In particular, after the final Code stage, it retains much stronger Math performance than sequential SFT, indicating reduced forgetting of earlier domains. The same trend is reflected in the general-domain average, where our method remains higher than sequential SFT (74.5 vs. 73.7). Overall, these results show that trajectory-guided conflict selection remains effective under continual fine-tuning.



Figure 6: Final performance after sequential adaptation on Math → Agent → Medical → Code. Each point reports performance in the corresponding domain after completing the full sequence. Arrows indicate the change from the pretrained base performance in each domain to its final performance after learning all four domains.

Table 4: Cross-domain performance comparison on code, medical, and agent adaptation. Domain Avg. denotes the average performance on the corresponding target-domain benchmarks, while Gen. Avg. denotes the average over MMLU, IFEval, and CLUEWSC.

Method	Code		Medical		Agent	
	Domain Avg.	Gen. Avg.	Domain Avg.	Gen. Avg.	Domain Avg.	Gen. Avg.
Base	69.7	77.7	72.4	77.7	70.4	77.7
SFT	67.5	75.7	65.6	76.2	66.3	76.7
TALR	68.6	76.1	65.6	76.6	65.5	76.7
ForeToken	**70.9**	**77.0**	**69.7**	**77.9**	**70.0**	**78.1**

6.3 Inference-Time Scaling

Sampling multiple candidate solutions improves reasoning only if additional samples can meaningfully explore the solution space, a method that merely sharpens its output distribution will gain at Pass@1 and lose that advantage as the budget grows Welleck et al. (2024); Charniak & Johnson (2005). Figure 7 tests this for ForeToken. Across two Math benchmarks, ForeToken achieves higher Pass@k than standard SFT and TALR at every budget from $k = 1$ to $k = 16$, and the gap does not close as k increases. Its gains therefore reflect a genuinely stronger set of candidate solutions rather than a sharper distribution over the ones standard SFT could already produce.

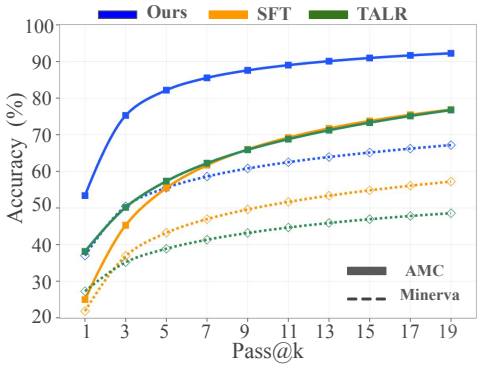


Figure 7: Inference-time scaling on Qwen3-8B across sampling budgets ($k = 1$ to $k = 16$).

6.4 Computational Overhead

Trajectory forecasting is a one-time preprocessing stage requiring one forward–backward pass to obtain the initial SFT direction D, followed by two forward-only passes at $\theta_0 \pm hD$, for a total cost of $3C_F + C_B$. This cost is incurred only once, before training begins. Once the mask is constructed, each subsequent optimization step has the same complexity as standard SFT, with no additional inference-time overhead. Importantly, because the number of forecasting passes is fixed and does not increase with training epochs or LoRA rank, its relative computational cost becomes negligible for longer or more expensive training settings, such as full-parameter fine-tuning.

6.5 Ablation

Table 5 isolates which conflicts should be masked. Masking alone does not explain the gains. Randomly masking the same number of tokens performs close to standard SFT (31.3 vs. 29.4 on AMC), whereas masking specific conflict groups yields much larger improvements. Masking every conflict is also suboptimal. Retaining Intermediate conflicts while masking Early and Late ones improves over masking all conflicts by 5.8 points on AMC, 5.1 points on AIME. Early and Late masking are complementary, as each gives a partial gain and their combination performs best. In contrast, masking only Intermediate conflicts stays close to SFT. Across all of these choices, general-domain performance remains stable.

Table 5: Effect of masking different subsets of conflicting tokens during SFT on target and general performance. Random masking provides little improvement, masking Early and Late conflicts gives the best results

Mask	AMC↑	AIME↑	MMLU↑
No Mask	29.4	2.5	72.1
Random	31.3	3.0	71.0
Early	38.9	5.0	71.9
Intermediate	28.6	2.5	72.3
Late	40.6	11.7	72.1
Early + Late	**55.2**	**12.5**	**73.5**
All	49.4	7.4	73.2

7 Conclusion

We demonstrate that conflicting tokens play different roles during SFT and that their learning trajectories provide a useful basis for selective supervision. Building on this observation, we introduce a pre-SFT forecasting framework ForeToken that identifies which conflicts to retain or suppress before training begins, without observing their actual trajectories during SFT. With a one-time forecasting cost and no additional training-step or inference-time overhead, our approach provides a practical framework for balancing adaptation and retention in language model fine-tuning.

AI USE STATEMENT

In preparing this work, we used generative AI tools to assist with language revision, including improving the grammar, clarity, and presentation of author-written text. All AI-assisted suggestions were independently reviewed and verified by the authors, and no AI-generated content was accepted without author evaluation. The authors take full responsibility for the final content of the paper.

REPRODUCIBILITY STATEMENT

We provide anonymized source code, training and evaluation configurations, and data-preparation scripts required to reproduce our experiments. Additional implementation details, hyperparameters, datasets, and evaluation protocols are described in the main paper and appendix.

REFERENCES

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Eugene Charniak and Mark Johnson. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pp. 173–180, 2005.

Hardy Chen, Haoqin Tu, Fali Wang, Hui Liu, Xianfeng Tang, Xinya Du, Yuyin Zhou, and Cihang Xie. Sft or rl? an early investigation into training r1-like reasoning large vision-language models. *arXiv preprint arXiv:2504.11468*, 2025.

Junying Chen, Zhenyang Cai, Ke Ji, Xidong Wang, Wanlong Liu, Rongsheng Wang, Jianye Hou, and Benyou Wang. Huatuogpt-o1, towards medical complex reasoning with llms. *arXiv preprint arXiv:2412.18925*, 2024.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*, 2025.

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.

Zorik Gekhman, Gal Yona, Roee Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. Does fine-tuning LLMs on new knowledge encourage hallucinations? In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 7765–7784, Miami, Florida, USA, November 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.444. URL https://aclanthology.org/2024.emnlp-main.444/.

Zorik Gekhman, Gal Yona, Roee Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. Does fine-tuning llms on new knowledge encourage hallucinations? In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pp. 7765–7784, 2024b.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 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.

Maggie Huan, Yuetai Li, Tuney Zheng, Xiaoyu Xu, Seungone Kim, Minxin Du, Radha Poovendran, Graham Neubig, and Xiang Yue. Does math reasoning improve general llm capabilities? understanding transferability of llm reasoning. *arXiv preprint arXiv:2507.00432*, 2025.

Yue Huang, Jiawen Shi, Yuan Li, Chenrui Fan, Siyuan Wu, Qihui Zhang, Yixin Liu, Pan Zhou, Yao Wan, Neil Gong, et al. Metatool benchmark for large language models: Deciding whether to use tools and which to use. In *International Conference on Learning Representations*, volume 2024, pp. 42978–43007, 2024.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 2567–2577, 2019.

Seongryong Jung, Suwan Yoon, DongGeon Kim, and Hwanhee Lee. Todi: Token-wise distillation via fine-grained divergence control. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 8089–8102, 2025.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pp. 1885–1894. Pmlr, 2017.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

Gaotang Li, Ruizhong Qiu, Xiusi Chen, Heng Ji, and Hanghang Tong. Beyond log likelihood: Probability-based objectives for supervised fine-tuning across the model capability continuum. *International Conference on Machine Learning*, 2026.

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.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Jiacheng Lin, Zhongruo Wang, Kun Qian, Tian Wang, Arvind Srinivasan, Hansi Zeng, Ruochen Jiao, Xie Zhou, Jiri Gesi, Dakuo Wang, et al. Sft doesn't always hurt general capabilities: Revisiting domain-specific fine-tuning in llms. In *International Conference on Learning Representations*, volume 2026, pp. 46954–46992, 2026.

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.

Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D Lee, Danqi Chen, and Sanjeev Arora. Fine-tuning language models with just forward passes. *Advances in Neural Information Processing Systems*, 36:53038–53075, 2023.

Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017.

NVIDIA. Nemotron-agentic-v1. https://huggingface.co/datasets/nvidia/Nemotron-Agentic-v1, 2025. Hugging Face dataset.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on health, inference, and learning*, pp. 248–260. PMLR, 2022.

Shishir G Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E Gonzalez. The berkeley function calling leaderboard (bfcl): From tool use to agentic evaluation of large language models. In *Forty-second International Conference on Machine Learning*, 2025.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. *Advances in neural information processing systems*, 33: 19920–19930, 2020.

Mengjie Ren, Boxi Cao, Hongyu Lin, Cao Liu, Xianpei Han, Ke Zeng, Wan Guanglu, Xunliang Cai, and Le Sun. Learning or self-aligning? rethinking instruction fine-tuning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6090–6105, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.330. URL https://aclanthology.org/2024.acl-long.330/.

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

Sunny Sanyal, Hayden Prairie, Rudrajit Das, Ali Kavis, and Sujay Sanghavi. Upweighting easy samples in fine-tuning mitigates forgetting. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 52922–52957. PMLR, 13–19 Jul 2025a.

Sunny Sanyal, Hayden Prairie, Rudrajit Das, Ali Kavis, and Sujay Sanghavi. Upweighting easy samples in fine-tuning mitigates forgetting. *arXiv preprint arXiv:2502.02797*, 2025b.

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. Rl's razor: Why online reinforcement learning forgets less. In *International Conference on Learning Representations*, volume 2026, pp. 59839–59864, 2026.

Antoine Simoulin, Namyong Park, Xiaoyi Liu, and Grey Yang. Memory-efficient fine-tuning of transformers via token selection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 21565–21580, 2024.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A Smith, and Yejin Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9275–9293, 2020.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*, 2018.

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, 2026.

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, Amanda Bertsch, Matthew Finlayson, Hailey Schoelkopf, Alex Xie, Graham Neubig, Ilia Kulikov, and Zaid Harchaoui. From decoding to meta-generation: Inference-time algorithms for large language models. *arXiv preprint arXiv:2406.16838*, 2024.

Yongliang Wu, Yizhou Zhou, Ziheng Zhou, Yingzhe Peng, Xinyu Ye, Xinting Hu, Wenbo Zhu, Lu Qi, Ming-Hsuan Yang, et al. On the generalization of sft: A reinforcement learning perspective with reward rectification. In *International Conference on Learning Representations*, volume 2026, pp. 27550–27571, 2026.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*, 2024.

Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, et al. Clue: A chinese language understanding evaluation benchmark. In *Proceedings of the 28th international conference on computational linguistics*, pp. 4762–4772, 2020.

Wenhao Yu, Shaohang Wei, Jiahong Liu, Yifan Li, Minda Hu, Aiwei Liu, Hao Zhang, and Irwin King. Probability-entropy calibration: An elastic indicator for adaptive fine-tuning. *arXiv preprint arXiv:2602.01745*, 2026.

Runjia Zeng, Qifan Wang, Qiang Guan, Ruixiang Tang, Lifu Huang, Zhenting Wang, Xueling Zhang, Cheng Han, and Dongfang Liu. Tokenseek: Memory efficient fine tuning via instance-aware token ditching. *arXiv preprint arXiv:2601.19739*, 2026.

Hao Zhang, Zhibin Zhang, Guangxin Wu, Wanyi Ning, Jiafeng Guo, and Xueqi Cheng. Egad: Entropy-guided adaptive distillation for token-level knowledge transfer. *arXiv preprint arXiv:2605.01732*, 2026.

Qihuang Zhong, Liang Ding, Xiantao Cai, Juhua Liu, Bo Du, and Dacheng Tao. KaFT: Knowledge-aware fine-tuning for boosting LLMs' domain-specific question-answering performance. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 24085–24100, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1235. URL https://aclanthology.org/2025.findings-acl.1235/.

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

CONTENTS OF THE APPENDIX

A Pseudocode **15**

B Experimental Details **15**

B.1 Training Datasets and Domains 15

B.2 Evaluation Benchmarks and Metrics 16

B.3 Training Configuration 17

B.4 Evaluation Configuration 17

C Additional Experimental Results **18**

C.1 Online vs. Post-hoc Conflict Masking 18

C.2 Detailed Mathematical Reasoning Results 19

C.3 Results Across Code, Medical, and Agent Domains 19

C.4 Sequential Continual Fine-Tuning 19

C.5 Scalability Across Model Sizes. 19

D Additional Analysis **20**

D.1 Forecasting Score Distribution. 20

D.2 Threshold Analysis 20

E Additional Qualitative Examples **20**

E.1 Reasoning Comparison 21

E.2 Examples of Conflict Trajectories 21

A PSEUDOCODE

Algorithm 1 Pre-SFT Trajectory Forecasting and Selective Fine-Tuning

Require: Pretrained model θ_0, SFT dataset $\mathcal{D}$, probability threshold τ_p, entropy threshold τ_H
Ensure: Fine-tuned model $\theta^\star$

Stage 1: Identify initial conflicts and compute SFT direction

$\mathcal{T} \leftarrow$ all supervised token positions in $\mathcal{D}$
Compute $p_i = p_i(\theta_0)$ and normalized top-K entropy $H_i = H_i(\theta_0)$ for all $i \in \mathcal{T}$
$\mathcal{C} \leftarrow \{i \in \mathcal{T} : p_i < \tau_p \wedge H_i < \tau_H\}$
$G \leftarrow \nabla_{\theta_{\mathrm{tr}}} \mathcal{L}_{\mathrm{SFT}}(\theta_0)$
$D \leftarrow -G$

Stage 2: Forecast token learning trajectories

$h \leftarrow \epsilon \dfrac{\|\theta_{0,\mathrm{tr}}\|_2}{\|D\|_2}$
Evaluate $\log p_i^+, H_i^+$ at $\theta_0 + hD$ for all $i \in \mathcal{C}$
Evaluate $\log p_i^-, H_i^-$ at $\theta_0 - hD$ for all $i \in \mathcal{C}$
for all $i \in \mathcal{C}$ **do**
    $v_i^p \leftarrow \dfrac{\log p_i^+ - \log p_i^-}{2h}$
    $v_i^H \leftarrow \dfrac{H_i^+ - H_i^-}{2h}$
    $d_i^p \leftarrow \log \tau_p - \log p_i$
    $d_i^H \leftarrow \tau_H - H_i$
    $s_i \leftarrow \max\left(\dfrac{v_i^p}{d_i^p}, \dfrac{v_i^H}{d_i^H}\right)$
end for

Stage 3: Assign predicted trajectories

$z_i \leftarrow \mathrm{sign}(s_i)\log(1 + |s_i|)$ for all $i \in \mathcal{C}$
Fit a three-component GMM to $\{z_i\}_{i \in \mathcal{C}}$
Obtain score boundaries $\gamma_L < \gamma_E$ from adjacent GMM posterior intersections
for all $i \in \mathcal{C}$ **do**
    if $s_i \le \gamma_L$ **then**
        $\hat{z}_i \leftarrow \mathrm{Late}$
    else if $s_i < \gamma_E$ **then**
        $\hat{z}_i \leftarrow \mathrm{Intermediate}$
    else
        $\hat{z}_i \leftarrow \mathrm{Early}$
    end if
end for

Stage 4: Construct mask and perform selective SFT

$m_i \leftarrow 1$ for all $i \in \mathcal{T}$
for all $i \in \mathcal{C}$ **do**
    if $\hat{z}_i \in \{\mathrm{Early}, \mathrm{Late}\}$ **then**
        $m_i \leftarrow 0$
    end if
end for
$\theta^\star \leftarrow \mathrm{SFT}(\theta_0, \mathcal{D}; \{m_i\})$ ▷ Keep the mask fixed throughout training
return $\theta^\star$

B EXPERIMENTAL DETAILS

B.1 TRAINING DATASETS AND DOMAINS

We evaluate domain-specific adaptation across four substantially different settings: mathematical reasoning, medical reasoning, agentic tool use, and code generation. For controlled comparison, we

sample 10,000 training examples from each domain and use the same examples across all methods and model backbones Table 6.

Mathematical Reasoning: NuminaMath. NuminaMath (Li et al., 2024) is a large-scale mathematical reasoning dataset containing competition-style problems paired with detailed solution traces. It spans a broad range of mathematical topics and difficulty levels, making it well suited for studying adaptation on multi-step reasoning. We use a fixed subset of 10,000 examples for training.

Medical Reasoning: Medical-O1-Reasoning-SFT. Medical-O1-Reasoning-SFT (Chen et al., 2024) contains medical questions paired with reasoning-oriented responses designed to encourage step-by-step clinical problem solving. The dataset requires both domain-specific medical knowledge and structured reasoning, providing a substantially different adaptation distribution from mathematics. We use 10,000 examples for training.

Agentic Tool Use: Nemotron Agentic Tool Use. Nemotron Agentic Tool Use (NVIDIA, 2025) contains interaction trajectories in which models must interpret user requests, select appropriate tools or functions, and produce structured tool calls. This domain introduces highly structured outputs and action-oriented supervision rather than free-form reasoning. We sample 10,000 trajectories for training.

Code Generation: Magicoder-OSS-Instruct-75K. Magicoder-OSS-Instruct-75K (Wei et al., 2023) is an instruction-tuning dataset constructed for code generation and programming-related reasoning. Its examples pair natural-language programming instructions with code-oriented responses across diverse programming tasks. We use a fixed subset of 10,000 examples for code-domain adaptation.

Table 6: Training datasets used for domain-specific adaptation.

Domain	Training Dataset	# Examples
Math	NuminaMath	10,000
Medical	Medical-O1-Reasoning-SFT	10,000
Agent	Nemotron Agentic Tool Use	10,000
Code	Magicoder-OSS-Instruct-75K	10,000

B.2 Evaluation Benchmarks and Metrics

We evaluate each fine-tuned model on benchmarks from its corresponding target domain and separately measure preservation of general capabilities. The evaluation benchmarks and metrics are summarized in Table 7.

Mathematical Reasoning. We evaluate mathematical reasoning on five benchmarks with complementary difficulty and problem distributions. **AMC** consists of competition-style mathematics problems requiring short mathematical reasoning; **AIME** provides substantially harder competition problems requiring precise numerical answers; **Minerva Math** (Lewkowycz et al., 2022) covers mathematical reasoning across areas such as algebra, calculus, and other quantitative domains; **MATH-OAI** (Lightman et al., 2024) evaluates multi-step mathematical problem solving; and **OlympiadBench** (He et al., 2024) contains challenging olympiad-level reasoning problems. We report both Pass@1 and Pass@16, measuring performance with a single sampled solution and with multiple candidate solutions, respectively.

Medical Reasoning. We evaluate medical adaptation using **MedMCQA** (Pal et al., 2022), **MedQA** (Jin et al., 2021), and **PubMedQA** (Jin et al., 2019). MedMCQA contains multiple-choice questions covering a broad range of medical subjects, while MedQA evaluates medical knowledge and clinical reasoning using medical examination questions. PubMedQA focuses on biomedical research questions grounded in scientific literature. We report accuracy on all three benchmarks.

Agentic Tool Use. We evaluate agent capabilities using **BFCL** (Patil et al., 2025) and **MetaTool** (Huang et al., 2024). BFCL evaluates whether models can correctly interpret requests and generate appropriate function calls, whereas MetaTool tests whether a model can determine whether tool use is necessary and select the appropriate tool. We report the official benchmark score for each dataset.

Code Generation. We use **HumanEval+** (Liu et al., 2023)and **MBPP+** (Austin et al., 2021) to evaluate functional code generation. Both extend standard code-generation benchmarks with additional test cases, providing a stricter measure of whether generated programs satisfy the intended functionality. We report Pass@1 based on execution-based correctness.

General Capabilities. We use **MMLU** (Hendrycks et al., 2020), **IFEval** (Zhou et al., 2023), and **CLUEWSC** (Xu et al., 2020) to quantify capability retention after domain-specific fine-tuning. MMLU evaluates broad knowledge and reasoning across diverse academic and professional subjects, IFEval measures instruction-following ability under objectively verifiable constraints, and CLUEWSC evaluates language understanding through Winograd-style coreference resolution. We report accuracy on all three benchmarks and use their mean as the general-domain average.

Table 7: Evaluation benchmarks and metrics used across adaptation domains.

Domain	Benchmarks	Metric
Math	AMC, AIME, Minerva Math, MATH-OAI, OlympiadBench	Pass@1 / Pass@16
Medical	MedMCQA, MedQA, PubMedQA	Accuracy
Agent	BFCL, MetaTool	Benchmark score
Code	HumanEval+, MBPP+	Pass@1
General	MMLU, IFEval, CLUEWSC	Accuracy

For compact comparison in the main paper, we additionally report the mean performance within each target domain and the average over MMLU and IFEval as the general-domain score.

B.3 Training Configuration

We use a consistent training configuration across all adaptation domains and compared methods. All models are fine-tuned using LoRA with rank 64 for two epochs. We use a maximum sequence length of 2,048 tokens and train on the fixed 10,000-example subset described in Appendix B.1.

The per-device batch size is 2 with 8 gradient-accumulation steps. We use an initial learning rate of 1×10^{-4} with cosine decay and a warmup ratio of 0.1. Training is performed in `bfloat16` precision with gradient checkpointing enabled. Table 8 summarizes the shared training configuration.

Table 8: Training configuration used across adaptation domains.

Hyperparameter	Value
Fine-tuning method	LoRA
LoRA rank	64
Training epochs	2
Training examples	10,000
Maximum sequence length	2,048
Per-device batch size	2
Gradient accumulation	8
Learning rate	1×10^{-4}
Learning-rate schedule	Cosine
Warmup ratio	0.1
Precision	bfloat16
Gradient checkpointing	Enabled

We use the tokenizer and chat template associated with each model family. For Qwen3-8B, we use the `qwen3` template, while Llama-3.1-8B-Instruct uses the `llama3` template. Apart from these model-specific formatting choices, the training configuration is kept fixed across model backbones and domains.

B.4 Evaluation Configuration

We keep the evaluation configuration fixed across Base, SFT, all baselines, and our method within each model backbone. Table 9 summarizes the decoding and scoring settings used across bench-

marks. We use vLLM for generation-based evaluation and the official evaluation procedure associated with each benchmark whenever available.

Table 9: Evaluation configuration used across benchmarks. All settings are kept fixed across compared methods within each model backbone.

Benchmark(s)	Decoding	# Samples	Max Tokens	Metric
Math	$T = 1.0$, top-$p = 1.0$	16	4096	Pass@1 / Pass@16
Medical	Deterministic	1	8192	Accuracy
BFCL	Official BFCL	1	$\leq$4096	BFCL score
MetaTool	Deterministic	1	512	MetaTool score
HumanEval+, MBPP+	Greedy ($T = 0$)	1	4096	Pass@1
MMLU	Likelihood scoring	–	–	Accuracy
IFEval	Greedy ($T = 0$)	1	1280	Strict accuracy
CLUEWSC	Greedy ($T = 0$)	1	1024	Accuracy

Mathematical Reasoning. For AMC, AIME, Minerva Math, MATH-OAI, and OlympiadBench, we generate 16 independent responses per problem using temperature $T = 1.0$ and top-$p = 1.0$. Final answers are extracted from the generated solutions and matched against the reference answers. We compute Pass@1 and Pass@16 from the same set of 16 generations. The sampling configuration and random seed are kept identical across all compared methods.

Medical Reasoning. We evaluate MedMCQA, MedQA, and PubMedQA in a zero-shot setting using their benchmark-specific multiple-choice formats. Predictions are generated deterministically and mapped to the corresponding answer choice. We report accuracy over the full evaluation set.

Agentic Tool Use. For BFCL, we follow the official benchmark pipeline: model responses are first generated for the tool-use requests and are subsequently evaluated using the official function-call parsing and scoring procedure. MetaTool is evaluated using its benchmark-specific prompts and scoring protocol. The same generation configuration is used for all compared methods.

Code Generation. We evaluate HumanEval+ and MBPP+ using the EvalPlus framework. We use greedy decoding ($T = 0$), generate one solution per problem, and allow up to 4,096 generated tokens. Generated programs are executed against the extended *plus* test cases, and we report Pass@1 based on functional correctness.

General Capabilities. MMLU is evaluated using multiple-choice likelihood scoring rather than free-form generation. For IFEval, we use the full 541-prompt zero-shot evaluation with greedy decoding and a maximum generation length of 1,280 tokens, and report prompt-level strict accuracy. CLUEWSC is evaluated using greedy decoding with a maximum generation length of 1,024 tokens, and we report classification accuracy.

Model-Specific Formatting. We preserve the intended inference behavior of each model family while keeping it fixed across methods. For Qwen3, reasoning is enabled for mathematical and medical reasoning tasks, while direct non-thinking generation is used for agent, code, IFEval, and CLUEWSC evaluations. Llama-3.1-8B-Instruct uses its native chat template, and DeepSeek-R1-0528-Qwen3-8B retains its native reasoning behavior. These model-specific settings are identical for the Base model and all corresponding fine-tuned variants.

as standard SFT.

C Additional Experimental Results

C.1 Online vs. Post-hoc Conflict Masking

The main motivation for this comparison is to understand whether Late conflicts can be handled effectively during training, or whether they must be identified beforehand. Since Late conflicts remain unresolved for a large portion of SFT, an online strategy can only recognize them after they have already contributed to many optimization updates. In contrast, post-hoc masking uses their complete observed trajectories from a previous SFT run and removes them from the start.

Table 10: Illustrative sample run comparing standard SFT, online masking, and post-hoc masking of Late conflicts on Qwen3-8B. Online masking is applied only after Late conflicts can be identified during training, whereas post-hoc masking uses their complete observed trajectories from a previous SFT run.

Method	AMC	MATH-OAI
SFT	29.37	56.44
Online	29.43	57.56
Post-hoc	40.60	69.42

Table 10 provides an illustrative sample run comparing standard SFT, online masking, and post-hoc masking of Late conflicts. Online masking remains close to standard SFT, while post-hoc masking yields much larger gains on both AMC and MATH-OAI. This supports the need to identify persistent conflicts before training begins, while avoiding the extra full SFT run required by a post-hoc strategy.

C.2 Detailed Mathematical Reasoning Results

Table 16 reports the complete results across all mathematical reasoning and general-domain benchmarks. ForeToken shows the strongest gains on Qwen3-8B and DeepSeek-R1-0528-Qwen3-8B, improving consistently across the math benchmarks while largely preserving general-domain performance. On Llama-3.1-8B-Instruct, the gains are more mixed, but ForeToken remains competitive and achieves strong results on several reasoning benchmarks. Overall, the detailed results show that the improvement is not driven by a single benchmark, but extends across multiple mathematical reasoning tasks.

C.3 Results Across Code, Medical, and Agent Domains

Tables 12–14 report the detailed results across code, medical, and agent domains. Across all three settings, ForeToken consistently improves over standard SFT and TALR on the target domain while largely preserving general-domain performance. It achieves the strongest fine-tuned performance on code and medical adaptation, and nearly matches the base model on the agent domain while obtaining the highest general-domain average. These results show that the gains extend beyond mathematical reasoning to diverse adaptation domains.

C.4 Sequential Continual Fine-Tuning

Table 15 reports stage-wise performance under the sequence Math $\rightarrow$ Agent $\rightarrow$ Medical $\rightarrow$ Code. Across the sequence, standard SFT shows substantial degradation on previously learned domains, whereas ForeToken retains much stronger performance while continuing to adapt to new domains. The advantage is especially clear after later stages, where ForeToken preserves substantially higher Math, Agent, Medical, Code, and general-domain averages than standard SFT. These results show that the proposed conflict selection remains effective even when adaptation is performed sequentially across multiple domains.

C.5 Scalability Across Model Sizes.

Figure 8 evaluates ForeToken on Qwen3 models ranging from 1.7B to 8B parameters. ForeToken improves Pass@1 over standard SFT at all three model sizes and on every benchmark. Although absolute performance does not increase monotonically with model size, the gains hold at every scale, indicating that the benefits of trajectory-guided masking are not tied to a particular model capacity.

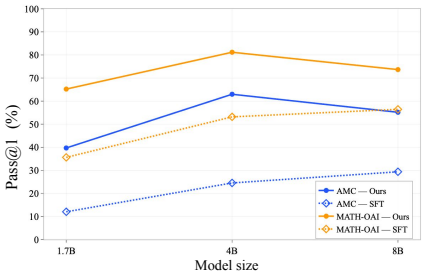


Figure 8: Scalibility.

D Additional Analysis

D.1 Forecasting Score Distribution.

Figure 9 examines how pre-SFT forecasting scores vary across the observed trajectory groups. The median score decreases from Early (0.153) to Intermediate (0.081) and Late (0.038), consistent with the expected relationship between higher scores and earlier conflict resolution. However, the distributions overlap substantially, indicating that the forecasting score captures a group-level tendency rather than providing a clear separation between individual trajectories.

D.2 Threshold Analysis

Threshold Sensitivity. Table 11 examines the effect of varying the probability and entropy thresholds independently. Although the proportion of identified conflicts changes only modestly (8.26–10.40%), performance varies noticeably, particularly for Pass@1. The default setting ($\tau_p = \tau_H = 0.30$) achieves the highest AMC Pass@1 and MATH-OAI Pass@15 among the tested configurations. Increasing τ_H to 0.35 improves MATH-OAI Pass@1 but reduces performance on the other reported metrics. These results highlight the importance of conflict selection and the trade-offs introduced by different thresholds.

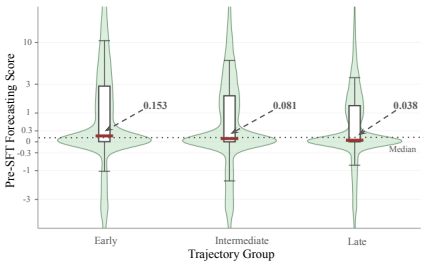


Figure 9: Distribution of pre-SFT forecasting scores across observed Early, Intermediate, and Late trajectory groups. Red horizontal markers indicate group medians.

Table 11: Threshold sensitivity on Qwen3-1.7B. We vary one conflict-identification threshold at a time while keeping the other fixed. q_p and q_H are the empirical percentile ranks of the corresponding raw thresholds, and Conflict denotes the percentage of supervised tokens satisfying both conflict conditions. Bold indicates the default configuration.

Conflict Thresholds					AMC		MATH-OAI	
τ_p	q_p (%)	τ_H	q_H (%)	Conflict (%)	Pass@1	Pass@15	Pass@1	Pass@15
0.30	**16.22**	**0.30**	**89.11**	**9.35**	**39.69**	**79.38**	**65.21**	**91.75**
0.25	15.29	0.30	89.11	8.96	39.38	79.22	63.30	90.74
0.35	17.11	0.30	89.11	9.74	37.81	79.53	59.06	90.25
0.30	16.22	0.25	86.30	8.26	32.97	79.06	56.96	89.15
0.30	16.22	0.35	91.42	10.40	37.34	77.19	72.30	89.23

E Additional Qualitative Examples

E.1 Reasoning Comparison

Figure 10 provides a qualitative example on AMC23. Both standard SFT and ForeToken reach the same intermediate equation, but standard SFT makes an algebraic error when collecting terms and propagates this mistake to the final answer. In contrast, ForeToken preserves the correct reasoning and reaches the ground-truth answer. The difference is also consistent across sampled generations: standard SFT succeeds in only $1/16$ samples, whereas ForeToken succeeds in all $16/16$.

AMC23: Qualitative Comparison

Question. A triangle has side lengths in arithmetic progression, with smallest side 6 and one angle 120°. Its area is $m\sqrt{n}$ with n squarefree. Find $m+n$. **Ground truth:** $\boxed{18}$

Shared reasoning. Both methods apply the Law of Cosines to the sides 6, $6+x$ and $6+2x$, and reach the same expanded equation:

$$(6+2x)^2 = 6^2 + (6+x)^2 + 6(6+x)$$

$$36 + 24x + 4x^2 \;=\; 36 + 36 + 12x + x^2 + 36 + 6x$$

	Standard SFT 1/16 correct	ForeToken 16/16 correct
Collect terms	$4x^2 - 6x - 36 = 0$	$3x^2 + 6x - 72 = 0$
Reduce	$2x^2 - 3x - 18 = 0$	$x^2 + 2x - 24 = 0$
Solve	$x = 3$ or -3	$(x+6)(x-4) = 0 \Rightarrow x = 4$
Side lengths	$6,\ 9,\ 12$	$6,\ 10,\ 14$
Area	$\frac{27\sqrt{3}}{2}$	$15\sqrt{3}$
Answer	$\boxed{30}$ ✗	$\boxed{18}$ ✓

Key observation. Both methods reach the same expanded equation. Standard SFT then collects terms incorrectly and subsequently misidentifies the roots of the resulting quadratic, while our method completes the derivation correctly.

Figure 10: **Qualitative comparison on AMC23 (Qwen3-8B).** Standard SFT solves this problem in 1 of 16 sampled generations, our method in 16 of 16; one generation from each is shown (SFT generation 1, ours generation 4). The problem statement and derivations are abridged, our method's variable d is renamed x for alignment, and the right-hand side of the shared equation is shown ungrouped, where our method writes $36 + 36$ as 72. No other changes were made.

E.2 Examples of Conflict Trajectories

We provide additional examples illustrating how initially conflicting tokens can exhibit different learning trajectories during SFT. Colored tokens denote the PRETRAINING–SUPERVISION CONFLICTS, while the gray token in $\langle\cdot\rangle$ denotes the pretrained model's top-1 prediction at the same supervised position.

Early **Intermediate** **Late**

Example 1: Eigenvalue

QUESTION
Using Kellogg's method, find the smallest eigenvalue of the kernel $K(x,t) = x^2t^2$, where $0 \le x, t \le 1$.

PRETRAINING–SUPERVISION CONFLICTS
To **calculate** ⟨**find**⟩ the smallest **characteristic** ⟨**eigen**⟩ **number** ⟨**value**⟩ (eigenvalue) of the kernel $K(x,t) = x^2t^2$, we perform the following steps to obtain the required eigenvalue.

Example 2: Scientific Notation

QUESTION
The number 0.000000125 can be expressed in scientific notation as:

$$\text{A: } 0.125 \times 10^{-7} \qquad \text{B: } 125 \times 10^{-6}$$
$$\text{C: } 1.25 \times 10^{-7} \qquad \text{D: } 0.125 \times 10^{-8}$$

PRETRAINING–SUPERVISION CONFLICTS
To express the number 0.000000125 in scientific notation, we **move** ⟨**need**⟩ the decimal point seven places to the right, which gives us 1.25. This movement is equivalent to **dividing** ⟨**multiplying**⟩ the **number** ⟨**original**⟩ by 10^7. Therefore,

$$0.000000125 = 1.25 \times 10^{-7}.$$

This matches option C in the given choices.

Example 3: Rectangles from Unit Squares

QUESTION
Find the sum of the areas of all distinct rectangles that can be formed from 9 unit squares, where the side length of each square is 1 cm.

PRETRAINING–SUPERVISION CONFLICTS
We need to find the sum of the areas of all **different** ⟨**distinct**⟩ rectangles that can be **composed** ⟨**formed**⟩ using these 9 squares.
If one side has length $a = 1$, then the other side can take any **value** ⟨**integer**⟩ from 1 to 9. We similarly enumerate the remaining possible side-length combinations and sum their areas.

Table 12: Performance comparison on code and general-domain benchmarks.

Method	Code Domain			General Domain			
	HumanEval+	MBPP+	Code Avg.	MMLU	IFEval	CLUEWSC	Gen. Avg.
Base	78.0	61.4	69.7	72.9	80.5	79.6	77.7
SFT	74.4	60.6	67.5	72.4	78.3	76.3	75.7
TALR	72	65.3	68.6	73.1	79.5	75.7	76.1
ForeToken	**78.7**	**63.0**	**70.9**	**72.6**	**80.2**	**78.3**	**77.0**

Table 13: Performance comparison on medical and general-domain benchmarks.

Method	Medical Domain				General Domain			
	MedMCQA	MedQA	PubMedQA	Medical Avg.	MMLU	IFEval	CLUEWSC	Gen. Avg.
Base	66.0	76.9	74.2	72.4	72.9	80.5	79.6	77.7
SFT	58.5	69.6	68.6	65.6	72.9	78.0	77.6	76.2
TALR	58.6	69.4	68.8	65.6	73.1	78.2	78.6	76.6
ForeToken	**62.4**	**75.0**	**71.6**	**69.7**	**72.9**	**80.4**	**80.5**	**77.9**

Example 4: Number Bases

QUESTION
Express the product of 1011_2 and 102_3 in base 10.

PRETRAINING–SUPERVISION CONFLICTS
First, convert 1011_2 to **decimal** ⟨base⟩:

$$1011_2 = 1(2^3) + 0(2^2) + 1(2^1) + 1(2^0) = 11.$$

Similarly,

$$102_3 = 1(3^2) + 0(3^1) + 2(3^0) = 11.$$

Next, **Calculate** ⟨Multiply⟩ the product of these decimal **values** ⟨numbers⟩:

$$11 \times 11 = 121.$$

Therefore, the answer is

$$\boxed{121}.$$

Example 5: Solid Geometry — Surface Area

QUESTION
The right figure is a three-view drawing of a solid. Based on the data in the figure, the surface area of the solid is:

A: 9π B: 10π C: 11π D: 12π.

The original problem contains a three-view drawing; we omit it here since our focus is on the token-level conflicts in the supervised response.

PRETRAINING–SUPERVISION CONFLICTS
Analysis: The solid is a combination of a cylinder and a **sphere** ⟨cone⟩. The radius of the sphere is 1, and the radius of the base of the cylinder, which has a height of **3** ⟨2⟩, is also 1. Using the surface-area formula, we calculate

$$S = 4\pi + \mathbf{3}_{\langle 2 \rangle} \times 2\pi + \pi + \pi = \boxed{12\pi}.$$

Table 14: Performance comparison on agent and general-domain benchmarks.

Method	Agent Domain			General Domain			
	BFCL	MetaTool	Agent Avg.	MMLU	IFEval	CLUEWSC	Gen. Avg.
Base	64.8	75.9	70.4	72.9	80.5	79.6	77.7
SFT	56.6	75.9	66.3	72.3	82.1	75.6	76.7
TALR	58.7	72.3	65.5	72.2	81.7	76.3	76.7
ForeToken	**64.2**	**75.7**	**70.0**	**72.7**	**82.5**	**79.0**	**78.1**

Table 15: Stage-wise continual learning performance under the sequence Math → Agent → Medical → Code. All values are percentages. **Avg.** denotes the mean across benchmarks within each domain.

		Math			Agent			Medical			Code			General			
Stage	Method	AMC	MATH-OAI	Avg.	BFCL	MetaTool	Avg.	MedMCQA	MedQA	Avg.	HumanEval+	MBPP+	Avg.	MMLU	IFEval	CLUEWSC	Avg.
	Base	34.5	65.1	49.8	64.8	76.0	70.4	66.0	76.9	71.5	78.0	61.4	69.7	73.0	80.6	79.6	77.7
After Math	SFT	29.4	56.4	42.9	58.9	78.0	68.4	58.5	64.6	61.5	78.7	60.1	69.4	72.1	79.1	75.9	75.7
	ForeToken	**55.2**	**73.7**	**64.4**	**60.4**	**78.0**	**69.2**	**62.7**	**74.2**	**68.5**	**75.6**	**63.2**	**69.4**	**73.5**	**80.6**	**77.0**	**77.0**
	Base	34.5	65.1	49.8	64.8	76.0	70.4	66.0	76.9	71.5	78.0	61.4	69.7	73.0	80.6	79.6	77.7
After Agent	SFT	34.1	64.2	49.1	58.8	63.0	60.9	58.0	74.2	66.1	74.4	57.9	66.2	71.8	77.5	75.3	74.9
	ForeToken	**59.7**	**63.9**	**61.8**	**61.3**	**75.0**	**68.2**	**63.3**	**75.5**	**69.4**	**75.0**	**61.9**	**68.5**	**72.0**	**80.0**	**76.3**	**76.1**
	Base	34.5	65.1	49.8	64.8	76.0	70.4	66.0	76.9	71.5	78.0	61.4	69.7	73.0	80.6	79.6	77.7
After Medical	SFT	25.6	63.5	44.6	38.8	69.9	54.4	57.5	71.4	64.5	77.4	60.8	69.1	72.5	74.9	75.7	74.3
	ForeToken	**63.1**	**80.0**	**71.6**	**61.6**	**73.3**	**67.5**	**60.2**	**71.8**	**66.0**	**76.2**	**62.7**	**69.5**	**72.2**	**77.5**	**77.3**	**75.7**
	Base	34.5	65.1	49.8	64.8	76.0	70.4	66.0	76.9	71.5	78.0	61.4	69.7	73.0	80.6	79.6	77.7
After Code	SFT	22.8	72.2	47.5	59.4	71.5	65.4	57.4	67.8	62.6	63.5	63.5	63.5	72.6	75.2	73.4	73.7
	ForeToken	**49.7**	**72.0**	**60.8**	**59.1**	**74.1**	**66.6**	**60.1**	**70.6**	**65.4**	**73.2**	**62.4**	**67.8**	**72.2**	**75.6**	**75.7**	**74.5**

Table 16: Performance comparison on mathematical reasoning and general-domain benchmarks. We report Pass@1 (P@1) and Pass@16 (P@16) for the math-domain benchmarks, and accuracy (Acc.) for MMLU, IFEval, and CLUEWSC. Best results among fine-tuned methods, excluding the base model, are shown in **bold**; ties at the reported precision are all highlighted.

Model	Method	Math Domain										General Domain		
		AMC		AIME		Minerva		MATH-OAI		Olympiad		MMLU	IFEval	CLUEWSC
		P@1	P@16	P@1	P@16	P@1	P@16	P@1	P@16	P@1	P@16	Acc.	Acc.	Acc.
Qwen3-8B	Base	34.5	70.0	6.0	26.7	31.4	48.5	65.1	87.4	27.2	51.1	73.0	80.6	79.6
	SFT	29.4	77.5	2.5	16.7	21.8	55.1	56.4	89.2	21.0	56.3	72.1	79.1	75.9
	EAFT	37.0	75.0	4.6	20.0	19.8	51.8	55.7	88.4	21.5	56.4	71.8	78.0	76.6
	DFT	41.1	72.5	4.2	16.7	26.1	40.1	69.2	85.0	32.6	54.8	72.6	**80.6**	76.6
	TALR	43.4	77.5	6.7	16.7	35.7	60.3	68.8	87.4	32.9	57.8	73.1	78.9	**77.6**
	RankTuner	46.6	**90.0**	6.7	30.0	28.0	52.9	69.5	89.0	34.1	60.3	73.1	79.7	74.3
	ForeToken	**55.2**	**90.0**	**12.5**	**46.7**	**37.2**	**65.4**	**73.7**	**94.6**	**37.6**	**69.9**	**73.5**	**80.6**	77.0
DeepSeek-R1-0528 Qwen3-8B	Base	25.6	65.0	3.5	23.3	19.6	44.1	38.4	73.6	11.5	35.4	69.7	68.6	90.5
	SFT	23.4	67.5	3.1	16.7	19.3	52.2	51.2	86.8	17.1	52.0	70.6	52.9	81.9
	EAFT	22.7	72.5	4.8	23.3	17.0	49.3	51.0	87.2	17.4	50.8	70.8	51.8	84.5
	DFT	33.4	65.0	**10.2**	23.3	25.4	51.1	65.3	86.6	28.5	54.2	**71.2**	60.4	**90.5**
	TALR	33.3	67.5	10.0	23.3	25.6	51.1	64.8	86.4	28.0	51.6	70.5	53.1	**90.5**
	RankTuner	38.4	82.5	9.2	**30.0**	24.5	53.3	64.2	90.0	28.0	57.6	68.2	**65.6**	75.7
	ForeToken	**44.5**	**87.5**	9.2	26.7	**27.8**	**60.7**	**66.1**	**91.8**	**30.6**	**62.1**	70.9	62.7	85.2
Llama-3.1-8B Instruct	Base	18.8	60.0	2.5	23.3	16.0	43.0	38.2	78.8	10.1	10.1	68.5	74.5	69.1
	SFT	13.8	55.0	0.8	10.0	13.1	43.4	38.7	77.2	10.6	39.4	66.0	71.2	64.1
	EAFT	14.4	62.5	2.9	23.3	13.2	43.8	37.1	77.8	10.1	39.3	66.1	69.9	60.5
	DFT	26.3	**72.5**	2.1	20.0	17.5	40.4	46.3	75.4	14.4	39.3	**66.6**	68.6	**72.4**
	TALR	**27.2**	67.5	**5.0**	**26.7**	**18.8**	41.2	46.3	75.6	**15.1**	40.9	66.0	71.7	68.8
	RankTuner	20.8	70.0	4.0	16.7	18.4	45.2	**46.7**	79.8	14.6	41.9	64.5	68.2	60.2
	ForeToken	21.2	**72.5**	2.3	16.7	16.0	**46.7**	43.7	**81.6**	13.6	**44.4**	66.3	**72.3**	71.7