LEARNING WHERE AND HOW TO TRAIN FROM MODEL HISTORY: SUPERVISED FINE-TUNING WITH HISTORICAL STATES

Anonymous authors
Paper under double-blind review

ABSTRACT

Supervised fine-tuning (SFT) adapts large language models through reference responses. We study how a stronger model can use its own weaker historical states to improve learning from the same supervision. We introduce *Weak-Driven Learning* and instantiate it as WMSS, which uses model history to guide example selection and token-level correction. **Weak-Driven Discrepancy Selection (WD-DS)** selects training examples using current predictive uncertainty and historical entropy differences. **Weak-Driven Joint-Logit Training (WD-JT)** mixes historical and current logits before applying cross-entropy to the original targets. Our analysis separates the scale and candidate allocation of the strong model's correction, characterizes the margin compensation required by the historical reference, and derives conditions under which the strong model retains the resulting margin gains after the reference is removed. Experiments across mathematical reasoning, code generation, and logical reasoning on four 3B–8B Qwen and Gemma models show consistent improvements over SFT and competitive baselines under the same nominal two-epoch allocation. WMSS requires no additional generated responses and preserves single-model inference.

1 INTRODUCTION

Supervised fine-tuning (SFT) adapts pretrained large language models to instruction following and reasoning using reference responses (Ouyang et al., 2022). The target tokens specify what the model should predict, while the model's earlier training states record how its predictions changed on the way to the current checkpoint. These weaker historical models can expose plausible mistakes that the current model no longer assigns much probability. We study whether the strong model can use that record to guide further learning from the same supervised targets.

A growing line of self-improvement methods follows this direction by constructing supervision from the model itself. However, in most such methods, the model-derived signal must first be converted into explicit supervision through additional computation, such as sampling self-generated responses for preference construction (Chen et al., 2024) or running auxiliary contexts to form teacher targets (Mitra & Ulukus, 2025). Ordinary SFT already leaves a useful record: earlier, weaker models from the strong model's own training trajectory. Their uncertain or incorrect predictions reveal alternatives that the current model can revisit under the original labels. We therefore ask: ***Can a strong model use weaker models from its training history as a reference to correct its own predictions?***

We study this principle as *Weak-Driven Learning* and instantiate it as WMSS (**W**eak models **M**ake **S**trong models **S**tronger). As illustrated in Figure 1, a weak historical model provides a reference for the strong model's next correction. Its value appears at key decision tokens, where it remains uncertain or assigns mass to plausible but incorrect alternatives. Comparing the two predictions reveals decisions the strong model can sharpen. WMSS brings the weak model's logits into a mixed-logit objective while keeping the ground-truth token as the target. In this way, the strong model uses its own weaker history to correct its behavior without additional deployment-time inference cost.

Our concrete implementation uses history for two decisions. **Weak-Driven Discrepancy Selection (WD-DS)** compares the predictive entropy of the strong model and its weaker historical reference to select examples for further SFT. **Weak-Driven Joint-Logit Training (WD-JT)** mixes their logits

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.

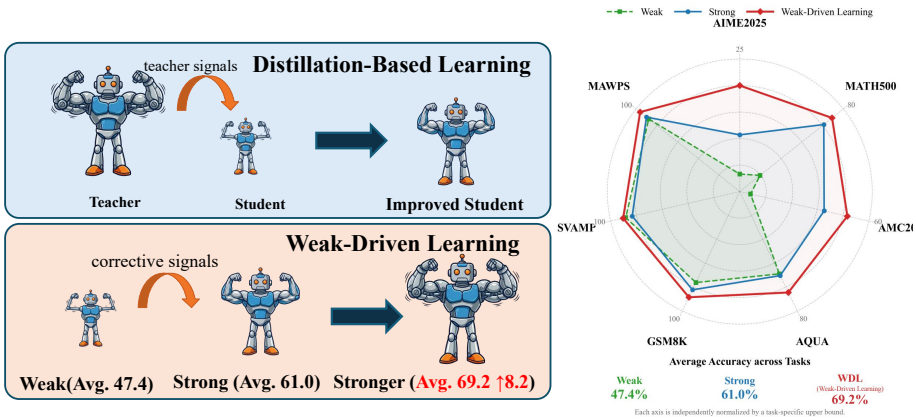

Figure 1: **Paradigm Comparison:** Distillation-Based Learning vs. Weak-Driven Learning.

on those examples and applies cross-entropy to the original ground-truth token. The weak model's plausible mistakes alter the correction signal faced by the strong model, especially at hard decisions. We characterize the strong branch's logit gradient as position-wise scaling and candidate redistribution, and test the two method components separately in our ablations.

More broadly, our findings show how a strong model can use weak models from its training history to improve SFT on existing supervision. Our contributions are summarized as follows:

- **Historical reference for SFT.** We introduce *Weak-Driven Learning*, in which a strong model uses weaker historical checkpoints to identify and correct hard predictions.
- **Training framework.** We instantiate this direction with WMSS, an SFT framework that combines **WD-DS** for entropy-discrepancy sample selection and **WD-JT** for weak-driven logit mixing. Together, these modules let the strong model correct its predictions using the weak historical reference without additional inference overhead.
- **Theoretical analysis.** We show that WD-JT equals SFT plus an exact per-position penalty, derive the margins that history requires of the strong branch, decompose its logit gradient into a position-wise scale and a candidate reallocation, and prove that the gain is retained by the strong model after the reference is removed.
- **Empirical performance.** Relative to the two-epoch SFT baseline, WMSS improves Math Avg by $+2.9$ to $+5.9$ points and Code Avg by $+2.4$ to $+4.4$ points across four backbones, with no additional deployment-time inference cost.

2 Related Work

Post-training: distillation and perturbation-based regularization. Post-training commonly starts from supervised fine-tuning (SFT) (Ouyang et al., 2022) and often uses knowledge distillation from a teacher (Hinton et al., 2015; Agarwal et al., 2024) or self-distillation (Zhang et al., 2019). These objectives provide useful supervision, while WMSS additionally compares current predictions with a weaker historical model to guide where and how the original targets are used. Perturbation regularizers such as **NEFTune** (Jain et al., 2024) inject noise for regularization rather than using this historical comparison.

Self-improvement and weak-to-strong supervision. Another line of work improves models using self-generated outputs or weaker variants. ReST (Gulcehre et al., 2023), **SPIN** (Chen et al., 2024), and **SSB** (Mitra & Ulukus, 2025) construct model-derived training signals. Weak-to-strong generalisation (Burns et al., 2024), weak-to-strong reasoning (Yang et al., 2024), and **LightReasoner** (Wang et al., 2026) use weaker supervision in different forms. Nearby work also studies contrastive-entropy data selection (Su et al., 2026).

Our position. Self-play methods such as SPIN (Chen et al., 2024) improve an SFT model by sampling responses from its previous iteration and training the next model to distinguish those

responses from the original demonstrations. WMSS shares the goal of self-improvement without a stronger external teacher, but uses the past self differently: a weaker historical model guides which existing examples receive further training and how their original targets correct token-level predictions. This turns the model's own training history into feedback for SFT without generating new responses or changing inference-time behavior.

3 WEAK-DRIVEN LEARNING

3.1 PRELIMINARIES

We consider an autoregressive language model $\mathcal{M}_\theta$ that maps a context sequence x to logits $z_t \in \mathbb{R}^{|\mathcal{V}|}$ over a vocabulary $\mathcal{V}$, inducing $P_\theta(\cdot \mid x) = \mathrm{Softmax}(z_t)$. **Supervised fine-tuning.** In standard supervised fine-tuning (SFT), we minimize the negative log-likelihood over a dataset $\mathcal{D}$:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\mathbb{E}_{(x,y)\sim\mathcal{D}}[\log P_\theta(y \mid x)]\,, \tag{1}$$

where y denotes the ground-truth next token under context x (we omit the time index when clear).
Predictive entropy. To quantify uncertainty and weak–strong discrepancy, we monitor the predictive entropy $H(P_\theta(\cdot \mid x)) = -\sum_{v\in\mathcal{V}} P_\theta(v \mid x) \log P_\theta(v \mid x)$. Let $\ell(x, y; \theta) = -\log P_\theta(y \mid x)$ be the per-token cross-entropy loss. Its gradient w.r.t. the logit of any token $k \in \mathcal{V}$ is $\partial\ell/\partial z_t[k] = P_\theta(k \mid x) - \mathbb{I}[k = y]$. In particular, for any negative token $k \neq y$, $|\partial\ell/\partial z_t[k]| = P_\theta(k \mid x)$, i.e., the gradient magnitude on a negative class is proportional to its assigned probability.

3.2 PROBLEM: WEAK-DRIVEN LEARNING

We define *Weak-Driven Learning* as SFT in which a strong model uses a weaker model from its own training history as a reference for correcting its predictions. Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ be the existing supervised dataset, $\mathcal{M}_{\text{strong}}$ the current model, and $\mathcal{M}_{\text{weak}}$ a historical model from its training trajectory. For an input x, let

$$o_{\text{strong}}(x) = \mathcal{M}_{\text{strong}}(x), \qquad o_{\text{weak}}(x) = \mathcal{M}_{\text{weak}}(x)$$

denote their predictions. In WMSS, these predictions provide entropy statistics for selecting examples and logits for forming the supervised correction.

The goal of weak-driven learning is to construct a discrepancy-aware training signal

$$\mathcal{S}_{\text{WD}} = \Phi(o_{\text{strong}}(x), o_{\text{weak}}(x), y)\,, \qquad (x, y) \in \mathcal{D}, \tag{2}$$

where Φ uses weak–strong discrepancies to shape the strong model's correction while preserving the original target y. The weak model provides a historical reference for the supervised correction, and the resulting stronger model $\mathcal{M}^+_{\text{strong}}$ is used downstream.

The historical weak model reveals uncertain choices and plausible mistakes. The strong model uses this reference to identify which examples need more attention and which incorrect tokens deserve a sharper correction.

4 METHOD: WMSS

WMSS is our practical instantiation of weak-driven learning. It consists of two modules: **Weak-Driven Joint-Logit Training (WD-JT)**, which uses weak-model logits to shape supervised correction, and **Weak-Driven Discrepancy Selection (WD-DS)**, which uses current uncertainty and historical entropy differences to choose examples for that training.

In this section, we describe how WMSS operationalizes weak-driven learning to further strengthen a strong model.

Weak and strong model initialization. In WMSS, the weak model $\mathcal{M}_{\text{weak}}$ is initialized from a historical checkpoint and provides weak logits during WD-JT. We begin with **Phase 1 (Initialization)**: starting from a base model $\mathcal{M}_0$, we perform standard SFT to obtain $\mathcal{M}_1$, and set

$$\mathcal{M}_{\text{weak}} \leftarrow \mathcal{M}_0, \qquad \mathcal{M}_{\text{strong}} \leftarrow \mathcal{M}_1. \tag{3}$$

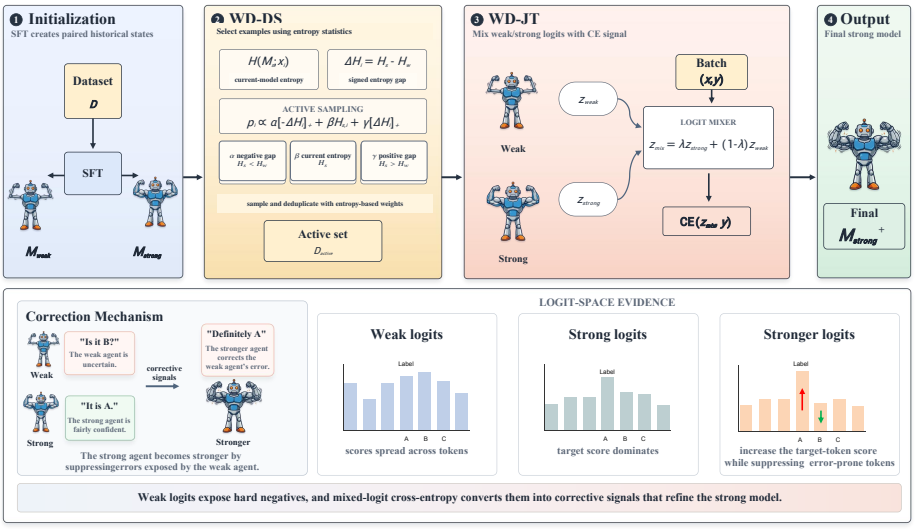

Figure 2: **Workflow of WMSS,** using a weaker historical model to select training examples and shape corrections from the original supervised targets.

The weak model supplies logits $z_{\text{weak}}(x) \in \mathbb{R}^{|\mathcal{V}|}$ that reveal plausible wrong alternatives from an earlier stage. For the iterative variant ($K>1$ in Algorithm 1), each stage draws its weak reference from the preceding historical checkpoint.

4.1 WD-JT: Weak-Driven Joint-Logit Training

The core weak-driven operation in WMSS is WD-JT: the strong model uses logits from a weaker historical model to form a supervised correction. The weak model exposes plausible errors, and the ground-truth target guides the strong model away from them. This module determines how the strong model uses its historical reference.

Weak-conditioned logit mixing. For a training pair (x, y), let $z_{\text{strong}}(x), z_{\text{weak}}(x) \in \mathbb{R}^{|\mathcal{V}|}$ denote the next-token logits produced by $\mathcal{M}_{\text{strong}}$ and $\mathcal{M}_{\text{weak}}$, respectively. We construct a mixed probability map by linearly mixing the logits:

$$z_{\text{mix}}(x) = \lambda\, z_{\text{strong}}(x) + (1-\lambda)\, z_{\text{weak}}(x), \tag{4}$$

where $\lambda \in [0, 1]$. We apply the original supervised target to the mixed-logit distribution $P_{\text{mix}}(\cdot \mid x) := \text{Softmax}(z_{\text{mix}}(x))$:

$$\mathcal{L}_{\text{mix}} = -\mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\log P_{\text{mix}}(y \mid x)\right] \tag{5}$$

This mixed-logit objective uses the weak model's predictions as a reference for the strong model's supervised correction. Historical logits change the probabilities assigned to competing alternatives under the original target. Section 5 derives the resulting strong-branch gradient, including the chain-rule coefficient λ, and characterizes how history changes its scale and candidate allocation.

4.2 WD-DS: Weak-Driven Discrepancy Selection

WD-DS is the entropy-based selector for WD-JT: current predictive entropy and its difference from the historical model determine which examples receive further training. These statistics describe uncertainty and prediction change; Appendix C relates them to the residual that WD-JT re-exposes and corrects.

Entropy-based selection. For each supervised sequence example $x_i = (u_i, y_i)$, let $\mathcal{T}_i$ denote the target-token positions included in the training loss and let $c_{i,t} = (u_i, y_{i,<t})$ be the prompt concatenated with the ground-truth target prefix before position t. We use the length-normalized

predictive entropy

$$H(\mathcal{M}; x_i) = -\frac{1}{|\mathcal{T}_i|} \sum_{t \in \mathcal{T}_i} \sum_{v \in \mathcal{V}} P_{\mathcal{M}}(v \mid c_{i,t}) \log P_{\mathcal{M}}(v \mid c_{i,t}). \tag{6}$$

The weak–strong discrepancy is

$$\Delta H_i = H(\mathcal{M}_{\text{strong}}; x_i) - H(\mathcal{M}_{\text{weak}}; x_i). \tag{7}$$

WD-DS assigns a score and a per-draw sampling probability to each example:

$$s_i = \alpha[-\Delta H_i]_+ + \beta H(\mathcal{M}_{\text{strong}}; x_i) + \gamma[\Delta H_i]_+, \qquad p_i = \frac{s_i}{\sum_{j=1}^{N} s_j}, \tag{8}$$

where $N = |\mathcal{D}|$ and $[u]_+ := \max(u, 0)$. The α and γ terms weight negative and positive entropy differences, while β weights current uncertainty. The defaults $\alpha = \gamma = 0.1$, $\beta = 0.8$ give $s_i = 0.8H(\mathcal{M}_{\text{strong}}; x_i) + 0.1|\Delta H_i|$. Entropy change is a selection cue rather than a label of forgetting or recovery.

Sampling and deduplication. Holding $\mathbf{p}$ fixed within a stage, draw $J_1, \ldots, J_N$ independently from $\text{Categorical}(\mathbf{p})$ and retain the unique index set $A = \{J_1, \ldots, J_N\}$. Let $\mathcal{D}_{\text{active}} = \{x_i : i \in A\}$ and $M = |A|$. The marginal inclusion probability and expected active-set size are

$$\pi_i := \Pr(i \in A \mid \mathbf{p}) = 1 - (1 - p_i)^N, \qquad \mathbb{E}[M \mid \mathbf{p}] = \sum_{i=1}^{N} \pi_i. \tag{9}$$

Thus p_i governs a single draw, whereas the π_i govern final inclusion and need not sum to one. Deduplication preserves the ordering of p_i but compresses relative inclusion differences. WD-JT makes one pass over the realized active set; repeated draws add no multiplicity weight.

4.3 Training Pipeline

Figure 2 summarizes WMSS and the roles of its two modules; Algorithm 1 in Appendix D.5 gives the full training loop.
Phase 1 (Initialization) trains $\mathcal{M}_0$ with SFT to obtain $\mathcal{M}_1$ and initializes the weak and strong models as in Eq. (3).
Phase 2 (WD-DS) computes $\mathbf{p}$ via Eq. (8), draws N indices with replacement, and deduplicates them into $\mathcal{D}_{\text{active}}$, with inclusion probabilities given by Eq. (9).
Phase 3 (WD-JT) makes one pass over $\mathcal{D}_{\text{active}}$ using the original targets and mixed-logit loss in Eq. (5), then retains $\mathcal{M}^{+}_{\text{strong}}$.

5 Mechanistic Analysis: Why Joint Training Improves the Strong Model Alone

Fix a context x, target y, finite logits, and $0 < \lambda < 1$. Write $p = P_{\text{strong}}(\cdot \mid x)$, $u = P_{\text{weak}}(\cdot \mid x)$, and $r = P_{\text{mix}}(\cdot \mid x)$. The branches have disjoint parameters, so for the unscaled losses $\ell_{\text{JT}} = -\log r_y$ and $\ell_{\text{SFT}} = -\log p_y$ at the same strong state,

$$g_{\text{JT}} := \nabla_{z_{\text{strong}}} \ell_{\text{JT}} = \lambda(r - e_y), \qquad g_{\text{SFT}} := \nabla_{z_{\text{strong}}} \ell_{\text{SFT}} = p - e_y. \tag{10}$$

For $k \neq y$, let $m_{s,k} = z_{\text{strong}}[y] - z_{\text{strong}}[k]$ and $m_{w,k} = z_{\text{weak}}[y] - z_{\text{weak}}[k]$ be the target margins and $\Delta_k = m_{s,k} - m_{w,k}$ the *historical margin gap*, positive when the historical model separates y from k less clearly. Let $q_k = p_k/(1 - p_y)$ and $\tilde{q}_k = r_k/(1 - r_y)$ be the non-target allocations of the strong and mixed predictions ($q_y = \tilde{q}_y = 0$), $\rho = \sum_{k \neq y} q_k e^{(1-\lambda)\Delta_k}$, and $D = p_y + (1 - p_y)\rho$. WD-JT optimizes ℓ_{JT} but deploys p alone. Proofs are in Appendix B.

What the objective demands. Since $r_y = p_y/D$, the two losses differ by an exact per-position term, $\ell_{\text{JT}} = \ell_{\text{SFT}} + \log D$, positive exactly when $\rho > 1$, i.e. when the historical model is confused on candidates the current model still entertains.

Proposition 1 (Historical margin compensation)**.** *Let* $0 < \varepsilon < \log 2$ *and* $b_\varepsilon = -\log(e^\varepsilon - 1)$. *If* $\ell_{\text{JT}} \leq \varepsilon$, *then for every* $k \neq y$, $m_{s,k} \geq b_\varepsilon + \frac{1-\lambda}{\lambda}(b_\varepsilon - m_{w,k})$.

Reaching the same loss under SFT requires only $m_{s,k} \geq b_\varepsilon$: the strong branch must add $(1-\lambda)/\lambda$ times any historical shortfall, and is relaxed where history already exceeds b_ε, which is rarely binding on an SFT trajectory (Appendix B.10).

How the correction is allocated.

Theorem 2 (Exact allocation of the strong-branch correction)**.**

$$g_{\mathrm{JT}} = S\, g_{\mathrm{SFT}} + \lambda(1-r_y)(\tilde{q}-q), \qquad S = \lambda w, \quad w = \frac{1-r_y}{1-p_y} = \frac{\rho}{D}, \qquad \tilde{q}_k = \frac{q_k e^{(1-\lambda)\Delta_k}}{\rho}, \tag{11}$$

where the second term has zero sum and zero target coordinate, $w > 1$ if and only if $\rho > 1$, and $\tilde{q} = \arg\max_v \{(1-\lambda)\,\mathbb{E}_{k\sim v}[\Delta_k] - \mathrm{KL}(v\|q)\}$, so that $\tilde{q}(H) > q(H)$ for every nontrivial upper level set $H = \{k \neq y : \Delta_k \geq h\}$.

The constant λ is shared by every position and absorbed into the step size; the position-specific w decides where the correction concentrates and $\tilde{q}$ which candidates receive it. Re-exposure ($w > 1$) requires the historical model to be less confident on the target by a log-odds gap exceeding a Rényi divergence between the two non-target allocations (Appendix B.4); absolute amplification is a separate, conditional property (Appendix B.5).

What the strong branch retains. A step along $-g_{\mathrm{JT}}$ raises every target margin by $\eta\lambda(1-r_y)(1+\tilde{q}_k) > 0$ and decreases the standalone loss to first order ($\langle g_{\mathrm{SFT}}, g_{\mathrm{JT}}\rangle > 0$): the reference never asks the strong branch to fit its errors. At equal ℓ_1 correction mass, the margin against k grows $(1+\tilde{q}_k)/(1+q_k)$ times as fast as under SFT, exceeding one exactly when $\tilde{q}_k > q_k$. SFT is the steepest-descent direction at a fixed ℓ_2 logit-update norm; at fixed ℓ_1 correction mass, candidate allocation can also change the standalone loss's first-order descent rate (Appendix B.8). The required margins are retained without the reference:

Proposition 3 (Gains survive removal of the reference)**.** *Let $m^0_{s,k}$ and $m^+_{s,k}$ be the strong margins before and after an interval of WD-JT, $m_{w,k}$ the historical margin at its end, $\Delta^0_k = m^0_{s,k} - m_{w,k}$, and $m^+_{\mathrm{mix},k} = \lambda m^+_{s,k} + (1-\lambda) m_{w,k}$. Then, exactly,*

$$m^+_{s,k} - m^0_{s,k} = \frac{1-\lambda}{\lambda}\,\Delta^0_k + \frac{m^+_{\mathrm{mix},k} - m^0_{s,k}}{\lambda}. \tag{12}$$

Hence if $m^+_{\mathrm{mix},k} \geq m^0_{s,k} - \varepsilon_k$ with $0 \leq \varepsilon_k < (1-\lambda)\Delta^0_k$, then $m^+_{s,k} - m^0_{s,k} \geq \big((1-\lambda)\Delta^0_k - \varepsilon_k\big)/\lambda > 0$; if no margin decreases and one increases, then $p^+_y > p^0_y$ and $\ell_{\mathrm{SFT}}(z^+_{\mathrm{strong}}) < \ell_{\mathrm{SFT}}(z^0_{\mathrm{strong}})$.

Mixing first lowers the mixed margin to $m^0_{s,k} - (1-\lambda)\Delta^0_k$; once training merely restores it to the initial strong separation, the strong branch alone has exceeded its initial margin by $(1-\lambda)\Delta^0_k/\lambda$. The target remains y; the initial strong model is only the yardstick.

History versus temperature. A uniform reference gives $r^{\mathrm{temp}} = \mathrm{Softmax}(\lambda z_{\mathrm{strong}})$ and also changes w and $\tilde{q}$; at equal temperature, history contributes exactly $(r_k\, r_l^{\mathrm{temp}})/(r_l\, r_k^{\mathrm{temp}}) = (u_k/u_l)^{1-\lambda}$ for $k, l \neq y$, the weak model's own relative odds among non-target tokens, a temperature transform only when $u \propto p^a$. These identities hold at one strong state, before the Jacobian and the optimizer; Sections 6.3 and 6.4 test the effect on accuracy.

6 Experiments

We test whether WMSS improves over SFT and offline baselines (Section 6.2), how its modules contribute and at what cost (Section 6.3), and whether simpler alternatives explain its gains (Section 6.4); Appendix A adds further experiments.

6.1 Experimental Setup

Backbones and tasks. We use four 3B–8B base models from two families: Qwen3-4B-Base, Qwen3-8B-Base, Qwen2.5-3B, and Gemma-3-4B-PT. We report accuracy on three domains: mathematical reasoning (AIME2025, MATH500, AMC23, AQuA, GSM8K, MAWPS, and SVAMP), code

Table 1: **Math and code results on four Qwen and Gemma backbones.** Accuracy (%).

Method	Math								Code				
	AIME25	MATH	AMC23	AQuA	GSM8K	MAWPS	SVAMP	Avg.	HE+	MBPP+	BCB	LCB	Avg.
Qwen3-4B-Base as base model.													
SFT	13.3	66.2	47.5	61.8	83.9	91.2	85.5	64.2	73.8	64.0	38.5	17.8	48.5
Random	13.3	63.2	40.0	55.1	82.9	90.8	84.1	61.3 ↓2.9	72.6	66.9	39.6	17.5	49.2 ↑0.7
NEFTune	16.7	68.2	42.5	59.1	86.7	93.7	87.7	64.9 ↑0.7	75.0	66.9	40.3	16.0	49.6 ↑1.1
SPIN	**20.0**	70.8	47.5	66.9	85.8	87.4	87.6	66.6 ↑2.4	76.2	66.7	39.1	18.8	50.2 ↑1.7
SSB	10.0	70.6	45.0	66.1	87.1	95.4	88.8	66.1 ↑1.9	78.0	67.5	42.5	18.0	51.5 ↑3.0
WMSS	**20.0**	**71.4**	**50.0**	**67.7**	**88.5**	**96.2**	**90.3**	**69.2 ↑5.0**	**78.7**	**68.3**	**43.5**	**21.0**	**52.9 ↑4.4**
Qwen3-8B-Base as base model.													
SFT	16.7	72.2	45.0	63.0	87.5	95.0	88.6	66.9	80.5	69.3	44.5	21.0	53.8
Random	10.0	71.6	47.5	66.9	89.9	96.6	91.2	67.7 ↑0.8	79.9	69.3	45.7	21.5	54.1 ↑0.3
NEFTune	16.7	73.8	40.0	71.3	90.3	97.5	91.5	68.7 ↑1.8	81.1	70.4	45.9	21.0	54.6 ↑0.8
SPIN	16.7	72.4	47.5	**78.3**	88.8	95.4	92.7	70.3 ↑3.4	81.7	70.1	46.3	21.8	55.0 ↑1.2
SSB	16.7	74.6	**52.5**	76.4	89.3	93.3	93.0	70.8 ↑3.9	81.1	66.9	46.0	20.8	53.7 ↓0.1
WMSS	**20.0**	**75.4**	**52.5**	77.2	**92.5**	**97.9**	**94.0**	**72.8 ↑5.9**	**84.1**	**71.4**	**47.1**	**22.3**	**56.2 ↑2.4**
Qwen2.5-3B as base model.													
SFT	3.3	47.0	20.0	51.2	76.6	91.6	84.4	53.4	58.5	59.0	27.5	11.0	39.0
Random	3.3	50.8	17.5	49.2	77.4	91.2	83.2	53.2 ↓0.2	56.1	59.5	28.9	**12.5**	39.3 ↑0.3
NEFTune	3.3	51.2	**22.5**	47.2	78.1	90.3	84.7	53.9 ↑0.5	56.1	59.5	29.6	11.5	39.2 ↑0.2
SPIN	3.3	51.6	20.0	49.6	77.4	91.2	**86.1**	54.2 ↑0.8	59.8	60.6	31.4	12.0	41.0 ↑2.0
SSB	3.3	51.4	**22.5**	52.0	79.9	92.0	84.9	55.1 ↑1.7	54.3	52.9	30.4	9.3	36.7 ↓2.3
WMSS	**6.7**	**52.0**	**22.5**	**53.9**	**80.9**	**92.4**	86.0	**56.3 ↑2.9**	**62.2**	**61.9**	**32.7**	**12.5**	**42.3 ↑3.3**
Gemma-3-4B-PT as base model.													
SFT	0.0	28.0	17.5	46.5	54.0	83.6	68.3	42.6	36.6	43.1	18.5	7.5	26.4
Random	0.0	31.8	20.0	43.7	58.8	86.6	**70.9**	44.5 ↑1.9	36.6	47.1	19.2	7.8	27.7 ↑1.3
NEFTune	0.0	32.6	17.5	46.1	59.2	86.6	70.8	44.7 ↑2.1	37.8	46.3	20.1	8.0	28.1 ↑1.7
SPIN	0.0	30.6	7.5	48.0	58.6	84.5	66.1	42.2 ↓0.4	37.2	34.7	19.6	7.5	24.8 ↓1.6
SSB	0.0	31.2	5.0	43.7	51.1	69.3	58.7	37.0 ↓5.6	39.0	39.7	19.0	7.3	26.3 ↓0.1
WMSS	0.0	**33.2**	**22.5**	**51.2**	**60.5**	**87.4**	69.5	**46.3 ↑3.7**	**41.5**	**48.9**	**21.8**	**9.0**	**30.3 ↑3.9**

generation (HumanEval+, MBPP+, BigCodeBench, and LiveCodeBench), and logical reasoning (LogiQA 2.0 and ReClor). All training sets are decontaminated against these benchmarks by exact problem-statement matching (Appendix D.2).

Baselines and training budget. We compare with standard SFT, Random, NEFTune (Jain et al., 2024), SPIN (Chen et al., 2024), and SSB (Mitra & Ulukus, 2025), all trained, like WMSS, under the same two-epoch allocation. Since WMSS is an offline post-training method, we restrict baselines to offline or off-policy methods, which learn from fixed supervision, alternate-context targets, or responses generated before the current update; we exclude on-policy RL, whose results depend strongly on online sampling and reward-model budgets. Details are in Appendix D.

6.2 Effectiveness and Generalization

Main comparison. **WMSS achieves the best Math Avg and Code Avg on all four backbones** (Table 1). Relative to two-epoch SFT, it gains 2.9–5.9 points on Math Avg and 2.4–4.4 on Code Avg, peaking on Qwen3-8B-Base for math ($66.9\% \rightarrow 72.8\%$) and Qwen3-4B-Base for code ($48.5\% \rightarrow 52.9\%$). Although the strongest competitor varies across blocks, WMSS leads both averages in every block, with at least a 1.2-point margin over the runner-up.

Performance under the training budget. The gains require no extra supervised epoch and only modest additional compute. All methods in Table 1 share the same nominal two-epoch allocation, which matches supervised passes rather than compute. On Qwen3-4B-Base, WMSS takes $218 + 123 = 341$ optimizer steps versus 436 for SFT, because the deduplicated WD-DS active set yields 123 rather than 218 second-stage updates. In exchange, it adds an entropy-scoring pass of both models over $\mathcal{D}$ and a weak-model forward at every WD-JT step. This totals roughly $1.2\times$ the forward-pass compute of two-epoch SFT (Appendix D.5); we do not train an SFT baseline to matched compute.

Table 2: **Three-run math results on Qwen3 backbones.** Models are trained with seeds 0, 1, and 2.

	Methods	AIME25	MATH	AMC23	AQuA	GSM8K	MAWPS	SVAMP	Avg$_7$
Qwen3-4B	SFT	$12.2_{\pm 5.09}$	$63.8_{\pm 2.62}$	$48.3_{\pm 1.44}$	$60.2_{\pm 1.42}$	$82.2_{\pm 1.63}$	$90.3_{\pm 0.73}$	$85.7_{\pm 1.21}$	$63.3_{\pm 0.88}$
	Random	$12.2_{\pm 5.09}$	$62.9_{\pm 1.22}$	$39.2_{\pm 3.82}$	$54.2_{\pm 2.68}$	$81.7_{\pm 1.09}$	$89.8_{\pm 1.35}$	$83.7_{\pm 0.84}$	$60.5_{\pm 1.24}$
	NEFTune	$16.7_{\pm 3.33}$	$66.1_{\pm 1.86}$	$44.2_{\pm 2.89}$	$56.2_{\pm 2.50}$	$85.5_{\pm 1.08}$	$92.6_{\pm 1.28}$	$87.4_{\pm 1.57}$	$64.1_{\pm 1.45}$
	SPIN	$18.9_{\pm 5.09}$	$72.3_{\pm 1.75}$	$48.3_{\pm 3.82}$	$65.4_{\pm 1.57}$	$84.9_{\pm 2.17}$	$87.4_{\pm 0.84}$	$87.4_{\pm 1.01}$	$66.4_{\pm 2.09}$
	SSB	$10.0_{\pm 3.33}$	$69.7_{\pm 1.50}$	$45.8_{\pm 1.44}$	$63.3_{\pm 2.53}$	$85.9_{\pm 1.12}$	$96.4_{\pm 1.70}$	$88.6_{\pm 1.76}$	$65.7_{\pm 0.62}$
	WMSS	$\mathbf{20.0}_{\pm 3.33}$	$\mathbf{72.7}_{\pm 1.81}$	$\mathbf{50.8}_{\pm 1.44}$	$\mathbf{69.3}_{\pm 1.57}$	$\mathbf{88.6}_{\pm 1.60}$	$\mathbf{97.5}_{\pm 1.11}$	$\mathbf{90.3}_{\pm 1.40}$	$\mathbf{69.9}_{\pm 1.31}$
Qwen3-8B	SFT	$15.6_{\pm 5.09}$	$71.2_{\pm 1.56}$	$44.2_{\pm 1.44}$	$60.9_{\pm 1.86}$	$86.4_{\pm 1.03}$	$93.7_{\pm 1.11}$	$89.0_{\pm 1.64}$	$65.8_{\pm 1.35}$
	Random	$10.0_{\pm 3.33}$	$71.1_{\pm 2.64}$	$48.3_{\pm 3.82}$	$65.2_{\pm 1.49}$	$89.1_{\pm 1.92}$	$95.2_{\pm 1.75}$	$91.0_{\pm 1.07}$	$67.1_{\pm 1.15}$
	NEFTune	$15.6_{\pm 5.09}$	$72.7_{\pm 1.36}$	$38.3_{\pm 2.89}$	$68.9_{\pm 2.19}$	$88.5_{\pm 1.62}$	$96.6_{\pm 0.84}$	$91.8_{\pm 0.93}$	$67.5_{\pm 1.33}$
	SPIN	$16.7_{\pm 3.33}$	$70.4_{\pm 2.00}$	$49.2_{\pm 2.89}$	$76.1_{\pm 1.94}$	$87.3_{\pm 1.34}$	$94.0_{\pm 1.21}$	$92.7_{\pm 1.90}$	$69.5_{\pm 1.50}$
	SSB	$16.7_{\pm 3.33}$	$74.7_{\pm 2.00}$	$51.7_{\pm 1.44}$	$73.8_{\pm 2.27}$	$88.7_{\pm 2.20}$	$92.2_{\pm 0.97}$	$93.0_{\pm 1.30}$	$70.1_{\pm 0.68}$
	WMSS	$\mathbf{20.0}_{\pm 3.33}$	$\mathbf{75.4}_{\pm 1.80}$	$\mathbf{53.3}_{\pm 1.44}$	$\mathbf{77.2}_{\pm 1.57}$	$\mathbf{92.7}_{\pm 1.60}$	$\mathbf{98.9}_{\pm 1.06}$	$\mathbf{94.0}_{\pm 1.50}$	$\mathbf{73.1}_{\pm 0.27}$

Table 3: **Logical reasoning across four backbones.** Accuracy (%) on LogiQA 2.0 and ReClor.

Methods	Qwen3-4B-Base			Qwen3-8B-Base			Qwen2.5-3B			Gemma-3-4B-PT		
	LogiQA2	ReClor	Avg.	LogiQA2	ReClor	Avg.	LogiQA2	ReClor	Avg.	LogiQA2	ReClor	Avg.
SFT	66.1	74.6	70.4	69.2	79.8	74.5	48.5	52.0	50.3	55.4	63.2	59.3
Random	69.1	75.4	72.3	71.3	81.8	76.6	54.8	60.0	57.4	55.7	67.4	61.6
NEFTune	68.2	73.0	70.6	72.3	83.8	78.1	54.1	59.0	56.6	56.1	68.2	62.2
SPIN	68.3	76.8	72.6	71.3	80.6	76.0	51.5	58.8	55.2	58.1	71.0	64.6
SSB	67.5	74.6	71.1	69.0	81.0	75.0	50.8	53.8	52.3	57.1	70.8	64.0
WMSS	**69.5**	**79.2**	**74.4**	**74.6**	**89.2**	**81.9**	**58.0**	**67.6**	**62.8**	**60.5**	**75.6**	**68.1**

Robustness and generalization. **The gains hold across training runs, model families, and task domains.** Over three Qwen3 runs (Table 2), WMSS reaches seven-benchmark math averages of $69.9\% \pm 1.31$ on 4B and $73.1\% \pm 0.27$ on 8B, ahead of the strongest alternatives (66.4%, 70.1%) by more than the run-to-run standard deviation. On Gemma-3-4B-PT, Math Avg and Code Avg improve by 3.7 and 3.9 points over SFT (Table 1). On logical reasoning, WMSS leads LogiQA 2.0 and ReClor on every backbone, with the largest Logic Avg gain of 12.5 points on Qwen2.5-3B ($50.3\% \rightarrow 62.8\%$; Table 3).

6.3 Component Analysis and Training Dynamics

Module ablation. **Both modules improve over SFT, and their gains are complementary** (Table 4, Qwen3-4B-Base). WD-DS alone lifts the three-benchmark mean from 54.4% to 56.3% via MATH500 ($+3.2$) and GSM8K ($+2.3$), leaving AIME2025 at 13.3%. WD-JT alone gains 3.8 points to 58.2% and improves all three benchmarks, including AIME2025 (16.7%). Combining both reaches 59.9%, 1.7 points above WD-JT alone, with 20.0%/71.4%/88.5% on AIME2025/MATH500/GSM8K. The AIME2025 result makes the interaction clear: WD-DS alone leaves accuracy unchanged, yet its addition to WD-JT brings another 3.3 points. WD-DS thus reallocates training across examples, while WD-JT reshapes token-level correction.

Table 4: **Module ablation** on Qwen3-4B-Base.

Method	AIME	MATH	GSM8K	Avg.
Baseline	13.3	66.2	83.9	54.4
+ WD-DS	13.3	69.4	86.2	56.3
+ WD-JT	16.7	70.2	87.6	58.2
WMSS	**20.0**	**71.4**	**88.5**	**59.9**

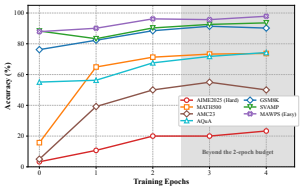

Figure 3: **Extended WMSS trajectory.** Four epochs extend beyond the main comparison.

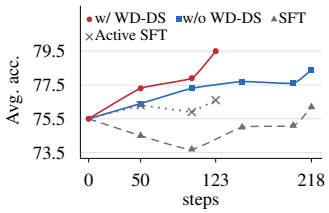

Figure 4: **WD-DS step reduction.** MATH500/GSM8K mean accuracy after shared initialization.

Most of the gain arrives within the two-epoch budget. Extending WMSS by two more weak-driven epochs (Figure 3) adds smaller improvements: every benchmark at Epoch 4 remains at or above its Epoch 2 score, although AMC23 peaks at Epoch 3. This pattern places the main gain within the planned budget. We report the Epoch 2 checkpoint fixed in advance, keeping the comparison independent of per-test-set stopping choices.

WD-DS reaches a stronger endpoint with 44% fewer second-stage updates. On the math pool, it draws $N = 111{,}657$ indices with replacement and trains on their distinct members. At global batch size 512, 123 steps provide 62,976 full-batch example slots (56.4% of N), a batching proxy for the active-set size rather than an exact unique-index count. Starting from the same MATH500/GSM8K checkpoint ($65.8\%/85.2\%$), WD-DS reaches $71.4\%/87.6\%$ after 123 steps; the full schedule reaches $69.4\%/87.3\%$ after 218 (Figure 4). Selection thus delivers 2.0/0.3 points higher accuracy with 95 fewer updates. Appendix D.5 includes WD-DS's one-time entropy-scoring pass in the compute accounting.

WMSS **attains the highest Math Avg at near-SFT wall-clock cost.** On Qwen3-4B-Base, Math Avg rises from SFT's 64.2% to 69.2% while training takes 4h49m, just 19 minutes (7%) longer than SFT's 4h30m (Figure 7). SPIN and SSB take about 7h53m and 18h, including response generation, and reach 66.6% and 66.1%. WMSS thus leads this comparison while keeping measured training time close to SFT; Appendix D.5 reports the corresponding forward-pass accounting.

6.4 Alternative Explanations

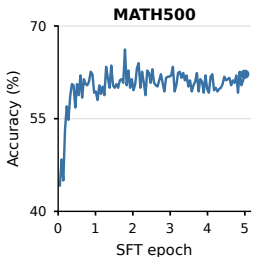

Figure 5: **Longer SFT trajectory.** MATH500 accuracy over 109 checkpoints.

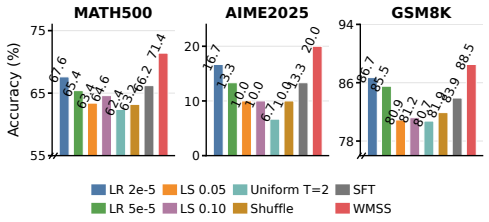

Figure 6: **Optimization and shuffle controls.** Each benchmark uses its own labeled y-axis start. LR: learning rate; LS: label smoothing.

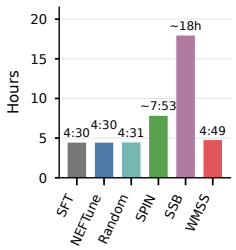

Figure 7: **Training time.** Qwen3-4B wall clock; SPIN/SSB include generation.

Longer SFT or best-checkpoint selection does not close the gap. Across all 109 checkpoints of a separate 1,090-step (about five-epoch) SFT run (Figure 5), MATH500 peaks at 66.2% at step 390 and ends at 62.2%; even this post-hoc best is 5.2 points below WMSS (71.4%). Its step-440 checkpoint (nearest two epochs) scores 59.6%, 6.6 points below the SFT baseline in Table 1, so this run is a diagnostic, not the budget-matched comparison.

Neither larger learning rates nor generic logit softening reproduces the gains. Figure 6 adds higher learning rates, vocabulary-uniform label smoothing, and a uniform weak-logit reference ($z_{\text{weak}} = 0$, $\lambda = 0.5$, so $z_{\text{mix}} = 0.5 z_{\text{strong}}$, i.e., temperature $T = 2$), which keeps logit softening but removes the sample-dependent weak distribution. On MATH500, learning rates 2×10^{-5} and 5×10^{-5} score 67.6% and 65.4% versus 66.2% for SFT, and label smoothing scores 63.4% and 64.6%; on AIME2025, 2×10^{-5} reaches 16.7% versus 13.3%. WMSS exceeds the shuffled weak-logit pairing (Shuffle) by 8.2, 10.0, and 6.6 points on MATH500, AIME2025, and GSM8K, so its gains come from the sample-specific weak reference rather than step size or softening alone.

7 Conclusion

We introduce *Weak-Driven Learning*, a new SFT paradigm in which a stronger model uses its weaker historical self to select examples and shape correction under the original targets. For further discussion, see Appendix E.

AI USE STATEMENT

Generative AI tools, including OpenAI Codex, assisted with refining the research framing and experimental design, literature search, checking mathematical derivations, code and figure preparation, and manuscript editing and translation. The authors independently verified the mathematical arguments and reported results, reviewed AI-assisted code and manuscript edits, and take full responsibility for the final text, claims, code, figures, and other artifacts.

REPRODUCIBILITY STATEMENT

Section 4 defines the WD-DS selection rule and WD-JT training objective, and Algorithm 1 specifies the training procedure. Appendix D documents the datasets, preprocessing, model backbones, training configurations, baselines, and evaluation protocols. Sections 6.3 and 6.4 report training time and optimization controls; Appendix A reports sensitivity analyses; Appendix B gives the proofs and recorded scale diagnostics for the strong-branch analysis, and Appendix C relates the WD-DS entropy statistics to the residual that WD-JT re-exposes.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/5be69a584901a26c521c2b51e40a4c20-Abstract-Conference.html.

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

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, Ilya Sutskever, and Jeffrey Wu. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4971–5012, 2024. URL https://proceedings.mlr.press/v235/burns24b.html.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 6621–6642, 2024. URL https://proceedings.mlr.press/v235/chen24j.html.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Gemma Team. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, Arnaud Doucet, Orhan Firat, and Nando de Freitas. Reinforced self-training (rest) for language modeling, 2023. URL https://arxiv.org/abs/2308.08998.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

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 (ICLR)*, 2025. URL https://arxiv.org/abs/2403.07974.

Neel Jain, Ping-yeh Chiang, Yuxin Wen, John Kirchenbauer, Hong-Min Chu, Gowthami Somepalli, Brian R. Bartoldson, Bhavya Kailkhura, Avi Schwarzschild, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Neftune: Noisy embeddings improve instruction finetuning. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2310.05914.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. Mawps: A math word problem repository. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 1152–1157, 2016. doi: 10.18653/v1/N16-1136.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 158–167, 2017. doi: 10.18653/v1/P17-1015. URL https://aclanthology.org/P17-1015/.

Hanmeng Liu, Jian Liu, Leyang Cui, Zhiyang Teng, Nan Duan, Ming Zhou, and Yue Zhang. LogiQA 2.0—an improved dataset for logical reasoning in natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2947–2962, 2023a. doi: 10.1109/TASLP.2023.3293046.

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 (NeurIPS)*, 2023b.

Purbesh Mitra and Sennur Ulukus. Semantic soft bootstrapping: Long context reasoning in LLMs without reinforcement learning. *arXiv preprint arXiv:2512.05105*, 2025.

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. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2080–2094, 2021. doi: 10.18653/v1/2021.naacl-main.168. URL https://aclanthology.org/2021.naacl-main.168/.

Junyou Su, He Zhu, Xiao Luo, Liyu Zhang, Hong-Yu Zhou, Yun Chen, Peng Li, Yang Liu, and Guanhua Chen. InstructDiff: Domain-adaptive data selection via contrastive entropy for efficient LLM fine-tuning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10630–10648, 2026. doi: 10.18653/v1/2026.acl-long.486. URL https://aclanthology.org/2026.acl-long.486/.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. TRL: Transformers reinforcement learning. GitHub repository, 2020. URL https://github.com/huggingface/trl.

Jingyuan Wang, Yankai Chen, Zhonghang Li, and Chao Huang. LightReasoner: Can small language models teach large language models reasoning? In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2635–2663, 2026. doi: 10.18653/v1/2026.acl-long.122. URL https://aclanthology.org/2026.acl-long.122/.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, 2020.

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, 2025. URL https://arxiv.org/abs/2505.09388.

Yuqing Yang, Yan Ma, and Pengfei Liu. Weak-to-strong reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 8350–8367, 2024. doi: 10.18653/v1/2024.findings-emnlp.490. URL https://aclanthology.org/2024.findings-emnlp.490/.

Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. Reclor: A reading comprehension dataset requiring logical reasoning. In *International Conference on Learning Representations (ICLR)*, 2020.

Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. doi: 10.1109/ICCV.2019.00381. URL https://arxiv.org/abs/1905.08094.

Han Zhao, Haotian Wang, Yiping Peng, Sitong Zhao, Xiaoyu Tian, Shuaiting Chen, Yunjie Ji, and Xiangang Li. 1.4 million open-source distilled reasoning dataset to empower large language model training, 2025. URL https://arxiv.org/abs/2503.19633.

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 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 (ICLR)*, 2025.

Appendix Contents

A. Additional Experiments 12

B. Supplementary Theory: Margin Compensation and Gradient Allocation 14

C. Supplementary Theory: Entropy Selection and Residual Re-exposure 18

D. Data Construction and Training Details 20

E. Discussion 24

A Additional Experiments

A.1 Fixed hyperparameters and post-hoc sensitivity

The headline experiments in Table 1 use fixed coefficients $\alpha = 0.1$, $\beta = 0.8$, $\gamma = 0.1$, and $\lambda = 0.5$ across the reported models and domains. Tables 5 and 6 examine these choices after the headline runs. The coefficient sweep reports AIME2025 and MATH500 separately; the λ sweep reports a six-benchmark average over AIME2025, MATH500, AQuA, GSM8K, MAWPS, and SVAMP. Within the latter sweep, the average is 75.5% at $\lambda = 0.42$ and 74.4% at the fixed default $\lambda = 0.5$. The default also gives the highest AIME2025 score in the coefficient sweep and ties the highest MATH500 accuracy. These sweeps verify that the headline results do not rely on per-task tuning: the fixed settings are used unchanged across every backbone and domain, and the measured variation around them is small.

A.2 Sensitivity of Entropy-Selection Coefficients

Table 5: **Post-hoc WD-DS coefficient sensitivity.** The weights α, β, and γ multiply the negative entropy difference, current entropy, and positive entropy difference, respectively. These sweeps are diagnostic and are not used to select the headline models in Table 1. Values are accuracies (%); "-" indicates no correct AIME2025 answer under that configuration.

Setup	Coefficients			Qwen3-4B-Base		Qwen2.5-3B	
	α	β	γ	AIME 2025	MATH 500	AIME 2025	MATH 500
Config A	0.2	0.7	0.1	10.0	**71.4**	-	44.6
Default	**0.1**	**0.8**	**0.1**	**16.7**	**71.4**	-	**48.2**
Config B	0.2	0.6	0.2	10.0	67.8	-	42.0
Config C	0.1	0.9	0.0	10.0	70.2	-	47.6
Config D	0.0	1.0	0.0	10.0	68.2	-	47.0
Config E	0.05	0.85	0.1	13.3	69.4	-	47.2

Table 5 shows that the fixed WD-DS weights perform strongly across both backbones. On Qwen3-4B-Base, the default gives the highest AIME2025 accuracy (16.7%) and ties the highest MATH500 accuracy (71.4%). On Qwen2.5-3B, it achieves the highest MATH500 accuracy (48.2%), compared with 47.0% when both historical entropy terms are removed (Config D).

A.3 Sensitivity Analysis on Mixing Coefficient λ

After fixing $\lambda = 0.5$ for the main experiments, we ran a post-hoc sensitivity sweep on Qwen3-4B-Base with training seed 42 and evaluated each setting using mean@3 accuracy. The six-benchmark mean ranges from 69.7% to 75.5% for $\lambda \in [0.4, 0.8]$ (Table 6); it peaks at $\lambda = 0.42$ (75.5%), while the main-experiment setting $\lambda = 0.5$ achieves 74.4% in this sweep.

Table 6: **Post-hoc sensitivity to the mixing coefficient λ.** Mean@3 accuracy (%) on Qwen3-4B-Base trained with seed 42. The main experiments fix $\lambda = 0.5$; this sweep peaks at $\lambda = 0.42$.

Dataset	0.1	0.2	0.3	0.4	**0.42**	0.44	0.46	0.48	0.5
AIME25	7.8	10.0	12.2	16.7	**20.0**	20.0	16.7	20.0	20.0
MATH500	71.4	67.1	70.8	73.1	**74.9**	74.5	73.2	73.3	73.3
AQUA	77.2	72.8	74.7	75.2	**73.9**	73.1	72.4	71.8	73.9
GSM8K	88.2	89.1	89.4	91.9	**91.8**	91.7	91.0	91.4	91.0
MAWPS	95.0	96.6	95.5	97.8	**98.2**	97.8	97.8	98.0	95.7
SVAMP	90.4	91.8	93.3	94.9	**94.1**	93.6	93.5	92.9	92.6
Avg.	71.7	71.2	72.7	74.9	**75.5**	75.1	74.1	74.6	74.4

Dataset	0.52	0.54	0.56	0.58	0.6	0.7	0.8	0.9
AIME25	21.1	16.7	16.7	16.7	10.0	20.0	20.0	12.2
MATH500	70.7	74.0	73.0	73.3	68.9	71.7	70.9	66.1
AQUA	71.0	71.3	71.9	70.2	64.8	66.9	61.4	60.8
GSM8K	91.2	91.4	90.6	90.9	89.2	89.3	88.1	85.7
MAWPS	96.5	97.2	96.8	97.5	94.7	96.2	95.5	93.6
SVAMP	92.4	93.5	92.8	92.9	90.8	91.9	89.6	87.4
Avg.	73.8	74.0	73.6	73.6	69.7	72.7	70.9	67.6

A.4 Pass@k Comparison with DPO

Using the same SFT warm-start and math data, we train DPO on pairs formed from the first correct and incorrect responses among four rollouts per prompt ($T = 0.8$), verified by `math_verify`. Prompts without a valid pair are omitted. We evaluate both methods with 64 samples per problem ($T = 0.7$, top-$p = 0.95$) using the unbiased pass@k estimator (Chen et al., 2021).

Table 7: **Pass@k evaluation of WMSS vs. DPO on Qwen3-4B-Base.** Sampling at $T = 0.7$, top-$p = 0.95$, $n = 64$ samples per problem; the k=1 entry is equivalent to mean@64.

k	AIME 2025		AMC23		MATH500	
	WMSS	DPO	WMSS	DPO	WMSS	DPO
1	**17.97**	14.20	**49.49**	42.80	**75.32**	70.90
4	**24.69**	22.40	**63.25**	56.70	**84.36**	84.00
8	**26.59**	24.90	**68.06**	61.00	87.20	**87.40**
16	**28.57**	27.30	**71.79**	65.20	89.48	**89.90**
32	**30.83**	29.10	**75.13**	69.80	91.49	**91.80**
64	**33.33**	30.00	**77.50**	75.00	93.20	**93.40**

WMSS outperforms DPO on AIME 2025 and AMC23 at every k in Table 7. At $k = 1$, the scores are 17.97 vs. 14.20 and 49.49 vs. 42.80; the lead remains at $k = 64$ (33.33 vs. 30.00 and 77.50 vs. 75.00). On MATH500, WMSS leads at $k = 1, 4$; DPO is ahead by at most 0.42 points for $k \geq 8$.

B Supplementary Theory: Margin Compensation and Gradient Allocation

This appendix proves the statements of Section 5, records the conditional amplification criteria and a counterexample, and states consequences for direct logit descent, for the retained gains, and for the population optimum. The analysis is pointwise at a fixed context and target. Both branches have finite logits and disjoint parameter sets; $\lambda \in (0, 1)$ is a fixed scalar. Comparisons use the same strong parameters and the same token-loss normalization; ℓ_{JT} is the cross-entropy on mixed logits without division by λ.

B.1 Chain rule, margin form, and compensation

Let $p = P_{\mathrm{strong}}$, $u = P_{\mathrm{weak}}$, and $r = P_{\mathrm{mix}}$. For $z_{\mathrm{mix}} = \lambda z_{\mathrm{strong}} + (1 - \lambda) z_{\mathrm{weak}}$, cross-entropy gives $\nabla_{z_{\mathrm{mix}}} \ell_{\mathrm{JT}} = r - e_y$, and $\partial z_{\mathrm{mix}} / \partial z_{\mathrm{strong}} = \lambda I$ gives Eq. (10). The historical branch need not be frozen over the whole run for this local identity; it must only have no derivative path through the strong parameters. Dividing the softmax denominator by $e^{z_{\mathrm{mix}}[y]}$,

$$\ell_{\mathrm{JT}} = \log\Big(1 + \sum_{k \neq y} e^{-\lambda m_{s,k} - (1-\lambda) m_{w,k}}\Big), \qquad z_{\mathrm{mix}}[y] - z_{\mathrm{mix}}[k] = \lambda m_{s,k} + (1 - \lambda) m_{w,k}. \tag{13}$$

Proof of Proposition 1. If $\ell_{\mathrm{JT}} \leq \varepsilon$ then $\sum_{k \neq y} e^{-\lambda m_{s,k} - (1-\lambda) m_{w,k}} \leq e^{\varepsilon} - 1$. Every summand is nonnegative, hence at most the right-hand side; taking logarithms gives $\lambda m_{s,k} \geq b_\varepsilon - (1 - \lambda) m_{w,k}$, which rearranges to the claim. The same argument applied to $\ell_{\mathrm{SFT}} \leq \varepsilon$ gives $m_{s,k} \geq b_\varepsilon$. With several non-target candidates these are necessary, not sufficient, coordinatewise inequalities. □

B.2 Probability identities

Softmax invariance to additive constants yields the normalized geometric mixture

$$r_k = \frac{p_k^{\lambda} u_k^{1-\lambda}}{\sum_j p_j^{\lambda} u_j^{1-\lambda}}. \tag{14}$$

The mixed margin is $m_{m,k} = \lambda m_{s,k} + (1 - \lambda) m_{w,k} = m_{s,k} - (1 - \lambda)\Delta_k$, so $r_k / r_y = (p_k / p_y) e^{(1-\lambda)\Delta_k}$. Summing over $k \neq y$ gives $(1 - r_y)/r_y = \rho\,(1 - p_y)/p_y$, and solving,

$$r_y = \frac{p_y}{D}, \qquad r_k = \frac{p_k e^{(1-\lambda)\Delta_k}}{D}, \qquad 1 - r_y = \frac{(1 - p_y)\rho}{D}, \qquad \tilde{q}_k = \frac{q_k e^{(1-\lambda)\Delta_k}}{\rho}. \tag{15}$$

Taking $-\log$ of $r_y = p_y / D$ proves the loss identity $\ell_{\mathrm{JT}} = \ell_{\mathrm{SFT}} + \log D$ of Section 5. Since D and Δ depend on the current strong logits, holding them fixed while differentiating would not reproduce WD-JT.

B.3 Proof of Theorem 2

Decomposition. Using $p - e_y = (1 - p_y)(q - e_y)$ and $r - e_y = (1 - r_y)(\tilde{q} - e_y)$,

$$g_{\mathrm{JT}} = \lambda(1 - r_y)(\tilde{q} - e_y) = \frac{\lambda(1 - r_y)}{1 - p_y}(p - e_y) + \lambda(1 - r_y)(\tilde{q} - q), \tag{16}$$

and Eq. (15) gives $w = (1 - r_y)/(1 - p_y) = \rho/D$. Both conditional distributions sum to one with zero target coordinate, so their difference has zero sum and zero target coordinate. Since $D = p_y + (1-p_y)\rho$ and $p_y > 0$, $w > 1 \iff \rho > p_y + (1-p_y)\rho \iff p_y(\rho - 1) > 0 \iff \rho > 1$. The absolute target coordinate and the sum of non-target coordinates both equal $\lambda(1 - r_y)$ for g_{JT} and $1 - p_y$ for g_{SFT}, so $S = \|g_{\mathrm{JT}}\|_1 / \|g_{\mathrm{SFT}}\|_1$; this is not in general an ℓ_2 or parameter-gradient norm ratio.

Scalar case. If $g_{\mathrm{JT}} = c\, g_{\mathrm{SFT}}$, the target coordinates force $c = S$ and Eq. (16) forces $\tilde{q} = q$. Since every $q_k > 0$, Eq. (15) then gives $e^{(1-\lambda)\Delta_k} = \rho$ for all $k \neq y$, i.e. constant Δ_k. The converse follows by substitution.

Variational characterization and monotone allocation. For any distribution v on the non-target candidates, Eq. (15) gives

$$(1 - \lambda)\, \mathbb{E}_v[\Delta] - \mathrm{KL}(v \| q) = \log \rho - \mathrm{KL}(v \| \tilde{q}), \tag{17}$$

so the variational objective of Theorem 2 is uniquely maximized at $v = \tilde{q}$. For any nondecreasing f and $a_k = e^{(1-\lambda)\Delta_k}$,

$$\mathbb{E}_{\tilde{q}}[f(\Delta)] - \mathbb{E}_q[f(\Delta)] = \frac{\mathrm{Cov}_q\left(f(\Delta), a\right)}{\rho}, \tag{18}$$
$$\mathrm{Cov}_q\left(f(\Delta), a\right) = \frac{1}{2} \sum_{k,l \neq y} q_k q_l \big[f(\Delta_k) - f(\Delta_l)\big]\big[a_k - a_l\big] \geq 0.$$

Taking $f(t) = \mathbf{1}\{t \geq h\}$ gives $\tilde{q}(H) \geq q(H)$ for $H = \{\Delta_k \geq h\}$; when both H and its complement have positive q-mass, every cross pair has strictly ordered gaps and the inequality is strict. Taking $f(t) = t$ gives $\mathbb{E}_{\tilde{q}}[\Delta] \geq \mathbb{E}_q[\Delta]$, strictly unless all gaps are equal. Pairwise, $(\tilde{q}_k/\tilde{q}_l)/(q_k/q_l) = e^{(1-\lambda)(\Delta_k - \Delta_l)}$.

B.4 The re-exposure gate and its shape

Write $O_{\mathrm{s}} = (1-p_y)/p_y$ and $O_{\mathrm{w}} = (1-u_y)/u_y$ for the odds against the target. Since $e^{-m_{s,k}} = O_{\mathrm{s}} q_k$ and $e^{-m_{w,k}} = O_{\mathrm{w}} \bar{u}_k$, the mixed margin gives

$$\rho = \sum_{k \neq y} q_k \Big(\frac{O_{\mathrm{w}} \bar{u}_k}{O_{\mathrm{s}} q_k}\Big)^{1-\lambda} = \Big(\frac{O_{\mathrm{w}}}{O_{\mathrm{s}}}\Big)^{1-\lambda} \sum_{k \neq y} q_k^{\lambda} \bar{u}_k^{1-\lambda} = \Big(\frac{O_{\mathrm{w}}}{O_{\mathrm{s}}}\Big)^{1-\lambda} e^{-(1-\lambda) D_\lambda(q \| \bar{u})}, \tag{19}$$

where $D_\lambda(q\|\bar{u}) = \frac{1}{\lambda - 1} \log \sum_{k \neq y} q_k^{\lambda} \bar{u}_k^{1-\lambda} \geq 0$ is the Rényi divergence of order λ, zero only when $q = \bar{u}$. Because $\log O = -\operatorname{logit} P(y)$, equivalently $\log \rho = (1-\lambda)\big[\operatorname{logit} p_y - \operatorname{logit} u_y - D_\lambda(q\|\bar{u})\big]$. Two consequences follow. First, $D_\lambda \geq 0$ gives $\rho \leq (O_{\mathrm{w}}/O_{\mathrm{s}})^{1-\lambda}$ (Hölder's inequality), so $u_y \geq p_y$ implies $\rho \leq 1$ and no re-exposure; the historical model must be less confident on the target. Second, Jensen's inequality on $t \mapsto e^{(1-\lambda)t}$ gives $\rho \geq e^{(1-\lambda)\mathbb{E}_q[\Delta]}$, so $\mathbb{E}_{k \sim q}[\Delta_k] > 0$ suffices for re-exposure; the uniform condition $\Delta_k \geq 0$ for all k is the special case in which every gap is nonnegative.

For the shape in p_y, Eq. (15) gives $p_y - r_y = p_y(1 - D^{-1}) = p_y(1 - p_y)(\rho - 1)/D$. For fixed $\rho > 1$, differentiating $p(1-p)/(\rho - (\rho - 1)p)$ in p and setting the numerator to zero gives $(\rho - 1)p^2 - 2\rho p + \rho = 0$, whose root in $(0, 1)$ is $p^* = \sqrt{\rho}/(1 + \sqrt{\rho})$; the drop vanishes at both endpoints. Meanwhile $w = \rho/D \to \rho$ as $p_y \to 1$, so the ratio w stays large at saturated positions while the absolute drop vanishes; the bounded drop, not the ratio, measures how much residual a position actually gains.

B.5 Absolute amplification is conditional

Proposition 4 (Coordinate and total amplification). *For $k \neq y$,*

$$R_k := \frac{|g_{\mathrm{JT},k}|}{|g_{\mathrm{SFT},k}|} = \lambda \frac{r_k}{p_k} = \frac{\lambda e^{(1-\lambda)\Delta_k}}{D} = S\,\frac{\tilde{q}_k}{q_k}, \qquad R_k > 1 \iff (1-\lambda)\Delta_k > \log D - \log \lambda, \tag{20}$$

and $S > 1$ exactly when $p_y > 1-\lambda$ and $\rho > p_y/(p_y + \lambda - 1)$.

Proof. $g_{\mathrm{JT},k} = \lambda r_k$ and $g_{\mathrm{SFT},k} = p_k$; substitute Eq. (15) and take logarithms. For the total, $S > 1 \iff \lambda\rho > p_y + (1-p_y)\rho \iff (p_y + \lambda - 1)\rho > p_y$; since $p_y, \rho > 0$, this holds exactly under the two stated conditions. At $\lambda = 1$, $r = p$ and all ratios equal one; at $\lambda = 0$ the strong branch receives zero gradient. □

For any nonempty proper subset H of non-target candidates, $R_H := \lambda \sum_{k\in H} r_k / \sum_{k\in H} p_k = S\,E_H$ with $E_H = \tilde{q}(H)/q(H) = \mathbb{E}_q[e^{(1-\lambda)\Delta} \mid H]/\rho$; enrichment $E_H > 1$ holds iff the conditional mean of $e^{(1-\lambda)\Delta}$ on H exceeds that on its complement, so a set chosen by historical top-K probability is not automatically enriched. At $\lambda = 0.5$, $R_k > 1$ requires $r_k > 2p_k$, and no position with $p_y \leq 0.5$ has $S > 1$. A binary example separates the three properties: with target first, $p = (0.9, 0.1)$, $u = (0.8, 0.2)$, $\lambda = \frac{1}{2}$ give $r = (6/7, 1/7)$, so the margin gap $\log 9 - \log 4$ is positive, the mixed residual on the negative token grows by $10/7$, yet $R_2 = S = 5/7$. Larger mixed residuals, enriched allocation, and larger strong-branch gradients are therefore distinct properties.

B.6 History versus temperature and label smoothing

If $z_{\mathrm{weak}} = c\mathbf{1}$, then $r^{\mathrm{temp}} = \mathrm{Softmax}(\lambda z_{\mathrm{strong}})$ and $\tilde{q}_k \propto p_k^{\lambda}$, the temperature-$1/\lambda$ transform ($T = 2$ at $\lambda = 0.5$). By Eq. (14), $r_k/r_l = (p_k/p_l)^{\lambda}(u_k/u_l)^{1-\lambda}$ and $r_k^{\mathrm{temp}}/r_l^{\mathrm{temp}} = (p_k/p_l)^{\lambda}$, which proves the temperature identity of Section 5. If $u_k \propto p_k^a$ then $r_k \propto p_k^{\lambda + a(1-\lambda)}$, a power transform of the current prediction, so history is indistinguishable from temperature exactly when the historical log-probabilities are an affine function of the current ones. For a uniform label distribution $\mathbf{v}$ and smoothing mass ϵ, label-smoothed SFT has gradient $p - e_y + \epsilon(e_y - \mathbf{v})$, a position-independent additive term. Both references are fixed functions of the current prediction; a historical reference replaces them by margin gaps from a second predictive state.

B.7 Parameter gradients and normalization

For fixed nonnegative token weights c_i with $\sum_i c_i = 1$, let $\mathcal{L}_{\mathrm{JT}} = \sum_i c_i \ell_{\mathrm{JT},i}$ and $J_i = \partial z_{\mathrm{strong}}(x_i)/\partial\theta$. Applying the chain rule position by position,

$$\nabla_\theta \mathcal{L}_{\mathrm{JT}} = \lambda \sum_i c_i w_i J_i^\top g_{\mathrm{SFT},i} + \lambda \sum_i c_i (1 - r_{y_i}) J_i^\top (\tilde{q}_i - q_i). \tag{21}$$

The leading λ multiplies every position and is absorbed into the effective step size; the per-position factors w_i and the tilt terms decide where the update concentrates. A fixed shared averaging factor cancels in all ratios above, whereas dividing only ℓ_{JT} by λ would change the compared objective and remove the leading λ in S and R_k; Eqs. (5) and (10) use the unscaled mixed cross-entropy. A model-dependent differentiable λ would add chain-rule terms outside this fixed-coefficient setting. All statements depend on λ through $(1-\lambda)\Delta_k$: as $\lambda \to 1$, $\rho \to 1$, $w \to 1$ and $\tilde{q} \to q$, recovering SFT; as λ decreases, the re-exposure bound $(O_{\mathrm{w}}/O_{\mathrm{s}})^{1-\lambda}$ and the pairwise reallocation $e^{(1-\lambda)(\Delta_k - \Delta_l)}$ both grow while the allocation departs from the current competition structure q. WD-DS changes the examples entering this aggregate; the identities characterize WD-JT conditional on a selected context, and Appendix C relates the WD-DS statistics to these identities.

B.8 Margin changes and standalone loss under direct logit descent

This subsection assumes freely optimized logits for one fixed example and is not a statement about a shared network. A step $z_{\mathrm{strong}}^{+} = z_{\mathrm{strong}} - \eta g_{\mathrm{JT}}$ changes $z_{\mathrm{strong}}[y]$ by $\eta\lambda(1 - r_y)$ and $z_{\mathrm{strong}}[k]$ by

$-\eta\lambda r_k$, so every target margin changes by

$$m^+_{s,k} - m_{s,k} = \eta\lambda\big[(1-r_y)+r_k\big] = \eta\lambda(1-r_y)(1+\tilde{q}_k) > 0, \qquad \frac{\Delta m^{\mathrm{JT}}_{s,k}}{\Delta m^{\mathrm{SFT}}_{s,k}} = S\,\frac{1+\tilde{q}_k}{1+q_k}, \quad (22)$$

where the SFT step of the same size changes the margin by $\eta(1-p_y)(1+q_k)$. Every margin increases and the standalone target probability rises. Normalizing the joint direction by S, so that both steps carry the same ℓ_1 correction, leaves the ratio $(1+\tilde{q}_k)/(1+q_k)$, which exceeds one exactly when $\tilde{q}_k > q_k$, i.e. $\Delta_k > \log\rho/(1-\lambda)$ by Eq. (15); the normalized step therefore changes the margin against k by $\eta(1-p_y)(\tilde{q}_k - q_k)$ more than the SFT step, and the differences sum to zero over k: history redirects the strong branch's margin growth toward candidates with above-average historical gap and away from the others.

For the standalone loss, the first-order decrease along $-g$ is $\mathcal{I}(g) = -\frac{d}{d\eta}\ell_{\mathrm{SFT}}(z_{\mathrm{strong}} - \eta g)|_{\eta=0} = \langle g_{\mathrm{SFT}}, g\rangle$. Expanding the inner products with $g_{\mathrm{SFT}} = (1-p_y)(q-e_y)$ and $g_{\mathrm{JT}} = \lambda(1-r_y)(\tilde{q}-e_y)$,

$$\mathcal{I}(g_{\mathrm{JT}}) = \lambda(1-p_y)(1-r_y)\big(1+\langle q,\tilde{q}\rangle\big) > 0, \qquad \frac{\mathcal{I}(g_{\mathrm{JT}})}{\mathcal{I}(g_{\mathrm{SFT}})} = S\,\frac{1+\langle q,\tilde{q}\rangle}{1+\|q\|_2^2}. \quad (23)$$

The joint step always decreases the standalone loss to first order. At a fixed ℓ_2 logit-update norm, $-g_{\mathrm{SFT}}$ is its steepest-descent direction: Cauchy–Schwarz gives $\langle g_{\mathrm{SFT}}, g\rangle \le \|g_{\mathrm{SFT}}\|_2\|g\|_2$, with equality for nonzero g exactly when g is a positive multiple of g_{SFT}. In particular,

$$\frac{\mathcal{I}(g_{\mathrm{JT}}/\|g_{\mathrm{JT}}\|_2)}{\mathcal{I}(g_{\mathrm{SFT}}/\|g_{\mathrm{SFT}}\|_2)} = \frac{1+\langle q,\tilde{q}\rangle}{\sqrt{(1+\|q\|_2^2)(1+\|\tilde{q}\|_2^2)}} \le 1.$$

The scale S, however, is the ℓ_1 norm ratio. At equal ℓ_1 correction mass ($S=1$), Eq. (23) exceeds one exactly when $\langle q, \tilde{q}-q\rangle > 0$. For example, with target $y=1$, $\lambda = 1/2$, $p = (0.8, 0.16, 0.04)$, and $u = (45, 81, 4)/130$, the geometric mixture is $r = (0.6, 0.36, 0.04)$. Its non-target allocations are $(0.8, 0.2)$ and $(0.9, 0.1)$, giving $S = 1$ but $\mathcal{I}(g_{\mathrm{JT}})/\mathcal{I}(g_{\mathrm{SFT}}) = 29/28 > 1$. Thus the unnormalized descent rate depends on both scale and allocation; at fixed ℓ_1 mass, allocation can increase or decrease it. History also directs margin growth through Eq. (22) and changes the required margins in Proposition 1, with retained gains characterized by Proposition 3.

B.9 Proof of Proposition 3

From $m^+_{\mathrm{mix},k} = \lambda m^+_{s,k} + (1-\lambda)m_{w,k}$, solve $m^+_{s,k} = \big(m^+_{\mathrm{mix},k} - (1-\lambda)m_{w,k}\big)/\lambda$ and subtract $m^0_{s,k} = \big(\lambda m^0_{s,k} + (1-\lambda)m^0_{s,k}\big)/\lambda$:

$$m^+_{s,k} - m^0_{s,k} = \frac{m^+_{\mathrm{mix},k} - m^0_{s,k}}{\lambda} + \frac{(1-\lambda)\big(m^0_{s,k} - m_{w,k}\big)}{\lambda}, \quad (24)$$

which is Eq. (12). The identity uses the historical margin at the end of the interval; a frozen reference is the special case in which it equals the historical margin at the start. Under the recovery condition $m^+_{\mathrm{mix},k} \ge m^0_{s,k} - \varepsilon_k$, the first term is at least $-\varepsilon_k/\lambda$, giving the stated bound, which is positive when $\varepsilon_k < (1-\lambda)\Delta^0_k$. Since $p_y = \big(1+\sum_{k\neq y} e^{-m_{s,k}}\big)^{-1}$ is strictly increasing in every margin, nondecreasing margins with one strict increase give $p^+_y > p^0_y$ and hence a strictly smaller ℓ_{SFT}. At the start of the interval the mixed margin equals $m^0_{s,k} - (1-\lambda)\Delta^0_k$, below the initial strong margin whenever $\Delta^0_k > 0$, so the recovery condition asks only that training close the gap opened by mixing on that candidate. The proposition assumes nothing about whether a network attains the condition; it states what the strong branch alone must have gained if it does.

B.10 Population optimum and the one-hot case

Fix a context with conditional label distribution μ, a fixed weak reference u, and an unrestricted prediction p. The expected mixed loss is $\mathcal{L}_{\mathrm{JT}}(p) = H(\mu) + \mathrm{KL}(\mu\|r(p))$. The map $p \mapsto r(p)$ in Eq. (14) is a bijection of the interior simplex with inverse $p_k \propto r_k^{1/\lambda} u_k^{-(1-\lambda)/\lambda}$, so for strictly positive μ and u the unique optimum satisfies $r = \mu$ and

$$p^*_k = \frac{\mu_k^{1/\lambda} u_k^{-(1-\lambda)/\lambda}}{\sum_j \mu_j^{1/\lambda} u_j^{-(1-\lambda)/\lambda}}, \qquad \mathcal{L}_{\mathrm{SFT}}(p^*) - \mathcal{L}_{\mathrm{SFT}}(\mu) = \mathrm{KL}(\mu\|p^*) > 0 \text{ unless } u = \mu. \quad (25)$$

The optimum is tilted *against* the historical reference: the strong branch is asked to be sharper than the label distribution exactly where history places mass. This is the mechanism of Proposition 1 at the level of the optimum, and it is also its cost: with soft labels the standalone strong model would be miscalibrated by $\mathrm{KL}(\mu\|p^*)$. WD-JT trains on one-hot targets, for which $\mu = e_y$ lies on the boundary; there $\mathcal{L}_{\mathrm{JT}} \to 0$ requires $\lambda m_{s,k} + (1-\lambda) m_{w,k} \to \infty$, i.e. $m_{s,k} \to \infty$ for every $k \neq y$, the same limit as SFT, and the relaxation of Proposition 1 at candidates with large historical margin does not change the standalone optimum. The tilt acts along the trajectory, concentrating the finite correction where history was confused, rather than at the endpoint.

B.11 Scale Factors from Stored Forward Probes

We recompute the exact chain-rule factors from an existing Qwen3-4B forward probe with $\lambda = 0.5$: the weak logits come from the base model and the strong logits from its SFT checkpoint at step 218. The probe uses the same teacher-forced contexts for both models on 1,000 mathematical training completions, with 64 evenly spaced positions per completion and a maximum sequence length of 2,048. Model forwards use BF16 and probability calculations use FP32. This is a fixed-checkpoint evaluation of the local WD-JT objective.

The stored target probabilities determine S. The records also contain conditional non-target mass for H_8, the historical model's top-eight non-target candidates, allowing the group ratio

$$R_{H_8} = \frac{\lambda \sum_{k \in H_8} r_k}{\sum_{k \in H_8} p_k} = S \frac{\tilde{q}(H_8)}{q(H_8)}. \tag{26}$$

We exclude 352 of 64,000 positions with $p_y \geq 1 - 10^{-7}$ to avoid ratios with a numerically saturated denominator. The target-drop subset uses $r_y < p_y - 10^{-10}$ and retains 43,206 positions after the same exclusion. Table 8 reports descriptive pointwise statistics from these stored probabilities.

Table 8: **Strong-branch logit-gradient scales at $\lambda = 0.5$.** S measures total correction; R_{H_8} measures summed correction on a historical candidate set. Percentages count ratios strictly greater than one.

Probe subset	Positions	Median S	$S > 1$ (%)	Median R_{H_8}	$R_{H_8} > 1$ (%)
All non-saturated	63,648	0.607	27.6	0.641	30.3
Target probability drops	43,206	0.847	40.6	0.901	44.3

Both attenuated and amplified strong-branch corrections occur in this probe. Even among positions where the target probability drops, the median total scale is below one; the chain-rule coefficient is therefore consequential for interpreting the correction. The historical-set statistic concerns a sum over candidates. Individual R_k values require aligned candidate probabilities, which are not retained in this record. Parameter updates across training require the corresponding Jacobians and optimizer state.

C Supplementary Theory: Entropy Selection and Residual Re-exposure

This appendix links the two modules of WMSS. It identifies the part of the WD-JT correction that the WD-DS statistics of Eq. (8) track, gives the exact condition under which a historical entropy difference marks re-exposure, and shows how sampling turns the score into retained correction. We keep the pointwise setting and notation of Appendix B, and write h_b for the binary entropy, σ for the logistic function, $a_s = \operatorname{logit} p_y$ and $a_w = \operatorname{logit} u_y$ for the target log-odds against all other tokens, $K_s = H(q)$ and $K_w = H(\bar{u})$ for the non-target entropies, and $E = H(u) - H(p)$ for the per-position entropy difference, whose average over the target positions of x_i is $-\Delta H_i$.

What selection has to track. The strong-branch correction $g_{\mathrm{JT}} = \lambda(r - e_y)$ has ℓ_1 mass $2\lambda(1 - r_y)$, and Eq. (15) splits the mixed residual into

$$1 - r_y = \underbrace{(1 - p_y)}_{\text{current residual}} + \underbrace{(p_y - r_y)}_{\text{re-exposed residual}}, \qquad p_y - r_y = \frac{p_y(1-p_y)(\rho - 1)}{D}. \tag{27}$$

The entropy chain rule $H(p) = h_b(p_y) + (1 - p_y)K_s$ separates the target-versus-rest uncertainty, which carries the residual that w rescales, from the candidate spread K_s, which the allocation $\tilde{q}$ redistributes (Theorem 2); moreover, $H(p) = \mathbb{E}_{y' \sim p}[\ell_{\mathrm{SFT}}(y')]$ is the strong model's expected standalone loss under its own prediction. WD-DS therefore connects to WD-JT when $H(p)$ tracks $1 - p_y$ and E tracks $p_y - r_y$.

Entropy difference as a change in target log-odds. With $F_K(a) = h_b(\sigma(a)) + (1 - \sigma(a))K$, the chain rule gives $H(p) = F_{K_s}(a_s)$ and $H(u) = F_{K_w}(a_w)$, and $h_b' = -\operatorname{logit}$ gives $F_K'(a) = -\sigma(a)(1 - \sigma(a))(a + K)$.

Lemma 5 (Exact decomposition of the entropy difference)**.** *Let* $a_t = a_w + t(a_s - a_w)$ *and* $\kappa = \int_0^1 \sigma(a_t)(1 - \sigma(a_t))(a_t + K_s)\, dt$*. Then*

$$E = \kappa\,(a_s - a_w) + (1 - u_y)(K_w - K_s), \tag{28}$$

and $\kappa > 0$ *whenever* $\min(p_y, u_y) > (1 + e^{K_s})^{-1}$*, in particular whenever both models give the target probability above* $\frac{1}{2}$*.*

Proof. Split $E = [F_{K_w}(a_w) - F_{K_s}(a_w)] + [F_{K_s}(a_w) - F_{K_s}(a_s)]$. The first bracket is $(1 - u_y)(K_w - K_s)$; the second is $\int_{a_w}^{a_s} \sigma(a)(1 - \sigma(a))(a + K_s)\, da = \kappa(a_s - a_w)$, whose integrand is positive on the segment when $\min(a_s, a_w) > -K_s$, i.e. $\min(p_y, u_y) > \sigma(-K_s)$. □

The identity holds for any two checkpoints: the entropy difference is a κ-weighted change in target log-odds plus a change in candidate spread.

Entropy difference and re-exposure. Substituting Eq. (28) into $\log \rho = (1 - \lambda)[a_s - a_w - D_\lambda(q\|\bar{u})]$ from Eq. (19) gives the bridge.

Proposition 6 (Historical entropy and re-exposure)**.** *If* $\kappa > 0$*, then*

$$\log \rho = \frac{1 - \lambda}{\kappa} \Big[E - (1 - u_y)(K_w - K_s) - \kappa\, D_\lambda(q\|\bar{u}) \Big], \tag{29}$$

so $r_y < p_y \iff w > 1 \iff E > (1 - u_y)(K_w - K_s) + \kappa\, D_\lambda(q\|\bar{u})$*. If the candidate structure is stable* ($q = \bar{u}$)*, then* $\log \rho = (1 - \lambda)E/\kappa$ *and* $H(u) > H(p) \iff r_y < p_y$*.*

Proof. Solve Eq. (28) for $a_s - a_w$ and substitute; $\rho > 1 \iff w > 1 \iff r_y < p_y$ by Theorem 2. If $q = \bar{u}$, then $K_s = K_w$ and $D_\lambda(q\|\bar{u}) = 0$. □

A higher historical entropy thus marks re-exposure exactly when the excess is not explained by a more dispersed historical candidate set or by a candidate mismatch; under a stable candidate structure the entropy drop alone decides, and $\log \rho$ is proportional to it. The certified quantity is the residual ratio w; the gradient carries the additional factor λ (Appendix B.5).

Reading the selection score. Under a stable candidate structure with $K = K_s = K_w$ and $a_s + K > 0$, expanding Eq. (27) with $\rho - 1 = (1 - \lambda)(a_s - a_w) + O((a_s - a_w)^2)$ and $\kappa = p_y(1 - p_y)(a_s + K) + O(a_s - a_w)$ gives

$$\begin{aligned} 1 - r_y &= (1 - p_y) + \frac{1 - \lambda}{a_s + K}\, E + O\big((a_s - a_w)^2\big), \\ U := (1 - p_y) + |p_y - r_y| &= (1 - p_y) + \frac{1 - \lambda}{a_s + K}\, |E| + O\big((a_s - a_w)^2\big), \end{aligned} \tag{30}$$

where $U \geq 1 - r_y$ with equality in the re-exposure direction. The default score of Eq. (8) mirrors U term by term. In the region $a_s + K > 0$, $H(p)$ increases with the current residual at fixed K; the α term follows the re-exposed residual; the γ term measures how strongly history lowers the correction signal where entropy rises; and the global weights stand in for the position-dependent factor $(1 - \lambda)/(a_s + K)$. Because $|\Delta H_i| \leq \frac{1}{|\mathcal{T}_i|} \sum_{t \in \mathcal{T}_i} |E_t|$, with equality when all positions shift in the same direction, the example-level score measures the net shift of a sequence and inherits Eq. (30) position by position when the shifts share a sign.

Division of labor. Entropy is target-free, whereas re-exposure depends on the target: $\ell_{\mathrm{JT}} - \ell_{\mathrm{SFT}} = \log D = (1-\lambda)\big[\log(p_y/u_y) - D_\lambda(p\|u)\big]$, with $D_\lambda(p\|u)$ the order-λ Rényi divergence of the full predictions, so the gate opens on the level set $\{y : p_y/u_y > e^{D_\lambda(p\|u)}\}$. For $p \neq u$ this set is a nonempty proper subset of the vocabulary, since $\mathbb{E}_{y\sim p}[\log(p_y/u_y)] = \mathrm{KL}(p\|u) > D_\lambda(p\|u)$ while $\sum_y p_y = \sum_y u_y$. Entropy also does not register confident disagreement: $p = (1-\epsilon, \epsilon)$, $u = (\epsilon, 1-\epsilon)$, $y = 1$, and $\lambda = \frac{1}{2}$ give equal entropies that vanish as $\epsilon \to 0$, yet $r = (\frac{1}{2}, \frac{1}{2})$ and $\|g_{\mathrm{JT}}\|_1 = \frac{1}{2}$. WD-DS therefore decides which examples carry supervised demand, and within each selected sequence WD-JT applies the correction that the target and the historical margin gaps determine through ρ and $\tilde{q}$.

Retained correction under sampling. At a fixed model state, let $C_i = \frac{1}{|\mathcal{T}_i|}\sum_{t\in\mathcal{T}_i} 2\lambda\big(1 - r_{i,t}[y_{i,t}]\big)$ be the average logit-level correction mass of example i.

Proposition 7 (Retained correction)**.** *With π_i from Eq. (9) and $\bar{\pi} = \frac{1}{N}\sum_i \pi_i$, so that uniform inclusion at rate $\bar{\pi}$ has the same expected active-set size,*

$$\mathbb{E}\Big[\sum_{i\in A} C_i \,\Big|\, \mathbf{p}\Big] - \bar{\pi}\sum_{i=1}^{N} C_i = N\,\mathrm{Cov}_{i\sim\mathrm{Unif}\{1,\dots,N\}}(\pi_i, C_i), \tag{31}$$

which is nonnegative whenever s_i and C_i are similarly ordered, since π_i increases with s_i.

Proof. $\mathbb{E}[\sum_{i\in A} C_i \mid \mathbf{p}] = \sum_i \pi_i C_i$, and $\sum_i(\pi_i - \bar{\pi})C_i = \sum_i(\pi_i - \bar{\pi})(C_i - \bar{C})$ with $\bar{C} = \frac{1}{N}\sum_i C_i$; nonnegativity is Chebyshev's sum inequality. □

At equal expected budget, WD-DS retains more fixed-state correction than uniform inclusion exactly when inclusion co-varies with correction mass, and Eq. (30) gives the regime in which the entropy score follows that mass position by position.

D Data Construction and Training Details

D.1 Evaluation Benchmarks and Baselines

Tasks and evaluation. For mathematical reasoning, we report performance on AIME2025, MATH500, AMC23, AQuA (Ling et al., 2017), GSM8K (Cobbe et al., 2021), MAWPS (Koncel-Kedziorski et al., 2016), and SVAMP (Patel et al., 2021). For code generation, we evaluate on HumanEval+ (HE+) and MBPP+, the augmented test-case versions of HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021) introduced by EvalPlus (Liu et al., 2023b), together with BigCodeBench (BCB) (Zhuo et al., 2025) and LiveCodeBench (LCB) (Jain et al., 2025) for broader, contamination-resistant coverage. For logical reasoning, we evaluate on LogiQA 2.0 (Liu et al., 2023a) and ReClor (Yu et al., 2020).

Baselines. **Standard SFT** optimizes the conventional next-token prediction objective without discrepancy-based sample selection or logit-space perturbation. **Random** adds zero-mean Gaussian noise to training logits before cross-entropy while retaining the original reference targets. **NEFTune** (Jain et al., 2024) injects random noise into embeddings during training to improve generalization. **SPIN** (Chen et al., 2024) employs a self-play mechanism where the model iteratively distinguishes its own generated responses from human-annotated data, progressively converting a weak policy into a stronger one without external reward models. **SSB** (Mitra & Ulukus, 2025) (Semantic Soft Bootstrapping) is an RL-free self-distillation method that uses the same base model as both teacher and student under different contexts: the teacher, conditioned on a hint trace, generates soft logit targets over the answer, which the student (without hints) learns to match via KL divergence, eliminating the need for preference data or RL training.

D.2 Training Data Sources

Our training mixture covers three domains: mathematical reasoning, code generation, and logical reasoning. All samples use a chat message format (user/assistant turns), with assistant completions

retaining the full <think>-wrapped chain of thought and a structured final answer. We use the full set of verified samples in each domain (Table 9); the math pool is by far the largest ($\sim$5.7$\times$ code, $\sim$7.5$\times$ logic).

Table 9: Training data across the three domains used in our experiments. Each domain uses the full set of verified samples from the corresponding source.

Domain	#Samples	Source
Math	111,657	Full AM-DeepSeek-R1-1.4M (Zhao et al., 2025) math subset, verified by `math_verify`
Code	19,457	AM-DeepSeek-R1-1.4M code subset, execution-verified against the provided unit tests
Logic	14,882	LogiQA 2.0 and ReClor distilled CoT, retained after answer-match verification (Section D.3)

Mathematical reasoning. Sourced from the AM-DeepSeek-R1-1.4M math subset ($\sim$111 K verified examples spanning algebra, geometry, number theory, combinatorics, and competition problems). We retain DeepSeek-R1's full <think> CoT together with the final \boxed{X} answer and discard samples that fail `math_verify` parsing, leaving 111,657 training examples.

Code generation. Sourced from the same corpus, filtered by executing the produced code against the provided unit tests. We retain the <think> reasoning block; this yields 19,457 training examples after execution verification.

Decontamination. We decontaminate every training set against all evaluation benchmarks in Section 6.1, including AIME2025, whose problems could otherwise enter a corpus assembled after the competition. After whitespace and case normalization, we remove any training example whose problem statement exactly matches an evaluation item. For logic, the LogiQA 2.0 `test` and ReClor `val` splits are additionally held out before CoT distillation (Appendix D.3).

D.3 Logic Data Synthesis

The Logic training set is synthesized in two steps from public benchmarks.

Step 1: merging raw questions. We combine the raw splits of LogiQA 2.0 (Liu et al., 2023a) (MRC version, `train`/`dev`/`test`) and ReClor (Yu et al., 2020) (`train`/`val`) into a unified JSON format ($\sim$17,205 items). Before any CoT distillation, the LogiQA 2.0 `test` split (1,572) and the ReClor `val` split (500) are removed from the training-candidate pool and held out for evaluation; we use ReClor `val` because the public ReClor `test` set has no released labels. The remaining training-candidate pool contains 15,133 items.

Step 2: CoT distillation with answer verification. We prompt Qwen3-32B in its `thinking` mode on each training-candidate question to obtain a long-form chain-of-thought answer. A sample is retained only if the extracted prediction matches the ground-truth option. This yields 14,882 verified Logic training samples, corresponding to a 98.3% retention rate after the held-out evaluation splits have already been removed.

D.4 Gemma-3-4B-PT: Tag-Free Preprocessing

In preliminary experiments, Gemma-3-4B-PT did not reliably follow the <think>/<answer> tag convention during training. We therefore adopt a tag-free variant *only for Gemma*.

Tag stripping. Starting from the original CoT outputs, we remove the literal <think>, </think>, <answer>, and </answer> markers while retaining the reasoning text and the inner answer payload. Thus Gemma still trains on tag-free step-by-step solutions ending with \boxed{X} for math, `The answer is (X)` for logic, or Python code blocks for code.

Fallback chat template. Gemma-3-4B-PT ships without a chat template (only the IT variant has one). We inject a Gemma-3-IT-style fallback in which the `assistant` role is mapped to `model`, a leading system message is prepended to the first user turn (rather than forming its own turn), and each turn is wrapped with `<start_of_turn>`/`<end_of_turn>`. These special tokens are already present in the PT vocabulary.

Tagged vs. tag-free system prompts. Qwen-series bases use a tag-style system prompt that instructs the model to emit `<think>···</think>` followed by `<answer>···</answer>`. Gemma uses a tag-free prompt that ends with "Solve the problem step by step. End your solution with the final answer enclosed in `\boxed{}`" (math), "End your response with 'The answer is (X).' " (logic), or "Provide clean, correct Python code" (code). The SPIN and SSB rollout/teacher phases are aligned to the base-appropriate prompt via their `--system-prompt` / `--student-system-prompt` flags so that the self-generated trajectories match the same formatting that the student is trained to emit.

vLLM multimodal wrap for evaluation. Training saves a `Gemma3ForCausalLM` (text-only) checkpoint, while vLLM v0.11 loads Gemma-3 only as `Gemma3ForConditionalGeneration` (multimodal). To make the saved checkpoint loadable for evaluation, we (i) re-key every weight with a `language_model.` prefix and (ii) inject the base model's `vision_tower.*` and `multi_modal_projector.*` weights unchanged. The wrapped checkpoint is functionally identical for text-only decoding but conforms to vLLM's expected module layout.

D.5 Training Configuration

Algorithm 1 Weak Models make Strong Models Stronger

Require: Dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, base model $\mathcal{M}_0$, iterations K, hyperparameters $\alpha, \beta, \gamma, \lambda$

$\mathcal{M}_1 \leftarrow \text{Train}(\mathcal{M}_0, \mathcal{D}, \mathcal{L}_{\text{SFT}})$ ▷ Initialization
for $t = 1$ to K **do**
  Compute token-averaged entropies of $\mathcal{M}_t$ and $\mathcal{M}_{t-1}$ on $\mathcal{D}$ ▷ WD-DS
  $\Delta\mathbf{H} \leftarrow \mathbf{H}(\mathcal{M}_t) - \mathbf{H}(\mathcal{M}_{t-1})$
  Calculate per-draw probabilities $\mathbf{p}$ via Eq. 8
  Draw $J_1, \ldots, J_N \overset{\text{iid}}{\sim} \text{Categorical}(\mathbf{p})$; $A \leftarrow \text{unique}(J_1, \ldots, J_N)$
  $\mathcal{D}_{\text{active}} \leftarrow \{(x_i, y_i) : i \in A\}$; $M \leftarrow |A|$ ▷ Record M and M/N
  $(\mathcal{M}^{(t)}_{\text{weak}}, \mathcal{M}^{(t)}_{\text{strong}}) \leftarrow (\mathcal{M}_{t-1}, \mathcal{M}_t)$ ▷ WD-JT
  for batch (x, y) in one pass over $\mathcal{D}_{\text{active}}$ **do**
    $z_{\text{weak}} \leftarrow \text{Forward}(\mathcal{M}^{(t)}_{\text{weak}}, x)$; $z_{\text{strong}} \leftarrow \text{Forward}(\mathcal{M}^{(t)}_{\text{strong}}, x)$
    $z_{\text{mix}} \leftarrow \lambda z_{\text{strong}} + (1 - \lambda) z_{\text{weak}}$
    Apply the mixed-logit correction $\text{CE}(z_{\text{mix}}, y)$
  end for
  $\mathcal{M}_{t+1} \leftarrow \mathcal{M}^{(t)}_{\text{strong}}$
end for
return $\mathcal{M}_{K+1}$

Implementation details. We use Qwen3-4B-Base, Qwen3-8B-Base (Yang et al., 2025), Qwen2.5-3B, and Gemma-3-4B-PT (Gemma Team, 2025) as backbone models, and fine-tune them on the AM-1.4M dataset (Zhao et al., 2025) for math and code, together with the Logic training set synthesised in Appendix D.3. Our implementation is based on the `TRL` library with the Qwen3 / Gemma architectures from `transformers` (Wolf et al., 2020). The main SFT and WMSS runs use full-parameter fine-tuning with AdamW, learning rate 1×10^{-5}, maximum sequence length 4096, global batch size 512, gradient clipping at 1.0. The longer-SFT run and optimization/softening controls use the same user-only, thinking-enabled greedy evaluation specification as the main comparison. For WMSS, we set $\alpha = 0.1$, $\beta = 0.8$, $\gamma = 0.1$, and $\lambda = 0.5$ in the main runs; Appendix A.1 reports a sensitivity sweep.

Per-domain training budget. All methods reported in Table 1 use the same base models and per-domain training budget. The total training budget is fixed to **2 epochs** for the main SFT-style comparisons, including the SFT baseline and WMSS.

Compute accounting. The two-epoch allocation fixes the number of supervised passes, not the compute. Let $N = |\mathcal{D}|$ and count one forward pass of one model on one example as one unit, with a backward pass costing two units. On the Qwen3-4B-Base math data ($N = 111{,}657$, global batch size 512), two-epoch SFT costs $2 \times 3N = 6N$ units in 436 optimizer steps. WMSS costs $3N$ for the initial SFT epoch, $2N$ for scoring $\mathcal{M}_{\text{weak}}$ and $\mathcal{M}_{\text{strong}}$ on $\mathcal{D}$ in WD-DS, and $4|\mathcal{D}_{\text{active}}|$ for WD-JT, which runs a strong forward and backward pass plus a weak forward pass. Using $123 \times 512 = 62{,}976$ full-batch example slots as a proxy for $M = |\mathcal{D}_{\text{active}}|$ gives about $7.3N$, or roughly $1.2\times$ two-epoch SFT, in $218 + 123 = 341$ optimizer steps. The exact active-set size counts unique sampled indices before batching and is distinct from both the N draws and the expected size $\sum_i \pi_i$ in Eq. (9). This count assumes equal per-example sequence length; WD-DS favors high-entropy examples, whose average length may differ from that of $\mathcal{D}$. The wall-clock times in Section 6.3 are not FLOP-matched.

Training schedule and early stopping. The SFT baseline is trained for 2 epochs. WMSS uses the same 2-epoch total budget: an initial SFT epoch producing $\mathcal{M}_1$, followed by 1 weak-driven epoch using WD-DS and WD-JT. This fixed budget is used consistently for the reported main comparisons rather than selected separately per benchmark. The convergence trajectories in Figure 3 are extended diagnostics beyond the main comparison: further training produces non-monotonic changes on small benchmarks such as AMC23.

D.6 Baseline Hyperparameters

We report the key hyperparameters we used for each baseline; all runs share the same base models and training data as WMSS, and follow the per-domain epoch budget described in Appendix D.5.

Random. We perturb training logits with independent Gaussian noise before applying cross-entropy to the original reference tokens, using a nominal scale of $\sigma = 1.0$. The model-level implementation adds $\sigma\varepsilon$, whereas the forward-hook implementation adds $\sigma\varepsilon/\sqrt{L}$ for sequence length L, with $\varepsilon \sim \mathcal{N}(0, I)$.

NEFTune. We adopt the paper-recommended embedding-noise magnitude $\alpha = 5$.

SPIN. SPIN is DPO-based self-play. Per the 2-epoch allocation in Appendix D.5, we initialise from the SFT-epoch-1 checkpoint and then run 1 epoch of DPO on SPIN-generated preference pairs. Hyperparameters were tuned on a small grid and we ended up using: $\beta = 0.1$ (recommended by SPIN/TRL's DPO and empirically preferred for iterative self-play, as low β allows the student to deviate further from the reference), learning rate 5×10^{-7} (DPO is highly learning-rate sensitive—1×10^{-6} was unstable in our runs), sampling temperature 1.0 (following SPIN §4 for rejected-sample diversity), and gradient-norm clip 0.5 (tighter than the TRL default of 1.0, which we found noticeably improves DPO stability). Following the original SPIN protocol, each iteration trains for exactly 1 epoch before resampling.

SSB. SSB is a self-bootstrapping distillation baseline. For every prompt we first draw 4 rollouts from the current student m_2 (sampling temperature 0.8, top-k 50) and retain one correct and one incorrect trajectory. A teacher (the same m_2 conditioned on a hint-augmented prompt, temperature 0.4, `max_model_len`= 16 384) then produces a corrective CoT. During optimisation we store the top-20 logits of each teacher token, which covers >95% of the softmax mass, and train the student with full-parameter AdamW using an equal mixture of KD ($T = 1.5$) and cross-entropy loss. Training uses the SFT optimizer settings: learning rate 1×10^{-5}, 2 epochs, global batch size 512, and gradient clipping at 1.0.

SSB on Gemma-3-4B-PT. Because the Gemma pretrained model alone cannot produce coherent rollouts for the self-play stage, we additionally initialise the SSB student from a 1-epoch SFT checkpoint and then run one further SSB epoch on top (total budget still = 2 epochs). The Qwen-series bases (Qwen2.5-3B, Qwen3-4B-Base, Qwen3-8B-Base) do not require this warm-up and run SSB directly from the base model.

E DISCUSSION

We introduce Weak-Driven Learning as a new paradigm for SFT: the stronger current model learns from feedback supplied by its own weaker historical state while retaining the original supervised targets. WD-DS uses this history to allocate further training to examples, and WD-JT uses it to shape token-level correction. WMSS thus turns an existing training trajectory into a reusable learning resource. Across four Qwen and Gemma backbones and mathematics, code, and logical reasoning, WMSS consistently improves over same-budget SFT and competing methods. It needs no additional training data, stronger external teacher, reward model, or generated rollouts, and adds no deployment-time inference cost.

Limitations. Our theory characterizes the local effect of history on mixed logits, target margins, and supervised residuals, with exact identities under the stated mixing assumptions (Appendix B). It does not yet prove that full-network optimization reaches the favorable margin conditions or that these local corrections necessarily improve held-out performance. Bridging this gap requires accounting for network Jacobians, interactions across examples and tokens, and optimizer dynamics. The gains across models and domains are established by the experiments and controls in Sections 6.2 and 6.4; an end-to-end convergence or generalization guarantee for historical self-feedback remains an open problem.