FAMIFT: FAMILIARITY-ADAPTIVE SUPERVISED FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Supervised fine-tuning (SFT) typically applies the same training objective across examples, while recent probability-based methods such as Dynamic Fine-Tuning (DFT) adapt token supervision using the model's online confidence. However, the appropriate strength of such token weighting may itself depend on how well the pretrained model already fits each training example. We first conduct a controlled study and find a consistent familiarity-dependent pattern: relatively familiar examples benefit from sharper probability-based weighting, whereas relatively unfamiliar examples benefit from flatter weighting. Motivated by this observation, we propose Familiarity-Adaptive Fine-Tuning (FamiFT), a simple extension of probability-weighted SFT that assigns each example a sample-specific exponent. FamiFT measures response familiarity once using the frozen initial model, centers it by the dataset median, and uses the resulting score to modulate online token probabilities throughout fine-tuning. Across mathematical reasoning and code generation, FamiFT consistently improves average performance over standard SFT and competitive token-weighting baselines, providing a simple way to incorporate pretrained-model familiarity into supervised fine-tuning.

1 INTRODUCTION

Supervised fine-tuning (SFT) plays an important role in large language model post-training, adapting pretrained models to user instructions and downstream tasks through dense supervision from reference responses (Chung et al., 2024; Ouyang et al., 2022). Standard SFT optimizes the negative log-likelihood of reference tokens and treats all target tokens uniformly within the training objective. Recent work has questioned whether such uniform weighting is always desirable. Dynamic Fine-Tuning (DFT) addresses this issue by weighting each target token according to its probability under the current model (Wu et al., 2026). As a result, low-probability targets receive weaker supervision, creating non-uniform token weights that reflect the model's current confidence. Building on this idea, subsequent methods further refine token supervision using model-derived signals, such as probability thresholding (Liu et al., 2026), token-level predictive entropy (Diao et al., 2026), and scores derived from a frozen pretrained reference model (Wang et al., 2026).

Despite these advances, DFT determines token weights primarily from local token-level probabilities, without explicitly incorporating response-level information about the training sample, such as how predictable the response is under the model. Recent extensions have begun to incorporate broader information into dynamic weighting. Lin et al. (2026) adjusts probability-based token weights using a temperature derived from batch-level response statistics, but the same transformation is applied to all examples in the batch. Zhou et al. (2026) instead incorporates response-level compatibility to rescale the contribution of individual examples, but this sample-level scaling does not alter the relative weights among tokens within a response. These limitations raise a natural question: can we design a sample-adaptive token-weighting rule that both distinguishes among training examples and reshapes the relative weights of tokens within each sample?

To investigate this question, we conduct a controlled analysis in Section 3 on how the preferred token-weighting strength varies across training samples. We first use the response negative log-likelihood under the initial model to divide training samples into relatively familiar and unfamiliar groups. Since the exponent naturally controls the relative differences among token weights, we introduce it into the probability-based weighting rule and vary it across the two groups to examine

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

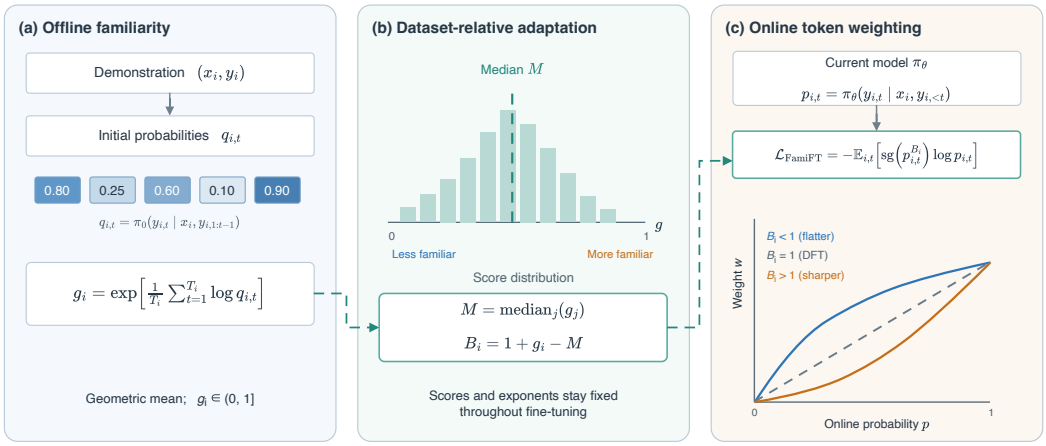

Figure 1: Overview of FamiFT. (a) The frozen initial model scores each demonstration using the geometric mean of response-token probabilities. (b) Dataset median familiarity sets a fixed exponent $B_i = 1 + g_i - M$. (c) The current model supplies online probabilities to the FamiFTobjective: lower familiarity produces flatter weighting, while higher familiarity produces sharper weighting. Token probabilities, the score distribution, and the curves are schematic; the curves use illustrative exponents 0.75, 1, and 1.25.

whether familiar and unfamiliar samples prefer different weighting strengths. We observe a clear asymmetric pattern: familiar samples benefit from sharper weighting that enlarges the relative differences among token weights, whereas unfamiliar samples benefit from flatter weighting that provides more balanced supervision across tokens. These findings show that the exponent effectively controls how strongly token weights are differentiated within each sample and suggest that its preferred value varies with sample familiarity.

Motivated by these observations, we propose Familiarity-Adaptive Fine-Tuning (FamiFT), which adapts the relative weight differences among tokens within each sample using a sample-specific exponent. We first estimate sample familiarity using the initial model, measuring how predictable the reference response is by the geometric mean of the probabilities assigned to its target tokens. We then compare each familiarity score with the dataset median and use their deviation to determine a sample-specific exponent. Less familiar samples receive a smaller exponent, producing flatter token weights and more balanced supervision across the response. In contrast, more familiar samples receive a larger exponent, amplifying the relative differences among token weights. The familiarity scores and their corresponding exponents remain fixed throughout training to preserve stable sample-level information from the initial model, while the online token probabilities evolve with the current model, allowing FamiFTto combine initial familiarity with dynamic token-level signals during fine-tuning.

Our contributions are threefold:

- We show that the exponent applied to token probabilities plays an important role in controlling token-weight differences and its preferred value varies with sample familiarity.
- We propose Familiarity-Adaptive Fine-Tuning (FamiFT), which uses sample familiarity under the initial model to determine a sample-specific exponent, thereby adapting the relative weight differences among tokens within each sample.
- We evaluate FamiFTon mathematical reasoning and code generation tasks, where it improves average performance over standard SFT and existing dynamic token-weighting baselines. Extensive ablations further validate the roles of sample familiarity and sample-specific weighting.

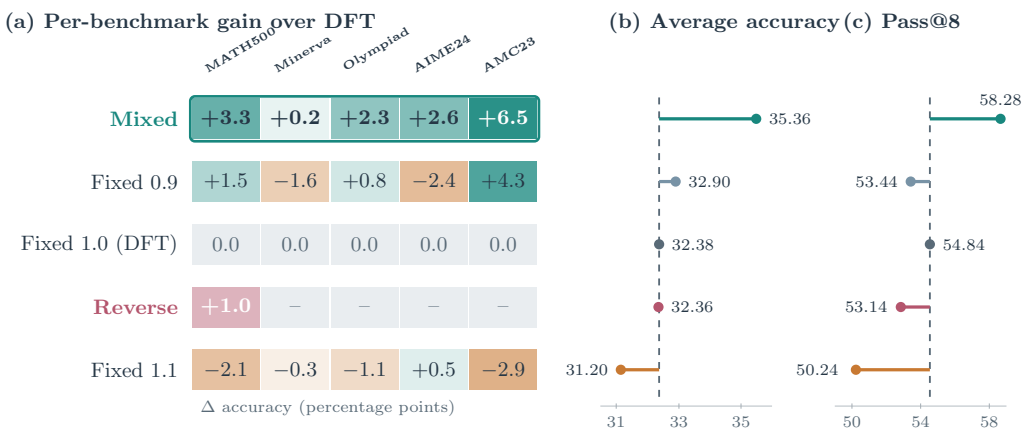

Figure 2: Familiarity-split exponent experiment on Qwen2.5-Math-7B. We form a 2K-example subset from the 1,000 most familiar and 1,000 least familiar training examples, ranked by initial-model response NLL (lowest NLL is most familiar). Mixed uses $B = 1.1$ for familiar and $B = 0.9$ for unfamiliar examples; fixed controls use the same exponent for all examples. (a) Per-benchmark accuracy differences from DFT. (b,c) Aggregate accuracy and sampled coverage; dashed lines mark DFT.

2 Related Work

2.1 Post-Training of Large Language Models

SFT adapts pretrained language models by maximizing the likelihood of reference responses. Instruction tuning improves instruction following and transfer to unseen tasks (Chung et al., 2024). RL instead optimizes reward feedback on model outputs. RLHF uses human preferences to train a reward model (Ouyang et al., 2022), while reasoning-focused RL can use verifiable outcomes to improve problem solving (Guo et al., 2025). These approaches are often combined: SFT establishes useful response behavior before RL further refines it.

Their generalization behavior depends on the training setting. Chu et al. (2025) find that RL generalizes better than SFT to unseen rule and visual variants in their studied tasks. Other work shows that SFT's loss of general capabilities can be reduced through learning-rate choices (Lin et al., 2026), and that its objective should reflect the model's existing capabilities (Li et al., 2025). These findings motivate improving supervised adaptation through its training objective. FamiFT follows this direction by adapting token weights within SFT.

2.2 Token-Level Training Objectives

DFT analyzes SFT through an RL formulation and rectifies its implicit reward structure through token-level probability weighting (Wu et al., 2026). Its objective multiplies each target-token NLL by the detached online probability, reducing the contribution of low-probability targets. ProFit uses a hard probability threshold to select tokens (Liu et al., 2026), while TALR adjusts probability-based weights through temperature scaling and a weight floor (Lin et al., 2026).

Other methods change the weighting signal or constrain distributional drift. EAFT uses predictive entropy to reduce updates at positions where the model is confident but disagrees with the target (Diao et al., 2026). ASFT adds KL regularization to DFT's reweighting to anchor training to the initial model (Zhu et al., 2026). PriFT derives token weights from a frozen pretrained reference, using target-token probability or cumulative probability mass (Wang et al., 2026). FamiFT also uses an initial-model signal, but aggregates it into response familiarity to control the exponent on online token probabilities.

Probability-based objectives also connect to generalized cross-entropy (Zhang & Sabuncu, 2018). In LLM fine-tuning, Li et al. (2025) show that suppressing low-probability tokens can help when

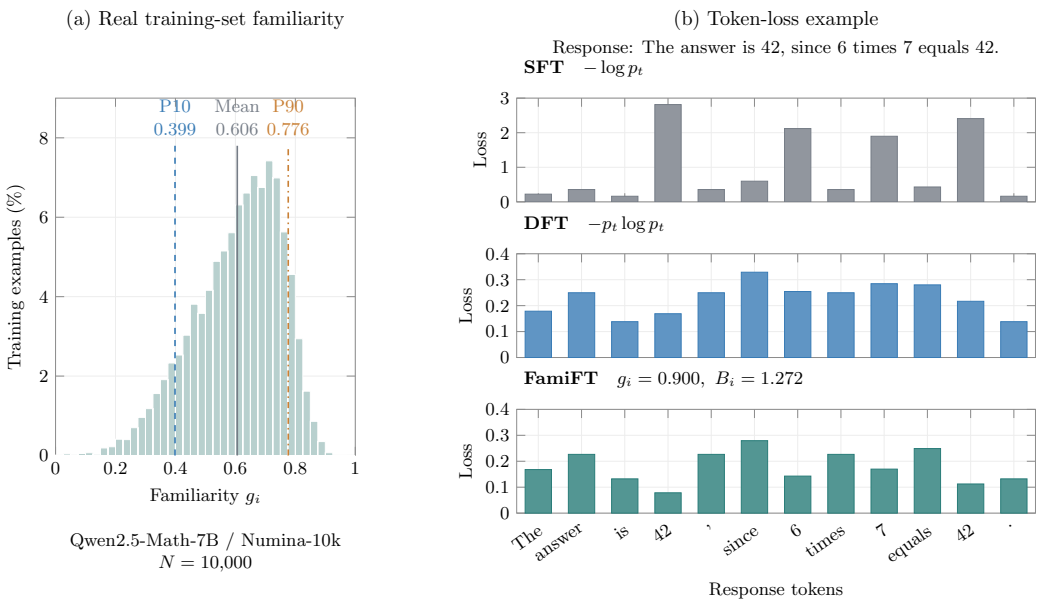

Figure 3: Familiarity and token-level loss contributions. Left: the empirical g_i distribution for Qwen2.5-Math-7B on Numina-10k, with P10, mean, and P90 marked; the method uses the median as its center. Right: token-level loss contributions under SFT, DFT, and FamiFT for an illustrative high-familiarity response. All three panels share a vertical scale; the token probabilities and losses are listed in Appendix D.2.

task-relevant priors are strong, whereas NLL can perform better when they are weak. FamiFT uses response predictability to adapt the weighting strength among examples within a single training dataset.

Sample weighting offers a related approach. FLOW uses initial-model losses to upweight easy examples and mitigate forgetting (Sanyal et al., 2025). CADFT scales each example's DFT loss using online response NLL normalized by batch statistics, and also explores rewriting incompatible responses (Zhou et al., 2026). A sample multiplier preserves relative token weights within a response. FamiFT instead uses familiarity relative to the dataset median to adjust their ratios through a sample-specific exponent. This also changes total sample weight; our matched-weight ablations examine both effects.

3 Token-weighting Strength and Sample Familiarity

In this section, we investigate whether different token-weighting strengths are more effective for training samples with different levels of familiarity. We first use the response negative log-likelihood under the initial model as a measure of sample familiarity and select the 1,000 most familiar and 1,000 least familiar training samples. We then introduce an exponent into the DFT objective to control the strength of probability-based token weighting, yielding the following generalized objective:

$$\mathcal{L} = -\sum_i \sum_t \mathrm{sg}(p_{i,t}^{B}) \log p_{i,t}. \tag{1}$$

The exponent efficiently controls the relative differences among token weights within each response. A larger exponent sharpens the weighting by increasing the relative gap between high- and low-probability tokens, whereas a smaller exponent produces flatter weights and more balanced supervision across tokens. We consider five settings: standard DFT with $B \equiv 1$, two globally fixed exponents with $B \equiv 0.9$ and $B \equiv 1.1$, a mixed setting that assigns $B = 1.1$ to familiar samples and $B = 0.9$ to unfamiliar samples, and a reversed mixed setting that assigns $B = 0.9$ to familiar samples and $B = 1.1$ to unfamiliar samples.

Figure 2 shows a clear familiarity-dependent pattern. The mixed setting with $B = 1.1$ for familiar samples and $B = 0.9$ for unfamiliar samples achieves the best performance, reaching 35.36 Avg@8 and 58.28 Pass@8. In comparison, the best globally fixed exponent reaches 32.90 Avg@8, while standard DFT achieves 32.38. Importantly, reversing the assignment, with $B = 0.9$ for familiar samples and $B = 1.1$ for unfamiliar samples, reduces Avg@8 to 32.36. These results indicate that the direction of the assignment matters: familiar samples benefit from a larger exponent, whereas unfamiliar samples benefit from a smaller one.

This pattern can be understood from both the sample level and the token level. At the sample level, since token probabilities lie in $(0, 1)$, a larger exponent decreases the overall token weights of familiar samples, while a smaller exponent increases those of unfamiliar samples, shifting relatively more training emphasis toward unfamiliar examples. Within each sample, the exponent also controls the relative differences among token weights. A larger exponent sharpens the weighting, increasing the relative preference for high-probability tokens in familiar samples. In contrast, a smaller exponent flattens the weighting for unfamiliar samples, providing more balanced supervision across their tokens. These observations suggest that samples with different familiarity levels may require different weighting strengths, motivating a sample-adaptive token-weighting rule.

4 Familiarity-Adaptive Fine-Tuning

4.1 Setup

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ be the training dataset, where x_i denotes the prompt and $y_i = (y_{i,1}, y_{i,2}, \ldots, y_{i,T_i})$ denotes the reference response. Standard SFT assigns equal weight to all response tokens and optimizes the objective

$$\mathcal{L}_{\mathrm{SFT}}(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\sum_{t=1}^{T} -\log \pi_\theta(y_t \mid x, y_{<t}) \right]. \tag{2}$$

DFT weights each token loss by its current target-token probability and optimizes the objective

$$\mathcal{L}_{\mathrm{DFT}}(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\sum_{t=1}^{T} -\mathrm{sg}(\pi_\theta(y_t \mid x, y_{<t})) \log \pi_\theta(y_t \mid x, y_{<t}) \right], \tag{3}$$

where $\mathrm{sg}(\cdot)$ denotes the stop-gradient operator.

Before training, we measure the familiarity of each training sample using the frozen initial model π_0. Specifically, for each $(x_i, y_i) \in \mathcal{D}$, we define

$$g_i = \exp\left(\frac{1}{T_i} \sum_{t=1}^{T_i} \log \pi_0(y_{i,t} \mid x_i, y_{i,<t}) \right), \tag{4}$$

where g_i is the geometric mean of the target-token probabilities assigned by the initial model to the reference response. Since $g_i \in (0, 1]$, it provides a normalized measure of sample familiarity, with a larger value indicating that the initial model is more familiar with the reference response. We then center each familiarity score by subtracting the median familiarity over the training dataset:

$$M = \mathrm{median}_{i\in\{1,\ldots,N\}}(g_i), \qquad \delta_i = g_i - M. \tag{5}$$

The centered score δ_i measures each sample's familiarity relative to the dataset median, with $\delta_i < 0$ indicating relatively lower familiarity and $\delta_i > 0$ indicating relatively higher familiarity. Both g_i and M are computed before fine-tuning using the initial model and remain fixed throughout training.

4.2 Familiarity-adaptive token weighting

Motivated by the findings in Section 3, we propose Familiarity-Adaptive Fine-Tuning (FamiFT), which uses the centered familiarity score of each sample to determine a sample-specific exponent for probability-based token weighting. The resulting objective is

$$\mathcal{L}_{\mathrm{FamiFT}}(\theta) = \mathbb{E}_{(x_i,y_i)\sim\mathcal{D}} \left[\sum_{t=1}^{T_i} -\mathrm{sg}\Big(\pi_\theta^{\,1+g_i-M}(y_{i,t} \mid x, y_{i,<t}) \Big) \log \pi_\theta(y_{i,t} \mid x, y_{i,<t}) \right]. \tag{6}$$

Table 1: Mathematical reasoning accuracy (Avg@16, %). Avg. is the mean over five benchmarks; Δ is its change from Base in percentage points. **Bold** and underlined scores mark the best and second-best distinct values within each model, including ties. Additional Pass@16 results are in Appendix C.

Model	Method	MATH500	Minerva	Olympiad	AIME24	AMC23	Avg.	Δ
Qwen2.5 Math-7B Avg@16	Base	41.20	13.40	17.70	6.20	30.30	21.76	—
	SFT	48.90	15.90	15.50	3.80	22.50	21.32	-0.44
	DFT	69.10	28.90	33.00	**11.20**	41.60	36.76	+15.00
	ProFit	67.60	25.30	31.80	6.20	36.90	33.56	+11.80
	TALR	68.50	27.60	31.80	5.80	42.20	35.18	+13.42
	ASFT	69.20	27.20	33.00	9.60	41.20	36.04	+14.28
	FamiFT	**70.60**	**29.50**	**34.40**	7.10	**44.40**	**37.20**	**+15.44**
Qwen3-8B Base Avg@16	Base	57.60	20.80	27.30	6.70	35.30	29.54	—
	SFT	49.20	17.20	16.90	3.30	23.80	22.08	-7.46
	DFT	**69.90**	24.80	**32.20**	7.10	38.80	34.56	+5.02
	ProFit	66.40	26.80	31.00	6.70	38.80	33.94	+4.40
	TALR	69.10	**28.80**	31.50	6.70	39.10	35.04	+5.50
	ASFT	66.70	26.90	31.50	8.80	36.20	34.02	+4.48
	FamiFT	**69.90**	27.80	31.50	**10.00**	**40.90**	**36.02**	**+6.48**

When $g_i = M$, we have $B_i = 1$, and the weighting rule of FamiFT reduces to the DFT. For relatively unfamiliar samples, $g_i < M$ and hence $B_i < 1$, which increases token weights and reduces the relative differences between high- and low-probability tokens, resulting in flatter weighting. For relatively familiar samples, $g_i > M$ and hence $B_i > 1$, which decreases token weights while amplifying their relative differences, resulting in sharper weighting. In this way, FamiFT adapts both the overall supervision strength of each sample and the relative weighting among its tokens according to sample familiarity.

Figure 1 summarizes the overall workflow of FamiFT and Figure 3 provides an illustrative example of how familiarity affects token weighting.

5 Experiments

5.1 Experimental setup

Mathematical reasoning. We fine-tune Qwen2.5-Math-7B (Yang et al., 2024) and Qwen3-8B-Base (Yang et al., 2025) for one epoch on Numina-10k, a fixed 10,000-example subset of NuminaMath-CoT (Li et al., 2024), with a learning rate of 5×10^{-5}. We evaluate on MATH500 (Hendrycks et al., 2021), Minerva Math (Lewkowycz et al., 2022), OlympiadBench (He et al., 2024), AIME24 (Mathematical Association of America, 2024), and AMC23 (Mathematical Association of America, 2023). Main comparisons use 16 sampled responses per question; analyses use eight. We report average accuracy, denoted Avg@n in tables, and give additional Pass@n results in Appendix C (Tables 5 and 6). Overall scores are unweighted means across benchmarks.

Code generation. We fine-tune Qwen2.5-Coder-3B-Instruct (Hui et al., 2024) on 10,000 examples from OpenCodeInstruct (Ahmad et al., 2025) for three epochs with a learning rate of 5×10^{-6}. We evaluate on HumanEval+ and MBPP+ (Liu et al., 2023), LiveCodeBench (LCB) v5 (Jain et al., 2025), and BigCodeBench-Hard (Zhuo et al., 2025). We report pass@1 and its unweighted mean across the four benchmarks. All methods use the same evaluation subsets; Appendix B.1 specifies their construction.

Baselines and training details. We compare FamiFT with the initial model, standard SFT, DFT (Wu et al., 2026), ProFit (Liu et al., 2026), TALR (Lin et al., 2026), and ASFT (Zhu et al., 2026). Within each task, all methods share the training data and configuration. We use full-parameter training with a global batch size of 256 and a maximum sequence length of 2,048. FamiFT computes

Table 2: Code generation results (pass@1, %) for Qwen2.5-Coder-3B-Instruct fine-tuned on OpenCodeInstruct-10k. Base denotes the initial checkpoint. Avg. is the mean over the four benchmarks; Δ is its change from Base in percentage points; both are rounded after calculation. **Bold** and underlined scores mark the best and second-best distinct values, including ties.

Model	Method	HumanEval+	MBPP+	LCB v5	BigCodeBench Hard	Avg.	Δ
Qwen2.5 Coder-3B Instruct	Base	76.22	65.08	9.58	12.77	40.91	—
	SFT	78.66	65.61	11.38	**15.60**	42.81	+1.90
	DFT	79.27	66.40	11.98	14.18	42.96	+2.05
	ProFit	79.27	66.14	12.57	13.48	42.87	+1.95
	TALR	**79.88**	**66.93**	11.98	14.18	43.24	+2.33
	ASFT	78.66	**66.93**	**13.17**	12.06	42.71	+1.79
	FamiFT	**79.88**	**66.93**	12.57	**15.60**	**43.75**	**+2.83**

familiarity scores once and keeps them fixed during training. Full configurations and baseline objectives are in Appendices B.1 and B.3.

5.2 Main results

Mathematical reasoning. FamiFT achieves the highest average accuracy on both math models (Table 1): 37.20 on Qwen2.5-Math-7B and 36.02 on Qwen3-8B-Base, improving over SFT by 15.88 and 13.94 percentage points. It also improves over the strongest token-weighting baseline for each model: DFT by 0.44 points on Qwen2.5-Math-7B and TALR by 0.98 points on Qwen3-8B-Base. These results show that adapting the weighting rule to example familiarity adds gains over existing token-weighting methods.

The gains extend across benchmarks. Relative to DFT, FamiFT improves four of five scores on Qwen2.5-Math-7B, including MATH500 from 69.10 to 70.60 and AMC23 from 41.60 to 44.40. On Qwen3-8B-Base, average accuracy rises from 34.56 to 36.02, with gains of 3.00 points on Minerva Math, 2.90 on AIME24, and 2.10 on AMC23. The improvement therefore spans several benchmarks on both models.

Code generation. FamiFT achieves the highest average accuracy across the four code benchmarks at 43.75 (Table 2), exceeding DFT by 0.79 percentage points and SFT by 0.93 points. It improves over DFT on all four benchmarks, with gains of 0.61 points on HumanEval+, 0.53 on MBPP+, 0.59 on LCB v5, and 1.42 on BigCodeBench-Hard. It also ties for the best score on HumanEval+, MBPP+, and BigCodeBench-Hard. Together with the math results, these comparisons support familiarity-adaptive weighting across both reasoning and code-generation tasks.

5.3 Mechanism Analysis

Example-specific familiarity. We test whether the example-adaptive exponent in FamiFT is necessary using two controls. First, we replace B_i with a shared exponent $B \in \{0.5, 0.75, 1, 1.25, 1.5\}$ for all examples; $B = 1$ recovers DFT. Second, we randomly permute the familiarity scores and use $B_i = 1 + g_{\sigma(i)} - M$. This preserves the exponent distribution and dataset center while breaking the correspondence between each example and its familiarity score. Figure 4 shows that FamiFT reaches 37.20 average accuracy. The best fixed setting is DFT at 36.76, while shuffled familiarity reaches 35.88. The 1.32-point gain over shuffling shows that matching familiarity to examples matters beyond using a distribution of varying exponents. The 0.44-point gain over the best shared exponent further supports adapting the weighting strength to individual examples.

Sample weight and relative token weights. An exponent changes both the mean weight of a response and the ratios between its token weights. At each model's current state, define

$$\bar{w}_i^F = \frac{1}{T_i}\sum_{t=1}^{T_i} p_{i,t}^{B_i}, \qquad \bar{w}_i^D = \frac{1}{T_i}\sum_{t=1}^{T_i} p_{i,t}. \tag{7}$$

We separate these effects with two controls:

$$w_{i,t}^{\text{scale}} = \text{sg}\left(\frac{\bar{w}_i^F}{\bar{w}_i^D + \epsilon} p_{i,t}\right), \qquad w_{i,t}^{\text{shape}} = \text{sg}\left(\frac{\bar{w}_i^D}{\bar{w}_i^F + \epsilon} p_{i,t}^{B_i}\right). \tag{8}$$

Scale-only matches FamiFT's mean response weight while retaining DFT's token ratios. Shape-only retains FamiFT's token ratios while matching DFT's mean response weight. All weights, including the matching factors, are detached. Table 3 shows that shape-only reaches 36.48 average accuracy, compared with 35.44 for scale-only. Combining both in FamiFT reaches 37.20, improving by 0.72 and 1.76 points, respectively. Thus, relative token weighting contributes more in this comparison, while the full rule gives the highest accuracy. Per-benchmark results are in Appendix D.4.

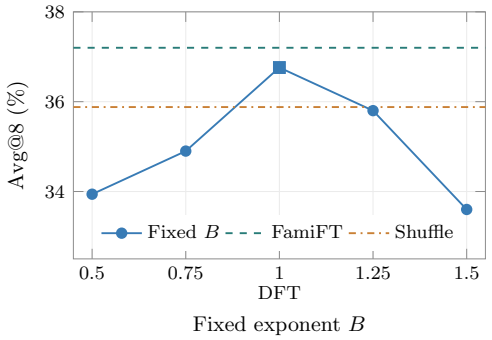

Figure 4: Fixed exponents and shuffled familiarity. The square marks DFT ($B = 1$); horizontal lines show FamiFT and its shuffled control.

Table 3: Matched scale and shape controls (%). Mean and ratios show the rule retained by each control: D for DFT and F for FamiFT.

Method	Mean	Ratios	Avg@8	Pass@8
DFT	D	D	36.76	55.24
Scale-only	F	D	35.44	53.84
Shape-only	D	F	36.48	55.50
FamiFT	F	F	**37.20**	**55.70**

5.4 Ablation Study

Exponent sensitivity. To further analyze FamiFT, we express its exponent in a more general form:

$$B_i = \beta + \lambda(g_i - M). \tag{9}$$

Here, β sets the exponent at the dataset median, and λ controls how strongly the exponent changes with familiarity. The default is $(\beta, \lambda) = (1, 1)$; $(1, 0)$ recovers DFT. We vary one parameter at a time on Qwen2.5-Math-7B.

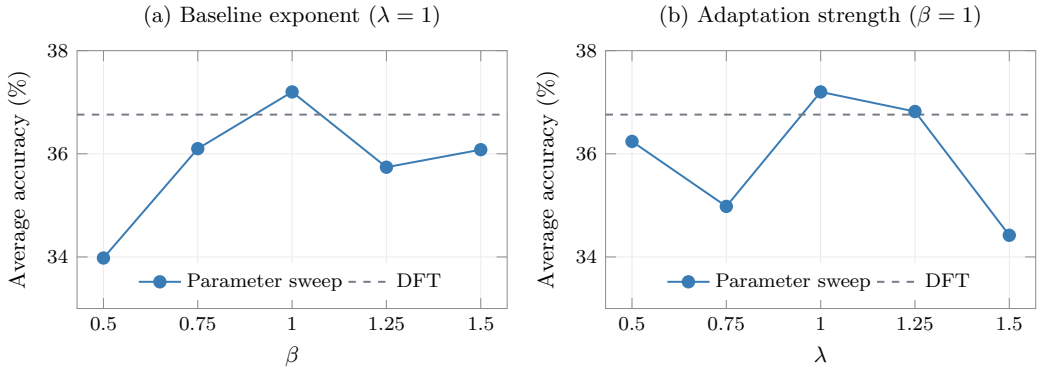

Figure 5: Exponent sensitivity on Qwen2.5-Math-7B. (a) Varying the baseline exponent $\beta \in [0.5, 1.5]$ with $\lambda = 1$. (b) Varying the positive adaptation strength $\lambda \in [0.5, 1.5]$ with $\beta = 1$. Blue curves show average accuracy; dashed lines mark DFT. Full settings and scores are in Appendix D.5.

The default $\beta = 1$ reaches 37.20 average accuracy, compared with 36.10 at $\beta = 0.75$ and 35.74 at $\beta = 1.25$ (Figure 5a). Its lead among the tested values supports centering the adaptive exponent

at DFT's weighting rule. The default $\lambda = 1$ reaches 37.20 average accuracy, followed by 36.82 at $\lambda = 1.25$ (Figure 5b). Removing adaptation ($\lambda = 0$) recovers DFT and gives 36.76. Reversing the assignment ($\lambda = -1$), so that less familiar examples receive sharper weighting and more familiar examples receive flatter weighting, reduces accuracy to 35.64. The default therefore gains 0.44 points over no adaptation and 1.56 points over the inverse assignment, supporting both the use and the direction of familiarity-based adaptation. Appendix D.5 reports the full sweep.

Weighting variants. Writing each token-level loss as $-w_{i,t} \log p_{i,t}$, we study several ways to construct $w_{i,t}$. We place these variants in the general family $w_{i,t} = \mathrm{sg}(A_i p_{i,t}^{B_i})$, where A_i changes the overall scale of an example and B_i changes its relative token weights (Table 4).

Table 4: Weighting variants on Qwen3-8B (%). We use $w_{i,t} = \mathrm{sg}(A_i p_{i,t}^{B_i})$ and $\delta_i = g_i - M$. Bold and underlined scores mark the best and second-best distinct values in each column.

Variant	A_i	B_i	MATH500	Minerva	Olympiad	AIME24	AMC23	Avg@8	Pass@8
DFT	1	1	**69.9**	24.8	**32.2**	7.1	38.8	34.56	50.84
Sample scaling	$1 + \delta_i$	1	69.3	27.6	30.8	**10.0**	38.4	35.22	51.34
Sample + exponent	$1 + \delta_i$	$1 + \delta_i$	<u>69.7</u>	<u>27.7</u>	<u>31.7</u>	<u>8.3</u>	**41.2**	<u>35.72</u>	<u>53.36</u>
FamiFT	1	$1 + \delta_i$	68.9	**27.8**	31.5	**10.0**	<u>40.9</u>	**35.82**	**54.08**

Sample scaling with $A_i = 1 + g_i - M$ and $B_i = 1$ reaches 35.22 average accuracy. Combining sample scaling with the familiarity-dependent exponent increases it to 35.72, while FamiFT's exponent-only weighting reaches 35.82. Among these variants, the exponent-only rule gives the highest average accuracy, supporting the choice $A_i = 1$ with a centered, example-specific exponent.

6 Conclusion

We studied how probability-based token weighting should vary across training examples and found a consistent relationship with pretrained-model familiarity: relatively familiar examples favor sharper weighting, whereas relatively unfamiliar examples favor flatter weighting. Building on this observation, we introduced FamiFT, which converts each example's initial-model response familiarity into a fixed, sample-specific exponent over online token probabilities. This simple design combines stable sample-level information from the pretrained model with dynamic token-level signals during fine-tuning. Across mathematical reasoning and code generation, FamiFT improves average performance over SFT and competitive token-weighting baselines. Controlled analyses further show that the correspondence between familiarity and individual examples matters, and adapting relative token weights within familarity is an important source of the improvement. Together, these results highlight weighting strength itself as an example-dependent design choice in supervised fine-tuning and suggest pretrained-model familiarity as a useful signal for determining how supervision should be distributed across tokens.

AI Use Statement

Generative AI tools assisted with refining the paper narrative and and drafting code for analysis and figures. The authors reviewed the AI-assisted outputs, independently checked all reported experimental results, and take responsibility for the paper's content, claims, and artifacts.

REFERENCES

Wasi Uddin Ahmad, Aleksander Ficek, Mehrzad Samadi, Jocelyn Huang, Vahid Noroozi, Somshubra Majumdar, and Boris Ginsburg. OpenCodeInstruct: A large-scale instruction tuning dataset for code LLMs. *arXiv preprint arXiv:2504.04030*, 2025. URL https://arxiv.org/abs/2504.04030.

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. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 10818–10838. PMLR, 2025. URL https://proceedings.mlr.press/v267/chu25c.html.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024. URL https://jmlr.org/papers/v25/23-0870.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.

Daya Guo, Dejian Yang, Haowei Zhang, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645:633–638, 2025. doi: 10.1038/s41586-025-09422-z. URL https://www.nature.com/articles/s41586-025-09422-z.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen 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)*, pp. 3828–3850. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.211. URL https://aclanthology.org/2024.acl-long.211/.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

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. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/94074dd5a072d28ff75a76dabed43767-Abstract-Conference.html.

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://proceedings.neurips.cc/paper_files/paper/2022/hash/18abbeef8cfe9203fdf9053c9c4fe191-Abstract.html.

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. *arXiv preprint arXiv:2510.00526*, 2025. URL https://arxiv.org/abs/2510.00526.

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 repository, 2024. URL https://huggingface.co/datasets/AI-MO/NuminaMath-CoT.

Jiacheng Lin, Zhongruo Wang, Kun Qian, Tian Wang, Arvind Srinivasan, Hansi Zeng, Ruochen Jiao, Xie Zhou, Jiri Gesi, Dakuo Wang, Yufan Guo, Kai Zhong, Weiqi Zhang, Sujay Sanghavi, Changyou Chen, Hyokun Yun, and Lihong Li. SFT doesn't always hurt general capabilities: Revisiting domain-specific fine-tuning in LLMs. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/4e447acb68f57e29234bc0eb19896f11-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. doi: 10.52202/075280-0943. URL https://papers.neurips.cc/paper_files/paper/2023/hash/43e9d647ccd3e4b7b5baab53f0368686-Abstract-Conference.html.

Tao Liu, Taiqiang Wu, Runming Yang, Shaoning Sun, Junjie Wang, and Yujiu Yang. ProFit: Leveraging high-value signals in SFT via probability-guided token selection. *arXiv preprint arXiv:2601.09195*, 2026. URL https://arxiv.org/abs/2601.09195.

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

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

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022. URL https://arxiv.org/abs/2203.02155.

Sunny Sanyal, Hayden Prairie, Rudrajit Das, Ali Kavis, and Sujay Sanghavi. Upweighting easy samples in fine-tuning mitigates forgetting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 52922–52957. PMLR, 2025. URL https://proceedings.mlr.press/v267/sanyal25a.html.

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.

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://proceedings.iclr.cc/paper_files/paper/2026/hash/2e53c02ea028cbf603f4b6b47fef3d97-Abstract-Conference.html.

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.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Zhilu Zhang and Mert R. Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/f2925f97bc13ad2852a7a551802feea0-Abstract.html.

Yucheng Zhou, Junwei Sheng, Qianning Wang, and Jianbing Shen. Compatibility-aware dynamic fine-tuning for large language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 29997–30008. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1383. URL https://aclanthology.org/2026.acl-long.1383/.

He Zhu, Junyou Su, Peng Lai, Ren Ma, Wenjia Zhang, Linyi Yang, and Guanhua Chen. Anchored supervised fine-tuning. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=PORko7QT64.

Terry Yue Zhuo, Minh Chien Vu, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widyasari, Imam Nur Bani Yusuf, Haolan Zhan, Junda He, Indraneil Paul, Simon Brunner, Chen Gong, Thong Hoang, Armel Randy Zebaze, Xiaoheng Hong, Wen-Ding Li, Jean Kaddour, Ming Xu, Zhihan Zhang, Prateek Yadav, Naman Jain, Alex Gu, Zhoujun Cheng, Jiawei Liu, Qian Liu, Zijian Wang, Binyuan Hui, Niklas Muennighoff, David Lo, Daniel Fried, Xiaoning Du, Harm de Vries, and Leandro von Werra. BigCodeBench: Benchmarking code generation with diverse function calls and complex instructions. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/a6a90bcc2aa470c3871b2d39a67d26e8-Abstract-Conference.html.

A Additional Method Details

A.1 Scoring and training

Offline scoring uses the same initial checkpoint, tokenizer, prompt format, response positions, and truncation as training. Each example receives one stored familiarity score g_i. The center is computed once from all training examples. Prompt and padding positions do not contribute to either scoring or training.

> **Offline:** Compute g_i using Equation 4. Set $M = \mathrm{median}_i(g_i)$ and store $B_i = 1 + g_i - M$.
> **For each optimizer step:** Zero gradients, then for each microbatch:
> 1. Compute target log probabilities $\log p_{i,t}$.
> 2. Compute $w_{i,t} = \mathrm{sg}[\exp(B_i \log p_{i,t})]$ in FP32.
> 3. Normalize the weighted NLL sum by the response-token count and backpropagate.
> After all microbatches, update the model parameters.

The recorded trainer normalizes each microbatch by its valid response-token count. With `balance_dp_token` enabled, the denominator instead sums these counts across data-parallel ranks at the same microstep, and each rank's local loss is multiplied by the number of ranks to compensate for gradient averaging. This balances tokens within that microstep, not across the full optimizer batch.

The trainer calls backward before dividing the returned loss by the number of microbatches. That outer division therefore affects logging only; gradients from successive microbatches are accumulated without this division. This differs from a single global token-mean loss, particularly when microbatches contain different numbers of response tokens. All methods in a controlled comparison share this training loop. These details follow the available implementation records; the original trainer source should be checked when reproducing the runs.

B Evaluation and Baseline Details

B.1 Training and evaluation configurations

Mathematical reasoning. We fine-tune Qwen2.5-Math-7B (Yang et al., 2024) and Qwen3-8B-Base (Yang et al., 2025) for one epoch on Numina-10k, a fixed 10,000-example subset of NuminaMath-CoT (Li et al., 2024). We use a learning rate of 5×10^{-5}. Evaluation covers MATH500 (Hendrycks et al., 2021), Minerva Math (Lewkowycz et al., 2022), OlympiadBench (He et al., 2024), AIME24 (Mathematical Association of America, 2024), and AMC23 (Mathematical Association of America, 2023). The main comparison uses 16 sampled responses per question for both Qwen2.5-Math-7B and Qwen3-8B-Base. Ablations use eight responses per question. Avg@n measures mean response accuracy over n responses, while Pass@n measures the fraction of questions solved by at least one response. The appendix also reports the existing 16-response evaluations for both models. Overall scores average the five benchmark scores equally. Appendix B.2 gives the metric definitions.

Code generation. We fine-tune Qwen2.5-Coder-3B-Instruct (Hui et al., 2024) on a fixed 10,000-example subset of OpenCodeInstruct (Ahmad et al., 2025), selected with judge-5 balanced sampling. Training runs for three epochs with a learning rate of 5×10^{-6}. We evaluate on HumanEval+ and MBPP+ (Liu et al., 2023), LiveCodeBench (LCB) v5 (Jain et al., 2025), and BigCodeBench-Hard (Zhuo et al., 2025). HumanEval+ and MBPP+ contain 164 and 378 questions, respectively; their evaluations use chat-instruct prompts and greedy decoding with a maximum of 4,096 generated tokens. We report pass@1 on each benchmark and use the unweighted mean of the four scores as the overall result. LCB v5 contains the 167 problems added between v4 and v5. BigCodeBench-Hard scores use the same 141-problem subset for every method.

Baselines and training details. We compare FamiFT with the initial model, standard SFT, DFT (Wu et al., 2026), ProFit (Liu et al., 2026), TALR (Lin et al., 2026), and ASFT (Zhu et al., 2026). DFT uses the target-token probability as a detached weight. ProFit retains tokens with probabilities at least 0.1. TALR adjusts probability weights using a temperature computed from the global batch's median response NLL, with a weight floor of 0.01. ASFT adds a forward KL term from

the frozen initial model with coefficient 0.05. Appendix B.3 specifies the objectives used in our implementation.

All fine-tuning runs use four-GPU full-parameter FSDP2 training, BF16, a global batch size of 256, a maximum sequence length of 2,048 tokens, and training seed 1. The code runs use a microbatch size of two examples per GPU. Within each task, all methods share the training data and training configuration. FamiFT scores the demonstrations once under the initial checkpoint and keeps the familiarity scores and dataset center fixed during training. Appendix A describes loss reduction and gradient accumulation.

B.2 Evaluation metrics

For a benchmark with Q questions, let $c_{q,j} \in \{0, 1\}$ indicate whether response j to question q is correct. With n sampled responses,

$$\text{avg}\,@n = \frac{100}{Qn} \sum_{q=1}^{Q} \sum_{j=1}^{n} c_{q,j}, \qquad \text{pass}\,@n = \frac{100}{Q} \sum_{q=1}^{Q} \mathbf{1}\left[\sum_{j=1}^{n} c_{q,j} > 0\right]. \tag{10}$$

Avg@n estimates single-sample success under the evaluation sampler, while pass@n measures success within the given sampling budget. The main table and appendix Pass@16 tables use $n = 16$ for both models; the ablations use $n = 8$.

B.3 Baseline objectives

All weighting baselines multiply target NLL by detached weights and use valid response-token counts as the local denominator. For padded tensors, let $m_{i,t}$ mark response positions, $T_i = \sum_t m_{i,t}$, and $Z = \sum_{i,t} m_{i,t}$ over the local normalization group. The weights are

$$\begin{aligned} w_{i,t}^{\text{SFT}} &= 1, & w_{i,t}^{\text{DFT}} &= p_{i,t}, \\ w_{i,t}^{\text{ProFit}} &= \mathbf{1}[p_{i,t} \geq 0.1], & w_{i,t}^{\text{TALR}} &= \max(p_{i,t}^{1/\tau}, 0.01). \end{aligned} \tag{11}$$

For TALR, let $u_i = -T_i^{-1} \sum_t m_{i,t} \log p_{i,t}$. The temperature is $\tau = \max(\text{median}_{i \in \mathcal{B}}\, u_i, 10^{-6})$ over the global optimizer batch. It is detached and recomputed online. Unlike FamiFT's exponent, it is shared by examples in that batch.

ASFT adds a reference-to-online forward KL term:

$$\mathcal{L}_{\text{ASFT}} = \frac{1}{Z} \sum_{i,t} m_{i,t} \left[-\,\text{sg}(p_{i,t}) \log p_{i,t} + 0.05\, D_{\text{KL}}(q_{i,t} \| P_{i,t}) \right], \tag{12}$$

where $q_{i,t}$ and $P_{i,t}$ are the full-vocabulary distributions of the initial and online models under the same reference prefix. The KL term is differentiated through $P_{i,t}$. It cannot generally be represented by a single detached scalar multiplying target NLL.

C Full Math Results

Table 1 reports per-benchmark Avg@16 for both models. Tables 5 and 6 report Pass@16 from the existing 16-response evaluations on the same benchmarks. All values are percentages. The bold overall scores mark the highest aggregate in each table.

D Ablation Details and Full Results

D.1 Protocol and control provenance

Unless stated otherwise, the new controls use Numina-10k, four-GPU full-parameter FSDP2 training in BF16, one epoch, learning rate 5×10^{-5}, global batch 256, microbatch one per GPU, maximum training length 2,048, warmup ratio 0.1, and training seed 1. The complete 10k runs take 39 optimizer steps. Evaluation uses `qwen-boxed` prompts, the slow tokenizer, eight samples per question, nominal seed 0, temperature 1, top-p 1, and at most 4,096 generated tokens. Aggregates weight the five benchmarks equally.

Table 5: Qwen2.5-Math-7B: pass@16 (%).

Method	MATH500	Minerva	Olympiad	AIME24	AMC23	Overall
ASFT	90.20	50.40	59.10	23.30	75.00	**59.60**
ProFit	88.80	46.00	55.90	16.70	77.50	56.98
FamiFT	88.20	46.00	56.00	13.30	75.00	55.70
Base	84.00	41.50	48.00	26.70	77.50	55.54
DFT	84.80	44.90	53.20	23.30	70.00	55.24
TALR	86.80	44.90	52.70	16.70	70.00	54.22
SFT	80.80	40.40	40.90	20.00	67.50	49.92

Table 6: Qwen3-8B-Base: pass@16 (%).

Method	MATH500	Minerva	Olympiad	AIME24	AMC23	Overall
Base	93.20	56.20	66.80	33.30	85.00	**66.90**
ASFT	91.20	54.40	64.70	23.30	85.00	63.72
ProFit	91.40	50.40	60.70	20.00	82.50	61.00
TALR	90.40	50.00	60.10	20.00	80.00	60.10
SFT	88.20	53.70	53.80	23.30	75.00	58.80
FamiFT	89.20	48.90	57.00	16.70	75.00	57.36
DFT	88.20	43.00	55.70	20.00	75.00	56.38

D.2 Token-level probabilities and losses

This table lists the recorded token probabilities and the corresponding per-token losses for one response; it is a mechanism display, not an aggregate result. For the FamiFT calculation, we use an observed training-set high-familiarity score nearest to 0.90, $g_i = 0.900490$. With median $M = 0.628053$, this gives $B_i = 1 + g_i - M = 1.272436$.

Table 7: Recorded probabilities and per-token losses for the response in Figure 3. FamiFT uses the observed high-familiarity setting $g_i = 0.900490$ and $B_i = 1.272436$.

Token	p_t	$\log p_t$	SFT	DFT	FamiFT
	0.8000	-0.2231	0.2231	0.1785	0.1680
	0.7000	-0.3567	0.3567	0.2497	0.2266
	0.8500	-0.1625	0.1625	0.1381	0.1322
	0.0600	-2.8134	2.8134	0.1688	0.0784
	0.7000	-0.3567	0.3567	0.2497	0.2266
	0.5500	-0.5978	0.5978	0.3288	0.2794
	0.1200	-2.1203	2.1203	0.2544	0.1428
	0.7000	-0.3567	0.3567	0.2497	0.2266
	0.1500	-1.8971	1.8971	0.2846	0.1697
	0.6500	-0.4308	0.4308	0.2800	0.2490
	0.0900	-2.4079	2.4079	0.2167	0.1125
	0.8500	-0.1625	0.1625	0.1381	0.1322

D.3 Shuffled familiarity and fixed exponents

Average accuracy and sampled coverage. FamiFT has the highest Avg@8 among the fixed-exponent and shuffled controls, while the fixed exponent $B = 0.5$ has the highest Pass@8 (59.00 versus 55.70 for FamiFT). Relative to shuffling, FamiFT gains 1.32 points in Avg@8 and 0.72 in Pass@8. These metrics therefore favor different settings; the main-text comparison focuses on average accuracy.

We sample one permutation σ with seed 42 and keep it fixed throughout training. Shuffled weights are

$$w_{i,t}^{\text{shuffle}} = \text{sg}\left(p_{i,t}^{\,1+g_{\sigma(i)}-M}\right). \tag{13}$$

The permutation preserves the exponent distribution over examples, but changes its assignment. Qwen2.5 fixed-exponent controls use $w_{i,t} = \text{sg}(p_{i,t}^{B})$ for $B \in \{0.5, 0.75, 1, 1.25, 1.5\}$.

D.4 MATCHED RESPONSE WEIGHT AND TOKEN RATIOS

The scale-only and shape-only weights are defined in Equations 7–8. The means are recomputed from each model's own online probabilities, and all weights and matching factors are detached. Loss normalization uses response-token counts. Tables 8 and 9 report all benchmark scores.

Table 8: Qwen2.5-Math-7B: matched controls, Avg@8 (%).

Method	MATH500	Minerva	Olympiad	AIME24	AMC23	Avg.
FamiFT	70.6	29.5	34.4	7.1	44.4	**37.20**
DFT	69.1	28.9	33.0	11.2	41.6	36.76
Shape-only matched	69.2	28.0	32.7	6.3	46.2	36.48
Scale-only matched	70.0	29.6	34.3	6.7	36.6	35.44

Table 9: Qwen2.5-Math-7B: matched controls, Pass@8 (%).

Method	MATH500	Minerva	Olympiad	AIME24	AMC23	Overall
FamiFT	88.2	46.0	56.0	13.3	75.0	**55.70**
Shape-only matched	87.0	44.9	53.9	16.7	75.0	55.50
DFT	84.8	44.9	53.2	23.3	70.0	55.24
Scale-only matched	85.8	46.0	53.2	16.7	67.5	53.84

D.5 BASELINE EXPONENT AND ADAPTATION STRENGTH

The default setting has the highest Avg@8 in both sweeps. Lowering β to 0.5 gives 33.98 Avg@8 and 60.82 Pass@8, compared with 37.20 and 55.70 for the default. Thus, the preferred setting can differ between average accuracy and sampled coverage. We evaluate $B_i = \beta + \lambda(g_i - M)$ by varying one parameter at a time.

Table 10: Qwen2.5-Math-7B: baseline exponent sweep. All values are percentages.

	Avg@8						Pass@8					
β	MATH500	Minerva	Olymp.	AIME24	AMC23	Overall	MATH500	Minerva	Olymp.	AIME24	AMC23	Overall
1.0	70.6	29.5	34.4	7.1	44.4	37.20	88.2	46.0	56.0	13.3	75.0	55.70
0.75	69.3	28.5	32.6	6.7	43.4	36.10	88.2	47.8	54.7	16.7	72.5	55.98
1.50	70.1	27.7	33.2	8.8	40.6	36.08	83.8	42.6	52.6	23.3	70.0	54.46
1.25	68.7	27.9	31.6	6.7	43.8	35.74	83.8	43.0	51.3	16.7	72.5	53.46
0.50	67.8	24.7	31.2	6.2	40.0	33.98	90.0	47.4	57.5	26.7	82.5	60.82

Table 11: Qwen2.5-Math-7B: adaptation-strength sweep. All values are percentages.

	Avg@8						Pass@8					
λ	MATH500	Minerva	Olymp.	AIME24	AMC23	Overall	MATH500	Minerva	Olymp.	AIME24	AMC23	Overall
1.0	70.6	29.5	34.4	7.1	44.4	37.20	88.2	46.0	56.0	13.3	75.0	55.70
1.25	69.4	30.8	33.8	6.7	43.4	36.82	86.6	46.7	55.9	16.7	75.0	56.18
0	69.1	28.9	33.0	11.2	41.6	36.76	84.8	44.9	53.2	23.3	70.0	55.24
0.50	69.3	30.1	33.7	6.2	41.9	36.24	84.0	46.7	53.5	16.7	82.5	56.68
-1.0	69.1	28.2	32.4	8.8	39.7	35.64	86.0	43.4	53.9	23.3	65.0	54.32
0.75	67.9	29.6	32.8	4.6	40.0	34.98	85.6	44.9	55.3	16.7	77.5	56.00
1.50	67.3	27.2	31.3	3.8	42.5	34.42	85.2	45.6	54.5	10.0	75.0	54.06