CREDITFILTER: CALIBRATED TOKEN WEIGHTS OVER THE WITHIN-RESPONSE ENTROPY RANK FOR RLVR

Anonymous authors
Paper under double-blind review

ABSTRACT

Critic-free reinforcement learning with verifiable rewards (RLVR) typically assigns a uniform advantage to every token. Token-reweighting heuristics like the 80/20 mask favor high-entropy tokens, but fail to measure a token's actual impact on the outcome. We formalize this impact as the forking value: the variance of the success probability over the policy's next-token choices. Because the expected forking values along a sequence sum exactly to the reward variance, token credit acts as a budget each response spends at its own forks; consequently, weights should redistribute credit strictly within a response. Mathematically, the forking value factorizes into a fork consequence, which requires costly continuations to estimate, and a fork probability, which is intrinsically provided by the softmax. On Qwen3-8B-Base, this fork probability decays exponentially with a token's within-response entropy rank at class-specific rates, revealing that most tokens carry negligible forking value. CreditFilter weights each token using this exponential envelope with no extra forward passes, stabilized by difficulty damping and an entropy-gated fallback. CreditFilter outperforms GRPO and DAPO by 4–5 points on MATH500 and 8–10 on the AIME sets, and its dynamically adaptive variant matches it. The ablations reveal that the majority of this gain stems from ranking tokens within each response rather than globally across the batch, the coordinate on which the 80/20 mask ranks.

1 INTRODUCTION

Credit assignment is the central open problem of reinforcement learning with verifiable rewards (RLVR) (Lambert et al., 2025; Guo et al., 2025): a scalar reward per response has to be spread over thousands of tokens. Per-token advantages can come from a learned value function (Schulman et al., 2016; 2017), from rollouts or perturbations at every prefix (Kazemnejad et al., 2025; Li et al., 2026), or from a process reward model (Lightman et al., 2024; Cui et al., 2026). However, each of these approaches costs a critic or extra generations. In practice, they have largely given way to the critic-free GRPO family (Shao et al., 2024; Yu et al., 2025), which assigns a uniform advantage to every token in a response.

Token-importance weighting keeps this shared advantage and rescales it token by token; the 80/20 mask of Wang et al. (2025), for example, keeps only the 20% highest-entropy tokens of a batch. The rescaling coefficient is non-negative, so every token preserves the sign of its response's overall advantage. The underlying heuristic is that the tokens capable of changing the course of the reasoning, the *forking* tokens (Bigelow et al., 2025), deserve more weight, with entropy serving as a proxy for this forking potential. With the current rescaling approaches, two questions remain open. **Which tokens represent valuable forks, and how to weight them?** Existing rules that avoid extra forward passes rely purely on proxies; none directly measure how much altering a token shifts the probability of a correct answer. So neither the shape of the weighting function nor the cutoff threshold is grounded in empirical measurement. **Ranked against which tokens?** Such masks rank tokens across the entire batch, so a long, uncertain and incorrect response retains most of its tokens while a confident, correct response loses all of them.

We answer the first question through empirical measurement and the second through a theoretical identity, supported by an ablation that compares ranking tokens within each response versus across the batch (Section 5.4). The foundation of our approach is the *forking value* ν: the variance of the

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

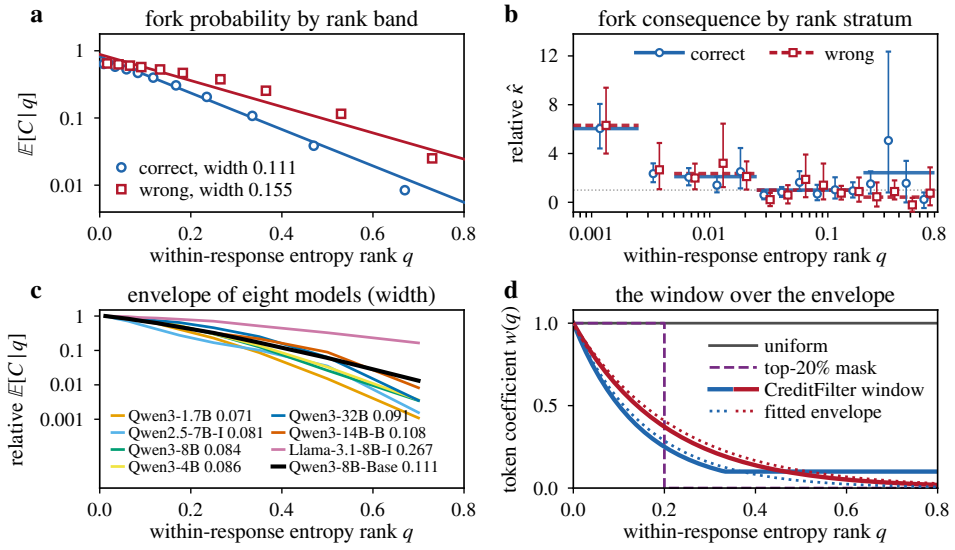


Figure 1: The forking spectrum of Qwen3-8B-Base and the window built from it. (a) Fork probability C in bands of the within-response entropy rank, with the exponential fits (widths 0.111 for correct and 0.155 for wrong responses). (b) Fork consequence $\hat{\kappa}$ in finer rank strata (log rank axis), relative to its class's level over ranks 2.5–20%; error bars: 95% intervals from a bootstrap over prompts; segments: values pooled over four rank ranges. (c) The envelope on eight models, each normalized to its top band; the legend gives each model's width fitted to both classes together, and for Qwen3-8B-Base the width of its correct class. (d) Uniform weights, the within-response top-20% mask and the CreditFilter windows (widths 0.10/0.14, floor 0.1 on correct responses) against the fitted envelopes of (a) (dotted).

probability of ending with a correct answer across the policy's next-token alternatives. The sum of the expected forking values along a response exactly equals the reward variance $V(1-V)$, where V is the prompt's overall success probability. Because this total variance is fixed by the prompt, credit is effectively a budget that each response spends at its own forks. A token weight should therefore redistribute credit strictly *within* the response; ranking tokens globally across a batch inappropriately moves credit *between* responses.

To determine exactly how to allocate this budget, we factorize the forking value into $\nu = C\kappa$. The *fork consequence* κ requires expensive continuations to measure (Section 5.1), but the *fork probability*, the chance that two independent draws of the next token differ, is intrinsically yielded by the softmax. On Qwen3-8B-Base (Yang et al., 2025), we find that this fork probability decays exponentially against the token's entropy rank within its response, exhibiting separate widths for correct and wrong responses (Figure 1a). We denote this profile the *envelope*.

CreditFilter converts this envelope into a soft window (Figure 1d) by scaling each token's advantage by $w_c(q) = \max(2^{-q/\lambda_c}, \varepsilon_c)$. This applies an exponential decay based on the token's within-response entropy rank q, parameterized by the envelope width λ_c of its respective class c (correct or wrong). By enforcing a minimum floor ε_c solely on correct responses, the method equalizes the mean weight across both classes. This window design suppresses the low-entropy tail, ensuring that tokens with a vanishing fork probability receive negligible credit. Our contributions can be summarized as follows:

- **Measurement.** We prove that the expected forking values along a response sum to the reward variance, factorize the forking value as $\nu = C\kappa$, and introduce an unbiased branching estimator to measure it. Empirically, on Qwen3-8B-Base we show that the *forking spectrum* forms an exponential envelope over the within-response entropy rank, with the fork consequence heavily concentrated in the top 0.25% of tokens; the envelope recurs on seven other models.

- **Method.** We prove that the theoretically optimal sign-preserving token weight targets the expected forking value. CreditFilter approximates this target without extra forward passes by weighting tokens using the exponential envelope of the fork probability—the component intrinsically yielded by the softmax. To stabilize training, the method incorporates difficulty damping and an entropy-gated fallback to uniform credit, alongside an adaptive variant (CreditFilter-D) that dynamically tracks the envelope.
- **Results.** On Qwen3-8B-Base, CreditFilter outperforms GRPO and DAPO by 4–5 points on MATH500 and 8–10 points on AIME24 and AIME25. In our reproductions, it exceeds the 80/20 mask by 13–15 points and beats STARE and SC-GRPO on all three sets. Ablations reveal that ranking tokens strictly *within* each response drives the vast majority of this gain. Finally, we demonstrate that our windowed update shifts the policy up to $2.3\times$ further in KL divergence than a uniform update, without increasing parameter displacement.

2 Related work

Per-token credit in RLVR. Value functions with generalized advantage estimation (Schulman et al., 2016; 2017), Monte-Carlo values from extra rollouts (VinePPO (Kazemnejad et al., 2025)), counterfactual token perturbations (OAR-P (Li et al., 2026)) and process reward models (Lightman et al., 2024; Cui et al., 2026) give each token its own advantage at the price of a critic, extra generations or extra forward passes. The critic-free GRPO family (Shao et al., 2024; Yu et al., 2025; Guo et al., 2025) keeps one scalar per response, and under shared prefixes the group advantage already acts as a Monte-Carlo process reward (Sullivan & Koller, 2026). CreditFilter stays in this family and moves the scalar within the response.

Token reweighting and entropy control. A second line of work rescales the shared advantage token-by-token using various proxies for the fork (Appendix G), including token entropy (Wang et al., 2025; Tan et al., 2026), covariance (Wang et al., 2026), gradient saliency (Li et al., 2026), mutual information (He et al., 2026), and self-conditioned KL (Shan et al., 2026). While CreditFilter adopts the entropy ordering of the 80/20 mask (Wang et al., 2025), it diverges by ranking tokens strictly within each response rather than across the batch, calibrates its envelope on measured fork probabilities, and needs no extra forward pass. Finally, rather than controlling entropy through asymmetric clipping (Yu et al., 2025; Park et al., 2025), covariance targeting (Cui et al., 2025), or target-entropy loops (Yu et al., 2026; Luo et al., 2026), CreditFilter stabilizes training via a simple batch-entropy-gated fallback to uniform credit.

Measuring forking tokens. Bigelow et al. (2025) identify forking tokens, and Lin et al. (2025) locate critical tokens within incorrect solutions, by resampling continuations for individual responses. In contrast, a rank-based weighting scheme requires the forking value as a generalized function of the token's rank. Our approach estimates this value as the variance across the policy's next-token alternatives, averaged by rank band (Section 5.1).

3 Method

Setting. We use DAPO as the base training recipe (Appendix B). For each prompt, the policy samples G responses. The response i receives one group-normalized advantage A_i from its final reward, the verifier outcome $R_i \in \{0, 1\}$ less an overlong penalty. The policy maximizes the batch token mean of the clipped surrogate $\tilde{A}_{i,t}\,\rho_{i,t}$, with $\rho_{i,t}$ the importance ratio. GRPO and DAPO use $\tilde{A}_{i,t} = A_i$. The position t of the response i has prefix $s_{i,t}$, the next-token law $\pi_{i,t}$ and entropy $H_{i,t}$ in nats. Its *within-response entropy rank* is

$$q_{i,t} = r_{i,t}/T_i \in (0, 1], \tag{1}$$

where $r_{i,t}$ is the descending rank of $H_{i,t}$ among the T_i tokens of the same response, so $q \approx 0$ marks the most uncertain positions of the response. The class $c(i) \in \{+, -\}$ of a response is defined as the sign of its group-normalized advantage A_i. This typically aligns with the verifier outcome (correct or wrong), except that a correct response is assigned to the negative class if the overlong penalty drops its reward below the group mean.

3.1 FORMULATION OF THE WINDOW

CreditFilter rescales the advantage of every token by a non-negative coefficient,

$$\tilde{A}_{i,t} = w_{i,t} A_i, \qquad w_{i,t} = g_i \Big[(1 - m_{\text{gov}}) + m_{\text{gov}}\, w_{c(i)}(q_{i,t}) \Big], \qquad w_c(q) = \max\big(2^{-q/\lambda_c},\, \varepsilon_c\big). \tag{2}$$

Here the kernel w_c decays exponentially with the rank, bounded by a per-class floor ε_c. Its width λ_c defines the rank interval over which the weight halves. The g_i factor acts as a difficulty damper, while m_{gov} serves as an entropy-gated fallback to uniform credit (Section 3.2); setting $g_i = m_{\text{gov}} = 1$ would give the unmodified window. Figure 1d visualizes this window over the measured envelope, and Algorithm 1 outlines the computation for one batch.

Shape and width from the envelope. The width parameter for each class is derived from the class envelope, the mean fork probability $\mathbb{E}[C \mid q, c]$ where $C = 1 - \sum_v \pi_v^2$. We fit it with a single exponential in the rank:

$$\lambda_c \approx \lambda_{C,c}, \qquad \text{the width of the class envelope profile } \mathbb{E}[C \mid q, c], \tag{3}$$

where this log-linear fit is performed across ten rank bands covering $q \leq 0.8$ (Appendix D.7). CreditFilter sets $\lambda_+ = 0.10$ and $\lambda_- = 0.14$, representing the rounded envelope widths derived from a subset of base-policy responses. The full measurement of Section 5.1 yields 0.111 and 0.155 (Table 9). The fork probability is the mathematical component of the forking value that is naturally provided by the softmax. The forking value itself is:

$$\nu_t = \mathrm{Var}_{x \sim \pi(\cdot \mid s_t)}\, Q(s_t, x), \tag{4}$$

which represents the variance of the success probability Q over the next token. As shown in Proposition 4, this factors into $\nu = C\kappa$, where the *fork consequence* κ requires evaluating continuations. Finally, Section 4 establishes the *forking spectrum* $\mathbb{E}[\nu \mid q, c]$ as the target of a non-negative coefficient (Proposition 3).

Class mass and the floor. While gradient clipping is inactive, only the shape within a class and the ratio of the two class masses $M_c = \mathbb{E}_q[w_c(q)]$ reach the Adam update (Lemma 8). The mass M has a closed form in terms of (λ, ε) (Proposition 7). CreditFilter equalizes the class masses by introducing a positive-class floor:

$$M(\lambda_+, \varepsilon_+) = M(\lambda_-, 0), \tag{5}$$

which has a unique solution. Given the widths $0.10/0.14$, setting the floor to $\varepsilon_+ = 0.1$ (the solution 0.106, rounded) brings the class masses to 0.196 and 0.200. This floor ensures a small positive update on the low-entropy tokens of correct responses, while the zero floor of the wrong class concentrates its negative gradient exclusively on the forking positions.

Why the within-response rank. Appendix D.5 gives the three reasons: every response of a given length receives the same mean coefficient, so the budget of Proposition 1 is redistributed within each response; the rank is invariant to monotone transforms of the uncertainty measure; and on the rank the exponential is the memoryless kernel, so a single width sets the concentration.

3.2 DAMPING AND FALLBACK

Difficulty damping scales correct responses by $g_i = (1 - \hat{p}_i)^\gamma$, where $\gamma = 1$ and $\hat{p}_i$ is the pass rate of the response's group, and leaves wrong responses undamped. In expectation over a prompt's responses, the damped update preserves the direction of the undamped one and shrinks it by a factor between $3/4$ and 1 (Proposition 9). Because g_i shrinks as $\hat{p}_i$ grows, the update on prompts the policy mostly solves rests mainly on the wrong responses. As a safeguard, an *entropy-gated fallback* reverts the update toward uniform credit as the batch mean entropy $\bar{H}$ rises,

$$m_{\text{gov}} = \mathrm{clip}\Big(\frac{H_{\text{hi}} - \bar{H}}{H_{\text{hi}} - H_{\text{lo}}},\, 0,\, 1\Big), \qquad H_{\text{lo}}/H_{\text{hi}} = 1.5/3.0 \text{ nats}. \tag{6}$$

Above H_{hi}, the window is entirely disabled in favor of uniform credit, though damping and clipping are retained. Over the CreditFilter training run detailed in Section 5, the window remains fully active in 64% of the steps and fully replaced by uniform credit in 8% (Appendix C).

3.3 RE-MEASURING THE WIDTH ONLINE (CREDITFILTER-D)

The calibration equation 3 is a property of the current policy. Because the envelope moves during training (Section 5.1), relying on widths fixed to the base policy creates a mismatch over the later stage of the training. Because C is derived from the same softmax as the entropies, every batch intrinsically yields a zero-cost estimate $\hat{\lambda}_{C,c,t}$ of each class width. CreditFilter-D tracks this using an exponential moving average:

$$\lambda_{c,t} = (1-\alpha)\,\lambda_{c,t-1} + \alpha\,\hat{\lambda}_{C,c,t}, \qquad \alpha = 0.1, \tag{7}$$

where the batch estimate and the average are clamped to a class-specific range, with the average initialized at the measured base widths $0.111/0.155$ (Appendix D.7). With moving widths, a fixed floor can no longer equalize the classes, so CreditFilter-D drops the floor and, at every step, rescales the window of the class with the smaller natural mass $M(\lambda_{c,t}, 0)$ up to match the larger mass. This equalizes the masses while preserving the exact shape of each window.

4 THEORY: THE TARGET OF A SIGN-PRESERVING TOKEN WEIGHT

Three theoretical results establish the optimal target for a sign-preserving token weight and explain why our windowing mechanism approximates it. Proposition 1 characterizes the expected total of the forking value across a response. Proposition 3 demonstrates that the conditional expectation of this forking value, given a local observable, serves as the ideal target for a non-negative token weight. Proposition 4 factors this value into two components: the fork probability, which the softmax gives for free and which bounds the forking value, and the fork consequence that needs to be estimated with continuations. The respective proofs are included in Appendix D.

Setting. Consider on-policy rollouts for a fixed prompt. At position t with prefix s_t, let $V_t = \mathbb{E}[R \mid s_t]$ denote the value, $Q(s_t, x)$ the action value of the next token x, and $\delta_t = Q(s_t, x_t) - V_t$ the credit increment of the sampled token. The conditional variance of this increment is the forking value equation 4. Our rescaling assigns token t of response i an adjusted advantage $w(o_{i,t})A_i$, where the non-negative coefficient $w \geq 0$ depends on an observable o of the token. When computing expectations over positions, we apply position weights, which are uniform across all tokens under a token-mean loss. For brevity, we define $N(o) = \mathbb{E}[\delta_t A \mid o]$, $D(o) = \mathbb{E}[A^2 \mid o]$ and $m(o) = \mathbb{E}[\nu_t \mid o]$.

Proposition 1 (credit decomposition)**.** *With increments padded by zero after termination,*

$$\textstyle\sum_t \delta_t = R - V_1, \qquad \mathbb{E}[\delta_t \mid s_t] = 0, \qquad \mathbb{E}[\delta_t \delta_u] = 0 \;\; (t \neq u), \qquad \mathbb{E}[\delta_t^2 \mid s_t] = \nu_t, \tag{8}$$

and $\sum_t \mathbb{E}[\nu_t] = \mathrm{Var}(R) = V_1(1-V_1)$*. Thus the reward variance equals the expected sum of the forking values along a response.*

The identity represents the orthogonal-increment decomposition of the bounded martingale V_t, and it provides the theoretical justification for ranking tokens within each response (Section 3.1).

Assumption 2 (rank look-ahead)**.** *Let* $\Gamma_t = (\#\{u > t : H_u > H_t\},\, T)$*, with ties broken by position (earlier first) and* T *the number of ranked tokens, so that the within-response rank* q_t *is a function of* (s_t, Γ_t)*. For every position t,* $\mathbb{E}[\delta_t \mid s_t, \Gamma_t, R] = \mathbb{E}[\delta_t \mid s_t, R]$*: given the prefix and the outcome, the number of later tokens more uncertain than the current one and the response length carry no further information about the current increment.*

Proposition 3 (positive-part MMSE coefficient and its target)**.** *Among* $w \geq 0$*, the mean-square error* $\mathbb{E}[(\delta_t - w(o)A)^2]$ *is minimized on* $\{D > 0\}$ *by* $w^*(o) = \max\{0,\, N(o)/D(o)\}$*. Let* R *be the verifier outcome,* $c = R$*,* $A = (R - V_1)/d$ *with* d *a constant of the prompt, and let the position weights be functions of* (s_t, Γ_t, R) *(token-mean and rollout-equal weights both qualify). Under Assumption 2, for a fixed prompt and* $o = (q_t, c)$*,*

$$N(o) = \frac{1}{d}\,\mathbb{E}\big[(1-V_1)\,\nu_t/V_t \,\big|\, o\big] \text{ (correct)}, \qquad N(o) = \frac{1}{d}\,\mathbb{E}\big[V_1\,\nu_t/(1-V_t) \,\big|\, o\big] \text{ (wrong)}, \tag{9}$$

and D *is constant within each class. Hence* w^* *is proportional to this value-weighted forking spectrum within each class, and to the forking spectrum* $\mathbb{E}[\nu_t \mid o]$ *when the* ν*-weighted mean of* $1/V_t$ *(correct) or* $1/(1-V_t)$ *(wrong) at rank* q *does not depend on* q*. For the group advantage of* G *rollouts the identities hold with the prompt-level scales* $\mathbb{E}[A \mid \mathrm{prompt}, R]$ *in place of* $(1-V_1)/d$ *and* $-V_1/d$ *(Appendix D.2).*

Proof sketch. The positive part minimizes the quadratic $Dw^2 - 2Nw$ over $w \geq 0$. Since δ_t is determined by s_{t+1} and $V_{t+1} = V_t + \delta_t$, $\mathbb{E}[\delta_t R \mid s_t] = \mathbb{E}[\delta_t V_{t+1} \mid s_t] = \mathbb{E}[\delta_t^2 \mid s_t] = \nu_t$, and with $\mathbb{E}[\delta_t \mid s_t] = 0$ this gives $\mathbb{E}[\delta_t \mid s_t, R = 1] = \nu_t/V_t$ and $\mathbb{E}[\delta_t \mid s_t, R = 0] = -\nu_t/(1 - V_t)$. Assumption 2 extends both to conditioning on (s_t, Γ_t, R), of which o and the position weights are functions, so the weights leave the conditional law of δ_t unchanged. On a correct response $A = (1 - V_1)/d$ is fixed by the prompt, which gives the first identity; on a wrong one $A = -V_1/d$ and the two signs cancel. D equals $(1-V_1)^2/d^2$ or V_1^2/d^2 on the two classes. The group-advantage case and the strata with $m(o) = 0$ are treated in Appendix D.2.

The target of the window is therefore the forking spectrum $\mathbb{E}[\nu_t \mid o]$ up to the value factor, and our window follows this profile over the rank. The criterion also determines the class masses: for a fixed prompt with equal position weights and $N \geq 0$, the per-token class mean of w^* evaluates to $d/\mathbb{E}[T \mid c]$, so classes of equal mean length receive equal masses, which is the property enforced by the class-mass equalization in Section 3.1. Our window purposely omits the value factor.

Proposition 4 (factorization and envelope)**.** *At a position with next-token distribution π and action values $Q_k \in [0, 1]$,*

$$\nu = \tfrac{1}{2} \sum_{k,l} \pi_k \pi_l (Q_k - Q_l)^2 = C\,\kappa, \qquad C = 1 - \sum_k \pi_k^2, \qquad \kappa = \tfrac{1}{2}\,\mathbb{E}\big[(Q_K - Q_L)^2 \,\big|\, K \neq L\big], \tag{10}$$

with $K, L \sim \pi$ independent, $\kappa \in [0, \frac{1}{2}]$ and $\kappa := 0$ where $C = 0$. With H the entropy in nats,

$$\nu \;\leq\; C/2 \;\leq\; 1 - \pi_{\max} \;\leq\; C \;\leq\; H. \tag{11}$$

κ represents half the mean squared value gap between two independent draws from π. The upper bound $\nu \leq C/2$ suppresses the low-C tail of the forking spectrum: a position with a vanishing fork probability carries zero fork value, regardless of its consequence. The value-weighted target naturally inherits the bound, scaled by the value factor. Across the remainder of the rank, weighting by the envelope represents a deliberate modeling choice. Suppose the mean fork consequence, conditioned on both the rank and the fork probability, is constant within a class (condition K). Then the forking spectrum would be proportional to the class envelope $\mathbb{E}[C \mid q, c]$, and, under the same condition on the value-weighted consequence, so would the target (Corollary 6). Section 5.1 measures the degree to which condition K holds and how closely the envelope matches the forking spectrum.

5 Experiments

Setup. All runs train Qwen3-8B-Base on the DAPO-17k prompts (BytedTsinghua-SIA, 2025) for 400 steps with 128 prompts and 16 responses per step, mini-batch 32 and learning rate 10^{-6}. Responses run up to 20,480 tokens, with dynamic sampling, the overlong penalty and clip ratios 0.2 below and 0.27 above (0.28 above for DAPO, symmetric 0.2 for GRPO; every setting in Appendix B). We reproduce the 80/20 mask of Wang et al. (2025) on its DAPO recipe as published, STARE (Luo et al., 2026) and SC-GRPO (Shan et al., 2026) with their published settings (Appendix C). We report mean@32 (temperature 1.0, top-p 0.7) on MATH500 (Hendrycks et al., 2021; HuggingFaceH4, 2024) ($n = 500$), AIME24 and AIME25 (Mathematical Association of America, 2024; 2025) ($n = 30$ each) averaged over checkpoints 350/375/400 (the late average) and at step 400.

5.1 The forking spectrum of Qwen3-8B-Base

At a sampled position of a base-policy response, an instrument branches on several next tokens, continues each branch using the training sampler, and evaluates the continuations. It estimates ν as the variance of the branch success probabilities under the policy, unbiased over the next tokens it branches on; Appendix E gives the two instruments, the protocol and the full tables.

The fork probability decays exponentially in the rank; the consequence peaks in the top 0.25% (Figure 1a, b). A single exponential fits the band means of C with widths 0.111 (correct) and 0.155 (wrong). On the four-branch instrument (Table 11), half of the correct class's forking value is concentrated in the top 10% of ranks, with the top 20% capturing 68% of it (63% for wrong

Table 1: mean@32 after training Qwen3-8B-Base, on MATH500 (n=500), AIME24 and AIME25 (n=30 each): the average over checkpoints 350/375/400 (late), the value at step 400, and the share of step-400 samples left unanswered at the 20,480-token cap. Bold: best accuracy in each column; the base-model row is step 0.

	MATH500		AIME24		AIME25		unanswered (%)		
Run	late	400	late	400	late	400	M500	A24	A25
CreditFilter	0.876	0.882	0.338	0.352	**0.277**	**0.283**	0.3	4.0	0.8
CreditFilter-D	**0.887**	**0.889**	**0.367**	**0.372**	0.268	0.276	0.7	5.0	1.9
GRPO	0.833	0.848	0.263	0.281	0.184	0.173	1.0	2.4	4.5
DAPO	0.830	0.837	0.244	0.259	0.177	0.195	0.9	4.6	1.5
80/20 mask	0.741	0.731	0.193	0.177	0.149	0.144	5.8	6.6	5.6
SC-GRPO	0.838	0.836	0.269	0.283	0.168	0.156	1.3	1.1	4.8
STARE	0.824	0.819	0.189	0.183	0.156	0.155	0.4	1.3	1.1
base model	0.507		0.055		0.047		10.6	22.5	14.8

responses). The fork consequence is flat over most of the rank and rises only at the head (Table 13). Compared with its mean over ranks 2.5–20%, it is about six times larger in the top 0.25% of the rank, which is about two tokens per response, and about twice as large over ranks 0.5–2.5%; beyond 20% the estimates are too noisy to show a trend. Condition K holds approximately: the consequence is flat over the rank apart from a spike in the top 2.5%, the most uncertain tokens of a response. There the envelope already gives those tokens their largest weight, so the spike would add only a further factor of about three on a handful of tokens (the next paragraph quantifies it). We therefore approximate the consequence as uniform over the rank. The exponential envelope of the fork probability recurs across seven other models, with widths from 0.071–0.108 on the Qwen models to 0.267 on Llama-3.1-8B-Instruct, so the width is calibrated per model (Figure 1c, Table 12).

What the window retains. We score a weight profile by the proportion of matched-filter gain it retains: its squared cosine with the measured forking spectrum over the rank, which is 1 for the spectrum itself. The CreditFilter window retains 0.61 (correct) and 0.64 (wrong), against 0.40 and 0.51 for the within-response top-20% mask and 0.21 and 0.16 for uniform weights (Figure 1d, Table 14). Applying a threefold weight explicitly to the top 0.25% would raise both shares to 0.79, a gain the window leaves out (Appendix E.4).

The envelope moves during training. On rollouts of the CreditFilter checkpoints sampled at the training length, the wrong-class width doubles to approximately 0.34 from step 250 onward (with the exception of 350). Conversely, the correct-class width halves to about 0.05 from step 300 onward (Table 8). Despite this shifting envelope, the fixed window maintains a squared cosine of at least 0.83 with the moving envelope (Appendix D.7).

5.2 Main results

CreditFilter leads every baseline on every set, and its adaptive variant matches it. Across the late checkpoints (Table 1) CreditFilter leads GRPO by 4, 8 and 9 points on MATH500, AIME24 and AIME25, respectively, and surpasses DAPO by 5, 9 and 10 points. It leads the 80/20 mask by 13–15 points, and exceeds STARE and SC-GRPO by 4–15 points. Furthermore, the dynamically tracked CreditFilter-D scores higher than CreditFilter on MATH500 and performs comparably on the AIME sets.

5.3 Entropy dynamics and local displacement

The windowed update moves the policy further without moving the parameters further. The batch entropy of the nine runs plotted in Figure 4 collapses to about 0.1 nats within 25 steps. CreditFilter then trains stably between 0.7 and 2.2 nats for two hundred steps, experiencing only a brief spike to 4.4 nats around step 290, which resolves by step 307. To quantify how successive updates affect the model (the *local displacement*), we evaluate three checkpoints from the CreditFilter run: step 250 (inside the fallback band), step 275 (at the onset of the entropy spike), and step 300 (during its descent). From each checkpoint, we compute two five-step continuations: one retaining the CreditFilter window, and one using uniform weights (with damping and clipping preserved). For

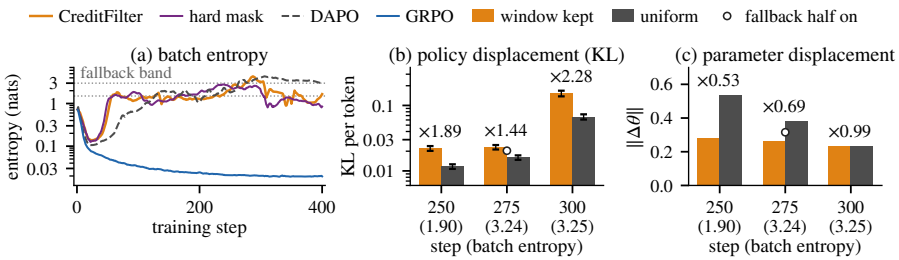


Figure 2: Training dynamics and local displacement. (a) Batch entropy (five-step moving average; dotted: the fallback band, 1.5–3 nats): GRPO collapses and DAPO stays above 3 nats after step 263. (b, c) Five-step continuations from three checkpoints of the CreditFilter run, each labeled with its step and batch entropy. Keeping the window moves the policy 1.4–2.3 times further in KL than uniform weights (b), and moves the parameters, $\|\Delta\theta\| = \|\theta_{t+5} - \theta_t\|_2$, no further (c). Numbers: window/uniform ratio; error bars: 95% prompt-bootstrap intervals (all intervals in Table 18); open circle: fallback half on ($m_{\text{gov}} \approx 0.5$).

Table 2: Ablations: each row changes one component of CreditFilter; the DAPO and 80/20 mask rows repeat Table 1, and both 80/20 rows run on the mask's DAPO recipe. Columns as in Table 1; the late columns add the difference in points from the row's reference, marked by the row shading (reference rows darker).

	MATH500		AIME24		AIME25		unanswered (%)		
Run	late	400	late	400	late	400	M500	A24	A25
CreditFilter	0.876	0.882	0.338	0.352	0.277	0.283	0.3	4.0	0.8
DAPO	0.830	0.837	0.244	0.259	0.177	0.195	0.9	4.6	1.5
80/20 mask	0.741	0.731	0.193	0.177	0.149	0.144	5.8	6.6	5.6
Window shape									
top 20% window	0.854 −2.2	0.858	0.380 +4.2	0.400	0.280 +0.3	0.274	3.3	10.2	13.4
wide window (0.20/0.28)	0.722 −15.4	0.750	0.218 −12.0	0.235	0.190 −8.7	0.197	8.8	9.5	8.6
Coordinate									
random rank	0.849 −2.7	0.849	0.255 −8.3	0.248	0.217 −6.0	0.220	0.9	5.2	2.6
reversed rank	0.694 −18.2	0.703	0.176 −16.2	0.171	0.119 −15.8	0.119	5.4	11.1	5.7
batch-level rank	0.826 −5.0	0.833	0.264 −7.4	0.294	0.194 −8.3	0.211	3.1	6.4	2.6
80/20 mask per-response	0.825 +8.4	0.834	0.322 +12.9	0.314	0.263 +11.4	0.241	5.7	25.5	20.8
Class mass									
no floor	0.862 −1.4	0.863	0.338 +0.0	0.333	0.254 −2.3	0.256	1.1	4.0	4.2
rescaled masses	0.840 −3.6	0.844	0.331 −0.7	0.328	0.243 −3.4	0.260	2.1	5.6	2.6
Stabilizers									
no fallback	0.842 −3.4	0.852	0.327 −1.1	0.315	0.233 −4.4	0.233	2.6	4.7	3.8
no damping	0.792 −8.4	0.765	0.304 −3.4	0.278	0.228 −4.9	0.220	11.4	15.8	13.6
damping only	0.713 −11.7	0.741	0.168 −7.6	0.190	0.138 −3.9	0.136	3.2	3.2	2.0

each we measure the token-level KL from the starting policy, evaluated on the starting policy's own rollouts, and the parameter displacement $\|\theta_{t+5} - \theta_t\|_2$ (Appendix H). At every checkpoint the windowed update, with a fifth of the weight mass, moves the parameters no further than the uniform one. Yet it shifts the policy 1.4–2.3 times further in KL (Figure 2b,c; Table 18). By contrast, at the step-300 checkpoint of the DAPO control run, the window/uniform KL ratio is 0.86 (Appendix H).

5.4 Ablations

Window and damping are complementary. Against DAPO, the window alone (the run without damping) loses 4 points on MATH500 and gains 6 on AIME24 (Table 2). Damping alone loses 4–12 points, and the two together gain 5–10. We read the window as moving a correct response's update onto its forking tokens, off its execution tokens (the low-entropy tokens that carry out steps already decided), including those that close the reasoning and box the answer. The pressure to terminate then weakens: reasoning grows longer, which pays off on AIME, and 11–16% of step-400 samples reach the 20,480-token cap unanswered. Damping shrinks the correct-class update on training prompts the policy mostly solves, so the few wrong responses carry the update on those

prompts; most MATH500 problems are of this kind. Their negative credit at the forks plausibly curbs digressions: the unanswered share falls to 0–4%, and the AIME gains remain.

The gain needs the entropy order and the within-response rank. Reversing the rank collapses the entropy to 0.01 nats, and a random within-response rank lands only 1–4 points above DAPO. Ranking across the batch is level with DAPO and 2 points below even the random rank on MATH500. We read the reversed rank as reinforcing the most confident tokens until the policy stops exploring. A random rank keeps the per-response budget of Proposition 1 but reaches the forks only by chance, so the entropy order supplies most of the gain. A batch-level rank breaks that budget: long uncertain responses, mostly wrong, keep most of their tokens, and concise correct ones lose theirs. With its cut taken within each response, the 80/20 mask on its own recipe gains 8–13 points.

The exact shape matters less than the width. The window retains $0.61/0.64$ of the matched-filter gain and the top 20% window $0.40/0.51$ (Table 14), yet their accuracies differ by at most 4 points. Doubling the widths to $0.20/0.28$ from the first step retains $0.57/0.48$ and costs 9–15 points. Training thus seems to respond mainly to where the mass sits, and both shapes put it at the head of the rank. The top 20% window's zero tail drops the small update on the closing tokens of correct responses, so 3–13% of its samples reach the cap unanswered. Early on, doubled widths spread credit into middle ranks outside the base policy's envelope ($0.11/0.16$), onto execution tokens and noise.

The floor, width tracking and the fallback each add a small gain. Dropping the floor costs 1–2 points; rescaling the class masses at fixed widths costs 4 on MATH500. The floor and the rescaling both equalize the class masses. The floor adds a small update in the tail of correct responses; the rescaling raises their head weights about 1.4-fold, which enlarges the correct-class update as removing the damping does. Adding the tracking to the rescaled run gives CreditFilter-D, 5 and 4 points higher on MATH500 and AIME24, plausibly by following the moving envelope. Without the fallback, the late accuracy is 3 and 4 points lower on MATH500 and AIME25. We hypothesize that high batch entropy marks human-like hesitation, whose high-entropy tokens are often interchangeable wording, so the rank stops finding forks; the fallback moves the update toward uniform credit in that regime, and the run without it keeps crediting noise.

6 Conclusion

Token credit in critic-free RLVR is a budget that each response allocates at its forks: the expected forking values along a response sum to the reward variance. The fork probability, which bounds the forking value and is naturally yielded by the softmax, decays exponentially in a token's within-response entropy rank at a class-specific rate. CreditFilter translates this envelope into a token weight, gaining 4–5 points on MATH500 and 8–10 points on the AIME sets over GRPO and DAPO with no extra forward passes. Ranking within each response drives most of its lead over our reproduction of the 80/20 mask: taking the mask's cut strictly within each response recovers 8–13 of the 13–15 points.

Limitations. Our primary empirical evidence relies on a single base model and a single prompt set, utilizing 30-problem AIME sets and a 20,480-token cap. Response length on the base model currently includes text generated after the answer: from Qwen3-8B-Base, CreditFilter's responses are 1.8–3.1 times longer than GRPO's and DAPO's (Table 15); from a post-trained Qwen3-1.7B, with widths measured on that model, they are 14–40% shorter than DAPO's (Table 16). The rank enters the theory through Assumption 2 and the class through a value factor the window ignores, yet neither is measured.

Reproducibility statement

All runs use released weights (Qwen3-8B-Base, and Qwen3-1.7B in Appendix F) with their original tokenizer and chat template and a fixed prompt set (Appendix B). Every accuracy in the paper is recomputed from the released per-sample evaluation files by the scripts in the supplementary material; the branching instrument, its per-position records and the analysis scripts that produce every figure and table are released with the paper.

AI USE STATEMENT

Large language model assistants (Claude Fable 5 and Fable 5.1) were used throughout this work: to write and debug the experiment, analysis and plotting code, to draft and edit the text under the authors' direction, and to produce simulated reviews of the draft. The authors designed the experiments, verified every number against the recorded per-sample outputs, and take full responsibility for the content.

REFERENCES

Eric Bigelow, Ari Holtzman, Hidenori Tanaka, and Tomer Ullman. Forking paths in neural text generation. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2412.07961.

BytedTsinghua-SIA. DAPO-Math-17k. Hugging Face dataset `https://huggingface.co/datasets/BytedTsinghua-SIA/DAPO-Math-17k`, 2025. Training prompts released with the DAPO paper (Yu et al., 2025).

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, et al. Process reinforcement through implicit rewards. *Transactions on Machine Learning Research*, 2026. arXiv:2502.01456.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645:633–638, 2025. arXiv:2501.12948.

Yinhan He, Yaochen Zhu, Mingjia Shi, Wendy Zheng, Lin Su, Xiaoqing Wang, Qi Guo, and Jundong Li. IAPO: Information-aware policy optimization for token-efficient reasoning. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026. arXiv:2602.19049.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2021. arXiv:2103.03874.

HuggingFaceH4. MATH-500. Hugging Face dataset `https://huggingface.co/datasets/HuggingFaceH4/MATH-500`, 2024. The 500-problem subset of the MATH test set (Hendrycks et al., 2021) selected by Lightman et al. (2024).

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. VinePPO: Refining credit assignment in RL training of LLMs. In *International Conference on Machine Learning (ICML)*, 2025. arXiv:2410.01679.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. arXiv:1412.6980.

Hynek Kydlíček. Math-Verify: Math verification library. `https://github.com/huggingface/Math-Verify`, 2025.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tülu 3: Pushing frontiers in open language model post-training. In *Conference on Language Modeling (COLM)*, 2025. arXiv:2411.15124.

Thanh-Long V. Le, Myeongho Jeon, Kim Vu, Viet Lai, and Eunho Yang. No prompt left behind: Exploiting zero-variance prompts in LLM reinforcement learning via entropy-guided advantage shaping. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2509.21880.

Ziheng Li, Liu Kang, Feng Xiao, Luxi Xing, Qingyi Si, Zhuoran Li, Weikang Gong, Deqing Yang, Yanghua Xiao, and Hongcheng Guo. Outcome-grounded advantage reshaping for fine-grained credit assignment in mathematical reasoning. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2026. ACL Anthology 2026.acl-long.1132; arXiv:2601.07408.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2305.20050.

Zicheng Lin, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xing Wang, Ruilin Luo, Chufan Shi, Siheng Li, Yujiu Yang, and Zhaopeng Tu. Critical tokens matter: Token-level contrastive estimation enhances LLM's reasoning capability. In *International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 37906–37918, 2025. arXiv:2411.19943.

Haipeng Luo, Qingfeng Sun, Songli Wu, Can Xu, Wenfeng Deng, Han Hu, and Yansong Tang. STARE: Surprisal-guided token-level advantage reweighting for policy entropy stability. *arXiv preprint arXiv:2606.19236*, 2026.

Mathematical Association of America. AIME 2024. American Invitational Mathematics Examination I and II, 2024; Hugging Face dataset `https://huggingface.co/datasets/Maxwell-Jia/AIME_2024` (30 problems), 2024.

Mathematical Association of America. AIME 2025. American Invitational Mathematics Examination I and II, 2025; Hugging Face dataset `https://huggingface.co/datasets/yentinglin/aime_2025` (30 problems), 2025.

Jaesung R. Park, Junsu Kim, Gyeongman Kim, Jinyoung Jo, Sean Choi, Jaewoong Cho, and Ernest K. Ryu. Clip-low increases entropy and clip-high decreases entropy in reinforcement learning of large language models. *arXiv preprint arXiv:2509.26114*, 2025.

John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations (ICLR)*, 2016. arXiv:1506.02438.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Yingyu Shan, Yuhang Guo, Zihao Cheng, Zeming Liu, Xiangrong Zhu, Xinyi Wang, Jiashu Yao, Wei Lin, Hongru Wang, and Heyan Huang. Learning from own solutions: Self-conditioned credit assignment for reinforcement learning with verifiable rewards. *arXiv preprint arXiv:2606.18810*, 2026.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Michael Sullivan and Alexander Koller. GRPO is secretly a process reward model. In *International Conference on Machine Learning (ICML)*, 2026. arXiv:2509.21154.

Hongze Tan, Zihan Wang, Jianfei Pan, Jinghao Lin, Hao Wang, Yifan Wu, Tao Chen, Zhihang Zheng, Zhihao Tang, and Haihua Yang. GTPO and GRPO-S: Token and sequence-level reward shaping with policy entropy. In *International Conference on Machine Learning (ICML)*, 2026. arXiv:2508.04349.

Cheng Wang, Qin Liu, Wenxuan Zhou, and Muhao Chen. Taming extreme tokens: Covariance-aware GRPO with Gaussian-kernel advantage reweighting. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 540–546, 2026. ACL Anthology 2026.acl-short.45; arXiv:2605.11538.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for LLM reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2506.01939.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, et al. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025. arXiv:2503.14476.

Song Yu, Li Li, Wenwen Zhao, and Zhisheng Yang. ERPO: Token-level entropy-regulated policy optimization for large reasoning models. *arXiv preprint arXiv:2603.28204*, 2026.

A THE CREDITFILTER COEFFICIENT STEP BY STEP

Algorithm 1 lists the per-batch computation of the coefficient equation 2; the lines marked $\star$ are the CreditFilter-D additions (tracked widths and the class-mass equalization by rescaling, which replaces the floor). Every quantity it uses is already computed by the trainer: the entropies come from the actor's forward pass and the per-token fork probabilities $C_{i,t}$ come from the same softmax, the advantages and group pass rates come from the reward, and the rank is a sort within each response.

Notation and terms. Table 3 collects the symbols and coined terms of the main text; a few symbols that a reproduced method or an appendix derivation reuses with a local meaning are flagged where they occur.

Algorithm 1 CreditFilter coefficient for one training batch, computed from the rollout entropies without gradient (CreditFilter-D lines marked $\star$)

Require: responses $i = 1, \dots, B$ with tokens $t \le T_i$ (format tokens excluded), entropies $H_{i,t}$ and fork probabilities $C_{i,t} = 1 - \sum_v \pi_{i,t}(v)^2$ from the same softmax, advantages A_i, group pass rates $\hat{p}_i$; widths $\lambda_\pm$ and floors $\varepsilon_\pm$ ($\star$: initial widths $0.111/0.155$, clamps $[\lambda_\pm^{\min}, \lambda_\pm^{\max}]$, floors 0); fallback band $(H_{\mathrm{lo}}, H_{\mathrm{hi}})$; damping exponent γ

1: $c(i) \leftarrow \mathrm{sign}(A_i)$; $q_{i,t} \leftarrow \mathrm{rank}_\downarrow(H_{i,t}$ within $i)/T_i$, ties broken deterministically; $q'_{i,t} \leftarrow q_{i,t} - 1/(2T_i)$ ▷ midpoint rank

2: $\star$ for each class c with at least 64 responses: average C per response in the ten q'-bands of Appendix D.7 ($q' \le 0.8$), then over responses; if every band mean is present and positive and the least-squares slope $\hat{\beta}$ of their logarithms on the band centres is negative: $\hat{\lambda}_{C,c} \leftarrow \mathrm{clip}(-\ln 2/\hat{\beta};\ \lambda_c^{\min}, \lambda_c^{\max})$, $\lambda_c \leftarrow (1-\alpha)\lambda_c + \alpha\hat{\lambda}_{C,c}$ equation 7

3: $k_{i,t} \leftarrow \max(2^{-q_{i,t}/\lambda_{c(i)}}, \varepsilon_{c(i)})$ ▷ kernel $w_{c(i)}(q_{i,t})$ of equation 2

4: $\star$ with c the class of smaller natural mass $M(\lambda, 0) = \lambda(1 - 2^{-1/\lambda})/\ln 2$ and $\bar{c}$ the other class: $k_{i,t} \leftarrow k_{i,t}\, M(\lambda_{\bar{c}}, 0)/M(\lambda_c, 0)$ for all i with $c(i) = c$

5: $\bar{H} \leftarrow$ token-mean entropy of the retained batch (after dynamic sampling); $m_{\mathrm{gov}} \leftarrow \mathrm{clip}\big((H_{\mathrm{hi}} - \bar{H})/(H_{\mathrm{hi}} - H_{\mathrm{lo}}), 0, 1\big)$

6: $g_i \leftarrow (1 - \hat{p}_i)^\gamma$ if $A_i > 0$ else 1

7: $w_{i,t} \leftarrow g_i\,[(1 - m_{\mathrm{gov}}) + m_{\mathrm{gov}} k_{i,t}]$ ▷ coefficient of equation 2

8: maximize the batch token mean of $\min\{\rho_{i,t} w_{i,t} A_i,\ \mathrm{clip}(\rho_{i,t}, 0.8, 1.27)\, w_{i,t} A_i\}$, with the dual clip of Table 4 for $A_i < 0$

Table 3: Notation and terms.

symbol or term	meaning
R_i, A_i, $\tilde{A}_{i,t} = w_{i,t}A_i$	verifier outcome of response i, its group-normalized advantage, and the advantage rescaled by the coefficient $w_{i,t}$
$c(i) = \text{sign}(A_i)$	class of a response, correct or wrong
$H_{i,t}$, $C_{i,t} = 1 - \sum_v \pi_v^2$	entropy and fork probability of token t, from the same softmax
$q_{i,t}$, q' (midpoint rank)	entropy rank of a token within its response, $1/T_i$ (near 0) at the highest entropy; the window uses $q = r_t/T_i$, the width fits use the midpoint rank $q' = (r_t - \frac{1}{2})/T_i$ (Appendix D)
$w_c(q) = \max(2^{-q/\lambda_c}, \varepsilon_c)$, $k_{i,t}$	the kernel, with λ_c the width (rank interval over which the weight halves) and ε_c the floor
$M(\lambda, \varepsilon)$	class mass, the mean of the kernel over the rank (Proposition 7)
$g_i = (1 - \hat{p}_i)^\gamma$, $\hat{p}_i$	difficulty damping of correct responses; the pass rate of the response's group
m_{gov}, $\bar{H}$, $[H_{lo}, H_{hi}]$	fallback mixing weight (1: window on, 0: uniform credit), batch entropy, and the entropy band that drives it (1.5/3.0 nats)
ν_t, C_t, $\kappa_t = \nu_t/C_t$	forking value, fork probability and fork consequence of a position (Section 4)
$Q(s, x)$, V	action value and value, the probability of ending correct
envelope, $\lambda_{C,c}$; λ_ν	the profile $\mathbb{E}[C \mid q, c]$ and the width of its exponential fit; the width fitted to the forking-value profile
condition K	the mean fork consequence given the rank and C is constant within a class, so the forking spectrum is proportional to the class envelope (Corollary 6)
J	continuations per branch (8 four-branch, 48 full-support)
$m(o)$	forking spectrum $\mathbb{E}[\nu_t \mid o]$, the mean-square target for prefix-measurable o with D constant (Appendix D)
δ_t; $N(o)$, $D(o)$, w^*	credit increment $Q(s_t, x_t) - V_t$ of the sampled token; $\mathbb{E}[\delta_t A \mid o]$, $\mathbb{E}[A^2 \mid o]$ and the mean-square coefficient $\max\{0, N/D\}$ (Proposition 3)
value-weighted spectrum, value factor	$\mathbb{E}[(1 - V_1)\,\nu_t/V_t \mid q, c]$ (correct) and $\mathbb{E}[V_1\,\nu_t/(1 - V_t) \mid q, c]$ (wrong), d times the target for the rank and the class under Assumption 2 (Proposition 5); the weight on ν_t is the value factor, which the window ignores
four-branch, full-support instrument	the sampled token and its three most probable alternatives; the head branches up to mass 0.9 (at most six) plus two tail draws (Appendix E)
matched-filter gain retained	squared cosine between a weight profile and the measured forking-value profile over the rank strata (Table 14)
format tokens	the end-of-text, end-of-turn and thinking-tag tokens, excluded from the rank and from the measurement

B TRAINING AND EVALUATION PROTOCOL

This appendix gives the training recipe shared by all runs (Table 4), the choice of the fallback band, the evaluation protocol and the statistics.

Choice of the fallback band. The upper bound of the fallback band, 3.0 nats, was set from pilot runs: runs whose batch entropy stayed above 3 nats for long stretches ended with lower accuracy.

Evaluation protocol. Every checkpoint is exported to HF format and sampled with vLLM 0.18.0 (bfloat16, tensor parallel 1, one replica per GPU, seed 0 plus the GPU index, context 21,504 tokens) with $k = 32$ samples per problem at temperature 1.0, top-p 0.7 and at most 20,480 new tokens, on MATH500 ($n = 500$), AIME24 ($n = 30$) and AIME25 ($n = 30$). The last boxed expression is extracted, normalized as in DAPO and graded with Math-Verify (Kydlíček, 2025) against the reference (the training reward applies exact match to the same normalized string). A sample is complete once it has produced a boxed answer; a sample that reaches the 20,480-token cap without one is unanswered and counts as wrong, and the tables report the share of such samples. The rule grades the last expression the model boxed, so generation that continues after an answer is not

Table 4: Settings shared by all runs; the runs differ only in the entries of Tables 5 and 6.

Model	Qwen3-8B-Base with its official tokenizer and ChatML chat template
Prompts	DAPO-17k (BytedTsinghua-SIA, 2025), boxed-answer variant (17,917 problems); prompts of at most 1,024 tokens; data seed 201
Batch	128 prompts $\times$ 16 rollouts per step; mini-batch 32 prompts, i.e. four optimizer updates per step
Schedule	400 steps; checkpoint every 25 steps
Optimizer	AdamW, learning rate 10^{-6} constant (no warm-up), $\beta = (0.9, 0.999)$, weight decay 0.01, gradient clipping at norm 1, one optimizer pass per mini-batch; no KL loss and no entropy bonus; token-mean loss aggregation
Coefficient	the window coefficient is computed from the rollout entropies and the verifier outcomes under no gradient and multiplies the advantage, which is itself a constant of the update
Rollouts	vLLM, temperature 1.0, top-p 1.0, no top-k; at most 20,480 response tokens
Reward	training grader (boxed answer, exact match after normalization, 0/1) plus the DAPO soft overlong penalty: with $L_{\max} = 20{,}480$ and buffer $L_{\mathrm{c}} = 4{,}096$, a response of ℓ tokens receives 0 for $\ell \le L_{\max} - L_{\mathrm{c}}$, $(L_{\max} - L_{\mathrm{c}} - \ell)/L_{\mathrm{c}}$ inside the buffer and -1 at the cap (factor 1.0)
Sampling	DAPO dynamic sampling: generation chunks of 128 prompts, only prompt groups with mixed outcomes are kept, at most 8 chunks per step
Advantage	group mean/standard-deviation normalization (DAPO); PPO ratio ρ clipped to $[1 - 0.2,\ 1 + 0.27]$ for the CreditFilter recipe; dual clip: for $A < 0$ the per-token objective is $\max\{\min(\rho A, \mathrm{clip}(\rho)A),\ 3A\}$
Hardware	two GPU types (H100 and B200); the runs differ only in rollout throughput settings and micro-batching, every other parameter being identical, so we treat them as one training configuration

penalized unless it boxes another expression; the overlong penalty of Table 4 shapes training and is not part of the evaluation. All checkpoints are evaluated with this protocol on the same problem sets, and contrasts between two runs are paired by problem.

Statistics. The per-problem accuracy of a checkpoint is the mean over its 32 samples; the late average of a run is the mean over checkpoints 350, 375 and 400 of that per-problem accuracy. Intervals are 95% percentile bootstrap intervals over problems with 3,000 resamples; a contrast between two runs resamples problems and takes the mean of the per-problem difference.

C RUN DEFINITIONS

All runs share the recipe of Table 4 and differ only in the entries of Table 5; Table 6 gives the settings of the reproduced baselines, Table 7 how often the fallback engaged, and the fidelity notes at the end how each reproduction departs from its publication. The masses M_+/M_- in Table 5 follow from the closed form of Proposition 7 (Appendix D.4).

Reproduced baselines: fidelity notes. *80/20 mask.* Both 80/20 runs normalize the loss over all response tokens instead of over the kept ones (a constant factor 0.2 that Adam removes; the gradient norm never reached the clipping threshold). *Absolute levels.* Our GRPO run's late MATH500 / AIME24 / AIME25 (0.833 / 0.263 / 0.184) match the 83.0 / 28.3 / 20.0 reported for GRPO on Qwen3-8B-Base by Le et al. (2026), who sample at the same temperature and top-p but score the best checkpoint with 8 samples per problem under an 8,192-token cap and clip ratios $0.20/0.28$. The DAPO baseline of Wang et al. (2025) on the same model and prompts, trained with 512 prompts per step for about 40 steps (640 gradient steps) and evaluated at its best checkpoint with 16 samples at temperature 1.0 (response cap and top-p not stated), reaches 89.2 / 33.3 / 25.4; CreditFilter-D reaches 0.887 / 0.367 / 0.268 with 128 prompts per step over 400 steps under our protocol (top-p 0.7, 20,480-token cap), and our DAPO run at that budget stays at 0.830 / 0.244 / 0.177. The published 80/20 mask matches its host there ($+0.5$ / $+1.3$ / $+0.8$) and trails it here by 9, 5 and 3 points. *STARE.* The O1 variant with the paper's defaults on the DAPO recipe; surprisals are taken from the rollout log-probabilities shared by the four mini-batches of a step, and the top-P quantile is computed over the batch. Its entropy gate opens only while the batch entropy is below the target

Table 5: Run definitions: CreditFilter, CreditFilter-D and the uniform-weight recipes (upper block) and the ablations of Table 2 (lower block); the reproduced baselines are in Table 6. Kernel: exp $= 2^{-q/\lambda_c}$; hard $= \mathbf{1}[q \leq 0.2]$; dyn. = tracked widths, the tracker fitting the fork-probability profile $\mathbb{E}[C \mid q, c]$ every step; none = uniform weights. Rank: within-response (resp.), batch-level (batch), random within-response (random) or reversed within-response (reversed) entropy rank. Fallback: band H_{lo}/H_{hi} in nats of the entropy-gated fallback ("off": $m_{\text{gov}} \equiv 1$); Damp.: γ_+/γ_-; Clip: upper clip ratio. M_+/M_-: expected class masses from the closed form on the trainer's rank grid ("$\rightarrow$": natural mass, then rescaled).

Run	Kernel	Rank	λ_+/λ_-	$\varepsilon_+/\varepsilon_-$	Fallback	Damp.	Clip	M_+/M_-
CreditFilter	exp	resp.	0.10/0.14	0.1/0	1.5/3.0	1/0	0.27	0.196/0.200
CreditFilter-D[†]	exp, dyn.	resp.	0.111/0.155	0/0	1.5/3.0	1/0	0.27	rescaled each step
DAPO	none	—	—	—	off	0/0	0.28	1/1
GRPO	none	—	—	—	off	0/0	0.20	1/1
hard mask (top 20%)	hard	resp.	—	0/0	1.5/3.0	1/0	0.27	0.200/0.200
wide window	exp	resp.	0.20/0.28	0.1/0	1.5/3.0	1/0	0.27	$0.293 \rightarrow 0.196$ / $0.370 \rightarrow 0.200$
random rank	exp	random	0.10/0.14	0.1/0	1.5/3.0	1/0	0.27	0.196/0.200
reversed rank	exp	reversed	0.10/0.14	0.1/0	1.5/3.0	1/0	0.27	0.196/0.200
batch-level rank	exp	batch	0.10/0.14	0.1/0	1.5/3.0	1/0	0.27	0.196/0.200 under a uniform rank
80/20 mask, per-response cut	hard	resp.	—	0/0	off	0/0	0.28	0.200/0.200
no floor	exp	resp.	0.10/0.14	0/0	1.5/3.0	1/0	0.27	0.144/0.200
rescaled masses	exp	resp.	0.10/0.14	0/0	1.5/3.0	1/0	0.27	$0.144 \rightarrow 0.200$ / 0.200
no fallback	exp	resp.	0.10/0.14	0.1/0	off	1/0	0.27	0.196/0.200
no damping	exp	resp.	0.10/0.14	0.1/0	1.5/3.0	0/0	0.27	0.196/0.200
damping only (window removed)	none	—	—	—	1.5/3.0	1/0	0.27	1/1

[†]Widths tracked from these initial values: clamps $[0.05, 0.30]/[0.07, 0.40]$, EMA $\alpha = 0.1$, ≥ 64 responses per class (Appendix D.7). Wide window: widths doubled; both class windows rescaled to CreditFilter's class masses (shape-only change). Random rank: fresh permutation of each response's positions every step. Reversed rank: lowest-entropy tokens ranked first. Batch-level rank: per-response mass varies. Rescaled masses: the fixed-width form of CreditFilter-D's equalization. Damping only: uniform token weights; the fallback then has no effect.

Table 6: Reproduced baselines: the 80/20 mask, STARE and SC-GRPO of Table 1. Token weight and rank as in Table 5; host recipe with its upper clip ratio (Clip); none of these runs carries the fallback or the damping (fallback off, damping $0/0$). Fidelity notes at the end of this appendix.

Method	Token weight	Rank	Host	Clip	Settings
80/20 mask	hard	batch	DAPO	0.28	published rule: top-20% of the batch; class masses 0.2 in expectation
STARE	surprisal quantiles	batch	DAPO	0.28	O1 variant, $W = 1.1$, $M = 1.0$, $P = 0.10$, $H_{\text{tgt}} = 0.3$ (STARE's symbols)
SC-GRPO	self-conditioned KL	—	GRPO	0.20	$f = D/(D + c)$, $c = \max(P_{75}, 10^{-4})$, $n_{c,\min} = 2$

0.3 nats, which in our run is the collapsed stretch between steps 10 and 39 (Figure 4); elsewhere the run reduces to its DAPO host. *SC-GRPO.* The self-conditioned teacher (the same weights, with a correct rollout of the group in the system prompt) and the full-vocabulary forward KL follow the paper; the "solve-none" branch does not exist under dynamic sampling (all-wrong groups are removed upstream), the math guide template is ours (the paper's is not released), responses beyond 8,192 (teacher) or 12,288 (KL) tokens fall back to unit weight, and the model is the base model rather than the paper's post-trained one, so the conditioning signal is weaker than in the paper (75th percentile of the KL about 0.007 in a short test run).

Table 7: Fallback engagement per run, reconstructed from the batch entropy with the band $1.5/3.0$: mean m_{gov} and the shares of steps (%) with the window fully on ($m_{\mathrm{gov}} = 1$) and fully replaced by uniform credit ($m_{\mathrm{gov}} = 0$). Damping only (uniform weights) is omitted; the wide-window, batch-level-rank and no-floor runs log 352, 371 and 384 steps.

Run	mean m_{gov}	fully on (%)	uniform (%)
CreditFilter	0.84	64	8
CreditFilter-D	0.69	30	10
hard mask (top 20%)	0.84	57	0
wide window (0.20/0.28)	0.65	37	9
random rank	0.60	42	8
reversed rank	1.00	100	0
batch-level rank	0.66	38	16
no floor	0.86	41	0
rescaled masses	0.83	64	3
no fallback	off	—	—
no damping	0.97	77	0

D Proofs and computations for Section 4

This appendix proves the propositions of Section 4 and the further results stated here (Assumption 2, Proposition 5, Corollary 6, Proposition 7, Lemma 8, Proposition 9); Appendix D.5 gives the rationale for the coordinate and the kernel, and Appendix D.7 the width tracker and the envelope along training.

D.1 Setting, sampling law and notation

(S1) A fixed prompt and policy π; P is the on-policy rollout law; the prefix s_t contains all generated tokens and everything the policy conditions on. (S2) $s_{t+1} = s_t \circ x_t$; termination is observable from the prefix, $T \le T_{\max}$ almost surely, and $R \in \{0, 1\}$ is a deterministic function of the completed response. Let $\mathcal{F}_t = \sigma(s_t)$, $V_t = \mathbb{E}_P[R \mid \mathcal{F}_t]$, $Q(s_t, a) = \mathbb{E}_P[R \mid s_t, x_t = a]$, $\delta_t = V_{t+1} - V_t$, $\nu_t = \mathrm{Var}_{a\sim\pi(\cdot|s_t)} Q(s_t, a)$; after termination $V_t = R$ and $\delta_t = \nu_t = 0$. Two advantage conventions: (I) the ideal $A = (R - p)/d$ with $p = V_1$ and fixed $d > 0$; (G) the group form $A_i = k(r_i - \bar{r}_{-i})/d$, $k = (G - 1)/G$, $\bar{r}_{-i}$ the mean of the other $G - 1$ rewards, with $A_i = 0$ on homogeneous groups. A *sampling law* over rollout–position pairs is $\mathbb{E}_\mu[f] = \mathbb{E}_P[\sum_{i,t} a_{i,t} f_{i,t}]/Z$ with $a_{i,t} \ge 0$, zero after termination, $0 < Z < \infty$; it is *predictable* (P) when each $a_{i,t}$ is $\mathcal{F}_{i,t}$-measurable (pooled active-token sampling is; drawing a rollout and then a uniform position is predictable only when the length is fixed, since its position weight is $1/T_i$). With $S_i = r_i - p = \sum_u \delta_{i,u}$: $N(o) = \mathbb{E}[\delta_t A \mid o]$, $D(o) = \mathbb{E}[A^2 \mid o]$, $m(o) = \mathbb{E}[\nu_t \mid o]$, the selection correction $\Delta_1(o) = \mathbb{E}[\delta_t^2 - \nu_t \mid o]$, the cross-position term $\Delta_2(o) = \mathbb{E}[\delta_t(S_i - \delta_t) \mid o]$ and the peer term $\Lambda(o) = \mathbb{E}[\delta_t(\bar{r}_{-i} - p) \mid o]$. Two rank coordinates are used: the rank $q_t = r_t/T$ (window and masses), with r_t the descending entropy rank within the response and deterministic tie-breaking (Assumption 2), and the *midpoint rank* $q'_t = (r_t - \frac{1}{2})/T$, on which the instrument and the width tracker of CreditFilter-D bin their profiles; both use all response entropies and are retrospective (prefix-measurable only in degenerate cases such as $T = 1$), whereas absolute entropy H_t and a causal rank among positions $\le t$ are prefix-measurable; the outcome class is retrospective as well (prefix-measurable only when $V_1 \in \{0, 1\}$).

D.2 Credit decomposition, MMSE and identification (Propositions 1, 3 and 5)

Decomposition and orthogonality. V_t is a bounded Doob martingale: $\mathbb{E}_P[V_{t+1} \mid \mathcal{F}_t] = V_t$, so $\mathbb{E}_P[\delta_t \mid \mathcal{F}_t] = 0$; given $\mathcal{F}_t$ the next token follows π and $V_{t+1} = Q(s_t, x_t)$, so $\mathbb{E}_P[\delta_t^2 \mid \mathcal{F}_t] = \nu_t$. For $t < u$, δ_t is $\mathcal{F}_u$-measurable and $\mathbb{E}_P[\delta_t \delta_u] = \mathbb{E}_P[\delta_t \, \mathbb{E}_P[\delta_u \mid \mathcal{F}_u]] = 0$. Telescoping, $\sum_{t \le n} \delta_t = V_{n+1} - V_1$, equal to $R - V_1$ for $n \ge T_{\max}$; squaring and using orthogonality, $\sum_t \mathbb{E}_P[\nu_t] = \mathrm{Var}_P(R) = p(1-p)$ for the fixed prompt. With almost-surely finite but unbounded T, bounded convergence gives $\mathbb{E}_P[(V_{n+1} - R)^2] \to 0$, so the sums converge in L^2. These statements are about P at fixed indices; retrospective selection can change the expectation measure.

Non-negative coefficient MMSE. For $w \geq 0$ the conditional risk is $\mathbb{E}[\delta_t^2 \mid o] - 2wN(o) + w^2 D(o)$, minimized on $D > 0$ by $w^* = \max\{0, N/D\}$ (unique a.s.); on $D = 0$, $A = 0$ a.s. conditionally and every finite weight is equivalent. Conditional Cauchy–Schwarz gives $N^2 \leq D\,\mathbb{E}[\delta_t^2 \mid o]$, so $w^* A$ has finite second moment. Expanding around w^*, $\mathrm{Risk}(w) - \mathrm{Risk}(w^*) = \mathbb{E}[D(w - w^*)^2] + 2\mathbb{E}[(-N)w\mathbf{1}[N < 0]]$, which reduces to $\mathbb{E}[D(w - w^*)^2]$ when $N \geq 0$. This is coefficient-prediction optimality under the chosen sampling law; the gradient-error bound below relates it to the update.

Clean identification. Under (I), (P) and o $\mathcal{F}_t$-measurable: $\mathbb{E}_P[\delta_t^2 - \nu_t \mid \mathcal{F}_t] = 0$; for $u < t$, δ_u is $\mathcal{F}_t$-measurable and $\mathbb{E}_P[\delta_t \delta_u \mid \mathcal{F}_t] = 0$; for $u > t$, $\mathbb{E}_P[\delta_t \delta_u \mid \mathcal{F}_t] = \mathbb{E}_P[\mathbb{E}_P[\delta_t \delta_u \mid \mathcal{F}_u] \mid \mathcal{F}_t] = 0$. Multiplying each identity by $a_{i,t}\phi(o_{i,t})$ (an $\mathcal{F}_{i,t}$-measurable multiplier for bounded ϕ), summing and dividing by Z transfers them to μ: $\Delta_1(o) = \Delta_2(o) = 0$ and $N(o) = m(o)/d$, hence $w^*(o) = m(o)/(dD(o))$; $m \geq 0$, so the positive part is inactive. Proportionality of w^* to m across observables additionally needs D constant over the compared observables; with a binary reward $D(o)$ depends on o through the outcome class and the pass rate, so the comparison is made within a class (the class masses are set separately, Proposition 7). The within-response rank and the outcome class are retrospective; Proposition 5 extends the identification to them under Assumption 2.

Finite group and random scale. Under (G), $\delta_t A_i = (k/d)[\delta_t^2 + \delta_t(S_i - \delta_t) - \delta_t(\bar{r}_{-i} - p)]$, so $N(o) = (k/d)[m + \Delta_1 + \Delta_2 - \Lambda](o)$ for any observable and sampling law. If rollouts are i.i.d. given the prompt and both $a_{i,t}$ and o depend only on the focal rollout Z_i, then $\mathbb{E}_P[\bar{r}_{-i} - p \mid Z_i] = 0$ and $\Lambda(o) = 0$ even for retrospective o; including $\mathrm{sign}(A_i)$ or peer information in o can make $\Lambda \neq 0$. Under the same hypotheses (i.i.d. rollouts, $a_{i,t}$ and o depending only on the focal rollout), for a common positive symmetric scale $d(\mathbf{r})$ of the group reward vector (e.g. the group standard deviation, with $A_i = 0$ on homogeneous groups; deleting such groups instead changes the sampling law) with $G > 1$ and $0 < p < 1$, symmetry and $\sum_i A_i = 0$ give $\mathbb{E}_P[A_i \mid Z_i] = \theta(r_i - p)$ with a prompt-dependent $\theta > 0$ (independence is needed for $\theta > 0$: if every reward equals one shared Bernoulli variable the group is exchangeable but every advantage vanishes), so $N(o) = \theta[m + \Delta_1 + \Delta_2](o)$ for focal-only observables; θ must be retained when prompts are pooled.

Assumption 2 (rank look-ahead) is stated in Section 4; the proposition below is the general form of the identification in Proposition 3, with the advantage (I) or (G).

Proposition 5 (identification for the rank and the class, general form)**.** *Let R be the binary verifier outcome, $c = R$ and A the advantage (I) or (G) of R, and let the sampling weights be functions of (s_t, Γ_t, R) (pooled active tokens and rollout-equal $1/T$ both qualify). Under Assumption 2, for a fixed prompt and $o = (q_t, c)$,*

$$N(o) = \frac{1}{d}\,\mathbb{E}\Big[(1-V_1)\frac{\nu_t}{V_t}\,\Big|\,o\Big] \text{ on correct responses}, \qquad N(o) = \frac{1}{d}\,\mathbb{E}\Big[V_1\frac{\nu_t}{1-V_t}\,\Big|\,o\Big] \text{ on wrong ones}, \tag{12}$$

and D is constant within each class. Hence w^* is proportional to the value-weighted forking spectrum, and to $\mathbb{E}[\nu_t \mid o]$ when the ν-weighted mean of $1/V_t$ (correct) or $1/(1-V_t)$ (wrong) over the positions at rank q does not depend on q.

Proof of Proposition 5. Since δ_t is $\mathcal{F}_{t+1}$-measurable, $\mathbb{E}_P[\delta_t R \mid \mathcal{F}_t] = \mathbb{E}_P[\delta_t V_{t+1} \mid \mathcal{F}_t] = \mathbb{E}_P[\delta_t(V_t + \delta_t) \mid \mathcal{F}_t] = \nu_t$; with $\mathbb{E}_P[\delta_t \mid \mathcal{F}_t] = 0$ this gives $\mathbb{E}_P[\delta_t \mid s_t, R = 1] = \nu_t/V_t$ and $\mathbb{E}_P[\delta_t \mid s_t, R = 0] = -\nu_t/(1 - V_t)$ on conditioning events of positive probability, and by Assumption 2 the same values for $\mathbb{E}_P[\delta_t \mid s_t, \Gamma_t, R]$. On a correct response $A = (1 - V_1)/d$ is fixed by the prompt; o and the sampling weights are functions of (s_t, Γ_t, R), so the tilt by the weights leaves the conditional law of δ_t given (s_t, Γ_t, R) unchanged, and $N(o) = \mathbb{E}_\mu[(1 - V_1)\,\mathbb{E}_P[\delta_t \mid s_t, \Gamma_t, R = 1] \mid o]/d = \mathbb{E}_\mu[(1 - V_1)\,\nu_t/V_t \mid o]/d$. With $A = -V_1/d$ the same steps give the wrong-class identity, the negative advantage and the negative conditional increment cancelling. D equals $(1 - V_1)^2/d^2$ or V_1^2/d^2 on the two classes. For the group advantage the rewards of the other rollouts are independent of the trajectory given the prompt, so the identities hold with the prompt-level scales $\mathbb{E}[A \mid \mathrm{prompt}, R]$ in place of $(1 - V_1)/d$ and $-V_1/d$. Strata with $m(o) = 0$ have $\nu_t = 0$ conditionally, hence $\delta_t = 0$ μ-a.s. and $N(o) = 0$. The overlong penalty of the training reward (Appendix B), under which A_i depends on the length and $c = \mathrm{sign}(A_i)$ can differ from R, lies outside the proposition.

Gradient-error bound. With $g_t = \nabla \log \pi(x_t \mid s_t)$, $e_t = \delta_t - w_t A$, $\|g_t\| \leq B$, positions padded after termination and L the maximum response length: $\|\sum_t e_t g_t\|^2 \leq (\sum_t e_t^2)(\sum_t \|g_t\|^2) \leq$

$LB^2 \sum_t e_t^2$ pathwise, so $\mathbb{E}\|\sum_t e_t g_t\|^2 \le LB^2 \mathbb{E}[\sum_t e_t^2]$, and by Jensen the squared norm of the expected discrepancy obeys the same bound. Under differentiability, a valid score identity and policy-independent reward, $\nabla \mathbb{E}_P[R] = \mathbb{E}_P[R \sum_t g_t] = \mathbb{E}_P[\sum_t \delta_t g_t]$ because $\mathbb{E}_P[R g_t] = \mathbb{E}_P[Q(s_t, x_t) g_t]$ and $\mathbb{E}_P[V_t g_t] = 0$. Under pooled active-token sampling $\mathbb{E}_P[\sum_t e_t^2] = \mathbb{E}_P[T]\,\mathrm{Risk}(w)$. The exact discrepancy contains $\sum_t \mathbb{E}[e_t^2 \|g_t\|^2] + 2\sum_{t<u} \mathbb{E}[e_t e_u \langle g_t, g_u \rangle]$; coefficient MMSE ignores score magnitudes, directions and cross-position cancellation, minimizes only this upper bound, and does not preserve unbiasedness.

D.3 Factorization and envelope (Proposition 4, Corollary 6)

Factorization. Fix a position, write $\pi_k = \pi(k \mid s_t)$ and $Q_k = Q(s_t, k) \in [0, 1]$, and let K, L be independent draws from π. The variance of a random variable is half the mean squared difference of two independent copies, so $\nu = \mathrm{Var}\, Q_K = \frac{1}{2}\mathbb{E}[(Q_K - Q_L)^2] = \frac{1}{2}\sum_{k,l} \pi_k \pi_l (Q_k - Q_l)^2$. Diagonal terms vanish, hence $\nu = \mathbb{P}(K \neq L) \cdot \frac{1}{2}\mathbb{E}[(Q_K - Q_L)^2 \mid K \neq L] = C\kappa$ with $C = \mathbb{P}(K \neq L) = 1 - \sum_k \pi_k^2$, and $\kappa := 0$ when $C = 0$ (a deterministic policy has $\nu = 0$). Since $(Q_k - Q_l)^2 \in [0, 1]$, $\kappa \in [0, \frac{1}{2}]$. In words, C is the probability that two independent draws of the policy differ (the fork probability) and κ is half the mean squared change in expected correctness caused by such a fork (the fork consequence).

Envelope chain. $\nu = C\kappa \le C/2$. From $\sum_k \pi_k^2 \ge \pi_{\max}^2$, $C \le 1 - \pi_{\max}^2 = (1 - \pi_{\max})(1 + \pi_{\max}) \le 2(1 - \pi_{\max})$. From $\sum_k \pi_k^2 \le \pi_{\max} \sum_k \pi_k = \pi_{\max}$, $1 - \pi_{\max} \le C$. For the entropy, $-\ln$ is convex, so Jensen gives $H = \mathbb{E}[-\ln \pi_K] \ge -\ln \mathbb{E}[\pi_K] = -\ln \sum_k \pi_k^2 = -\ln(1 - C) \ge C$, using $-\ln(1 - x) \ge x$ on $[0, 1)$; the intermediate quantity $-\ln \sum_k \pi_k^2$ is the collision entropy. Equality $1 - \pi_{\max} = C$ holds iff π is uniform on its support, and $C = H$ only for a deterministic law. Two further bounds: $\nu \le V_t(1 - V_t)$ because $Q_K \in [0, 1]$ has mean V_t (Bhatia–Davis), and $\sum_t \mathbb{E}[\nu_t] = p(1 - p)$ (Proposition 1), so the total forking value of a prompt vanishes as its pass rate approaches 0 or 1. All quantities in each identity refer to one and the same action law; a restricted-law ν cannot replace the full-policy ν in the martingale budget.

Corollary 6 (condition K: the envelope window)**.** *If within a class $\mathbb{E}[\kappa \mid q, c, C] = \bar{\kappa}_c > 0$ depends neither on the rank nor on C (condition K), then $\mathbb{E}[\nu \mid q, c] = \bar{\kappa}_c\, \mathbb{E}[C \mid q, c]$. If, within a class, $\mathbb{E}[\kappa_t / V_t \mid q, +, C]$ (correct) and $\mathbb{E}[\kappa_t / (1 - V_t) \mid q, -, C]$ (wrong) depend neither on the rank nor on C (condition K for the value-weighted consequence), then under Assumption 2*

$$w^*(q, c) \;\propto\; \mathbb{E}[C \mid q, c]\,: \tag{13}$$

the window is the class envelope profile.

Condition K and the envelope window. Within a class, $\mathbb{E}[\nu \mid q, c] = \mathbb{E}[C\kappa \mid q, c] = \mathbb{E}[C\, \mathbb{E}[\kappa \mid q, c, C] \mid q, c]$. Under condition K, $\mathbb{E}[\kappa \mid q, c, C] = \bar{\kappa}_c$ (it need only be imposed where $C > 0$), so $\mathbb{E}[\nu \mid q, c] = \bar{\kappa}_c\, \mathbb{E}[C \mid q, c]$; a decay scale is identified only when $\bar{\kappa}_c > 0$, the envelope profile is not identically zero and at least two ranks occur (constant action values give $\kappa = m = w^* = 0$ despite $C > 0$). Under the clean identification of Appendix D.2 (prefix-measurable observable, predictable sampling, constant D), $w^* = m/(dD) \propto \mathbb{E}[C \mid q, c]$; for the rank and the outcome class, Proposition 5 gives the value-weighted numerators $\mathbb{E}[(1 - V_1)\, C_t \kappa_t / V_t \mid q, +]/d$ and $\mathbb{E}[V_1\, C_t \kappa_t / (1 - V_t) \mid q, -]/d$ with D constant within a class, and conditioning on C as above turns them into multiples of $\mathbb{E}[C \mid q, c]$ under condition K for the value-weighted consequence, which condition K does not supply. In both cases the shape within a class is the envelope profile; the criterion fixes the class amplitudes (Section 4) and the class mass rule sets the implementation amplitudes equal. The weaker condition that suffices wherever $\mathbb{E}[C \mid q, c] > 0$ is that the C-weighted band mean $\bar{\kappa}_c(q) = \mathbb{E}[C\kappa \mid q, c] / \mathbb{E}[C \mid q, c]$ be rank-independent; this ratio is what the instrument estimates as $\mathbb{E}[\hat{\nu} \mid q, c] / \mathbb{E}[C \mid q, c]$, and it differs from the unweighted mean of κ when κ and C are correlated within a band. Condition K is substantive: for two independent fair tokens with reward equal to the second token, both positions have $C = \frac{1}{2}$ and $H = \ln 2$ while $\nu_1 = 0$ and $\nu_2 = \frac{1}{4}$, so identical envelopes can carry different forking value and an envelope alone cannot identify it. If $C_{0,c} > 0$, at least two ranks occur and $\mathbb{E}[C \mid q, c] = C_{0,c} 2^{-q/\lambda_{C,c}}$ exactly, the window is the exponential family with $\lambda_c = \lambda_{C,c}$ and no floor; with a fitted exponential, λ_C is the decay scale of the ten-band fit (Appendix D.7). Section 5.1 and Appendix E assess condition K; its value-weighted form is not assessed.

D.4 WINDOW MASS AND THE POSITIVE-CLASS FLOOR (PROPOSITION 7)

Proposition 7 (window mass and the positive-class floor)**.** *Conditional on the length T, the class and the window parameters, a response's mean coefficient does not depend on its entropy values and lies within $[w(0)-w(1)]/T$ of a continuum mass $M(\lambda,\varepsilon)$ with a closed form (below); for $\lambda_- > \lambda_+$ the equal-mass condition equation 5 has a unique solution on the active-floor interval.*

Per-response mass. With ranks $q_t = r_t/T$ the ranks of a response of length T are a permutation of $\{j/T : j = 1,\dots,T\}$, so conditional on T, the class and the window parameters the response's mean coefficient $\frac{1}{T}\sum_t w(q_t) = M_T(\lambda,\varepsilon) = \frac{1}{T}\sum_{j=1}^{T}\max\{2^{-j/(\lambda T)},\varepsilon\}$ is independent of its entropy values. M_T depends on T and converges to the continuum mass $M(\lambda,\varepsilon)=\int_0^1 \max\{2^{-q/\lambda},\varepsilon\}\,dq$ with $0 \le M - M_T \le [w(0)-w(1)]/T$, because the right-endpoint sum of a non-increasing integrand lies below its integral by at most one cell of height $w(0)-w(1)$. The ratio of normalized class mean weights therefore approaches $M(\lambda_+,\varepsilon_+)/M(\lambda_-,\varepsilon_-)$ when both classes are present with sufficiently long responses; total class masses also scale with the class token counts and are not equalized. A batch-wide rank does not have the per-response property (the batch-level rank ablation of Table 2).

Closed form. For $0 < \lambda < \infty$, $0 \le \varepsilon \le 1$ and uniform q on $[0,1]$, let $b = 2^{-1/\lambda}$ and $q^* = -\lambda \log_2 \varepsilon$ (the point where the exponential meets the floor; $q^* = +\infty$ when $\varepsilon = 0$). If $\varepsilon \le b$ the floor is inactive and $M = \int_0^1 2^{-q/\lambda}dq = \frac{\lambda}{\ln 2}(1-b)$; if $b \le \varepsilon \le 1$, $M = \int_0^{q^*} 2^{-q/\lambda}dq + \varepsilon(1-q^*) = \frac{\lambda}{\ln 2}(1-\varepsilon) + \varepsilon(1+\lambda\log_2\varepsilon)$, using $2^{-q^*/\lambda} = \varepsilon$; the two branches agree at $\varepsilon = b$, and the floored expression is not valid outside $[b,1]$. Values: $M(0.10, 0.1) = 0.1966$ ($q^* = 0.332$), $M(0.14, 0) = 0.2005$, $M(0.10, 0) = 0.1441$, so the floor 0.1 of the CreditFilter configuration adds 0.0525 of continuum mass, all of it on $q \ge 0.33$.

Positive-class floor. For fixed λ, $\varepsilon \mapsto M(\lambda,\varepsilon)$ is continuous, constant on $[0,b]$ and strictly increasing on $[b,1]$ ($\partial M/\partial\varepsilon = 1 + \lambda\log_2\varepsilon = 1 - q^*$, zero at $\varepsilon = b$ and positive for $\varepsilon > b$) up to $M(\lambda,1) = 1$, and $\lambda \mapsto M(\lambda,0)$ is strictly increasing. Hence for $\lambda_- > \lambda_+$ the equation $M(\lambda_+,\varepsilon_+) = M(\lambda_-,0)$ has exactly one solution on the active-floor interval $[b_+,1)$ (for $\lambda_- = \lambda_+$ every $\varepsilon \in [0,b_+]$ solves it, so uniqueness comes only from selecting the active-floor interval), and by implicit differentiation the solution increases when λ_+ narrows or λ_- widens. Bisection gives $\varepsilon_+ = 0.106$ for $(0.10, 0.14)$ (target mass 0.2005); the floor 0.10 of the CreditFilter widths equalizes only approximately (continuum mass ratio $0.1966/0.2005 = 0.98$). Within the clamps of CreditFilter-D a floor equalizes the classes only by flattening the window: $\lambda_+ = 0.05$ with $\lambda_- = 0.40$ needs mass 0.475, and even a floor of 0.3 gives only $M(0.05, 0.3) = 0.324$; CreditFilter-D therefore drops the floor and equalizes the classes by rescaling (Section 3.3).

D.5 WHY THE WITHIN-RESPONSE RANK AND THE EXPONENTIAL KERNEL

The window's coordinate and kernel follow from three requirements on a per-token coefficient, none of which uses the measurement.

A rank weight redistributes a fixed budget. By Proposition 1 the expected forking value of a prompt's rollouts, summed over positions, is $V(1-V)$; the identity fixes this mean over rollouts, not the value of a single response, and a rank weight redistributes it within each response without moving weight between responses. A coefficient that is a function of the entropy values gives two responses of the same length and outcome different total coefficients (for $w_t = C_t$ the total is the length times the response's mean fork probability, which moves by an order of magnitude along training as the batch entropy runs from 0.1 to 3 nats), so the size of a response's update would follow its uncertainty, which its budget does not. A coefficient that is a function of the within-response ranks leaves the total a function of the length alone (Proposition 7, with or without the floor) and redistributes it: the window's shape becomes one object across responses, training regimes and models, and under Adam only that shape and the class ratio reach the update (Lemma 8).

The rank is invariant and bounded. The rank of a token within its response is the same under any monotone transform of the uncertainty measure (the entropy, its exponential the perplexity, any

power of either), whereas a value-based coefficient commits to one scale of the softmax. A kernel on the rank lies in $[0, 1]$ by construction, which keeps the coefficient sign-preserving, the premise of Proposition 3, and bounded by one, so no token receives more than the shared advantage; a z-score of the entropy within the response is signed, and a ratio to the response mean is unbounded.

The exponential is the memoryless kernel. Among kernels on the rank, the exponential $w(q) = 2^{-q/\lambda}$ is the one whose weight ratio between two positions depends only on their rank difference, $w(q+\delta)/w(q) = 2^{-\delta/\lambda}$ (Cauchy's equation for a monotone kernel): it prefers no scale along the rank, and one number, the width, sets the concentration. This is what makes the width a per-model calibration that a tracker can re-measure in the loop and that Table 12 compares across models, and it gives the mass its closed form (Appendix D.4). The measured envelope has this form (R^2 0.99 and 0.95 over the ten bands, the same form on eight models), so the fitted width is the calibration.

D.6 Lemma on Adam and gradient scale

Lemma 8 (Adam and gradient scale)**.** *With gradient clipping inactive and zero initial moments, on coordinates with nonzero second moment (a zero coordinate steps by zero under the convention $0/0 := 0$), multiplying the entire gradient history by a constant $k > 0$ leaves the bias-corrected Adam update (Kingma & Ba, 2015) unchanged at $\epsilon_{\mathrm{Adam}} = 0$ (and replaces ϵ_{Adam} by $\epsilon_{\mathrm{Adam}}/k$ otherwise). For a constant signed coordinate gradient multiplied by $k \geq 1$ after t_0 baseline updates, at $\epsilon_{\mathrm{Adam}} = 0$ the update n steps later relative to the unchanged reference is $F_n = [1 + (k-1)a_1]/\sqrt{1 + (k^2-1)a_2} \leq k$, $a_j = (1-\beta_j^n)/(1-\beta_j^{t_0+n})$.*

Adam keeps its moment estimates (a local use of M and V) $M_t = \beta_1 M_{t-1} + (1-\beta_1)g_t$, $V_t = \beta_2 V_{t-1} + (1-\beta_2)g_t^2$ and steps by $\alpha \hat{M}_t/(\sqrt{\hat{V}_t} + \epsilon_{\mathrm{Adam}})$ with the usual bias corrections; replacing g_t by kg_t throughout gives kM_t and k^2V_t, hence the first claim (decoupled weight decay is unaffected). For the step change, $M_{t_0+n} = \bar{g}(1-\beta_1^{t_0+n})[1 + (k-1)a_1]$ and $V_{t_0+n} = \bar{g}^2(1-\beta_2^{t_0+n})[1 + (k^2-1)a_2]$, giving F_n; with $r = \epsilon_{\mathrm{Adam}}/|\bar{g}|$ the general ratio is $[1 + (k-1)a_1](1+r)/[\sqrt{1 + (k^2-1)a_2} + r]$, still at most k. For $k = 4$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $t_0 = 100$ and $r = 0$: $F_n = 1.89, 1.59, 1.34, 1.12$ at $n = 10, 50, 100, 300$, peak 1.93 near $n = 15$; with equilibrated moments the peak is 3.23. *Consequence for the design.* A common amplitude fixed for the whole run is therefore immaterial under these assumptions but not outside them (a factor introduced mid-run, a non-negligible ϵ_{Adam}, active clipping), whereas changing the ratio of the class mean weights can change the aggregate gradient direction and Adam's moments. CreditFilter equalizes the class mean weights, and the no-floor and rescaled-mass ablations of Table 2 compare placing the equalizing mass on the tail (floor) or on the head (rescaling).

D.7 Tracking the envelope

Estimator and update. At every step, for each class c with at least 64 responses in the batch, the tokens of each response are binned by their midpoint rank q' into the ten bands with edges $(0, .025, .05, .075, .1, .15, .2, .3, .4, .6, .8)$, the band mean of the fork probability $C = 1 - \sum_v \pi_v^2$ (per token, from the same softmax pass as the entropies) is taken per response, the response means are averaged over the responses that have the band (rollout-equal), and $\hat{\lambda}_{C,c,t} = \ln 2/(-\text{slope})$ from an ordinary least-squares fit of the log band means on the band centers; the width is updated only when every band mean is present and positive and the fitted slope is finite and negative; otherwise, including for a flat profile, the previous width is retained. The tracked width is the exponential moving average $\lambda_{c,t} = (1-\alpha)\lambda_{c,t-1} + \alpha\,\hat{\lambda}_{C,c,t}$ with $\alpha = 0.1$, the target and the average both clamped to $[0.05, 0.30]$ (correct) and $[0.07, 0.40]$ (wrong) in the CreditFilter-D run, whose initial widths are $0.111/0.155$; the state is persisted so that a restart resumes the average.

Envelope along training. Table 8 applies this estimator to training-length rollouts of the CreditFilter checkpoints every 50 steps. At step 350, in the aftermath of the entropy excursion, 28% of the rollouts hit the cap and are excluded, and the width of the remaining, shorter wrong responses (0.08) is not comparable with the other rows. CreditFilter-D's tracker, which follows the envelope of its own batches, has widened the wrong-class width to its clamp of 0.40 by step 290 and ends training at 0.28–0.37 over its last 75 updates; its correct-class width is 0.16 at step 290 and 0.13–0.18 at the

Table 8: Envelope widths along the CreditFilter run on training-length rollouts of its checkpoints (224 questions $\times$ 4 rollouts, 20,480-token cap; the tracker's estimator with 95% rollout-bootstrap intervals over 300 replicates). n: rollouts per class among those that finished under the cap; capped: share of the 896 rollouts cut at the cap and excluded (†: more than a fifth). Base-policy widths for reference: 0.111 (correct) and 0.155 (wrong), Table 9.

	correct		wrong		
step	n	λ_C	n	λ_C	capped (%)
50	431	0.060 [0.057, 0.065]	440	0.126 [0.112, 0.142]	3
100	302	0.100 [0.090, 0.108]	578	0.218 [0.208, 0.229]	2
150	388	0.104 [0.091, 0.114]	473	0.243 [0.230, 0.256]	4
200	391	0.099 [0.087, 0.112]	498	0.223 [0.210, 0.236]	1
250	400	0.076 [0.059, 0.088]	493	0.346 [0.320, 0.372]	0
300	483	0.053 [0.051, 0.055]	294	0.327 [0.284, 0.369]	13
350†	471	0.058 [0.054, 0.061]	173	0.082 [0.076, 0.087]	28
400	424	0.054 [0.052, 0.056]	447	0.341 [0.317, 0.365]	3

end. The widths that fail when fixed from the first step, $0.20/0.28$ (Table 2), are close to those the tracker reaches after step 290. The fixed CreditFilter window stays close to the moving envelope: its squared cosine with the exponential of the tracked width is 0.84–0.95 for the correct class and 0.83–1.00 for the wrong class across the checkpoints of this table, against 0.65–0.91 and 0.86–1.00 for the wide window and 0.66–0.81 and 0.55–0.81 for the hard mask.

D.8 Damping multiplier (Proposition 9)

Proposition 9 (damping multiplier)**.** *For binary R with pass probability $p \in (0,1)$, $A = (R-p)/\sqrt{p(1-p)}$ and an unclipped full-trajectory score S, under a valid score identity with a policy-independent reward and with p, the normalization and the damping coefficients held fixed, scaling correct responses by $(1-p)^\gamma$ gives $\mathbb{E}[g(R)AS] = [p + (1-p)^{\gamma+1}]\,\nabla p/\sqrt{p(1-p)}$.*

Damping. With true pass probability $0 < p < 1$, $A = (R-p)/\sigma$, $\sigma = \sqrt{p(1-p)}$, the unclipped full-trajectory score S, a valid score identity with a policy-independent reward, and p, σ and the damping coefficients treated as fixed: $g(1) = (1-p)^\gamma$, $g(0) = 1$; $\mathbb{E}[RS] = \nabla p$ and, since $\mathbb{E}[S] = 0$, $\mathbb{E}[(1-R)S] = -\nabla p$; on $R = 1$, $g(R)A = (1-p)^{\gamma+1}/\sigma$ and on $R = 0$, $g(R)A = -p/\sigma$, so $\mathbb{E}[g(R)AS] = \sigma^{-1}\{(1-p)^{\gamma+1}\nabla p + p\nabla p\} = [p + (1-p)^{\gamma+1}]\nabla p/\sigma$. For $\gamma = 1$ the multiplier $1 - p + p^2 = \frac{3}{4} + (p - \frac{1}{2})^2$ lies in $[\frac{3}{4}, 1)$; the population ascent direction is preserved. The identity does not cover estimated group pass rates, random normalization, windowing, clipping or Adam.

E The forking-value instrument

This appendix describes the two instruments of Section 5.1: the four-branch instrument, whose measurement on Qwen3-8B-Base is summarized in Table 9 and drawn band by band in Figure 3, and the full-support instrument of Appendix E.4.

E.1 Estimator

Setting. At a position t with prefix s_t and next-token law π, the branch set is the sampled token together with the three most probable alternatives under π (four branches). With $\mathrm{cover} = \sum_k \pi(x_k)$ the weights are $\omega_k = \pi(x_k)/\mathrm{cover}$. Each branch is continued J times, independently, with the training sampler (temperature 1, top-p 1.0, a budget of the instrument's 4608-token context less the prefix), and every continuation is graded $g_{k,j} \in \{0,1\}$ by the training grader (strict boxed-answer match against the reference); a length-truncated continuation is graded 0. Let $Q_k = \mathbb{E}[g_{k,j}]$ be the action value of branch k under that sampler, $S_k = \sum_j g_{k,j}$, $\hat{Q}_k = S_k/J$ and $U_k = S_k(S_k -$

Table 9: The forking spectrum of Qwen3-8B-Base by response class (Figure 1a and Figure 3; the numbers behind Section 5.1): widths of the fork-probability envelope and of the forking value; the fork consequence pooled over bands 2–10, its relative slope (count-weighted least-squares slope of $\hat{\kappa}$ over the band centres of bands 2–10, divided by $\bar{\kappa}$) and the top-band gain; 95% rollout-bootstrap intervals. $\hat{\kappa}$ and $\hat{\nu}$ in units of 10^{-3}.

	correct	wrong
responses / positions	294 / 7056	672 / 16128
envelope width λ_C	0.111 [0.099, 0.123]	0.155 [0.146, 0.165]
forking-value width λ_ν	0.125 [0.063, 0.207]	0.146 [0.086, 0.332]
top-band forking value $\mathbb{E}[\hat{\nu} \mid q \le 0.025]$	3.93 [2.64, 5.24]	1.07 [0.55, 1.52]
consequence $\bar{\kappa}$ (bands 2–10)	2.12 [1.21, 3.13]	0.70 [0.43, 0.98]
relative slope of $\hat{\kappa}$ (bands 2–10)	-2.1 [-5.2, 1.1]	0.3 [-3.3, 4.6]
top-band gain $\hat{\kappa}_1/\bar{\kappa}$	2.9 [1.7, 5.6]	2.4 [1.3, 3.9]

$1)/(J(J-1))$. The estimate is

$$\hat{\nu}_t = \sum_k \omega_k(1-\omega_k)\, U_k - \sum_{k\neq l} \omega_k\omega_l\, \hat{Q}_k\hat{Q}_l, \tag{14}$$

signed, and exactly zero when all $4J$ continuations agree.

Unbiasedness. U_k is the fraction of ordered pairs of distinct continuations of branch k that both succeed, a U-statistic of order two, so $\mathbb{E}[U_k] = Q_k^2$; independence across branches gives $\mathbb{E}[\hat{Q}_k\hat{Q}_l] = Q_kQ_l$ for $k \neq l$. Hence the estimate $\hat{\nu}_t$ of equation 14 has expectation

$$\begin{aligned}\mathbb{E}\Big[\sum_k \omega_k(1-\omega_k)U_k - \sum_{k\neq l}\omega_k\omega_l\hat{Q}_k\hat{Q}_l\Big] &= \sum_k \omega_k Q_k^2 - \sum_k \omega_k^2 Q_k^2 - \sum_{k\neq l}\omega_k\omega_l Q_kQ_l \\ &= \sum_k \omega_k Q_k^2 - \Big(\sum_k \omega_k Q_k\Big)^2,\end{aligned}$$

the variance of Q under ω on the branch support: the quantity ν_t of Section 4 restricted to the retained support, with $C_t = 1 - \sum_k \omega_k^2$ the matching fork probability (the identity $\nu = C\kappa$ of Proposition 4 holds on any finite support). The instrument therefore measures ν, C and κ under the renormalized four-branch law; the omitted mass $1 - \mathrm{cover}$ can carry forking value that the retained support does not show, and the full-support instrument of Appendix E.4 includes it. The band estimator $\hat{\kappa} = \mathbb{E}[\hat{\nu} \mid q]/\mathbb{E}[C \mid q]$ inherits the sign of $\hat{\nu}$ and can be negative (Table 11), although $\kappa = \nu/C \in [0, \frac{1}{2}]$.

E.2 Cover and truncation

Cover, the probability mass retained by the four branches, has mean 0.902 (correct) and 0.795 (wrong) and rises with the rank, from $0.69/0.48$ in the top band to at least 0.96 beyond $q = 0.3$: the low-cover positions are those of the head, where the policy spreads mass beyond four tokens, and the top-band gain of Table 9 is a statement about $0.69/0.48$ of the mass at those positions. Summed over positions, $\hat{\nu}$ is 2.4 (correct) and 3.0 (wrong) times the budget $V(1-V)$ of Proposition 1; the window uses only the shape of each class profile, which a class-wide factor leaves unchanged. Of the $4J = 32$ continuations per position, $2.1\%/3.7\%$ are length-truncated and graded 0, $35\%/48\%$ of positions have at least one truncated continuation, and $89.7\%/73.3\%$ of continuations end with a boxed answer.

Continuation horizon. The base-policy continuations run inside the 4,608-token context and a continuation that reaches its end is graded 0, whereas training generates under a 20,480-token budget. Table 10 repeats the spectrum statistics under two alternative treatments of the stored records: only the positions none of whose 32 continuations was truncated (65% of the correct-class and 52% of the wrong-class positions), and every truncated continuation graded 1 instead of 0. The envelope width does not move (it depends on the softmax only); the forking-value width stays inside its interval, and its interval contains the envelope width; the pooled consequence moves within a factor of

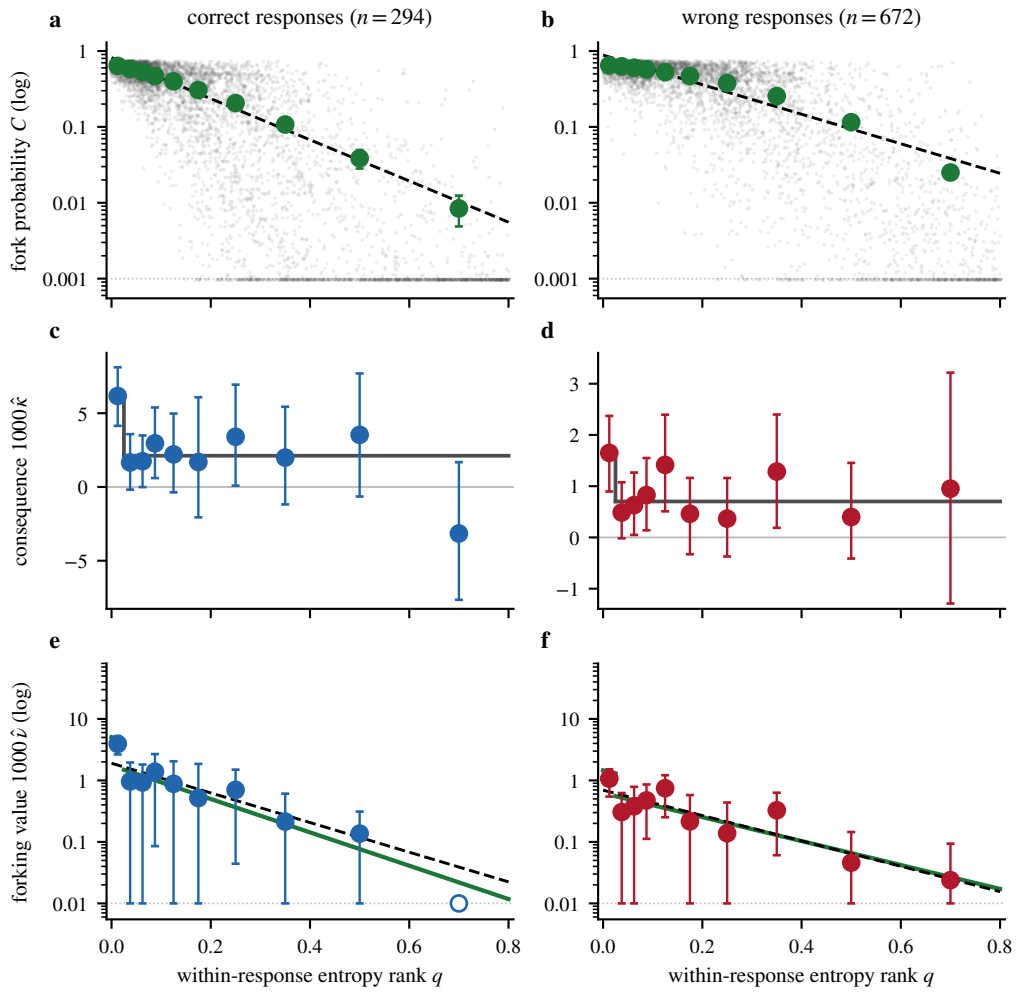


Figure 3: The factorization $\nu = C\kappa$ on Qwen3-8B-Base (left: correct, right: wrong responses; 95% rollout-bootstrap intervals). (a, b) Fork probability C: gray, 6000 positions per class (values below 10^{-3} drawn at 10^{-3}); green, band means; dashed, the exponential fit ($\lambda_C = 0.111$, $R^2 = 0.99$; $\lambda_C = 0.155$, $R^2 = 0.95$). (c, d) Fork consequence $\hat{\kappa}(q) = \mathbb{E}[\hat{\nu} \mid q]/\mathbb{E}[C \mid q]$ per band; gray line, the pooled value $\bar{\kappa}$ over bands 2–10 with the top-band gain on band 1 (values in Table 9). (e, f) Forking value $\mathbb{E}[\hat{\nu} \mid q]$ per band, hollow at the floor for a band mean ≤ 0; green, the reconstruction $\bar{\kappa}\,\hat{C}(q)$ with the top-band gain; dashed, the direct exponential fit ($\lambda_\nu = 0.125$, $R^2 = 0.83$; $\lambda_\nu = 0.146$, $R^2 = 0.81$).

1.5 with overlapping intervals; the top-band gain stays above one in every treatment; the head-to-tail ratio of $\hat{\kappa}$ resolves no trend in any; and the class ratio $\bar{\kappa}_+/\bar{\kappa}_-$ ranges from 3.0 to 4.8.

E.3 Protocol

Corpus. 288 training prompts with 8 responses each at temperature 1, top-p 1.0 (13.7% of responses correct); 640 responses were selected with at least 200 non-format tokens and prompt plus response within 4160 tokens (294 correct, 346 wrong). The wrong class is the union of the wrong responses from mixed-outcome prompts of this draw and of a second draw of the same prompts, 672 responses. Format tokens are excluded from the rank; positions carry the midpoint

Table 10: Continuation-horizon check on the base policy: the spectrum statistics under three treatments of the length-truncated continuations (positions kept; envelope and forking-value widths over the ten bands; fork consequence pooled over bands 2–10, in units of 10^{-3}; top-band gain $\hat{\kappa}_1/\bar{\kappa}$; head-to-tail ratio of $\hat{\kappa}$, $q \leq 0.2$ against $q > 0.2$ within bands 2–10; 95% rollout bootstrap, replicated independently of Table 9).

class	treatment	positions	λ_C	λ_ν	$\bar{\kappa}$	top-band gain	head-to-tail
correct	all positions (paper)	7056	0.111 [0.098, 0.124]	0.125 [0.069, 0.221]	2.12 [1.21, 3.13]	2.9 [1.7, 5.6]	0.67 [0.24, 3.55]
correct	no truncated continuation	4577	0.109 [0.095, 0.121]	0.087 [0.061, 0.240]	3.15 [1.91, 4.49]	2.1 [1.2, 3.6]	0.69 [0.29, 2.81]
correct	truncated graded 1	7056	0.111 [0.098, 0.124]	0.136 [0.080, 0.226]	2.21 [1.27, 3.21]	2.8 [1.6, 5.0]	0.70 [0.29, 3.53]
wrong	all positions (paper)	16128	0.155 [0.146, 0.165]	0.146 [0.093, 0.344]	0.70 [0.46, 0.96]	2.4 [1.3, 3.9]	1.15 [0.42, 6.63]
wrong	no truncated continuation	8443	0.150 [0.141, 0.161]	0.110 [0.074, 0.823]	0.66 [0.29, 1.08]	3.4 [1.5, 7.7]	1.05 [0.28, 10.84]
wrong	truncated graded 1	16128	0.155 [0.146, 0.165]	0.164 [0.089, 0.665]	0.51 [0.22, 0.79]	3.0 [1.3, 6.8]	1.20 [0.26, 12.70]

rank $q' = (r - \frac{1}{2})/T$ of Appendix D.1, with r the 1-based descending entropy rank, T the number of ranked tokens and ties broken by position.

Positions. Per response, $3, 3, 3, 2, 2, 2, 3, 2, 2, 2$ positions are drawn uniformly at random from the ten bands $(0, .025]$, $(.025, .05]$, $\ldots$, $(.6, .8]$, 24 in total, so every response has the same design in every band, and a band mean $\sum_r S_{r,b} / \sum_r N_{r,b}$ over responses r ($S_{r,b}$: the sum of the estimates at the positions of response r in band b; $N_{r,b}$: their number) weights responses equally (the per-band counts of Table 11 differ from this design because positions are re-binned by the midpoint rank q', which shifts some positions across band edges); the tail $q > 0.8$ is not sampled. Table 11 lists the ten-band measurement for both classes.

Table 11: Ten-band measurement on Qwen3-8B-Base (top: 294 correct responses, 7056 positions; bottom: 672 wrong responses, 16128 positions). n: positions per band. Band means with 95% rollout-bootstrap intervals of the forking value $\hat{\nu}$, the fork probability C, the entropy H (nats) and the fork consequence $\hat{\kappa} = \mathbb{E}[\hat{\nu} \mid q]/\mathbb{E}[C \mid q]$; $\hat{\nu}$ and $\hat{\kappa}$ in units of 10^{-3}.

band q	n	$\mathbb{E}[\hat{\nu} \mid q]$	$\mathbb{E}[C \mid q]$	$\mathbb{E}[H \mid q]$	$\hat{\kappa}(q)$
correct responses					
0–0.025	904	3.93 [2.64, 5.24]	0.637 [0.630, 0.643]	2.49 [2.35, 2.62]	6.17 [4.14, 8.12]
0.025–0.05	886	0.96 [−0.08, 1.96]	0.579 [0.570, 0.589]	1.81 [1.70, 1.92]	1.66 [−0.18, 3.58]
0.05–0.075	875	0.93 [0.00, 1.81]	0.530 [0.517, 0.542]	1.51 [1.40, 1.62]	1.75 [−0.01, 3.49]
0.075–0.1	580	1.39 [0.09, 2.68]	0.469 [0.452, 0.488]	1.28 [1.18, 1.38]	2.96 [0.60, 5.39]
0.1–0.15	582	0.88 [−0.03, 2.05]	0.398 [0.376, 0.419]	1.05 [0.95, 1.15]	2.21 [−0.36, 4.98]
0.15–0.2	587	0.52 [−0.58, 1.85]	0.306 [0.285, 0.328]	0.76 [0.69, 0.83]	1.69 [−2.06, 6.08]
0.2–0.3	880	0.70 [0.04, 1.50]	0.206 [0.185, 0.231]	0.50 [0.44, 0.58]	3.40 [0.09, 6.94]
0.3–0.4	586	0.21 [−0.16, 0.61]	0.107 [0.091, 0.125]	0.28 [0.23, 0.34]	1.99 [−1.18, 5.44]
0.4–0.6	589	0.14 [−0.04, 0.31]	0.039 [0.028, 0.050]	0.11 [0.08, 0.14]	3.53 [−0.65, 7.69]
0.6–0.8	587	−0.03 [−0.08, 0.01]	0.008 [0.005, 0.012]	0.02 [0.02, 0.03]	−3.15 [−7.65, 1.68]
wrong responses					
0–0.025	2071	1.07 [0.55, 1.52]	0.647 [0.644, 0.651]	3.65 [3.55, 3.77]	1.65 [0.90, 2.37]
0.025–0.05	2011	0.31 [−0.02, 0.62]	0.628 [0.623, 0.633]	2.81 [2.71, 2.91]	0.49 [−0.02, 1.08]
0.05–0.075	2005	0.38 [−0.01, 0.79]	0.602 [0.594, 0.608]	2.41 [2.32, 2.50]	0.63 [0.05, 1.27]
0.075–0.1	1323	0.47 [0.11, 0.86]	0.572 [0.564, 0.581]	2.12 [2.03, 2.22]	0.83 [0.14, 1.55]
0.1–0.15	1347	0.75 [0.25, 1.22]	0.527 [0.517, 0.537]	1.77 [1.69, 1.85]	1.42 [0.51, 2.40]
0.15–0.2	1335	0.22 [−0.12, 0.58]	0.468 [0.457, 0.481]	1.45 [1.38, 1.53]	0.46 [−0.33, 1.16]
0.2–0.3	2015	0.14 [−0.13, 0.44]	0.377 [0.364, 0.391]	1.06 [1.00, 1.13]	0.37 [−0.37, 1.16]
0.3–0.4	1342	0.33 [0.06, 0.63]	0.255 [0.241, 0.270]	0.70 [0.64, 0.76]	1.29 [0.19, 2.40]
0.4–0.6	1337	0.05 [−0.04, 0.14]	0.115 [0.105, 0.128]	0.34 [0.30, 0.39]	0.40 [−0.41, 1.45]
0.6–0.8	1342	0.02 [−0.03, 0.09]	0.025 [0.020, 0.031]	0.09 [0.07, 0.12]	0.95 [−1.29, 3.22]

Widths. $\lambda_X = \ln 2/(-\text{slope})$ of an ordinary least-squares fit of $\ln \mathbb{E}[X \mid q]$ on the band centers over the bands with positive mean, ten bands unless stated (for $X = C$ every band mean is positive;

for $X = \hat{\nu}$ a negative tail band is dropped, which can lengthen λ_ν, whereas λ_C, the quantity the window uses, is unaffected). The log band means of C are concave: the local half-width falls from 0.18–0.20 over the first bands to 0.09 in the last for correct responses and from 0.58 to 0.09 for wrong ones, and a fit restricted to $q \leq 0.2$ gives widths 0.15/0.34 against the global widths of Table 9; the single exponential is a global summary, narrower than the head and wider than the tail.

Table 12: Forking spectrum on further models with the same instrument (120 responses × 24 positions, 60 for Qwen3-32B, four branches, $J = 8$, top-p 1.0; classes pooled): per model, the number of responses, the ten-band forking-value width λ_ν with its interval, the envelope width λ_C at the same positions, and the coarse-group means of $\hat{\nu}$ (G_1: $q \leq 0.1$, G_2: $0.1 < q \leq 0.4$, G_3: $q > 0.4$). With 60–120 responses the forking-value width intervals are wide, whereas λ_C is resolved on every model. A width point estimate is undefined when the ten-band fit does not decay; its interval is then over the replicates whose fit does.

model	n	λ_ν [95% CI]	λ_C	$\mathbb{E}[\hat{\nu} \mid G]$ (10^{-3}): $G_1/G_2/G_3$
Qwen3-1.7B	120	0.119 [0.047, 1.057]	0.071	0.65 / 0.44 / −0.03
Qwen3-4B	120	– [0.062, > 2]	0.086	0.37 / 0.26 / −0.02
Qwen3-14B-Base	120	0.255 [0.072, 0.913]	0.108	0.79 / 0.04 / 0.16
Qwen3-32B	60	0.307 [0.059, > 2]	0.091	0.25 / 0.17 / 0.03
Qwen3-8B (post-trained)	120	0.218 [0.043, 1.499]	0.084	0.31 / 0.34 / −0.01
Qwen2.5-7B-Instruct	120	0.130 [0.048, 1.225]	0.081	0.41 / 0.12 / −0.02
Llama-3.1-8B-Instruct	120	0.306 [0.094, 1.592]	0.267	0.68 / 0.25 / 0.20

Summary. The width of an exponential fitted to the forking value itself lies within 20% of the envelope's, though its intervals span a factor of three to four (Table 9). Under the full-support instrument, the top 20% of the rank captures 62% (correct) and 82% (wrong) of the forking value (Table 13).

E.4 A FULL-SUPPORT INSTRUMENT AT HIGHER RESOLUTION

Design. The full-support instrument is run once on the base policy. Responses: 16 rollouts of 512 training prompts at the training sampler with a 5,800-token cap; from the mixed-outcome prompts, 400 responses per class (at most two per prompt and class, at least 200 ranked tokens; 296/297 prompts). Positions: one uniform draw from each of 16 rank strata per response (edges 0, 0.0025, 0.005, 0.01, 0.015, 0.025, 0.0375, 0.05, 0.075, 0.1, 0.15, 0.2, 0.3, 0.4, 0.6, 0.8, 1) plus one draw from each of the top global-entropy strata, with recorded inclusion probabilities; every bin is a Horvitz–Thompson mean, so the estimates are unbiased in any coordinate. Branch law: the most probable tokens in descending order until their mass reaches 0.9, with at least two and at most six branches (so the head covers less than 0.9 at the highest-entropy positions, column cover of Table 13), plus two independent draws from the remaining mass r; with $D_{ab} = U_a + U_b - 2\hat{Q}_a\hat{Q}_b$ the estimate $\hat{\nu} = \sum_{a<b} p_a p_b D_{ab} + r \sum_k p_k (D_{k,y_1} + D_{k,y_2})/2 + \frac{1}{2} r^2 D_{y_1,y_2}[y_1 \neq y_2]$ is unbiased for the full-policy ν (checked on 4×10^4 random laws), and the head-restricted pair $(\hat{\nu}_{\mathrm{head}}, C_{\mathrm{head}})$ is kept for comparison with the four-branch instrument. Each branch is continued $J = 48$ times under the training sampler with a budget of 5,800 tokens minus the position index; C, H and π come from a forward pass over every token. Responses are weighted so that a class-c response of a prompt stands for all eligible class-c rollouts of that prompt, the population a training batch sees; intervals are 95% cluster bootstraps over prompts (2,000 replicates). The run took 3.58 million continuations and 2.97 billion generated tokens on 16 GPUs in 6.2 hours; 1.7%/2.7% of the continuations hit the budget.

Where the consequence concentrates (Table 13). The fork consequence is concentrated in the top 0.25% of the rank, about two tokens per response. Pooled over ranks 0.5–2.5% it is 3.29 $[2.17, 4.81]$ against 1.56 $[1.12, 2.08]$ over ranks 2.5–20%, a ratio of 2.10 $[1.29, 3.45]$; for wrong responses 0.73 $[0.44, 1.13]$ against 0.31 $[0.18, 0.47]$, ratio 2.36 $[1.27, 4.62]$. Beyond $q = 0.2$ the fork probability is below 0.17 for correct and 0.39 for wrong responses, and the estimates of $\kappa = \nu/C$ widen: 3.79 $[1.56, 7.33]$ for correct responses (the 2.5–20% value is 0.41 $[0.19, 1.05]$ of it; the estimate is carried by the strata 0.3–0.6 whose C is 0.03–0.08) and 0.13 $[0.01, 0.31]$ for wrong ones

Table 13: The full-support instrument on the base policy, training-population weights: per class and rank stratum, sampled positions, cover of the head branches, fork probability C over all positions of the stratum, forking value $\hat{\nu}$ and consequence $\hat{\kappa}$ under the full law (in units of 10^{-3}, 95% prompt bootstrap) and the consequence under the head-restricted law.

class	rank q	n	cover	C	$\hat{\nu}$ [95%]	$\hat{\kappa}$ [95%]	$\hat{\kappa}_{\mathrm{head}}$
correct	0–0.0025	460	0.64	0.869	8.21 [6.04, 10.86]	9.45 [6.90, 12.61]	11.45
correct	0.0025–0.005	392	0.72	0.832	3.05 [2.15, 4.15]	3.67 [2.58, 4.99]	4.15
correct	0.005–0.01	408	0.76	0.796	2.55 [1.73, 3.48]	3.20 [2.17, 4.38]	3.12
correct	0.01–0.015	408	0.80	0.763	1.68 [0.98, 2.49]	2.20 [1.27, 3.28]	2.40
correct	0.015–0.025	407	0.83	0.720	2.82 [1.26, 5.04]	3.92 [1.78, 6.95]	3.87
correct	0.025–0.0375	401	0.87	0.671	0.62 [0.26, 1.03]	0.93 [0.38, 1.53]	0.78
correct	0.0375–0.05	405	0.89	0.626	0.80 [0.43, 1.21]	1.28 [0.68, 1.96]	1.27
correct	0.05–0.075	405	0.91	0.563	1.44 [0.82, 2.23]	2.56 [1.45, 3.99]	2.53
correct	0.075–0.1	401	0.93	0.485	0.52 [0.13, 1.08]	1.07 [0.27, 2.24]	0.93
correct	0.1–0.15	401	0.95	0.384	0.62 [0.19, 1.22]	1.60 [0.48, 3.20]	1.38
correct	0.15–0.2	401	0.96	0.279	0.41 [0.17, 0.71]	1.47 [0.61, 2.56]	1.43
correct	0.2–0.3	400	0.98	0.169	0.40 [0.18, 0.67]	2.36 [1.11, 3.96]	2.05
correct	0.3–0.4	400	0.99	0.082	0.65 [0.10, 1.63]	7.91 [1.26, 19.31]	8.55
correct	0.4–0.6	400	1.00	0.027	0.07 [0.00, 0.15]	2.44 [−0.01, 5.31]	2.37
correct	0.6–0.8	400	1.00	0.004	0.00 [0.00, 0.00]	0.36 [−0.73, 1.26]	0.30
correct	0.8–1	400	1.00	0.000	0.00 [0.00, 0.00]	1.93 [0.07, 5.20]	3.01
wrong	0–0.0025	534	0.41	0.943	1.84 [1.17, 2.73]	1.95 [1.24, 2.91]	2.23
wrong	0.0025–0.005	411	0.48	0.921	0.76 [0.30, 1.38]	0.82 [0.33, 1.50]	0.81
wrong	0.005–0.01	422	0.56	0.897	0.55 [0.30, 0.88]	0.62 [0.33, 0.98]	0.60
wrong	0.01–0.015	410	0.60	0.872	0.86 [0.34, 1.74]	0.99 [0.38, 1.99]	0.91
wrong	0.015–0.025	417	0.66	0.843	0.55 [0.27, 0.87]	0.65 [0.32, 1.04]	0.53
wrong	0.025–0.0375	417	0.70	0.805	0.05 [−0.08, 0.19]	0.07 [−0.10, 0.24]	0.10
wrong	0.0375–0.05	406	0.75	0.771	0.14 [−0.02, 0.34]	0.18 [−0.03, 0.43]	−0.03
wrong	0.05–0.075	407	0.80	0.724	0.42 [0.09, 0.88]	0.58 [0.13, 1.21]	0.59
wrong	0.075–0.1	403	0.83	0.668	0.29 [0.05, 0.65]	0.43 [0.07, 0.98]	0.41
wrong	0.1–0.15	406	0.87	0.594	0.14 [0.04, 0.25]	0.23 [0.07, 0.42]	0.29
wrong	0.15–0.2	402	0.90	0.505	0.14 [0.01, 0.32]	0.27 [0.02, 0.63]	0.28
wrong	0.2–0.3	404	0.93	0.389	0.05 [−0.04, 0.20]	0.14 [−0.11, 0.50]	0.18
wrong	0.3–0.4	400	0.95	0.254	0.07 [0.02, 0.14]	0.27 [0.07, 0.56]	0.23
wrong	0.4–0.6	401	0.98	0.111	−0.01 [−0.02, 0.01]	−0.07 [−0.24, 0.11]	−0.08
wrong	0.6–0.8	400	1.00	0.022	0.01 [0.00, 0.02]	0.23 [−0.08, 0.89]	0.32
wrong	0.8–1	400	1.00	0.002	0.00 [0.00, 0.01]	1.20 [0.01, 5.42]	2.73

(the 2.5–20% value is 2.4 $[0.9, 15]$ times it). Under question-balanced weights the head ratios are 1.53 $[0.99, 2.36]$ and 2.26 $[1.20, 4.40]$. In the top stratum the full-law consequence lies below the head-law one (9.45 against 11.45 for correct and 1.95 against 2.23 for wrong responses), so the probability mass beyond the head dilutes the fork there; below it the two laws are close. Aggregated to the ten bands of the four-branch instrument, the top band gives 4.05 $[3.04, 5.31]$ against 6.17 $[4.14, 8.12]$ there, overlapping. On this corpus the envelope widths over the ten bands are 0.092 (correct) and 0.138 (wrong) and the forking-value widths 0.080 and 0.113: the forking value decays somewhat faster than the fork probability, and the envelope width is a property of the corpus and sampler on which it is measured (the four-branch corpus gives 0.111 $[0.099, 0.123]$), which is why the width is calibrated per model and the adaptive variant re-measures it.

Table 14: Matched-filter gain retained by weight profiles against the forking-value profile of the full-support instrument (16 strata, width-weighted; the exact profile retains 1). Brackets: 95% question-cluster bootstrap intervals (2,000 replicates).

weight profile	correct	wrong
uniform weights	0.21 [0.12, 0.28]	0.16 [0.10, 0.23]
hard mask on the top fifth of the rank	0.40 [0.23, 0.53]	0.51 [0.31, 0.61]
top 0.25% of the rank only	0.35 [0.17, 0.52]	0.31 [0.13, 0.48]
CreditFilter window (exponential with its floor)	0.61 [0.44, 0.69]	0.64 [0.44, 0.70]
wide window ($0.20/0.28$ with its floor)	0.57 [0.41, 0.65]	0.48 [0.31, 0.58]
reversed rank (exponential in $1 - q$)	0.04 [0.02, 0.05]	0.00 [0.00, 0.00]
exponential fitted to the forking value	0.66 [0.45, 0.74]	0.67 [0.47, 0.73]
CreditFilter window with a threefold top-0.25% gain	0.79 [0.57, 0.84]	0.79 [0.57, 0.83]
piecewise $C\kappa$ (fitted on the same estimates)	0.93 [0.69, 0.95]	0.91 [0.66, 0.92]

What the envelope window retains (Table 14). With the forking-value profile $m(q) = \mathbb{E}[\hat{\nu} \mid q]$ of the 16 strata as the target and the strata widths as position weights, a weight profile w retains the fraction $\langle w, m\rangle^2/(\|w\|^2\|m\|^2)$ of the matched-filter gain of m itself (the mean-square criterion of Proposition 3 with D constant, ignoring the value factor). An exponential refitted to the forking value gains little over the window, so the width is not the limiting factor, whereas a threefold gain on the top 0.25% gains more; the piecewise profile $C\kappa$ with the measured consequence per region is fitted on the same estimates it is scored against, so its value is an in-sample reference. Question-bootstrap intervals (Table 14) are wide, and resampling adds noise to the target, so they lie below the point values; the paired differences are better resolved: the window retains 0.21 $[0.13, 0.25]$ (correct) and 0.13 $[0.02, 0.23]$ (wrong) more than the hard mask, 0.40 $[0.20, 0.50]$ and 0.48 $[0.33, 0.51]$ more than uniform weights, and 0.04 $[-0.05, 0.11]$ and 0.16 $[0.09, 0.18]$ more than the wide window. The top-rank gain is not used in training: the top stratum is where the four-branch cover is lowest and it holds two tokens per response.

E.5 Bootstrap

All intervals of the four-branch instrument (Tables 9–12) are 95% percentile intervals of a bootstrap over responses (the 24 positions of a response are resampled together): 400 replicates for band means, 600 for the κ statistics and 300 for widths, where replicates whose fit does not decay are excluded (none for the base spectra and envelopes).

F Additional training results

This appendix gives the training curves of the nine runs (Figure 4), lengths and unanswered shares at step 400 (Table 15) and the comparison repeated on a post-trained model (Table 16).

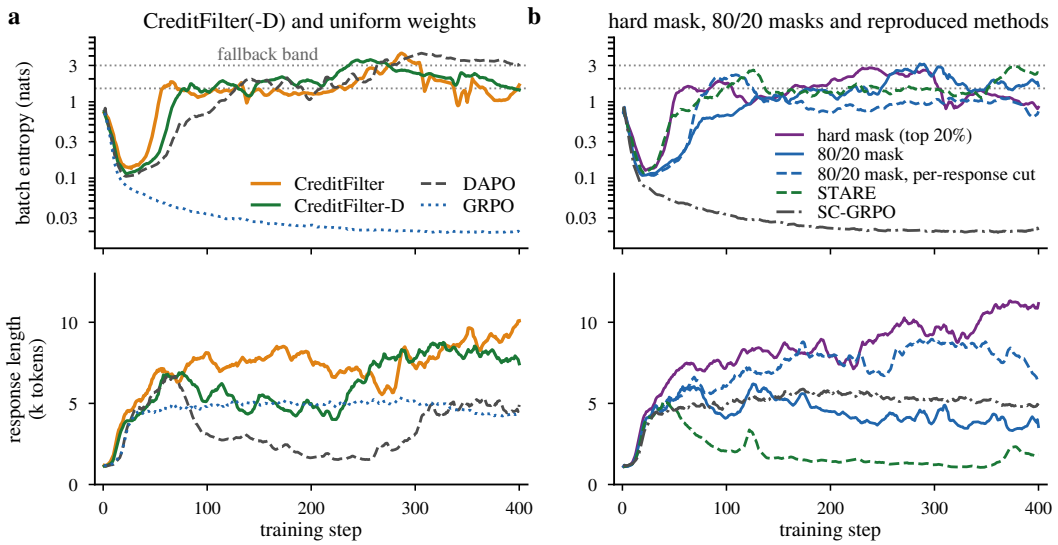


Figure 4: Training dynamics of CreditFilter against the baselines. Top: batch entropy (five-step moving average, log axis; horizontal dotted lines: the fallback band). Bottom: mean response length (five-step moving average). (a) CreditFilter and CreditFilter-D against the uniform-weight recipes DAPO and GRPO; (b) the hard mask on the CreditFilter recipe, the 80/20 mask (batch cut and per-response cut) and the reproduced methods STARE and SC-GRPO.

Length and unanswered samples. The share of samples that reach the cap (Table 15) is larger than the unanswered share: at the late checkpoints CreditFilter reaches the cap in 3.8–20.6% of its MATH500 samples and leaves 0.3–1.8% unanswered, and its checkpoint at step 350, taken shortly after the entropy excursion of Figure 4, has the largest cap share and 1.8 / 5.4 / 3.0% unanswered.

The base model already reaches the cap after having boxed an answer in 3.8–5.0% of its samples, and the trained runs inherit the habit: at step 400, 3.3–3.6% of CreditFilter's samples and 7.5–11.6% of DAPO's reach the cap after their answer, while GRPO, whose entropy has collapsed (Figure 4), stops at the answer (0.0–0.3%). Generated length therefore mixes reasoning length with continuation after the answer; Table 16 separates the two on a post-trained model.

Table 15: Step-400 checkpoints under the mean@32 protocol: mean generated tokens and share of samples (%) that reach the 20,480-token cap without a boxed answer (unans.) or after having boxed one (at cap), per set (MATH500 $n = 500$, AIME24 $n = 30$, AIME25 $n = 30$; 32 samples per problem). The base model is given at step 0.

	MATH500			AIME24			AIME25		
Run	tokens	unans.	at cap	tokens	unans.	at cap	tokens	unans.	at cap
CreditFilter	6,346	0.3	3.6	9,797	4.0	3.5	9,188	0.8	3.3
hard mask (top 20%)	8,467	3.3	15.4	12,109	10.2	4.0	11,250	13.4	3.5
80/20 mask, per-response cut	6,228	5.7	13.3	10,972	25.5	4.8	9,699	20.8	4.6
80/20 mask	2,446	5.8	0.9	3,961	6.6	0.6	3,112	5.6	0.3
SC-GRPO	2,430	1.3	0.4	5,215	1.1	0.4	4,954	4.8	0.1
STARE	867	0.4	0.7	1,480	1.3	0.5	1,169	1.1	0.3
DAPO	2,848	0.9	7.5	5,361	4.6	11.6	3,667	1.5	7.8
GRPO	2,020	1.0	0.1	4,553	2.4	0.0	4,266	4.5	0.3
base model (step 0)	3,347	10.6	3.8	6,225	22.5	5.0	4,581	14.8	4.4

Table 16: Post-trained model (Qwen3-1.7B, non-thinking template, 8,192-token cap; mean@32 with temperature 1.0 and top-p 0.7; one run per configuration): accuracy, mean generated tokens and share of samples unanswered at the cap (%), each averaged over checkpoints 350/375/400 (the untrained model at step 0), per set (MATH500 $n = 500$, AIME24 $n = 30$, AIME25 $n = 30$). Single-checkpoint 95% half-widths are about 3 points on MATH500 and 9–13 points on the AIME sets.

	MATH500			AIME24			AIME25		
Run	acc.	tokens	unans.	acc.	tokens	unans.	acc.	tokens	unans.
untrained (step 0)	0.778	2,035	7.4	0.260	6,751	59.2	0.196	6,289	57.4
DAPO	0.749	1,663	8.9	0.226	6,166	58.2	0.159	4,632	41.9
CreditFilter, Qwen3-8B-Base widths ($0.10/0.14$, floor 0.1)	0.773	1,581	1.5	0.257	5,490	15.6	0.184	4,693	16.2
CreditFilter, measured widths ($0.059/0.061$, floor 0.006)	0.785	1,432	0.2	0.267	3,701	1.7	0.210	3,344	1.1
CreditFilter-D, tracked widths (init. $0.059/0.061$, clamps 0.5–2×)	0.797	2,474	10.9	0.227	6,163	52.7	0.209	6,206	51.8

Post-trained model. Table 16 repeats the comparison on Qwen3-1.7B, a post-trained model that stops after its answer, with an 8,192-token response cap in training and evaluation, a 2,048-token overlong buffer, 32 prompts with 16 responses per step and mini-batch 8, otherwise the recipe of Appendix B. CreditFilter is run with the widths of the Qwen3-8B-Base study ($0.10/0.14$, floor 0.1) and with widths measured on this model by the envelope fit of Appendix E ($0.059/0.061$ per class from 600 training prompts with eight responses each; Table 12 gives the class-pooled width 0.071 of the same model from 120 responses at 24 positions each; the floor 0.006 equalizes the class masses). The fallback band of all three CreditFilter runs on this model is $0.58/1.16$, twice and four times the model's token entropy of 0.29. The measured widths keep the accuracy of the untrained model, lead DAPO on MATH500 and are level with it on the AIME sets. The measured-width run's responses are 14% shorter than DAPO's on MATH500 and 28–40% shorter on the AIME sets, and 0.2–1.7% of its samples are unanswered against DAPO's 9–58%; the Qwen3-8B-Base widths give lengths close to DAPO's and unanswered shares of 1.5–16%. CreditFilter-D is run with its widths initialized at the measured values and clamped to 0.5–2× them (0.030–0.118 / 0.031–0.122), no floor and the class masses equalized by rescaling. Its late accuracy is level with the measured-width run on MATH500 and AIME25 and below it on AIME24 (Table 16).

G RELATED-WORK COMPARISON

Table 17 places CreditFilter among the token-level methods of Section 2.

Table 17: Token reweighting and entropy control in critic-free RLVR, as each paper describes its method. Among prior methods only OAR-P measures the fork (how much replacing the token shifts the final-answer distribution), per token and with one extra forward pass per token; CreditFilter measures it offline, on the base policy and aggregated over the entropy rank, and builds its weight on the fork probability alone, with no extra pass.

Method	Forking proxy	Hard/soft	Fixed/adaptive	Fork measured?	Entropy control
Token reweighting					
80/20 mask† (Wang et al., 2025)	token entropy, batch top 20%	hard	fixed fraction	no	none
GTPO / GRPO-S (Tan et al., 2026)	entropy share	soft	fixed rule	no	none
Cov.-aware GRPO (Wang et al., 2026)	probability–advantage covariance	soft	data-dependent kernel	no	stable entropy (stated aim)
OAR-P (Li et al., 2026)	counterfactual perturbation	soft	fixed rule	yes, per token (one forward pass per token)	none
OAR-G (Li et al., 2026)	input-gradient saliency	soft	fixed rule	no	none
IAPO (He et al., 2026)	conditional MI with the answer	soft	fixed rule	no	none
SC-GRPO† (Shan et al., 2026)	self-conditioned per-token KL	soft	fixed rule	no	none
Entropy control					
Clip-higher / clip-low (Yu et al., 2025; Park et al., 2025)	none	—	—	no	asymmetric trust region
Clip-Cov / KL-Cov (Cui et al., 2025)	covariance (control only)	hard / soft	fixed fraction	no	clip / penalize high-covariance tokens
ERPO (Yu et al., 2026)	entropy (decision pivots)	soft	adaptive gate	no	entropy-aware gating
STARE† (Luo et al., 2026)	batch surprisal quantiles	piecewise	adaptive, closed loop	no	target-entropy loop
CreditFilter (ours)	fork-probability envelope over the within-response entropy rank	soft	fixed widths; class masses equalized by the floor	offline: base policy, aggregated over the rank	entropy-gated fallback to uniform credit
CreditFilter-D (ours)	same	soft	widths tracked per batch; class masses equalized by rescaling	same	same

Entropy control: what a method adds to its host recipe; every run keeps its host's upper clip ratio (Tables 5 and 6). Extra cost per response: OAR-P adds a forward pass per token, SC-GRPO a teacher forward pass and OAR-G a saliency pass. †Reproduced and compared in Table 1.

H LOCAL DISPLACEMENT OF THE POLICY

This appendix gives the protocol and every measurement (Table 18) behind the local displacement of Section 5.3 (Figure 2).

Protocol. The states are three checkpoints of the CreditFilter run, steps 250, 275 and 300 (batch entropies 1.90, 3.24 and 3.25 nats), and, as a control, the step-300 checkpoint of the DAPO run (4.10 nats). From each checkpoint two continuations of five steps are trained with the checkpoint's optimizer state and data order: one keeps the window with the fallback switched off ($m_{\text{gov}} \equiv 1$), the other uses uniform credit ($m_{\text{gov}} \equiv 0$, the damping $(1-\hat{p}_i)^{\gamma}$ kept); damping, clipping, dynamic sampling and every other setting are those of training. At step 275 a third continuation freezes the fallback near $m_{\text{gov}} = 0.5$ (band $1.24/5.24$ nats, realized $m_{\text{gov}} = 0.49$–0.57 over the five steps). For the measurement, 128 training prompts $\times$ 16 rollouts are sampled from the starting policy with the training sampler (temperature 1.0, top-p 1.0, 20,480-token cap), and a teacher-forced pass under the starting policy and under each continued policy records the sampled token's log-probability and the conditional entropy at every response position. The displacement is the per-token mean of $\log \pi_t(x_t \mid s_t) - \log \pi_{t+5}(x_t \mid s_t)$, the estimator k_1, which is unbiased for $\mathbb{E}_{s\sim\pi_t}\mathrm{KL}(\pi_t(\cdot \mid s) \,\|\, \pi_{t+5}(\cdot \mid s))$ on the starting policy's samples; $k_3 = r - 1 - \log r$ with $r = \pi_{t+5}/\pi_t$ is reported as a second estimator. Intervals are a bootstrap over prompts (2,000 draws); the class split uses the

verifier outcome of the rollout; ΔH is the per-token mean change of the conditional entropy on the same contexts. The parameter displacement is the Euclidean norm of $\theta_{t+5} - \theta_t$ over all 8.19B actor parameters. The uniform continuation at step 275 was trained twice; the two runs agree within 1% in KL and 2% in parameter displacement.

Table 18: Every continuation of the displacement measurement: parameter displacement $\|\theta_{t+5} - \theta_t\|_2$, KL per token from the starting policy (in units of 10^{-3}; k_1 with its 95% prompt bootstrap, k_3), change of the conditional entropy on the starting contexts (nats per token) and the KL split by the rollout's class (correct / wrong, same units); $\bar{H}$: token-mean entropy of the starting batch (nats).

State	$\bar{H}$	continuation	$\|\Delta\theta\|$	KL k_1 [95%]	k_3	ΔH	KL correct / wrong
step 250 (in band)	1.90	window kept	0.284	22.1 [20.2, 24.0]	22.5	+0.017	5.4 / 28.4
step 250 (in band)	1.90	uniform	0.535	11.7 [10.7, 12.6]	12.4	+0.077	3.3 / 14.8
step 275 (excursion onset)	3.24	window kept	0.264	23.1 [21.3, 24.8]	23.3	−0.085	4.9 / 29.4
step 275 (excursion onset)	3.24	uniform	0.383	16.1 [14.8, 17.3]	16.8	−0.016	3.5 / 20.4
step 275 (excursion onset)	3.24	fallback $m_{\mathrm{gov}} \approx 0.5$	0.316	20.4 [18.7, 21.9]	20.7	+0.023	4.2 / 25.8
step 300 (late excursion)	3.25	window kept	0.232	152.9 [137.6, 167.9]	148.2	−0.102	22.5 / 188.7
step 300 (late excursion)	3.25	uniform	0.234	67.1 [60.9, 73.2]	63.9	−0.232	10.6 / 82.6
DAPO step 300 (control)	4.10	window kept	0.198	56.7 [53.9, 59.2]	55.3	−0.219	36.4 / 64.4
DAPO step 300 (control)	4.10	uniform	0.250	65.9 [62.7, 68.7]	64.5	−0.248	42.2 / 74.9

Results. At step 300 the gradient norm of the uniform continuation exceeds the clipping norm at every step and that of the windowed continuation in four of the five, which equalizes their parameter steps (ratio 0.99, Figure 2c). Per unit of squared parameter displacement the windowed update moves the policy 2–7 times further at the three CreditFilter states and 1.4 times further at the DAPO-trained control. There the KL ratio itself is 0.86, because the windowed continuation takes the smaller parameter step. At step 275 the frozen-fallback continuation lies between the two in both quantities, 3% from the linear interpolation in m_{gov}. At step 250 the batch entropy of the continuations separates within five steps (window kept $2.08 \to 3.27$ nats, uniform $2.07 \to 2.58$) while the conditional entropy on the fixed starting contexts rises slightly under both ($+0.017$ nats per token under the window and $+0.077$ under uniform credit): the rise under the window comes from the change of the generated contexts. The excess displacement of the windowed update is spread over the whole response: its share in the top 20% of the entropy rank is 0.24–0.36 against 0.29–0.37 for the uniform update, and its ratio to the uniform update is largest in the lower half of the rank at steps 250 and 300.