BREW: Progress-Aware Token Weighting for Open-Ended Writing

Anonymous authors
Paper under double-blind review

Abstract

Supervised fine-tuning (SFT) for open-ended writing learns from a single recorded realization even where several continuations are plausible. Under uniform SFT, every supervised token keeps weight one throughout training, so continued fitting concentrates probability on the recorded realization even after the token becomes predictable, a phenomenon we call *single-reference overcommitment*. Existing confidence-based weighting partly eases this problem by downweighting low-confidence tokens, but its weight can keep rising as the target becomes more predictable, leaving no signal to reduce weighting after substantial progress. We introduce BREW (**B**ase-**RE**lative **W**eighting), which compares each token's current loss with its loss under the frozen base model. The current loss measures how learnable the token is now, while the base-relative ratio measures progress from its own starting point. This keeps the cautious entry of confidence weighting while adding a progress-aware exit. In the primary setting, BREW performs best on every writing metric, improving WritingBench by 2.1 and LongBench-Write by 3.7 over the strongest baseline. Its WritingBench advantage holds across models and datasets, and the same configuration remains competitive on answer-determined mathematical tasks.

1 Introduction

Supervised fine-tuning for open-ended writing repeatedly optimizes every token of a single recorded response, even at positions where several continuations are equally plausible (Fig. 1, top). Learning

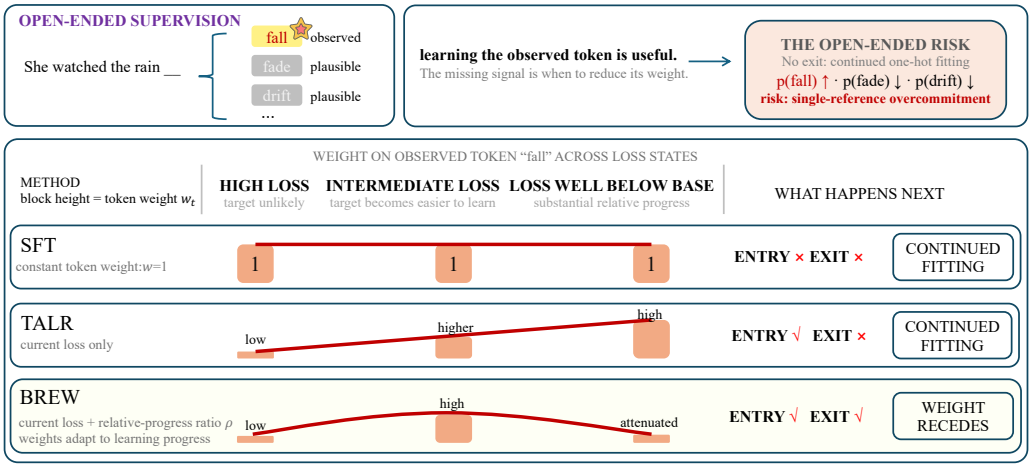

Figure 1: Weight profiles across the states a supervised token passes through, ordered by current loss. **Top:** single-reference overcommitment; continued fitting keeps raising $p(\text{fall})$ although other continuations are plausible. **Bottom:** reading current loss alone gives cautious entry but no exit; BREW adds the base-relative ratio $\rho = \ell_\theta/\ell_b$, so weight peaks at intermediate loss and recedes after substantial relative progress. Setting γ=0 recovers TALR.

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.

the reference is necessary, but one-hot supervision treats the recorded token as the only correct target throughout training: under uniform SFT, every supervised token keeps weight one from start to finish. At locally underdetermined positions, the recorded token may be only one of several reasonable realizations, so continued fitting can drift from learning writing ability toward reinforcing one specific way of writing; repeated fitting has been linked to homogenization in open-ended generation (Jiang et al., 2025; Karouzos et al., 2026). No signal tells the model when a token has been learned well enough.

More data improves coverage but not this training-time problem: each prompt still provides one recorded realization. Multi-reference supervision directly reduces the ambiguity, but each additional long-form realization is costly, while model-generated references inherit the generator's own stylistic tendencies. Nor do extra references tell the optimizer when further fitting of a particular observed token has become unnecessary. This motivates an objective-side signal that changes with training progress. Since tokens become learnable and progress at different rates, token-level weighting is a natural way to express it.

Existing token-level weighting methods mostly base their weights on the student's current confidence: DFT favors tokens the model already predicts well (Wu et al., 2026), TALR tempers this confidence with a self-referential temperature (Lin et al., 2026), and InfoSFT applies a band-pass anchored at a global constant (Sabbaghi et al., 2026). They mainly ask whether a token is learnable now, downweighting low-confidence targets. For low-confidence tokens, this reduces the emphasis on the single recorded continuation and partly eases single-reference overcommitment. In our setting, these rules outperform uniform SFT on two external writing benchmarks (§4.2).

But the benefit weakens as the observed token becomes more predictable. These rules read weight from the current state alone, so two tokens with the same current loss receive the same weight even when their histories are completely different. One may still sit near the base model's level with little progress. The other may have descended from a much higher loss and already moved substantially toward the recorded realization. Treating them identically ignores how much of the supervision the model has already learned. This matters even more for rules whose weight rises with confidence, such as DFT and TALR: the better the target is fitted, the higher its weight, so the recorded realization keeps receiving more emphasis after it becomes predictable. The training dynamics show this pattern directly: in the bin of tokens the base finds moderately hard, TALR's mean weight rises as those tokens make substantial progress (Fig. 3B). To tell whether a token has been learned enough, knowing how easy it is now is not sufficient; one also needs to know how far it has come from its own starting point.

BREW (**B**ase-**RE**lative **W**eighting) builds on TALR's confidence-based weighting by adding a progress term measured relative to the base model's starting point, supplying the missing exit signal while keeping cautious entry (Fig. 1, bottom). Before training begins, we cache the base model's loss ℓ_b on every supervised token; during training we compare the student's current loss ℓ_θ against it. The two signals capture different aspects of learning: the current loss reflects how learnable the token is now, while the base-relative ratio tracks progress from its starting point. Together they create a progress-aware learning window: weight is low while a difficult token is still hard to learn, rises as it becomes learnable, and recedes after substantial progress from its starting point, reducing single-reference overcommitment. Across tokens, these state-dependent weights form a moving curriculum (§4.3.2).

In the primary Qwen3.5-9B setting, BREW performs best on every writing metric at the same training budget, improving WritingBench by 2.1 over the strongest baseline TALR ($76.3 \rightarrow 78.4$) and LongBench-Write by 3.7. Checkpoint replay shows the predicted weight dynamics: initially hard positions first receive less of the training budget; as they become learnable, that budget is restored; once they have improved substantially relative to their starting points, their weights recede. By the end of training, near-zero fitting is 7.5–17.3 percentage points lower than TALR's in every base-defined difficulty bin. BREW's WritingBench advantage extends to a second model and a second writing corpus (§4.4), and the same configuration remains competitive on answer-determined mathematics (§4.4).

Contributions. **Method:** a base-relative token weighting rule that pairs cautious entry with a progress-aware exit by comparing current loss against a token-specific frozen-base reference (§3). **Mechanism:** checkpoint replay showing how token weights shift with learning progress, decreasing

after substantial relative progress and reducing near-zero fitting across all base-defined difficulty bins (§3.3, §4.3.2). **Evidence:** equal-budget writing comparisons across two models and two datasets, a component ablation isolating the relative-progress term, and transfer to answer-determined mathematics (§4.2, §4.4, §4.4).

2 Related Work

Homogenization in open-ended generation. Post-training can homogenize open-ended generation, with collapse appearing during SFT or preference optimization and persisting in model weights (Jiang et al., 2025; Klypa & Cherednichenko, 2026; Karouzos et al., 2026). Existing mitigations modify preference optimization or objective-level entropy (Chung et al., 2025; Li et al., 2025), whereas BREW changes the per-token weighting of the supervised objective.

Token-weighted supervised fine-tuning. A growing family of methods keeps the demonstration data fixed and selects or reweights tokens instead. The signal is read from the student's current state (Wu et al., 2026; Lin et al., 2026; Sabbaghi et al., 2026), from a held-out model (Mindermann et al., 2022; Lin et al., 2024) or an earlier self (Qin et al., 2026), or from pretrained token probabilities or cumulative probability mass (Wang et al., 2026b). AlphaToken (Qing et al., 2026) selects tokens using gradient-based valuation with a Fisher-drift stability proxy. A related line pursues stability through KL anchoring, proximal updates, or emphasis on easy samples (Zhu et al., 2026a;b; Sanyal et al., 2025), and Xie et al. (2026) unifies these rules by the target distribution each induces. These rules differ in where the signal comes from (Shen et al., 2026), but none tracks a token's progress relative to its own pre-training loss: the weight is fixed before training, read from a current snapshot, or set against a reference that is not the token's own starting point. BREW compares the current loss with the token's frozen pre-training loss, so the weight can decrease after substantial relative progress (§3.1).

Objectives using a frozen reference. A separate line of work introduces frozen reference models into the objective. LP-SFT (Wang et al., 2026d) preserves the relative distribution over adaptively selected non-target alternatives through a locally normalized KL objective. For open-ended writing, TCER (Wang et al., 2026c) uses frozen specialist and generalist reference policies to build a corrected endogenous reward for reinforcement fine-tuning. BREW instead compares the current student's loss with its frozen starting loss to adjust token weights during SFT.

Curriculum and self-paced learning. Curriculum learning orders training examples from easy to hard under a fixed schedule (Bengio et al., 2009); self-paced learning instead selects examples dynamically by the learner's current loss (Kumar et al., 2010), and competence-based variants tie the pacing to a competence measure (Platanios et al., 2019). These are all *sample-level* schedulers driven by the learner's own state. BREW is curriculum-like but operates at the *token* level and is paced by progress relative to a frozen base rather than by the learner's absolute loss, so its notion of "easy" is anchored to what the base already knew instead of to the student's transient confidence.

Data-side and decoding-side work. Data-side approaches improve supervision by constructing richer long-form targets or reasoning traces (Bai et al., 2025; Wang et al., 2026a). These methods complement rather than replace token-level weighting. At inference time, Verbalized Sampling mitigates mode collapse without changing training (Zhang et al., 2026).

3 Method

3.1 Base-relative weighting objective

Let

$$\ell_\theta(t) = -\log p_\theta(y_t \mid x, y_{<t})$$

be the student's loss on supervised token y_t, and let

$$\ell_b(t) = -\log p_b(y_t \mid x, y_{<t})$$

be the loss of the frozen base model on the same token. We compute ℓ_b once before training and cache it for all supervised tokens.

BREW measures progress relative to this token-specific starting point (pipeline in Fig. 2):

$$\rho = \frac{\ell_\theta}{\ell_b}, \qquad \ell_{\text{eff}} = \max(0,\ \ell_\theta - \gamma \log \rho)\,. \tag{1}$$

For each batch, we compute the sequence-level temperature

$$\tau = \text{median}_{\text{seq}}(\ell_{\text{eff}}^{-}), \tag{2}$$

following TALR, and assign each supervised token

$$w = \max\Big(w_{\min}, e^{-\ell_{\text{eff}}/\tau}\Big)\,. \tag{3}$$

The training objective is

$$\mathcal{L}_{\text{BREW}} = \frac{1}{N}\sum_t \text{sg}(w_t)\,\ell_\theta(t), \tag{4}$$

where sg denotes stop-gradient.

The two losses play different roles. The current loss ℓ_θ measures how learnable the token is now, while the ratio $\rho = \ell_\theta/\ell_b$ measures how much progress it has made from its own starting point. When the student improves relative to the base ($\rho < 1$), the progress term lowers the token's weight relative to confidence-only weighting at the same current loss. When the student is worse than the base ($\rho > 1$), it receives additional weight, capped at the full SFT weight 1 by the floor on ℓ_{eff}.

We measure substantial progress by $\rho < 0.25$, a fourfold reduction in surprisal relative to the frozen base. Near-zero fitting is a separate endpoint statistic: $\ell_\theta < 0.01$, i.e. final $q > 0.990$, indicating near-deterministic prediction of the observed token.

We use $\gamma = 0.75$ and $w_{\min} = 0.1$ throughout the main experiments. Because the base losses are cached once, BREW adds no base-model forward pass during training.

3.2 Cautious entry and progress-aware exit

On the improvement domain ($\ell_\theta < \ell_b$), before applying the weight floor,

$$\tilde{w} = (q \cdot (\ell_\theta/\ell_b)^\gamma)^{1/\tau}\,, \qquad q = e^{-\ell_\theta}. \tag{5}$$

This form separates two effects. The confidence term $q^{1/\tau}$ gives initially difficult tokens lower weight, providing cautious entry as in TALR. The relative-progress term $(\ell_\theta/\ell_b)^{\gamma/\tau}$ reduces weight after the token has improved substantially from its starting point, providing the missing exit signal.

As a difficult token becomes easier to predict, the confidence term first raises its weight. After enough relative progress, the progress term dominates and the weight falls again. Thus, for tokens with $\ell_b > \gamma$, the unfloored weight peaks at the intermediate loss $\ell_\theta = \gamma$ (Proposition 1). If $\ell_b \le \gamma$, the peak lies at the boundary instead. Setting $\gamma = 0$ removes the progress term and recovers TALR exactly.

The floor $w_{\min}$ preserves a minimum learning signal for base-unexpected tokens. Across tokens, these state-dependent weights form a moving curriculum, which we examine empirically in §4.3.2.

3.3 Properties

The key shape property is the rise-then-fall profile described above.

Proposition 1 (Rise-then-fall; γ is the peak location)**.** Fix ℓ_b, τ with $0 < \gamma < \ell_b$, and consider the unfloored weight $\tilde{w} = e^{-\ell_{\text{eff}}/\tau}$ of Eq. 1. On $\ell_\theta \in (0, \ell_b)$, $\tilde{w}$ has a unique maximum at $\ell_\theta = \gamma$, with peak value $\tilde{w}_{\max} = (\gamma/\ell_b)^{\gamma/\tau} e^{-\gamma/\tau}$. If $\ell_b \le \gamma$, $\tilde{w}$ is strictly increasing on $(0, \ell_b)$ and its supremum is approached as $\ell_\theta \to \ell_b^{-}$.

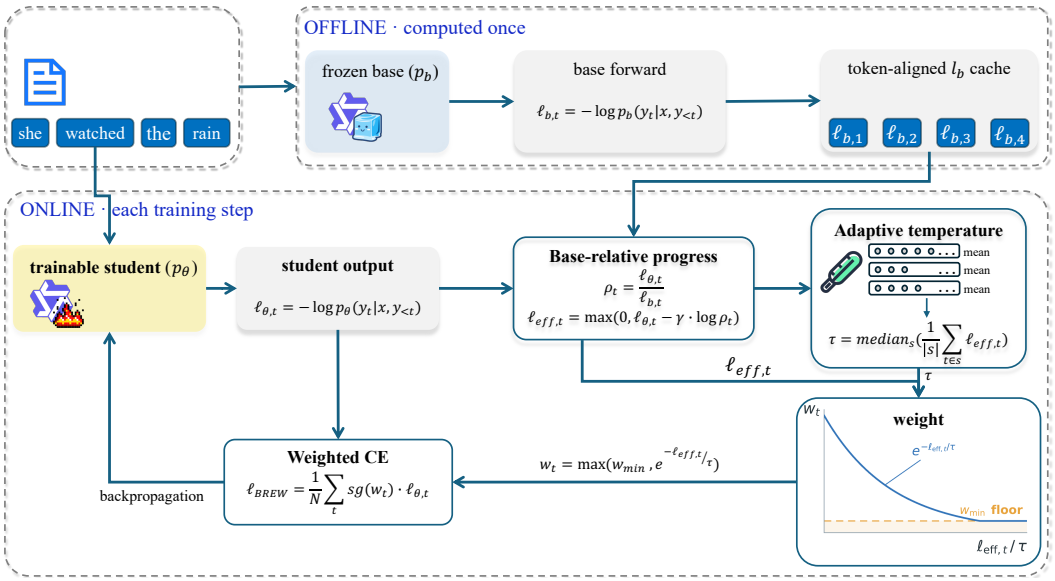

Figure 2: BREW training pipeline: a one-time frozen-base cache supplies ℓ_b, and each training step derives ρ, ℓ_{eff}, τ, and the detached token weight w that scales the cross-entropy update (details in §3).

In our setting, $\gamma = 0.75$ places the peak at $q = e^{-0.75} \approx 0.47$. The peak height decreases with ℓ_b, while the floor $w_{\min}$ keeps a minimum learning signal for difficult targets.

Two additional properties are useful in practice. First, under memorization TALR's token-level statistic tends toward zero, whereas BREW's effective loss moves upward. For the sequence-level temperature used in training, the resulting direction depends on how these changes are distributed across sequences; we examine the temperature trajectories empirically in Appendix C. Second, the positive floor prevents extremely low-confidence tokens from receiving vanishingly small updates. Formal statements and proofs are given in Appendix A.

4 Experiments

4.1 Setup

Data and evaluation. Our primary setting is Qwen3.5-9B fine-tuned on the DeepWriting dataset of Wang et al. (2026a) (20,395 usable long-form samples with think-format supervision). We test model transfer with Qwen3-8B (Yang et al., 2025) on the same dataset and data transfer with Qwen3.5-9B on the LongWriter dataset of Bai et al. (2025) (5,993 usable samples, no-think supervision). Dataset provenance and preprocessing are given in Appendix B; transfer to mathematics is evaluated in §4.4.

Baselines. We compare against standard SFT (uniform token weighting, w=1) and four representative weighting objectives: DFT (w=q) (Wu et al., 2026), KL-anchored ASFT (Zhu et al., 2026a), TALR with its sequence-median temperature (Lin et al., 2026), and InfoSFT (Sabbaghi et al., 2026).

Training protocol. All methods use the same data order, packing, optimizer, and learning-rate schedule, and are trained for four epochs. Results are averaged over three independent runs. BREW uses (γ=0.75, $w_{\min}$=0.1), chosen on the main setting and kept unchanged across all transfer settings. Full training details are in Appendix B; sensitivity and training trajectories are reported in Appendices D and E.

Writing evaluation. We evaluate open-ended writing with three complementary benchmark families. WritingBench evaluates generative writing across six domains with its Critic-7B rubric scorer (Wu et al., 2025): (A) Academic & Engineering, (B) Finance & Business, (C) Politics & Law,

Table 1: Main writing results on Qwen3.5-9B with the DeepWriting dataset. WB-A–F are the six WritingBench domains.

Method	WB	LB	HB-A	HB-B	WB-A	WB-B	WB-C	WB-D	WB-E	WB-F
SFT	74.3	73.1	55.1	61.5	72.2	75.4	71.4	75.5	75.4	76.9
DFT	65.6	75.7	74.1	66.6	68.5	66.5	64.0	61.2	67.8	67.3
ASFT	72.7	78.1	78.7	72.0	73.8	73.6	71.4	71.6	73.9	72.6
InfoSFT	75.7	81.4	75.6	75.2	76.2	76.9	74.3	74.6	77.0	76.0
TALR	76.3	83.4	78.2	77.4	77.4	76.8	74.3	75.7	76.7	77.5
BREW	**78.4**	**87.1**	**79.9**	**78.4**	**79.1**	**78.9**	**76.7**	**78.6**	**78.5**	**79.1**

Table 2: Component ablations on the main setting; one component changed per row. Bold marks the per-column best.

	WB	WB-A	WB-B	WB-C	WB-D	WB-E	WB-F
BREW (full)	**78.4**	**79.1**	**78.9**	**76.7**	**78.6**	**78.5**	**79.1**
(a) Reference: per-token and frozen							
no reference (γ=0 $\rightarrow$ TALR)	76.3	77.4	76.8	74.3	75.7	76.7	77.5
per-token ℓ_b $\rightarrow$ corpus constant	76.6	77.4	77.2	74.7	76.2	77.8	76.9
frozen ℓ_b $\rightarrow$ refreshed every $1/2$ epoch	76.1	76.5	76.4	74.7	75.9	76.6	77.2
(b) Control: reduced effective step size							
SFT at matched effective LR ($5.6\text{e}{-6}$)	58.5	52.4	56.2	55.8	66.4	55.9	65.4
(c) Role of the weight floor							
no floor ($w_{\min}$=0)	77.3	78.1	78.2	75.2	77.5	77.4	77.9

(D) Literature & Arts, (E) Education, (F) Advertising & Marketing. LongBench-Write evaluates long-form generation, while HelloBench contributes its Open-Ended QA (HB-A) and Heuristic Text Generation (HB-B) subsets (Bai et al., 2025; Que et al., 2024); both follow their official prompts and aggregation with `gpt-4o-2024-08-06` as judge (Appendix K). All writing scores are reported on a 0–100 scale. We additionally report IFEval (Zhou et al., 2023) for instruction-following retention (Appendix J).

4.2 Main results: Qwen3.5-9B with the DeepWriting dataset

Writing quality. BREW performs best on all ten metrics in Table 1. It improves WritingBench by 2.1 over TALR, the strongest baseline (76.3 $\rightarrow$ 78.4), with gains of 1.6–2.9 across all six domains. It also improves LongBench-Write by 3.7, HB-A by 1.2, and HB-B by 1.0 over the corresponding runner-up.

The baseline results also show why both entry and exit matter. DFT gives very little weight to initially unlikely targets and falls to 65.6 WB, below uniform SFT. TALR tempers this behavior and reaches 76.3, providing a much better entry policy. But its weight continues to rise as targets become more predictable. BREW adds a progress-aware exit to TALR's cautious entry. The following ablations and training trajectories examine the contribution of this term and how it changes token weights.

The gain is broad rather than concentrated in one category: BREW improves over TALR in all six WritingBench domains, with the largest increase on Literature & Arts (+2.9). DFT, TALR, and InfoSFT all outperform uniform SFT on the three external metrics, so confidence-based weighting helps before a base-relative exit is added; on LongBench-Write BREW further improves both length compliance and judged quality over TALR (S_l: 76.8 $\rightarrow$ 81.3; S_q: 89.9 $\rightarrow$ 93.0; Appendix K).

4.3 Mechanism and ablations

4.3.1 Component ablations

Table 2 changes one component at a time at the identical budget, grouped by the question each row answers.

(a) The reference must be both per-token and frozen. The ablations show that both properties matter. Setting γ=0 reduces BREW to TALR under the same temperature estimator and floor, dropping WB from 78.4 to 76.3 and isolating the contribution of the relative-progress term. Replacing each token's ℓ_b with the corpus-mean NLL (1.16) gives 76.6, showing that the gain does not come from the rise-then-fall shape alone. Refreshing ℓ_b every half epoch gives 76.1: because the reference moves with the student, $\rho = \ell_\theta/\ell_b$ stays near 1 and no longer records cumulative progress. Together, these results show that the gain comes from comparing each token with its own frozen starting point.

Consistently, removing the progress term increases terminal near-zero fitting by 7.5–17.3 percentage points across the base-defined bins (as shown in the training-dynamics analysis).

(b) Effective step size and training length. BREW has a mean token weight of 0.37 over training. With the default learning rate 1.5×10^{-5}, its average effective step is approximately matched by uniform SFT at 5.6×10^{-6}. SFT reaches only 58.5 WB at this learning rate, compared with 74.3 under the standard SFT setting and 78.4 for BREW.

Across one to four epochs, the best baseline reaches 76.3 WB, while BREW already reaches 77.6 after two epochs (Appendix E). These comparisons show that BREW's gain comes from reallocating updates across tokens rather than uniformly reducing the update size or training length.

(c) Role of the weight floor. Removing the weight floor lowers WB from 78.4 to 77.3. The no-floor variant still exceeds TALR (76.3), separating the additional benefit of the floor from the larger gain provided by base-relative progress. Appendix D reports sensitivity around the default $w_{\min} = 0.1$.

4.3.2 Progress-aware weighting over training

We next examine how the base-relative progress term changes the training dynamics behind this gain. BREW dynamically reallocates token weights during training: initially hard tokens receive less weight, regain weight as they become learnable, and their weights decrease after substantial relative progress. Across tokens, this state-dependent reallocation forms a moving curriculum: which tokens receive more weight changes with their training state rather than following a fixed easy-to-hard order.

To test this mechanism directly, we track token groups defined only by the frozen base model (Appendix C): tokens with $\ell_b \in [1, 3)$ are split into low- and high-entropy halves, while tokens with $\ell_b \geq 3$ form the hard tail. The effect applies broadly: 78.3% of audited positions reduce their loss by at least 50% relative to the base, and on these positions the base-relative term lowers the mean weight from 0.940 under matched confidence-only weighting to 0.401.

(a) Training budget shifts as tokens become learnable. Early in training, BREW allocates 2.5× as much relative gradient contribution per token (the per-step share of $w\|p - e_y\|$) to the intermediate bins as to the hard tail, compared with 0.77× under SFT. The hard tail receives 14.7% of the total share, versus 36.8% under SFT, and regains share as those tokens become more learnable. Thus, BREW does not permanently suppress hard targets; the allocation changes with their training state (Fig. 3A).

(b) TALR keeps increasing weight after progress; BREW reduces it. In the intermediate low-entropy bin, the fraction of tokens reaching $\rho < 0.25$ rises from 25.8% to 74.7% between steps 210 and 870. Over the same interval, TALR's mean weight rises, whereas BREW's falls from 0.436 to 0.397. TALR therefore continues to increase weight as these targets become predictable, while BREW begins to reduce it after substantial relative progress (Fig. 3B). The same difference appears in the generation-time examples below.

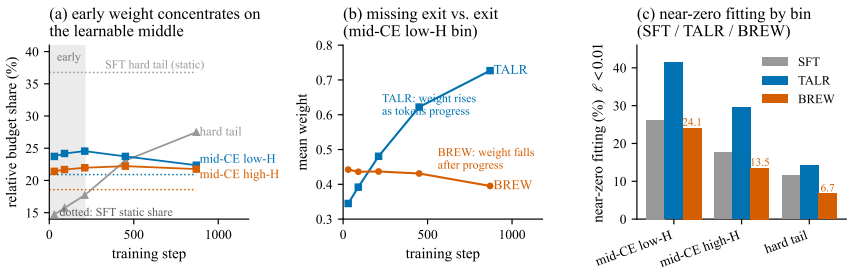

Figure 3: Progress-aware curriculum (base-defined bins; protocol in Appendix C). **A:** BREW's budget share vs. SFT's static shares (dotted). **B:** intermediate low-entropy bin; TALR's weight rises as tokens make progress, BREW's declines. **C:** terminal near-zero fitting rate (ℓ_θ<0.01) per bin; BREW lowest everywhere.

Table 3: Next-token distributions at two branching points from the same prompt ("I am creating a collection of 10 short science fiction stories..."). The shown prefixes allow multiple natural continuations. All models are evaluated on the same prefix. We report the generated token (bold) and one alternative, without renormalization; full eight-candidate results are in Appendix H.

	(a) `... stories that`		(b) `Alright,\n`	
Method	`explore`	`focus`	`the`	`let`
SFT	**0.822**	0.143	**0.621**	0.376
TALR	**0.915**	0.066	**0.931**	0.067
BREW	**0.647**	0.306	**0.559**	0.436

(c) BREW reduces near-zero fitting across difficulty levels. At the end of training, BREW has the lowest fraction of near-zero-loss tokens ($\ell_\theta < 0.01$) in every base-defined difficulty bin (Fig. 3C). TALR exceeds even SFT in the two intermediate bins, matching its rising weight curve in Fig. 3B. Appendix C reports the same comparison under alternative loss cutoffs.

4.3.3 Generation-time examples

The training dynamics above predict that reducing token weights after substantial progress should preserve more probability on alternative continuations. We examine two branching points from the same prompt, where the local context allows multiple natural continuations. Table 3 reports the next-token distributions of the three main-setting checkpoints on the same prefixes.

At both positions, all three models generate the same token, but TALR assigns it more probability than SFT. BREW assigns 0.306 to `focus` and 0.436 to `let`, compared with TALR's 0.066 and 0.067. The generated token remains the most likely choice under BREW, while the alternative receives more probability than under either baseline. These examples provide a generation-level view of the same pattern observed in the training dynamics: BREW reduces excessive concentration on a single recorded continuation while keeping the observed token as the most likely choice.

4.4 Generalization

Across models and datasets. We next evaluate BREW with a second base model and a second training corpus. Table 4 reports Qwen3-8B on DeepWriting and Qwen3.5-9B on LongWriter, using the same BREW configuration as in the main setting. The first changes the base model while keeping DeepWriting fixed; the second changes the training corpus while keeping Qwen3.5-9B fixed.

On WritingBench, BREW remains the strongest method in both settings. It reaches 75.7 on Qwen3-8B, 1.6 points above the runner-up, and 70.8 on LongWriter, a 1.1-point gain. The improvement is also broad across domains: BREW is best or tied in 11 of the 12 domain cells. The strongest baseline changes across settings, while BREW remains highest on WritingBench.

Table 4: Generalization to other models and datasets. Best per column in bold; BREW underlined where it places second.

Method	WB	LB	HB-A	HB-B	WB-A	WB-B	WB-C	WB-D	WB-E	WB-F
Qwen3-8B, DeepWriting dataset										
SFT	69.5	72.3	53.1	47.2	69.3	69.9	67.2	68.5	72.0	71.8
DFT	66.1	75.0	72.1	69.3	67.3	66.6	64.7	64.0	68.2	66.9
ASFT	73.6	80.2	**77.8**	72.4	74.8	73.4	72.6	71.9	74.7	75.6
InfoSFT	74.1	**82.7**	73.4	72.7	75.7	74.1	72.3	73.1	75.3	75.1
TALR	74.0	77.1	69.6	68.5	**76.2**	74.6	71.6	72.3	74.9	75.3
BREW	**75.7**	<u>82.1</u>	<u>76.2</u>	**75.0**	**76.2**	**75.4**	**74.6**	**75.5**	**76.7**	**76.6**
Qwen3.5-9B, LongWriter dataset										
SFT	64.2	59.8	51.7	65.6	61.4	61.5	63.4	70.4	64.7	64.4
DFT	63.9	65.8	72.4	64.5	64.8	65.4	63.8	61.7	65.1	62.4
ASFT	69.7	**80.6**	76.7	77.6	70.3	69.3	68.9	70.1	71.1	**69.3**
InfoSFT	68.3	74.8	75.0	74.9	69.1	68.6	68.1	68.6	68.3	66.3
TALR	69.1	73.5	76.8	**79.3**	69.0	69.1	69.1	69.7	70.3	66.9
BREW	**70.8**	<u>78.6</u>	**77.5**	<u>77.7</u>	**71.3**	**70.8**	**70.1**	**72.6**	**71.3**	<u>68.0</u>

On the external benchmarks, BREW has the highest judged-quality score S_q in all three writing settings (93.0, 88.3, and 91.8; Appendix K), although InfoSFT on Qwen3-8B and ASFT on LongWriter achieve higher overall LongBench-Write scores through better length compliance. On HelloBench it leads HB-B on Qwen3-8B and HB-A on LongWriter.

Appendix F extends the comparison to a larger dense model (Qwen3.5-27B) and a mixture-of-experts model (Qwen3-30B-A3B), while Appendix G evaluates Llama-3.1-8B-Instruct on LongWriter. BREW achieves the highest WritingBench score in all three additional settings. Together with Table 4, this gives six model–dataset settings in which the same configuration leads on WritingBench.

Transfer to answer-determined mathematics. We also transfer the writing configuration, without retuning, to Mistral-7B-v0.3 (Jiang et al., 2023) trained on NuminaMath-CoT-100K. This setting differs sharply from open-ended writing: many target tokens encode constrained reasoning steps or required answers, creating a different weighting environment. We use mathematics to evaluate the same weighting rule beyond open-ended writing.

At four epochs, BREW scores 28.2 against SFT's 28.4 on MATH500 (acc@1) and 11.4 against 10.1 on OlympiadBench (pass@1). All methods use the same training recipe and data budget; full results are in Appendix L.

The weight floor is especially relevant in this setting because base-unexpected tokens can contain answer-critical information. Before training, 30.7% of tokens inside `\boxed` answer spans fall at $w_{\min}$, so they retain a minimum learning signal rather than being heavily suppressed by confidence weighting. The other weighted baselines all fall below SFT on both math benchmarks, with DFT dropping by 20.6 points on MATH500 acc@1.

5 Conclusion

SFT for open-ended writing needs both an entry policy and an exit policy. Confidence-based weighting provides cautious entry; BREW adds a progress-aware exit by reducing a token's weight once its loss has fallen well below the frozen-base value. A positive floor preserves a minimum learning signal for base-unexpected tokens. BREW outperforms confidence-only weighting on WritingBench across all six model-dataset combinations and transfers competitively to answer-determined mathematics without retuning (limitations in Appendix M). The broader implication is that token weighting need not be determined only by the model's current confidence: a reference drawn from the model's own starting point gives a simple way to measure progress and to adapt token weights across different stages of learning.

AI use statement

Large language models were used as assistants for editing and polishing the text of this paper and for writing analysis and evaluation tooling; all methods, experiments, and claims were designed, executed, and verified by the authors. LLM judges used *as measurement instruments* (WritingBench Critic-7B; `gpt-4o-2024-08-06` for LongBench-Write and HelloBench) are disclosed in §4 and Appendix K.

Ethics statement

This work uses publicly released corpora, models, and benchmarks under their original licenses; no human subjects, personal data, or proprietary text are involved. LLM judges are used for automated evaluation, with model versions and evaluation protocols reported in §4 and Appendix K.

Reproducibility statement

Training hyperparameters, optimizer settings, packing protocol, checkpoint schedule, and base-NLL cache construction are reported in Appendix B; hyperparameter sensitivity in Appendix D; the mathematics protocol, grading procedure, and temperature audit in Appendix L; and judge versions, prompts, and aggregation for the external benchmarks in Appendix K.

References

Yushi Bai, Jiajie Zhang, Xin Lv, Linzhi Zheng, Siqi Zhu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongWriter: Unleashing 10,000+ Word Generation from Long Context LLMs. In *International Conference on Learning Representations (ICLR)*, 2025.

Yoshua Bengio, Jason Louradour, Ronan Collobert, and Jason Weston. Curriculum Learning. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, pp. 41–48, 2009.

John Joon Young Chung, Vishakh Padmakumar, Melissa Roemmele, Yuqian Sun, and Max Kreminski. Modifying Large Language Model Post-Training for Diverse Creative Writing. *arXiv preprint arXiv:2503.17126*, 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*, 2024.

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 Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.211. URL `https://aclanthology.org/2024.acl-long.211/`.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.

Liwei Jiang, Yuanjun Chai, Margaret Li, Mickel Liu, Raymond Fok, Nouha Dziri, Yulia Tsvetkov, Maarten Sap, and Yejin Choi. Artificial Hivemind: The Open-Ended Homogeneity of Language Models (and Beyond). In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS D&B)*, 2025.

Constantinos Karouzos, Xingwei Tan, and Nikolaos Aletras. Where does output diversity collapse in post-training? *arXiv preprint arXiv:2604.16027*, 2026.

Roman Klypa and Oleksandr Cherednichenko. Diversity in Large Language Models under Supervised Fine-Tuning. *arXiv preprint arXiv:2605.00195*, 2026.

M. Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-Paced Learning for Latent Variable Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 23, pp. 1189–1197, 2010.

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 (ICLR)*, 2025.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's Verify Step by Step. In *International Conference on Learning Representations (ICLR)*, 2024.

Jiacheng Lin, Zhongruo Wang, Kun Qian, Tian Wang, Arvind Srinivasan, Hansi Zeng, Ruochen Jiao, Xie Zhou, Jiri Gesi, Dakuo Wang, et al. SFT Doesn't Always Hurt General Capabilities: Revisiting Domain-Specific Fine-Tuning in LLMs. In *International Conference on Learning Representations (ICLR)*, 2026.

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Yelong Shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, et al. Rho-1: Not All Tokens Are What You Need. *arXiv preprint arXiv:2404.07965*, 2024.

Ximing Lu, Melanie Sclar, Skyler Hallinan, Niloofar Mireshghallah, Jiacheng Liu, Seungju Han, Allyson Ettinger, Liwei Jiang, Khyathi Chandu, Nouha Dziri, et al. AI as Humanity's Salieri: Quantifying Linguistic Creativity of Language Models via Systematic Attribution of Machine Text against Web Text. In *International Conference on Learning Representations (ICLR)*, 2025.

Sören Mindermann, Jan Brauner, Muhammed Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N. Gomez, Adrien Morisot, Sebastian Farquhar, et al. Prioritized Training on Points that are Learnable, Worth Learning, and Not Yet Learnt. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, pp. 15630–15649, 2022.

Samuel Paech, Allen Roush, Judah Goldfeder, and Ravid Shwartz-Ziv. Antislop: A Comprehensive Framework for Identifying and Eliminating Repetitive Patterns in Language Models. In *International Conference on Learning Representations (ICLR)*, 2026.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabás Póczos, and Tom M. Mitchell. Competence-Based Curriculum Learning for Neural Machine Translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 1162–1172, 2019. doi: 10.18653/v1/N19-1119.

Xiaohan Qin, Xiaoxing Wang, Ning Liao, Cancheng Zhang, Xiangdong Zhang, Mingquan Feng, Jingzhi Wang, and Junchi Yan. ssToken: Self-modulated and Semantic-aware Token Selection for LLM Fine-tuning. In *International Conference on Learning Representations (ICLR)*, 2026.

Liu Qing, Ou Wu, and Yi Du. AlphaToken: Decoupling Adaptation and Stability for Path-Aware Response Token Valuation in LLM Post-Training. *arXiv preprint arXiv:2606.01635*, 2026.

Haoran Que, Feiyu Duan, Liqun He, Yutao Mou, Wangchunshu Zhou, Jiaheng Liu, Wenge Rong, Zekun Moore Wang, Jian Yang, Ge Zhang, et al. HelloBench: Evaluating Long Text Generation Capabilities of Large Language Models. *arXiv preprint arXiv:2409.16191*, 2024.

Mahdi Sabbaghi, George Pappas, Adel Javanmard, and Hamed Hassani. InfoSFT: Learn More and Forget Less with Information-Aware Token Weighting. *arXiv preprint arXiv:2605.14967*, 2026.

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 (ICML)*, 2025.

Chantal Shaib, Tuhin Chakrabarty, Diego Garcia-Olano, and Byron C. Wallace. Measuring AI "Slop" in Text. In *Third Conference on Language Modeling (COLM)*, 2026. URL `https://openreview.net/forum?id=UMipfwK6Ar`.

Zhanming Shen, Zeyu Qin, Jiaqi Hu, Wentao Ye, Hao Chen, Xiaomeng Hu, Haokai Xu, Gang Chen, Yi R. Fung, and Haobo Wang. Supervised Fine-Tuning Needs to Unlock the Potential of Token Priority. *arXiv preprint arXiv:2602.01227*, 2026.

Haozhe Wang, Haoran Que, Qixin Xu, Minghao Liu, Wangchunshu Zhou, Jiazhan Feng, Wanjun Zhong, Wei Ye, Tong Yang, Wenhao Huang, Ge Zhang, and Fangzhen Lin. Reverse-Engineered Reasoning for Open-Ended Generation. In *International Conference on Learning Representations (ICLR)*, 2026a.

Ke Wang, Shuangqi Li, Mathieu Salzmann, and Pascal Frossard. PriFT: Prior-Support Guided Supervised Fine-Tuning. *arXiv preprint arXiv:2606.09396*, 2026b.

Xinda Wang, Zhengxu Hou, Yangshijie Zhang, Bingren Yan, Jialin Liu, Chenzhuo Zhao, Zhibo Yang, Bin-Bin Yang, and Feng Xiao. Triviality Corrected Endogenous Reward. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19334–19355, 2026c. doi: 10.18653/v1/2026.acl-long.883.

Yueyang Wang, Baolong Bi, Shuo Lu, Jingyuan Zhang, and Jiajun Shi. LP-SFT: Local-Preserving Supervised Fine-Tuning via Multimodal Entropy Structure. *arXiv preprint arXiv:2607.04733*, 2026d.

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 (ICLR)*, 2026.

Yuning Wu, Jiahao Mei, Ming Yan, Chenliang Li, Shaopeng Lai, Yuran Ren, Zijia Wang, Ji Zhang, Mengyue Wu, Qin Jin, et al. WritingBench: A Comprehensive Benchmark for Generative Writing. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025.

Tong Xie, Yuanhao Ban, Yunqi Hong, Sohyun An, Yihang Chen, and Cho-Jui Hsieh. A Unifying Lens on Supervised Fine-Tuning Through Target Distribution Design. *arXiv preprint arXiv:2606.11189*, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, 2025.

Jiayi Zhang, Simon Yu, Derek Chong, Anthony Sicilia, Michael R. Tomz, Christopher D. Manning, and Weiyan Shi. Verbalized Sampling: How to Mitigate Mode Collapse and Unlock LLM Diversity. In *International Conference on Machine Learning (ICML)*, 2026. URL `https://openreview.net/forum?id=zloIrd77G5`.

Yiming Zhang, Harshita Diddee, Susan Holm, Hanchen Liu, Xinyue Liu, Vinay Samuel, Barry Wang, and Daphne Ippolito. NoveltyBench: Evaluating Language Models for Humanlike Diversity. In *Second Conference on Language Modeling (COLM)*, 2025. URL `https://openreview.net/forum?id=XZm1ekzERf`.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-Following Evaluation for Large Language Models. *arXiv preprint arXiv:2311.07911*, 2023.

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 (ICLR)*, 2026a.

Wenhong Zhu, Ruobing Xie, Rui Wang, Xingwu Sun, Di Wang, and Pengfei Liu. Proximal Supervised Fine-Tuning. In *International Conference on Learning Representations (ICLR)*, 2026b.

A Theory: statements and proofs

Throughout, ℓ_θ is the student CE on the gold token (nats), $q = e^{-\ell_\theta}$, ℓ_b the frozen-base CE, $\delta_+ = \max(\ell_b - \ell_\theta, 0)$, and $\rho = \ell_\theta/\ell_b$ the relative-loss ratio (Eq. 1). Assumptions: (A1) improvement domain $\delta > 0$ unless stated; (A2) floor inactive unless stated; (A3) frozen reference; (A4) decoupled per-token dynamics $d\ell/dt = -\eta w(\ell)G(\ell)$; and (A5) $G = \Theta(1)$ for $\ell \gtrsim 1$.

A.1 Proof of Proposition 1 (rise-then-fall)

From Eq. 1, $d\ell_{\text{eff}}/d\ell_\theta = 1 - \gamma/\ell_\theta$: negative for $\ell_\theta < \gamma$, positive for $\ell_\theta > \gamma$, so ℓ_{eff} has a unique minimum at $\ell_\theta = \gamma$ with $\ell^*_{\text{eff}} = \gamma(1 + \log(\ell_b/\gamma))$ and $w_{\max} = e^{-\ell^*_{\text{eff}}/\tau} = (\gamma/\ell_b)^{\gamma/\tau} e^{-\gamma/\tau}$. □

The peak location is independent of ℓ_b, τ (but requires $\ell_b > \gamma$; for $\ell_b \le \gamma$ the unfloored weight increases monotonically on $(0, \ell_b)$ and the supremum is at $\ell_\theta \to \ell_b^-$); the peak height decays as $\ell_b^{-\gamma/\tau}$. Empirical check (400 samples / 1.15M tokens): the empirical $\arg\max$ falls in the 0.45–0.60 bin for both ℓ_b bands, and the higher band peaks lower (0.548 vs. 0.489), in the predicted direction.

Proposition 2 (Opposite asymptotic drift under memorization)**.** In the memorization limit $\ell_\theta \to 0^+$ of a learned token, TALR's statistic $\ell^{\text{TALR}}_{\text{eff}} = \ell_\theta \to 0$ pulls the batch statistic toward 0, sharpening the contrast between low- and high-loss tokens (approaching a binary gate only in the limit $\tau \to 0$); under Eq. 1, $\ell_{\text{eff}} = \ell_\theta + \gamma \log(\ell_b/\ell_\theta) \to +\infty$, so learned tokens push the token-level statistic *up*. If more than half of the sequences in a batch have their mean ℓ_{eff} dominated by such drift (e.g., because memorized content is not confined to a minority of sequences), the sequence median rises with training; otherwise its trajectory is not fixed by the token-level result alone.

A.2 Proof of Proposition 2 (opposite asymptotic drift)

For TALR the token-level statistic is ℓ_θ, so a memorized token satisfies $\ell_\theta \to 0$. Under Eq. 1, by contrast, $\ell_{\text{eff}} = \ell_\theta + \gamma \log(\ell_b/\ell_\theta) \to +\infty$ as $\ell_\theta \to 0^+$ for fixed $\ell_b > 0$. Thus the two token-level statistics have opposite asymptotic directions. Translating this result to the sequence median requires the distributional condition stated in Proposition 2: the median rises only when more than half of the sequences have mean effective losses dominated by this drift. □

Empirically, this condition is consistent with the main run: across five matched checkpoints, TALR's temperature decreases from 1.30 to 0.64, while BREW's temperature has an overall upward trend from 2.24 to 2.97.

Proposition 3 (Dead-zone escape dichotomy)**.** Model a base-unknown token by $d\ell/dt = -\eta\, w(\ell)\, G(\ell)$ with $G = \Theta(1)$ on $\ell \ge 1$ and initial surprisal ℓ_0. For any rule with $w(\ell) \le e^{-\ell}\,\text{poly}(\ell)$ (DFT, InfoSFT small-q), the escape time grows as $e^{\ell_0}/\text{poly}(\ell_0)$, so the reachable set within budget ηT satisfies $\ell_0 \lesssim \log(\eta T) + O(\log\log)$. With a floor $w \ge w_{\min}$, escape time is at most $(\ell_0 - \ell_1)/(w_{\min}\,\eta\,\Theta(1))$: reachability is linear, $\ell_0 \lesssim w_{\min}\,\eta T + \ell_1$.

A.3 Proof of Proposition 3 (escape dichotomy)

With (A4)/(A5), the time to move from ℓ_0 down to ℓ_1 is $T_{\text{esc}}[w] = \eta^{-1} \int_{\ell_1}^{\ell_0} d\ell/(w(\ell)G(\ell))$. For $w(\ell) = e^{-\ell}$ (DFT): $T = \Theta(e^{\ell_0})$. For InfoSFT's small-q form $w \approx e^{-\ell}(\ell + c)$, $c = \text{logit}(\bar{p})$: $T \approx e^{\ell_0}/(\ell_0 + c)$ by Laplace asymptotics (numerically 936 vs. asymptotic 847 at $\ell_0 = 9.21$, i.e. $q_b = 10^{-4}$). For tempered $w = e^{-\ell/\tau}$: $T = \tau(e^{\ell_0/\tau} - e^{\ell_1/\tau})$, still exponential with rate $1/\tau$; a decreasing τ (Prop. 2 for TALR) can steepen this raw confidence weighting, with the binary-gate regime reserved for the theoretical limit $\tau \to 0$. For $w \ge w_{\min}$: $T \le (\ell_0 - \ell_1)/(w_{\min}\,\eta\,\min_{[\ell_1,\ell_0]} G)$. Inverting each T at budget ηT gives the stated reachability classes. □

The floor therefore changes the escape-time scaling from exponential to linear for low-confidence tokens, even when it binds on only a minority of observed tokens. Appendix B quantifies how often the floor applies in the training data.

B Implementation and protocol details

Training. All runs are full-model SFT in bf16 with FlashAttention and ZeRO-3, AdamW (β_2=0.95, weight decay 0.01, gradient clip 1.0), learning rate 1.5×10^{-5} under one fixed cosine schedule with 10% warmup, padding-free packing to length 17,670 with position-id-aware sequence boundaries (attention never crosses document boundaries), 16 GPUs × per-device batch 1 × accumulation 1. Each DeepWriting comparison is run with training seeds 1–3; reported scores are averages over the three runs. Every method in a given setting shares the same data order and schedule and is trained for four epochs, the comparison endpoint used throughout (DeepWriting, 9B: step 870; DeepWriting,

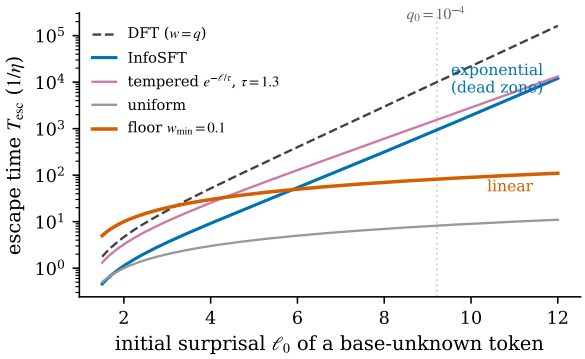

Figure 4: Dead-zone escape dichotomy (Prop. 3; $G{\equiv}1$, with the target level set to $\ell_1{=}1$, i.e. $q_b = e^{-1} \approx 0.37$): the time to learn a base-unknown token grows exponentially in its initial surprisal ℓ_0 under every confidence-only family, and linearly under a weight floor.

8B: step 908; LongWriter: step 390); no per-method schedule tuning or checkpoint selection is performed.

Matched-step-size control. Uniform SFT was also trained at 5.6×10^{-6}, the learning rate that matches BREW's run-average token weight of 0.37 (§4.3). It reaches 58.5 WB, with a think-segment closure rate of 68.8% and a median response length of approximately 21,000 characters.

BREW. Frozen-base NLLs are computed once offline and cached (training adds no second forward); $\gamma{=}0.75$, $w_{\min}{=}0.1$, the log-ratio applied without truncation at $\rho{=}1$, τ = sequence median of ℓ_{eff} (median over sequences of the per-sequence mean, TALR's convention). On DeepWriting, the cache contains 61.6M float32 token NLLs across 3,488 packed streams and occupies 246 MB. A batch-1 SDPA measurement on one MI308X processes 2.04M tokens in 6.4 minutes ($\sim$ 5,300 tokens/s), corresponding to approximately 3.2 single-GPU hours for the full cache, or about 12 minutes when amortized across the 16-GPU training allocation. This is a one-time preprocessing cost; subsequent epochs reuse the cache. The weights and the temperature are computed with stop-gradient and treated as constants during backpropagation, matching the decoupled dynamics of the analysis; and ℓ_{eff} is floored at 0 in Eq. 1 before entering the temperature statistic and Eq. 3, so $w \in [w_{\min}, 1]$ and τ's argument is nonnegative. The initialization value is $\tau = 1.21$; it rises to 2.24 by the earliest measured checkpoint at step 30 and then follows an overall upward, though non-monotone, trajectory of 2.24/2.19/2.22/2.66/2.97 through step 870. Mechanism probes replay the production $\gamma{=}0.75$ checkpoints; the math setting uses $\gamma{=}0.75$ at the writing default $w_{\min}{=}0.1$ (Appendix L).

Floor audit. Proposition 3 gives different low-confidence escape-time scaling with and without a positive weight floor. We measure how often the floor matters in DeepWriting using 577,067 supervised positions and their cached base-model losses. Write $q_b \equiv e^{-\ell_b}$ for the frozen base's probability of the observed token. Before training we set $\ell_\theta = \ell_b$ (with $\tau = 1$) and evaluate the initial weight assigned by each rule, taking InfoSFT at its default $\bar{p} = 0.93$. Among the 6.6% of positions with $q_b < 0.01$ (equivalently $\ell_b > 4.61$), BREW assigns the floor $w_{\min} = 0.1$ to every token, while InfoSFT assigns a mean weight of 0.0092 on the same positions, a 10.9$\times$ difference in gradient budget. Confidence-weighted objectives can therefore assign very small weights to unfamiliar markup such as <think> (Sabbaghi et al., 2026). On the main writing setting, removing the floor lowers WB by 1.1 points (Table 2).

Baselines. DFT via the trainer's native per-token weight $w{=}q$; ASFT with KL coefficient 0.03; TALR with its sequence-median temperature (faithful reimplementation); InfoSFT with the authors' recommended $\bar{p}{=}0.93$; all under the identical optimizer, packing, and budget.

Generation and scoring. One shared protocol for WritingBench, LongBench-Write, and HelloBench: temperature 0.7, top-p 0.8, top-k 20, repetition penalty 1.2, max 16,000 new tokens; think-on for the DeepWriting settings and think-off for LongWriter, with think segments stripped before

scoring. WritingBench uses the official Critic-7B rubric scorer (n=1000 prompts); instruction-following retention uses the official IFEval evaluator.

Data provenance. The DeepWriting dataset (Wang et al., 2026a) contributes 20,395 usable long-form samples with reverse-engineered reasoning traces and think-format supervision. The LongWriter dataset (Bai et al., 2025) contributes 5,993 usable samples of mixed Chinese–English long-form instructions in no-think format. Both are used as distributed by their authors; no additional filtering or modification is applied.

C Additional mechanism evidence

This appendix gives the frequency-bin view, the bin definitions and statistics behind Fig. 3, and the temperature trajectories.

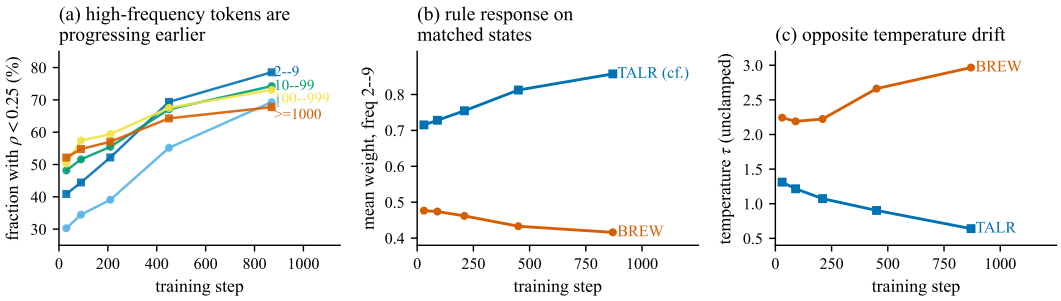

Figure 5: Frequency-bin view of the mechanism (full details in Appendix C). **A:** higher-frequency 4-grams reach $\rho < 0.25$ (a fourfold surprisal reduction) earlier, although frequency is not itself an input to BREW. **B:** on the largest bin (freq 2–9), BREW's mean weight declines as those tokens make progress while the TALR rule applied to the same states (counterfactual) rises; the same qualitative direction holds in all five frequency-defined bins. **C:** opposite temperature drift. Both methods use the identical sequence-mean-then-median temperature; only the input statistic differs.

Table 5: Terminal fitting (%) per base-defined bin at three cutoffs on ℓ_θ. Intermediate bin is $\ell_b \in [1, 3)$ split at the band's base-entropy median; hard tail is $\ell_b \geq 3$.

	$\ell_\theta < 0.01$ (convention)			$\ell_\theta < 0.001$ (stricter)			$\ell_\theta < 0.1$ (looser)		
Cohort	SFT	TALR	BREW	SFT	TALR	BREW	SFT	TALR	BREW
mid-ℓ_b, low ent. (n=193,762)	26.2	41.4	24.1	13.5	18.4	7.8	47.2	64.8	54.3
mid-ℓ_b, high ent. (n=193,762)	17.6	29.6	13.5	6.8	10.5	2.8	39.1	56.0	44.2
hard tail, $\ell_b \geq 3$ (n=303,390)	11.6	14.2	6.7	4.4	5.9	1.9	27.6	25.4	20.0

Replay protocol. All curves are replayed from checkpoints of the main BREW run (training seed 1) using the cached frozen-base NLLs; no additional training is involved. We track the first 128 packed sequences of the DeepWriting training order, packed with the same greedy rule as training (cap 17,670 tokens), comprising 2,038,120 supervised positions, and evaluate five approximately log-spaced checkpoints of the 1,308-step run at steps 30/90/210/450/870. Temperatures use the same sequence-mean-then-median statistic as in training, applied to ℓ_{eff} (BREW) or ℓ_θ (TALR); because position ids reset at every training sample inside a packed sequence, the "sequence" unit throughout is the training sample. Confidence intervals come from 200 bootstrap resamples over that same unit (analysis seed 0).

Cohort definitions. The Fig. 3 bins are defined from the base state alone, so membership cannot be an effect of training. The intermediate band $\ell_b \in [1, 3)$ collects positions the base finds neither easy nor near-impossible (q_b between 5% and 37%) and is split at the band's median base entropy (2.02 nats) into low- and high-entropy halves; base entropy serves as a model-relative proxy for local

underdetermination. Positions with $\ell_b \geq 3$ ($q_b < 5\%$) form the hard tail. Positions with $\ell_b < 1$ ($q_b > 37\%$) have limited room for further loss reduction and are omitted from the three plotted bins, but they remain in the denominator of the globally normalized budget shares, so those three curves do not sum to 100%. The frequency bins of Fig. 5 are likewise fixed *pre-training*: we count every token 4-gram (a conventional order) over the full DeepWriting dataset as rendered by the Qwen3.5-9B chat template, assign each tracked position its corpus count, and bucket by decade: $\{1\}$ (n=516,699), $\{2\text{–}9\}$ (1,127,491), $\{10\text{–}99\}$ (227,596), $\{100\text{–}999\}$ (82,150), $\{\geq 1000\}$ (82,317).

Budget share. Relative budget share uses $w \cdot \|p - e_y\|$, with the gradient-norm proxy evaluated at the base state ($\|p - e_y\|^2 = \sum_v p_v^2 + 1 - 2p_y$) and normalized per step; the SFT reference uses $w \equiv 1$.

Thresholds. We use $\rho < 0.25$ to denote a fourfold reduction in surprisal relative to the frozen base: the student retains at most a quarter of the base's surprisal on that token. *Near-zero fitting* uses $\ell_\theta < 0.01$ ($q > 0.990$); Table 5 also reports the stricter (0.001, $q > 0.999$) and looser (0.1) cutoffs. Under the stricter cutoff the ordering is unchanged in every bin. At the looser cutoff $\ell_\theta < 0.1$, BREW exceeds SFT in the two intermediate bins while remaining lowest on the hard tail. Together with the stricter cutoffs, this shows that BREW places more intermediate tokens in the well-fitted range while fewer reach near-deterministic loss. The writing-quality results in §4 report the corresponding generation performance. *The within-band entropy split* is the median; moving it to the 40th or 30th percentile leaves the entropy interaction's sign positive below ℓ_b=3 and negative above it.

TALR comparisons. In Fig. 5B, the TALR curve is counterfactual: we apply the TALR weighting rule to BREW's training states, using the same ℓ_θ values and a TALR temperature computed from those states. This compares the two weighting rules at matched model states. In Fig. 3B, we instead replay TALR's own checkpoints (steps $30/90/210/450/870$) on the same base-defined mid-ℓ_b low-entropy bin and compare its observed weight trajectory with BREW's.

D Hyperparameter sensitivity

We vary the progress-term strength γ and the floor $w_{\min}$ around the operating point (γ=0.75, $w_{\min}$=0.1) on the main setting (Qwen3.5-9B with DeepWriting, 4-epoch point, think-on protocol, one Critic-7B scoring pass each).

Table 6: Sensitivity of WritingBench and instruction-following retention to γ and $w_{\min}$, in the metric layout of Table 1 (WB = overall, WB-A–F = domain scores; IFEval as in Table 11). The center row is the main-table operating point.

	WB	WB-A	WB-B	WB-C	WB-D	WB-E	WB-F	IFEval
$\gamma = 0.5$	77.9	78.5	78.5	75.7	78.1	**78.9**	78.5	78.93
$\gamma = 0.75$ (center)	**78.4**	**79.1**	**78.9**	**76.7**	**78.6**	78.5	**79.1**	80.22
$\gamma = 1.5$	77.7	78.6	77.7	76.1	77.8	78.6	78.5	**82.44**
$w_{\min} = 0$ (no floor)	77.3	78.1	78.2	75.2	77.5	77.4	77.9	**80.41**
$w_{\min} = 0.01$	77.7	78.2	77.8	76.0	77.8	78.3	78.8	78.00
$w_{\min} = 0.1$ (center)	**78.4**	**79.1**	**78.9**	**76.7**	**78.6**	**78.5**	**79.1**	80.22
$w_{\min} = 0.36$	78.0	78.3	78.8	76.3	78.2	77.9	78.5	80.22

Performance is stable around the default setting. Varying γ from 0.5 to 1.5 changes WB by at most 0.7 points, and varying $w_{\min}$ from 0.01 to 0.36 changes it by at most 0.7 points. Every tested setting remains above the strongest baseline (76.3 WB), and the domain-level results show the same overall stability.

The two hyperparameters affect writing quality and instruction following differently. Increasing γ from 0.75 to 1.5 lowers WB by 0.7 points while increasing IFEval by 2.2 points. For the floor, WB rises from 77.3 at $w_{\min} = 0$ to 78.4 at $w_{\min} = 0.1$, then remains close at $w_{\min} = 0.36$ (78.0). We use (γ=0.75, $w_{\min}$=0.1) in the main experiments; γ=0.75 places the unfloored weight peak at $q = e^{-0.75} \approx 0.47$, and $w_{\min}$=0.1 provides a small but nonzero minimum weight.

E Training-budget trajectory

All methods are compared at four epochs (§4). We evaluate all methods at each epoch boundary to compare their performance over training, under the identical generation and scoring protocol.

BREW reaches 77.6 WB after two epochs, already above the best score achieved by any baseline at any budget (TALR, 76.3 at four epochs; SFT, 75.9 at three). It also has the highest overall WB at each of the four epoch budgets.

The trajectories also differ across methods. SFT improves from 73.1 at two epochs to 75.9 at three, then drops to 74.3 at four; ASFT shows a similar decline from three to four epochs (73.2 → 72.7). TALR changes little over the same interval (76.2 → 76.3), while BREW rises from 76.8 to 78.4. The BREW–TALR gap remains positive at all four budgets ($+2.0/+3.9/+0.6/+2.1$).

Table 7: WritingBench across four training budgets (Qwen3.5-9B with the DeepWriting dataset, n=1000 per cell); columns are WB overall and the six domains A–F of Table 1. Bold marks the per-column best within each budget block.

Budget	Method	WB	A	B	C	D	E	F
1 epoch	SFT	65.4	57.7	63.9	64.0	70.4	66.6	70.2
	DFT	59.8	62.2	63.7	58.7	51.6	61.0	62.6
	ASFT	68.1	69.9	69.2	66.5	66.1	68.8	68.1
	InfoSFT	73.1	**75.8**	**74.0**	71.7	71.0	**74.1**	72.7
	TALR	72.0	73.2	72.9	71.6	70.6	70.9	72.7
	BREW	**74.0**	75.2	**74.0**	**72.8**	**74.6**	73.5	**74.1**
2 epochs	SFT	73.1	69.1	74.3	70.8	75.0	75.8	75.0
	DFT	63.2	66.1	66.1	63.3	53.2	66.0	66.0
	ASFT	72.4	73.1	72.7	70.3	71.7	74.8	73.3
	InfoSFT	73.4	74.7	74.0	71.8	71.9	73.9	74.8
	TALR	73.7	74.5	73.7	71.4	74.3	74.0	75.2
	BREW	**77.6**	**78.0**	**77.6**	**76.2**	**78.1**	**78.7**	**77.9**
3 epochs	SFT	75.9	74.6	76.5	74.3	76.1	77.9	77.0
	DFT	63.3	64.9	66.3	62.4	58.5	65.2	63.3
	ASFT	73.2	75.3	73.6	71.4	72.5	73.7	73.1
	InfoSFT	73.9	75.1	74.8	72.0	72.6	75.2	74.2
	TALR	76.2	76.6	76.4	74.1	**76.8**	76.9	**77.6**
	BREW	**76.8**	**78.4**	**76.6**	**74.9**	76.6	**78.2**	77.1
4 epochs	SFT	74.3	72.2	75.4	71.4	75.5	75.4	76.9
	DFT	65.6	68.5	66.5	64.0	61.2	67.8	67.3
	ASFT	72.7	73.8	73.6	71.4	71.6	73.9	72.6
	InfoSFT	75.7	76.2	76.9	74.3	74.6	77.0	76.0
	TALR	76.3	77.4	76.8	74.3	75.7	76.7	77.5
	BREW	**78.4**	**79.1**	**78.9**	**76.7**	**78.6**	**78.5**	**79.1**

At two and four epochs, BREW has the highest score in all six WritingBench domains (Table 7). At one epoch, InfoSFT is higher in Academic & Engineering and Education and ties BREW on Finance & Business; at three epochs TALR is slightly higher on Literature & Arts and Advertising & Marketing by 0.2 and 0.5. These differences disappear by four epochs.

Politics & Law is BREW's lowest-scoring domain at every budget, although it still achieves the highest score there among the compared methods. DFT shows a different pattern: Literature & Arts is consistently its weakest domain, trailing its average over the other five domains by 5.6–12.3 points.

F Additional model families: 27B dense and 30B-A3B MoE

Beyond the transfer settings of §4.4, we evaluate BREW on a larger dense model (Qwen3.5-27B) and a mixture-of-experts model (Qwen3-30B-A3B), both trained for two epochs under the configuration of §3.1, think-on, scored by Critic-7B. The MoE setting runs the full baseline set; the 27B setting

compares against uniform SFT and TALR, the strongest baseline in the main setting. Table 8 reports overall WritingBench and the six domains.

Table 8: Two additional model families, trained for two epochs. Best per column in bold. BREW is first in every column on both bases.

Base	Method	WB	A	B	C	D	E	F
Qwen3-30B-A3B (MoE)	SFT	63.97	64.9	63.9	61.8	62.4	66.5	66.4
	DFT	63.47	65.4	65.7	63.0	57.7	63.1	66.6
	ASFT	72.54	73.9	72.0	70.5	72.5	73.7	73.9
	InfoSFT	73.77	75.1	74.1	71.8	73.2	74.6	74.6
	TALR	74.01	75.2	74.4	71.3	74.8	74.4	74.6
	BREW	**75.63**	**77.1**	**75.7**	**74.0**	**75.4**	**76.8**	**75.5**
Qwen3.5-27B (dense)	SFT	71.76	66.7	71.2	71.5	76.6	69.8	74.4
	TALR	76.82	77.9	76.3	74.6	78.5	76.4	77.7
	BREW	**78.38**	**78.7**	**77.8**	**76.1**	**79.9**	**79.3**	**79.6**

BREW achieves the highest WritingBench score on both additional models, with TALR second in each setting. The gain over uniform SFT is $+11.7$ points on Qwen3-30B-A3B and $+6.6$ on Qwen3.5-27B, compared with $+4.1$ in the main 9B setting. The improvement is broad across domains: BREW exceeds SFT by 9–13 points in all six MoE domains and by 3–12 points across the six 27B domains.

G A non-Qwen base: Llama-3.1-8B-Instruct on LongWriter

We ran the writing baselines on Llama-3.1-8B-Instruct (Grattafiori et al., 2024) with the LongWriter dataset for four epochs under the configuration of §3.1, no-think protocol, scored by Critic-7B.

Table 9: Llama-3.1-8B-Instruct with the LongWriter dataset, four-epoch endpoint. Best per column in bold.

Method	WB	WB-A	WB-B	WB-C	WB-D	WB-E	WB-F
SFT	63.55	65.47	64.77	61.45	63.33	64.23	62.02
ASFT	61.87	64.22	61.95	59.96	59.99	64.09	62.38
InfoSFT	64.92	67.55	66.21	63.95	63.46	64.79	63.05
TALR	66.79	69.40	67.34	65.65	65.70	**66.67**	65.92
BREW	**67.52**	**70.08**	**68.20**	**66.91**	**66.77**	66.04	**66.37**

BREW leads overall and in five of the six WritingBench domains; TALR is higher only on Education, by 0.6 points.

H Full next-token distributions

Table 10 reports probabilities for the base model's top-eight candidates at both branching points in §4.3.3.

I Judge-free writing diagnostics

Figure 6 compares output length and judge-free diagnostics on the main-setting generations. BREW produces responses at roughly half of SFT's mean length while scoring higher on WritingBench. Its gains are positive across the reported criterion groups, with larger improvements on the creative subset. Its template mass is $10\times$ lower than SFT's (Shaib et al., 2026; Paech et al., 2026), its distinct-4 is among the highest, and anchored novelty (the share of response 5-grams unseen in the training corpus, weighted by prompt-content coverage) remains similar across methods.

Table 10: Next-token distributions at two branching points for the three main-setting checkpoints. All models are evaluated on the same prefix. We report probabilities for the base model's top-8 candidates, ordered by SFT probability; bold marks the generated token.

(a) Prefix: `... stories that` [next: `explore`]

Model	explore	focus	blend	delve	rev	each	dive	highlight
SFT	**0.822**	0.143	0.013	0.015	0.003	0.002	0.001	0.000
TALR	**0.915**	0.066	0.013	0.001	0.001	0.002	0.000	0.000
BREW	**0.647**	0.306	0.037	0.004	0.003	0.001	0.000	0.000

(b) Prefix: `Alright,\n` [next: `the`]

Model	the	let	I	so	maybe	we	less	this
SFT	**0.621**	0.376	0.001	0.001	0.000	0.000	0.000	0.000
TALR	**0.931**	0.067	0.001	0.000	0.000	0.000	0.000	0.000
BREW	**0.559**	0.436	0.004	0.001	0.000	0.000	0.000	0.000

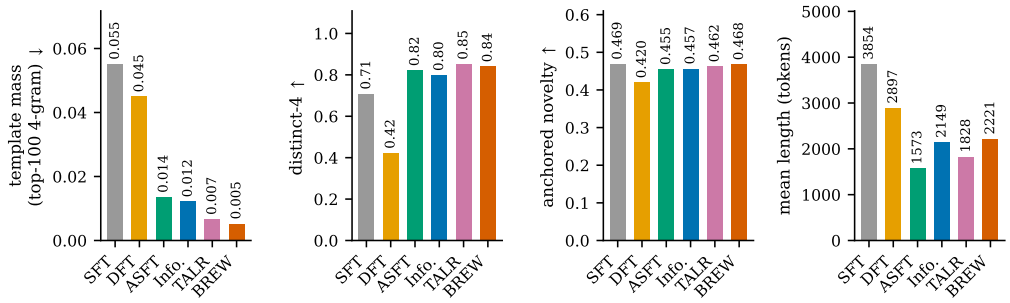

Figure 6: Judge-free diagnostics on the main-setting generations (n=1000; same outputs as Table 1). BREW has the lowest template mass, distinct-4 among the highest, and anchored novelty comparable to the other methods; its mean output length is roughly half of SFT's. DFT has the highest template mass among the weighted methods and the lowest distinct-4 and anchored novelty.

J Capability retention: instruction following

All methods are evaluated at the same checkpoints with the official IFEval strict-prompt evaluator.

Table 11: IFEval retention (same checkpoints as Table 1).

Method	IFEval ↑
base (no SFT)	**84.84**
SFT	80.41
DFT	73.57
ASFT	77.63
InfoSFT	77.45
TALR	81.33
BREW (ours)	80.22

K External benchmark protocols and judge analysis

We evaluate all eighteen writing checkpoints on two additional long-form benchmarks, following the DeepWriter/TCER benchmark suite (Wang et al., 2026c). LongBench-Write (Bai et al., 2025) contains 120 length-constrained prompts and reports $\bar{S} = (S_l + S_q)/2$, where S_l measures length compliance and S_q is the six-dimension judged-quality score. HelloBench (Que et al., 2024)

contributes the Open-Ended QA subset (HB-A, 200 prompts) and Heuristic Text Generation subset (HB-B, 123 prompts, scored by HelloEval's regression-weighted checklist).

Generation follows the same protocol as WritingBench: think-on for the DeepWriting settings and think-off for LongWriter, with think segments removed before scoring. The judges originally specified by these benchmarks (`gpt-4o-2024-05-13` and `gpt-4o`) are no longer available, so we use `gpt-4o-2024-08-06` for every system with the official prompts and aggregation. Each output is judged once.

Tables 1 and 4 report the overall LB and HB scores; Table 12 below gives the LongBench-Write S_l/S_q decomposition.

Table 12: LongBench-Write S_l/S_q decomposition on the three writing settings (same checkpoints as Tables 1 and 4). Bold marks the per-column best; BREW underlined where it places second.

Method	LB S_l	LB S_q
Qwen3.5-9B, DeepWriting dataset		
SFT	68.6	77.6
DFT	67.6	83.9
ASFT	68.7	87.5
InfoSFT	76.5	86.2
TALR	76.8	89.9
BREW	**81.3**	**93.0**
Qwen3-8B, DeepWriting dataset		
SFT	70.7	73.9
DFT	69.8	80.3
ASFT	74.0	86.5
InfoSFT	**79.4**	85.9
TALR	74.4	79.8
BREW	75.9	**88.3**
Qwen3.5-9B, LongWriter dataset		
SFT	48.8	70.7
DFT	53.4	78.3
ASFT	**73.2**	87.9
InfoSFT	62.1	87.5
TALR	60.0	86.9
BREW	65.4	**91.8**

In the two transfer settings where BREW does not lead the overall LB score, the difference comes from S_l: InfoSFT has the highest length-compliance score on Qwen3-8B (79.4), and ASFT on LongWriter (73.2). BREW has the highest judged-quality score S_q in all three writing settings (93.0, 88.3, and 91.8).

Results across judge families. WritingBench uses its dedicated Critic-7B rubric scorer, whereas LongBench-Write and HelloBench use `gpt-4o-2024-08-06`. In the main setting, BREW has the highest score on all four judged metrics (WB, LB, HB-A, and HB-B). Across the three model–dataset settings, it ranks first on 8 of the 12 judged metrics and second on the remaining four (Tables 1 and 4).

The judge-free diagnostics of Appendix I provide a separate view of the same generations, including output length, template mass, distinct-n, and anchored novelty.

L Math benchmarks: protocol and full results

Corpus: NuminaMath-CoT-100K (100K sampled, seed 1); base Mistral-7B-v0.3; packing to 8,192; otherwise the training recipe of Appendix B (lr 1.5×10^{-5}, cosine, seed 1, bf16 full-model). BREW uses γ=0.75 with the production temperature statistic (per-sample mean→median), at the writing default floor $w_{\min}$=0.1; results are reported at four epochs. Benchmarks are MATH500 (Lightman et al., 2024) and OlympiadBench (He et al., 2024) (OE_TO_maths_en_COMP, 674 problems). We report acc@1 under greedy decoding and pass@8 with T=0.7 and top-p=0.95. Answers are graded

with `math_verify` using brace-matched `\boxed` extraction. All six methods use the same recipe (global batch 64) and the same 4-epoch data budget, and are compared at the shared endpoint (§4.4).

Table 13: Six methods on the two math benchmarks, averaged over three independent runs. Bold marks the per-column best; BREW underlined where it places second.

	MATH500		OlympiadBench	
Method	acc@1	pass@8	pass@1	pass@8
SFT	**28.4**	55.6	10.1	33.5
DFT	7.8	17.6	3.0	10.1
ASFT	22.0	41.8	8.0	24.5
InfoSFT	26.4	52.8	9.3	30.6
TALR	23.8	51.8	8.0	31.4
BREW ($w_{\min}$=0.1)	<u>28.2</u>	**56.0**	**11.4**	**34.6**

DFT failure mode. DFT's training loss decreases and token accuracy increases throughout training, but its generations frequently enter repetition loops before producing a parseable answer. On OlympiadBench, 32.8% of DFT outputs contain no parseable answer, compared with 4.2% for BREW. The corresponding failure counts out of 674 problems are SFT 10, BREW 28, InfoSFT 41, TALR 47, ASFT 110, and DFT 221. This matches InfoSFT's own report that DFT's output entropy converges toward zero on math.

Pre-training floor audit. We compute the initial floor assignment on a corpus sample from the cached base-model losses and the production weight formula, before any training update; at initialization $\ell_{\text{eff}} = \ell_\theta = \ell_b$. All quantities use the same sequence-mean-then-median temperature statistic as the training runs.

Table 14: Pre-training floor audit on Mistral-7B with NuminaMath and Qwen3.5-9B with DeepWriting. The two settings have similar overall floor rates, but the floored tokens occur in different parts of the supervision.

	Math (Mistral)	Writing (Qwen3.5-9B)
τ (per-sample mean→median)	0.70	1.21
ℓ_b p90/τ	3.4×	3.0×
Floored-token share (at init., $w_{\min}$=0.1)	14.3%	14.5%
Answer-span (`\boxed`) floor rate	30.7%	—
Floor rate, first / last third of response	16.5% / 12.1%	18.0% / 9.3%
Floored-token location	incl. **answer** tokens	more concentrated early in the response

The overall floor rate is similar in the two domains (14.3% vs. 14.5%), but its location differs: 30.7% of tokens inside math answer spans are floored, whereas in writing the floored tokens are more concentrated toward the beginning of the response.

M Limitations

(1) Our writing evaluation relies partly on LLM judges. Judge-free diagnostics provide complementary measurements, but judge-specific biases may remain. (2) The creativity-related metrics are proxies based on Zhang et al. (2025); Lu et al. (2025) and are computed from a single sample per prompt. (3) Base uncertainty has a different role in answer-determined tasks. The writing configuration transfers competitively to mathematics without retuning (§4.4), but we do not explore task-specific hyperparameter tuning. (4) We use fixed γ and $w_{\min}$ across the experiments. γ=0.75 places the unfloored weight peak at $q \approx 0.47$, while $w_{\min}$=0.1 sets the minimum token weight; other tasks may favor different values. (5) The frozen reference measures progress from the initial model. Its behavior over training horizons substantially longer than those studied here remains untested.