MoRe: Sensing Prompt Utility Through Gate Modulation for Sample-Efficient RLVR

Anonymous authors
Paper under double-blind review

Abstract

Reinforcement learning with verifiable rewards (RLVR) must allocate a limited rollout budget across prompts. We observe systematic differences in full-parameter gradient energy across prompts, even after accounting for observed success counts and prompt length. To characterize one model-internal source of this variation, we introduce *modulation reactivity* (MoRe), a closed-form measure of final-layer prompt-cache sensitivity to gate modulation that preserves bilinear interactions. MoRe uses only current weights and prefill activations, requiring no additional rollouts or backward passes. The same Jacobian connects cache responses to gradient feedback, giving MoRe a normalized expected gradient-energy interpretation under isotropic reference feedback. Among prompts matched on observed success rate and length, the high-MoRe group exhibits up to twice the cache-path feedback energy of the low-MoRe group. Fixing within-prompt attention proportions removes more than half of this matched feedback gap without changing MoRe. These findings motivate a two-stage sampler that uses reward beliefs to retain prompts likely to yield mixed rewards, then allocates a fixed number of rollout slots using current MoRe scores. On mathematical reasoning benchmarks, the sampler improves average accuracy by 1.92 percentage points over vanilla GRPO.

1 Introduction

Reinforcement learning (RL) has substantially advanced the reasoning capabilities of large language models (LLMs), particularly in mathematics and coding (Shao et al., 2024; Guo et al., 2025). In these domains, automatically checked outcomes allow models to learn from their own generated solutions. Yet RL with verifiable rewards (RLVR) remains computationally demanding because it relies on repeated response generation. Making better use of this rollout budget requires selecting prompts that provide informative training signals.

Prompt difficulty offers a natural starting point: in GRPO, identical rewards within an answer group yield zero advantages, so prior work prioritizes prompts that produce reward contrast. Dynamic sampling (Yu et al., 2025) retains groups with mixed rewards but still pays for discarded rollouts. Predictive samplers (Qu et al., 2026a; Mao et al., 2026; Qu et al., 2026b; Zhu et al., 2026) reduce this overhead by estimating success rates from training history and selecting prompts before generation. However, maximizing the likelihood of nonzero signals only addresses the availability of learning opportunities. Figure 1(c) shows that, after adjusting for observed success counts and prompt length, differences in full-parameter GRPO gradient energy remain reproducible across independent answer groups. This raises a complementary question: *can we identify, among prompts with comparable estimated success rates, those to which the current policy is more sensitive before generating any response?*

The key is how learning signals propagate: reward contrast determines the advantages, while the model's computation determines how those advantages reach its parameters. Per-example gradient magnitude has long served as a measure of example importance (Katharopoulos & Fleuret, 2018; Paul et al., 2021). In RLVR, existing gradient-based selectors estimate gradients with additional rollouts (Yang et al., 2026) or fix an ordering offline (Wang et al., 2025). HIVE (Wu et al., 2026) instead scores candidates at prefill under the current policy, but its prompt-token entropy characterizes the predictive distribution rather than how gradient feedback reaches the parameters. We therefore

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.

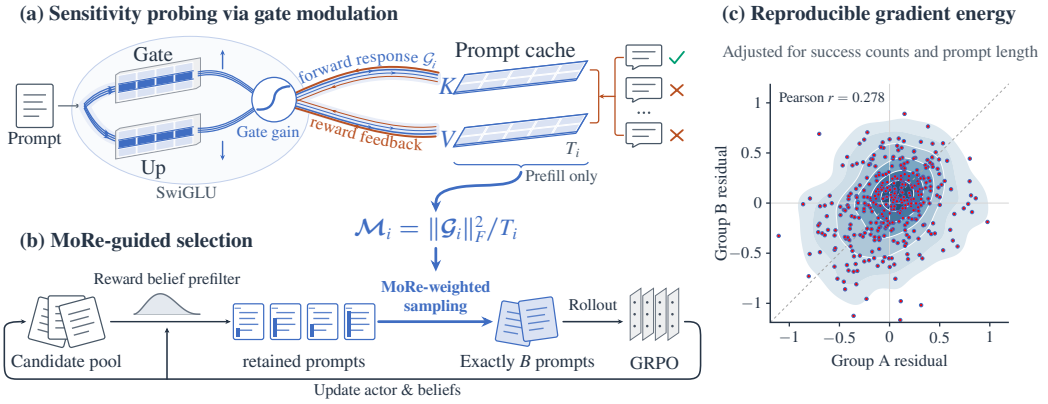


Figure 1: MoRe-guided prompt selection. (a) Gate modulation probes prompt-cache sensitivity during prefill. (b) Reward beliefs prefilter candidates; MoRe-weighted sampling selects exactly B prompts. (c) Log-energy residuals of full-parameter GRPO gradients across independent answer groups at step 100 ($n = 368$ prompts with mixed rewards in both groups). Residuals adjust for observed success count, prompt-length stratum, and both groups' success counts. Shading shows density; the dashed line marks equal residuals.

seek a low-cost online signal that reflects differences in gradient feedback strength across prompts under the current model.

To construct this signal, we probe the prompt key–value (KV) cache of the final attention block, which provides a short path from prompt representations to the answer distribution. Our probe varies gate gain in the preceding SwiGLU layer (Shazeer, 2020), analogous to neuronal gain modulation that changes response amplitude (Salinas & Thier, 2000). Reciprocal gate/up scaling changes this gain while preserving each channel's bilinear interaction for every input (Figure 1(a); Appendix A.2). This construction yields a closed-form Jacobian from normalized gate coordinates to the prompt KV cache. We define the *modulation reactivity* (MoRe) score $\mathcal{M}$ as its squared Frobenius norm per prompt token.

The same Jacobian links the MoRe score to gradient feedback: its transpose carries the GRPO gradient from the prompt cache back to the gate coordinates. Under isotropic reference feedback, MoRe is proportional to the expected squared norm of this cache-path gradient per prompt token. The score uses only current weights and prefill activations, so it is available before decoding without additional rollouts or backward passes.

Actual reward-weighted feedback need not be isotropic, so we test its association with MoRe using feedback energy, defined as the mean squared cache-path gradient norm. After matching observed success rate and prompt length, the high-MoRe group has twice the feedback energy of the low-MoRe group at step 100 (Figure 3(a)). At this checkpoint, grouping by current MoRe scores produces larger differences in mean feedback energy than grouping by initialization scores (Section 2.2).

To understand these cache-path feedback differences, we examine how attention combines value responses and changes their readout weights (Section 2.3). Fixing within-prompt attention proportions removes more than half of the feedback gap between the matched high- and low-MoRe groups while leaving MoRe unchanged. This intervention identifies attention redistribution as a major contributor.

We assign reward beliefs and MoRe to complementary decisions in a two-stage sampler (Figure 1(b)). Reward beliefs first retain prompts likely to produce mixed rewards. Within this pool, MoRe scores from the current actor allocate rollout slots according to model sensitivity, with a correction for the success-rate-dependent scaling of finite-group advantages. Each step selects a fixed number of prompts, and only selected prompts receive rollouts. On mathematical reasoning benchmarks, the sampler improves average accuracy by 1.92 percentage points over vanilla GRPO.

We make three contributions:

- **A closed-form MoRe score.** We derive MoRe to quantify prefill KV-cache sensitivity. The same Jacobian governs the forward cache response and backward GRPO feedback along the cache path; under isotropic reference vectors, MoRe equals the normalized expected squared response in either direction.
- **Empirical and mechanistic evidence.** We show that MoRe captures differences in cache-path feedback energy among prompts matched for observed success rate and length. We further find that shared channels produce coherent V-cache responses, and use controlled-attention interventions to isolate the contribution of attention redistribution to the resulting feedback gap.
- **A fixed-budget two-stage sampler.** We propose a sampler that combines reward beliefs with the current policy's MoRe scores and accounts for the success-rate-dependent scaling induced by finite-group advantage normalization.

2 From Gate Modulation to Gradient Feedback

Gradient feedback along the final-layer prompt KV cache path combines a Jacobian determined during prefill with cache gradients that depend on sampled answers and rewards. A tractable measure of this Jacobian would expose part of the model's sensitivity before decoding. We construct such a measure through gate modulation and relate it to feedback under an isotropic reference. Its association with actual reward-weighted feedback is then an empirical question, which we examine through matched comparisons and attention readout.

2.1 Measuring Modulation Reactivity

Each sampled answer contributes its own gradient at the cache, while all answers share the same cache Jacobian at a fixed prompt and actor. To probe this map before decoding, we choose parameter directions whose effect on the cache is interpretable and available in closed form. The preceding SwiGLU FFN provides a local probe because its residual output feeds directly into the normalization and K/V projections that form the final-layer cache.

Within this FFN, varying gate gain while holding each channel's bilinear interaction fixed isolates how changes in gain propagate to the cache. A bias-free channel k (Shazeer, 2020) produces $d_k(g_{tk}u_{tk})\sigma(g_{tk})$ at prompt position t, with $g_{tk}u_{tk}$ as its bilinear interaction (Pearce et al., 2025) and $\sigma(g_{tk})$ as its gate gain. Here $g_{tk} = w_{g,k}^\top h_t$ and $u_{tk} = w_{u,k}^\top h_t$ are the gate and up preactivations for FFN input h_t, d_k is a down-projection column, and σ is the sigmoid. Let α_k control the modulation of channel k, with $\alpha_k = 0$ at the current weights. We vary the two input projections by reciprocal factors:

$$w_{g,k}(\alpha_k) = e^{\alpha_k/s_k} w_{g,k}, \qquad w_{u,k}(\alpha_k) = e^{-\alpha_k/s_k} w_{u,k}, \qquad s_k^2 = \|w_{g,k}\|^2 + \|w_{u,k}\|^2. \tag{1}$$

This preserves $g_{tk}u_{tk}$ at every position and changes only the gain. Normalization by s_k makes the perturbation scale comparable across channels: each coordinate has a unit-norm tangent in the Euclidean parameter metric, and disjoint parameter rows make these directions orthogonal. A single coordinate α_k acts across all prompt positions. At a fixed actor, every prompt is probed along the same parameter directions and at the same scale. For linearly independent gate and up rows, reciprocal scaling is the unique local direction, up to scale, that preserves the product for every input (Appendix A.2).

Let $\alpha = (\alpha_k)_k$ collect these channel coordinates and $\mathbf{C}$ denote the final-layer prompt KV cache, vectorized over prompt positions and KV heads. The cache Jacobian at the current weights is $\mathcal{G} = \partial\mathbf{C}/\partial\alpha|_{\alpha=0}$. For G on-policy answers Y_j to prompt x with binary-reward GRPO advantages A_j, the cache-path gradient in these coordinates, with other paths held at their baseline values, is

$$\widehat{g}^{\text{cache}} = \mathcal{G}^\top e, \qquad e = \frac{1}{G}\sum_{j=1}^{G} A_j \nabla_{\mathbf{C}} \log P(Y_j \mid x). \tag{2}$$

Here $\mathcal{G}$ depends only on current weights and prompt activations, while e contains the answer- and reward-dependent part of the gradient. Because the parameter tangents are orthonormal, $\|\widehat{g}^{\text{cache}}\|^2$

equals the squared norm of the cache-path parameter gradient projected onto the gain-modulation subspace (Appendix A.2). The same Jacobian governs the forward cache response, $\delta\mathbf{C} \simeq \mathcal{G}\,\delta\alpha$.

The MoRe score measures this sensitivity as the Jacobian's response energy per prompt token:

$$\mathcal{M} = \frac{\|\mathcal{G}\|_F^2}{T} = \mathcal{M}_K + \mathcal{M}_V, \tag{3}$$

where T is the prompt length and the subscripts separate the K and V contributions. Dividing by T compares the average cache-response energy per prompt token along the common modulation directions.

The score factors into a local gate response and its propagation to the cache. Differentiating the sigmoid gain gives a scalar channel response z_{tk}. Let L_t be the Jacobian of RMSNorm (Zhang & Sennrich, 2019) followed by the K/V projections at position t. Then

$$z_{tk} = (g_{tk}u_{tk})\sigma'(g_{tk})\frac{g_{tk}}{s_k} = \frac{u_{tk}g_{tk}^2\sigma'(g_{tk})}{s_k}, \qquad \mathcal{M} = \frac{1}{T}\sum_{t,k} z_{tk}^2\|L_t d_k\|^2.$$

Here z_{tk} is the scalar source response. The factor $\|L_t d_k\|^2$ is the squared norm of the cache response to a unit source amplitude. The RMSNorm derivative and projection weights yield an explicit expression for this factor (Appendix A.1). MoRe can therefore be evaluated from current weights and prefill activations without constructing $\mathcal{G}$ or performing additional rollouts or backward passes.

To relate this prefill sensitivity to actual feedback, take the second moment of Equation (2) at a fixed prompt and actor:

$$\mathbb{E}\|\widehat{g}^{\text{cache}}\|^2 = \operatorname{tr}\big(\mathcal{G}^\top \mathbb{E}[ee^\top]\mathcal{G}\big)\,. \tag{4}$$

The cache Jacobian determines how feedback propagates, while $\mathbb{E}[ee^\top]$ weights the cache directions according to the sampled answers and rewards. A reference feedback vector ξ with isotropic second moment $\mathbb{E}[\xi\xi^\top] = \sigma_e^2 I$ isolates the Jacobian's response energy:

$$\mathbb{E}_\xi\|\mathcal{G}^\top\xi\|^2 = \sigma_e^2\|\mathcal{G}\|_F^2 = \sigma_e^2 T\mathcal{M}. \tag{5}$$

Thus, MoRe gives the reference feedback energy normalized by its per-coordinate input second-moment scale and prompt length. The same Frobenius norm determines the leading-order mean squared forward response to isotropic gate perturbations (Appendix A.3). For actual feedback, the second-moment matrix in Equation (4) also matters. Section 2.2 tests whether MoRe distinguishes this energy among prompts with comparable observed success rates and lengths.

2.2 Feedback Beyond Reward History

Reward counts determine the GRPO advantages, but the resulting feedback also depends on the cache gradients of correct and incorrect answers. For a mixed-reward group with K successes, the binary-reward contrast identity (Bay & Yearick, 2026) gives

$$\widehat{g}^{\text{cache}} = c_K(\bar{s}_+ - \bar{s}_-), \qquad c_K = \frac{K(G-K)}{G^2\delta_K}, \tag{6}$$

where $\bar{s}_+$ and $\bar{s}_-$ average the per-answer cache-path log-probability gradients over correct and incorrect answers, and δ_K is the sample reward standard deviation plus the stabilizer. Fixing the success count fixes c_K, while the gradient contrast still depends on the prompt and current policy (Appendix A.3).

We test whether MoRe distinguishes this feedback using Qwen2.5-0.5B-Instruct (Qwen et al., 2025) along a uniform-GRPO trajectory on GSM8K (Cobbe et al., 2021). Of 1,536 training prompts, 512 were used to develop the analysis; the remaining 1,024 supply the main diagnostic results. At initialization and step 100, one set of 16 answers estimates each prompt's success rate, and an independent set forms two feedback groups of eight. Feedback energy is the mean squared cache-path gradient norm across groups, including identical-reward groups with zero feedback. This measures both the occurrence and magnitude of feedback.

Within strata defined by observed success count and prompt-length tertile, we divide prompts into tertiles of current MoRe and assign the three groups the same stratum weights. The high-MoRe

group's feedback energy is about 1.4 times the low-MoRe group's at initialization and about 2 times at step 100 (Figure 3(a)). The ordering persists on the development set and within mixed-reward groups, showing an energy difference among groups that already provide reward contrast. Mean answer length is similar across groups at initialization and slightly shorter for the high-MoRe group at step 100.

Feedback energy combines the squared norm of mean feedback with variation across sampled answer groups. In an independent resampling study on 72 matched prompts at step 100, the high-MoRe group has about 2.1 times the feedback energy of the low-MoRe group, while the difference in the squared norm of mean feedback remains unresolved (Appendix C.1).

The relationship between MoRe and feedback energy evolves with the policy. To trace this evolution on the same prompts, Figure 2 fixes bins using the MoRe scores of 1,024 prompts at step 100 and follows their feedback at steps 0, 20, and 100. At initialization, feedback energy varies little across these fixed bins. By step 100, the highest-score bins have several times the value of the lowest.

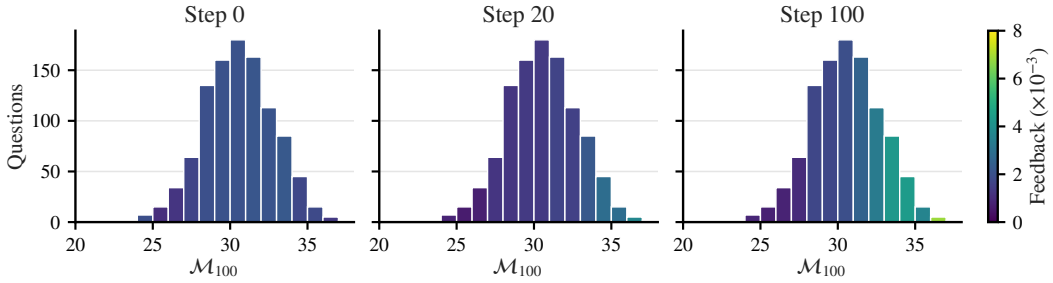


Figure 2: Feedback energy for the same 1,024 prompts, binned by their step-100 MoRe score. Bar heights give prompt counts and are identical across panels; colors give the mean squared cache-path gradient norm at each checkpoint, with identical-reward groups counted as zero. Steps 0 and 100 use two groups from the second answer set; step 20 uses one group from each set.

At step 100, we directly compare current and initialization MoRe scores with the feedback measurements, observed success-count and length strata, and the same 1,024 prompts held fixed. Grouping by current scores yields a high–low feedback-energy gap about 3.1 times that obtained with initialization scores (Appendix B.2). This comparison supports recomputing MoRe with the current actor when allocating rollouts.

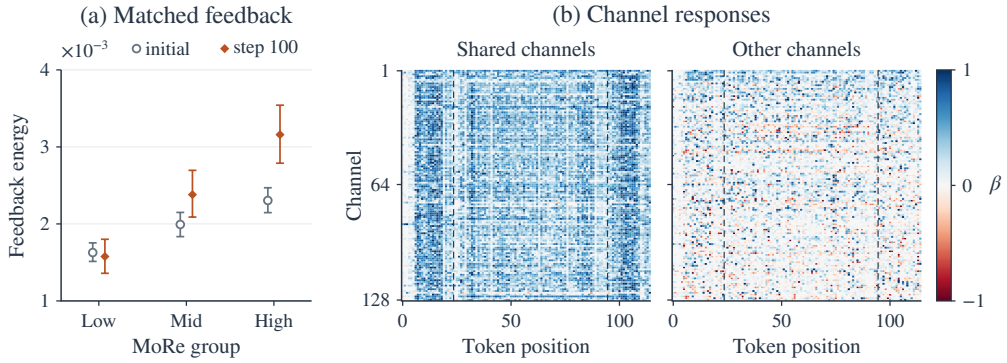


Figure 3: Feedback association and value-response structure. (a) Feedback energy (mean squared cache-path GRPO gradient norm) at initialization and step 100. Groups are matched on observed success count and prompt-length tertile; error bars indicate 95% bootstrap confidence intervals. (b) V responses along the common direction at step 200 for one of the 64 evaluation prompts. Left: the 128 shared channels; right: 128 channels taken at equal rank spacing from the remainder. Colors show the coefficient β, with the sign of each channel chosen on the selection prompts and each row normalized by its maximum. Dashed lines separate the system prompt, the question, and the template suffix.

2.3 How Attention Shapes Gradient Feedback

The cache gradients in Equation (4) depend on attention readout. We examine how V responses combine across prompt positions and how K-mediated changes in attention weights contribute to the matched feedback gap.

Coherent value responses and shared channels. Each channel acts through the same down-projection column at every prompt position, giving its V responses a common projected component and an RMSNorm correction. Coefficients of the same sign reinforce the common component under attention. For the full V responses at a fixed answer position, let E be the squared norm of their attention-weighted sum and D the sum of the squared norms of its terms. Their difference, $C = E - D$, collects cross-position inner products: positive contributions reinforce and negative contributions cancel (Appendix A.4).

We select 128 channels, about 2.6% of the FFN, by their positive contributions to C at step 200. They account for roughly 60% of C on two prompt sets excluded from selection (Appendix C.2), with coherent sign bands along the token axis (Figure 3(b)). The same channels carry about 36% of feedback energy at initialization and 44% at step 100, including both K and V paths despite selection from V responses alone. Their coherent response extends across more prompt positions during training (Figure 6).

Attention redistribution and the feedback gap. K accounts for about a fifth of MoRe, whereas its share of the sum of K-only and V-only means is roughly three fifths for both Fisher information and feedback energy (Figure 4a). Fisher measures local answer-distribution sensitivity along the cache path (Appendix A.3); its response becomes increasingly concentrated at a small fraction of answer positions during training (Figure 4b). The larger K share after readout motivates testing how attention weights contribute to the feedback gap.

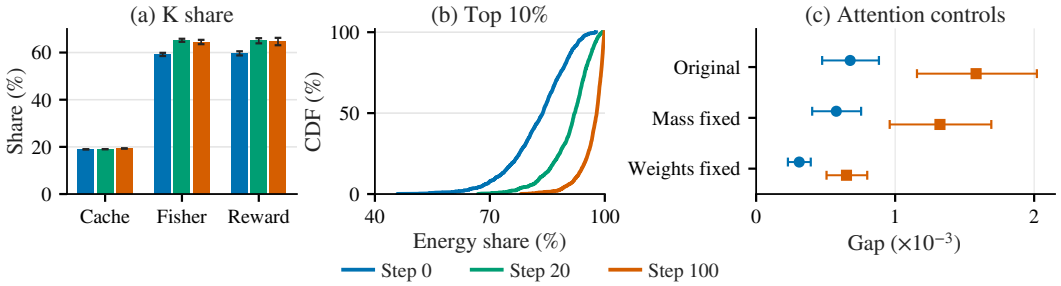


Figure 4: Cache responses and downstream feedback. (a) K's share of the sum of the K-only and V-only means for the MoRe score, per-token Fisher information, and squared GRPO gradient norm at steps 0, 20, and 100. (b) Distribution across prompts of the Fisher share carried by the 10% most sensitive answer positions at the same checkpoints. (c) Matched differences in mean squared cache-path gradient norm between high- and low-MoRe groups under original attention, fixed prompt mass, and fixed within-prompt proportions; 1,022 prompts at step 0 and 1,024 at step 100. Error bars show 95% bootstrap intervals, paired across controls in (c).

A common shift of the rotated prompt keys can change only the prompt's total attention mass; differences across positions also redistribute it (Appendix A.6). We separate these effects by interventions in the manner of activation patching (Heimersheim & Nanda, 2024). "Mass fixed" preserves the baseline total while allowing relative proportions to change. "Weights fixed" preserves those proportions while allowing the total to change. With the same answers, rewards, MoRe scores, and unperturbed forward pass, we recompute the gradient under each intervention. The feedback gap is the difference in mean squared cache-path gradient norm between the matched high- and low-MoRe groups.

Fixing the total mass leaves most of the high–low gap, whereas fixing within-prompt proportions removes more than half of it at both checkpoints (Figure 4c). At step 100, the latter control leaves 41.1% of the original gap. These interventions identify within-prompt redistribution as a major contributor to the matched feedback gap.

3 SAMPLING WITH MORE

The comparisons of feedback energy in Section 2.2 motivate applying current-actor MoRe within a pool selected by MoPPS. The first stage uses reward beliefs to retain moderately difficult prompts; the second allocates a fixed number of rollout slots within this pool using prefill MoRe scores and a correction for finite-group advantage scaling. Both decisions precede answer generation.

3.1 PREFILTERING WITH MOPPS

We adopt the posterior-sampling rule of MoPPS (Qu et al., 2026a) as our first-stage prefilter. From N training prompts, draw a candidate set $\mathcal{U}$ of size $\min(N, 16B)$ uniformly without replacement, where B is the final prompt budget. We maintain a $\mathrm{Beta}(a_i, b_i)$ belief over each candidate's success probability p_i. Independently sample a success rate and compute its squared distance from the target $1/2$:

$$\widetilde{p}_i \sim \mathrm{Beta}(a_i, b_i), \qquad d_i = (\widetilde{p}_i - \tfrac{1}{2})^2, \qquad i \in \mathcal{U}. \tag{7}$$

Retain the $H = \min(4B, |\mathcal{U}|)$ candidates with the smallest d_i to form $\mathcal{P}$. This uses the MoPPS selection criterion to produce an intermediate pool of up to $4B$ prompts, leaving the final budget allocation to MoRe. Targeting intermediate success rates favors reward contrast, while posterior sampling lets uncertainty in the reward beliefs affect which prompts enter the pool.

3.2 ALLOCATING THE GENERATION BUDGET

Within the retained pool, reward normalization itself reweights prompts: binary GRPO induces $\mathbb{E}[\widehat{g}_i] = \kappa_G(p_i)\nabla_\theta p_i$ (Davis & Recht, 2025). Let $\bar{\kappa}_i$ be its Beta-belief average, derived in Appendix A.7. We account for this multiplier when assigning the remaining preference to the MoRe score. For $|\mathcal{P}| > B$, compute current-actor MoRe scores $\mathcal{M}_i$ and set

$$\pi_i(\mathcal{P}) = \min\left\{1, c_{\mathcal{P}} \frac{\mathcal{M}_i}{\bar{\kappa}_i}\right\}, \qquad \sum_{i \in \mathcal{P}} \pi_i(\mathcal{P}) = B. \tag{8}$$

Uncapped prompts then satisfy $\pi_i \bar{\kappa}_i = c_{\mathcal{P}} \mathcal{M}_i$: their inclusion probabilities compensate the belief-averaged normalization scale and retain a preference proportional to the prefill response. Dependent rounding (Gandhi et al., 2006) preserves these conditional marginals and selects exactly B prompts. If $|\mathcal{P}| = B$, all are selected without MoRe scoring. Only the selected prompts receive rollouts, with G answers per prompt.

We use $\mathcal{M}_K + \mathcal{M}_V$ for allocation. Although K and V differ in feedback contribution, their MoRe scores rank the diagnostic prompts similarly, with Spearman correlations above 0.99 at steps 0, 100, and 200. At step 100, their narrow ratio range means that using K or V alone changes conditional inclusion probabilities by less than 5% relative to their sum, with the retained pool, budget, and advantage-scale correction fixed (Appendix A.8). This limited effect supports the unweighted sum on this diagnostic pool.

3.3 UPDATING REWARD BELIEFS

We maintain the reward beliefs with the discounted Beta–Bernoulli update of MoPPS (Qu et al., 2026a), initialized at $a_i = b_i = 1$. After the actor update, incorporate each selected prompt's K_i successes among G answers:

$$a_i \leftarrow 1 + \rho(a_i - 1) + K_i, \qquad b_i \leftarrow 1 + \rho(b_i - 1) + G - K_i, \tag{9}$$

where $\rho = \exp[-B/(2N)]$. Old evidence is discounted only when the prompt is selected; unselected beliefs are unchanged.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets and models. We evaluate prompt selection on Countdown and mathematical reasoning. Countdown requires combining three or four given integers with arithmetic operations to reach

a target, using each integer exactly once. We train on 20,000 Countdown-34 prompts and evaluate on 1,024 prompts, split equally between Countdown-34 and Countdown-4. For mathematical reasoning, we train on the 40,311 prompts in DeepScaleR (Luo et al., 2025) and evaluate on MATH500 (Hendrycks et al., 2021; Lightman et al., 2024), AIME 2024/2025, AMC 2023, Minerva Math (Lewkowycz et al., 2022), and OlympiadBench (He et al., 2024). We use Qwen3-0.6B and Qwen3-4B (Yang et al., 2025) for both tasks, and additionally Qwen2.5-Math-1.5B (Yang et al., 2024) for DeepScaleR.

Training protocol. All methods use GRPO in verl (Sheng et al., 2025) with $B = 256$ prompts per training batch and $G = 8$ responses per prompt. We use AdamW (Loshchilov & Hutter, 2019) with a constant learning rate of 10^{-6}, weight decay 0.01, and one PPO epoch per batch. Countdown and DeepScaleR use PPO minibatches of 64 and 128 prompts, with training budgets of 120 and 200 steps, respectively. Their maximum prompt/response lengths are 256/1,024 and 1,024/4,096 tokens. Training responses are sampled at temperature 1.0 and top-p 1.0, with Qwen3 thinking mode disabled. We use binary correctness rewards, no KL or entropy regularization, and seed 42.

Evaluation and reporting. We evaluate every 10 training steps on Countdown and every five steps on DeepScaleR, using temperature 0.6 and top-p 0.95. Avg@k denotes average correctness over k sampled responses per problem. We report Avg@16 for Countdown, AIME 2024/2025, and AMC 2023; Avg@4 for Minerva Math; and Avg@1 for MATH500 and OlympiadBench. Countdown accuracy averages its two evaluation subsets equally. For Table 1, we select the evaluated checkpoint with the highest unweighted mean across the six mathematical benchmarks and report every subset at that same checkpoint. Rollout tokens and GPU-hours accumulate to the selected checkpoint. Token counts include all DS resampling. GPU-hours are the cumulative logged training-step duration in hours multiplied by the number of H200 GPUs used by the run, excluding validation.

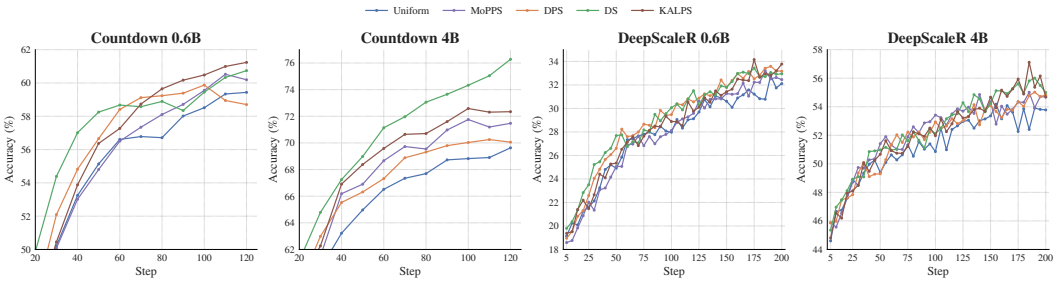


Figure 5: Validation accuracy (%) over training steps, comparing Uniform, MoPPS, DPS, DS, and KALPS (seed 42). From left to right: Countdown 0.6B, Countdown 4B, DeepScaleR 0.6B, and DeepScaleR 4B. Countdown reports mean@16; DeepScaleR reports mean@1.

Baselines. We compare with four baselines: **Uniform**, which samples prompts uniformly; **DS** (Yu et al., 2025), which resamples answer groups to prioritize prompts with mixed rewards; **MoPPS** (Qu et al., 2026a), which uses Beta–Bernoulli reward beliefs and posterior sampling to select moderately difficult prompts; and **DPS** (Mao et al., 2026), which uses a hidden Markov model to predict prompt-solving states and prioritize partially solved prompts. DS allows at most four generation batches per training step. MoPPS and DPS draw candidate pools of up to $16B$ and $32B$ prompts, respectively, capped by the training-set size.

4.2 Main Results

Figure 5 shows validation accuracy over training steps for Qwen3-0.6B and Qwen3-4B trained on Countdown and DeepScaleR.

Table 1: Average pass@1 (%) on six mathematical reasoning benchmarks after training on DeepScaleR (seed 42). Each row reports all benchmarks at the checkpoint maximizing their unweighted mean (Avg.). Costs accumulate to that checkpoint. GPU-hours equal cumulative logged training-step hours multiplied by the number of H200 GPUs, excluding validation. Results use the logs available on September 26, 2026.

	Accuracy (%) ↑							**Training cost ↓**	
Method	AIME 2024 Avg@16	AIME 2025 Avg@16	AMC 2023 Avg@16	MATH 500 Avg@1	Minerva Math Avg@4	Olympiad Bench Avg@1	**Avg.**	Rollout tokens (M)	GPU hours
				Qwen3-0.6B					
Uniform	8.96	15.21	45.47	70.00	17.28	39.90	32.80	925.16	29.79
DS	9.75	16.18	46.94	69.20	18.11	40.41	33.43	2306.54	61.08
MoPPS	**10.21**	**16.46**	45.63	68.80	18.01	40.24	33.22	845.94	27.23
DPS	6.67	15.63	**47.50**	69.40	**19.30**	43.01	33.58	833.09	26.02
Ours	9.04	15.54	47.38	**70.40**	19.21	**43.40**	**34.16**	814.30	34.12
				Qwen2.5-Math-1.5B					
Uniform	15.42	8.96	54.69	68.80	20.86	36.10	34.14	344.89	16.39
DS	17.50	10.00	54.06	68.40	20.68	36.44	34.51	559.82	17.53
MoPPS	15.21	7.71	51.56	**72.00**	22.61	**37.48**	34.43	335.28	31.18
DPS	**18.33**	7.92	54.38	70.80	**23.07**	36.10	35.10	355.84	31.84
Ours	17.63	**10.79**	**55.06**	69.80	22.88	37.44	**35.60**	372.95	15.82
				Qwen3-4B					
Uniform	39.38	28.54	76.41	85.20	33.36	62.18	54.18	579.33	90.28
DS	41.04	33.13	79.84	85.40	34.47	62.18	56.01	2440.10	178.61
MoPPS	34.79	32.71	80.63	85.60	34.19	62.18	55.02	737.87	58.27
DPS	35.83	31.04	81.41	85.40	33.09	63.39	55.03	853.42	116.58
Ours	**41.54**	**33.92**	**81.58**	**86.80**	**35.01**	**63.79**	**57.11**	770.15	61.35

Effect of score ordering. We reverse the MoRe-score ordering for prompt selection and train Qwen3-0.6B and Qwen3-4B on Countdown. Compared with the standard MoRe-guided sampler, this ablation lowers validation accuracy by 1.8 percentage points on average across the two models. This comparison supports prioritizing higher MoRe scores in these settings.

5 Conclusion

Reward history and computational response describe different aspects of an RLVR prompt. We isolate gate modulation that preserves bilinear interactions, derive its prefill MoRe score, and trace the same Jacobian into reward feedback. Matched comparisons reveal differences in feedback energy beyond observed success rate and length. The clearer energy separation obtained with current scores motivates using current-actor MoRe alongside reward beliefs. The attention analysis identifies coherent V responses in shared channels and the contribution of within-prompt redistribution to the matched feedback gap. In the sampler, reward beliefs determine the candidate pool and current-model sensitivity determines the allocation within it. The central design principle is to combine the availability of reward contrast with the computation that transmits it, while keeping both selection decisions ahead of rollout.

AI use statement

We used generative AI tools to assist with experimental design, literature search, and language editing. We reviewed and refined the suggested experimental designs and selected the final settings. We consulted the original papers to verify the references and their support for the cited claims, and carefully reviewed the edited text to ensure that it preserved the intended meaning. We take responsibility for the final manuscript, including all claims, results, and accompanying artifacts.

Ethics statement

This work studies prompt selection for reinforcement learning with verifiable rewards using existing language models and mathematical reasoning benchmarks. The experiments use automated reward verification and do not involve human-participant studies or the collection of personal data. We do not identify specific ethical concerns arising from the proposed prompt-selection method.

References

Yong Yi Bay and Kathleen A. Yearick. GRPO, Dr. GRPO, and DAPO are three operations on one number: The group-standard-deviation identity, 2026. URL https://arxiv.org/abs/2607.00152.

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

Damek Davis and Benjamin Recht. What is the objective of reasoning with reinforcement learning? *arXiv preprint arXiv:2510.13651*, 2025. URL https://arxiv.org/abs/2510.13651.

Rajiv Gandhi, Samir Khuller, Srinivasan Parthasarathy, and Aravind Srinivasan. Dependent rounding and its applications to approximation algorithms. *Journal of the ACM*, 53(3):324–360, 2006. doi: 10.1145/1147954.1147956. URL https://www.cs.umd.edu/~srin/PDF/2006/depround-jou.pdf.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025. doi: 10.1038/s41586-025-09422-z. URL https://doi.org/10.1038/s41586-025-09422-z.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.211. URL https://aclanthology.org/2024.acl-long.211/.

Stefan Heimersheim and Neel Nanda. How to use and interpret activation patching, 2024. URL https://arxiv.org/abs/2404.15255.

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 *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

Angelos Katharopoulos and Francois Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2525–2534. PMLR, 2018. URL https://proceedings.mlr.press/v80/katharopoulos18a.html.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 3843–3857. Curran Associates, Inc., 2022. doi: 10.52202/068431-0278. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/18abbeef8cfe9203fdf9053c9c4fe191-Abstract.html.

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*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/aca97732e30bcf1303bc22ac3924fd16-Paper-Conference.pdf.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Erran Li, Raluca Ada Popa, and Ion Stoica. DeepScaleR: Surpassing O1-Preview with a 1.5b model by scaling RL, 2025. URL https://huggingface.co/datasets/agentica-org/DeepScaleR-Preview-Dataset.

Yixiu Mao, Yun Qu, Qi Wang, Heming Zou, and Xiangyang Ji. Dynamics-predictive sampling for active RL finetuning of large reasoning models. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/2ae33575c3374050654ae7802326c81d-Abstract-Conference.html.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. In *Advances in Neural Information Processing Systems*, volume 34, pp. 20596–20607. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/ac56f8fe9eea3e4a365f29f0f1957c55-Paper.pdf.

Michael T. Pearce, Thomas Dooms, Alice Rigg, Jose M. Oramas, and Lee Sharkey. Bilinear MLPs enable weight-based mechanistic interpretability. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/7504142a20a3e1fe9dd7de42f475828c-Abstract-Conference.html.

Yun Qu, Qi Wang, Yixiu Mao, Vincent Tao Hu, Björn Ommer, and Xiangyang Ji. Can prompt difficulty be online predicted for accelerating RL finetuning of reasoning models? In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, pp. 1240–1250. ACM, 2026a. doi: 10.1145/3770854.3780263. URL https://doi.org/10.1145/3770854.3780263.

Yun Qu, Qi Wang, Yixiu Mao, Heming Zou, Yuhang Jiang, Weijie Liu, Clive Bai, Kai Yang, Yangkun Chen, Saiyong Yang, and Xiangyang Ji. Small generalizable prompt predictive models can steer efficient RL post-training of large reasoning models. In *International Conference on Machine Learning*, 2026b. URL https://openreview.net/forum?id=vCHOBek0dv.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Emilio Salinas and Peter Thier. Gain modulation: A major computational principle of the central nervous system. *Neuron*, 27(1):15–21, 2000. doi: 10.1016/S0896-6273(00)00004-0. URL https://doi.org/10.1016/S0896-6273(00)00004-0.

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. URL https://arxiv.org/abs/2402.03300.

Noam Shazeer. GLU variants improve transformer, 2020. URL https://arxiv.org/abs/2002.05202.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A flexible and efficient RLHF framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297. ACM, 2025. doi: 10.1145/3689031.3696075. URL https://doi.org/10.1145/3689031.3696075.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding, 2021. URL https://arxiv.org/abs/2104.09864.

Qinsi Wang, Jinghan Ke, Hancheng Ye, Yueqian Lin, Yuzhe Fu, Jianyi Zhang, Kurt Keutzer, Chenfeng Xu, and Yiran Chen. Angles don't lie: Unlocking training-efficient RL through the model's own signals. In *Advances in Neural Information Processing Systems*, volume 38, pp. 87119–87149. Curran Associates, Inc., 2025. doi: 10.52202/085713-2914. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/7e0af0d1bc0ec2a90fc294be2e00447e-Abstract-Conference.html.

Jiahao Wu, Ning Lu, Shengcai Liu, Kun Wang, Yanting Yang, Bailong Lin, Chen Jason Zhang, Qing Li, and Ke Tang. Train at the moving edge: Efficient RL for large reasoning models via rollout selection, 2026. URL https://arxiv.org/abs/2603.25184.

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

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Ningyuan Yang, Weihua Du, Weiwei Sun, Sean Welleck, and Yiming Yang. GradAlign: Gradient-aligned data selection for LLM reinforcement learning. In *Third Conference on Language Modeling*, 2026. URL https://openreview.net/forum?id=bGUY6CJSbu.

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, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*, volume 38, pp. 113222–113244. Curran Associates, Inc., 2025. doi: 10.52202/085713-3775. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/a4277440d50f1f15d2cb4c14f7e0c0d2-Abstract-Conference.html.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/1e8a19426224ca89e83cef47f1e7f53b-Abstract.html.

Haodong Zhu, Yangyang Ren, Yanjing Li, Sheng Xu, Haiguang Liu, Linlin Yang, and Baochang Zhang. Kalman meets curriculum: Efficient dynamic prompt selection for adaptive RL finetuning. *arXiv preprint arXiv:2607.27610*, 2026. URL https://arxiv.org/abs/2607.27610.

A Theoretical Details

A.1 Gate Coordinates and Cache Responses

At fixed FFN input, reciprocal scaling leaves $g_{tk}u_{tk}$ constant. Differentiating its sigmoid gain gives the source response

$$z_{tk} = \partial_{\alpha_k}[(g_{tk}u_{tk})\sigma(e^{\alpha_k/s_k}g_{tk})]\Big|_0 = \frac{u_{tk}g_{tk}^2\sigma'(g_{tk})}{s_k}. \tag{10}$$

The parameter tangent has nonzero rows $(w_{g,k}, -w_{u,k})/s_k$, hence unit norm; different channels occupy disjoint rows.

Let y_t be the post-FFN residual of dimension d. Write the next RMSNorm as $\Gamma y_t/\rho_t$, with $\rho_t^2 = \|y_t\|^2/d + \epsilon_N$, and stack the projections as $W_{\mathrm{mem}} = [W_K; W_V]$. Then

$$L_t = W_{\mathrm{mem}}\Gamma\left(\frac{I}{\rho_t} - \frac{y_t y_t^\top}{d\rho_t^3}\right), \qquad j_{tk} = L_t d_k z_{tk}, \qquad Q = \frac{1}{T}\sum_{t,k}\|j_{tk}\|^2. \tag{11}$$

Each KV head is counted once, before replication across query heads. RoPE (Su et al., 2021) at a given position is orthogonal and preserves these norms. With $p_k^{\mathrm{mem}} = W_{\mathrm{mem}}\Gamma d_k$, $v_t^{\mathrm{mem}} = W_{\mathrm{mem}}\Gamma y_t$, and $\chi_{tk} = y_t^\top d_k$, expanding the square gives

$$\|L_t d_k\|^2 = \frac{\|p_k^{\mathrm{mem}}\|^2}{\rho_t^2} - \frac{2\chi_{tk}(p_k^{\mathrm{mem}})^\top v_t^{\mathrm{mem}}}{d\rho_t^4} + \frac{\chi_{tk}^2\|v_t^{\mathrm{mem}}\|^2}{d^2\rho_t^6}. \tag{12}$$

Multiplying by z_{tk}^2 and summing evaluates Q from prefill activations and weights without constructing the full Jacobian.

A.2 Content-Preserving Gate Directions

For a bias-free channel, let $a = w_g \neq 0$, $b = w_u \neq 0$, and $\theta = (a, b)$. With the input h and down-projection column fixed, write $g = a^\top h$, $u = b^\top h$, $\phi_h = gu$, and $\gamma_h = \sigma(g)$, so the channel output is $f_h = \phi_h\gamma_h$. Under the Euclidean parameter metric, reciprocal scaling preserves the bilinear matrix $B = ab^\top$ and has unit tangent

$$B(\alpha) = e^{\alpha/s}a(e^{-\alpha/s}b)^\top = B, \qquad v = \frac{1}{s}\begin{pmatrix} a \\ -b \end{pmatrix}, \qquad s^2 = \|a\|^2 + \|b\|^2. \tag{13}$$

When a and b are linearly independent, this is also the unique direction, up to scale, that preserves the product function for every input:

$$D\phi_h[\delta a, \delta b] = 0 \text{ for every } h \quad \Longleftrightarrow \quad (\delta a, \delta b) = (ca, -cb) \text{ for some } c. \tag{14}$$

To prove this, decompose each perturbation into $\mathrm{span}\{a, b\}$ and its orthogonal complement. Varying the orthogonal component of h forces both perturbations' orthogonal components to vanish, since $a^\top h$ and $b^\top h$ can vary independently. Write $\delta a = xa + yb$ and $\delta b = za + wb$. For $A = a^\top h$ and $U = b^\top h$, the derivative is $zA^2 + (x + w)AU + yU^2$, which vanishes identically only if $z = y = 0$ and $w = -x$. The converse follows by substitution. The constraint concerns all inputs; a finite set of prompt positions can admit additional directions.

Preserving B implies preserving every ϕ_h, and v preserves B. Thus, under the same noncollinearity condition, $J_B = \partial\,\mathrm{vec}(B)/\partial\theta$ has $\ker J_B = \mathrm{span}\{v\}$. For $P_g = vv^\top$, the product rule then gives

$$v^\top\nabla_\theta\phi_h = 0, \qquad P_g\nabla_\theta f_h = \phi_h P_g\nabla_\theta\gamma_h, \qquad v^\top\nabla_\theta f_h = \frac{ug^2\sigma'(g)}{s}. \tag{15}$$

The last expression recovers Equation (10). Noncollinearity holds for all 4,864 source-layer channels at steps $0, 10, \ldots, 100, 125, 150, 175, 200$ of the diagnostic model, with maximum absolute gate/up row cosine 0.737.

Across channels, embed the unit tangents as $V = [v_1, \ldots, v_m]$. Their disjoint parameter rows give $V^\top V = I$ and the projector $P_g = VV^\top$. For the cache Jacobian J_C in gate/up parameters,

$$\mathcal{G} = J_C V, \qquad TQ = \|J_C V\|_F^2 = \|J_C P_g\|_F^2 = \mathrm{tr}(J_C P_g J_C^\top). \tag{16}$$

At a fixed actor, V is shared across prompts, while J_C and prompt length T vary. Writing $g_\theta^{\mathrm{cache}} = J_C^\top e$ also yields

$$\widehat{g}^{\mathrm{cache}} = V^\top g_\theta^{\mathrm{cache}}, \qquad \|\widehat{g}^{\mathrm{cache}}\|^2 = \|P_g g_\theta^{\mathrm{cache}}\|^2. \tag{17}$$

This identifies the gain contribution measured by the cache-path probe. Other parameter directions can also change the gain, and orthogonal parameter directions need not produce orthogonal cache responses.

A.3 Reward Feedback Along the Cache Path

Only the final layer's prompt KV cache varies with α; queries, residual inputs, and generated-position caches take their baseline values for each prefix. For G independent on-policy answers, let $K = \sum_j R_j$ and $A_j = (R_j - K/G)/\delta_K$, with $\delta_k = \sqrt{k(G-k)/(G(G-1))} + 10^{-6}$. Treating advantages as constants gives $\widehat{g}^{\mathrm{cache}} = \mathcal{G}^\top e$. Taking its second moment at a fixed prompt and actor yields Equation (4); substituting the isotropic reference second moment gives Equation (5).

For the forward response, let r have independent, equally likely ± 1 entries and set $\alpha = \varepsilon r$. Since $\mathbb{E}[rr^\top] = I$, the local response $\delta\mathbf{C} \simeq \varepsilon \mathcal{G} r$ satisfies

$$\mathbb{E}_r \|\delta\mathbf{C}\|^2 \simeq \varepsilon^2 \|\mathcal{G}\|_F^2 = \varepsilon^2 T\mathcal{M}. \tag{18}$$

This relation holds to leading order in ε. Together with Equation (5), it gives the same mean squared response $\mathcal{M}$ in both directions after normalization by per-coordinate input second-moment scale and prompt length.

The normalization in Equation (5) uses the per-coordinate second-moment scale and prompt length. With $d_C = \dim \mathbf{C}$ and $\sigma_e^2 > 0$, the corresponding ratio of mean output to input energies for the reference feedback is

$$\frac{\mathbb{E}_\xi \|\mathcal{G}^\top \xi\|^2}{\mathbb{E}_\xi \|\xi\|^2} = \frac{\|\mathcal{G}\|_F^2}{d_C} = \frac{T\mathcal{M}}{d_C}.$$

For a fixed architecture, d_C/T is the cache dimension per prompt token.

For arbitrary feedback, the deterministic norm bound also holds:

$$\|\widehat{g}^{\mathrm{cache}}\|^2 = e^\top \mathcal{G}\mathcal{G}^\top e \le \|\mathcal{G}\|_2^2 \|e\|^2 \le T\mathcal{M}\,\|e\|^2.$$

Fix $0 < p = \Pr(R = 1) < 1$ and let $s(Y) = \nabla_\alpha \log P_\alpha(Y \mid x)|_0$. Denote its correct/incorrect conditional means and covariances by $\mu_\pm, \Sigma_\pm$, and put $\Delta\mu = \mu_+ - \mu_-$. Equation (6) follows by collecting the correct and incorrect terms in the reward-weighted average; set $c_0 = c_G = 0$ for identical-reward groups. Conditional on $K = k$, the two sample means have means $\mu_\pm$ and covariances Σ_+/k and $\Sigma_-/(G-k)$. With expectations over $K \sim \mathrm{Binomial}(G, p)$, define $m_G = \mathbb{E}[c_K]$, $b_G = \mathbb{E}[c_K^2]$, and

$$a_G^+ = \sum_{k=1}^{G-1} \Pr(K = k) \frac{c_k^2}{k}, \qquad a_G^- = \sum_{k=1}^{G-1} \Pr(K = k) \frac{c_k^2}{G-k}.$$

Taking the conditional moments and averaging gives

$$\begin{aligned} \mathbb{E}[\widehat{g}^{\mathrm{cache}}] &= m_G \Delta\mu, \\ \mathbb{E}\|\widehat{g}^{\mathrm{cache}}\|^2 &= b_G \|\Delta\mu\|^2 + a_G^+ \operatorname{tr}\Sigma_+ + a_G^- \operatorname{tr}\Sigma_-. \end{aligned} \tag{19}$$

Success probability fixes the coefficients, but the score contrast and conditional variation still depend on the model through $s = \mathcal{G}^\top \nabla_{\mathbf{C}} \log P(Y \mid x)$. The squared mean feedback is $m_G^2 \|\Delta\mu\|^2$; subtracting it from the second moment gives the trace of the sampling covariance. Appendix C.1 measures these two quantities separately.

At answer position r, let p_r be the vocabulary distribution and D_r the logit Jacobian with respect to α. The Fisher response satisfies

$$F_r = \operatorname{tr}\left[D_r^\top (\operatorname{diag} p_r - p_r p_r^\top) D_r\right], \qquad \mathbb{E}_Y \|\nabla_\alpha \log P_\alpha(Y \mid x)|_0\|^2 = \mathbb{E}_Y \sum_r F_r. \tag{20}$$

Each on-policy token score has zero conditional mean, so cross-position terms vanish in this expectation. Joint K/V Fisher still includes the cross term between the two paths.

A.4 Shared Responses and Cross-Position Interactions

At a fixed answer position, let a_{ht} be the attention to prompt token t and j^V_{htk} the V response mapped to query head h. Before the output projection,

$$E = \sum_{h,k} \Big\| \sum_t a_{ht} j^V_{htk} \Big\|^2, \qquad D = \sum_{h,t,k} a_{ht}^2 \|j^V_{htk}\|^2,$$
$$C = E - D = 2 \sum_{h,k} \sum_{t<t'} a_{ht} a_{ht'} (j^V_{htk})^\top j^V_{ht'k}. \tag{21}$$

The common-direction term of Equation (11) is $p_k \beta_{tk}$, where $p_k = W_V \Gamma d_k$ and $\beta_{tk} = z_{tk}/\rho_t$; the remaining term is the RMSNorm radial correction. For this common term, put $w_{htk} = a_{ht}\beta_{tk}$ and define, on a channel set $\mathcal{S}$,

$$D^{(0)}_{\mathcal{S}} = \sum_{h,k\in\mathcal{S}} \|p_{hk}\|^2 \sum_t w_{htk}^2, \qquad E^{(0)}_{\mathcal{S}} = \sum_{h,k\in\mathcal{S}} \|p_{hk}\|^2 \Big(\sum_t w_{htk}\Big)^2,$$
$$E_{\mathrm{abs},\mathcal{S}} = \sum_{h,k\in\mathcal{S}} \|p_{hk}\|^2 \Big(\sum_t |w_{htk}|\Big)^2, \qquad U_{\mathcal{S}} = \frac{E_{\mathrm{abs},\mathcal{S}}}{D^{(0)}_{\mathcal{S}}}, \quad R_{\mathcal{S}} = \frac{E^{(0)}_{\mathcal{S}}}{E_{\mathrm{abs},\mathcal{S}}}. \tag{22}$$

Here p_{hk} is the direction read by head h, and $E^{(0)}_{\mathcal{S}}/D^{(0)}_{\mathcal{S}} = U_{\mathcal{S}} R_{\mathcal{S}}$. Comparisons across checkpoints use the common scale $s_k^{(200)}$, whereas Q and the gradient at a single checkpoint use that model's own s_k. With magnitudes and attention unchanged, changing the signs of the up preactivations preserves Q and D and changes C through the inner products above.

A.5 Attention Mass and Within-Prompt Redistribution

For one head, let $m = \sum_t a_t v_t$ and let P contain the prompt positions. Define $w_P = \sum_{t\in P} a_t$, $m_P = w_P^{-1} \sum_{t\in P} a_t v_t$, and $\overline{\delta s}_P = w_P^{-1} \sum_{t\in P} a_t \delta s_t$, with $\delta s_t = q^\top \delta k_t/\sqrt{d_h}$. The softmax derivative gives $\delta m_K = \sum_{t\in P} a_t \delta s_t (v_t - m)$. Centering scores and values within the prompt yields

$$\delta m_K = \underbrace{w_P \overline{\delta s}_P (m_P - m)}_{\text{prompt attention mass}} + \underbrace{\sum_{t\in P} a_t (\delta s_t - \overline{\delta s}_P)(v_t - m_P)}_{\text{within-prompt redistribution}}. \tag{23}$$

The two interventions constrain either the total mass on the prompt or the within-prompt proportions to their unperturbed values, with the same V perturbation and cache Jacobian. At the first answer position, $w_P = 1$ and the mass term vanishes.

A.6 Spatial Modes of the Cache Response

A common rotated-key increment makes δs_t constant within the prompt, eliminating the redistribution term in Equation (23). A common value increment instead gives $\delta m_V = w_P \delta v$. We separate position means from differences to test their readout at equal cache energy.

For each branch separately, let J_t be its cache Jacobian block at input position t, using rotated K and native V coordinates. Decompose

$$J_t = \bar{J} + (J_t - \bar{J}), \qquad Q_{\mathrm{mean}} = \|\bar{J}\|_F^2, \qquad Q_{\mathrm{diff}} = \frac{1}{T} \sum_t \|J_t - \bar{J}\|_F^2, \tag{24}$$

where $\bar{J} = T^{-1} \sum_t J_t$ and $Q = Q_{\mathrm{mean}} + Q_{\mathrm{diff}}$ within the branch. To compare readout at the same branch energy, retain one mode a and multiply its Jacobian by $\sqrt{Q/Q_a}$. Its Fisher response becomes $\widetilde{F}_a = (Q/Q_a) F_a$. The two cache modes are orthogonal, but their readout responses can have nonzero cross terms, so their separate Fisher values are not additive shares.

We apply this control on 256 held-out prompts at steps 0, 100, and 200 of the same diagnostic trajectory. Each prompt is read at the position with highest next-token entropy in its saved natural

answer. This selects the position without using its cache Fisher response. Table 2 reports step 100; the ordering between the two modes within each branch is the same at steps 0 and 200.

Table 2: Readout after matching each mode to its branch's original Q, on 256 prompts at step 100. Values divide the mean matched Fisher by the mean original branch Fisher; intervals use prompt bootstrap.

Branch	Retained spatial mode	Relative Fisher	95% interval
K	Position mean	0.61	[0.39, 0.89]
K	Position differences	1.35	[1.14, 1.51]
V	Position mean	2.15	[1.96, 2.37]
V	Position differences	0.78	[0.74, 0.83]

A.7 Finite-Group Scaling

For G independent on-policy answers with $0 < p < 1$, the conditional-score argument above also applies to $s_j = \nabla_\theta \log P_\theta(Y_j \mid x)$. Normalization of the policy gives $\mathbb{E}[s \mid R = 1] = \nabla p/p$ and $\mathbb{E}[s \mid R = 0] = -\nabla p/(1-p)$, so $\Delta\mu = \nabla p/[p(1-p)]$. Substituting this into the conditional mean from Equation (6) and using $k(G-k)\binom{G}{k} = G(G-1)\binom{G-2}{k-1}$ yields

$$\mathbb{E}[\widehat{g}] = \kappa_G(p)\nabla_\theta p, \qquad \kappa_G(p) = \frac{G-1}{G}\sum_{j=0}^{G-2}\binom{G-2}{j}\frac{p^j(1-p)^{G-2-j}}{\delta_{j+1}}. \tag{25}$$

Using $\mathbb{E}[p^u(1-p)^v] = \mathrm{B}(a+u, b+v)/\mathrm{B}(a,b)$ for a Beta belief then yields

$$\bar{\kappa}_i = \frac{G-1}{G}\sum_{j=0}^{G-2}\frac{\binom{G-2}{j}}{\delta_{j+1}}\frac{\mathrm{B}(a_i+j, b_i+G-2-j)}{\mathrm{B}(a_i,b_i)}. \tag{26}$$

A.8 Relative K/V Weighting and Conditional Allocation

K and V differ in readout contribution but rank the 1,024 diagnostic prompts similarly: the Spearman correlations between Q_K and Q_V are 0.995, 0.993, and 0.992 at steps 0, 100, and 200. At step 100, Q_K/Q_V ranges approximately from 0.233 to 0.247. This narrow ratio range limits the effect of relative branch weights on conditional allocation.

Fix a retained pool larger than B, positive $Q_{K,i}, Q_{V,i}$, and $\bar{\kappa}_i$. Replacing $Q_{K,i}+Q_{V,i}$ by $\lambda_K Q_{K,i} + \lambda_V Q_{V,i}$, for nonnegative weights not both zero, multiplies the allocation score by $f(r_i) = (\lambda_K r_i + \lambda_V)/(1+r_i)$, where $r_i = Q_{K,i}/Q_{V,i}$. If $0 < m \le f(r_i) \le M$, monotonicity of the capped normalization gives $c/M \le c' \le c/m$, hence

$$\frac{m}{M}\pi_i \le \pi_i' \le \frac{M}{m}\pi_i. \tag{27}$$

The largest M/m over branch weights occurs when using K or V alone. The measured step-100 ratios give the bound $0.95 \le \pi_i'/\pi_i \le 1.05$, rounded outward, at fixed budget and $\bar{\kappa}_i$ on this pool.

B Measurement Details

B.1 Models, Checkpoints, and Prompt Sets

All measurements in Section 2 use Qwen2.5-0.5B-Instruct and checkpoints from uniform GRPO training on the GSM8K training set, with a constant learning rate of 10^{-6}, 256 prompts and 8 answers per step, and 512-token limits on prompt and answer. The perturbed FFN is in layer 22 and has 4,864 channels; the cache is that of layer 23, the final layer.

For the channel analysis, 128 prompts were split into two sets of 64. Channels were ranked on the first set at step 200 by their positive contribution to C, and all reported shares are computed on the

second set, with a further 256 prompts as a check. Each prompt is read at three answer positions, before answer tokens 1, 32, and $\min(128, L)$ for an answer of length L, and shares are averaged over positions and then over prompts. The selected channels and the step-200 parameter scale are held fixed across checkpoints; the resulting measurements are reported in Appendix C.2.

B.2 Answer Groups and Matched Comparisons

The gradient analysis uses 1,536 prompts drawn at random from the GSM8K training set. The analysis was developed on the first 512, and the gradient results in Sections 2.2 and 2.3 come from the remaining 1,024 unless stated; the 512 were then re-evaluated with the same procedure. At initialization and step 100, each prompt has two independent sets of 16 answers sampled at temperature 1. The first set gives the observed success count. The second set forms two groups of eight; advantages use the binary reward, the sample standard deviation, and $\epsilon = 10^{-6}$, and groups with identical rewards contribute zero. For matched comparisons, strata are defined by success count in the first set and by prompt-length tertile. Within each stratum, prompts are split into tertiles of current Q, and the three Q groups receive the same stratum weights. Strata with fewer than three prompts are dropped, which leaves 1,022 prompts at initialization and 1,024 at step 100. Mean answer lengths for the high- and low-Q groups are 312 and 315 tokens at initialization and 262 and 278 tokens at step 100.

For the initial/current-score comparison at step 100, both groupings use the same observed success counts, prompt-length strata, and feedback measurements. Tertiles are formed separately using initialization or current MoRe scores. We retain strata that contain all three groups under both scores and use their common prompt support with the same stratum weights. This leaves the same 1,024 prompts in the main diagnostic set for both groupings.

B.3 Attention Interventions

Both interventions recompute the gradient with the same answers, rewards, and unperturbed forward pass. In the first, the perturbed attention weights on prompt positions are rescaled so that their total equals the unperturbed total, and the weights on generated positions are rescaled to their unperturbed total. In the second, the unperturbed within-prompt proportions are multiplied by the perturbed total on the prompt, and generated positions keep their unperturbed proportions. Effects are reported as the remaining matched feedback gap divided by the original gap.

K shares divide the mean K-only quantity by the sum of the mean K-only and V-only quantities: about 19% for Q and 59–65% for the Fisher information and the squared gradient norm. The 10% most sensitive answer positions carry on average 82.5% and 96.7% of the Fisher information at initialization and step 100. The shared-channel shares of 35.6% and 44.1% divide the squared gradient norm restricted to the 128 channels' coordinates, averaged over groups and prompts, by the all-channel mean.

B.4 Estimates and Confidence Intervals

On the matched diagnostic prompts, the high-to-low feedback-energy ratios are 1.42 at initialization and 2.00 at step 100, with 95% bootstrap intervals $[1.28, 1.57]$ and $[1.68, 2.41]$. Constraining the total attention mass on the prompt leaves 85.3% and 83.6% of the respective high–low gaps; constraining the within-prompt proportions leaves 45.8% and 41.1%, with paired-bootstrap 95% intervals $[40.0, 53.3]\%$ and $[36.4, 47.3]\%$. Intervals use 4,000 resamples of prompts within each stratum-by-group cell. Ratios are recomputed from the group means in each resample, and the three intervention conditions of a prompt are resampled together.

C Additional Mechanism Results

C.1 Feedback Energy and Mean Feedback

To distinguish feedback energy in Equation (19) from the squared norm of mean feedback, we generate 256 fresh answers for each of 72 prompts at step 100. Selection uses a separate set of 16 answers: prompts with observed success rates between $1/8$ and $7/8$ form 24 triplets matched on

success count and prompt-length stratum, with one prompt from each current-Q tertile. Generation and score measurement use the same FP32 diagnostic model, with batches of 64, temperature 1, top-$p = 1$, no top-k truncation, repetition penalty 1, and a 512-token answer limit. Rewards use the same final-numeric-answer GSM8K verifier as the other feedback measurements.

Each prompt's answers form $J = 32$ independent groups of eight, including identical-reward groups with zero feedback. For group vectors u_j, the sample average of $\|u_j\|^2$ estimates the second moment, while

$$\widehat{S} = \frac{1}{J(J-1)} \sum_{j \neq \ell} u_j^\top u_\ell \tag{28}$$

unbiasedly estimates $\|\mathbb{E}u\|^2$. We retain negative finite-sample estimates and give prompts equal weight. Subtracting $\widehat{S}$ from the estimated second moment gives the estimated trace of the sampling covariance.

Using the mean across prompts in each Q group, the high-to-low ratio is 2.15 for feedback energy and 1.71 for the squared norm of mean feedback, with 95% intervals $[1.59, 2.93]$ and $[0.44, 7.15]$, respectively. Each interval uses 4,000 bootstrap resamples of the 24 matched triplets, with all three Q groups resampled together and the ratio recomputed in each resample.

C.2 Shared Channels and Response Magnitudes

The measures U and R are defined in Equation (22). Using the fixed-channel and fixed-scale protocol in Appendix B.1, Figure 6 tracks checkpoints 0, 50, 100, 150, and 200 on the same 64 evaluation prompts. The effective number of contributing positions U rises from 6.0 to 10.5, the sign-retention ratio R from 94.2% to 97.1%, and the shared channels' share of C from 33.7% to 59.8%. At step 200, these channels account for 33.4% of the summed squared V response and 53.9% of E. Their C share on the further 256 prompts is 59.7% (Figure 7). Figure 8 reports the magnitudes of E, D, and C before taking shares.

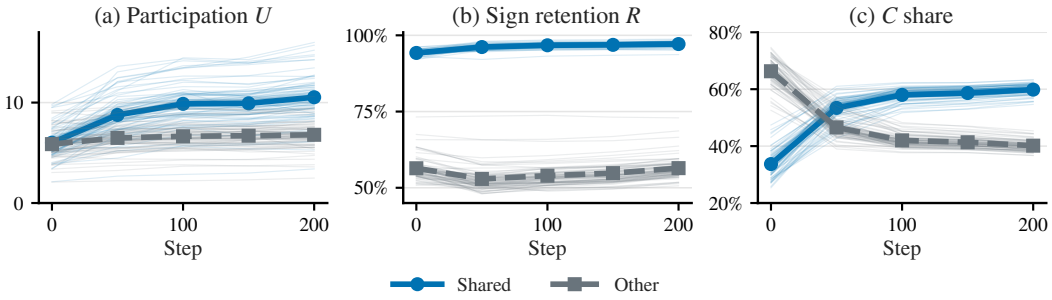


Figure 6: The 128 shared channels across training, on the 64 evaluation prompts. (a) U, the effective number of contributing positions; (b) R, the sign-retention ratio of the common-direction response; (c) the channels' share of C. Thin lines are individual prompts and thick lines are means, with paired-bootstrap 95% bands. All checkpoints use the step-200 parameter scale.

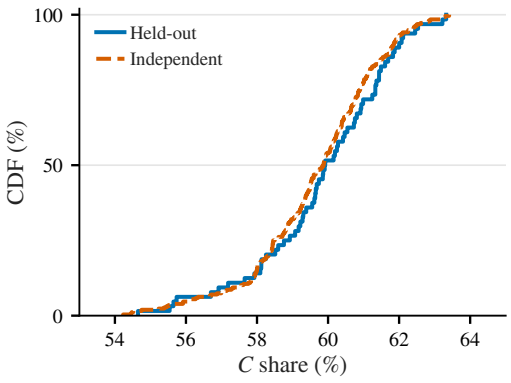


Figure 7: Distribution across prompts of the 128 shared channels' share of C, on the 64 evaluation prompts and on the further 256 prompts.

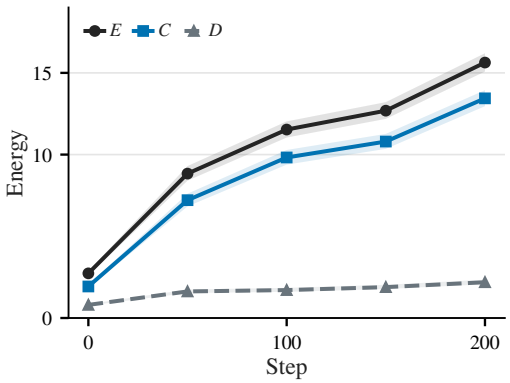


Figure 8: Mean E, D, and C of the 128 shared channels across checkpoints; 95% bootstrap intervals.

C.3 Sign Interventions and Downstream Readout

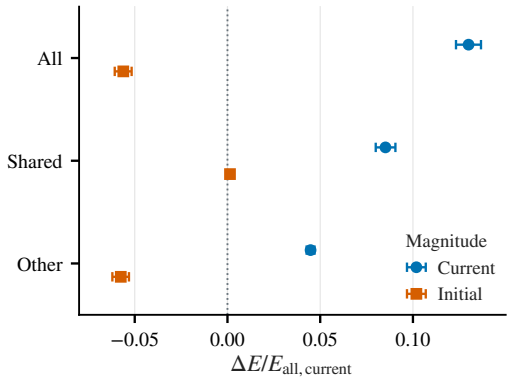


Figure 9: Paired changes in E when the signs of the up preactivations are switched from their initial to their current values, under current magnitudes and under initial magnitudes rescaled to match; 95% bootstrap intervals.

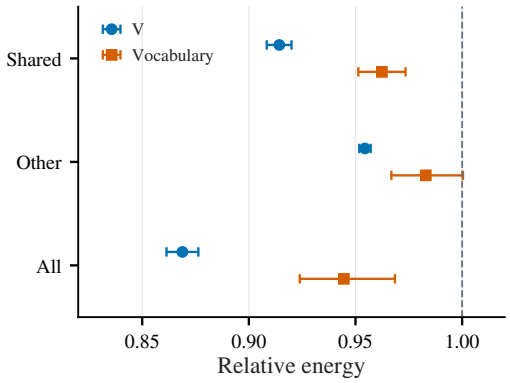


Figure 10: Relative V response and output-distribution sensitivity after the sign interventions; 95% bootstrap intervals.

C.4 Source, Propagation, and Attention Exchanges

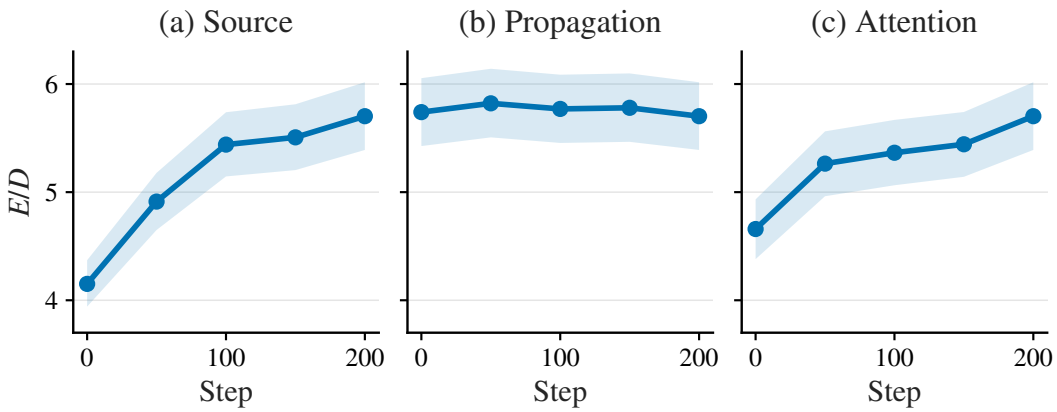


Figure 11: Each panel varies one factor across checkpoints with the other two at their step-200 values. Shading shows 95% bootstrap intervals.

D Case Studies of Gating Interventions

D.1 Gate Effects across Ten Prompts

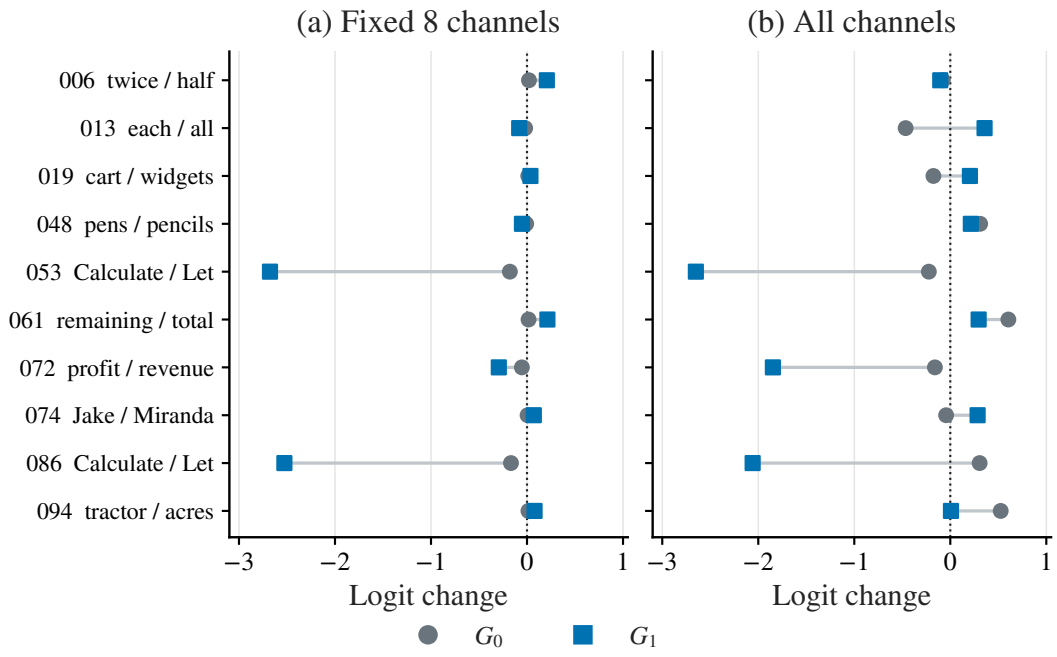


Figure 12: Effects of switching initial to current gain under initial (G_0) or current (G_1) interactions; positive values favor the first token. Ten inspected prompts, with eight channels selected on television case 086 (053: tax). Interventions act at saved answer branches of the frozen step-200 model.

D.2 Channel and First-Token Controls

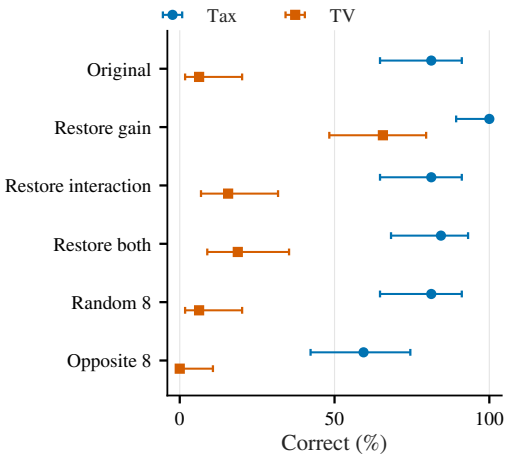


Figure 13: Gain and interaction restorations, with random and opposite-channel controls. Explicit-answer accuracy over 32 continuations; 95% Wilson intervals.

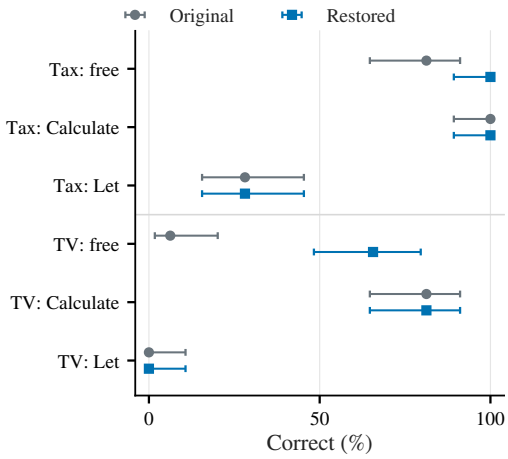


Figure 14: First-answer-token controls with paired seeds and 95% Wilson intervals. With the first token constrained, 127/128 continuation pairs have identical token sequences.

D.3 Numerical and Context Variants

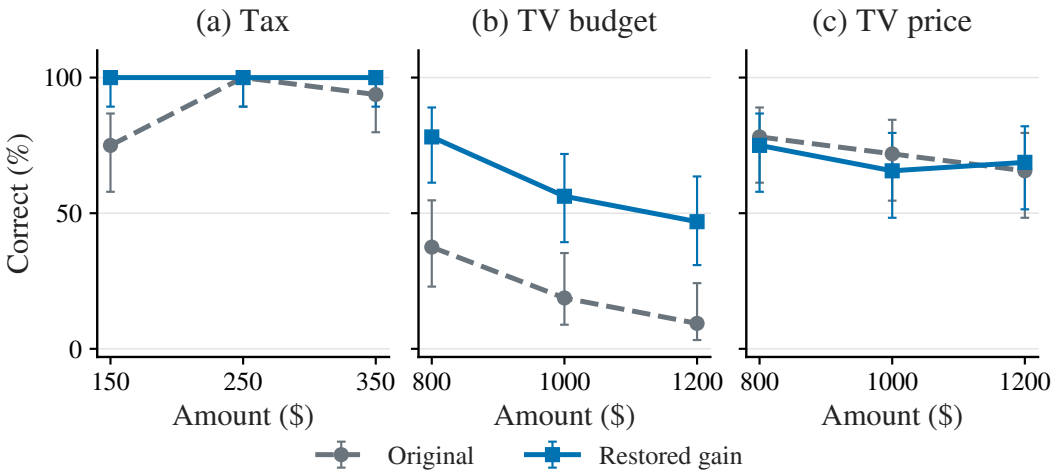


Figure 15: The same eight channels across tax, television-budget, and explicit-price variants. Each condition uses 32 continuations; whiskers show 95% Wilson intervals.