# UNDERSTANDING AND STEERING SUPERVISED FINE-TUNING THROUGH ENTROPY FLOW

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

## ABSTRACT

Supervised fine-tuning increases the likelihood of externally provided demonstrations, but likelihood improvement alone does not explain how supervision reshapes a model's predictive distribution. We investigate this process through entropy flow: the local entropy responses to target-token supervision and their evolution throughout training. This perspective connects model-state-dependent selection and weighting with the allocation of supervision across tokens and examples, while distinguishing local logit-space responses from their effects under shared model parameters. Our aim is to understand and steer fine-tuning through these evolving responses rather than to assess learning solely by the likelihood of observed tokens.

## 1 INTRODUCTION

Supervised fine-tuning (SFT) adapts a pretrained language model by increasing the likelihood of externally supplied demonstrations. These demonstrations provide concrete examples of desired behavior, but only limited coverage of the possible responses to a prompt: valid answers may differ in wording, style, and reasoning path (Li et al., 2025). Learning from them therefore does more than improve target likelihood. It also redistributes predictive support among alternative continuations, shaping which paths the model will follow during generation.

The model's readiness to follow those paths matters. Semantically equivalent expressions need not be equally easy for the current model to continue reliably. Increasing the probability of entering a demonstrated path may outpace learning its subsequent conditional distributions. Work on entropy calibration links autoregressive error accumulation to increasing uncertainty during generation, highlighting the distinction between prediction under observed contexts and reliable free generation (Cao et al., 2025). This distinction also motivates comparisons with on-policy reinforcement learning (RL), where candidate responses are sampled from the current policy and reward-driven updates reinforce or suppress sampled behavior. SFT instead learns from fixed external responses. These different sources of supervision motivate asking how demonstrations interact with the model's evolving predictive distribution (Zhang et al., 2026b; Wang et al., 2026).

Recent approaches address this interaction by assigning nonuniform weights to supervised tokens. They use signals such as target probability (Wu et al., 2026), predictive entropy (Diao et al., 2026), frozen pretrained support (Wang et al., 2026), and centered log-likelihood (Zhang et al., 2026b). These methods establish useful ways to characterize and select supervision. Our focus is the effect of the update itself: *how would learning from a particular demonstrated token locally reshape the predictive distribution, and how does this response evolve throughout training?*

We study this question using the local entropy response along the target-token negative-log-likelihood (NLL) descent direction in logit space. An entropy-decreasing response identifies supervision that increases target likelihood while locally concentrating predictions; we interpret this as supervision-supported consolidation of the model's current predictive preferences. An entropy-increasing response instead reduces concentration and changes the balance among candidates, potentially supporting content that the model does not yet sufficiently favor. Related RL research also uses update-induced entropy changes to interpret training dynamics and guide token-level interventions (Hao et al., 2025; Xu et al., 2026; Zhang et al., 2026a). Here, we examine these responses under external demonstration-driven supervision, connecting their evolution to fitting behavior and the allocation of learning signals.



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.

As a simple starting point, we introduce Compatibility-Aware Fine-Tuning (CAFT), which prioritizes entropy-decreasing supervision and continuously weights each token by its response magnitude. The weights are recomputed from the current model and detached during backpropagation, preserving each token's NLL gradient direction while changing its contribution. This provides a response-based allocation rule rather than a fixed classification of demonstrations. We use a binary CAFT-Hard variant to distinguish selection from magnitude weighting, and evaluate CAFT against existing fine-tuning methods.

Beyond the initial weighting rule, we track *entropy flow*: the evolution of positive and negative local responses across fixed prefixes and training checkpoints. Our analyses show that SFT can reach a stable state with substantial, nearly balanced responses on Gold prefixes, while Gold and self-generated contexts exhibit different response magnitudes and predictive entropies. Model-expectation diagnostics and update-rule switching experiments help distinguish statistical matching from the mechanisms that shape this balance. We then examine how controlling entropy-increasing supervision and reallocating CAFT weights across examples affect training and downstream behavior. Together, these analyses treat entropy response as both a diagnostic of the model–supervision relationship and a signal for steering fine-tuning.

## 2 Related Work

**Supervision selection and reweighting.** Adaptive weighting uses model-dependent signals to allocate learning effort. Active Bias emphasizes samples using prediction-history uncertainty (Chang et al., 2017), RHO-LOSS prioritizes estimated reductions in holdout loss (Mindermann et al., 2022), and Rho-1 selects pretraining tokens by excess loss relative to a reference model (Lin et al., 2024). Recent SFT methods use target probability (Wu et al., 2026), predictive entropy (Diao et al., 2026), or centered log-likelihood (Zhang et al., 2026b). PriFT derives token support from a frozen pretrained reference (Wang et al., 2026), while ASFT combines probability weighting with reference-model anchoring (Zhu et al., 2025). CAFT uses the current entropy response to target-directed supervision, linking token weighting to the distributional effect of the update direction.

**Learning dynamics and update effects.** Training trajectories reveal model-dependent differences among examples. Dataset Cartography uses confidence and its variability across epochs (Swayamdipta et al., 2020), while forgetting-event analysis tracks transitions from correct to incorrect predictions (Toneva et al., 2019). Gradient-based approaches estimate how training examples influence other predictions: TracIn aggregates gradient interactions across checkpoints (Pruthi et al., 2020), and LESS uses optimizer-aware gradient similarity for targeted instruction-data selection (Xia et al., 2024). More directly, Ren & Sutherland (2025) decompose update-induced prediction changes during language-model fine-tuning, and Razin et al. (2025) connect likelihood displacement in preference optimization to representation geometry. Our analysis follows the evolution of local entropy responses and their positive and negative aggregate contributions under different supervision rules.

**Entropy dynamics and generation.** Entropy provides complementary views of learning and generation. High-entropy minority tokens have been linked to reasoning branches and effective RL updates (Wang et al., 2025). Other RL studies analyze update-induced entropy changes, signed contributions, and entropy-flow interventions (Hao et al., 2025; Xu et al., 2026; Zhang et al., 2026a). In supervised fine-tuning, GEM studies diversity preservation (Li et al., 2025), while entropy-calibration research relates reference-text log loss to uncertainty under model generation (Braverman et al., 2020; Cao et al., 2025). We bring these perspectives to demonstration-driven learning by tracking signed entropy responses across model states and answer sources, and using them to study the strength and allocation of supervision.

## 3 Method

### 3.1 Motivation and Entropy Response

Consider a supervised response position $i$, conditioned on the prompt and the preceding demonstration tokens. Let $V$ be the vocabulary size, $v \in \{1, \ldots, V\}$ index a candidate token, and $y_i$ denote

the observed demonstration target. The model produces a logit vector $z_i \in \mathbb{R}^V$ and the predictive distribution $p_i = \mathrm{softmax}(z_i)$, where $p_{i,v}$ is the probability assigned to candidate $v$. The target negative log-likelihood (NLL), $\ell_i$, and the Shannon entropy of the predictive distribution, $H_i$, are

$$\ell_i = -\log p_{i,y_i}, \qquad H_i = -\sum_{v=1}^{V} p_{i,v} \log p_{i,v}. \tag{1}$$

Learning the demonstration target follows the NLL descent direction in logit space, $u_i = -\nabla_{z_i} \ell_i = e_{y_i} - p_i$, where $e_{y_i}$ is the one-hot vector for the target token. For a virtual step of size $\epsilon$ along $u_i$, we define the local entropy response as

$$d_i = \frac{\mathrm{d}}{\mathrm{d}\epsilon} H(\mathrm{softmax}(z_i + \epsilon u_i))\Big|_{\epsilon=0} = \nabla_{z_i} H_i^\top u_i. \tag{2}$$

Thus $d_i < 0$ indicates local entropy decrease under target-directed supervision, whereas $d_i > 0$ indicates local entropy increase. Unlike entropy itself, which describes the current distribution, $d_i$ describes its response to a particular target.

This response admits a closed-form computation. Let $h_{i,v} = \partial H_i / \partial z_{i,v}$ denote the sensitivity of predictive entropy to the logit of candidate $v$. Then

$$h_{i,v} = -p_{i,v}(\log p_{i,v} + H_i), \qquad d_i = h_{i,y_i} - \sum_{v=1}^{V} p_{i,v} h_{i,v}. \tag{3}$$

It can therefore be evaluated using the current predictive distribution and vocabulary-wise reductions, without an additional model forward pass. We use this logit-space response as a state-dependent signal for allocating supervision; the effects of shared parameters on the resulting updates are examined in Section B.4.

### 3.2 Compatibility-Aware Fine-Tuning

We begin by assigning greater weight to targets whose local NLL directions more strongly concentrate the predictive distribution. For a prompt–response pair $(x, y)$ from the demonstration dataset $\mathcal{D}$, CAFT assigns weight $w_t = [-d_t]_+$ to response position $t$ and uses the weighted sequence objective

$$\mathcal{L}_{\mathrm{CAFT}}(\theta) = -\mathbb{E}_{(x,y)\sim\mathcal{D}} \left[ \sum_{t=1}^{|y|} \mathrm{sg}\big([-d_t]_+\big) \log p_\theta(y_t \mid x, y_{<t}) \right]. \tag{4}$$

Here $d_t$ is the local entropy response at the demonstration prefix $(x, y_{<t})$, as defined in Equation 2, and sg denotes stop-gradient.

CAFT provides a continuous allocation rule: a stronger entropy-decreasing response receives greater weight, and the weight approaches zero as the response approaches the sign boundary. Detaching the weights preserves each token's NLL gradient direction while controlling its contribution to the aggregate update. The weights are recomputed from the current model at every training step, allowing the same demonstration position to receive different emphasis as learning proceeds.

To separate selection from response-magnitude weighting, we use CAFT-Hard as a binary control, replacing $w_t$ by $\mathbf{1}[d_t < 0]$. Both rules initially prioritize entropy-decreasing supervision. This provides a simple reference point for the subsequent entropy-flow analysis, which studies how selectively restoring entropy-increasing updates changes learning and downstream performance.

## 4 Experiments

### 4.1 Experimental Setup

**Models and training.** We evaluate mathematical reasoning, code generation, and medical question answering. Our mathematical reasoning experiments build on the setup of DFT (Wu et al., 2026), fine-tuning Qwen2.5-Math-1.5B (Yang et al., 2024) on a subset of NuminaMath-CoT (Li

et al., 2024) for 390 optimizer steps with a global batch size of 256. We also evaluate Qwen2.5-Math-7B fine-tuned on OpenR1-Math-220k (Hugging Face, 2025), which pairs problems from NuminaMath 1.5 with reasoning traces generated by DeepSeek-R1. For code generation, we fine-tune Qwen2.5-Coder-3B (Hui et al., 2024) on 34,999 Tulu Code examples (Allen Institute for AI, 2024); for medical question answering, we fine-tune Qwen2.5-7B Base (Qwen Team, 2024) on 100,000 MedMCQA training examples (Pal et al., 2022). Both use one training epoch. Within each experiment, all methods share the initial model, training data, training budget, and evaluation protocol. We use AdamW throughout these experiments. The mathematical training configuration and loss normalization are detailed in Appendix A.

**Baselines.** We compare CAFT with standard NLL-based SFT and five representative fine-tuning methods. DFT (Wu et al., 2026) weights supervision by the current target-token probability; EAFT (Diao et al., 2026) uses predictive entropy; and IDFT (Zhang et al., 2026b) modulates weights using centered target log-likelihood. We also include PriFT (Wang et al., 2026) using its mass-based variant, which selects tokens using cumulative probability mass under a frozen pretrained reference, and ASFT (Zhu et al., 2025), which combines probability reweighting with reference-model anchoring. These comparisons cover both current-model signals and frozen-reference guidance. We implement the methods within each domain-specific protocol.

**Evaluation.** For mathematics, we evaluate on MATH500 (Lightman et al., 2023), Minerva Math (Lewkowycz et al., 2022), OlympiadBench (He et al., 2024), AIME24, and AMC23 (Mathematical Association of America, 2024; 2023) using 16 responses per problem at temperature 1. Avg@16 is the mean fraction of correct responses, and Pass@16 is the fraction of problems with at least one correct response; aggregate scores are unweighted macro-averages across the five benchmarks. We track 12 shared mathematical checkpoints, with greedy results in Appendix C. For code generation, we report Pass@1 on HumanEval+ and MBPP+ (Liu et al., 2023), and LiveCodeBench v6 (Jain et al., 2024), evaluating one completion per problem against executable tests. For medicine, we report answer-choice accuracy on the MedMCQA validation set, the MedQA English four-option test set (Jin et al., 2020), and 12 medical-related MMLU test subjects (Hendrycks et al., 2020). Medical predictions use the highest-probability answer option under the evaluation protocol of ASFT (Zhu et al., 2025); MMLU-Med accuracy is pooled over questions across the 12 subjects.

### 4.2 Main Results

Table 1 reports mathematical reasoning performance on Qwen2.5-Math-1.5B.

Table 1: Mathematical reasoning performance of Qwen2.5-Math-1.5B (%). Avg and Pass denote Avg@16 and Pass@16. Average is the unweighted mean over five benchmarks. The best value in each column is bold, including ties.

| Method | MATH500 | | Minerva | | OlympiadBench | | AIME24 | | AMC23 | | Average | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Avg | Pass | Avg | Pass | Avg | Pass | Avg | Pass | Avg | Pass | Avg | Pass |
| SFT | 42.8 | 81.8 | 12.1 | 42.3 | 12.8 | 46.5 | 1.4 | 10.0 | 18.4 | 67.5 | 17.50 | 49.62 |
| DFT | 65.2 | 88.8 | 23.1 | 45.6 | 28.6 | 55.6 | 5.8 | 20.0 | 39.2 | 70.0 | 32.38 | 56.00 |
| EAFT | 43.9 | 79.6 | 12.1 | 39.7 | 13.3 | 46.2 | 2.1 | 13.3 | 19.1 | 65.0 | 18.10 | 48.76 |
| IDFT | 65.7 | 85.2 | 23.4 | 42.6 | 28.0 | 53.2 | **9.2** | 16.7 | 38.4 | 70.0 | 32.94 | 53.54 |
| PriFT† | 64.3 | **89.4** | 19.0 | 47.4 | 30.0 | **57.8** | 7.7 | 23.3 | **45.8** | 82.5 | 33.36 | 60.08 |
| ASFT† | 60.0 | **89.4** | 16.9 | 45.6 | 24.4 | 57.6 | 5.4 | **26.7** | 35.0 | 72.5 | 28.34 | 58.36 |
| CAFT | **66.5** | 88.0 | **25.4** | **48.2** | **30.5** | 57.3 | 7.3 | 20.0 | 40.3 | **87.5** | **34.00** | **60.20** |

† Uses a frozen reference model.

CAFT achieves the highest macro-averaged Avg@16 and Pass@16, reaching 34.00% and 60.20%, respectively. At the benchmark level, CAFT achieves the highest Avg@16 on MATH500, Minerva Math, and OlympiadBench, including strong performance on challenging olympiad-level problems. It also obtains the highest Pass@16 on Minerva Math and AMC23, reaching 87.5% on the latter.

On Qwen2.5-Math-7B fine-tuned on OpenR1 (Table 2), CAFT achieves the highest macro-averaged Avg@16 at 44.46%, together with a near-best Pass@16 of 70.58%. It leads in Avg@16 on four of

Table 2: Mathematical reasoning performance of Qwen2.5-Math-7B fine-tuned on OpenR1 (%). Avg and Pass denote Avg@16 and Pass@16. Average is the unweighted mean over five benchmarks. The best value in each column is bold, including ties.

| Method | MATH500 | | Minerva | | OlympiadBench | | AIME24 | | AMC23 | | Average | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Avg | Pass | Avg | Pass | Avg | Pass | Avg | Pass | Avg | Pass | Avg | Pass |
| SFT | 68.3 | 88.4 | 29.1 | 54.4 | 30.4 | 57.6 | 7.7 | 16.7 | 45.0 | 72.5 | 36.10 | 57.92 |
| DFT | 57.5 | 89.6 | 26.0 | 52.6 | 25.2 | 60.0 | 5.0 | 20.0 | 39.4 | 82.5 | 30.62 | 60.94 |
| EAFT | 66.7 | 88.2 | 27.8 | 55.1 | 29.4 | 57.5 | 7.3 | 23.3 | 42.8 | 72.5 | 34.80 | 59.32 |
| IDFT | 53.9 | 89.4 | 27.8 | 53.7 | 27.3 | 62.2 | 5.2 | 16.7 | 34.8 | 70.0 | 29.80 | 58.40 |
| PriFT† | 75.0 | 92.2 | 32.1 | 57.0 | 37.9 | 64.0 | 13.1 | 43.3 | **61.6** | 90.0 | 43.94 | 69.30 |
| ASFT† | 72.8 | **94.2** | 30.9 | **58.5** | 37.1 | **68.4** | 12.3 | 40.0 | 58.8 | **92.5** | 42.38 | **70.72** |
| **CAFT** | **77.7** | 92.2 | **34.3** | 55.1 | **40.7** | 66.4 | **13.5** | **46.7** | 56.1 | **92.5** | **44.46** | 70.58 |

† Uses a frozen reference model.

the five benchmarks: MATH500, Minerva Math, OlympiadBench, and AIME24. On competition benchmarks, CAFT also achieves the highest Pass@16 on AIME24 and ties for the highest on AMC23, demonstrating strong solution coverage alongside its average-accuracy gains.

**Code generation.** Table 3 summarizes the code and medical results. On code generation, CAFT achieves the highest Pass@1 on HumanEval+ and LiveCodeBench v6, reaching 70.73% and 15.73%, respectively, and improving over the base model by 3.66 and 2.65 percentage points. MBPP+ exhibits a different pattern: none of the fine-tuning methods surpasses the base model, and only SFT matches its score. CAFT largely preserves the base model's MBPP+ performance, scoring 64.02% versus 65.87%, alongside its gains on the other two benchmarks.

**Medical question answering.** CAFT achieves the highest MMLU-Med accuracy at 75.54%. On MedQA, it reaches 62.06%, outperforming all other reference-free fine-tuning methods by at least 5.34 percentage points. On MedMCQA, CAFT ranks second at 59.00%, just 0.29 percentage points behind ASFT; these are the only two fine-tuning methods that improve over the base model.

### 4.3 Performance throughout Training

Figure 1 shows Avg@16 and Pass@16 over 12 shared checkpoints. Greedy trajectories are provided in Figure 15.

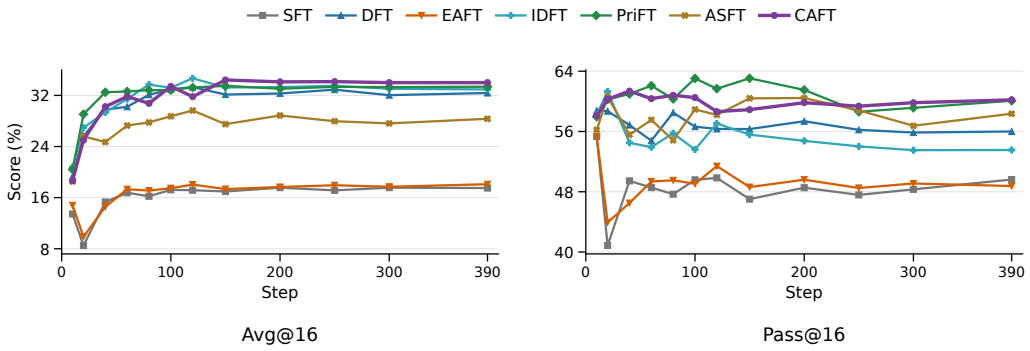



Figure 1: Mathematical reasoning performance throughout fine-tuning of Qwen2.5-Math-1.5B. Panels show Avg@16 and Pass@16, each as a five-benchmark macro-average (%) at the evaluated checkpoints. Colors and markers identify methods consistently across panels.

The training curves complement the endpoint comparison. CAFT maintains a high Avg@16 throughout the later checkpoints while retaining a relatively stable Pass@16, in contrast to the pro-

nounced early drop in Pass@16 under SFT and EAFT. Its endpoint performance therefore reflects a sustained region of the training trajectory rather than an isolated peak.

Table 3: Results on code generation and medical question answering (%). Code results report Pass@1, and medical results report accuracy. The best value in each column is bold, including ties.

| | Code: Qwen2.5-Coder-3B | | | Medicine: Qwen2.5-7B Base | | |
|---|---|---|---|---|---|---|
| Method | HumanEval+ | MBPP+ | LCB v6 | MedMCQA | MedQA | MMLU-Med |
| Base | 67.07 | **65.87** | 13.08 | 58.28 | 62.37 | 75.14 |
| SFT | 66.46 | **65.87** | 11.00 | 31.58 | 37.00 | 51.77 |
| DFT | 62.20 | 63.23 | 12.23 | 49.56 | 56.72 | 74.04 |
| EAFT | 59.76 | 65.08 | 11.56 | 35.74 | 48.70 | 57.21 |
| IDFT | 63.41 | 64.81 | 10.62 | 46.35 | 48.23 | 62.43 |
| PriFT† | 69.51 | 64.81 | 15.07 | 54.48 | 62.77 | 64.51 |
| ASFT† | 62.20 | 61.90 | 10.62 | **59.29** | **63.00** | 71.27 |
| **CAFT** | **70.73** | 64.02 | **15.73** | 59.00 | 62.06 | **75.54** |

† Uses a frozen reference model.

## 5 Analysis

**Analysis setup.** We use Qwen2.5-Math-1.5B and 1,024 fixed NuminaMath-CoT questions with 436,443 Gold response positions. Generated answers are sampled once per question from the final SFT model at temperature 1 and top-$p$ 1, then held fixed across checkpoints. Statistics cover all response positions; entropy is in nats.

### 5.1 Understanding SFT through Entropy Flow

We track positive and negative local entropy responses to characterize evolving supervision, complementing entropy-change analyses in RL (Xu et al., 2026; Zhang et al., 2026a). For $N$ positions and detached training weights $w_i$, define

$$M_{\pm} = \frac{1}{N}\sum_i [\pm d_i]_+ , \qquad M_{\pm}^{(w)} = \frac{1}{N}\sum_i w_i\,[\pm d_i]_+ , \tag{5}$$

where $+$ and $-$ denote entropy-increasing and entropy-decreasing responses. $M_{\pm}$ aggregates unweighted responses over all $N$ valid response tokens, while $M_{\pm}^{(w)}$ applies the training weights to the same positions with the same denominator. We also track sign fractions and predictive entropy $H$.

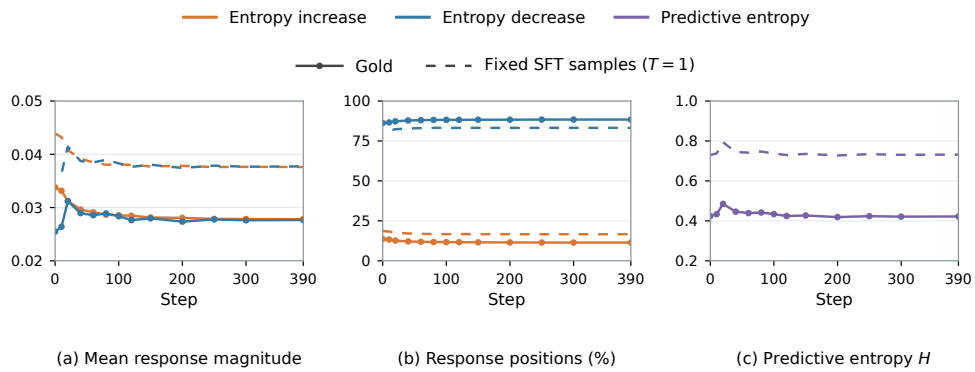



Figure 2: Gold answers (solid) and fixed SFT samples (dashed): response masses, sign fractions, and predictive entropy.

**SFT balances responses across training histories.** Initially, entropy-increasing Gold positions constitute 13.6% of tokens but contribute 57.0% of the absolute response, echoing the disproportionate role of minority tokens in RL (Wang et al., 2025). SFT brings $M_+$ and $M_-$ toward nonzero balance (Figure 2). This also occurs after switching from increase-only or decrease-only training to SFT at step 40 without resetting the optimizer (Figure 3a,b). Conversely, switching SFT to increase-only supervision raises $M_-$ while reducing $M_+$; decrease-only supervision reduces both (c,d). Thus the update rule reshapes the balance and the available entropy-decreasing response.

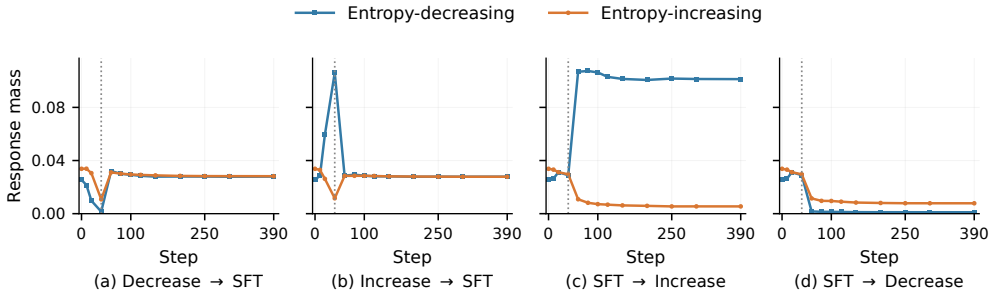



Figure 3: Gold response masses after switching supervision at step 40 (dotted line). Increase/Decrease restrict updates by response sign.

**Connecting balance to distribution matching.** SFT's distribution-matching objective (Gneiting & Raftery, 2007) gives a reference. At a fixed prefix, the zero-mean NLL descent direction (Mohamed et al., 2020) implies

$$\mathbb{E}_{v\sim p}[d(v)] = \nabla_z H(p)^\top \sum_v p_v(e_v - p) = 0, \tag{6}$$

with equal expected one-sided masses

$$R(p) = \sum_v p_v \left[d(v)\right]_+ = \sum_v p_v \left[-d(v)\right]_+ . \tag{7}$$

Gold observations instead sample targets from $q$, giving

$$\mathbb{E}_{v\sim q}[d(v)] = \nabla_z H(p)^\top (q - p). \tag{8}$$

Matching $p = q$ therefore yields zero expected Gold response. Probability, log-loss, and response diagnostics support close statistical matching on Gold prefixes (Appendix B). SFT's fixed generated answers also exhibit balance, but with larger masses and higher $H$ than Gold (Figure 2), linking fitting under demonstrated contexts to distinct generation behavior (Braverman et al., 2020; Cao et al., 2025). SFT's weaker downstream performance motivates examining how supervision shapes this relationship.

**Historically familiar answers.** Base-generated answers reflect pretrained support but are external to the evolving SFT model. Their masses initially balance; during SFT, $M_-$ rises while $M_+$ stays comparatively stable. For these historically familiar self-generated answers, entropy-decreasing mass dominates, opposite to Base's initial entropy-increasing dominance on Gold (Figure 4).

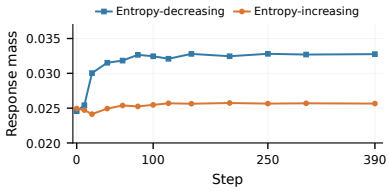



Figure 4: Response masses on fixed Base-generated answers during SFT.

**Weighting changes the effective supervision.** At each model state, nonnegative detached weights define target probabilities proportional to $wq$, with total weight setting each prefix's contribution. PriFT's frozen gate fixes selection; DFT and CAFT adapt it. Their entropy-decreasing masses decline to different levels (Figure 5): CAFT approaches zero, DFT's two sides decline and approach each other, and PriFT approaches balance. ASFT's total response, including reference anchoring, retains a gap.

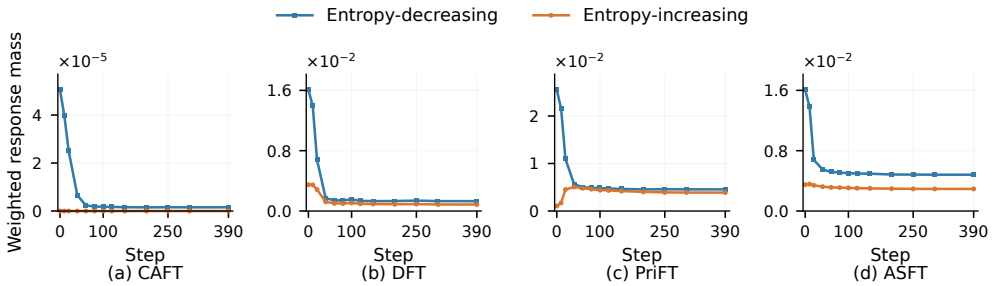



Figure 5: Training-weighted response masses. Original weight scales are retained; vertical ranges differ.

### 5.2 Allowing Entropy-Increasing Updates

The switching results motivate partial retention of entropy-increasing supervision. Keeping the decreasing branch at unit weight, we use detached current-model weights

$$w_i^{(\lambda)} = \mathbf{1}[d_i < 0] + \lambda \mathbf{1}[d_i \geq 0], \quad \lambda \in \{0, 0.03, 0.1, 0.3, 1\}, \tag{9}$$

recovering CAFT-Hard at $\lambda = 0$ and SFT at $\lambda = 1$. Increasing $\lambda$ raises the later entropy-decreasing mass despite its unchanged branch weight. Partial retention maintains $M_-^{(w)} > M_+^{(w)}$, whereas SFT approaches balance (Figure 6a). At $\lambda = 0.03$, Pass@16 gains 3.52 points over Hard while Avg@16 loses only 0.52; $\lambda = 0.1$ attains the highest Pass@16, 58.70%, with Avg@16 of 29.20%. Larger coefficients reduce accuracy without further coverage gains (b,c). Entropy-increase strength thus adjusts both the retained response level and the accuracy–coverage trade-off.

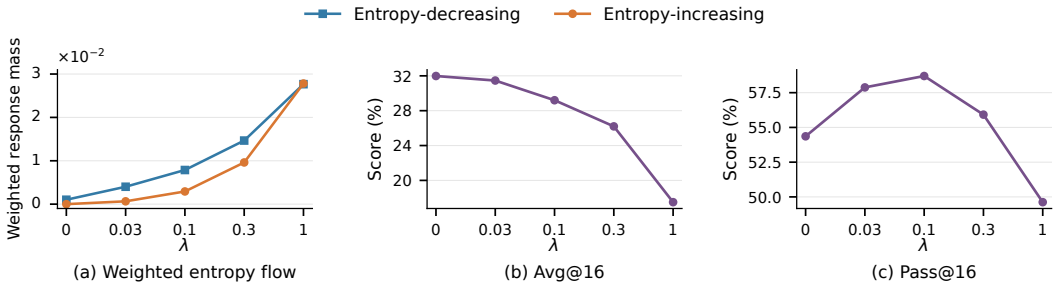



Figure 6: Final weighted response masses and performance at the five evaluated $\lambda$ settings.

### 5.3 Allocation across Questions

Prior work uses model-dependent signals to select or reweight training examples (Chang et al., 2017; Mindermann et al., 2022). We examine whether aggregating entropy responses can guide supervision across questions. For a complete answer $s$ with $n_s$ valid tokens, let $D_s^{\pm} = \sum_t [\pm d_{s,t}]_+$. Base responses vary substantially across the 1,024 Gold questions, with 163 (15.9%) satisfying $D_s^- > D_s^+$. The sum and token-normalized views in Figure 7a,d motivate two complementary allocation rules.

**Question selection.** We rank questions by their signed response $D_s^+ - D_s^-$ and discard the highest four within each group of 16, applying ordinary NLL to the retained tokens. Compared with ran-

domly discarding four questions, response-based selection improves final Avg@16 and Pass@16 by 1.32 and 4.42 points; it also exceeds full SFT by 0.54 and 1.44 points (Figure 7b,e).

**Question weighting.** We further multiply token weights by the accumulated entropy-decreasing support of their answer. With $a_{s,t} = 0.01\,[-d_{s,t}]_+$ and $C_s = \sum_t a_{s,t}$, the two detached weighting rules are

$$w_{s,t}^{\text{Hard+QW}} = C_s \mathbf{1}[d_{s,t} < 0], \qquad w_{s,t}^{\text{CAFT+QW}} = C_s a_{s,t}. \tag{10}$$

Both retain the entropy-decreasing token gate. Question weighting improves final Avg@16/Pass@16 by 2.36/4.48 points over CAFT-Hard and 0.60/1.20 points over CAFT (Figure 7c,f). These results show that reallocating supervision across questions can improve performance beyond token-level selection and weighting alone.

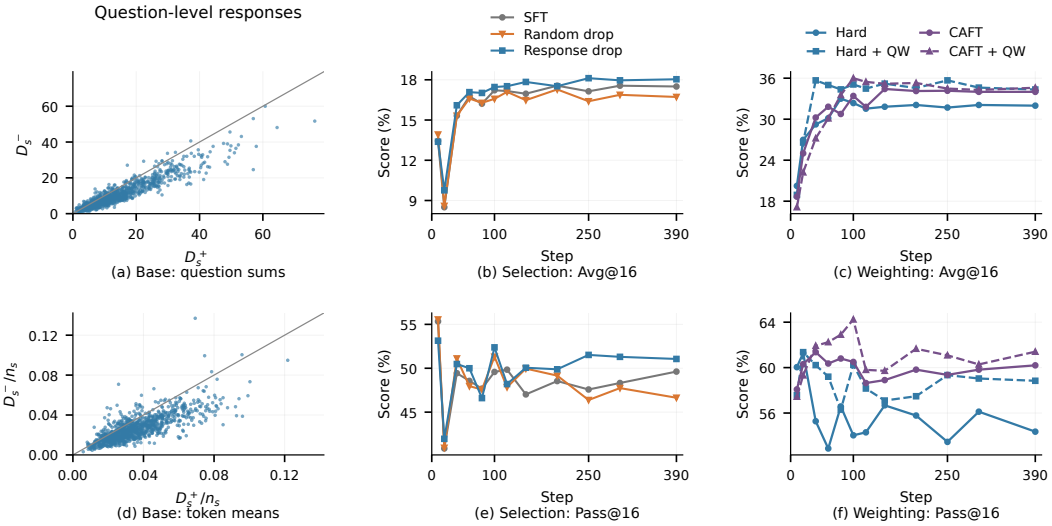



Figure 7: Question-level allocation. Left: Base response sums and token means; each point is one answer, and the diagonal marks zero net response. Middle: full SFT, random dropping, and response-based dropping. Right: CAFT-Hard (Hard) and CAFT, with and without question weighting (QW).

## 6 Conclusion

*[Draft note: Write after the main claims and supporting experiments are finalized.]*

### AI use statement

We used generative AI tools, including ChatGPT/Codex, to assist with developing and discussing conceptual frameworks and mathematical formulations, refining research hypotheses and experimental designs, implementing and debugging research code, interpreting experimental results, and translating and drafting manuscript text. These tools also assisted with literature discovery, organization of research notes, and document formatting. The authors retain responsibility for the final text, mathematical claims, experimental results, citations, and code.

*[Draft note: Author verification required: confirm the final scope of AI use, including any synthetic-data generation, dataset cleaning, qualitative analysis, or proof assistance; describe the human verification actually performed. Do not assert that all AI-assisted work has already been reviewed until that review is complete. Ensure the disclosure in the submission form agrees with the final manuscript.]*

### Ethics statement

This work studies how supervision changes language-model behavior. Improvements on mathematical reasoning benchmarks should not be interpreted as guarantees of correctness, safety, or reliability outside the evaluated settings. State-dependent supervision can also reinforce undesirable model preferences, so its effects should be assessed rather than inferred solely from confidence or entropy.

*[Draft note: Before submission, verify dataset and model licenses, relevant data provenance and privacy considerations, and any additional ethical issues introduced by the final experimental scope. Do not insert unverified claims about consent, institutional approval, or the absence of sensitive data.]*

### Reproducibility statement

*[Draft note: Finalize this paragraph after Appendices A–C are complete. Reference the exact sections containing model/data revisions, preprocessing and masks, normalization, optimizer and evaluation settings, mathematical assumptions, and per-seed results. Link code only if an anonymous artifact has actually been prepared and checked.]*

## References


Allen Institute for AI. Tulu-3 SFT Personas Code. Hugging Face dataset, 2024. URL https://huggingface.co/datasets/allenai/tulu-3-sft-personas-code.

Mark Braverman, Xinyi Chen, Sham Kakade, Karthik Narasimhan, Cyril Zhang, and Yi Zhang. Calibration, entropy rates, and memory in language models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1089–1099. PMLR, 2020. URL https://proceedings.mlr.press/v119/braverman20a.html.

Steven Cao, Gregory Valiant, and Percy Liang. On the entropy calibration of language models. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2511.11966.

Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. Active bias: Training more accurate neural networks by emphasizing high variance samples. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/2f37d10131f2a483a8dd005b3d14b0d9-Abstract.html.

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. URL https://arxiv.org/abs/2601.02151.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437. URL https://doi.org/10.1198/016214506000001437.

Zhezheng Hao, Hong Wang, Haoyang Liu, Jian Luo, Jiarui Yu, Hande Dong, Qiang Lin, Can Wang, and Jiawei Chen. Rethinking entropy interventions in RLVR: An entropy change perspective. *arXiv preprint arXiv:2510.10150*, 2025. URL https://arxiv.org/abs/2510.10150.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 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)*, 2024. URL https://aclanthology.org/2024.acl-long.211/.

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. URL https://arxiv.org/abs/2009.03300.

Hugging Face. OpenR1-Math-220k. Hugging Face dataset, 2025. URL https://huggingface.co/datasets/open-r1/OpenR1-Math-220k.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, Kai Dang, Yang Fan, Yichang Zhang, An Yang, Rui Men, Fei Huang, Bo Zheng, Yibo Miao, Shanghaoran Quan, Yunlong Feng, Xingzhang Ren, Xuancheng Ren, Jingren Zhou, and Junyang Lin. Qwen2.5-Coder technical report. *arXiv preprint arXiv:2409.12186*, 2024. URL https://arxiv.org/abs/2409.12186.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024. URL https://arxiv.org/abs/2403.07974.

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. *arXiv preprint arXiv:2009.13081*, 2020. URL https://arxiv.org/abs/2009.13081.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://arxiv.org/abs/2206.14858.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. NuminaMath. Hugging Face dataset, 2024. URL https://huggingface.co/datasets/AI-MO/NuminaMath-CoT.

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*, 2025. URL https://arxiv.org/abs/2408.16673.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023. URL https://arxiv.org/abs/2305.20050.

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Yelong Shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, and Weizhu Chen. Not all tokens are what you need for pretraining. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://papers.nips.cc/paper_files/paper/2024/hash/3322a9a72a1707de14badd5e552ff466-Abstract-Conference.html.

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. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2305.01210.

Mathematical Association of America. American mathematics competitions (AMC), 2023. URL https://maa.org/student-programs/amc/.

Mathematical Association of America. American invitational mathematics examination (AIME), 2024. URL https://maa.org/maa-invitational-competitions/.

Sören Mindermann, Jan M. Brauner, Muhammed T. Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N. Gomez, Adrien Morisot, Sebastian Farquhar, and Yarin Gal. Prioritized training on points that are learnable, worth learning, and not yet learnt. In

*Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 15630–15649. PMLR, 2022. URL https://proceedings.mlr.press/v162/mindermann22a.html.

Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. Monte carlo gradient estimation in machine learning. *Journal of Machine Learning Research*, 21(132):1–62, 2020. URL https://jmlr.org/papers/v21/19-346.html.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. MedMCQA: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pp. 248–260, 2022. URL https://proceedings.mlr.press/v174/pal22a.html.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/e6385d39ec9394f2f3a354d9d2b88eec-Abstract.html.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

Noam Razin, Sadhika Malladi, Adithya Bhaskar, Danqi Chen, Sanjeev Arora, and Boris Hanin. Unintentional unalignment: Likelihood displacement in direct preference optimization. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.08847.

Yi Ren and Danica J. Sutherland. Learning dynamics of LLM finetuning. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/afe1aa79e5eea7955f553c61a307273e-Abstract-Conference.html.

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*, pp. 9275–9293. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.746. URL https://aclanthology.org/2020.emnlp-main.746/.

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. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BJlxm30cKm.

Ke Wang, Shuangqi Li, Mathieu Salzmann, and Pascal Frossard. PriFT: Prior-support guided supervised fine-tuning. *arXiv preprint arXiv:2606.09396*, 2026. URL https://arxiv.org/abs/2606.09396.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for LLM reasoning. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://arxiv.org/abs/2506.01939.

David Widmann, Fredrik Lindsten, and Dave Zachariah. Calibration tests in multi-class classification: A unifying framework. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/1c336b8080f82bcc2cd2499b4c57261d-Abstract.html.

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

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 54104–54132. PMLR, 2024. URL https://proceedings.mlr.press/v235/xia24c.html.

Huimin Xu, Shuai Zhao, Xiaobao Wu, and Anh Tuan Luu. Understanding and preventing entropy collapse in RLVR with on-policy entropy flow optimization. *arXiv preprint arXiv:2605.11491*, 2026. URL https://arxiv.org/abs/2605.11491.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-Math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024. URL https://arxiv.org/abs/2409.12122.

Jiazheng Zhang, Ziche Fu, Junrui Shen, Yunbin Zhao, Yunke Zhang, Zhiheng Xi, Long Ma, Chenxin An, Zhihao Zhang, Shichun Liu, Dingwei Zhu, Shihan Dou, Shaofan Liu, Han Li, Wiggin Zhou, Aiden Adams, Tao Gui, Fei Huang, Qi Zhang, and Xuanjing Huang. Entropy polarity in reinforcement fine-tuning: Direction, asymmetry, and control. *arXiv preprint arXiv:2605.11775*, 2026a. URL https://arxiv.org/abs/2605.11775.

Miaosen Zhang, Yishan Liu, Shuxia Lin, Xu Yang, Qi Dai, Chong Luo, Weihao Jiang, Peng Hou, Anxiang Zeng, Xin Geng, and Baining Guo. Towards on-policy SFT: Distribution discriminant theory and its applications in LLM training. *arXiv preprint arXiv:2602.12222*, 2026b. URL https://arxiv.org/abs/2602.12222.

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

# A Experimental Details

**Shared mathematical-reasoning configuration.** For the Qwen2.5-Math-1.5B comparison, we use full-parameter fine-tuning with AdamW, a learning rate of $5 \times 10^{-5}$, $(\beta_1, \beta_2) = (0.9, 0.95)$, weight decay 0.01, and gradient clipping at 1.0. The learning rate follows a cosine schedule over 390 updates with a warmup ratio of 0.1. The global batch size is 256 and the per-device micro-batch size is 4. The maximum sequence length is 2,048, with right truncation. All compared runs use BF16 model computation and Liger-enabled training.

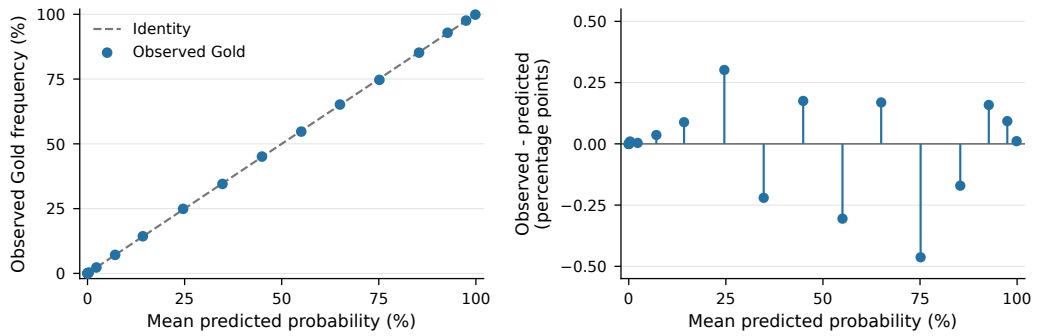
**Loss normalization and CAFT scale.** We sum weighted token losses across all micro-batches and devices contributing to an optimizer step and divide by the total number of supervised response tokens, including tokens assigned zero weight. Prompt and padding tokens are excluded. The denominator is not the sum of weights, nor do we average separately normalized sequence losses. For the initial CAFT comparison, the detached weight is $0.01[-d_t]_+$; entropy statistics for this weighting are computed in FP32. Other method-specific settings will be documented alongside their implementation details.

*[Draft note: Data and model revisions; preprocessing; effective batches; local-valid versus global normalization; optimizer, numerical precision, clipping, checkpoint selection, and evaluation.]*

# B Derivations and Additional Analyses

## B.1 Probability and entropy matching on Gold prefixes

**Statistical matching on Gold prefixes.** We compare model probabilities with empirical target frequencies using probability-bin calibration (Widmann et al., 2019). For every Gold prefix, we include all vocabulary candidates and group the resulting prefix–candidate events into 18 probability bins. Within each bin, we compare the mean predicted probability with the fraction of events whose candidate is the observed Gold token. Figure 8 shows close agreement across the probability range: the maximum absolute bin-wise gap is 0.463 percentage points. For example, in the $[0.9, 0.95)$ bin, the predicted probability is 92.71% and the observed Gold frequency is 92.87%. These results show that SFT's near-balanced entropy flow is accompanied by close probability–frequency matching on the Gold diagnostic set.




(a) Probability calibration (b) Bin-wise residuals

Figure 8: Probability-bin calibration of the final SFT model on Gold prefixes. Each point represents one of 18 bins containing full-vocabulary prefix–candidate events. Both panels use the bin's mean predicted probability as the horizontal coordinate. Left: empirical Gold frequency against prediction, with the identity line. Right: observed minus predicted frequency, in percentage points, shown on an expanded vertical scale to make small deviations visible.

**Mean log loss and entropy.** On the same Gold diagnostic set, the final SFT model has a mean NLL of $0.4152$ nats per token and a mean predictive entropy of $0.4218$ nats, giving an NLL–entropy gap of approximately $-0.0066$ nats per token.

### B.2 Entropy-response matching on Gold prefixes

For each Gold prefix $i$, the model-expected joint histogram assigns mass $p_{i,v}/N$ to the cell containing $(R(p_i), d_i(v))$ for every vocabulary candidate $v$. The observed histogram assigns mass $1/N$ to the cell containing $(R(p_i), d_i(y_i))$. Both histograms use the same $100 \times 1{,}200$ grid and each has total mass one. Denoting their cell masses by $E_b$ and $O_b$, respectively, the binned total variation distance is $\frac{1}{2}\sum_b |O_b - E_b| = 0.009257$. All histogram statistics use the original, unsmoothed data.

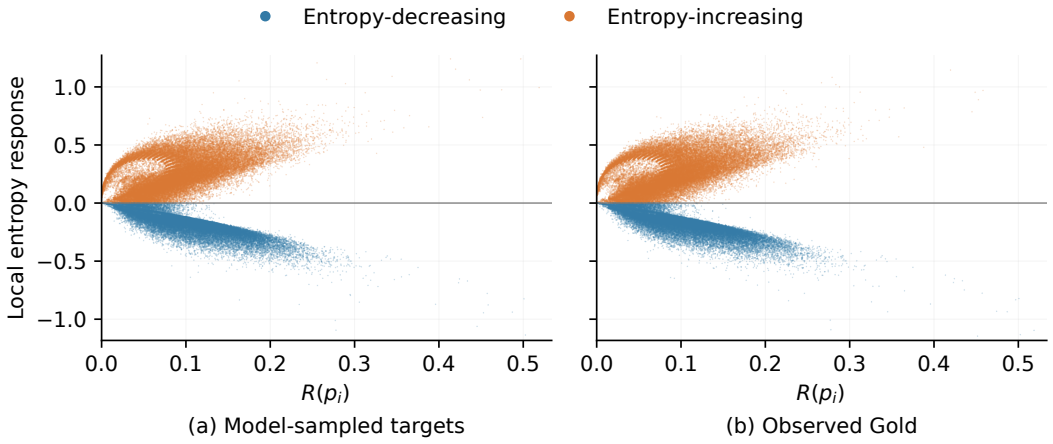



Figure 9: Joint response structure for the final SFT model on the same 436,443 Gold prefixes. Each panel plots $R(p_i)$ against the response to (a) one target sampled from $p_i$ or (b) the observed Gold target. Sampled targets do not alter subsequent prefixes. Orange and blue denote entropy-increasing and entropy-decreasing responses. All positions are shown with shared axes, without smoothing, jitter, or subsampling.

The sampled-target panel uses one full-categorical draw per position at temperature one, with random seed 42. The quantity $R(p_i)$ is computed exactly over the vocabulary in both panels. The total variation distance reported above compares Gold with the exact model-expected histogram; the sampled targets provide a pointwise visualization of that reference distribution.

*[Draft note: Derivations with explicit assumptions and supporting diagnostics. Preserve distinctions between finite probes, analytic responses, and parameter-space effects.]*

### B.3 Complete entropy-flow comparisons across methods

We report the seven training trajectories used in the main comparison on the same 1,024 Gold questions and 436,443 response positions. Each method includes its Base starting point and all 12 evaluated checkpoints. Full-position statistics use the common Gold-NLL direction, while training-rule statistics incorporate each method's detached weights or gate. Every mass is normalized by all response positions, including positions assigned zero weight. Predictive entropy uses natural logarithms, and exact zero responses are recorded separately.

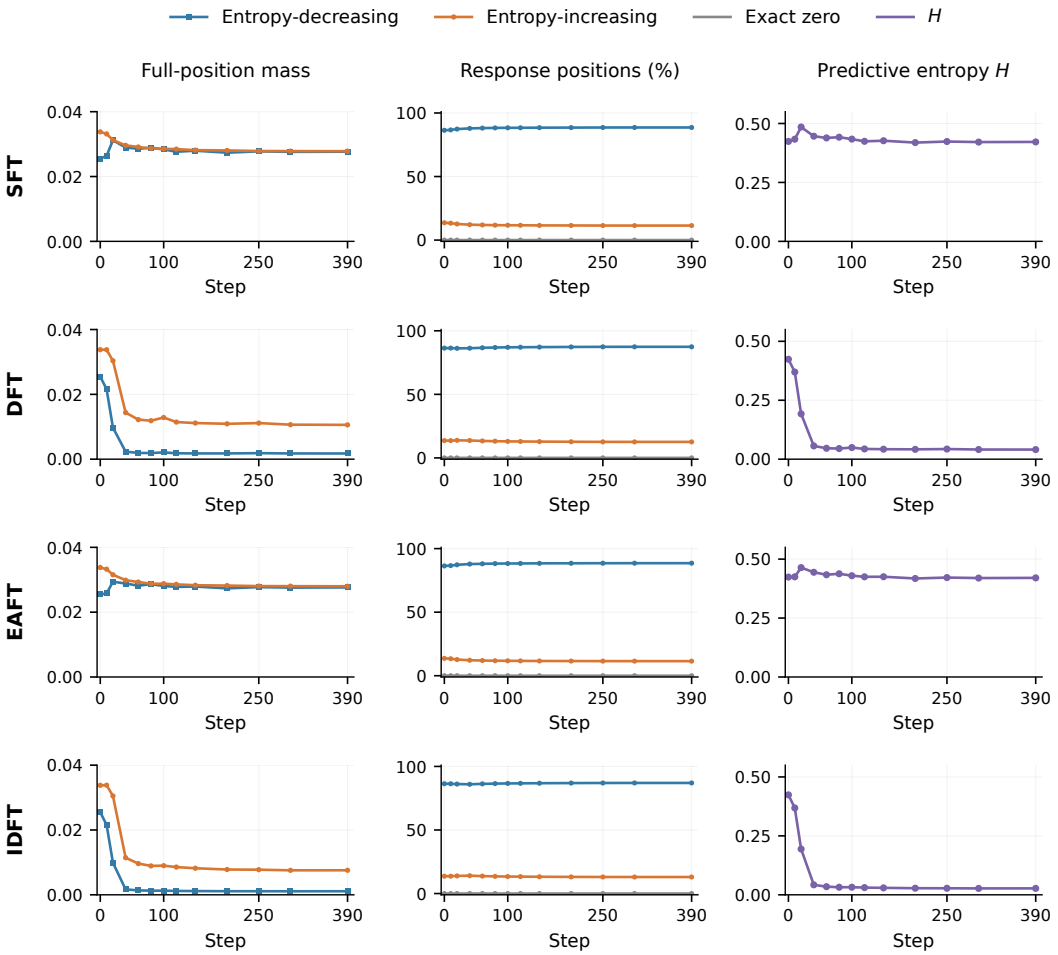



Figure 10: Full-position entropy flow for SFT, DFT, EAFT, and IDFT. Columns show the two-sided masses $M_+$ and $M_-$, response-sign fractions, and predictive entropy $H$. Axis ranges are shared across methods within each column.

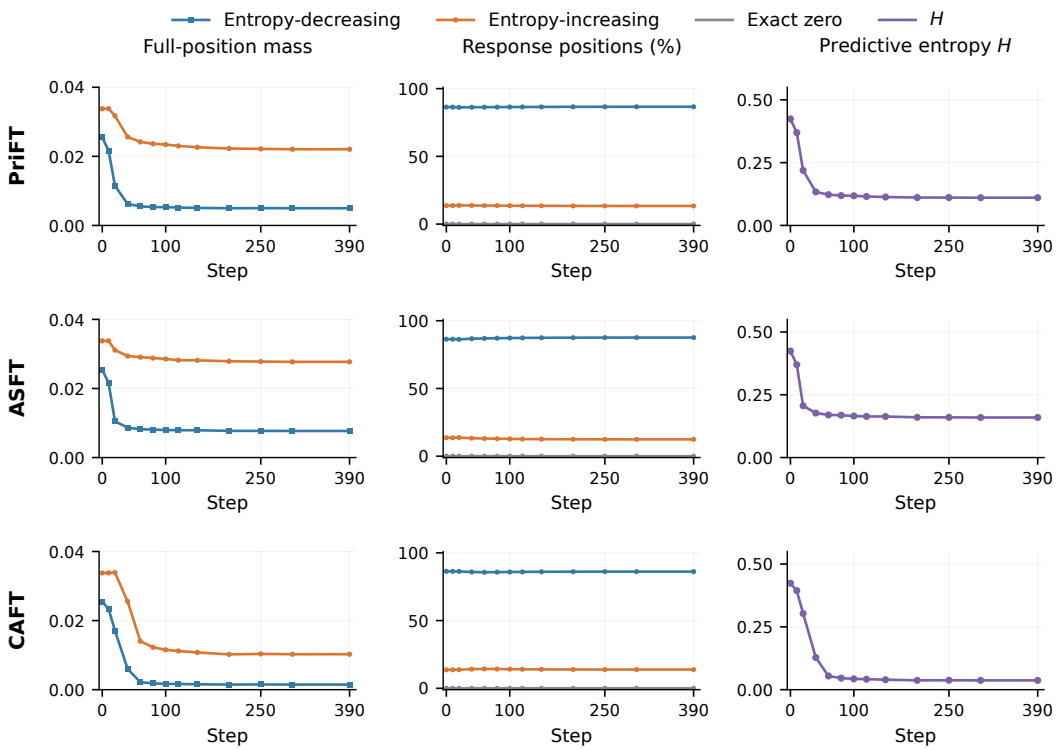



Figure 11: Full-position entropy flow for PriFT, ASFT, and CAFT, using the same statistics and axis ranges as Figure 10.

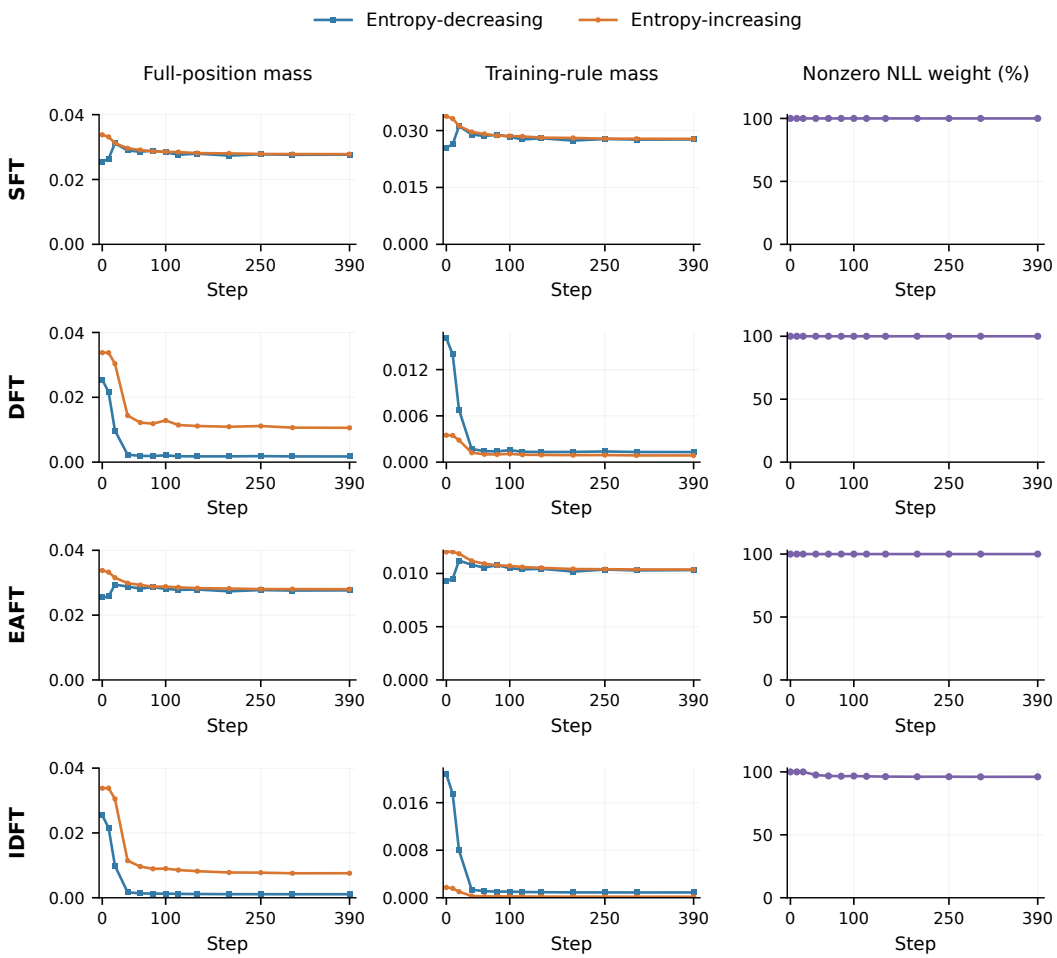



Figure 12: Full-position and training-rule response masses for SFT, DFT, EAFT, and IDFT. The last column reports the fraction of positions with nonzero effective NLL weight, including the effect of IDFT's clamp. Actual weight scales are retained; the middle-column vertical ranges vary by method.

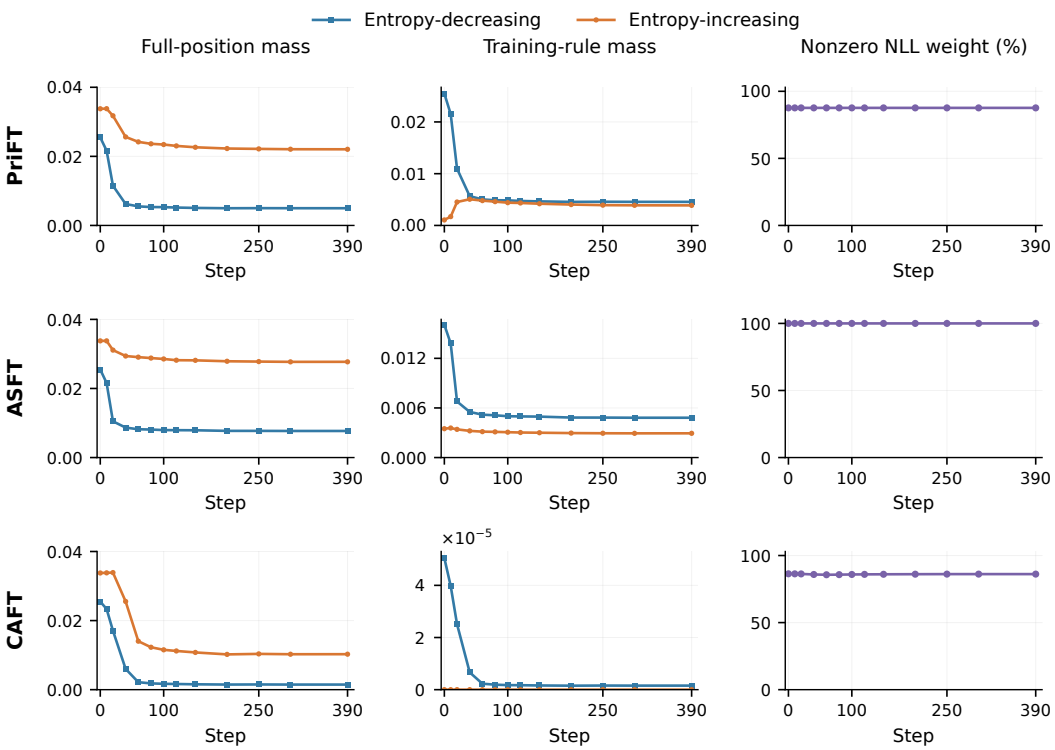



Figure 13: Full-position and training-rule response masses for PriFT, ASFT, and CAFT. CAFT retains its training scale of 0.01. ASFT's middle column includes its reference-model KL contribution; its nonzero-weight fraction refers only to the NLL branch.

**ASFT response decomposition.** For ASFT, the signed response at each position is $p_{i,y_i} d_i + 0.05 \nabla_{z_i} H(p_i)^\top (p_i^{\text{base}} - p_i)$. We add the NLL and KL contributions at each position before separating the total into positive and negative masses. Figure 14 shows the components and their total on a common vertical scale.

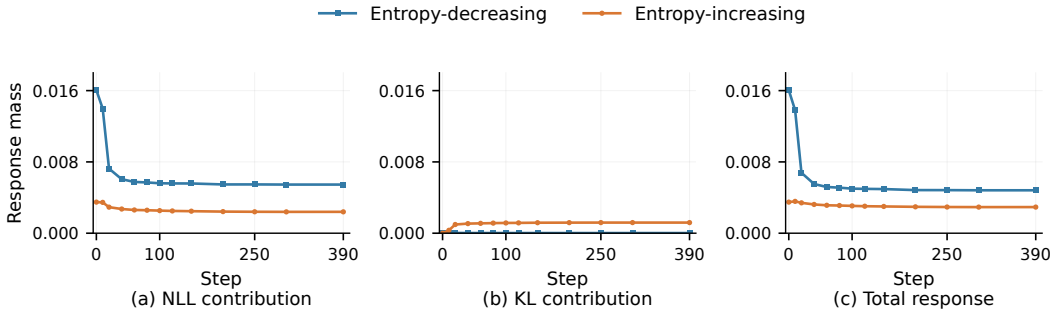



Figure 14: ASFT's NLL, KL, and combined local entropy responses.

### B.4 Shared-Parameter Effects and Reference States

*[Draft note: Select only diagnostics that explain the allocation comparisons. Keep local entropy response, parameter-gradient interaction, and actual Adam updates distinct.]*

### B.5 Stable Stages and Continued Fine-Tuning

*[Draft note: Discuss the operational meaning of dynamic balance and the SFT restart controls. Identify old versus continuous-two-epoch B and the fixed evaluation prefixes. Do not claim an isolated optimizer-reset effect.]*

### B.6 Discussion and Limitations

*[Draft note: Consolidate scope and evidence limitations here: directional rather than complete mismatch, shared parameters, optimization and normalization, finite sampling, and generalization beyond the studied settings. This heading is our organizational choice, not a prescribed ICLR section.]*

## C Additional Results

Table 4: Greedy accuracy of Qwen2.5-Math-1.5B at step 390 (%). Average is the unweighted mean over five benchmarks. The best value in each column is bold, including ties.

| Method | MATH500 | Minerva | OlympiadBench | AIME24 | AMC23 | Average |
|---|---|---|---|---|---|---|
| SFT | 61.4 | **26.5** | 28.9 | **10.0** | 45.0 | 34.36 |
| DFT | 65.6 | 23.2 | 28.7 | 3.3 | 37.5 | 31.66 |
| EAFT | 62.2 | 20.2 | 28.0 | 6.7 | 45.0 | 32.42 |
| IDFT | 67.2 | 21.3 | 27.6 | 6.7 | 32.5 | 31.06 |
| PriFT | **68.8** | 18.4 | 32.0 | **10.0** | **50.0** | 35.84 |
| ASFT | 64.2 | 14.3 | 29.6 | **10.0** | 37.5 | 31.12 |
| CAFT | 67.4 | 26.1 | **33.9** | **10.0** | 42.5 | **35.98** |

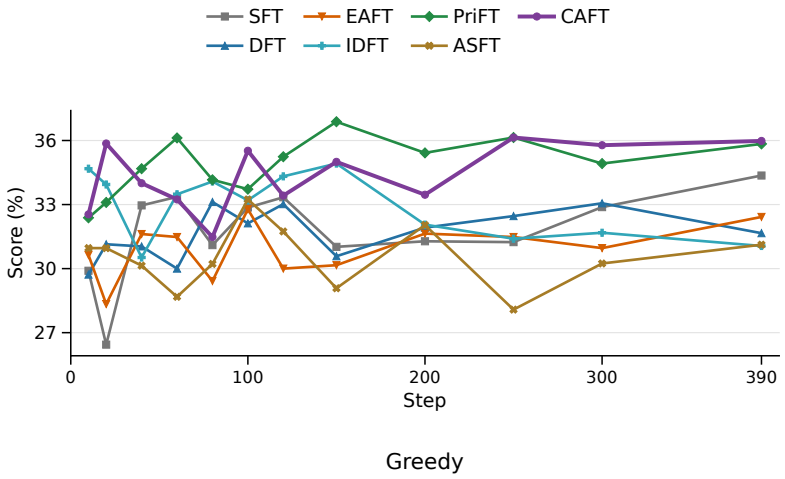



Figure 15: Greedy accuracy throughout fine-tuning of Qwen2.5-Math-1.5B. Scores are unweighted macro-averages over the five mathematical reasoning benchmarks (%), evaluated at the same 12 checkpoints as the main-text sampling metrics. Method colors and markers match Figure 1.

*[Draft note: Per-seed and per-checkpoint metrics, supplementary trajectories, and additional metrics where relevant.]*