FISHER-INFORMED RECALIBRATION FOR FEEDBACK-BASED ON-POLICY SELF-DISTILLATION OF LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Feedback-based on-policy self-distillation has emerged as a promising approach for enabling foundation models, more specifically Large Language Models (LLMs), to learn from their own outputs under external feedback, with a single model serving as both teacher and student. However, such methods can exhibit unstable optimization, conducive to performance collapse during training. To address this limitation, we propose **FIRE** (**F**isher-**I**nformed **RE**calibration), *a dual-branch framework that recalibrates the supervision applied to correct and incorrect on-policy outputs* during fine-tuning. For correct responses, FIRE replaces self-distillation with re-weighted on-policy SFT, while for incorrect ones FIRE identifies feedback components that disproportionately influence the teacher-induced update and recalibrates the feedback-conditioned target accordingly. Both branches are influenced by a token-level radius derived in part from a softmax Fisher trace. FIRE separates *which direction* feedback should move the model from *how far* the model should move in that direction, while leaving well-behaved feedback supervision unchanged. Our experiments demonstrate that FIRE provides substantially more stable self-distillation while maintaining strong downstream performance, particularly in settings where standard feedback-conditioned distillation becomes unstable.

1 INTRODUCTION

Foundation Models (Bommasani et al., 2021), which are models trained on large amounts of data, have become increasingly influential and adopted in practice, with Large Language Models (LLMs) particularly enjoying widespread adoption (Minaee et al., 2024). In practice, many available LLMs are pre-trained (Grattafiori et al., 2024; Yang et al., 2025), with fine-tuning being a method for adapting these models to specific use cases, updating only a small number of parameters relative to the total model (Hu et al., 2021; Zhang et al., 2025).

In parallel, fine-tuning such LLMs from *their own outputs* has become an emerging research direction (Lee et al., 2026; Hübotter et al., 2026). In particular, feedback-based on-policy self-distillation (Zhao et al., 2026b) has emerged as an increasingly popular paradigm. The model first generates an initial response, receives feedback (e.g., user interactions), and conditions on this feedback to produce a revised response. The feedback-conditioned model then serves as a teacher, with the original model optimized to minimize the Kullback-Leibler (KL) (Kullback & Leibler, 1951) divergence between its output distribution and the feedback-conditioned distribution.

Challenges. Despite its promise, feedback-based on-policy self-distillation fine-tuning presents multiple challenges. Firstly, not every on-policy response should be trained from feedback-conditioned teacher supervision in the same manner. Applying self-distillation to responses that are already correct can be unnecessary and may contribute to model divergence (Zheng et al., 2026), since the response itself already provides an unambiguous positive learning signal. Conversely, when a response is incorrect, feedback is necessary to determine how the model should revise its prediction, but directly matching the feedback-conditioned teacher blindly, whose revised outputs may still be incorrect, can potentially introduce excessively large and poorly directed updates, which also induce performance collapse (Lee et al., 2026). Moreover, standard KL-based self-distillation treats the entire feedback context as an infallible source of supervision and provides no mechanism

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.

for determining whether a particular feedback component disproportionately drives the resulting update. Lastly, teacher-student discrepancy could substantially affect optimization depending on the student's local uncertainty and the learning rate. These observations motivate two complementary requirements: the training procedure should distinguish between correct and incorrect on-policy trajectories, and it should constrain feedback-induced updates according to the student's local predictive geometry while preserving informative feedback whenever its induced update is already well behaved.

1.1 CONTRIBUTIONS

To address these challenges, we propose **FIRE** (**F**isher-**I**nformed **RE**calibration), a *dual-branch framework for stabilizing feedback-based on-policy self-distillation fine-tuning of LLMs*. FIRE distinguishes between correct and incorrect on-policy responses and recalibrates their training signals according to the student's local predictive geometry. For correct responses, FIRE replaces feedback-conditioned self-distillation with Fisher-informed re-weighted on-policy supervised fine-tuning (SFT). For incorrect responses, FIRE attributes the influence of individual feedback components and recalibrates the resulting teacher target before distillation. Overall, our main contributions are as follows:

- *Dual-branch on-policy self-distillation.* We introduce an outcome-routed framework that applies differing supervision to correct and incorrect on-policy responses. In contrast with existing work that adopts correct-branched SFT, correct responses are optimized with gradient and Fisher-informed re-weighted on-policy SFT. Incorrect responses retain feedback-conditioned self-distillation, allowing feedback to guide revision only where it is needed.
- *Fisher-informed feedback recalibration.* For incorrect responses, FIRE uses leave-one-feedback-field-out counterfactuals to quantify how individual feedback components influence the teacher-induced update in the gradient space. We combine this attribution with a token-level radius derived from the student's softmax Fisher trace and the learning rate, separating *which direction* feedback should move the model from *how far* the model should move in that direction. Importantly, FIRE leaves the original full-feedback supervision unchanged whenever its induced update already lies within the permitted radius.
- *Minimal hyperparameter tuning.* Outside of the *nominal rate*, which controls how aggressively the radius contracts as the learning rate increases, FIRE introduces no additional hyperparameters. Therefore, FIRE does not require meticulous hyperparameter tuning to fully take advantage of its performance improvements.
- *Bounded and stable self-distillation.* We show that FIRE bounds the token-level output-logit gradient in both the correct and incorrect branches by the same student-dependent radius. Empirically, we additionally show that this recalibration substantially improves training stability over standard feedback-conditioned self-distillation and numerous relevant baselines, particularly when standard self-distillation becomes unstable.

2 RELATED WORK

Knowledge Distillation (KD). Knowledge Distillation (KD) (Hinton et al., 2015) is a common technique employed in deep learning that is designed to distill capabilities from one model (the teacher) to another (the student) (Lee et al., 2025). More specifically, the student seeks to match the teacher's output distribution through a KL-divergence (Kullback & Leibler, 1951) based loss. KD-based training has been demonstrated to be an effective method of aligning the student's distribution to the teacher's (Phuong & Lampert, 2019), and has been used extensively in foundation model settings such as LLMs (Xu et al., 2024). Therefore, such distillation has become an increasingly adopted option for models trained on synthetic outputs generated by themselves or another model (Jang et al., 2026; Xie et al., 2026; Xing et al., 2026). In our work, we consider the first option.

On-Policy Self-Distillation and Stabilization. On-Policy Self-Distillation is a KD-based methodology in which a model learns from supervision constructed around *its own generated outputs* (Zhao et al., 2026b). This is in contrast with on-policy distillation, which assumes the teacher and student can be differing models (Xie et al., 2026; Xing et al., 2026). For on-policy self-distillation, the

current student policy first generates an output, producing an on-policy trajectory (Shenfeld et al., 2026). Then this original generation and feedback context is provided again, which produces a new target output conditioned on the student's behavior and feedback (Lee et al., 2026), which is what is distilled upon. As the student changes throughout training, both the trajectories presented to the teacher and the resulting distillation targets likewise change. With existing research demonstrating distilling on all samples can induce performance collapse (Li et al., 2026a; Zheng et al., 2026), existing literature seeks to stabilize the self-distillation process by dual-branching correct and incorrect trajectories, with Li et al. (2026a) and Zheng et al. (2026) considering Group Relative Policy Optimization (GRPO) (Shao et al., 2024) and a perplexity-based version of supervised fine-tuning (SFT) (Ouyang et al., 2022) on correct trajectories respectively. In addition, works such as Li et al. (2026b) measure the disagreement between the teacher and student, geometrically shifting the teacher closer to the student when the differences are unreconcilable. Outside of the self-distillation setting, stabilization of on-policy distillation has also been explored under the idea of a trust region, constraining unreliable and overly large updates (Xie et al., 2026; Xing et al., 2026). Compared to prior work, FIRE applies Fisher-informed re-weighted on-policy SFT to correct trajectories, while recalibrating incorrect-trajectory self-distillation through feedback-field attribution and a Fisher-informed update radius. Further details are provided in Sec. 3 and Appendix E.

Parameter Efficient Fine-tuning (PEFT). When fine-tuning foundation models, it is desirable to limit the number of parameters which are updated for computational efficiency. For this, parameter efficient fine-tuning methods (PEFT) are particularly popular and have demonstrated good performance across a wide range of domains (Fang et al., 2025; Liu et al., 2024). In addition, it is demonstrated to be useful in resource-constrained settings (Fang et al., 2025; Lee et al., 2025). Common methods to employ PEFT include prefix-tuning (Li & Liang, 2021) and LoRA (Hu et al., 2021), with the latter becoming the most popular option and the approach we adopt.

3 Proposed Methodology

3.1 Feedback-Based On-Policy Self-Distillation

We define $\pi_{\boldsymbol{\theta}_k}$ as the trainable *student* at step k, and denote $\pi_{\bar{\boldsymbol{\theta}}_k}$ as an exponential moving average (EMA) *teacher*, which we adopt from Hübotter et al. (2026) for stability. The teacher is initialized from the student, $\bar{\boldsymbol{\theta}}_0 = \boldsymbol{\theta}_0$, and is updated only through EMA rather than backpropagation. Given a prompt $\mathbf{x} \sim \mathcal{D}$, with $\mathcal{D}$ denoting an incoming data stream (e.g., user interaction), the current student first samples one on-policy response, i.e.,

$$\mathbf{y} = (y_1, \ldots, y_T) \sim \pi_{\boldsymbol{\theta}_k}(\cdot \mid \mathbf{x}). \tag{1}$$

A feedback mechanism then produces privileged feedback $\mathbf{c} = \mathcal{F}(\mathbf{x}, \mathbf{y})$ about the sampled response. For example, $\mathbf{c}$ may contain a critique, correction, reasoning hint, or reference information. This feedback is available only during training and is provided to the teacher, not the student. Then, for response position t, both models score the same sampled token prefix $\mathbf{y}_{<t}$. We define the live student distribution $p^t_{\boldsymbol{\theta}}(v)$ and detached step-k teacher distribution $q^t_{\mathbf{c},k}(v)$ as

$$p^t_{\boldsymbol{\theta}}(v) = \pi_{\boldsymbol{\theta}}(v \mid \mathbf{x}, \mathbf{y}_{<t}), \quad q^t_{\mathbf{c},k}(v) = \mathrm{sg}\left[\pi_{\bar{\boldsymbol{\theta}}_k}(v \mid \mathbf{x}, \mathbf{c}, \mathbf{y}_{<t})\right], \qquad v \in \mathcal{V}, \tag{2}$$

where $\mathcal{V}$ is the vocabulary and sg denotes a stop-gradient. At step k, $p^t_{\boldsymbol{\theta}}(v)$ is instantiated as $\boldsymbol{\theta} = \boldsymbol{\theta}_k$ to denote that it is the live distribution, while $q^t_{\mathbf{c},k}(v)$ is a detached target. We omit v when referring to either distribution. Importantly, evaluating $q^t_{\mathbf{c},k}$ does not require an additional autoregressive rollout. Because the complete student trajectory $\mathbf{y}$ is already available, $q^t_{\mathbf{c},k}$ is obtained simultaneously through a single teacher forward pass under the causal mask. Therefore, the EMA teacher scores the existing student trajectory rather than generating a new response. Next, the student is trained through reverse-KL distillation, i.e.,

$$\mathcal{L}_{\mathrm{KL}}(\boldsymbol{\theta}) = \frac{1}{T}\sum_{t=1}^{T} \mathrm{KL}\big(p^t_{\boldsymbol{\theta}} \,\big\|\, q^t_{\mathbf{c},k}\big)\,. \tag{3}$$

After backpropagating, the student is updated first. Then the EMA teacher tracks the student, i.e.,

$$\boldsymbol{\theta}_{k+1} = \mathrm{OptStep}\left(\boldsymbol{\theta}_k, \nabla_{\boldsymbol{\theta}_k}\mathcal{L}_{\mathrm{KL}}\right), \qquad \bar{\boldsymbol{\theta}}_{k+1} = \mu\bar{\boldsymbol{\theta}}_k + (1-\mu)\boldsymbol{\theta}_{k+1}, \tag{4}$$

where $\mathrm{OptStep}$ is the update rule (e.g., SGD with momentum) and $\mu \in [0, 1)$ is the EMA decay.

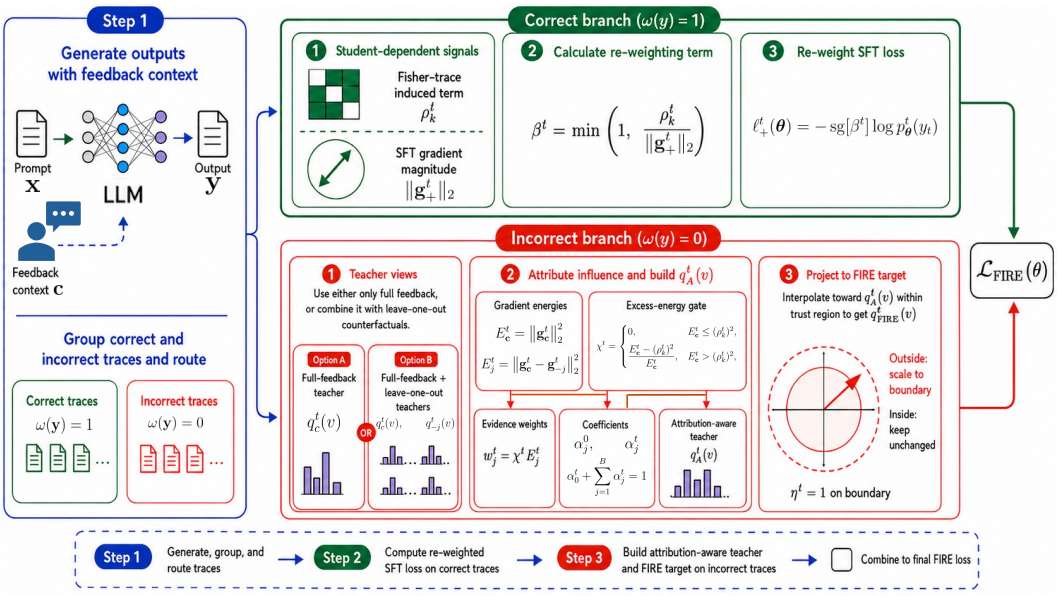


Figure 1: The fine-tuning process of the proposed FIRE methodology. Firstly, we separate traces produced from the model into correct and incorrect groups. Next, Fisher-informed re-weighted on-policy SFT is conducted on correct traces while incorrect traces engage in recalibrated self-distillation, re-weighting based on leave-one-out counterfactuals, full feedback, and gradient insights. Both branches are influenced by the Fisher-informed term ρ_k^t. The two branches are combined to form the final loss. Specifics on differences from similar works is found in Appendix E.

3.2 FIRE: FISHER-INFORMED RECALIBRATION

In this subsequent section, we outline the training procedure of FIRE, which modifies how the feedback-conditioned supervision in Sec. 3.1 is applied. FIRE routes correct and incorrect rollouts differently while constraining both branches with a shared student-dependent output-gradient radius. Complete derivations are provided in Appendix F.

Step 1: Outcome routing and teacher views. For responses that are already correct, existing work shows that applying self-distillation to those samples can lead to collapse (Li et al., 2026a; Zheng et al., 2026). Therefore, after sampling $\mathbf{y}$, a verifier $\omega(\mathbf{y}) \in \{0, 1\}$ indicates whether the initial output is correct. Correct responses receive Fisher-informed re-weighted on-policy SFT, whereas incorrect responses receive feedback-conditioned distillation. Intuitively, a verified response already provides an unambiguous positive target, while an incorrect response requires the teacher's feedback to determine how the prediction should change.

For target construction, we define $p^t(v) = \mathrm{sg}[p^t_{\boldsymbol{\theta}_k}(v)]$ as the detached current student distribution. On an incorrect response, FIRE represents the feedback as fields $\mathbf{c} = (b_1, \ldots, b_B)$, where B is the number of feedback fields. For notational simplicity, we suppress step index k on detached targets, writing $q^t_{\mathbf{c}}(v) := q^t_{\mathbf{c},k}(v)$. We use p^t and $q^t_{\mathbf{c}}$ without an argument to denote the corresponding distributions. The EMA teacher produces this full-feedback target from equation 2, together with the leave-one-field-out targets

$$q^t_{-j}(v) = \mathrm{sg}\left[\pi_{\bar{\boldsymbol{\theta}}_k}(v \mid \mathbf{x}, \mathbf{c}_{\setminus j}, \mathbf{y}_{<t})\right], \tag{5}$$

where $\mathbf{c}_{\setminus j}$ removes feedback field b_j. Each leave-one-field-out view is a counterfactual that asks *how the teacher's supervision would change without field* b_j, allowing FIRE to attribute influence to individual feedback components rather than treating the context as one block. We demonstrate that despite this overhead, our method achieves the best time-accuracy tradeoff in Appendix A.2. For any detached target q^t, we define its induced student-logit gradient as

$$\mathbf{g}^t(q^t) := \nabla_{\mathbf{z}^t} \mathrm{KL}\big(\mathrm{softmax}(\mathbf{z}^t) \,\big\|\, q^t\big)\big|_{\mathbf{z}^t = \mathbf{z}^t_k}. \tag{6}$$

where $\mathbf{z}_k^t$ denotes the current student logits at token position t. This converts each teacher distribution into the actual local training direction it would impose on the student. FIRE performs attribution in the gradient space since a change in teacher probabilities matters only in that it changes the update the student will receive. FIRE bounds this local signal using

$$\rho_k^t = \frac{\sqrt{1 - \|p^t\|_2^2}}{\max(1, a_k/a_0)}, \tag{7}$$

where a_k is the current learning rate and a_0 is a hyperparameter we call the *nominal rate*. The numerator is the square root of the softmax Fisher trace, which measures how spread out the student's prediction is and gives FIRE a natural reference for choosing the update size. When the student places most of its probability on only a few tokens, the radius is smaller. When its probability is spread across many tokens, the radius is larger. The denominator further reduces the radius when the learning rate is higher, preventing the larger learning rate from making the update too strong and causing divergence. Importantly, while existing work also considers trust-region-style stabilization for on-policy distillation (Xie et al., 2026; Xing et al., 2026), these methods consider *conventional teacher–student distillation rather than self-distillation* and *do not use the student's softmax Fisher trace to calibrate the local update radius* (additional details in Appendix E). The explicit derivation is provided in Appendix F.3.

Step 2: Correct branch training. Next, for $\omega(\mathbf{y}) = 1$, the standard on-policy SFT gradient is $\mathbf{g}_+^t = p^t - \mathbf{e}_{y_t}$, where $\mathbf{e}_{y_t}$ is the one-hot vector for token y_t. FIRE then re-weights the SFT loss via

$$\beta^t = \min\left(1, \frac{\rho_k^t}{\|\mathbf{g}_+^t\|_2}\right), \qquad \ell_+^t(\boldsymbol{\theta}) = -\,\mathrm{sg}[\beta^t] \log p_{\boldsymbol{\theta}}^t(y_t). \tag{8}$$

Then, with equation 8, we balance the uncertainty of the update (ρ_k^t) with the magnitude of the gradient ($\|\mathbf{g}_+^t\|_2$). Therefore, FIRE on correct traces balances student uncertainty with the magnitude of the induced gradient, unlike existing work (Zheng et al., 2026), which weights correct trajectories using student perplexity, in contrast with *an explicit Fisher-informed gradient-magnitude constraint* (additional details in Appendix E).

Step 3: Incorrect branch training. For incorrect responses, i.e., $\omega(\mathbf{y}) = 0$, we let $\mathbf{g}_{\mathbf{c}}^t = \mathbf{g}^t(q_{\mathbf{c}}^t)$ and $\mathbf{g}_{-j}^t = \mathbf{g}^t(q_{-j}^t)$. For an incorrect trace, we define the full-teacher gradient energy and the leave-one-field-out influence energy as

$$E_{\mathbf{c}}^t = \left\|\mathbf{g}_{\mathbf{c}}^t\right\|_2^2, \qquad E_j^t = \left\|\mathbf{g}_{\mathbf{c}}^t - \mathbf{g}_{-j}^t\right\|_2^2, \tag{9}$$

with $E_{\mathbf{c}}^t$ measuring the total strength of the update requested by the complete feedback-conditioned teacher, while E_j^t measures how much removing field b_j changes that update. A large E_j^t indicates influence, but does not by itself imply that the field is harmful. We require that the full teacher must first violate the local budget before FIRE attenuates any field. FIRE therefore defines the excess-energy gate

$$\chi^t = \begin{cases} 0, & E_{\mathbf{c}}^t \le (\rho_k^t)^2, \\ \dfrac{E_{\mathbf{c}}^t - (\rho_k^t)^2}{E_{\mathbf{c}}^t}, & E_{\mathbf{c}}^t > (\rho_k^t)^2, \end{cases} \qquad w_j^t = \chi^t E_j^t. \tag{10}$$

The gate separates influence from unsafe influence. If the full teacher already lies within the permitted energy radius, then $\chi^t = 0$ and the full-feedback target is retained exactly. If not, $\chi^t \in (0, 1)$ measures the fraction of its gradient energy lying beyond the boundary. Multiplying each E_j^t by this common fraction preserves the relative ranking of field influence while making the overall attenuation strength proportional to the severity of the budget violation. After computing the field weights $\{w_j^t\}_{j=1}^B$, FIRE normalizes the full-teacher anchor and leave-one-field-out evidence, i.e.,

$$Z^t = (\rho_k^t)^2 + \sum_{j=1}^{B} w_j^t, \qquad \alpha_0^t = \frac{(\rho_k^t)^2}{Z^t}, \qquad \alpha_j^t = \frac{w_j^t}{Z^t}. \tag{11}$$

The coefficients are nonnegative and satisfy $\alpha_0^t + \sum_{j=1}^B \alpha_j^t = 1$. The term $(\rho_k^t)^2$ acts as evidence for retaining the complete teacher, while each w_j^t acts as evidence for moving toward the counterfactual

teacher in which field b_j is absent. Because both quantities are measured in gradient-energy units, their normalization yields dimensionless weights that balance retaining the full-feedback teacher against reducing reliance on influential fields. In the degenerate case $Z^t = 0$, we adopt the convention $\alpha_0^t = 1$ and $\alpha_j^t = 0$. With these coefficients, FIRE constructs an attribution-aware teacher

$$q_A^t(v) = \frac{q_{\mathbf{c}}^t(v)^{\alpha_0^t} \prod_{j=1}^B q_{-j}^t(v)^{\alpha_j^t}}{\sum_{u \in \mathcal{V}} q_{\mathbf{c}}^t(u)^{\alpha_0^t} \prod_{j=1}^B q_{-j}^t(u)^{\alpha_j^t}}, \tag{12}$$

with the geometric mean utilized because averaging in log-probability space induces the corresponding weighted combination of reverse-KL logit gradients, so the attribution weights have a direct interpretation in the resulting optimization direction (see Appendix F.5).

Then we measure the gradient induced by the attribution-aware target and define the radial contraction coefficient

$$\eta^t = \begin{cases} 1, & \|\mathbf{g}^t(q_A^t)\|_2 \le \rho_k^t, \\ \dfrac{\rho_k^t}{\|\mathbf{g}^t(q_A^t)\|_2}, & \|\mathbf{g}^t(q_A^t)\|_2 > \rho_k^t, \end{cases} \tag{13}$$

where η^t is the interpolation coefficient that keeps the attribution-corrected gradient within the local radius. It equals one when no contraction is necessary and decreases only enough to place an oversized signal on the boundary. Finally, FIRE interpolates geometrically between the detached student and the attribution-aware teacher:

$$q_{\mathrm{FIRE}}^t(v) = \frac{p^t(v)^{1-\eta^t} q_A^t(v)^{\eta^t}}{\sum_{u \in \mathcal{V}} p^t(u)^{1-\eta^t} q_A^t(u)^{\eta^t}}. \tag{14}$$

This construction scales the induced gradient exactly as $\mathbf{g}^t(q_{\mathrm{FIRE}}^t) = \eta^t \mathbf{g}^t(q_A^t)$. Field attribution therefore determines which feedback-dependent direction should be followed, while the final interpolation determines how far the student may move in that direction. Computing η^t after attribution guarantees that the final signal satisfies the radius. The corresponding derivation is provided in Appendix F.6. Importantly, FIRE's incorrect branch training differs from recent work that considers teacher-student geometric interpolation (Li et al., 2026b) by considering (i) *how each individual feedback field influences the updates* and (ii) *readjusting for excess gradient energy*. Additional specifics are provided in Appendix E.

Step 4: FIRE Loss Function. As a last step, we define $m_{i,t}$ to mark valid response tokens and $M_i = \sum_t m_{i,t}$, with i denoting the batch sample index. For a minibatch of N responses, the final objective of FIRE for the student update is

$$\mathcal{L}_{\mathrm{FIRE}}(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^N \frac{1}{M_i} \sum_t m_{i,t} \Big[\omega(\mathbf{y}_i) \ell_{+,i}^t(\boldsymbol{\theta}) + \big(1 - \omega(\mathbf{y}_i)\big) \mathrm{KL}\big(p_{\boldsymbol{\theta},i}^t \,\big\|\, \mathrm{sg}[q_{\mathrm{FIRE},i}^t]\big) \Big], \tag{15}$$

with both branch gradients bounded by $\rho_{k,i}^t$, as shown in Appendix F.7. The EMA teacher update remains unchanged from equation 4.

4 Experimental Results

Datasets, Models, Basic Setup. For evaluation of the FIRE methodology, we consider Qwen2.5-Instruct from the well-established Qwen family of foundation models (Qwen Team, 2024). Specifically, we consider sizes of 1.5B and 3B, with a larger model size (Qwen2.5-14B) considered in Appendix A.3. For Qwen2.5-1.5B-Instruct, we consider the GSM8K (Cobbe et al., 2021), AQuA-RAT (Ling et al., 2017), and ASDiv (Miao et al., 2020) datasets. For Qwen2.5-3B-Instruct, we consider ARC-Challenge (Clark et al., 2018), SuperGLUE WiC (Wang et al., 2019; Pilehvar & Camacho-Collados, 2019), and WinoGrande (Sakaguchi et al., 2021). We consider disjoint datasets across model sizes because some benchmarks become saturated at larger scales, exhibiting little to no improvement over the pretrained model during fine-tuning. For all experiments, we conduct linear warmup on the learning rate, with cosine decay after the peak learning rate is reached. Additionally, all experiments utilize AdamW as the optimizer (Loshchilov & Hutter, 2017). We utilize LoRA (Hu et al., 2021) as the PEFT method. For time efficiency, we evaluate on subsets of the test data. These specifics and more (e.g., additional hyperparameter settings) can be found in Appendices C and D.

Table 1: Final accuracy (%) across Qwen2.5-1.5B-Instruct and Qwen2.5-3B-Instruct and their associated datasets over three runs. FIRE exhibits superior performance to baselines across multiple datasets and model sizes.

Model	Algorithm	GSM8K	ASDiv	AQuA-RAT	Avg.
Qwen2.5-1.5B-Instruct	Full Context	52.43±2.10	76.04±3.25	53.47±1.67	60.65
	On-Policy SFT	50.35±3.35	77.26±0.30	50.17±1.83	59.26
	Veto	50.35±5.22	69.97±1.31	54.86±2.57	58.39
	TOP-D	0.00±0.00	0.00±0.00	12.50±3.25	4.17
	TrOPD	45.66±4.84	66.84±2.87	48.78±2.10	53.76
	SRPO	59.55±1.97	80.56±3.14	49.65±3.39	63.25
	DemoPSD	55.21±5.49	77.78±0.80	54.51±3.55	62.50
	SCOPE	60.07±5.76	77.60±2.27	50.00±3.12	62.56
	FIRE (Ours)	61.11±2.35	81.94±1.31	57.81±1.04	**66.96**
Model	**Algorithm**	**ARC-Challenge**	**WinoGrande**	**WiC**	**Avg.**
Qwen2.5-3B-Instruct	Full Context	82.12±2.17	67.71±2.90	63.54±0.52	71.12
	On-Policy SFT	83.51±3.05	64.58±0.52	63.54±0.52	70.54
	Veto	83.51±2.46	65.28±2.87	62.85±1.08	70.54
	TOP-D	83.33±1.88	67.53±2.96	55.21±4.51	68.69
	TrOPD	82.29±1.80	64.76±2.41	60.76±0.60	69.27
	SRPO	82.81±1.88	63.72±2.46	64.06±1.88	70.20
	DemoPSD	82.47±1.59	67.36±4.05	62.33±0.80	70.72
	SCOPE	84.38±2.71	64.41±1.67	65.97±0.80	71.59
	FIRE (Ours)	86.11±2.46	68.23±6.83	64.76±2.10	**73.03**

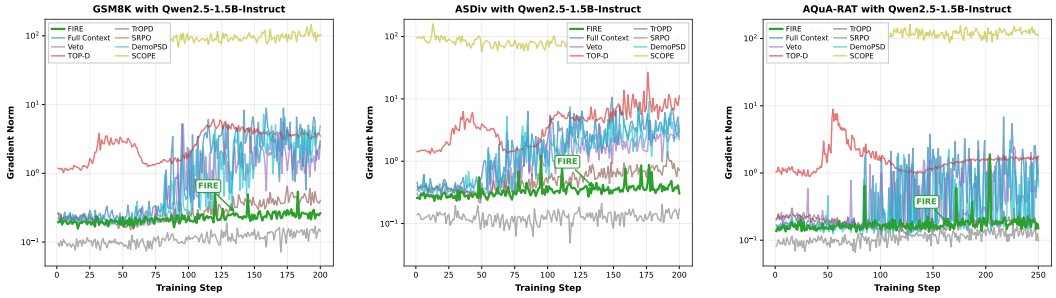
Baselines. To benchmark FIRE's performance, we consider eight relevant and recent baselines: standard feedback-based on-policy self-distillation (Full-Context), Sample Routed Policy Optimization (SRPO) (Li et al., 2026a), Trust Region Policy Distillation (TOP-D) (Xie et al., 2026), Trust Region On-Policy Distillation (TrOPD) (Xing et al., 2026), DemoPSD (Li et al., 2026b), Signal-Calibrated On-Policy Distillation Enhancement (SCOPE) (Zheng et al., 2026), Veto (Jang et al., 2026), and On-Policy Supervised Fine-tuning (On-Policy SFT) (Zhao et al., 2026a). We consider baselines that adopt approaches adjacent to FIRE to better quantify FIRE's performance differences. For example, TOP-D and TrOPD operate in a trust region similar to FIRE's Fisher-informed radius, while SCOPE conducts correct-incorrect branch self-distillation and SFT routing. DemoPSD similarly relies on calibration between the student and teacher distributions. Moreover, the baselines cover both self-distillation and regular on-policy distillation-based methods to better emphasize FIRE's advantages in a specifically self-distillation-based setting. More detailed specifics on how FIRE's methodology differs from the baselines can be found in Appendix E.

Hardware and Additional Results. All experiments were conducted on a server with an NVIDIA A100-40GB GPU, utilizing the PyTorch (Paszke et al., 2019) and HuggingFace (Jain, 2022) libraries. Additional experimental results, such as analysis on a larger model, ablation studies, etc. can be found in Appendix A.

4.1 Results

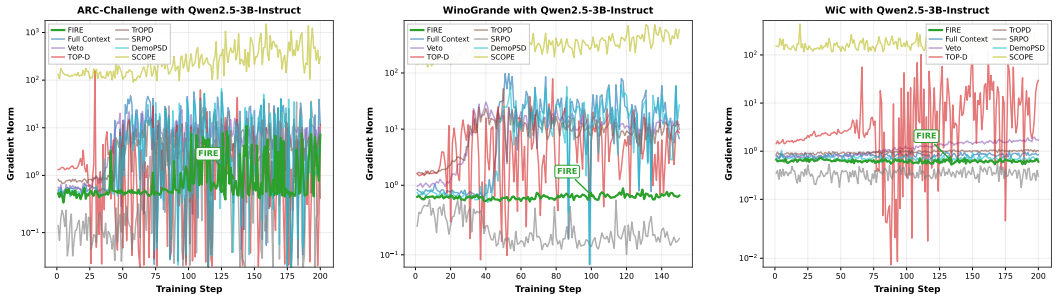
Performance in Comparison to Baselines. From the results presented in Table 1, we note the superiority in performance of the proposed FIRE method over the baselines. FIRE achieves the highest average performance across the datasets on both Qwen2.5-1.5B-Instruct and Qwen2.5-3B-Instruct. This high performance exhibited by FIRE across all settings is in contrast to some of the baselines (e.g., TOP-D on ASDiv). This indicates that FIRE's methodology is more conducive to stable performance during training, in comparison to other on-policy self-distillation and regular on-policy distillation methods. We verify this trend holds for a significantly larger model in Appendix A.3.

Gradient Norm Stability. In Figure 2, we examine whether FIRE's gradient-aware training induces more stable optimization dynamics. Although gradient clipping is applied to all methods



GSM8K ASDiv AQuA-RAT

(a) Qwen2.5-1.5B-Instruct



ARC-Challenge WinoGrande WiC

(b) Qwen2.5-3B-Instruct

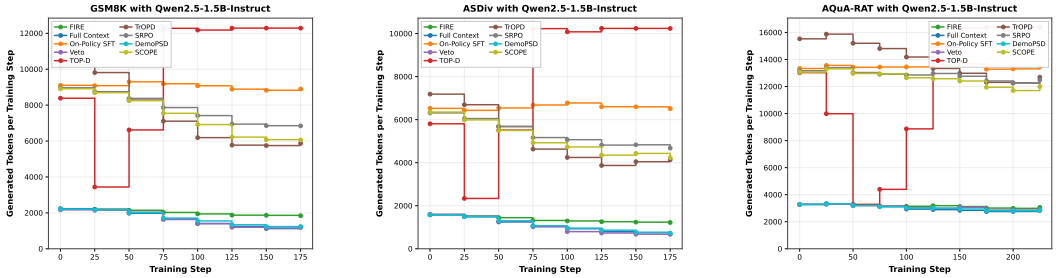
Figure 2: Gradient norm on (a) Qwen2.5-1.5B-Instruct and (b) Qwen2.5-3B-Instruct. The plotted values show the gradient norms before clipping. During training, gradient clipping constrains the norm to 1.0. The gradient norms are presented on a log-scale. FIRE exhibits much more stable training dynamics, with generally lower magnitude and less erratic oscillation.

during training, we report the *pre-clipping* gradient norms, i.e., the norms before clipping. Clipping can prevent individual updates from becoming excessively large, but it can also mask underlying optimization instability. We exclude On-Policy SFT from this analysis because it does not perform distillation.

Across the evaluated settings, FIRE exhibits substantially more stable pre-clipping gradient norms than the distillation baselines. In particular, several methods (e.g., Veto and DemoPSD) develop highly variable gradients as training progresses, despite being subject to the same clipping procedure. In contrast, FIRE maintains nearly constant gradients on five of the six datasets. ARC-Challenge is the only setting in which FIRE exhibits oscillation. Importantly, however, this behavior emerges later in training and remains considerably smaller in magnitude (log-scale plotting) than the fluctuations observed for several baselines. SRPO is the only other method that demonstrates consistent gradient stability, but this stability comes at the cost of a substantially more expensive training procedure, as shown in Appendix A.1.

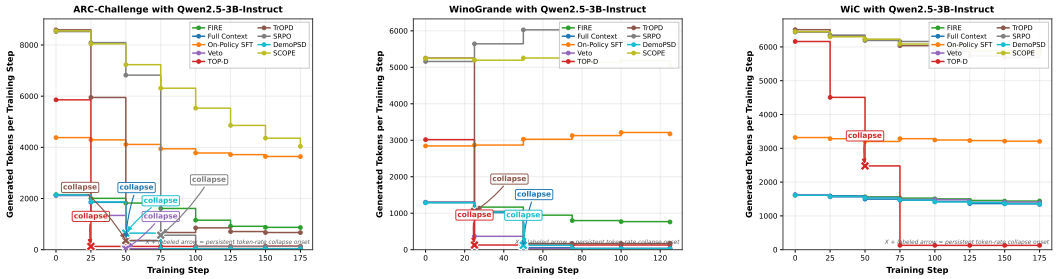
These results provide evidence that FIRE improves the *underlying optimization dynamics*, rather than relying on gradient clipping to suppress otherwise unstable updates. While clipping bounds the magnitude of the update, it does not make the training trajectories intrinsically stable. FIRE instead produces gradients that remain well-behaved even before this safeguard is applied, supporting the idea of its gradient-aware mechanism in stabilizing self-distillation.

Number of Tokens Generated per Step. We next examine the number of tokens generated at each training step in Fig. 3, which provides another view of training stability. While Fig. 2 characterizes stability at the optimization level, generation length reveals whether instability manifests in the model's output behavior. In particular, a sudden and sustained reduction in generated tokens can indicate *response collapse*, where training drives the model toward increasingly short responses.



GSM8K ASDiv AQuA-RAT

(a) Qwen2.5-1.5B-Instruct



ARC-Challenge WinoGrande WiC

(b) Qwen2.5-3B-Instruct

Figure 3: Number of auto-regressively generated tokens at each training step for each algorithm on (a) Qwen2.5-1.5B-Instruct and (b) Qwen2.5-3B-Instruct. FIRE's response length remains more resistant to collapse during training, maintaining its reasoning steps, in contrast to most baselines.

As seen in Fig. 3, FIRE is relatively robust to this behavior. On Qwen2.5-1.5B-Instruct, its generation lengths remain relatively steady across all three datasets, while on ARC-Challenge and WinoGrande with Qwen2.5-3B-Instruct, FIRE exhibits only a gradual decline rather than an abrupt collapse. In contrast, several baselines (e.g., TOP-D, TrOPD, Full Context) exhibit sharp falls in response length during training. On-Policy SFT is the only baseline exhibiting similar characteristics, but requires more generated tokens, making it computationally inefficient in comparison to FIRE.

Overall, both Figs. 2 and 3 show that FIRE's stability is not limited to the gradient norms but is simultaneously exhibited in response outputs. FIRE simultaneously maintains controlled pre-clipping gradients and avoids the abrupt response-length collapse exhibited by several baselines. Importantly, we additionally note the collapse correlates closely with the onset of increased gradient instability in Fig. 2.

5 Conclusion

In this paper, we introduced FIRE (**F**isher-**I**nformed **RE**calibration), a novel methodology for *stabilizing feedback-based on-policy self-distillation in LLMs*. FIRE routes correct and incorrect on-policy responses through dual-branching, whereby correct responses are trained via Fisher-informed re-weighted on-policy SFT, while incorrect responses retain feedback-conditioned self-distillation with attribution-aware teacher recalibration. The attribution-aware recalibration identifies *which direction* feedback should move the model, with the softmax Fisher trace primarily, in conjunction with gradient magnitudes and the learning rate, governing *how far* updates should move the model across both branches. Across multiple datasets and model sizes, FIRE consistently achieves strong downstream performance after fine-tuning while exhibiting substantially more stable pre-clipping gradient norms and greater resistance to response-length collapse than existing baselines. Potential future work could explore more efficient methodologies for characterizing leave-one-field-out counterfactuals when the number of feedback fields is large.

AI USE STATEMENT

In this work, we used generative AI tools for improving clarity of the manuscript (paraphrasing, rewording, partial creation of Fig. 1, and creation of tables in the main text and Appendix). Moreover, it was used as an aid for the derivations in Appendix F and code implementation (e.g., implementation of proposed methodology and baselines). Research ideation of the proposed methodology was derived from the authors, and the rest of the required disclosure tasks (e.g., generation of new synthetic datasets) are not applicable to this work. We have reviewed all AI-assisted work. For example, we have confirmed the accuracy of the derivations of Appendix F and have checked the accuracy of the re-worded text. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We have provided the implementation of the FIRE methodology in full in the supplementary material of the submission, including configuration files. Moreover, details on hyperparameter and prompt template settings can be found in Sec. 4, Appendix C, and Appendix D. The pseudocode of the FIRE method can additionally be found in Appendix B. Lastly, the derivations for the equations in Sec. 3.2 can be found in full in Appendix F.

REFERENCES

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*, 2018.

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.

Wenzhi Fang, Dong-Jun Han, Liangqi Yuan, Seyyedali Hosseinalipour, and Christopher G Brinton. Federated sketching lora: On-device collaborative fine-tuning of large language models. *arXiv preprint arXiv:2501.19389*, 2025.

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

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

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. *arXiv preprint arXiv:2106.09685*, 2021.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, et al. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026.

Shashank Mohan Jain. Hugging face. In *Introduction to transformers for NLP: With the hugging face library and models to solve problems*, pp. 51–67. Springer, 2022.

Ijun Jang, Jewon Yeom, Juan Yeo, Hyunggyu Lim, and Taesup Kim. Stable on-policy distillation through adaptive target reformulation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 42217–42227, 2026.

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Seohyun Lee, Wenzhi Fang, Dong-Jun Han, Seyyedali Hosseinalipour, and Christopher G Brinton. Tap: Two-stage adaptive personalization of multi-task and multi-modal foundation models in federated learning. *arXiv preprint arXiv:2509.26524*, 2025.

Seohyun Lee, Wenzhi Fang, Dong-Jun Han, Seyyedali Hosseinalipour, and Christopher G Brinton. Self-play enhancement via advantage-weighted refinement in online federated llm fine-tuning with real-time feedback. *arXiv preprint arXiv:2605.07977*, 2026.

Gengsheng Li, Tianyu Yang, Junfeng Fang, Mingyang Song, Mao Zheng, Haiyun Guo, Dan Zhang, Jinqiao Wang, and Tat-Seng Chua. Unifying group-relative and self-distillation policy optimization via sample routing. *arXiv preprint arXiv:2604.02288*, 2026a.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: Long papers)*, pp. 4582–4597, 2021.

Yunhe Li, Hao Shi, Wenhao Liu, Mengzhe Ruan, Hanxu Hou, Zhongxiang Dai, Shuang Qiu, and Linqi Song. Demopsd: Disagreement-modulated policy self-distillation. *arXiv preprint arXiv:2607.02502*, 2026b.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *ACL*, 2017.

Qidong Liu, Xian Wu, Xiangyu Zhao, Yuanshao Zhu, Derong Xu, Feng Tian, and Yefeng Zheng. When moe meets llms: Parameter efficient fine-tuning for multi-task medical applications. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pp. 1104–1114, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Shen-yun Miao, Chao-Chun Liang, and Keh-Yih Su. A diverse corpus for evaluating and developing english math word problem solvers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 975–984, 2020.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey. *arXiv preprint arXiv:2402.06196*, 2024.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Mary Phuong and Christoph Lampert. Towards understanding knowledge distillation. In *International conference on machine learning*, pp. 5142–5151. PMLR, 2019.

Mohammad Taher Pilehvar and Jose Camacho-Collados. WiC: The word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of NAACL-HLT*, 2019.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL `https://qwenlm.github.io/blog/qwen2.5/`.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*, 2026.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint 1905.00537*, 2019.

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.

Zhengpeng Xie, Li Lyna Zhang, Zeke Xie, and Mao Yang. Trust region policy distillation. *arXiv preprint arXiv:2607.04751*, 2026.

Xingrun Xing, Haoqing Wang, Boyan Gao, Ziheng Li, and Yehui Tang. Trust region on-policy distillation. *arXiv preprint arXiv:2606.01249*, 2026.

Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. A survey on knowledge distillation of large language models. *arXiv preprint arXiv:2402.13116*, 2024.

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.

Dan Zhang, Tao Feng, Lilong Xue, Yuandong Wang, Yuxiao Dong, and Jie Tang. Parameter-efficient fine-tuning for foundation models. *arXiv preprint arXiv:2501.13787*, 2025.

Anhao Zhao, Ziyang Chen, Junlong Tong, Yingqi Fan, Fanghua Ye, Shuhao Li, Yunpu Ma, Wenjie Li, and Xiaoyu Shen. On-policy supervised fine-tuning for efficient reasoning. *arXiv preprint arXiv:2602.13407*, 2026a.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026b.

Binbin Zheng, Xing Ma, Yiheng Liang, Jingqing Ruan, Xiaoliang Fu, Kepeng Lin, Benchang Zhu, Ke Zeng, and Xunliang Cai. Scope: Signal-calibrated on-policy distillation enhancement with dual-path adaptive weighting. *arXiv preprint arXiv:2604.10688*, 2026.

Appendix

A Additional Experimental Results **14**

A.1 Auto-regressively Generated Tokens during Training vs. Accuracy 14

A.2 Training Time vs. Accuracy 14

A.3 Consideration for Large Model Size (14B) 15

A.3.1 Accuracy of Qwen2.5-14B-Instruct 15

A.3.2 Gradient Norm of Qwen2.5-14B-Instruct 16

A.4 Ablation Study on FIRE components 16

A.5 Learning Rate Impact 17

A.5.1 Ablation on Differing Learning Rates with Adjusted Nominal Rate 17

A.5.2 Gradient Norm Stability Across Learning Rates 17

B Pseudocode of FIRE **19**

C Hyperparameter Settings **19**

C.1 Model and Dataset-Specific Settings 19

C.2 Method-Specific Settings 20

C.3 Micro-Batch Sizes and Gradient Accumulation 20

D Prompt and Feedback Templates **21**

D.1 Per-dataset Templates 21

D.2 Feedback Field Construction 25

E Comparison with Closely Related Methods **26**

F Additional FIRE Details **27**

F.1 Execution Order and Feedback Conditioning 27

F.2 Exact Reverse-KL Logit Gradient 27

F.3 Fisher Scale and Learning-Rate Calibration 28

F.4 Correct-Branch Re-weighting as Gradient Projection 28

F.5 Exact Leave-One-Field-Out Influence 28

F.6 Incorrect-Branch Radial Projection 29

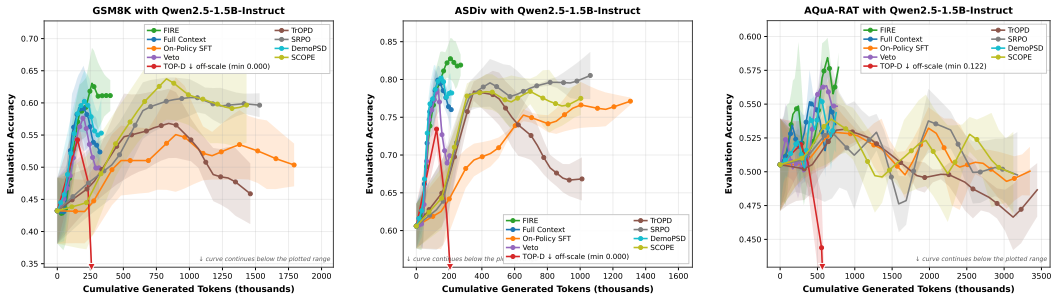
F.7 Unified Bound and Its Scope 30

F.8 Computation and Memory 30

A Additional Experimental Results

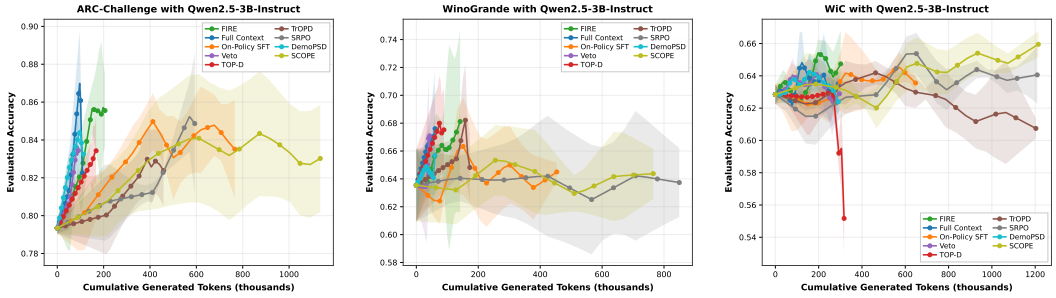
Here, we present additional experimental results in addition to those presented in Sec. 4. If hyperparameter settings differ for certain experiments, they will be stated in their corresponding subsections.

A.1 Auto-regressively Generated Tokens during Training vs. Accuracy



GSM8K ASDiv AQuA-RAT

(a) Qwen2.5-1.5B-Instruct



ARC-Challenge WinoGrande WiC

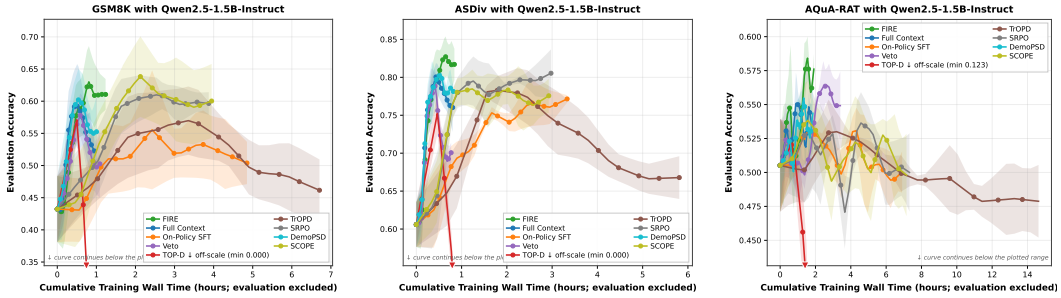
(b) Qwen2.5-3B-Instruct

Figure 4: Number of cumulatively auto-regressively generated tokens versus accuracy of each algorithm on (a) Qwen2.5-1.5B-Instruct and (b) Qwen2.5-3B-Instruct. FIRE offers the best tradeoff between training efficiency and accuracy.

Here, we consider the number of cumulatively generated tokens auto-regressively versus test accuracy across FIRE and the baselines in Fig. 4. For the curves, each run's trajectory is linearly interpolated onto a shared grid over the range jointly covered by all runs for that method. From Fig. 4, we note that across five of the six datasets, FIRE offers the best tradeoff between training efficiency and accuracy. While SCOPE attains a higher final accuracy than FIRE on WiC, it is computationally more expensive to reach that point. Moreover, observations noted in Sec. 4, specifically that FIRE induces better stability properties can also be seen in Fig. 4 in comparison to SCOPE (e.g., continuous decrease in accuracy of SCOPE on AQuA-RAT). This is true for other datasets as well. For example, while TOP-D comes close to beating FIRE on WinoGrande, it is the worst performing method on all Qwen2.5-1.5B-Instruct-based datasets and WiC; it also lags behind on ARC-Challenge. Additional examples with similar observations include TrOPD, DemoPSD, and Full Context. Methods that exhibit more stability, such as SRPO or On-Policy SFT, are computationally much more expensive than FIRE. This is because methods such as SRPO rely on group calculations, which require generating multiple completions for one input sample.

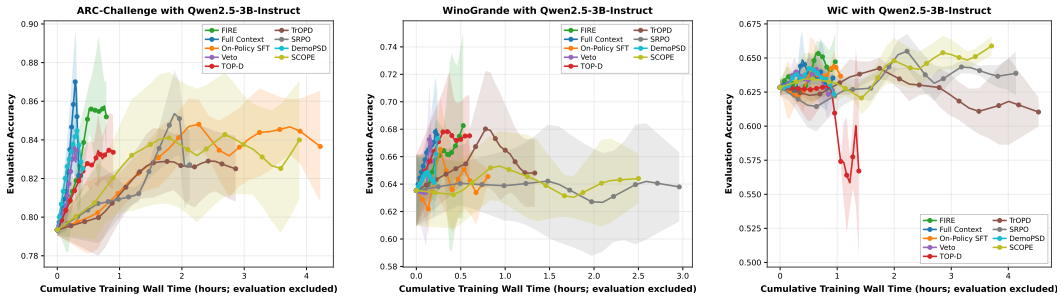
A.2 Training Time vs. Accuracy

In Fig. 5, we consider whether the observations in the previous subsection (Appendix A.1) still hold in relation to the wall-clock training time. Similarly to Appendix A.1, each run's trajectory is



GSM8K ASDiv AQuA-RAT

(a) Qwen2.5-1.5B-Instruct



ARC-Challenge WinoGrande WiC

(b) Qwen2.5-3B-Instruct

Figure 5: Training time versus accuracy of each algorithm on (a) Qwen2.5-1.5B-Instruct and (b) Qwen2.5-3B-Instruct. FIRE offers the best tradeoff between training time and accuracy.

linearly interpolated onto a shared grid over the range jointly covered by all runs for that method. We can see from Fig. 5 that the observations still hold. FIRE offers the best tradeoff between training time and accuracy, with FIRE being multiple hours faster than some of the better performing baselines (e.g., SRPO). Similarly to Appendix A.1, while we note SCOPE does better in final accuracy than FIRE on WiC, FIRE *consistently performs well on all settings*, in comparison to Full Context, DemoPSD, SCOPE, and TOP-D, for example. Some of the baselines' slower performance can be attributed to requiring a smaller micro-batch size and therefore more gradient accumulation steps, as outlined in Appendix C. However, this configuration was necessitated purely by resource constraints. Importantly, the wall-clock results exhibit characteristics similar to those in Fig. 4. A consequence of these results is that despite the $\mathcal{O}(B)$ operational overhead for the leave-one-out counterfactuals in Sec. 3.2, FIRE still exhibits a better wall-clock time-accuracy tradeoff.

A.3 Consideration for Large Model Size (14B)

In comparison to the smaller model sizes exhibited in Sec. 4, we seek to see if FIRE's advantages will persist on large model sizes. Therefore, we consider Qwen2.5-14B-Instruct, fine-tuned on six categories of the more challenging MMLU-Pro (Wang et al., 2024) dataset. Specifically, we consider categories of biology, chemistry, computer science (CS), economics, engineering, and physics, which were picked via random selection.

A.3.1 Accuracy of Qwen2.5-14B-Instruct

In Table 2, we present the overall performance of FIRE in comparison to the baselines on six categories of MMLU-Pro on Qwen2.5-14B-Instruct. From Table 2 we can note that FIRE's performance advantages persist on a model that is significantly larger than the 1.5B and 3B sizes explored in Sec. 4. Moreover, we can note that FIRE is the best performing method across four of the six explored categories, with high performance overall. Overall, these results indicate that FIRE's methodology

Table 2: Final accuracy (%) on MMLU-Pro across subject domains using Qwen2.5-14B-Instruct. FIRE achieves the strongest overall performance across the evaluated methods.

Model	Algorithm	Biology	Chemistry	CS	Economics	Engineering	Physics	Overall
Qwen2.5-14B-Instruct	Full Context	75.00±2.95	46.88±1.47	54.17±2.95	64.58±0.00	40.62±1.47	40.62±4.42	53.65±0.25
	TOP-D	73.96±7.37	46.88±4.42	53.12±4.42	66.67±0.00	38.54±4.42	42.71±7.37	53.65±0.74
	TrOPD	68.75±0.00	48.96±7.37	53.12±4.42	62.50±0.00	40.62±7.37	43.75±0.00	52.95±1.72
	SRPO	73.96±1.47	45.83±2.95	52.08±2.95	65.62±1.47	40.62±4.42	42.71±1.47	53.47±0.98
	DemoPSD	77.08±2.95	46.88±1.47	54.17±2.95	65.62±1.47	39.58±0.00	40.62±4.42	53.99±1.23
	SCOPE	70.83±8.84	20.83±0.00	47.92±8.84	55.21±7.37	19.79±13.26	27.08±5.89	40.28±7.37
	FIRE (Ours)	77.08±2.95	52.08±2.95	60.42±5.89	62.50±0.00	42.71±1.47	38.54±1.47	**55.56**±1.47

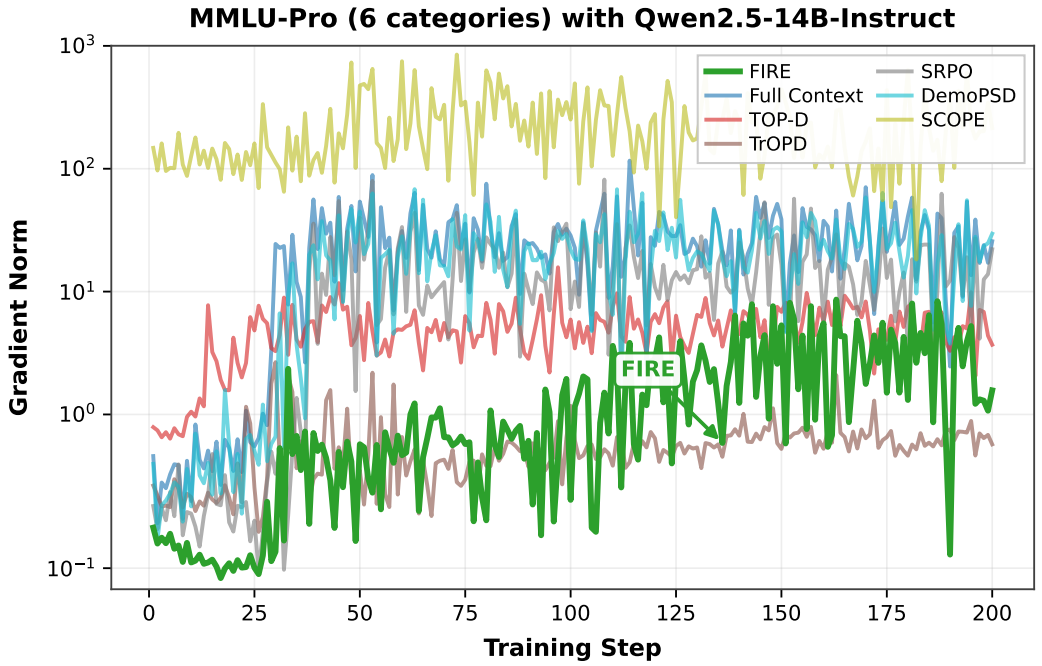


Figure 6: Gradient norms before clipping on six MMLU-Pro categories. Consistent with Sec. 4, FIRE maintains controlled gradient magnitudes at the 14B scale relative to most baselines.

is robust to differing model sizes, with performance not dependent on larger or smaller parameter sets.

A.3.2 Gradient Norm of Qwen2.5-14B-Instruct

Similar to Sec. 4, we examine whether FIRE's optimization stability persists at a significantly larger model scale (Qwen2.5-14B-Instruct) in Fig. 6. FIRE maintains substantially lower gradient norms than most baselines, with only TrOPD exhibiting consistently smaller magnitudes. While TOP-D shows somewhat lower oscillation, its gradient norms remain larger than FIRE's. Both TOP-D and TrOPD operate under a trust-region regime, which may explain their smoother gradient behavior. FIRE instead regulates updates through its Fisher-trace-based radius, yielding still relatively controlled gradients at the 14B scale. Although FIRE is not the smoothest method in this setting, the results remain consistent with the broader optimization stability observed in Sec. 4. Moreover, TOP-D and TrOPD are not consistently smooth across all settings, in contrast with FIRE.

A.4 Ablation Study on FIRE Components

Here, we conduct ablation studies on the individual components of the FIRE methodology. For No Attribution, we consider the exclusion of leave-one-out counterfactuals and utilize the full feedback (i.e., no equation 12). For No Projection, the final radial projection for incorrect responses is not conducted (i.e., no equation 13 and equation 14). Hard Gradient Excision corresponds to remov-

Table 3: Final accuracy (%) for FIRE component ablations across Qwen2.5-1.5B-Instruct and Qwen2.5-3B-Instruct and their associated datasets over three runs. Removing attribution, projection, or replacing projection with hard gradient excision degrades average performance, demonstrating the contribution of the full FIRE objective.

Model	Algorithm	GSM8K	ASDiv	AQuA-RAT	Avg.
Qwen2.5-1.5B-Instruct	No Attribution	61.11±1.20	74.13±3.05	50.35±4.21	61.86
	No Projection	56.77±7.22	69.10±11.14	51.04±0.52	58.97
	Hard Gradient Excision	58.33±4.13	66.32±13.94	53.12±1.80	59.26
	FIRE (Ours)	60.94±2.71	78.30±3.84	52.26±2.46	**63.83**
Model	**Algorithm**	**ARC-Challenge**	**WinoGrande**	**WiC**	**Avg.**
Qwen2.5-3B-Instruct	No Attribution	85.94±2.27	65.45±5.22	65.28±0.80	72.22
	No Projection	86.63±2.17	66.32±6.04	64.06±2.76	72.34
	Hard Gradient Excision	86.28±2.57	66.49±5.00	64.41±1.20	72.40
	FIRE (Ours)	86.11±2.46	68.23±6.83	64.76±2.10	**73.03**

ing the feedback field with the largest leave-one-out influence on the teacher-induced gradient and distilling on the resulting counterfactual target. For the Qwen2.5-1.5B-Instruct, we use a uniformly higher learning-rate setting across all variants to better emphasize component behavior advantages of FIRE in more unstable optimization landscapes. For AQuA-RAT, the peak $a_k = 5 \times 10^{-6}$ with $a_0 = 5\times10^{-7}$ over 200 steps. For GSM8K and ASDiv, it is 6×10^{-6}/1×10^{-6} and 1×10^{-5}/1×10^{-6} respectively.

From the results in Table 3, we note FIRE's performance advantages. In particular, we observe that FIRE outperforms all three ablations across both Qwen2.5-1.5B-Instruct and Qwen2.5-3B-Instruct on average. Therefore, the individual components that are introduced in the FIRE methodology result in an overall net positive improvement of performance.

A.5 Learning Rate Impact

Since the nominal rate a_0 hyperparameter is designed to cope with potential instability in higher learning rate regimes, we examine whether a_0 properly plays the role of stabilizing training across differing learning rates. For this, we consider a_0 set to 1.0×10^{-7}, 5.0×10^{-7}, 1.0×10^{-8}, 5.0×10^{-7}, 5.0×10^{-7}, and 5.0×10^{-8} for GSM8K, ASDiv, AQuA-RAT, ARC-Challenge, WinoGrande, and WiC respectively.

A.5.1 Ablation on Differing Learning Rates with Adjusted Nominal Rate

In Table 4, we sweep over four differing learning rates across Qwen2.5-1.5B-Instruct and Qwen2.5-3B-Instruct. While we note that setting the learning rate too high can still lead to suboptimal performance, we observe that FIRE successfully prevents collapse across all observed learning rates and still maintains acceptable levels of performance in all settings.

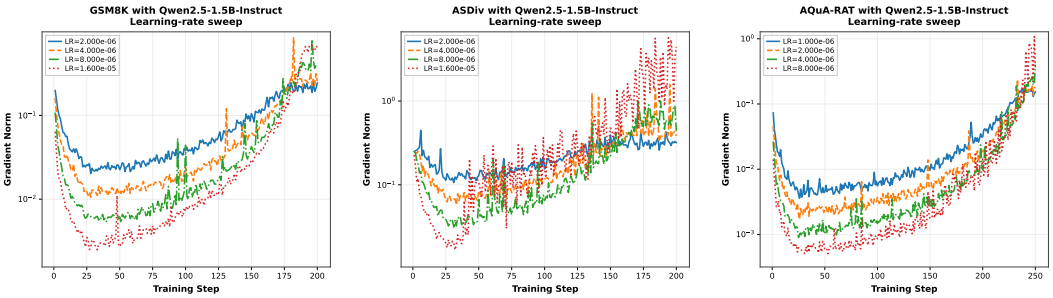
A.5.2 Gradient Norm Stability Across Learning Rates

In addition, we consider how the varying learning rates affect the norms of the gradients in Fig. 7. We note across all datasets with the exception of ARC-Challenge, the gradient norms stay relatively stable across both lower and larger learning rate regimes, with most settings below the clipping threshold of 1.0. This indicates that the observations in Appendix A.5.1 correspond mostly with the observations in Fig. 7. While we note relatively high oscillation for ARC-Challenge across three differing learning rates, we would emphasize the relative magnitude of the gradients *are small in comparison to the baselines*. For example, with ARC-Challenge, the magnitude never peaks past 10.0, while they nearly reach 100.0 on DemoPSD, Full Context, Veto, and TOP-D, with the gradient norms exceeding 100.0 for SCOPE in Fig. 2. Therefore, even in the higher learning rate scenario, FIRE achieves substantially more constrained gradients than multiple baselines under a more stable learning rate regime.

Table 4: Learning-rate ablation for FIRE. We sweep the learning rate over $\{0.5\times, 1\times, 2\times, 4\times\}$ the default learning rate utilized for each dataset-model pair.

Qwen2.5-1.5B-Instruct			
LR Multiplier	**GSM8K**	**ASDiv**	**AQuA-RAT**
$0.5\times$	54.51 ± 5.22	76.91 ± 4.55	54.86 ± 1.59
$1\times$	63.72 ± 3.01	81.94 ± 3.35	53.82 ± 1.59
$2\times$	61.63 ± 4.21	81.77 ± 2.39	52.78 ± 2.41
$4\times$	54.34 ± 4.21	78.47 ± 1.67	50.17 ± 1.31
LR ($1\times$): GSM8K/ASDiv $= 4.0 \times 10^{-6}$; AQuA-RAT $= 2.0 \times 10^{-6}$.			

Qwen2.5-3B-Instruct			
LR Multiplier	**ARC-Challenge**	**WinoGrande**	**WiC**
$0.5\times$	83.85 ± 2.39	65.80 ± 0.30	59.72 ± 5.35
$1\times$	83.16 ± 1.08	67.53 ± 0.60	61.46 ± 4.51
$2\times$	82.64 ± 1.59	66.84 ± 2.41	58.51 ± 5.76
$4\times$	81.77 ± 1.88	65.28 ± 3.01	55.56 ± 6.01
LR ($1\times$): ARC-Challenge/WinoGrande $= 4.0 \times 10^{-6}$; WiC $= 1.0 \times 10^{-6}$.			



GSM8K ASDiv AQuA-RAT

(a) Qwen2.5-1.5B-Instruct



ARC-Challenge WinoGrande WiC

(b) Qwen2.5-3B-Instruct

Figure 7: Gradient norms on (a) Qwen2.5-1.5B-Instruct and (b) Qwen2.5-3B-Instruct across varying learning rates.

B PSEUDOCODE OF FIRE

Following the notation of Sec. 3.2, Algorithm 1 gives the optimizer-step implementation of FIRE. All quantities used to construct β^t, α_j^t, η^t, and the teacher targets are detached. Gradients are taken only through the live student distribution in the final branch loss.

Algorithm 1 FIRE at optimizer step k

Require: Student $\pi_{\boldsymbol{\theta}_k}$, EMA teacher $\pi_{\bar{\boldsymbol{\theta}}_k}$, minibatch $\mathcal{B}$, verifier ω, feedback builder $\mathcal{F}$, learning rate a_k, nominal rate a_0

for all $\mathbf{x}_i \in \mathcal{B}$ **do**
  Sample one on-policy response $\mathbf{y}_i \sim \pi_{\boldsymbol{\theta}_k}(\cdot \mid \mathbf{x}_i)$.
  Score the fixed response with the live student, set $p_i^t \leftarrow \mathrm{sg}[p_{\boldsymbol{\theta}_k,i}^t]$ and compute $\rho_{k,i}^t$ by equation 7
  if $\omega(\mathbf{y}_i) = 1$ **then** ▷ Correct branch
    Compute $\mathbf{g}_{+,i}^t = p_i^t - \mathbf{e}_{y_{i,t}}$ and $\beta_i^t = \min(1, \rho_{k,i}^t / \|\mathbf{g}_{+,i}^t\|_2)$
    $\ell_i \leftarrow$ token-averaged re-weighted on-policy SFT from equation 8
  else ▷ Incorrect branch
    Build $\mathbf{c}_i = \mathcal{F}(\mathbf{x}_i, \mathbf{y}_i) = (b_{i,1}, \ldots, b_{i,B})$
    Teacher-force the EMA on the same fixed $\mathbf{y}_i$ under $\mathbf{c}_i$ and each $\mathbf{c}_{i,\setminus j}$ to obtain $q_{\mathbf{c},i}^t$ and $\{q_{-j,i}^t\}_{j=1}^B$
    Compute gradients and energies $\mathbf{g}^t(\cdot)$, $E_{\mathbf{c},i}^t$, and $\{E_{j,i}^t\}_{j=1}^B$ using equation 6 and equation 9
    Compute χ_i^t, $\{w_{j,i}^t\}_{j=1}^B$, and $\{\alpha_{j,i}^t\}_{j=0}^B$ using equation 10–equation 11
    Construct $q_{A,i}^t$, then η_i^t and $q_{\mathrm{FIRE},i}^t$ using equation 12–equation 14
    $\ell_i \leftarrow M_i^{-1} \sum_t m_{i,t} \mathrm{KL}\big(p_{\boldsymbol{\theta},i}^t \big\| \mathrm{sg}[q_{\mathrm{FIRE},i}^t]\big)$
  end if
end for
$\mathcal{L}_{\mathrm{FIRE}} \leftarrow |\mathcal{B}|^{-1} \sum_{i \in \mathcal{B}} \ell_i$
$\boldsymbol{\theta}_{k+1} \leftarrow \mathrm{OptStep}(\boldsymbol{\theta}_k, \nabla_{\boldsymbol{\theta}_k} \mathcal{L}_{\mathrm{FIRE}}, a_k)$
$\bar{\boldsymbol{\theta}}_{k+1} \leftarrow \mu \bar{\boldsymbol{\theta}}_k + (1-\mu)\boldsymbol{\theta}_{k+1}$

C HYPERPARAMETER SETTINGS

This section reports the settings used for the primary experiments in Sec. 4 and scaling experiments. The optimizer learning rate is shared by FIRE and the corresponding baselines for a given model-dataset pair. FIRE's nominal rate a_0 affects only the FIRE radius in equation 7.

Across experiments, we use AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.95$, weight decay 0.01, linear warmup followed by cosine decay, and gradient clipping at norm 1.0. We use `bfloat16`, gradient checkpointing, and LoRA with rank 16, LoRA scaling 32, and dropout 0.0 on the `q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, and `down_proj` modules. The effective batch size is 8 in all resolved runs. Generation uses top-$p = 0.95$ and temperature 0.8, except On-Policy SFT, which uses temperature 1.0. On-Policy SFT uses a differing sampling temperature because its sampled trajectories are directly reused as hard supervised targets. All distillation-based methods use the same sampling temperature. The maximum prompt and feedback budgets are 1280 and 768 tokens, respectively. FIRE uses four feedback fields during implementation.

C.1 MODEL AND DATASET-SPECIFIC SETTINGS

Table 5: Resolved model- and dataset-specific settings. The EMA column reports the student update weight $1 - \mu$ in equation 4. The nominal rate a_0 is used only by FIRE.

Model	Dataset	Peak LR	a_0	$1-\mu$	Warmup	Steps	Max context	Max new	Train pool	Eval. n
Qwen2.5-1.5B	GSM8K	4×10^{-6}	4×10^{-6}	0.03	25	200	2560	384	1024	192
Qwen2.5-1.5B	ASDiv	4×10^{-6}	4×10^{-6}	0.03	25	200	2304	320	1600	192
Qwen2.5-1.5B	AQuA-RAT	2×10^{-6}	2×10^{-6}	0.03	25	250	3072	512	2048	192
Qwen2.5-3B	ARC-Challenge	4×10^{-6}	4×10^{-6}	0.01	20	200	2560	320	2048	192
Qwen2.5-3B	WinoGrande	4×10^{-6}	4×10^{-6}	0.01	15	150	2560	320	2048	192
Qwen2.5-3B	WiC	1×10^{-6}	1×10^{-6}	0.01	15	200	2560	320	2048	192
Qwen2.5-14B	MMLU-Pro (six categories)	1×10^{-5}	2×10^{-6}	0.01	20	200	3072	256	1536	288

C.2 METHOD-SPECIFIC SETTINGS

Table 6 lists only the principal method-specific hyperparameters. Pure implementation constants such as numerical epsilons are excluded. We retain the notation of the original baseline papers for baseline-specific hyperparameters. Some symbols may therefore overlap with notation used elsewhere in this paper, but such symbols are constrained to only Table 6 and should not be interpreted as referring to the quantities defined in the main text.

Table 6: Important method-specific hyperparameters for run configurations. "Small" denotes the Qwen2.5-1.5B/3B experiments, and "scaling" denotes the larger model experiments.

Method	**Method-specific settings**
Full Context	No additional method-specific hyperparameter beyond the shared optimizer, generation, EMA, and LoRA settings.
On-Policy SFT	Group size 4 (1.5B) / 2 (3B).
Veto	β decays linearly from 0.8 to 0.0, forward-KL objective.
TOP-D	Group size 4, proximal coefficient $\alpha = 0.2$, PPO clipping $\epsilon = 0.2$, one off-policy epoch, one prompt per internal minibatch.
TrOPD	Group size 4, guidance coefficient $\beta = 10^{-3}$, maximum guided prefix length 384.
SRPO	Group size 4 for the small-model experiments and 2 for scaling, entropy coefficient $\beta = 1.0$, PPO clipping $\epsilon = 0.2$, ratio cap 2.0.
DemoPSD	Maximum leakage coefficient $\alpha_{\max} = 0.15$, disagreement scale $\beta = 50$.
SCOPE	Group size 4 for the small-model experiments and 2 for scaling, group-weight temperature 1.0.
FIRE (ours)	Four structured feedback fields, nominal rate a_0 as in Table 5, excess-ℓ_2 gradient-energy attribution, learning-rate step normalization and Fisher-informed re-weighted correct-branch SFT enabled.

C.3 MICRO-BATCH SIZES AND GRADIENT ACCUMULATION

Table 7: Resolved micro-batch size / gradient-accumulation steps. All entries have effective batch size 8. A dash denotes a method not used in that reported model-scale comparison.

Method	**Qwen 1.5B**	**Qwen 3B**	**Qwen 14B**
Full Context	8/1	8/1	2/4
On-Policy SFT	2/4	2/4	–
Veto	8/1	8/1	–
TOP-D	2/4	2/4	1/8
TrOPD	2/4	2/4	1/8
SRPO	2/4	2/4	1/8
DemoPSD	8/1	8/1	2/4
SCOPE	2/4	2/4	1/8
FIRE (ours)	8/1	8/1	2/4

Table 7 reports the micro-batch size and number of gradient accumulation steps used for each method while maintaining the same effective batch size. For baselines requiring more accumulation steps than FIRE or Full Context, this choice was necessitated solely by resource constraints, rather than to confer a computational advantage on FIRE or Full Context. Moreover, Appendix A.1 and Fig. 4 evaluate accuracy as a function of the number of autoregressively generated tokens, demonstrating that FIRE achieves a more favorable accuracy-generation trade-off. This provides evidence that FIRE's computational efficiency gains are intrinsic to the method rather than due to the batching configuration.

D PROMPT AND FEEDBACK TEMPLATES

D.1 PER-DATASET TEMPLATES

We provide the prompt templates used in our experiments below. Text in {blue braces} denotes fields populated from the corresponding dataset or sampled trajectory. For FIRE, the teacher feedback context shown below is used for incorrect responses. Correct responses are instead routed to Fisher-informed re-weighted on-policy SFT as described in Sec. 3.

Prompt Template — GSM8K

```
Solve the following task.

Work through the problem step by step, showing your arithmetic.
Then, on a new final line, write the final answer as ####  followed
by the number only (for example #### 42).  Write nothing after that
line.

{question}
```

Teacher feedback context:

```
You are the feedback-conditioned teacher in on-policy
self-distillation.  The earlier student attempt will appear as the
assistant continuation.  Use the structured feedback context to
revise the continuation distribution for the same task.  Preserve
useful reasoning tokens and end with the task's normal final-answer
line.  Do not quote or describe the context.

Task:
{dataset task prompt}

Structured feedback context:
[Context block 1]
{verifier result}
[Context block 2]
{parser record}
[Context block 3]
{verifier provenance}
[Context block 4]
{response-format diagnostics}
```

Prompt Template — ASDiv

```
Solve the following task.

Work through the problem step by step, showing your arithmetic.
Then, on a new final line, write the final answer as ####  followed
by the number only (for example #### 42).  Write nothing after that
line.

{body}
{question}
```

Teacher feedback context:

```
You are the feedback-conditioned teacher in on-policy
self-distillation.  The earlier student attempt will appear as the
assistant continuation.  Use the structured feedback context to
revise the continuation distribution for the same task.  Preserve
useful reasoning tokens and end with the task's normal final-answer
line.  Do not quote or describe the context.

Task:
```

{dataset task prompt}

Structured feedback context:
[Context block 1]
{verifier result}
[Context block 2]
{parser record}
[Context block 3]
{verifier provenance}
[Context block 4]
{response-format diagnostics}

Prompt Template — AQuA-RAT

Solve the following task.

Reason step by step about the options. Then, on a new final line, write Answer: followed by the single option letter only (for example Answer: B). Write nothing after that line.

{question}

Choices:
A. {choice A}
B. {choice B}
C. {choice C}
D. {choice D}
E. {choice E}

Teacher feedback context:

You are the feedback-conditioned teacher in on-policy self-distillation. The earlier student attempt will appear as the assistant continuation. Use the structured feedback context to revise the continuation distribution for the same task. Preserve useful reasoning tokens and end with the task's normal final-answer line. Do not quote or describe the context.

Task:
{dataset task prompt}

Structured feedback context:
[Context block 1]
{verifier result}
[Context block 2]
{parser record}
[Context block 3]
{verifier provenance}
[Context block 4]
{response-format diagnostics}

Prompt Template — ARC-Challenge

Solve the following task.

Reason step by step about the options. Then, on a new final line, write Answer: followed by the single option letter only (for example Answer: B). Write nothing after that line.

{question}

Choices:

{normalized lettered choices}

Teacher feedback context:

You are the feedback-conditioned teacher in on-policy self-distillation. The earlier student attempt will appear as the assistant continuation. Use the structured feedback context to revise the continuation distribution for the same task. Preserve useful reasoning tokens and end with the task's normal final-answer line. Do not quote or describe the context.

Task:
{dataset task prompt}

Structured feedback context:
[Context block 1]
{verifier result}
[Context block 2]
{parser record}
[Context block 3]
{verifier provenance}
[Context block 4]
{response-format diagnostics}

Prompt Template — WinoGrande

Solve the following task.

Reason step by step about the options. Then, on a new final line, write Answer: followed by the single option letter only (for example Answer: B). Write nothing after that line.

Choose the option that best fills the blank in the sentence. Resolve the reference using commonsense reasoning.

Sentence:
{sentence}

Choices:
A. {option 1}
B. {option 2}

Teacher feedback context:

You are the feedback-conditioned teacher in on-policy self-distillation. The earlier student attempt will appear as the assistant continuation. Use the structured feedback context to revise the continuation distribution for the same task. Preserve useful reasoning tokens and end with the task's normal final-answer line. Do not quote or describe the context.

Task:
{dataset task prompt}

Structured feedback context:
[Context block 1]
{verifier result}
[Context block 2]
{parser record}
[Context block 3]
{verifier provenance}
[Context block 4]
{response-format diagnostics}

Prompt Template — WiC

```
Solve the following task.

Reason step by step about the options. Then, on a new final line,
write Answer: followed by the single option letter only (for
example Answer: B). Write nothing after that line.

Consider the word '{word}' in the two sentences below. Determine
whether it has the same meaning in both sentences.

Sentence 1:
{sentence 1}

Sentence 2:
{sentence 2}

Choices:
A. Different meaning
B. Same meaning
```

Teacher feedback context:

```
You are the feedback-conditioned teacher in on-policy
self-distillation. The earlier student attempt will appear as the
assistant continuation. Use the structured feedback context to
revise the continuation distribution for the same task. Preserve
useful reasoning tokens and end with the task's normal final-answer
line. Do not quote or describe the context.

Task:
{dataset task prompt}

Structured feedback context:
[Context block 1]
{verifier result}
[Context block 2]
{parser record}
[Context block 3]
{verifier provenance}
[Context block 4]
{response-format diagnostics}
```

Prompt Template — MMLU-Pro

```
Solve the following task.

Reason step by step about the options. Then, on a new final line,
write Answer: followed by the single option letter only (for
example Answer: B). Write nothing after that line.

{question}

Choices:
{lettered non-N/A choices}
```

Teacher feedback context:

```
You are the feedback-conditioned teacher in on-policy
self-distillation. The earlier student attempt will appear as the
assistant continuation. Use the structured feedback context to
revise the continuation distribution for the same task. Preserve
useful reasoning tokens and end with the task's normal final-answer
line. Do not quote or describe the context.

Task:
```

```
{dataset task prompt}

Structured feedback context:
[Context block 1]
{verifier result}
[Context block 2]
{parser record}
[Context block 3]
{verifier provenance}
[Context block 4]
{response-format diagnostics}
```

D.2 Feedback Field Construction

For feedback-conditioned teacher supervision, we represent the feedback context as four structured fields, $\mathbf{c} = (b_1, b_2, b_3, b_4)$. Each field captures a distinct aspect of the feedback signal, allowing FIRE to measure its contribution independently through the leave-one-field-out targets in equation 5.

Verifier result (b_1). The verifier field provides the primary outcome-level feedback. It records whether the sampled response is correct, the student's normalized submitted answer, and the accepted or expected final answer. For an incorrect response, it additionally specifies the corrected final-answer line and instructs the teacher to revise the final answer while retaining the task-specific output format. Therefore, this field contains the most direct information about *what* in the final prediction must change.

Parser record (b_2). The parser field describes how the student's final prediction was extracted from its sampled response. It records the task and expected answer types, required final-answer marker, number of occurrences of that marker, whether parsing succeeded, the extracted normalized answer, and a snapshot of the final response line. For numerical tasks, it also records nearby numerical tokens. This field therefore exposes whether an apparent error may be related to answer extraction or the structure of the submitted response.

Verifier provenance (b_3). The provenance field supplies task and verifier-level context used to interpret the verification decision. It includes the dataset adapter and task type, a task fingerprint, available choice labels or numerical cues from the prompt, the checker mode (e.g., exact matching or symbolic equivalence), and the answer-normalization procedure. It additionally summarizes basic response statistics such as character, nonempty-line, and numerical-token counts. This field provides contextual evidence about *how* the correctness decision was produced without changing the verifier outcome itself.

Response-format diagnostics (b_4). The final field captures compliance with the required response format. It records the expected final-answer marker and parsed result, identifies issues such as a missing marker, repeated markers, or a non-canonical final line, and indicates whether reasoning precedes the final answer. For arithmetic responses, it also summarizes detected arithmetic expressions. The field concludes with the task-level requirement that the response terminate with exactly one valid final-answer line and no subsequent text.

Illustrative feedback instance. To make the structured feedback representation more understandable, the following box shows an illustrative example for an incorrect mathematical response on GSM8K.

Example Structured Feedback Context

```
[Context block 1: Verifier result]
Verifier feedback: source=reference_checker; status=incorrect;
decision=final; expected/accepted final answer='42'; submitted
```

```
normalized answer='40'; replace the submitted final answer with
'42'.
```

[Context block 2: Parser record]

```
Parser diagnostics: source=response_parser; task
uid='example_gsm8k_001'; task type=math; expected answer
type=numeric; final marker='####'; marker count=1; parse
success=yes; extracted normalized final answer='40'; final line
snapshot='#### 40'; numeric tokens near parsed span=[40].
```

[Context block 3: Verifier provenance]

```
Context provenance: source=environment_audit; dataset
adapter=gsm8k; task type=math; task fingerprint='example_001';
prompt numeric cues=[6, 7]; checker mode=numeric-equivalence;
normalization=task_adapter_final_answer; submitted normalized
answer='40'; response chars=184; response nonempty lines=5; response
numeric-token count=8.
```

[Context block 4: Response-format diagnostics]

```
Response-format diagnostics: source=format_checker; required final
marker='####'; final-line parse result='40'; format issues=none;
reasoning text before final line=yes; arithmetic expression count=3;
arithmetic expression snippets='6 × 7 = 40'; instruction=end with
exactly one task-normal final-answer line and no text after it.
```

E Comparison with Closely Related Methods

Several recent methods address instability in on-policy distillation, but differ from FIRE in both setting and mechanism. In particular, some methods below study standard teacher-student on-policy distillation, whereas FIRE targets feedback-based *self-distillation*, where the teacher and student are derived from the same model. In this section, we clarify in detail how FIRE differs from existing works.

SRPO and SCOPE. SRPO (Li et al., 2026a) and SCOPE (Zheng et al., 2026) also route correct and incorrect trajectories differently. SRPO operates in a self-distillation setting, sending correct samples to GRPO and failed samples to self-distillation. SCOPE instead uses conventional teacher-student on-policy distillation, with student-perplexity-weighted on-policy SFT for correct trajectories and teacher-perplexity-weighted distillation for incorrect ones. FIRE, by contrast, uses Fisher-informed re-weighted on-policy SFT for correct responses and, for incorrect responses, attributes the induced gradient to individual feedback fields. Neither method performs individual feedback-field attribution or constrains both branches with Fisher and gradient-derived insights.

DemoPSD. DemoPSD (Li et al., 2026b) is also an on-policy self-distillation method and is the closest baseline in terms of target recalibration. It measures teacher-student disagreement using per-token Jensen-Shannon Divergence (JSD) and geometrically shifts the privileged teacher toward the student when disagreement is large. FIRE, by contrast, measures the gradient induced by the feedback-conditioned teacher. The full teacher is unchanged when its gradient lies within the local budget, with only unsafe updates triggering feedback-field attribution and, when still necessary, subsequent contraction. Therefore, consideration for individual feedback field influence via leave-one-out counterfactuals is not considered with DemoPSD.

Veto. Veto (Jang et al., 2026) studies conventional teacher-student on-policy distillation rather than self-distillation. It constructs a geometric teacher-student target with a scheduled interpolation coefficient. FIRE is instead token-specific and determined by the attribution-aware gradient relative to a Fisher-derived radius. FIRE also routes by correctness and attributes unsafe updates to individual feedback fields.

TOP-D and TrOPD. TOP-D (Xie et al., 2026) and TrOPD (Xing et al., 2026) likewise address conventional on-policy distillation with distinct teacher and student models, rather than the feedback-based self-distillation setting. While TOP-D and TrOPD also constrain unreliable or overly large distillation updates, their trust-region mechanisms differ from FIRE's Fisher-informed local gradient radius. TOP-D constructs a proximal teacher and uses internal trust-region iterations, while TrOPD identifies teacher-verifiable regions from teacher-student agreement and treats outliers separately. FIRE instead directly constrains the token-level gradient induced by a feedback-conditioned self-teacher using the student's local Fisher geometry. In addition, neither TOP-D nor TrOPD routes trajectories by correctness into separate correct and incorrect-response objectives.

Key distinction. FIRE combines outcome routing, feedback-field counterfactual attribution in gradient space, and a token-level Fisher-informed update bound. The attribution determines *which feedback-dependent direction* to follow, while the Fisher radius determines *how far* the update may move. Importantly, FIRE leaves the original full-feedback target unchanged whenever its induced update is already within this radius and outperforms all the aforementioned baselines in Sec. 4.

F Additional FIRE Details

This section gives the derivations and implementation details omitted from Sec. 3. All coefficients and target distributions are constructed from detached model outputs. Backpropagation is performed only through the live student distribution $p^t_{\boldsymbol{\theta}}$.

F.1 Execution Order and Feedback Conditioning

At optimizer step k, the self-distillation process inspired by Hübotter et al. (2026) performs the following operations:

1. Sample a prompt $\mathbf{x}$ and generate one response $\mathbf{y} \sim \pi_{\boldsymbol{\theta}_k}(\cdot \mid \mathbf{x})$ using the current student.
2. Construct privileged feedback $\mathbf{c} = \mathcal{F}(\mathbf{x}, \mathbf{y})$ for that response.
3. Score the fixed response with the live student conditioned on $(\mathbf{x}, \mathbf{y}_{<t})$ and with the EMA teacher conditioned on $(\mathbf{x}, \mathbf{c}, \mathbf{y}_{<t})$.
4. Backpropagate the reverse-KL loss and update the student.
5. Update the EMA teacher from the newly updated student.

The teacher therefore supplies feedback-conditioned supervision on samples visited by the student. It neither produces the training trajectory nor receives gradients. FIRE preserves this rollout, student-update, and EMA-update ordering, while feedback construction and EMA-teacher scoring are performed only for incorrect responses after outcome routing.

F.2 Exact Reverse-KL Logit Gradient

Let $p^t_{\boldsymbol{\theta}} = \mathrm{softmax}(\mathbf{z}^t)$ and let q^t be a detached target. The token-level reverse KL is

$$\mathrm{KL}\big(p^t_{\boldsymbol{\theta}} \,\|\, q^t\big) = \sum_v p^t_{\boldsymbol{\theta}}(v)\left[\log p^t_{\boldsymbol{\theta}}(v) - \log q^t(v)\right]. \tag{16}$$

Using

$$\frac{\partial p^t_{\boldsymbol{\theta}}(v)}{\partial z^t(u)} = p^t_{\boldsymbol{\theta}}(v)\big(\mathbf{1}[u = v] - p^t_{\boldsymbol{\theta}}(u)\big), \tag{17}$$

the logit gradient evaluated at the detached student snapshot $p^t(v) = \mathrm{sg}[p^t_{\boldsymbol{\theta}_k}(v)]$ is

$$\big[\mathbf{g}^t(q^t)\big]_v = p^t(v)\left[\log \frac{p^t(v)}{q^t(v)} - \mathbb{E}_{u\sim p^t}\left[\log \frac{p^t(u)}{q^t(u)}\right]\right]. \tag{18}$$

The expectation centers the gradient so that $\sum_v \big[\mathbf{g}^t(q^t)\big]_v = 0$. FIRE compares feedback fields using this gradient because it represents the update the target would actually apply to the current student. Probability changes in regions assigned negligible student mass have little immediate effect, whereas smaller changes on high-mass tokens may strongly affect training.

F.3 Fisher Scale and Learning-Rate Calibration

For a categorical student distribution p^t, the softmax Fisher matrix is

$$\mathbf{F}(p^t) = \mathrm{diag}(p^t) - p^t(p^t)^\top, \tag{19}$$

whose trace is

$$\phi^t = \mathrm{tr}\,\mathbf{F}(p^t) = 1 - \|p^t\|_2^2. \tag{20}$$

For an on-policy token $y_t \sim p^t$, the hard-label logit gradient is $p^t - \mathbf{e}_{y_t}$, and, using $\mathbb{E}_{y_t\sim p^t}[p^t(y_t)] = \|p^t\|_2^2$,

$$\begin{aligned}\mathbb{E}_{y_t\sim p^t}\|p^t - \mathbf{e}_{y_t}\|_2^2 &= \mathbb{E}_{y_t\sim p^t}\left[1 + \|p^t\|_2^2 - 2p^t(y_t)\right] \\ &= 1 + \|p^t\|_2^2 - 2\|p^t\|_2^2 = 1 - \|p^t\|_2^2 = \phi^t.\end{aligned} \tag{21}$$

Therefore, $\sqrt{\phi^t}$ is the root-mean-square on-policy hard-label gradient norm. FIRE defines

$$s_k = \max\left(1, \frac{a_k}{a_0}\right), \qquad \rho_k^t = \frac{\sqrt{\phi^t}}{s_k}. \tag{22}$$

When the scheduled learning rate is at or below the hyperparameter nominal rate value, FIRE retains the Fisher scale. When $a_k > a_0$, the radius contracts inversely with the learning-rate increase. The maximum prevents a decaying scheduler from enlarging the permitted gradient.

F.4 Correct-Branch Re-weighting as Gradient Projection

For a verified token, define

$$\mathbf{g}_+^t = p^t - \mathbf{e}_{y_t}, \qquad \beta^t = \min\left(1, \frac{\rho_k^t}{\|\mathbf{g}_+^t\|_2}\right). \tag{23}$$

Because β^t is detached from the computation graph,

$$\nabla_{\mathbf{z}^t}\ell_+^t = \beta^t\big(p^t - \mathbf{e}_{y_t}\big). \tag{24}$$

Consequently,

$$\left\|\nabla_{\mathbf{z}^t}\ell_+^t\right\|_2 = \min\big(\|\mathbf{g}_+^t\|_2, \rho_k^t\big)\,. \tag{25}$$

Below the radius, FIRE is identical to ordinary on-policy SFT. Above it, the gradient lands exactly on the boundary without changing its direction.

We note the exact activation condition for the correct branch. Since $\|\mathbf{g}_+^t\|_2^2 = 1 + \|p^t\|_2^2 - 2p^t(y_t)$, clipping at the hyperparameter ($s_k = 1$, $(\rho_k^t)^2 = \phi^t$) occurs precisely when

$$p^t(y_t) < \|p^t\|_2^2 = \mathbb{E}_{u\sim p^t}\left[p^t(u)\right]. \tag{26}$$

Clipping is therefore applied exactly to verified tokens that the current student assigned below-average probability, i.e. to the tokens carrying the largest hard-label gradient in an already correct trace. This is intended: the radius equalizes the per-token step scale between the two branches, so that a single surprising token inside a correct response cannot dominate the update relative to the feedback-conditioned branch. The cost is that the most informative correct tokens are the ones rescaled, which trades per-token learning speed on the correct branch for comparability of step size across branches.

F.5 Exact Leave-One-Field-Out Influence

For field b_j, define the full-versus-removed log-probability residual

$$r_j^t(v) = \log q_{-j}^t(v) - \log q_{\mathbf{c}}^t(v). \tag{27}$$

Subtracting equation 18 for the two teacher views gives

$$\left[\mathbf{g}_{\mathbf{c}}^t - \mathbf{g}_{-j}^t\right](v) = p^t(v)\left[r_j^t(v) - \mathbb{E}_{u\sim p^t}\left[r_j^t(u)\right]\right]. \tag{28}$$

Therefore,

$$E_j^t = \left\|\mathbf{g}_{\mathbf{c}}^t - \mathbf{g}_{-j}^t\right\|_2^2 \tag{29}$$

is the exact change in the local training signal caused by removing field b_j. FIRE multiplies this influence by

$$\chi^t = \begin{cases} \dfrac{\left[E_{\mathbf{c}}^t - (\rho_k^t)^2\right]_+}{E_{\mathbf{c}}^t}, & E_{\mathbf{c}}^t > 0, \\ 0, & E_{\mathbf{c}}^t = 0, \end{cases} \tag{30}$$

where $E_{\mathbf{c}}^t = \|\mathbf{g}_{\mathbf{c}}^t\|_2^2$. This agrees with equation 10 in every case, and $\chi^t \in [0, 1)$ since $(\rho_k^t)^2 > 0$ whenever p^t is non-deterministic. The numerator is exactly the full teacher's gradient energy outside the permitted energy ball. If the complete feedback-conditioned target is already inside the ball, all field weights vanish and the full teacher is retained.

With

$$Z^t = (\rho_k^t)^2 + \sum_j w_j^t, \qquad \alpha_0^t = \frac{(\rho_k^t)^2}{Z^t}, \qquad \alpha_j^t = \frac{w_j^t}{Z^t}, \tag{31}$$

adopting the convention $\alpha_0^t = 1$, $\alpha_j^t = 0$ in the degenerate case $Z^t = 0$, the attribution target is the normalized geometric mean

$$q_A^t(v) = \frac{q_{\mathbf{c}}^t(v)^{\alpha_0^t} \prod_{j=1}^B q_{-j}^t(v)^{\alpha_j^t}}{\sum_{u \in \mathcal{V}} q_{\mathbf{c}}^t(u)^{\alpha_0^t} \prod_{j=1}^B q_{-j}^t(u)^{\alpha_j^t}}. \tag{32}$$

Its log probability satisfies

$$\log q_A^t(v) = C^t + \alpha_0^t \log q_{\mathbf{c}}^t(v) + \sum_j \alpha_j^t \log q_{-j}^t(v), \tag{33}$$

where C^t is constant across vocabulary elements. Since equation 18 is affine in $\log q$ after centering, and since $\alpha_0^t + \sum_j \alpha_j^t = 1$,

$$\mathbf{g}^t(q_A^t) = \alpha_0^t \mathbf{g}_{\mathbf{c}}^t + \sum_j \alpha_j^t \mathbf{g}_{-j}^t. \tag{34}$$

The geometric mean therefore converts the field-attribution coefficients into an exact weighted average of the corresponding training signals. An arithmetic probability mixture would not preserve this identity.

Two consequences of equation 34 arise. Firstly, writing $\boldsymbol{\delta}_j^t = \mathbf{g}_{\mathbf{c}}^t - \mathbf{g}_{-j}^t$, the attribution step can be expressed as

$$\mathbf{g}^t(q_A^t) = \mathbf{g}_{\mathbf{c}}^t - \frac{\chi^t}{Z^t} \sum_j \|\boldsymbol{\delta}_j^t\|_2^2 \, \boldsymbol{\delta}_j^t. \tag{35}$$

Second, because equation 34 is a convex combination,

$$\|\mathbf{g}^t(q_A^t)\|_2 \le \max\left(\|\mathbf{g}_{\mathbf{c}}^t\|_2, \max_j \|\mathbf{g}_{-j}^t\|_2\right), \tag{36}$$

but $\|\mathbf{g}^t(q_A^t)\|_2 > \|\mathbf{g}_{\mathbf{c}}^t\|_2$ is possible when some $\boldsymbol{\delta}_j^t$ is anti-aligned with $\mathbf{g}_{\mathbf{c}}^t$. Attribution is therefore a reweighting of the correction direction and carries no norm guarantee on its own. The radius is enforced by the subsequent projection.

F.6 Incorrect-Branch Radial Projection

Let

$$\mathbf{g}_A^t = \mathbf{g}^t(q_A^t), \qquad \eta^t = \min\left(1, \frac{\rho_k^t}{\|\mathbf{g}_A^t\|_2}\right), \tag{37}$$

with $\eta^t = 1$ for a zero gradient. FIRE constructs

$$q_{\mathrm{FIRE}}^t(v) = \frac{p^t(v)^{1-\eta^t} q_A^t(v)^{\eta^t}}{\sum_{u \in \mathcal{V}} p^t(u)^{1-\eta^t} q_A^t(u)^{\eta^t}}. \tag{38}$$

Up to a vocabulary-independent normalizer C_η^t,

$$\log q_{\mathrm{FIRE}}^t = (1 - \eta^t) \log p^t + \eta^t \log q_A^t + C_\eta^t, \tag{39}$$

so the log-ratio appearing in equation 18 reduces to

$$\log p^t - \log q^t_{\mathrm{FIRE}} = \eta^t \left(\log p^t - \log q^t_A\right) - C^t_\eta. \tag{40}$$

The constant C^t_η cancels under the centering in equation 18, and the remaining factor η^t passes through linearly, yielding

$$\mathbf{g}^t(q^t_{\mathrm{FIRE}}) = \eta^t \mathbf{g}^t_A. \tag{41}$$

Therefore,

$$\left\|\mathbf{g}^t(q^t_{\mathrm{FIRE}})\right\|_2 = \min\left(\|\mathbf{g}^t_A\|_2, \rho^t_k\right). \tag{42}$$

Attribution changes which feedback-dependent direction is followed, and radial projection then restricts how strongly that direction is applied. The two stages are sequential rather than independent, since η^t is determined by $\|\mathbf{g}^t_A\|_2$ and therefore by the outcome of the attribution step.

F.7 Unified Bound and Its Scope

Equations 25 and 42 imply that either routed branch satisfies

$$\|\mathbf{g}^t_{\mathrm{branch}}\|_2 \le \rho^t_k = \frac{\sqrt{\phi^t}}{\max(1, a_k/a_0)}. \tag{43}$$

When $a_k \ge a_0$,

$$a_k \|\mathbf{g}^t_{\mathrm{branch}}\|_2 \le a_0 \sqrt{\phi^t}. \tag{44}$$

This is a local output-logit bound. It does not directly bound the full parameter update, next-token KL, or global convergence, which additionally depend on the model Jacobian, optimizer preconditioning and momentum, gradient accumulation, and parameter-gradient clipping.

F.8 Computation and Memory

Correct responses require one student rollout and one student scoring pass, but no teacher evaluation. Incorrect responses use the same rollout, one live student scoring pass, one full-feedback EMA-teacher score, and B leave-one-field-out teacher scores. None of the teacher passes generates a new response.

The leave-one-field-out views can be streamed. Define

$$S^t_w = \sum_j w^t_j, \qquad R^t_w = \sum_j w^t_j \left(\log q^t_{-j} - \log q^t_{\mathbf{c}}\right). \tag{45}$$

Then, since $1 - S^t_w/Z^t = (\rho^t_k)^2/Z^t$, the attribution target can be computed as

$$\log q^t_A = \log \mathrm{softmax}\left(\log q^t_{\mathbf{c}} + \frac{R^t_w}{(\rho^t_k)^2 + S^t_w}\right). \tag{46}$$

The implementation therefore stores one residual accumulator rather than all B removed-field distributions simultaneously. FIRE introduces no additional student rollout, student forward pass, or backward pass beyond the underlying self-distillation loop.

Two numerical caveats apply. First, the exact gradient identities assume full-vocabulary reverse KL, while truncating the vocabulary support makes them approximate. Second, because equation 32 is a geometric mean, $\log q^t_A(v)$ diverges whenever any teacher view with $\alpha^t_j > 0$ assigns vanishing mass to v, so q^t_A effectively inherits the narrowest support among the active views. In practice all teacher log-probabilities are floored at a small constant before accumulation into R^t_w, which bounds the residual and leaves the identities in Appendices F.5 and F.6 accurate to that floor.