RETHINKING TOKEN REWEIGHTING FOR SFT: SUPPRESS, REVERSE, AND EXTRAPOLATE LEARNED FEATURES

Anonymous authors
Paper under double-blind review

ABSTRACT

Supervised fine-tuning (SFT) learns most aggressively from tokens that the model deems least likely. This helps acquire new behaviors, but also amplifies noisy or conflicting supervision and can overwrite useful pretrained knowledge. Through a unified policy-loss view, we revisit existing token-reweighting methods and show that they assign nonnegative coefficients to demonstrated tokens. Consequently, they can suppress or amplify supervised updates, but cannot reverse harmful features once learned. Moreover, larger training weights do not amount to feature extrapolation, since they change the optimization trajectory rather than scale a fixed SFT direction. We argue that reversal and extrapolation require a stable reference frame defined by a fixed SFT delta. Motivated by this, we propose SCALE (**S**elective **C**ontrol of **A**daptation via **L**ocal **E**ntropy), an entropy-guided adaptation-strength-control method that freezes the pretrained model and the SFT delta and learns bounded token- and module-specific gates by minimizing predictive entropy alone. These gates suppress, reverse, or extrapolate frozen SFT features according to their alignment with entropy reduction. Across Qwen2.5-Math-1.5B, Qwen2.5-Math-7B, and Qwen3-4B-Base, SCALE achieves mathematical-reasoning averages of 37.84, 43.60, and 36.57, exceeding the strongest corresponding baselines while remaining competitive on general-retention benchmarks. It also attains the best average code-generation performance across HumanEval, HumanEval+, and MBPP for all three models. These results suggest that effective SFT correction can benefit from controlling how already learned residuals are used, rather than only modifying how they are learned.

1 INTRODUCTION

Post-training pipelines of large language models (LLMs) (Zeng et al., 2026; Yang et al., 2025; Xu et al., 2026) commonly combine supervised fine-tuning (SFT) on curated demonstrations with subsequent exploration through reinforcement learning with verifiable rewards (RLVR) (Yang et al., 2025) and multi-domain integration with on-policy distillation (OPD) (Agarwal et al., 2024). Within this pipeline, SFT provides an essential transition from the general next-token prediction capability acquired during pretraining to crafted task-specific behaviors, structured reasoning patterns, and instruction-following capabilities (Wei et al., 2022; Ouyang et al., 2022; Chung et al., 2024). Despite this central role, the SFT objective remains remarkably simple: it minimizes the average negative log-likelihood of all demonstrated tokens. Let $c_t = (x, y_{<t})$ denote the prefix context and y_t the demonstrated token. Standard SFT optimizes $\mathcal{L}_{\mathrm{SFT}}(\theta) = -\sum_t \log \pi_\theta(y_t \mid c_t)$, whose reference-token logit gradient is $\pi_\theta(y_t \mid c_t) - 1$. Consequently, the correction toward a demonstrated token becomes stronger as the probability assigned to that token decreases, while tokens that the model already predicts confidently receive relatively small updates.

This confidence-dependent correction is a major source of both the effectiveness and the fragility of SFT (Jin et al., 2025; Zhou et al., 2023; Chu et al., 2025). It allows the model to rapidly acquire useful patterns that are unlikely to be predicted by the base model, but it also conflates informative novelty with unreliable supervision. Natural language supervision is inherently one-to-many, and a single demonstration may combine essential reasoning with interchangeable lexical choices, formatting conventions, stylistic preferences, and occasional factual or logical errors. A low-probability

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.

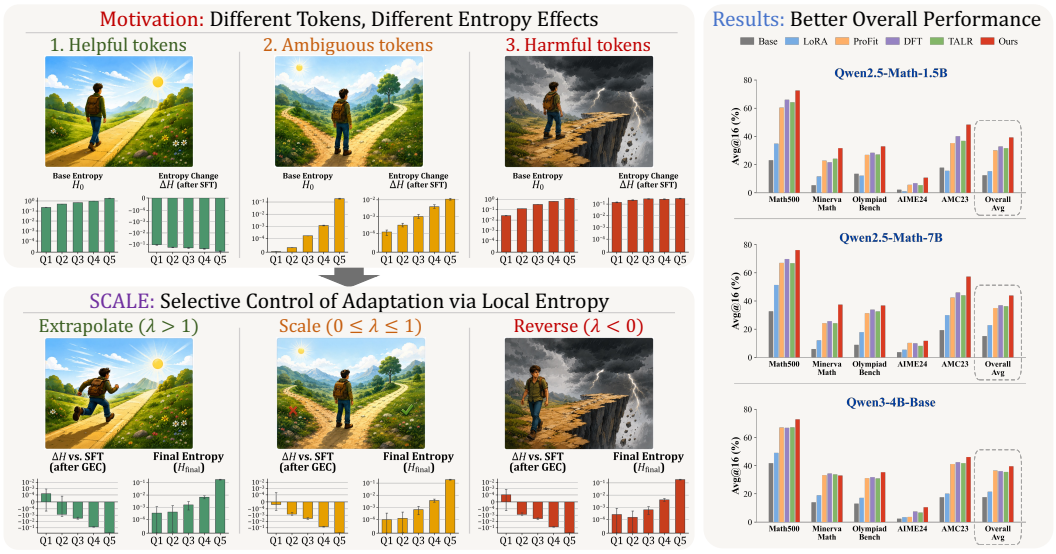

Figure 1: Motivation, illustration and performance of SCALE. The upper left panels show that different tokens induce distinct entropy changes after SFT, motivating our method. The lower left panels illustrate how the learned gates exploit this signal. The twelve bar plots summarize entropy changes before and after gating on held-out examples, providing empirical evidence. The right panels report the resulting improvements across model scales.

reference token can therefore represent two qualitatively different cases. The model may be uncertain, and the reference may provide valuable new information, or the model may confidently prefer a coherent alternative, and the reference may be noisy. Strong correction is desirable in the former case, but in the latter case, it can overwrite useful pretrained knowledge and contribute to suboptimal performance. SFT nevertheless treats both cases identically, assigning its strongest corrections precisely to the tokens whose reliability is most ambiguous.

A growing body of work addresses this problem by removing or reducing the influence of potentially harmful tokens. Data-side approaches identify low-value tokens before training using influence scores or perturbation-based saliency (Pang et al., 2025); training-side approaches instead reweight each token's contribution to the loss. These strategies differ in when they intervene and in the reliability signal they use. Representative methods include DFT (Wu et al., 2026), which assigns a soft weight from the model probability of the reference token; ProFit (Liu et al., 2026b), which masks low-probability tokens to reduce overfitting to a single reference; and EAFT (Diao et al., 2026), which combines reference probability with predictive entropy to attenuate confident conflicts. Despite these differences, these representative methods share one structure: a nonnegative coefficient w_t on the demonstrated token's loss, with standard SFT setting $w_t = 1$ and data filtering the special case $w_t \in \{0, 1\}$. As our analysis in Section 4 shows, this common structure also has a shared limitation. A nonnegative token weight controls the magnitude of the demonstrated-token supervised gradient; it can suppress or amplify this direct CE component but cannot reverse its sign.

Furthermore, token weights act while the task update itself is still being learned: changing w_t changes the optimization trajectory rather than scaling a fixed post-SFT delta. Reversal and extrapolation become well defined only after SFT, when the pretrained model and the learned delta are frozen to form a stable reference frame. For a gated delta $W_\ell h_{\ell,t} + \lambda_{\ell,t} \Delta W_\ell h_{\ell,t}$, values $0 \le \lambda_{\ell,t} < 1$ suppress its contribution, $\lambda_{\ell,t} < 0$ reverse it, and $\lambda_{\ell,t} > 1$ extrapolate it beyond its original SFT strength. We use predictive entropy to decide how these frozen deltas should be scaled. Minimizing entropy induces the signed full-action advantage $A_t^{\text{ent}}(a) = \log \pi(a \mid c_t) + H_t$, providing directional credit over the model's predictive distribution, while SCALE restricts this signal to gate-only adaptation over already learned SFT deltas. Under an alignment condition between entropy advantage and task value within this restricted gate subspace, entropy descent locally strengthens task-beneficial delta directions and weakens task-harmful ones. Predictive entropy is therefore not treated as a correctness oracle, but as an unsupervised local selection signal for a bounded family of frozen delta interventions.

Building on these observations, we propose SCALE, an entropy-guided adaptation-strength-control method for SFT (Figure 1). We first train a LoRA adapter using the standard SFT objective and then freeze both the pretrained model and the learned adapter. We introduce lightweight token- and module-specific gates $\lambda_{\ell,t} \in [\lambda_{\min}, \lambda_{\max}]$ with $\lambda_{\min} < 0$ and $\lambda_{\max} > 1$, modulating each learned LoRA delta as $u_{\ell,t} = W_\ell h_{\ell,t} + \lambda_{\ell,t} \Delta_{\phi^\star,\ell}(h_{\ell,t})$. The value $\lambda_{\ell,t} = 1$ preserves the original SFT delta, values in $[0, 1)$ suppress it, negative values reverse it, and values above one extrapolate it beyond its original strength. The gates are optimized exclusively by minimizing the predictive entropy of the gated model, while all pretrained and SFT parameters remain fixed. We evaluate SCALE on mathematical reasoning, general retention, and code generation using Qwen2.5-Math-1.5B, Qwen2.5-Math-7B, and Qwen3-4B-Base. It achieves mathematical-reasoning Avg@16 averages of 37.84%, 43.60%, and 36.57%, exceeding DFT by 5.03, 6.08, and 1.91 points; remains competitive on general retention; and obtains the best code-generation averages across HumanEval, HumanEval+, and MBPP for all three models (34.72%, 47.35%, and 47.34%). Our contributions are:

1. We present a unified policy-loss analysis of token reweighting for SFT, showing that representative methods such as DFT, ProFit, and EAFT can be expressed as demonstration-action objectives with nonnegative reference-token coefficients.
2. We identify a structural distinction between training-time token reweighting and post-hoc feature correction. Token weights can suppress or amplify supervised gradients, but they cannot reverse learned harmful features or extrapolate beneficial features within a fixed base-to-SFT coordinate system.
3. We propose SCALE, an entropy-guided adaptation-strength-control method that freezes both the pretrained model and the SFT delta, then learns bounded gates $\lambda_{\ell,t} \in [\lambda_{\min}, \lambda_{\max}]$ to perform suppression, reversal, and extrapolation using predictive entropy alone.
4. Experiments across three model settings demonstrate consistent improvements over standard SFT and strong token-reweighting baselines, while ablations confirm the importance of extending the correction range beyond suppression.

2 Related Work

Token-level reweighting for SFT. A growing line of work improves SFT by recognizing that demonstration tokens should not contribute equally to training. Token Cleaning identifies uninformative tokens through their estimated training influence (Pang et al., 2025); DFT rescales token losses using the model probability to improve the implicit reward structure of SFT (Wu et al., 2026); and ProFit masks low-probability tokens to reduce overfitting to replaceable expressions (Liu et al., 2026b). More recent methods incorporate uncertainty or distributional preservation: EAFT uses predictive entropy to identify confident conflicts (Diao et al., 2026), TALR adaptively reweights token losses to balance domain adaptation and general capabilities (Lin et al., 2026), and EKSFT combines entropy- and KL-based token selection with additional regularization (Liu et al., 2026a). Beyond suppression and reweighting, Forgetting explicitly applies negative updates to tokens judged harmful (Taheri et al., 2026). These approaches intervene while the task update is being learned. Our setting is complementary: SCALE first completes ordinary SFT, freezes the resulting task delta, and then learns how strongly that fixed delta should act on each input and module. The distinction is therefore not simply positive versus signed token weights, but training-time modification of the learning trajectory versus post-hoc control of an already learned feature direction.

Post-training model editing and delta manipulation. A line of work studies how to manipulate fine-tuning updates after training. Task Arithmetic represents fine-tuned models through their parameter displacements from a pretrained model and composes multiple task vectors through arithmetic operations (Ilharco et al., 2023). TIES-Merging further addresses interference among independently trained task vectors by resolving redundant and conflicting updates (Yadav et al., 2023). More recent methods such as AdaMerging and WEMoE learn how to combine multiple frozen task updates for multi-task transfer (Yang et al., 2024b; Shen et al., 2024). These methods focus on selecting or composing different learned updates. In contrast, SCALE considers a different post-SFT control problem: given a single completed SFT delta, how should its contribution be modulated for each input and module? By learning token-conditioned gates around the original SFT endpoint,

SCALE enables suppression, reversal, and extrapolation of the same frozen delta rather than merging multiple experts.

3 PRELIMINARIES

Let $c_t = (x, y_{<t})$ be a demonstrated prefix, y_t its next token, and $\pi_{\theta,t}(a) = \mathrm{softmax}(z_\theta(c_t))_a$ the next-token policy. We average over a fixed collection $\mathcal{T}$ of response positions, writing $\frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} f_t = |\mathcal{T}|^{-1}\sum_{t\in\mathcal{T}} f_t$. All logarithms are natural. Standard supervised fine-tuning (SFT) minimizes

$$\mathcal{L}_{\mathrm{SFT}}(\theta) = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} \ell_t(\theta), \qquad \ell_t = -\log \pi_{\theta,t}(y_t), \qquad \nabla_{z_t}\ell_t = \pi_{\theta,t} - e_{y_t}. \tag{1}$$

Our policy-loss view treats a prefix as a state and a vocabulary token as an action. With an action distribution μ_t and action coefficient A_t, define

$$\mathcal{L}_{\mathrm{policy}}(\theta) = -\frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}}\sum_{a\in\mathcal{V}} \mathrm{sg}[\mu_t(a)A_t(a)]\log \pi_{\theta,t}(a). \tag{2}$$

The coefficients are evaluated at the current iterate and held fixed during its backward pass. Thus, this is a gradient surrogate, not necessarily an equality of scalar objectives. SFT uses $\mu_t = \delta_{y_t}$ and $A_t(y_t) = 1$; token reweighting changes this coefficient. The equivalent on-policy representation and its importance-weighting interpretation are given in Appendix B. Contexts remain teacher-forced: an on-policy action expectation does not imply on-policy trajectories.

3.1 PREDICTIVE ENTROPY AS A SIGNED POLICY LOSS

For $\pi_t = \pi_{\theta,t}$, let $H_t = -\sum_a \pi_t(a)\log \pi_t(a)$.

Lemma 1 (Entropy advantage)**.** *For any differentiable parameterization,*

$$\nabla_\theta H_t = -\mathbb{E}_{a\sim\pi_t}\left[A_t^{\mathrm{ent}}(a)\nabla_\theta \log \pi_t(a)\right], \qquad A_t^{\mathrm{ent}}(a) = \log \pi_t(a) + H_t. \tag{3}$$

Consequently, Eq. equation 2 with $\mu_t = \pi_t$ and $A_t = A_t^{\mathrm{ent}}$ has exactly the entropy gradient. The advantage is centered, $\mathbb{E}_{\pi_t} A_t^{\mathrm{ent}} = 0$, and positive when $\pi_t(a) > \exp(-H_t)$. Its direct logit gradient is $\partial H_t/\partial z_t(a) = -\pi_t(a)A_t^{\mathrm{ent}}(a)$. This identity supplies signed action-level feedback, but does not identify correct actions. Our use of it will be restricted to scaling already learned, frozen deltas, under an explicit feature-alignment assumption.

4 REVISITING TOKEN CORRECTION METHODS FOR SFT

4.1 WHAT NONNEGATIVE TOKEN WEIGHTS CONTROL

A demonstration-action corrector has a weighted CE component

$$\mathcal{L}_w = -\frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} \mathrm{sg}[w_t]\log \pi_{\theta,t}(y_t), \qquad w_t \geq 0. \tag{4}$$

DFT and ProFit use reference probabilities, whereas EAFT uses predictive entropy to weight supervision (Wu et al., 2026; Liu et al., 2026b; Diao et al., 2026). Appendix C specifies their coefficients and distinguishes weighted CE from additional loss terms.

Proposition 1 (Local one-sidedness of weighted CE)**.** *Holding w_t fixed, a token's direct logit descent direction is*

$$-\nabla_{z_t}(w_t\ell_t) = w_t(e_{y_t} - \pi_t). \tag{5}$$

It raises the demonstrated logit, lowers competing logits, or vanishes. Nonnegative reweighting cannot reverse this particular CE direction.

This is a statement about one loss component, not an impossibility theorem for shared-parameter training. The distinction we need is more specific: w_t controls learning an update; it does not directly control the application of a completed update.

4.2 A FIXED DELTA DEFINES SUPPRESSION, REVERSAL, AND EXTRAPOLATION

After SFT, freeze both the pretrained map W_m and the learned delta ΔW_m. A scalar gate at module m and position s gives

$$u_{m,s} = W_m h_{m,s} + \lambda_{m,s} \Delta W_m h_{m,s}. \tag{6}$$

At the same module input, the delta is exactly $\lambda_{m,s} \Delta W_m h_{m,s}$: $\lambda = 0$ disables it, $0 < \lambda < 1$ suppresses it, $\lambda = 1$ preserves it, $\lambda < 0$ reverses it, and $\lambda > 1$ extrapolates its strength. This extends the allowable coefficients of a fixed delta, not the set of models obtainable by every training-time reweighting method. We use feature for this module-level delta contribution, not an individually disentangled semantic unit. Its operator is frozen; hidden states and final logit directions can still change with the gates.

4.3 WHY ENTROPY CAN SELECT FIXED FEATURES

Our premise concerns feature contributions: locally, task-supporting deltas reduce uncertainty, while interfering deltas increase it. This is not assumed for unrestricted feature learning. A binary margin model makes the premise concrete. Let the correct-versus-incorrect margin be $m(\lambda) = m_0 + \sum_i \lambda_i a_i$, with fixed contributions a_i and correct probability $p = \mathrm{sigmoid}(m)$. Then

$$\frac{\partial H}{\partial \lambda_i} = -m\, p(1-p) a_i = -m \frac{\partial p}{\partial \lambda_i}. \tag{7}$$

Whenever $m > 0$, entropy descent strengthens helpful contributions $a_i > 0$ and weakens harmful ones $a_i < 0$. Correctness defines helpfulness independently of entropy; the positive margin supplies the needed semantic reference. When $m < 0$, the conclusion reverses. Appendix H gives the multiclass extension. This conditional mechanism is consistent with theoretical work on avoiding spurious features through self-training (Chen et al., 2020), without importing its distributional guarantees into a Transformer. We next express the required alignment in policy-loss form for shared, input-conditioned gates.

5 METHOD

As shown in Figure 2, SCALE separates delta acquisition from delta calibration: standard SFT learns the delta, and predictive entropy subsequently selects its strength.

5.1 STAGE I: LEARNING THE TASK DELTA

LoRA (Hu et al., 2022) parameterizes the delta operator as

$$\Delta_{\phi,m}(h) = \Delta W_m h, \qquad \Delta W_m = \frac{\alpha}{r} B_m A_m. \tag{8}$$

We train only its factors $\phi = \{A_m, B_m\}_m$ with Eq. equation 1, obtaining $\phi^\star$. Stage II freezes the base, output head, and $\phi^\star$. No delta direction is relearned during calibration.

5.2 STAGE II: IDENTITY-CENTERED SIGNED GATES

Each targeted module has an independent scalar gate generated by a linear projection. Unlike continued SFT, this stage cannot modify the learned delta; it only changes the coefficient applied to the frozen delta. For module m, let $w_m \in \mathbb{R}^{d_m}$ and $b_m \in \mathbb{R}$ denote the gate weight and bias:

$$g_{m,s} = w_m^\top \mathrm{sg}[h_{m,s}] + b_m, \qquad v_{m,s} = \tanh(g_{m,s}/\tau), \qquad \tau > 0. \tag{9}$$

We map this response to a bounded coefficient using

$$\lambda_{m,s} = 1 + (1-L)\min(v_{m,s}, 0) + (U-1)\max(v_{m,s}, 0), \qquad L < 0 < 1 < U. \tag{10}$$

Its image is (L, U), and zero gate weights and biases recover the exact Stage-I model. In the experiments, we use a deterministic 10^{-6}-scale perturbation and check initialization parity. The bounded range limits module-level intervention magnitude; it is not an output-divergence guarantee. Appendix G specifies the implementation, including the nonsmooth join and range-ablation endpoints.

5.3 PARAMETER EFFICIENCY

The proposed gating component introduces negligible extra parameters on top of the frozen LoRA delta. For each targeted module m, the gate adds a linear projection (w_m, b_m), where $w_m \in \mathbb{R}^{d_m}$ and $b_m \in \mathbb{R}$, resulting in only $d_m + 1$ additional trainable parameters. Therefore, the total number of Stage-II parameters is $\sum_{m \in \mathcal{M}} (d_m + 1)$, where $\mathcal{M}$ denotes the gated modules. In contrast, the Stage-I LoRA adapter contains matrix-valued factors A_m and B_m with $r(d_m + d'_m)$ parameters per module. Thus, the proposed calibration optimizes only a small fraction of the parameters required to learn the original SFT delta, while keeping the base model and LoRA adapter completely frozen during Stage II.

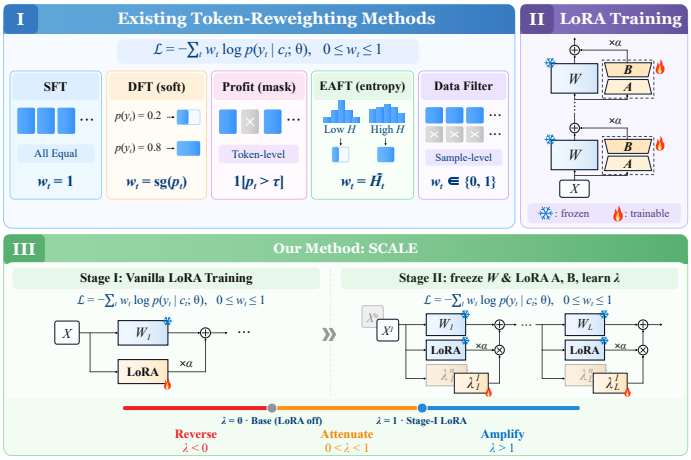

Figure 2: Overview of baseline methods and SCALE.

5.4 Entropy-only training on response prefixes

For $k > 1$, let $S_{k,t}$ be the indices of the largest k logits and define

$$q_{\psi,t}(a) = \frac{\exp z_{\psi,t}(a)}{\sum_{b \in S_{k,t}} \exp z_{\psi,t}(b)}, \quad a \in S_{k,t}, \qquad F_k(\psi) = \frac{1}{\log k} \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} H(q_{\psi,t}). \tag{11}$$

We use $k = 100$ and response positions only. There is no CE, KL, or budget term in Stage II, although the prefixes still come from demonstrations. Taking $k = |\mathcal{V}|$ recovers normalized full-vocabulary entropy. For $k < |\mathcal{V}|$, Eq. equation 11 is a conditional-entropy objective, not the full entropy; all following top-k derivatives are local to an unchanged support (Appendix E).

5.5 Policy credit assignment within the frozen-delta family

Write $q_t = q_{\psi,t}$. Lemma 1 gives $A^{\mathrm{ent}}_{k,t} = \log q_{\psi,t} + H(q_{\psi,t})$. For a gate coordinate i, let $d_{t,i} = \partial z_{\psi,t} / \partial \lambda_i$ denote its local intervention tangent. Then

$$-\frac{\partial F_k}{\partial \lambda_i} = \frac{1}{\log k} \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \mathbb{E}_{q_t} \left[A^{\mathrm{ent}}_{k,t}(a) \frac{\partial \log q_t(a)}{\partial \lambda_i} \right] = \frac{1}{\log k} \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \mathrm{Cov}_{q_t}(z_t, d_{t,i}). \tag{12}$$

Equivalently, Eq. equation 2 uses $\mu_t = q_t$ and $A_t = A^{\mathrm{ent}}_{k,t} / \log k$ (Appendix E). The average includes every causally affected response position, not only the gate's own position. This is signed policy credit transmitted through delta controls, rather than an unconstrained update to task features.

To characterize when entropy provides a useful correction signal, introduce fixed, unobserved action values $r_t(a)$ and define $V_k(\psi) = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \mathbb{E}_{q_t} r_t(a)$. These values are used only for analysis. Let $A^r_t = r_t - \mathbb{E}_{q_t} r_t$ denote the corresponding task advantage and define $s_t(a) = \nabla_\psi \log q_t(a)$. For some $\beta > 0$, the difference between the entropy advantage and the task advantage in the gate parameter space can be written as

$$e_\psi = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \mathbb{E}_{q_t} \left[(A^{\mathrm{ent}}_{k,t} - \beta A^r_t) s_t \right], \qquad -\log k \, \nabla_\psi F_k = \beta \nabla_\psi V_k + e_\psi. \tag{13}$$

Assumption 1 (Gate-subspace entropy–task consistency)**.** *At the operating point, the mismatch term satisfies* $\|e_\psi\| \le \rho\beta \|\nabla_\psi V_k\|$ *for some* $0 \le \rho < 1$.

The condition only requires consistency between entropy and task value along the directions accessible through the gate parameters; it does not require pointwise agreement between model confidence and correctness.

Theorem 1 (Local task improvement by gate-only entropy descent)**.** *Under Assumption 1, differentiability, and $\nabla_\psi V_k \neq 0$, the complete-gradient update $\psi^+ = \psi - \eta \nabla_\psi F_k$ increases V_k for sufficiently small $\eta > 0$. Specifically,*

$$V_k(\psi^+) - V_k(\psi) \geq \frac{\eta\beta(1-\rho)}{\log k} \|\nabla_\psi V_k\|^2 + o(\eta). \tag{14}$$

Thus, entropy provides a valid local credit signal for scaling frozen deltas when its directional preference is consistent with task value in the gate subspace. The guarantee concerns local proxy-value improvement under this condition, while the practical effectiveness of the method is evaluated empirically.

5.6 Why allow reversal and extrapolation after SFT?

Proposition 2 (Removing attenuation-boundary obstructions)**.** *Let F be differentiable near a local minimizer $\bar{\lambda}$ on $[0,1]^M$. If $\nabla F(\bar{\lambda}) \neq 0$, then, for $L < 0 < 1 < U$, an arbitrarily close point in $(L,U)^M$ has strictly smaller F.*

At $\bar{\lambda}_i = 0$, a positive derivative calls for reversal; at $\bar{\lambda}_i = 1$, a negative derivative calls for extrapolation. A decrease from $\lambda = 1$ alone does not justify crossing zero. The proposition concerns independent intervention coordinates; a shared gate must also express the useful direction. It neither places the nonlinear optimum at L or U nor guarantees that optimization finds it.

Freezing $\phi^\star$ fixes the delta operators and the meaning of these coefficients throughout calibration. Interleaving CE would change the operators being controlled, whereas post-hoc training isolates their terminal use (Appendix F). This design complements task arithmetic and entropy-based coefficient learning (Ilharco et al., 2023; Yang et al., 2024b); its distinctive intervention is token- and module-conditioned signed scaling of one completed SFT delta.

6 Experiments

In this section, we evaluate SCALE across mathematical reasoning and code generation on multiple backbone models. Additional experiments, including ablation studies, realized gate behavior, and training-time analysis, are provided in Appendix G.

6.1 Setup

Experimental models and baselines. We conduct experiments on three widely used LLMs, including Qwen2.5-Math-1.5B (Yang et al., 2024a), Qwen2.5-Math-7B (Yang et al., 2024a), and Qwen3-4B-Base (Yang et al., 2025). Based on these backbones, we compare our proposed method SCALE against representative adaptation baselines, including parameter-efficient fine-tuning methods such as LoRA (Hu et al., 2022) and DoRA (Liu et al., 2024), as well as token-level optimization methods such as ProFit (Liu et al., 2026b), DFT (Wu et al., 2026), EAFT (Diao et al., 2026), and TALR (Lin et al., 2026).

Datasets and evaluation. We organize our experiments by adaptation domains, including mathematical reasoning and code generation. For mathematical reasoning, we follow DFT (Wu et al., 2026) in the choice of NuminaMath-CoT (Li et al., 2024) for fine-tuning and the Math500 (Hendrycks et al., 2021), Minerva (Lewkowycz et al., 2022), OlympiadBench (He et al., 2024), AIME24, and AMC23 benchmarks. We further evaluate general capability retention after adaptation on MMLU-Pro (Wang et al., 2024), BBH (Suzgun et al., 2023), and OpenBookQA (Mihaylov et al., 2018). For code generation, we follow MoE^2-LoRA (Yang et al., 2026) in the choice of the Magicoder-OSS corpus (Wei et al., 2023) for fine-tuning. We evaluate on HumanEval (Chen et al., 2021), HumanEval+ (Liu et al., 2023), and MBPP (Austin et al., 2021).

Implementation details. We use two training stages throughout. In Stage I, a LoRA adapter with rank $r = 16$, scaling factor $\alpha = 16$, and zero dropout is trained on all linear layers for one epoch with total batch size 128, AdamW, peak learning rate 5×10^{-4} for math or 5×10^{-5} for code, cosine decay, and 10% warmup. In Stage II, the base model and LoRA adapter are frozen; only the lightweight $\tanh$-bounded gate is trained for a short period of time. To better understand the

Table 1: Mathematical reasoning and general-capability retention results (%). Mathematical-reasoning scores are Avg@16. Boldface and underlining mark the best and second-best adapted scores in each column.

	Mathematical reasoning						General Retention			
Method	Math500	Minerva	Olympiad	AIME24	AMC23	Avg	MMLU-P	BBH	OBQA	Avg
Qwen2.5-Math-1.5B (Yang et al., 2024a)										
Base	21.87	5.24	9.18	1.66	12.81	10.15	20.24	26.32	30.40	25.65
LoRA (Hu et al., 2022)	39.60	9.49	10.80	0.41	17.03	15.47	27.89	31.76	33.80	31.15
DoRA (Liu et al., 2024)	40.10	9.39	10.69	1.24	16.88	15.66	27.58	32.20	33.60	31.13
ProFit (Liu et al., 2026b)	56.12	21.93	22.69	2.29	31.56	26.92	28.02	29.78	33.80	30.53
DFT (Wu et al., 2026)	66.49	21.13	27.61	8.34	40.47	32.81	28.76	32.92	33.80	31.83
EAFT (Diao et al., 2026)	38.79	8.56	9.44	0.21	15.00	14.40	28.78	32.93	**34.80**	32.17
TALR (Lin et al., 2026)	60.28	18.54	23.29	3.32	35.31	28.15	**28.83**	32.06	33.80	31.56
Ours	**71.60**	**31.44**	**31.34**	**8.55**	**46.25**	**37.84**	28.55	**38.67**	32.80	**33.34**
Qwen2.5-Math-7B (Yang et al., 2024a)										
Base	20.57	7.89	6.49	3.33	11.88	10.03	37.86	45.75	34.40	39.34
LoRA (Hu et al., 2022)	49.94	13.91	16.43	3.75	25.62	21.93	38.55	49.12	35.40	41.02
DoRA (Liu et al., 2024)	49.90	12.20	15.86	3.12	25.00	21.22	38.69	50.35	35.40	41.48
ProFit (Liu et al., 2026b)	65.71	24.14	31.32	7.93	43.12	34.44	39.47	49.85	36.00	41.77
DFT (Wu et al., 2026)	68.95	27.75	33.67	8.96	48.28	37.52	39.84	50.08	33.20	41.04
EAFT (Diao et al., 2026)	50.41	14.10	18.67	4.38	26.09	22.73	38.71	50.12	36.20	41.68
TALR (Lin et al., 2026)	68.19	23.92	33.16	7.51	46.09	35.77	40.25	51.15	33.80	41.73
Ours	**76.97**	**35.51**	**36.12**	**12.51**	**56.88**	**43.60**	**40.87**	**64.52**	30.80	**45.40**
Qwen3-4B-Base (Yang et al., 2025)										
Base	46.36	12.67	19.29	6.87	24.84	22.01	46.39	47.47	37.80	43.89
LoRA (Hu et al., 2022)	47.30	11.54	15.90	3.54	21.72	20.00	51.54	55.54	37.40	48.16
DoRA (Liu et al., 2024)	46.53	11.29	16.07	1.86	22.34	19.62	51.39	55.37	36.80	47.85
ProFit (Liu et al., 2026b)	65.75	21.53	30.16	5.42	40.47	32.67	52.04	50.24	37.00	46.43
DFT (Wu et al., 2026)	68.26	25.00	32.27	7.92	39.84	34.66	52.08	49.27	35.40	45.58
EAFT (Diao et al., 2026)	46.92	12.80	15.68	1.45	21.88	19.75	52.04	55.56	**37.60**	**48.40**
TALR (Lin et al., 2026)	66.62	20.18	32.11	9.59	40.94	33.89	51.78	53.11	33.60	46.16
Ours	**69.85**	**25.41**	**32.77**	**9.99**	**44.84**	**36.57**	**52.25**	**55.97**	36.80	48.34

Table 2: Code-generation results (%). Boldface and underlining mark the best and second-best adapted scores in each column.

	Qwen2.5-Math-1.5B				Qwen2.5-Math-7B				Qwen3-4B-Base			
Method	HE	HE+	MBPP	Avg	HE	HE+	MBPP	Avg	HE	HE+	MBPP	Avg
Base	28.66	17.68	32.80	26.38	42.28	39.02	34.20	38.50	43.09	39.02	37.00	39.70
LoRA (Hu et al., 2022)	34.15	25.61	30.80	30.19	48.37	43.29	**38.00**	43.22	51.22	45.73	40.20	45.72
DoRA (Liu et al., 2024)	31.91	28.05	31.20	30.39	46.14	43.29	36.20	41.88	50.81	45.12	40.00	45.31
DFT (Wu et al., 2026)	34.96	29.88	33.60	32.81	48.58	42.07	36.00	42.22	44.51	38.41	37.20	40.04
ProFit (Liu et al., 2026b)	35.57	30.49	33.80	33.29	47.56	43.29	35.60	42.15	48.78	43.90	40.00	44.23
EAFT (Diao et al., 2026)	31.30	29.27	32.80	31.12	46.75	41.46	36.40	41.54	45.73	40.85	39.80	42.13
TALR (Lin et al., 2026)	36.59	**33.54**	**34.00**	34.71	49.39	44.51	37.00	43.63	44.72	39.63	37.80	40.72
Ours	**41.67**	31.10	31.40	**34.72**	**57.11**	**46.95**	**38.00**	**47.35**	**53.25**	**47.56**	**41.20**	**47.34**

mechanism of the range of the gating component, we evaluate all 42 combinations of lower bounds $L \in \{-6, \ldots, 0\}$ and upper bounds $U \in \{1, \ldots, 6\}$, train a separate gate for each interval, and **report the result of every interval in Figure 3**. The Ours rows in Tables 1 and 2 additionally present the best intervals in Figure 3 on the corresponding evaluation benchmarks.

6.2 Main results

Mathematical reasoning and general-capability retention. Table 1 reports the mathematical reasoning and general-capability retention results. On mathematical reasoning, SCALE achieves state-of-the-art performance among all compared adaptation methods across all three backbones, obtaining the best results on every benchmark. This consistent improvement across different backbone families demonstrates the effectiveness of learning a bounded correction over the adaptation delta for enhancing reasoning ability. Meanwhile, SCALE achieves strong general-capability retention with the highest retention Avg on both Qwen2.5-Math backbones and the second-highest on Qwen3-4B-Base. These results show that SCALE combines improved mathematical reasoning with strong average general-capability retention.

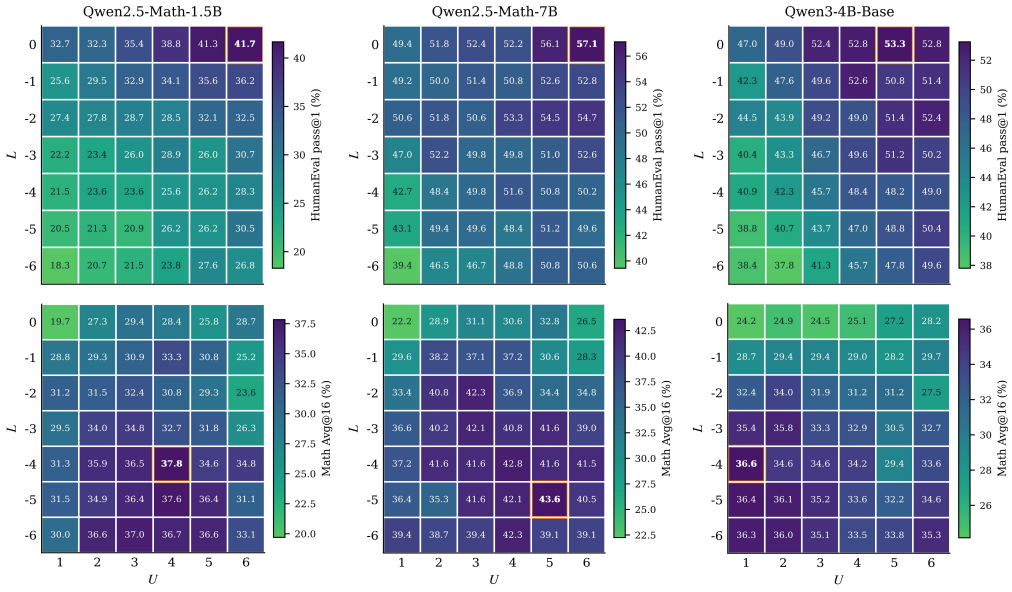

Figure 3: Gate-range sensitivity across backbones. We evaluate all endpoint pairs to characterize the intervention space. The selected maxima are reported as oracle sweep results, while fixed-range results are used for controlled comparisons.

Code generation and downstream generalization. Table 2 reports the code-generation performance of SCALE. SCALE consistently achieves the best overall performance across different backbone families, obtaining the highest Avg score on Qwen2.5-Math-1.5B, Qwen2.5-Math-7B, and Qwen3-4B-Base. These results extend the benefits of adaptation-delta control to code generation across the three backbones.

6.3 Discussion

Gate-range preferences across backbones. Figure 3 shows a task-dependent pattern across all three backbones. High-scoring mathematical configurations occupy the negative-L region. The two Qwen2.5-Math models also favor upper bounds above one, whereas Qwen3-4B-Base reaches its mathematical peak with a smaller upper bound. In the coding task, the high-scoring region lies near $L = 0$ with larger positive upper bounds. The preferred balance of reversal and extrapolation therefore varies with both the task and the backbone.

What the bounds contribute. At fixed $U = 1$, negative lower bounds improve math on all three backbones; larger U adds gains for both Qwen2.5-Math models. Expanding $[0, 1]$ to $[0, 6]$ improves code pass@1 on all three, while even $[0, 1]$ exceeds LoRA in math. These patterns support reversal for math and extrapolation for code. Appendix G.3 reports realized gates and inference-time interventions.

Interpreting the task difference. Math and code may use pretrained and newly adapted features differently. Qwen2.5-Math's specialization (Yang et al., 2024a) may make some SFT features worth correcting, while code gains more from amplifying them. Qwen3-4B-Base (Yang et al., 2025) shows the same split, suggesting that task–delta alignment also matters.

7 Conclusion

Supervised fine-tuning leaves a delta with both useful and harmful effects. SCALE freezes that delta and learns an entropy-guided signed gate to suppress, reverse, or extrapolate it. Across three backbones, this control improves mathematical reasoning and code generation while retaining general capability, showing the value of calibrating a completed adaptation.

AI Assistance Statement

We used large language models to assist with language polishing, improving the clarity of presentation, and supporting parts of the software development process, including writing and debugging auxiliary code such as visualization scripts and analysis utilities. All technical content, including the methodology, experimental design, implementation decisions, results interpretation, and conclusions, was developed, verified, and validated by the authors.

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*, volume 2024, pp. 21246–21263, 2024.

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, 2021. URL https://arxiv.org/abs/2108.07732.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Yining Chen, Colin Wei, Ananya Kumar, and Tengyu Ma. Self-training avoids using spurious features under domain shift, 2020. URL https://arxiv.org/abs/2006.10032.

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

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25:1–53, 2024.

Muxi Diao, Lele Yang, Wuxuan Gong, Yutong Zhang, Zhonghao Yan, Yufei Han, Kongming Liang, Weiran Xu, and Zhanyu Ma. Entropy-adaptive fine-tuning: Resolving confident conflicts to mitigate forgetting, 2026. URL https://arxiv.org/abs/2601.02151.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations*, 2023.

Hangzhan Jin, Sitao Luan, Tianwei Ni, Sicheng Lyu, Guillaume Rabusseau, Reihaneh Rabbany, Doina Precup, and Mohammad Hamdaqa. Rl fine-tuning heals ood forgetting in sft. *arXiv preprint arXiv:2509.12235*, 2025.

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

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q. Jiang, Ziju Shen, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. NuminaMath: The largest public dataset in AI4Maths with 860k pairs of competition math problems and solutions, 2024.

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

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *Advances in neural information processing systems*, 36:21558–21572, 2023.

Qi Liu, Mingdi Sun, Yongyi He, Zhi Zheng, Tong Xu, Yi Zheng, Zhefeng Wang, and Enhong Chen. Entropy-kl divergence-based token masking: A novel approach for selective fine-tuning of large language models. *arXiv preprint arXiv:2605.29303*, 2026a.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. DoRA: Weight-decomposed low-rank adaptation. In *International Conference on Machine Learning*, 2024.

Tao Liu, Taiqiang Wu, Runming Yang, Shaoning Sun, Junjie Wang, and Yujiu Yang. Profit: Leveraging high-value signals in sft via probability-guided token selection, 2026b. URL `https://arxiv.org/abs/2601.09195`.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2381–2391, 2018.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022.

Jinlong Pang, Na Di, Zhaowei Zhu, Jiaheng Wei, Hao Cheng, Chen Qian, and Yang Liu. Token cleaning: Fine-grained data selection for llm supervised fine-tuning. *arXiv preprint arXiv:2502.01968*, 2025.

Li Shen, Anke Tang, Enneng Yang, Guibing Guo, Yong Luo, Lefei Zhang, Xiaochun Cao, Bo Du, and Dacheng Tao. Efficient and effective weight-ensembling mixture of experts for multi-task model merging. *arXiv preprint arXiv:2410.21804*, 2024.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed H Chi, Denny Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051, 2023.

Ali Taheri, Alireza Taban, Qizhou Wang, Shanshan Ye, Abdolreza Mirzaei, Tongliang Liu, and Bo Han. Forgetting: A new mechanism towards better large language model fine-tuning, 2026. URL `https://arxiv.org/abs/2508.04329`.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. In *ICLR*, 2022.

Yuxiang Wei, Zihao Wang, Ji Li, et al. Magicoder: Source code is all you need, 2023. URL https://arxiv.org/abs/2312.02120.

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

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems*, 2023.

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

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

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. Adamerging: Adaptive model merging for multi-task learning, 2024b. URL https://arxiv.org/abs/2310.02575.

Qingyu Yang, Haonan He, Minglei Li, Jingqi Ye, Tao Chen, Lei Bai, and Peng Ye. MoE^2-LoRA: When MoE models meet MoE-style low-rank adaptation, 2026. URL https://arxiv.org/abs/2607.21978.

Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. Lima: Less is more for alignment, 2023. URL https://arxiv.org/abs/2305.11206.

A NOTATION

We list the notation used in this paper in Table 3.

Table 3: Notation used throughout the paper.

Symbol	Definition
x	Input prompt
$y = (y_1, \ldots, y_T)$	Demonstrated response tokens
$c_t = (x, y_{<t})$	Prefix context at token position t
$a \in \mathcal{V}$	Next-token action from vocabulary $\mathcal{V}$
$\pi_\theta(a \mid c_t)$	Model next-token distribution parameterized by θ
z_t	Output logits at position t
H_t	Predictive entropy of $\pi_\theta(\cdot \mid c_t)$
$A_t^{\mathrm{ent}}(a)$	Entropy advantage: $\log \pi_\theta(a \mid c_t) + H_t$
θ_0	Pretrained model parameters
$\phi^\star$	Frozen LoRA parameters after Stage-I SFT
$\Delta\phi^\star$	Learned SFT delta)
W_ℓ	Frozen pretrained weight of layer ℓ
ΔW_ℓ	LoRA delta added to layer ℓ
$h_{\ell,t}$	Hidden representation at layer ℓ, position t
$\lambda_{\ell,t}$	Gate coefficient applied to the frozen delta
ψ	Trainable gate parameters
$g_{\psi,\ell,t}$	Gate logit before transformation
w_t	Token weight used by training-time reweighting methods
$\mathcal{L}_{\mathrm{SFT}}$	Standard supervised fine-tuning loss
$\mathcal{L}_w$	Weighted token-level SFT objective
$F_k(\psi)$	Top-k entropy objective optimized by the gate
q_t	Renormalized top-k predictive distribution
$V_k(\psi)$	Analytical task-value proxy under gate parameters
e_ψ	Entropy–task advantage mismatch in gate space
L, U	Lower and upper bounds of the signed gate range
$\lambda < 0$	delta reversal regime
$0 \le \lambda < 1$	delta suppression regime
$\lambda = 1$	Original SFT delta strength
$\lambda > 1$	delta extrapolation regime

B POLICY-LOSS IDENTITIES AND THEIR ASSUMPTIONS

B.1 AN EXACT GRADIENT IDENTITY FOR SFT

Let $q_t^{\mathrm{demo}}(a) = \mathbf{1}[a = y_t]$ and assume a finite vocabulary with strictly positive softmax probabilities. Since demonstrated contexts are held fixed,

$$\begin{aligned}\nabla_\theta \mathcal{L}_w &= -\frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \sum_a q_t^{\mathrm{demo}}(a) w_t \nabla_\theta \log \pi_t(a) \\ &= -\frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \mathbb{E}_{a \sim \pi_t} \left[\frac{w_t \mathbf{1}[a = y_t]}{\pi_t(a)} \nabla_\theta \log \pi_t(a) \right]. \end{aligned} \tag{15}$$

Equation equation 15 is the same expected gradient, not the derivative of an expected reward obtained by differentiating the importance ratio. The coefficient is stopped in the surrogate. The factor $1/\pi_t(y_t)$ is cancelled in expectation by the sampling probability; it does not make the deterministic CE logit gradient unbounded. This representation is related to the RL analysis motivating DFT (Wu et al., 2026).

An action-independent baseline can be subtracted because $\mathbb{E}_{\pi_t} \nabla \log \pi_t = 0$. For example, subtracting w_t gives a signed coefficient on the full vocabulary without changing the weighted CE gradient. Therefore, signed coefficients alone are not an invariant distinction between SFT and entropy minimization. The relevant objects are the induced gradients and their trainable coordinates: one is a

Method	CE coefficient w_t	Signal / qualification
SFT	1	Uniform demonstration weighting.
DFT	$\mathrm{sg}[p_t]$	Probability-weighted CE (Wu et al., 2026).
ProFit	$\mathbf{1}[\mathrm{sg}[p_t] > \tau_p]$	Probability mask (Liu et al., 2026b).
EAFT	$H_t^{\text{top-20}} / \log 20$	Normalized top-20 entropy; frozen-coefficient CE interpretation (Diao et al., 2026).
TALR	$\max\{w_{\min}, \mathrm{sg}\left[e^{-\ell_t/\tau_\ell}\right]\}$	Loss-dependent weight, dynamic temperature, and floor (Lin et al., 2026).
Offline token mask	$m_t \in \{0, 1\}$	Fixed retained positions, e.g., token cleaning (Pang et al., 2025).

Table 4: Nonnegative coefficients of demonstration-action CE components. Entries describe CE components. The EAFT row reports its published entropy weight without specifying a stop-gradient convention.

reweighted demonstration CE direction; the other is entropy feedback applied to frozen-delta controls. Neither representation makes the demonstrated prefix distribution on-policy.

B.2 Reference-token entropy threshold

For a distribution p and reference token y, consider the *probability-space* path $p_\epsilon = (1-\epsilon)p + \epsilon e_y$. Direct differentiation gives

$$\frac{\mathrm{d}}{\mathrm{d}\epsilon} H(p_\epsilon)\bigg|_0 = -\log p_y - H(p) = -A^{\mathrm{ent}}(y). \tag{16}$$

Thus $p_y > \exp[-H(p)]$ identifies entropy decrease along this particular mixture path. This threshold connects the reference token to the signed entropy advantage; it is not an optimal weighting rule for ordinary SFT. Indeed, a CE step in logits follows $z(\eta) = z + \eta(e_y - p)$ and induces

$$\frac{\mathrm{d}p}{\mathrm{d}\eta}\bigg|_0 = C(p)(e_y - p), \qquad C(p) = \mathrm{diag}(p) - pp^\top, \tag{17}$$

which generally differs from $e_y - p$. A parameter update additionally depends on the network Jacobian. Our method consequently uses the actual gate sensitivity of entropy rather than treating the reference threshold as a sufficient test for CE or feature correction.

C Token Correction Objectives

Table 4 lists coefficients of the weighted CE component. Here $p_t = \pi_t(y_t)$, $\ell_t = -\log p_t$, and m_t is a preprocessing mask. A method belongs to Eq. equation 4 as a gradient objective only when its coefficient is held fixed during differentiation.

Differentiated weights. If a coefficient $w_t(\theta)$ is differentiated, then

$$\nabla_\theta\big(w_t(\theta)\ell_t(\theta)\big) = w_t\nabla_\theta\ell_t + \ell_t\nabla_\theta w_t. \tag{18}$$

Proposition 1 applies to the first term only. The published EAFT loss uses normalized predictive entropy as its weighting statistic, rather than an explicit two-argument function of reference probability and entropy. Reference probability still enters the CE factor. Its implementation's gradient convention must be specified independently. For TALR, the practical rule uses a weight floor and a temperature based on batch loss statistics; the nonnegative-ray conclusion is unchanged.

Table 5: Zero-threshold entropy-sign ablation on QWEN2.5-MATH-1.5B. Negative and Positive suppress deltas where $\Delta H < 0$ and $\Delta H > 0$, respectively. Scores are Avg@16. Boldface and underlining mark the best and second-best scores in each column.

Method	Math500	Minerva	Olympiad	AIME24	AMC23	Avg@16
Negative	30.85	6.54	8.85	1.24	15.47	12.59
LoRA	39.60	9.49	10.80	0.41	17.03	15.47
Positive	<u>47.99</u>	<u>15.24</u>	<u>20.08</u>	<u>4.16</u>	<u>25.47</u>	<u>22.59</u>
Ours	**71.60**	**31.44**	**31.34**	**8.55**	**46.25**	**37.84**

Objectives outside this class. EKSFT includes masked CE together with entropy and KL terms (Liu et al., 2026a). Its CE component can be entered in Eq. (4), but its entire objective cannot. Likewise, Taheri et al. (2026) explicitly subtract a loss on negative tokens. That method is not constrained by nonnegative token weights. The one-sidedness result therefore applies to nonnegative weighted CE components.

D EMPIRICAL MOTIVATION: DELTA ENTROPY AND LOCAL FEATURE SELECTION

Before training the gate, we ask whether the entropy effect of a *completed* LoRA delta carries directional information about how that delta should be used. For matched teacher-forced prefixes, define

$$\Delta H_t = H(\pi_{\mathrm{LoRA},t}) - H(\pi_{\mathrm{Base},t}). \tag{19}$$

The diagnostic suppresses delta contributions at positions selected by the sign of ΔH_t, with the base model and LoRA adapter frozen. This is a hard forward intervention without gate training. Table 5 compares sign-based suppression with LoRA and Ours across five math benchmarks.

Suppressing entropy-decreasing delta contributions yields 12.59 Math Avg@16, compared with 15.47 for LoRA, while suppressing entropy-increasing contributions yields 22.59. This ordering motivates using entropy change to select which frozen delta contributions to weaken and learning a continuous correction gate. The diagnostic measures finite differences between frozen models; the local alignment conditions for task-improving entropy descent are developed separately in Appendix H.

E CAUSAL GATE CALCULUS, TOP-k ENTROPY, AND STOPPED GRADIENTS

E.1 INTERVENTION COORDINATES VERSUS SHARED GATE PARAMETERS

Collect all module–position gate values for the fixed examples into $\boldsymbol{\lambda}$. Let $G_t(\boldsymbol{\lambda}; \theta_0, \phi^\star)$ be the logits when these values are supplied as external controls. The learned model satisfies $z_{\psi,t} = G_t(\Lambda(\psi))$, where Λ includes the dependence of gates on their current hidden inputs. At a differentiable operating point,

$$z_t(\boldsymbol{\lambda} + \delta\lambda) = z_t(\boldsymbol{\lambda}) + J_t^\lambda \delta\lambda + o(\|\delta\lambda\|), \qquad J_t^\psi = J_t^\lambda B, \quad B = \frac{\partial \Lambda}{\partial \psi}. \tag{20}$$

A column $d_{t,i}$ of J_t^λ is zero when gate i is outside the causal computation of prediction t. Otherwise it may affect that prediction even if located at an earlier sequence position. Consequently,

$$\frac{\partial F_k}{\partial \lambda_i} = \frac{1}{\log k} \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \langle \nabla_{z_t} H(q_t), d_{t,i} \rangle \,. \tag{21}$$

No globally affine base-to-SFT logit model is assumed. Frozen ΔW_m fixes an operator, not $\Delta W_m h_{m,s}$ across inputs or training steps. A scalar gate rescales the entire module delta at a position; it cannot independently separate all semantic components within that delta. Likewise, $B^\top$ is a chain-rule pullback, not generally an orthogonal projection. An advantageous independent intervention need not be expressible by shared gate parameters.

E.2 Exact policy loss for the implemented conditional entropy

Fix a neighborhood with unchanged top-k support S_t. Extend q_t by zero outside S_t, and let $\mathcal{H}_t = H(q_t)$. Then

$$\frac{\partial \mathcal{H}_t}{\partial z_t(a)} = \begin{cases} -q_t(a)\big(\log q_t(a) + \mathcal{H}_t\big), & a \in S_t, \\ 0, & a \notin S_t. \end{cases} \tag{22}$$

Since $\mathbb{E}_{q_t} A^{\mathrm{ent}}_{k,t} = 0$ and $\log q_t(a) = \log \pi_t(a) - \log \sum_{b \in S_t} \pi_t(b)$,

$$\nabla_\psi \mathcal{H}_t = -\mathbb{E}_{q_t}[A^{\mathrm{ent}}_{k,t} \nabla_\psi \log q_t] = -\mathbb{E}_{q_t}[A^{\mathrm{ent}}_{k,t} \nabla_\psi \log \pi_t]. \tag{23}$$

Thus the implemented loss has the policy surrogate in Eq. equation 2 with $\mu_t = q_t$ and $A_t = A^{\mathrm{ent}}_{k,t} / \log k$. It uses all actions in the retained support, not the full vocabulary when $k < |\mathcal{V}|$. At support ties, the smooth proof is replaced by a branchwise analysis; the reported procedure recomputes top-k every forward pass.

For support mass $m_t = \sum_{a \in S_t} \pi_t(a)$ and normalized tail $\widetilde{\pi}_t$, full entropy decomposes as

$$H(\pi_t) = h_{\mathrm{b}}(m_t) + m_t H(q_t) + (1 - m_t) H(\widetilde{\pi}_t). \tag{24}$$

Decreasing $H(q_t)$ does not by itself decrease these other terms. Similarly, V_k is a conditional action value. It is not the full-vocabulary expected value unless the corresponding tail effects are controlled.

E.3 What detaching a gate input changes

For one module $u = Wh + \lambda(h)\Delta W h$, the complete hidden-state Jacobian is

$$\frac{\partial u}{\partial h} = W + \lambda \Delta W + (\Delta W h)(\nabla_h \lambda)^\top. \tag{25}$$

Using $\lambda = \lambda(\mathrm{sg}[h])$ drops the last term from the backward graph. Hidden states still depend on upstream gates, so freezing W and ΔW does not remove this difference. Write $\mathrm{D}_\psi F_k$ for the resulting semi-gradient and $\nabla_\psi F_k$ for the ordinary derivative of the numerical forward function. Entropy and policy surrogates remain gradient-equivalent *within the same stopped graph*, but ordinary descent guarantees require an additional check.

Define scaled complete and implemented directions by

$$G_k = -\log k \, \nabla_\psi F_k, \qquad \widehat{G}_k = -\log k \, \mathrm{D}_\psi F_k, \qquad b_{\mathrm{sg}} = \widehat{G}_k - G_k. \tag{26}$$

For the conditional task value, Eq. equation 13 becomes

$$\widehat{G}_k = \beta \nabla_\psi V_k + e_\psi + b_{\mathrm{sg}}. \tag{27}$$

Hence Theorem 1 also holds for the semi-gradient step if $\|e_\psi + b_{\mathrm{sg}}\| \le \rho\beta \|\nabla_\psi V_k\|$, with $\rho < 1$. This is an additional condition, not a consequence of freezing.

For the full value $V = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \mathbb{E}_{\pi_t} r_t$, define $b_{\mathrm{tail}} = \beta(\nabla_\psi V_k - \nabla_\psi V)$. Then

$$\widehat{G}_k = \beta \nabla_\psi V + \underbrace{e_\psi + b_{\mathrm{sg}} + b_{\mathrm{tail}}}_{e_{\mathrm{total}}}. \tag{28}$$

The sufficient condition becomes $\|e_{\mathrm{total}}\| \le \rho\beta \|\nabla_\psi V\|$. These terms separate feature-selection mismatch, stopped-gradient error, and support truncation. The experiments do not directly measure them. Complete-gradient gate training sets $b_{\mathrm{sg}} = 0$ but is a distinct backward-pass variant; it is not substituted for the reported method here.

F Post-hoc Calibration and a Fixed delta Reference

Write the entropy objective as $F_\phi(\psi)$ to expose its dependence on the LoRA factors. SCALE first obtains $\phi^\star$ and then optimizes the fixed forward objective $F_{\phi^\star}(\psi)$. The delta operators, not necessarily the logit tangents, remain fixed. In particular, setting every gate to one continues to identify

the same Stage-I model, and setting every gate to zero recovers the base when all other parameters are unchanged.

For interleaved iterates $(\phi^{(j)}, \psi^{(j)})$, adding and subtracting $F_{\phi^{(j)}}(\psi^{(j+1)})$ yields

$$\begin{aligned} &F_{\phi^{(j+1)}}(\psi^{(j+1)}) - F_{\phi^{(j)}}(\psi^{(j)}) \\ &= \underbrace{F_{\phi^{(j)}}(\psi^{(j+1)}) - F_{\phi^{(j)}}(\psi^{(j)})}_{\Delta^{(j)}_{\text{gate}}} + \underbrace{F_{\phi^{(j+1)}}(\psi^{(j+1)}) - F_{\phi^{(j)}}(\psi^{(j+1)})}_{\Delta^{(j)}_{\text{delta}}}. \end{aligned} \tag{29}$$

The second term has no fixed sign. Freezing removes it and isolates the question of how to use a completed delta. This identity does not establish that post-hoc optimization outperforms joint or alternating optimization: a joint CE–entropy objective can also be a fixed function of its combined parameters.

Even for frozen $\phi^\star$, monotone loss descent requires a genuine descent direction and an appropriate step size. The deterministic, complete-gradient statements in Appendix H are local to smooth branches; minibatch noise, optimizer state, top-k changes, and stopped gate inputs require their own conditions. The post-hoc design fixes the intervention reference, not these optimization issues.

G More Experiments

G.1 More Implementation Details

Stage-I training. For math, the first-stage LoRA is trained for one epoch on 100k NuminaMath-CoT examples, following the data choice of DFT (Wu et al., 2026). For code, a separate first-stage LoRA is trained for one epoch on 75,197 Magicoder-OSS examples, following the data choice of MoE^2-LoRA (Yang et al., 2026). Both use total batch size 128, rank 16, alpha 16, all linear modules as targets, and dropout 0; the learning rates are 5×10^{-4} for math and 5×10^{-5} for code. Stage II treats the resulting LoRA operators as the fixed delta family to be calibrated; it does not relearn their factors.

Stage-II gate training. For math, the gate is trained on the first 20k NuminaMath-CoT examples; for code, it is trained on a 20k-example subset of Magicoder-OSS. Each gate run uses 157 optimizer steps with total batch size 128, learning rate $5 \cdot 10^{-4}$, cosine schedule, zero warmup, and a cutoff length of 4096 tokens. Each targeted module owns a linear map

$$g_{\psi,m,t} = w_m^\top \operatorname{sg}(x_{m,t}) + b_m$$

from its detached current input to a scalar logit, and these gate parameters are the only Stage-II trainable components. The reported runs use a fixed deterministic FP32 tiny initialization for the gate weights and zero bias. Step-zero parity with the frozen Stage-I model is verified before optimization. The normalized top-k entropy uses $k = 100$ and response-token predictions only.

Detaching gate inputs removes the derivative through their dependence on upstream gates. Stage II therefore uses the semi-gradient analyzed in Appendix E. Likewise, the top-k objective is the conditional entropy of the retained support, not the full-vocabulary entropy. The policy-loss identity remains exact on a locally fixed support within the same stopped graph; transferring a complete-gradient task-improvement statement to the implemented update requires the additional error condition in Eq. equation 27.

Evaluation details. Mathematical-reasoning and general-capability retention evaluations use task-native zero-shot prompts without a system message and EOT-aligned assistant termination. All methods are evaluated with vLLM inference under the same decoding configuration.

G.2 Ablation studies

Entropy provides a stronger calibration signal. Table 6 compares cross-entropy (CE) and entropy gate calibration with pure LoRA continued on the same 20k examples. The gate uses signed tanh with range $[-6, 3]$; continued LoRA has no gate. Entropy reaches 37.84 Math Avg@16, versus 16.27 for continued LoRA and 16.17 for CE.

Table 6: Calibration-loss ablation on QWEN2.5-MATH-1.5B (%). Scores are Avg@16; boldface and underlining mark the best and second-best values.

Method	Math500	Minerva	Olympiad	AIME24	AMC23	Avg@16
LoRA	39.60	9.49	10.80	0.41	17.03	15.47
LoRA +20k	40.12	9.72	<u>11.58</u>	<u>1.04</u>	<u>18.91</u>	<u>16.27</u>
SCALE w/ CE	<u>41.97</u>	<u>10.31</u>	11.36	0.82	16.41	16.17
SCALE w/ Entropy	**71.60**	**31.44**	**31.34**	**8.55**	**46.25**	**37.84**

Table 7: Three-seed mathematical-reasoning results on QWEN2.5-MATH-1.5B (%). Scores are Avg@16, reported as mean ± sample standard deviation.

Method	Math500	Minerva	Olympiad	AIME24	AMC23	Avg@16
LoRA	39.52 ± 0.07	9.60 ± 0.33	10.59 ± 0.39	0.62 ± 0.21	16.72 ± 1.43	15.41 ± 0.44
DFT	<u>65.30 ± 1.03</u>	<u>21.42 ± 1.89</u>	<u>27.22 ± 0.37</u>	<u>7.64 ± 1.78</u>	<u>39.95 ± 0.65</u>	<u>32.31 ± 0.92</u>
SCALE	$\mathbf{71.73 \pm 0.57}$	$\mathbf{27.18 \pm 2.47}$	$\mathbf{31.56 \pm 0.31}$	$\mathbf{10.21 \pm 1.08}$	$\mathbf{48.44 \pm 1.18}$	$\mathbf{37.83 \pm 0.20}$

Stable performance across training seeds. Across seeds 42, 3407, and 2025, SCALE uses the fixed $[-6, 3]$ gate range and achieves 37.83 ± 0.20 Math Avg@16 (Table 7). It leads on all five benchmarks.

G.3 REALIZED GATE BEHAVIOR

The mathematical-reasoning gates use $[-6, 3]$ for QWEN2.5-MATH-1.5B and $[-4, 4]$ for QWEN2.5-MATH-7B. The code-generation gates on QWEN2.5-MATH-7B use $[0, 6]$ and $[-3, 3]$. The multiplier λ scales the frozen LoRA delta: negative values reverse it, values between zero and one suppress or preserve it, and values above one extrapolate it.

Table 8: Distribution of gate multipliers on generated mathematical-reasoning tokens (%).

Backbone	$\lambda < 0$	$0 \le \lambda \le 1$	$\lambda > 1$
QWEN2.5-MATH-1.5B	39.36	17.81	42.82
QWEN2.5-MATH-7B	44.34	19.11	36.55

Table 8 shows that the learned gates realize both reversal and extrapolation during generation. The two Qwen2.5-Math backbones assign $\lambda < 0$ to 39.36% and 44.34% of generated-token coefficients, respectively, and $\lambda > 1$ to 42.82% and 36.55%. Thus, most generated-token coefficients lie outside the unit interval in both models.

The forward interventions in Table 9 show that removing reversal lowers Math Avg@16 by 16.95 and 9.15 points on the 1.5B and 7B models. Removing extrapolation lowers it by 6.99 and 12.46 points. Constraining the gate to $[0, 1]$ produces the largest reduction for each model. Both realized reversal and extrapolation contribute to mathematical-reasoning accuracy in these checkpoints.

Table 10: Gate interventions for code generation on QWEN2.5-MATH-7B. Scores are HumanEval pass@1 (%).

Gate range	Gate applied at inference	pass@1	Change
$[0, 6]$	Original λ	57.11	—
	Unit interval, $\mathrm{clip}(\lambda, 0, 1)$	47.15	-9.96
$[-3, 3]$	Original λ	49.80	—
	No reversal, $\max(\lambda, 0)$	52.64	$+2.85$
	No extrapolation, $\min(\lambda, 1)$	45.12	-4.67
	Unit interval, $\mathrm{clip}(\lambda, 0, 1)$	47.97	-1.83

Table 9: Mathematical-reasoning gate interventions. Scores are Math Avg@16; parentheses show the change from the original gate in percentage points.

Gate applied at inference	QWEN2.5-MATH-1.5B	QWEN2.5-MATH-7B
Original λ	37.05	42.52
No reversal, $\max(\lambda, 0)$	20.10 (-16.95)	33.37 (-9.15)
No extrapolation, $\min(\lambda, 1)$	30.06 (-6.99)	30.06 (-12.46)
Unit interval, $\mathrm{clip}(\lambda, 0, 1)$	19.44 (-17.61)	24.07 (-18.45)

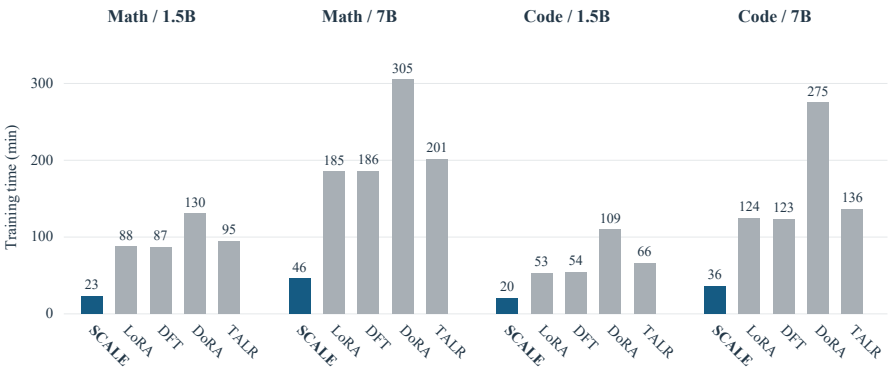

Figure 4: Single-GPU training time (minutes) for mathematical reasoning and code generation on Qwen2.5-Math-1.5B and 7B.

Table 10 shows a 9.96-point decrease when the $[0, 6]$ gate is restricted to the unit interval. With the $[-3, 3]$ gate, removing extrapolation lowers pass@1 by 4.67 points, while removing reversal raises it by 2.85 points. The code results favor extrapolation rather than reversal in these checkpoints.

G.4 TRAINING TIME

Figure 4 compares training time on one H200 GPU for the two Qwen2.5-Math backbones on mathematical reasoning and code generation. SCALE denotes Stage-II gate training, averaged across five gate ranges with one run per range; LoRA, DFT, DoRA, and TALR denote their standalone Stage-I training runs. Full SCALE training also includes its preceding LoRA stage. Excluding queueing and evaluation, gate training takes 20.0–46.0 minutes across the four settings, compared with 53.0–305.1 minutes for the baseline runs.

H PROOFS AND STRUCTURAL CONDITIONS FOR SECTIONS 2–4

H.1 PROOF OF THE ENTROPY-ADVANTAGE IDENTITY

Proof of Lemma 1. For a strictly positive finite softmax distribution,

$$\begin{aligned}\nabla H(p) &= -\sum_a p(a)(\log p(a) + 1)\nabla \log p(a) \\ &= -\sum_a p(a)(\log p(a) + H(p))\nabla \log p(a), \end{aligned} \tag{30}$$

because $\sum_a p(a)\nabla \log p(a) = \nabla \sum_a p(a) = 0$. Moreover, $\mathbb{E}_p[\log p + H(p)] = 0$. Since $\partial \log p(a)/\partial z(b) = \mathbf{1}[a = b] - p(b)$, $\partial H/\partial z(b) = -p(b)(\log p(b) + H(p))$. Applying the same computation to the softmax restricted to a locally fixed support proves Eqs. equation 22 and equation 23. □

H.2 Local one-sidedness under weighted CE

Proof of Proposition 1. Differentiating a weighted CE term with fixed weight gives $-\nabla_z(w\ell) = w(e_y - p)$. Its y coordinate is $w(1-p_y) \geq 0$, and its other coordinates are $-wp_a \leq 0$. For any $a \neq y$, the direct logit-margin change is proportional to $w(1 - p_y + p_a) \geq 0$. Thus changing a nonnegative weight rescales a single direct CE direction. □

This property is not preserved as a coordinatewise sign guarantee after shared-network propagation. If $K_t = \partial z_t/\partial\theta$, then one parameter update on several tokens has first-order effect

$$\Delta z_t = \frac{\eta}{|\mathcal{T}|}\sum_s w_s K_t K_s^\top (e_{y_s} - \pi_s) + o(\eta). \tag{31}$$

The cross-token matrices need not preserve signs. Neither this proposition nor the on-policy representation rules out unlearning through parameter sharing or through a different loss.

H.3 Policy credit and the task-alignment identity

At a fixed support, let $d_i = \partial z/\partial\lambda_i$. The score derivative is

$$\frac{\partial \log q(a)}{\partial \lambda_i} = d_i(a) - \mathbb{E}_q d_i. \tag{32}$$

Since $A^{\mathrm{ent}} = z - \mathbb{E}_q z$, Lemma 1 yields $-\partial H(q)/\partial\lambda_i = \mathrm{Cov}_q(z, d_i)$. For fixed action values r, differentiation likewise gives $\partial\mathbb{E}_q r/\partial\lambda_i = \mathrm{Cov}_q(r, d_i)$. Averaging over all affected positions proves Eq. equation 12. For shared parameters,

$$\nabla_\psi V_k = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} \mathbb{E}_{q_t}[A_t^r s_t], \qquad -\log k\, \nabla_\psi F_k = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} \mathbb{E}_{q_t}[A_{k,t}^{\mathrm{ent}} s_t]. \tag{33}$$

Subtracting β times the first identity from the second proves Eq. equation 13.

A structural sufficient condition. Write, on each current support,

$$z_t(a) = \beta r_t(a) + b_t + \varepsilon_t(a), \qquad \beta > 0. \tag{34}$$

Here r_t is fixed independently of entropy. The decomposition alone is always possible and imposes no restriction. Its useful condition is that the delta error has small effect on the accessible directions. Indeed,

$$A_{k,t}^{\mathrm{ent}} - \beta A_t^r = \varepsilon_t - \mathbb{E}_{q_t}\varepsilon_t, \qquad e_\psi = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} (J_t^\psi)^\top C(q_t)\varepsilon_t. \tag{35}$$

The centering matrix is $C(q_t) = \mathrm{diag}(q_t) - q_t q_t^\top$, restricted to the support. Large logit error in directions annihilated by $(J_t^\psi)^\top C(q_t)$ does not corrupt the gate signal. Freezing the feature operators restricts this accessible family; it does not automatically make the delta small.

For an individual intervention coordinate, let $h_i = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} \mathrm{Cov}_{q_t}(r_t, d_{t,i})$ and $b_i = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} \mathrm{Cov}_{q_t}(\varepsilon_t, d_{t,i})$. Then

$$-\log k\, \frac{\partial F_k}{\partial \lambda_i} = \beta h_i + b_i. \tag{36}$$

If $h_i \neq 0$ and $|b_i| < \beta|h_i|$, entropy gives the correct sign of task contribution. The vector assumption in the main text instead guarantees aggregate improvement with shared gates; it need not give a correct sign for every coordinate separately.

H.4 Proof of local task improvement and a finite step bound

Proof of Theorem 1. Set $v = \nabla_\psi V_k$ and $G = \beta v + e_\psi$. The step is $\psi^+ = \psi + \eta G/\log k$. By Cauchy–Schwarz and Assumption 1,

$$\langle v, G\rangle \geq \beta\|v\|^2 - \|v\|\|e_\psi\| \geq \beta(1-\rho)\|v\|^2 > 0. \tag{37}$$

The first-order expansion of V_k establishes Eq. equation 14; a sufficiently small step stays within the differentiable neighborhood and makes the remainder smaller than the strictly positive leading term. □

If ∇V_k is L_V-Lipschitz in that neighborhood, the stronger bound is

$$V_k(\psi^+) - V_k(\psi) \geq \left[\frac{\eta\beta(1-\rho)}{\log k} - \frac{L_V\eta^2\beta^2(1+\rho)^2}{2(\log k)^2}\right]\|v\|^2. \tag{38}$$

It is positive for $0 < \eta < 2(1-\rho)\log k/[L_V\beta(1+\rho)^2]$, provided the segment stays in that neighborhood. This is a deterministic Euclidean-gradient statement, not an unconditional guarantee for an optimizer with momentum or finite minibatch noise.

The same proof with $G = \widehat{G}_k$ establishes the semi-gradient and full-value variants in Appendix E, replacing the delta by $e_\psi + b_{\text{sg}}$ or e_{total} respectively. For any actual implemented update vector u, the elementary local task-improvement test is $\langle \nabla V, u\rangle > 0$. A low numerical entropy alone does not certify it.

H.5 A MARGIN MODEL FOR HELPFUL AND HARMFUL FEATURES

In the binary setting, $p = \text{sigmoid}(m)$ gives

$$\frac{\mathrm{d}H}{\mathrm{d}p} = -m, \qquad \frac{\mathrm{d}p}{\mathrm{d}m} = p(1-p), \qquad \frac{\partial m}{\partial \lambda_i} = a_i, \tag{39}$$

which proves Eq. equation 7. For $m > 0$, the sign of $-\partial H/\partial\lambda_i$ equals the sign of $\partial p/\partial\lambda_i$. With values $r(\text{correct}) = 1$ and $r(\text{wrong}) = 0$, this is exactly task-value improvement. At $m = 0$ there is no first-order entropy signal; at $m < 0$ it has the wrong orientation. Thus a reliable semantic reference, not freezing alone, is needed.

For a multiclass distribution p, let a fixed feature tangent be $d_i = a_i e_y + b_i\mathbf{1}$, where y is the independently defined correct action. Then

$$\frac{\partial H}{\partial \lambda_i} = -a_i p_y \sum_{b\neq y} p_b(z_y - z_b), \qquad \frac{\partial p_y}{\partial \lambda_i} = a_i p_y(1-p_y). \tag{40}$$

The sign correspondence holds if $z_y > \sum_{b\neq y} p_b z_b/(1-p_y)$. A strictly highest correct logit suffices. This model concerns features that change correct-versus-incorrect evidence in the specified form; arbitrary multiclass deltas require the directional condition in Eq. equation 36. For language modeling, a demonstration token is not automatically the unique correct action. Values can instead be assigned by a fixed external criterion or a fixed reference continuation policy, but the fixed-prefix guarantee then remains local and does not by itself prove improved free-running accuracy.

H.6 SIGNED PARAMETERIZATION

For $v = \tanh(g/\tau)$, the two branches of Eq. equation 10 are

$$\lambda(v) = \begin{cases} 1 + (1-L)v, & v < 0, \\ 1 + (U-1)v, & v \geq 0. \end{cases} \tag{41}$$

Both slopes are positive when $L < 0 < 1 < U$; the branches agree at zero and approach L and U as $v \to -1$ and $v \to 1$. The image is (L, U) with closure $[L, U]$. Moreover, $\lambda = 0$ at $v = -1/(1-L)$; more negative values reverse the delta, and every positive v extrapolates it. At $g = 0$, $\lambda = 1$ exactly. The map is generally not differentiable at that join, as its one-sided slopes in g differ.

For fixed input h, the module-level displacement obeys

$$\|u(h;\lambda) - u(h;1)\| \leq \max\{1-L, U-1\}\,\|\Delta W h\|. \tag{42}$$

This controls local delta scaling, not global output KL or task loss. If $\Delta W h \neq 0$, expanding from $[0, 1]$ to (L, U) strictly expands that module's attainable delta contributions. Degenerate deltas or downstream cancellation need not enlarge the final prediction set. Training-time token weights define a different object and are not covered by this set inclusion.

H.7 PROOF OF THE ATTENUATION-BOUNDARY RESULT

Proof of Proposition 2. Every $\bar{\lambda} \in [0,1]^M$ lies in the interior of $[L,U]^M$. For $g = \nabla F(\bar{\lambda}) \neq 0$ and sufficiently small $\epsilon > 0$, $\lambda' = \bar{\lambda} - \epsilon g$ belongs to $(L,U)^M$. Differentiability gives

$$F(\lambda') = F(\bar{\lambda}) - \epsilon \|g\|^2 + o(\epsilon) < F(\bar{\lambda}). \tag{43}$$

At a box-constrained local minimum, $g_i \geq 0$ if $\bar{\lambda}_i = 0$, $g_i = 0$ if $0 < \bar{\lambda}_i < 1$, and $g_i \leq 0$ if $\bar{\lambda}_i = 1$. A nonzero gradient therefore identifies at least one blocked outward direction: reversal from zero or extrapolation from one. □

If $\bar{\lambda}$ is a global minimum over the attenuation box, the proposition also gives a strictly lower attainable objective in the extended box. It does not imply endpoint optimality for nonlinear entropy. Its realization by a shared gate requires a parameter direction v with $\langle \nabla_\lambda F, Bv \rangle < 0$. Moreover, the sign must be evaluated at the relevant boundary: a positive derivative at one only supports attenuation initially, not an immediate jump to a negative coefficient. A zero derivative supplies no first-order verdict and does not exclude curvature, parameter-sharing, or subsequent-iterate effects.

I LIMITATION

Despite the promising results, this work has several limitations that motivate future research. First, the current evaluation is limited to relatively small-scale language models. Our experiments are conducted on models up to 7B parameters, and we have not yet validated SCALE on larger-scale models (e.g., 30B+ parameters). Although the proposed gated delta calibration mechanism is parameter-efficient and theoretically independent of model size, its effectiveness and scalability on larger models remain to be investigated. Future work should evaluate whether the observed benefits persist in larger-scale settings with more complex adaptation dynamics.

Second, our experiments are currently restricted to models from the Qwen family. Although we evaluate multiple generations and variants of Qwen models, including Qwen2.5-Math and Qwen3-Base, further validation on models from other families is necessary. In particular, evaluating SCALE on architectures and pretraining distributions beyond Qwen would provide a stronger assessment of its generality and robustness across different foundation models.

Third, the current gate selection mechanism is not fully adaptive. In SCALE, the gate range is selected through empirical evaluation, and our experiments demonstrate that different gate ranges can effectively control the degree of delta suppression, reversal, and extrapolation. However, automatically determining the optimal intervention range remains an open problem. Developing more principled adaptive gate selection strategies, potentially based on task signals, uncertainty estimation, or learned optimization criteria, is an important direction for future research.