School of Computation, Information and Technology
Technical University of Munich

Does a Weaker Twin Know Where an LLM's Reasoning Matters? An Interventional Test of Master–Novice Divergence for Reward Densification

Ana Carolina Ramos Cunha Leonhard Driesch Leonard Evers

School of Computation, Information and Technology
Technical University of Munich

ana.cunha@tum.de leo.driesch@tum.de ge95gih@mytum.de

Abstract: Reinforcement learning for large language models on verifiable tasks suffers from reward sparsity: only the final outcome receives a signal, leaving credit assignment across long chains of thought unresolved. We investigate a Master/Novice framework that compares a strong policy to a structurally matched, capacity-reduced *Novice*, obtained by tensor disentanglement of the Master's MLP weights, and asks whether top-1 disagreement (positions where the Novice would emit a different next token) identifies the steps that deserve credit. We contribute (i) the conceptual framework and its central, testable hypothesis; (ii) an end-to-end pipeline that scales tensor disentanglement to Qwen3.5-4B and yields Novices at any rank fraction from a single checkpoint; and (iii) a hypothesis-first interventional evaluation on MATH-500. Three interventions of increasing strength (handing generation to the Novice, perturbing single tokens, corrupting whole reasoning steps) consistently fail to reject the null: top-1 disagreement positions carry no more outcome leverage than control positions of similar depth where both models agree. The only reliable effect is local: disagreement positions are more fragile in the immediate continuation, and this effect persists when controlling for Master entropy. We report these negative results, characterize a *coherence cliff* (the rank fraction below which Novice generations lose coherence), and discuss the implications for densifying rewards with top-1 disagreement.

Keywords: reinforcement learning, verifiable rewards, credit assignment, tensor disentanglement

1 Introduction

Large language models (LLMs) trained with reinforcement learning (RL) often receive only a sparse, verifiable reward at trajectory end (correct or incorrect) while intermediate tokens receive no direct feedback. This *reward sparsity* impedes sample efficiency and *credit assignment*, the problem of deciding which of a trajectory's many actions deserve credit for its single terminal reward: the policy cannot tell which reasoning steps were load-bearing and which were routine (Cui et al., 2025).

Recent work densifies rewards without human step labels by exploiting structure already present in the task or the model: partial test-case success in code, worth up to +8.83 pass@1 (share of problems solved at the first attempt) over outcome-only training (Wang et al., 2026); implicit process rewards trained from outcome labels alone, with a reported 2.5× sample-efficiency gain (Cui et al., 2025); teacher–student divergence in distillation (Xu et al., 2026); and the observation that a minority of high-entropy "forking" tokens concentrates most of the learning signal in RL with verifiable rewards (RLVR) (Wang et al., 2025).

This paper explores a complementary idea: construct a weaker *Novice* from the same base weights as the *Master* via tensor disentanglement, compare their next-token distributions on successful Master rollouts, and upweight positions where the two diverge. Unlike an externally trained reward model, the Novice shares architecture and tokenizer with the Master and offers a high-resolution knob (the retained rank fraction) for how weak the comparator should be.

This approach relies on the premise that such points of disagreement are ones where the Master relies on knowledge the Novice lacks and therefore deserve higher credit. We express this premise with the following Hypothesis:

H1 (outcome leverage). *Positions where the Novice diverges from the Master have a larger effect on final-answer correctness than depth-matched positions where the two agree.*

The goal of this paper is to test this hypothesis: if it

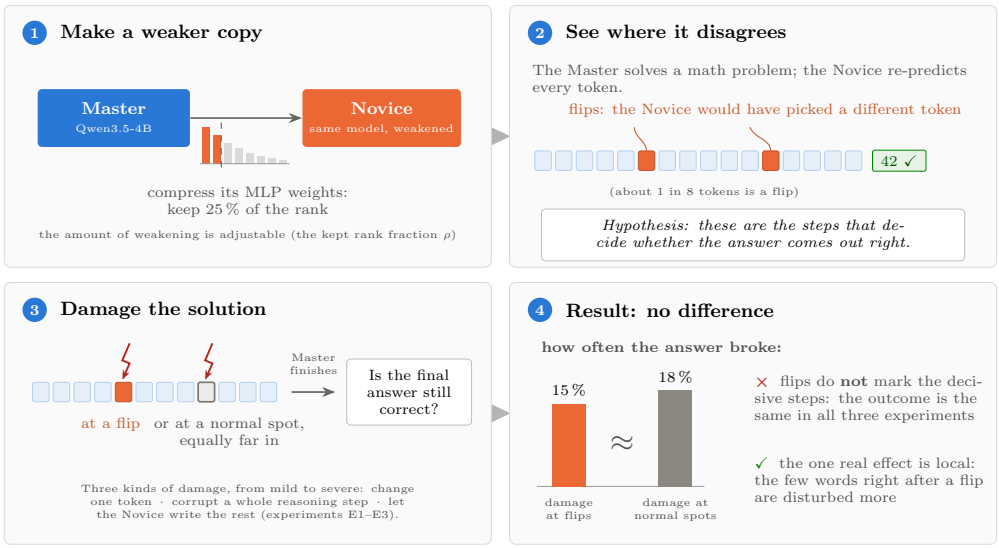

Figure 1: **(1)** Compress the Master's MLP weights into a weakened *Novice*; **(2)** the Novice re-predicts the Master's correct MATH-500 solutions, marking disagreements (*flips*); **(3)** solutions are damaged at flips versus depth-matched normal positions and the Master finishes. **(4)** *Takeaway:* the answer breaks equally often either way (shown: whole-step damage; all three experiments agree), so flips mark where the Novice's missing capacity acts, not where the outcome is decided, and the top-1 flip signal cannot densify the reward.

holds, the idea has merit and is worth exploring further. If not, the assignment of dense per-position rewards by Master/Novice disagreement cannot be justified at this time.

Figure 1 summarizes the approach and its outcome.

Our contributions are of three kinds. *Conceptually*, we formalize the Master/Novice framework and reduce its promise to a single testable hypothesis: divergence positions have higher leverage on the final outcome than matched non-divergence positions. *Methodologically*, we implement an end-to-end `NoviceModel` pipeline. It disentangles, saves/loads, bakes to a chosen rank, and finally generates. This pipeline is validated from MLP-scale prototypes through Qwen3.5-4B, together with a ladder of three intervention designs for testing per-token credit signals before committing to RL training. *Empirically*, we characterize the disentanglement and Novice-quality landscape with a learning-rate sweep, an MMLU rank sweep and a generation-level coherence cliff which identifies a rank fraction at which the Novice's output degenerates to the point of being unusable. We report a consistent, statistically quantified negative result on the central hypothesis, alongside one genuine positive: a local fragility effect at disagreement positions that is not explained by Master entropy alone.

2 Related Work

We group prior work by where the dense signal comes from: the *task* (verifiable structure) or the *model* (internal statistics). For each line of work we focus on its relation to our approach; we refer to the original papers for full results.

2.1 Dense rewards from verifiable task structure

VeRPO (Wang et al., 2026) densifies code-RL rewards using partial test-case success. Its central observation is cautionary: the naive dense signal (the raw test-suite pass rate) can *underperform* sparse outcome rewards, because redundant easy tests dominate the reward by sheer count (a *cardinality bias*). VeRPO corrects this with a dynamically calibrated per-test weighting and re-anchors the result to the end-to-end outcome, outperforming both outcome-only GRPO (a widely used policy-optimization algorithm for LLM RL) and baselines built on learned reward models (RMs), itself without any learned reward model. We selected it both as the clearest instance of extracting density from the *verifier* side and for its lesson that denser is not automatically better. This same lesson motivates our interventional validation of the divergence signal before any training. This approach as a whole differs from ours in scope and granularity: the signal exists only where partial success is measurable (currently code) and is assigned per turn, whereas a Master/Novice comparator is task-agnostic and per token.

PRIME (Cui et al., 2025) obtains per-token process rewards without step labels by training an outcome reward model whose log-ratio to a reference policy is reinterpreted as an implicit process reward, updated online alongside the policy. It substantially improves sample efficiency over outcome-only RL on math and code. PRIME is the closest point of comparison for our goal (token-level credit from outcome supervision only); the difference is that PRIME *learns* its dense signal, whereas we test whether a fixed, training-free capacity-reduced copy of the policy can supply one. Supervised process reward models reach the same granularity from explicit step la-

bels (Lightman et al., 2023, Wang et al., 2023); a training-free comparator sits at the opposite end of this supervision spectrum.

2.2 Token-importance signals inside the model

High-entropy minority tokens. Wang et al. (2025) show that in RLVR most chain-of-thought tokens are low-entropy "followers," while a minority of high-entropy forking tokens carries the learning signal: restricting policy gradients to the top 20% highest-entropy tokens matches or exceeds full-gradient training. We selected this work because Master entropy is the natural single-model baseline that any two-model divergence signal must beat; our experiments therefore control for entropy explicitly.

TIP (Xu et al., 2026) studies on-policy distillation and identifies informative tokens along two axes, student entropy and teacher-student divergence: entropy-only selection is structurally blind to positions where the student is confidently wrong, and training on under 10% of tokens, from the low-entropy, high-divergence region alone, nearly matches full-token distillation. TIP is the closest conceptual relative of our hypothesis in the sense that divergence between a strong and a weak model marks important tokens, but it operates in distillation, where the teacher's distribution is the training target itself. Whether the same signal identifies *outcome-critical* steps for RL credit assignment is exactly the question our experiments address; our negative results suggest the transfer is not automatic.

Compared with all four, the Master/Novice comparator requires no additional training, no task-specific verifier beyond the outcome check, and applies to any domain with verifiable answers; the price, as our results show, is that the signal it produces must be validated through interventions rather than assumed.

3 Methodology

3.1 Setup and hypothesis

We treat autoregressive generation as a Markov decision process, the standard sequential-decision formalism of RL: state s_t is the prompt plus generated prefix $y_{<t}$, action a_t is the next token, and the reward is binary verifiable correctness at episode end. The **Master** is the full-capacity policy π_M. The **Novice** is π_N, derived from the same checkpoint by truncating disentangled MLP (multi-layer perceptron) factors to a rank fraction $\rho \in (0, 1]$, the share of each layer's internal rank that is kept (defined precisely in Section 3.2); ρ=1 reproduces the Master and smaller ρ yields a weaker Novice. On a Master rollout $\tau = (y_1, \ldots, y_T)$ that reaches a correct answer, per-token divergence is

$$\delta_t = D\big(\pi_M(\cdot \mid y_{<t}),\, \pi_N(\cdot \mid y_{<t})\big), \tag{1}$$

where D measures how different the two next-token distributions are, for example the Kullback–Leibler (KL) or Jensen–Shannon divergence; or, in the discrete limit used throughout our experiments, D reduces to the *top-1 flip* indicator

$$\mathrm{flip}_t = \mathbb{1}\big[\arg\max \pi_N(\cdot \mid y_{<t}) \neq y_t\big], \tag{2}$$

i.e., whether the Novice would have emitted a different token at position t. The top-1 flip is one instantiation of the framework, chosen for the experiments because it is parameter-free and directly interpretable; the framework itself is agnostic to the divergence measure. This setup, and the top-1 flip as its discrete disagreement indicator, follow the Observer pattern of the weight-matrix disentanglement (WMD) introspection framework (Till, 2026).

The intended use of δ_t is as a shaping signal that concentrates credit on high-divergence steps of verified-correct rollouts. Before such a signal enters any RL objective, however, the premise H1 we defined earlier must hold. The experiments in this paper are direct tests of H1 at increasing intervention strength. We deliberately do not evaluate any RL training run: if H1 fails, advantages weighted by the divergence between Master and Novice cannot be shown to concentrate gradient where the outcome is decided, and the training experiment would be uninformative.

3.2 Constructing the Novice via weight matrix disentanglement

The Novice must be *policy-preserving*: it should retain as much of the Master's behavior as a given rank budget permits, so that its remaining disagreements are attributable to missing capacity rather than to a lossy construction. The classical route, a truncated singular value decomposition (SVD) of each weight matrix, treats the matrix as a flat 2D array and ignores the multi-dimensional structure of neural weights. We instead use *Weight Matrix Disentanglement* (WMD) (Till, 2026), following the Master/Novice framework of Till (sebis/TUM), which adapts tensor disentanglement from numerical linear algebra (Wei et al., 2025) to neural network layers.

Let $X \in \mathbb{R}^{lr \times bc}$ be a decoder-MLP weight matrix whose row and column counts are each written as a product of two integers, lr split into the near-balanced pair (l, r) and bc into (b, c); this lets the flat matrix be read as a four-index tensor rather than a 2D array, which is what exposes the multi-dimensional structure that plain SVD ignores. Writing M for the reshape of X into the 4th-order tensor $\mathcal{X} \in \mathbb{R}^{l \times r \times b \times c}$ and P for the index permutation $(l, r, b, c) \to (l, c, r, b)$, WMD re-bipartitions the tensor with the operator

$$A \equiv M^{-1} \circ P \circ M\colon\ \mathbb{R}^{lr \times bc} \to \mathbb{R}^{lc \times rb}, \tag{3}$$

exposing cross-dimensional correlations that the flat matricization hides. A learnable orthogonal *gauge* $Q \in O(lr)$ (the orthogonal group; the word *gauge* signals that Q is later undone by $Q^\top$ during reconstruction, re-expressing the layer's coordinates without changing the layer itself) is applied to the row space before re-bipartitioning, and is optimized to concentrate the spectrum of $A(QX)$ in its leading singular directions, i.e., to make a few large singular values carry most of the energy. For a target rank k, the optimal disentangler minimizes the energy of the discarded singular tail:

$$Q_k^* \;=\; \arg\min_{Q \in O(lr)} \sum_{i>k} \sigma_i^2\big(A(QX)\big). \tag{4}$$

The tail sum $\sum_{i>k} \sigma_i^2$ is exactly the squared error of replacing $A(QX)$ by its best rank-k approximation, so a gauge that shrinks it makes the layer more compressible. Because A is an isometry (a norm-preserving map) and Q is orthogonal, the total singular energy of $A(QX)$ is invariant, so (4) directly minimizes the Frobenius reconstruction error (the sum of squared entrywise differences) between the original and truncated layer; the identity gauge $Q{=}I$ recovers plain SVD truncation, so the optimized gauge is never worse. The rank-k Novice weight is then reconstructed by inverting the pipeline, a step we call *baking*,

$$\widetilde{X}^{(k)} \;=\; Q^{*\top} A^{-1}\big(\mathrm{trunc}_k(A(Q^* X))\big), \tag{5}$$

and the per-layer rank is set by a single global rank fraction: $k(\rho) = \max(1, \lceil \rho\, m \rceil)$, where $m = \min(lc, rb)$ is the largest possible rank of $A(QX)$. Thus ρ is literally the fraction of singular directions kept, and the outer max with 1 guarantees at least one is retained.

The optimization problem (4) admits a closed-form gradient that reuses one SVD of $A(QX)$ per step (Wei et al., 2025). Our pipeline accelerates it by roughly 30–50× per layer (Adam on a Cayley-parameterized Q, SVD reuse, 32-bit precision; details in Appendix A), with truncation cost within ∼10% of the double-precision reference; independent layers are processed in parallel, which made 4B-scale Novices feasible.

WMD also supplies a theoretical link to the divergence signal of Section 3.1: the per-state KL divergence between a policy and its truncated counterpart is bounded by the discarded singular tail, and at the rank the gauge was optimized for, the optimized gauge yields the tightest such bound (Till, 2026). WMD rather than plain SVD thus keeps the observed Master/Novice divergence dominated by genuinely missing capacity rather than by avoidable truncation error. How the bound applies at our bake rank, and the empirical check that the gauge's advantage transfers across ranks, are given in Appendix C.

3.3 The `NoviceModel` pipeline

Figure 2 summarizes the `NoviceModel` lifecycle from base model to generating Novice.

The `NoviceModel` lifecycle

1. Load the base causal LM (the Master).
2. Swap every decoder-MLP linear for its factored form.
3. Optimize the gauge Q per layer via (4); store the factors (Q, U, s, V_h) to disk.
4. Bake: truncate the stored factors at the chosen ρ and reconstruct dense weights via (5).
5. Generate with the baked model unchanged.

Figure 2: The `NoviceModel` lifecycle. Step 3 runs once; because the stored factors retain the full decomposition, step 4 yields a Novice at any ρ by cheap re-truncation.

A worked example of the factorization for one layer is given in Appendix A. Sharded checkpoints are loaded and baked one layer at a time, keeping peak memory near a single model. An equality test confirms that a Novice baked at $\rho{=}1$ reproduces the base model's logits. The pipeline was validated incrementally on toy weight matrices, MNIST- and CIFAR-scale MLPs, a small GPT-2-style transformer, Qwen3.5-0.8B-Base, and finally Qwen3.5-4B, the model used in all experiments below.

4 Experiments

4.1 Models, data, and grading

Models. Master: `Qwen/Qwen3.5-4B` (full rank). Novice: the disentangled checkpoint of the same model (gauge optimized at $\rho{=}0.1$), baked at $\rho{=}0.25$; the bake-rank choice is justified empirically in Section 5.1.

Data. MATH-500 (`HuggingFaceH4/MATH-500`), problems 0–49, prompted via the chat template with the standard instruction to place the final answer in `\boxed{}`; the box gives the grader and the interventions below a machine-checkable answer location. All decoding is greedy, hence deterministic. The only randomness in the pipeline is the fixed seed that selects control positions and free-run subsets, so every experiment is exactly reproducible. Master rollouts are capped at 512 new tokens in E1 and 700 in E2/E3 (the shared prefixes are identical, and the same 23 problems are solved), which suffices for most rollouts to terminate naturally.

Grading. A balanced-brace `\boxed{}` extractor with LaTeX-normalizing equality (fraction, spacing, and delimiter normalization; numeric fallback). The Master solved 23/50 problems; all credit-assignment experiments operate on these 23 verified-correct rollouts. The solved problems skew easy: 20 of 23 are MATH difficulty levels 1–3 (mean level 2.5, versus 3.9 for the unsolved problems). Across them, the Novice is teacher-forced on the Master's tokens and flips are recorded at every generated position (10,895 scored tokens; flip rate 12.3%). The disentangled factor checkpoint is publicly available, and Appendix A collects all remaining reproduction details (seeds, prompts, evaluation configuration); code and per-trial artifacts are available from the authors.

4.2 Novice characterization experiments

Learning-rate selection. Disentanglement quality is sensitive to the optimizer learning rate; a sweep on a representative Qwen3.5-0.8B MLP layer selects 10^{-3}, used for all subsequent disentanglement runs (Appendix B).

MMLU rank sweep. Novice quality as a function of ρ is measured with `lm-eval` on the MMLU benchmark (Massive Multitask Language Understanding, a standard multiple-choice knowledge test), 5-shot, limited to 5 questions per subtask (a fixed-budget subset; the same protocol at every rank, so relative comparisons are unaffected by the limit).

Generation coherence. Because MMLU scores likelihood ranking rather than generation, we separately inspect greedy free-running generations of the Novice at $\rho \in \{0.1, 0.25, 0.5, 0.75\}$ on MATH-500 prompts.

4.3 Intervention experiments

All three experiments compare *flip* positions against *depth-matched non-flip controls*. The *depth* of a position is its relative location within the generated part of the rollout, from 0 (first generated token) to 1 (last). Depth matching is essential because an intervention late in a rollout is much easier to recover from, since most of the correct solution is already in place; how far into a rollout an intervention occurs in fact dominates its effect (Section 5.2), so comparisons at unmatched depths would be confounded. Controls are drawn once per flip with a fixed random seed.

E1: Release-from-here. At a chosen position p the Novice takes over: it receives the Master's prefix $y_{<p}$ and free-runs greedily to completion (budget: remaining rollout length plus 48 tokens); the completed trajectory is graded.

Release positions are selected as follows. Within each rollout, all flips are ranked by their *confidence gap*: the probability the Novice assigns to its own preferred token minus the probability it assigns to the Master's token at that position. A near-zero gap means a borderline disagreement, a large gap a confident one. We release at the 5 most confident flips per rollout, since these are the flips most likely to matter if H1 is true; selecting them biases the test toward finding an effect, which makes a null result more conclusive. Although the selection is relative to each rollout, the selected flips are confident in absolute terms: all 115 have a confidence gap of at least 0.5 (median 0.93, versus a median of 0.44 over all flips). Each selected flip is then paired with the not-yet-used non-flip position closest to it in depth, and every control is used at most once. With 23 rollouts and 5 pairs each, this yields $23 \times 5 = 115$ pairs, i.e., 230 Novice free runs; five pairs per rollout balance per-problem coverage against generation cost. If H1 holds, releasing at a flip should derail the answer more often than at its matched control.

E2: Single-token perturbation. E1 is a coarse test: after the handoff, a whole weak model free-runs in both arms, so the effect of the single position is diluted by the long shared continuation. E2 instead perturbs exactly one token and lets the *Master* continue. It reuses the same 5 flip/control pairs per rollout as E1, and each of the 115 pairs produces three trials, one per *arm*. An arm is a group of trials that all receive the same intervention; here, each arm splices one kind of token at one kind of position, and outcomes are compared between arms:

- `flip_novice` splices the Novice's token at the flip position;
- `flip_rank2` splices the Master's own second most probable token at the same flip position;
- `control_rank2` splices the second most probable token at the matched control position.

The second-choice token is used because it exists at every position and is a perturbation of comparable strength in both groups. Comparing the two `rank2` arms therefore isolates the *position* effect under an identical perturbation type, while comparing the two flip arms isolates the effect of the Novice's *specific* token at an identical (hence entropy-matched) position. In total this gives $23 \times 5 \times 3 = 345$ trials, 115 per arm.

Three metrics are recorded per trial:

- *Local disruption*: the mean drop in the log-probability the Master assigns to its own next K=8 tokens when the perturbed prefix replaces the clean one, in nats per token (nats defined in Appendix D); the short window keeps the metric local while averaging out single-token noise.
- *Answer log-probability drop*: the same drop on the Master's answer tokens (the `\boxed{}` content), with the reasoning in between teacher-forced, i.e., held fixed, so only the effect that reaches the outcome is measured; trials perturbed after the answer span are excluded (no answer tokens remain).
- *Answer-break*: whether the Master, free-running greedily from the perturbed prefix, no longer reaches the correct answer; each free run is a full generation, so this is evaluated on a random 90-of-345 subset to bound compute.

Because flips concentrate at positions where the Master itself is uncertain, and perturbing an uncertain position disturbs the continuation more regardless of the position's importance, all position comparisons are additionally adjusted for entropy by ordinary least squares (OLS), $y = \beta_0 + \beta_1 \,\mathrm{isflip} + \beta_2 \, H_M$, where isflip indicates whether the position is a flip and H_M is the Shannon entropy (in nats) of the Master's next-token distribution at the perturbed position. The regression is fitted over the two `rank2` arms only, so that both groups received the identical perturbation type; β_1 then estimates the difference between a flip and a control position at equal entropy.

E3: Span perturbation. Single tokens may be the wrong granularity for reasoning. E3 corrupts an entire reasoning step, approximated here as a span of consecutive tokens ending at a line break, since the Master's solutions place line breaks between short units such as a sentence or a displayed equation. Spans shorter than 4 tokens are discarded as trivial fragments. It then lets the Master free-run to the answer, so the downstream reasoning may adapt. Steps are ranked by *flip density*, the fraction of their tokens that are flips; a fraction rather than a raw count, so that long steps are not favored over short steps with concentrated disagreement. Each of the 3 steps with the highest flip density per rollout is paired with a step of similar depth but low flip density (three per rollout to balance coverage of each problem against generation cost, as in E1), giving $23 \times 3 = 69$ step pairs.

Arms mirror E2: generic corruption (per-position second-choice tokens) of a flip-dense step, generic corruption of a control step, and replacement of the flip-dense step by the *Novice's version* of it (its preferred token at every position), for $69 \times 3 = 207$ trials in total, 69 per arm. The primary metric is the binary answer-break rate of the Master's free run, evaluated on 150 of the 207 trials; the teacher-forced answer drop and the entropy-adjusted regression (with the step's mean entropy as covariate) are reported as in E2, where one step pair without answer tokens after the span is excluded.

4.4 Statistical analysis

We attach a standard test to each comparison so that a null can be quantified rather than merely asserted; rates carry 95% Wilson confidence intervals (CIs). E1's arms are *paired* (each flip has its own depth-matched control) and are compared with McNemar's exact test. E2 and E3's free-run arms are *unpaired*, because their compute-bounded subsets were drawn uniformly over trials without preserving the pairing, fixed by seed before any outcome was generated (Appendix A); they are compared with Fisher's exact test. For regression coefficients we report the estimate, its standard error (SE), and the t-value; at our sample sizes $|t| \gtrsim 2$ corresponds to $p \lesssim 0.05$. Appendix D explains each test.

5 Results

5.1 Novice characterization

Gauge quality across bake ranks. The checkpoint's gauge was optimized at ρ=0.1 while the interventions bake at ρ=0.25, so the KL bound's tightness is not guaranteed at the bake rank (Appendix C). Measured on the stored spectra, the advantage transfers: the optimized gauge's reduction of discarded tail energy relative to identity-gauge (plain SVD) truncation is large, if heterogeneous across layers, and essentially rank-independent between ρ=0.1 and ρ=0.25 (Appendix C).

MMLU vs. rank. On Qwen3.5-4B (5-shot, 5 questions per subtask), accuracy stays near the base model's 0.737 down to $\rho \approx 0.5$, then degrades monotonically to 0.42 at ρ=0.1: still above the 0.25 random baseline (Figure 3).

Coherence cliff. Free-running generation reveals more than likelihood-based MMLU: at ρ=0.1 every inspected generation collapses into degenerate repetition (the model loops on the same words or phrases instead of continuing the solution), while at $\rho \geq 0.25$ generations are fluent, structured solutions. Consistently, the Novice's teacher-forced flip rate on Master rollouts falls from 44.0% at ρ=0.1 (disagreement smeared over half the tokens, i.e., noise) to 12.3% at ρ=0.25 (concentrated disagreement). We therefore run all interventions at ρ=0.25, the sharpest coherent operating point: the Novice is markedly weaker than the Master generatively (released on the Master's own trajectories it completes them correctly only 26% of the time, against the Master's 100% by construction; E1 below), yet its disagreements are attributable to specific positions rather than global incoherence.

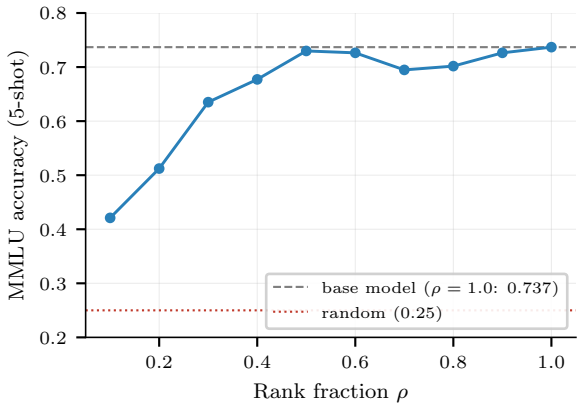

Figure 3: MMLU accuracy vs. Novice rank fraction on Qwen3.5-4B (`lm-eval`, 5-shot, 5 questions per subtask). Accuracy is preserved until $\rho \approx 0.5$, then degrades; note that at ρ=0.1 the Novice still scores above chance on likelihood ranking even though its free-running generation is degenerate.

5.2 Intervention experiments

Table 1 summarizes the outcome-level results of all three experiments; Table 2 reports E2's graded per-token metrics.

Table 1: Outcome-level intervention results. E1 reports the Novice's completion success rate after taking over at the given position (H1 predicts *lower* success at flips). E2/E3 report the rate at which the Master's free run no longer reaches the correct answer after the perturbation (H1 predicts *higher* break rates at flips). E2 counts reflect the 90-trial free-run subset drawn uniformly, to bound compute, across its three arms (29/30/31 trials), not all 345 perturbations. No comparison is significant; exact tests in the text.

Exp.	Arm	Rate	Count
E1 (success↑)	release at flip	0.261	30/115
	release at control	0.270	31/115
E2 (break↓)	flip, Novice token	0.241	7/29
	flip, rank-2 token	0.100	3/30
	control, rank-2	0.194	6/31
E3 (break↓)	flip-dense step	0.151	8/53
	control step	0.176	9/51
	flip-dense, Novice version	0.152	7/46

Table 2: Graded metrics for E2's three perturbation arms (means over 115 trials per arm; the answer-drop column excludes trials perturbed after the answer span). Local disruption: mean drop in the Master's log-probability of its own next K=8 tokens after the one-token perturbation; answer drop: the same drop on the final-answer tokens with the reasoning in between held fixed; H1 predicts larger drops at flips. H_M: mean Master entropy at the perturbed position; flips sit at high-entropy positions, motivating the entropy adjustment. Local disruption differs between flip and control positions, but the answer-tied drop is negligible for every arm.

E2 arm	Local disr. (nats/tok)	Answer drop (nats/tok)	H_M
flip, Novice token	2.56	0.0009	0.41
flip, rank-2	2.55	0.0009	0.41
control, rank-2	1.60	0.0013	0.08

E1: no release-point effect. Success after handing over at a flip was 30/115 (0.261; Wilson 95% CI $[0.189, 0.348]$) versus 31/115 (0.270; CI $[0.197, 0.357]$) at depth-matched controls; paired difference +0.009. Since each flip has its own depth-matched control, the informative comparison is within each pair: in 108 of the 115 pairs both releases had the same outcome, and the seven discordant pairs split 4 (control succeeded, flip failed) versus 3 (the reverse). McNemar's exact test on these discordant pairs gives $p = 1.00$: a 4-versus-3 split is what chance produces. Success is instead governed by release depth (Figure 4): in the deepest quartile of release positions both arms succeed at ≈ 0.7, in the earliest at $\approx$ 0–0.1. H1 is not supported at this granularity.

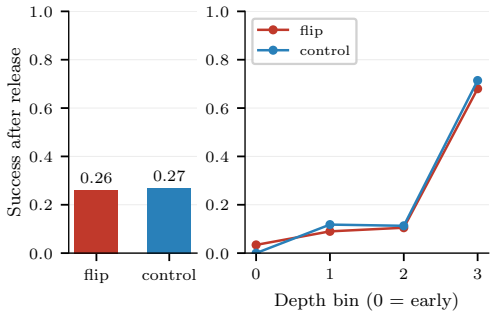

Figure 4: E1: the Novice takes over a correct Master solution at a chosen position and finishes it. A flip is a position where the Novice's next token differs from the Master's; H1 predicts lower success when releasing at flips. Left: Novice completion success after taking over at flip vs. depth-matched control positions. Right: success by release-depth quartile. Both arms rise identically with depth: the outcome is governed only by how much of the Master's trajectory the Novice inherits.

E2: local disruption, no effect on the answer. Perturbing at a flip disrupts the Master's immediate continuation substantially more than at a matched control (2.55 vs. 1.60 nats/token, Table 2), a raw gap of +0.94. Flips sit at higher-entropy positions (H_M 0.41 vs. 0.08), and entropy independently predicts disruption ($\beta_2 = 0.97$, SE 0.36, $t = 2.69$), but the flip effect persists after the adjustment: $\beta_1 = 0.62$, SE 0.26, $t = 2.35$ (n=230, the two `rank2` arms). The raw gap thus splits into roughly 0.32 nats from the groups' entropy difference and 0.62 nats tied to flip status; the flip>control ordering also holds within every entropy tercile (Appendix D). The effect does not reach the outcome, however: the answer-tied drop is $\sim$0.001 nats/token in *every* arm, with no flip effect ($\beta_1 = -0.0002$, SE 0.0017, $t = -0.14$, n=225 after answer-span exclusions), and the free-run answer-break subset shows no coherent ordering (7/29, 3/30, and 6/31 broken answers per arm; Fisher OR 0.46, $p = 0.47$; Appendix D). The Novice's specific token adds nothing over the Master's own second choice: the paired difference is +0.017 nats/token locally (Novice larger in only 18% of pairs) and +0.0001 on the answer metric (11%).

E3: no step-level effect. Corrupting a flip-dense reasoning step broke the final answer in 8/53 free runs (0.151; CI $[0.079, 0.271]$) versus 9/51 (0.176; CI $[0.096, 0.303]$) for depth-matched low-disagreement steps. Fisher's exact test gives an odds ratio of 0.83 ($p = 0.80$): the flip-dense step broke the answer slightly *less* often, opposite to H1's prediction. Replacing a flip-dense step by the Novice's own version was indistinguishable from generic corruption of the same step (7/46 vs. 8/53; Fisher $p = 1.00$), and the entropy-adjusted answer-drop OLS is likewise null ($\beta_1 = -0.0018$, SE 0.0023, $t = -0.79$, n=137). Notably, the Master recovered the correct answer after whole-step corruption in 84% of all trials (126/150), independent of the step's flip density.

6 Discussion

6.1 What the results mean for H1

Across three intervention designs of increasing strength, we consistently fail to reject the null, even in E3, whose free-running outcome metric gives downstream reasoning every chance to amplify a corrupted step. Our tests provide no evidence that where the Novice disagrees with the Master is where the outcome is decided: flip positions and flip-dense steps showed no more outcome leverage than depth-matched controls, and the Novice's specific alternatives were no more damaging than the Master's own second choices. On this evidence, naive top-1 flips used as a per-token or per-step credit weight cannot be expected to concentrate learning where it matters; an RL run built on them would likely match uniform weighting at extra cost. These conclusions are scoped to the top-1 flip instantiation of δ_t. Graded variants that weight positions by KL or JS divergence remain untested and could rank positions differently; for them, H1 is open rather than unsupported.

Two genuine findings survive alongside the negative. First, disagreement positions are *locally* special: perturbations there disrupt the immediate continuation about 60% more than at matched positions before any adjustment, and roughly two thirds of that gap (0.62 of +0.94 nats/token) persists at equal Master entropy. The flip sig-

nal thus measures something real about where the truncated capacity acts, but it is local fragility, not outcome leverage. Second, the Master is notably robust on MATH-500: even corrupting an entire reasoning step leaves the final answer intact 84% of the time. Only a small minority of steps is load-bearing, and identifying them evidently requires more than capacity-difference signals.

On entropy: our data do *not* show that entropy is a sufficient credit signal either. Flips and entropy are correlated, each carries independent *local* information, and neither, in our setting, identifies outcome-critical positions; this is consistent with reading high-entropy tokens as forks whose importance emerges over many training rollouts (Wang et al., 2025) rather than within a single trajectory.

6.2 Why the hypothesis may fail

The premise behind H1 conflates two notions of importance: a step can be hard *for a weaker model to produce* without being critical *to the outcome*. Much of what extra MLP capacity buys appears to be fluency and confidence at delicate positions, which is exactly what the local-disruption effect detects. Meanwhile the mathematically decisive content is either redundantly encoded across the trajectory (the Master re-derives it after corruption) or concentrated in a few steps that flips do not preferentially mark (Section 7 notes the related caveat that our interventions may probe recoverability more than necessity). In distillation the same signal can still be valuable, because "hard for the weaker model" *is* the training target (Xu et al., 2026); for RL credit assignment, outcome leverage must be established separately, and our tests do not support it.

6.3 Outlook

The infrastructure result stands independently of the negative: Master/Novice pairs can be produced repeatably at 4B scale, with rank fraction as a smooth, characterized control knob and a documented coherent operating range. Beyond credit assignment, the pipeline is a reusable primitive: one run yields capacity-controlled variants at every rank for distillation or analysis work. The Novice itself is a diagnostic construction, not a deployment artifact: discarded at test time, it only scores positions during training and never generates, so its weakness carries no inference-time cost; the same tests apply to any weak comparator sharing the Master's tokenizer, such as an earlier checkpoint or a smaller pretrained model. Given H1's failure, we see three defensible directions: (i) aggregate use of divergence (e.g., trajectory-level divergence mass as a difficulty signal); (ii) combining divergence's independent local information with entropy-based token selection (Wang et al., 2025); and (iii) benchmarking any future training-free comparator against learned dense rewards in the PRIME family (Cui et al., 2025). Each should be preceded by the same interventional validation applied here.

7 Limitations

Our conclusions are bounded by the setting: one Master (Qwen3.5-4B), one Novice family (MLP-only disentanglement, gauge optimized at ρ=0.1, evaluated chiefly at ρ=0.25), one dataset (MATH-500; 23 Master-correct rollouts), greedy decoding, and top-1 flips as the divergence instantiation. Conditioning on Master-correct rollouts matches the intended use of a shaping signal but skews the evaluated set easy (20 of 23 solved problems are MATH levels 1–3); harder problems with longer chains of dependent steps might weight individual steps more, the 84% recovery rate may partly reflect this easier regime, and our conclusions are scoped to problems the Master solves. The gauge optimization stopped after 40 compute-bounded steps with cost still decreasing (Figure 5), so some observed divergence may be avoidable truncation error, though the gauge already clearly beats plain SVD (Appendix C). The binary outcome metrics have modest power, so their nulls rule out only large effects. At 80% power, E1's 115 pairs could detect a flip–control success asymmetry of about 11 percentage points, E3's 53 vs. 51 free runs a rise in the step break rate from 0.18 to about 0.43, and E2's 30-trial free-run arms only larger effects still (definitions and assumptions in Appendix D); smaller but practically relevant effects would go undetected, so the consistent nulls argue against a large hidden effect, not a modest one. The regression standard errors treat positions as independent although the 230 perturbed positions are nested within 23 rollouts; such clustering typically understates standard errors, so $t = 2.35$ is best read as an upper bound on the evidence, with a rollout-clustered analysis the appropriate follow-up. The MMLU sweep's fixed 5-question-per-subtask budget supports relative comparisons across ρ, not absolute benchmark claims. The answer-tied drop metric teacher-forces intervening reasoning and is conservative by design; the E1/E3 free-run metrics are the complementary, unconstrained tests. Three further measurement caveats apply: newline-delimited spans are a syntactic proxy for a semantic reasoning step; rank-2 substitutions may produce disturbances that the Master repairs mechanically rather than reasoning through, so E2 and E3 may measure recoverability from local perturbation more than the necessity of a step's content; and the local-disruption window K=8 is a heuristic default, so the local-fragility result's sensitivity to K remains unchecked. Finally, continuous divergence measures (e.g., per-trajectory Jensen–Shannon mass), sampled rather than greedy rollouts, and other Novice constructions remain untested and are natural follow-ups.

8 Conclusion

We formalize the Master/Novice framework for densifying verifiable rewards, build the tensor-disentanglement pipeline that makes it practical at 4B scale, and subject its central premise to a ladder of three intervention tests on Master-solved MATH-500 problems. The premise is not supported: we find no evidence that top-1 disagreement between Master and Novice marks outcome-critical positions or steps (McNemar p=1.00; Fisher p=0.80), and the Novice's specific token choices carry no more

leverage than generic perturbations. What disagreement does mark robustly, and beyond Master entropy, is local fragility of the immediate continuation. We report this negative result with exact statistics because it narrows an otherwise attractive idea: a training-free capacity-reduced comparator cannot be assumed to target the right tokens, and interventional validation of any credit signal should precede RL integration. The released pipeline (one disentanglement run, Novices at any rank), the coherence-cliff characterization, and the three reusable intervention designs make exactly that validation cheap.

AI-Use Statement

AI assistants were used for limited drafting and coding assistance: boilerplate code may have been LLM-generated, and LLM tools assisted with editing the manuscript. All content was reviewed, modified, and verified by the authors, who take full responsibility for the final work.

References

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, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

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. *arXiv preprint arXiv:2305.20050*, 2023.

Tristan Till. On agent introspection via policy-preserving weight matrix disentanglement. Working paper, sebis, Technical University of Munich, 2026.

Longwen Wang, Yirui Liu, Xuan'er Wu, Xiaohui Hu, Yuankai Fan, Kaidong Yu, Qizhen Weng, Wei Xi, and Xuelong Li. Beyond binary: Turning partial success into dense verifiable rewards for reinforcement learning in code generation. *arXiv preprint arXiv:2601.03525*, 2026.

Peiyi Wang, Lei Li, Zhihong Shao, R.X. Xu, Damai Dai, Yifei Li, Deli Chen, Y. Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce LLMs step-by-step without human annotations. *arXiv preprint arXiv:2312.08935*, 2023.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui 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. *arXiv preprint arXiv:2506.01939*, 2025.

Julia Wei, Alec Dektor, Chungen Shen, Zaiwen Wen, and Chao Yang. Numerical optimization for tensor disentanglement. *arXiv preprint arXiv:2508.19409*, 2025.

Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, Zhipeng Wang, and Alborz Geramifard. TIP: Token importance in on-policy distillation. *arXiv preprint arXiv:2604.14084*, 2026.

A Reproducibility details

Table 3 collects every hyperparameter and configuration value used in the paper; the paragraphs below add only the details that need explanation rather than lookup.

Table 3: All hyperparameters and configuration values. Together with the published factor checkpoint, these settings specify every experiment in the paper; settings not listed use the released code's defaults.

Disentanglement (published checkpoint)	
Master / base model	`Qwen/Qwen3.5-4B`
Factor checkpoint	`leodriesch/novice-qwen3.5-4b-rf0.1` (Hugging Face Hub)
Factored layers	96 MLP linears (32 decoder layers × {gate, up, down})
Gauge optimization rank	$\rho = 0.1$
Optimizer, learning rate	Adam (Cayley-parameterized, fp32), 10^{-3}
Steps per layer	40 (compute-budget cutoff; Figure 5)
Bake rank, all interventions	$\rho = 0.25$
Data and decoding	
Dataset	MATH-500 (`HuggingFaceH4/MATH-500`), test split, problems 0–49
Decoding	greedy; rollouts capped at 512 new tokens (E1) / 700 (E2, E3)
Model precision	bfloat16
Prompt format	chat template, thinking mode disabled
Interventions	
Seed for controls and subsets	0 (Python `random.Random`)
E1 releases	5 flip/control pairs per rollout; budget = remainder + 48 tokens
E2 perturbations	same 5 pairs; local window $K = 8$; free-run subset 90 of 345
E3 spans	3 pairs per rollout; min. 4 tokens; free-run subset 150 of 207
MMLU evaluation	
Harness	`lm-eval`, task `mmlu`
Protocol	5-shot, 5 questions per subtask, batch size 1, max. length 2048

Checkpoint. The disentangled factors are stored sharded (one `*.factors.pt` file per linear plus a manifest), together with the per-step tuning trace (cost, entropy, effective rank per matrix); Figure 5 shows the run's cost trajectories for all 96 matrices. The step count was set by our compute budget, not by convergence: longer optimization would likely improve the factors further.

Implementation. The Cayley transform expresses the orthogonal gauge Q through unconstrained parameters that an ordinary optimizer can update freely while keeping Q exactly orthogonal; its backward pass is far cheaper than that of the matrix exponential it replaces, whose autograd gradient requires a Fréchet-derivative computation. Combined with reusing the gradient's SVD within each step and 32-bit precision, this yields the 30–50× per-layer speed-up reported in Section 3.2.

Worked example. For Qwen3.5-4B's `down_proj` (one of the three MLP projection matrices in each decoder layer, $X \in \mathbb{R}^{2560\times 9216}$) the factorization is $(l, r, b, c) = (40, 64, 96, 96)$, so that $40 \times 64 = 2560$ and $96 \times 96 = 9216$ recover the matrix dimensions; then $Q \in O(2560)$, $A(QX) \in \mathbb{R}^{3840\times 6144}$, and $m = 3840$. Baking at ρ=0.25 keeps k=960 singular directions; the stored factors are the gauge together with the singular vectors and values U, s, V_h of $A(Q^*X)$.

Prompting. Problems are formatted with the model's chat template, with the instruction "Please reason step by step, and put your final answer within `\boxed{}`." appended to the problem statement.

Seeds and subsampling. All randomness beyond greedy decoding, i.e., the selection of depth-matched control positions and steps and of free-run subsets, uses Python's `random.Random` with seed 0; free-run subsets are drawn uniformly without replacement via `rng.sample`.

B Learning-rate sweep

Disentanglement quality is sensitive to the optimizer learning rate. We swept $\{5\times10^{-4}, 10^{-3}, 5\times10^{-3}, 7\times10^{-3}, 10^{-2}, 2\times10^{-2}, 5\times10^{-2}\}$ for 200 steps on a representative Qwen3.5-0.8B MLP layer, tracking reconstruction cost (the discarded singular energy of (4), so lower is better) and effective rank (a soft count of how many singular directions carry meaningful energy). Small rates converge slowly; rates $\geq 10^{-2}$ diverge (reconstruction cost increases). A mid-range rate near 10^{-3} minimizes cost while reaching a stable effective rank (Figure 6). All disentanglement runs in the paper use this regime.

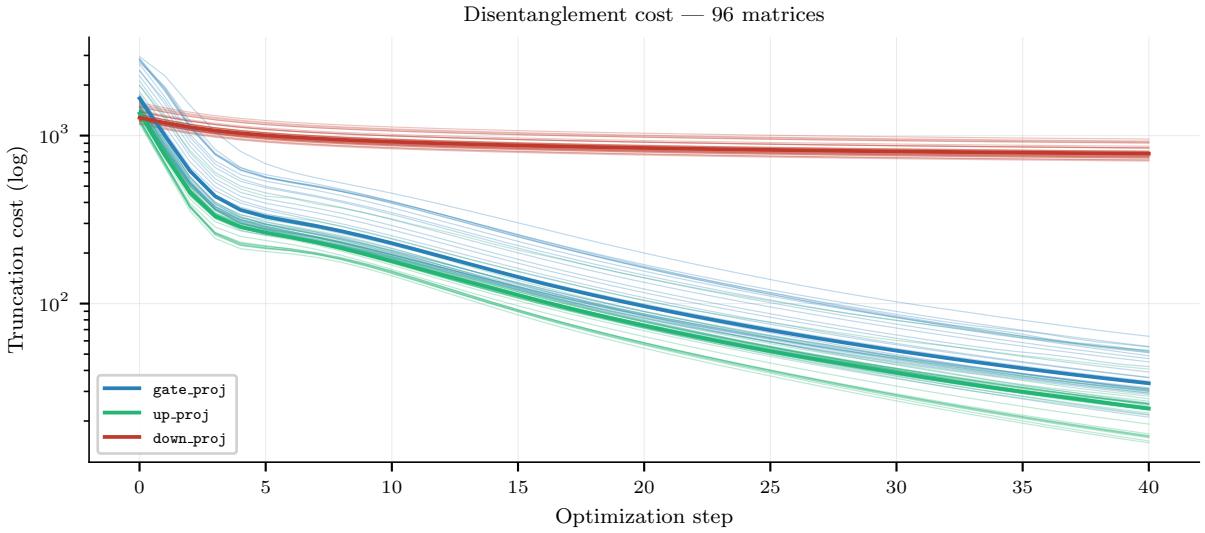

Figure 5: Disentanglement cost per optimization step for all 96 MLP matrices of the published ρ=0.1 run (thin lines: individual matrices; thick lines: per-projection medians). `gate_proj` and `up_proj` compress by more than an order of magnitude, while `down_proj` barely improves: the layer heterogeneity of the gauge advantage (Appendix C) is already visible during optimization. Cost is still decreasing when the run stops at the 40-step compute budget, so the gauges are not fully converged.

C The KL bound and gauge quality across bake ranks

For a policy and its rank-k truncated counterpart, the per-state KL divergence is bounded by the discarded singular tail, $D_{\mathrm{KL}}\big(\pi_M(\cdot \mid s) \,\|\, \pi_N(\cdot \mid s)\big) \le \frac{L}{2} \sum_{i>k} \sigma_i^2\big(A(Q^* X)\big)$, with L a network-wide Fisher-information Lipschitz constant, a fixed factor that converts discarded weight energy into a bound on how much the output distribution can change; at the rank the gauge was optimized for, the optimized gauge yields the tightest such bound (Till, 2026). Two properties of our pipeline determine how this applies to the Novices used in the experiments. First, baking truncates the *stored factors* of $A(Q^* X)$ before inverting the reshape, exactly the construction in (5), so the bound itself holds at every bake rank ρ, with the tail read directly off the stored spectrum. Second, because (4) depends on k, the *tightness* guarantee is rank-specific: a Novice baked at a different ρ than the gauge was optimized for inherits a valid but not provably minimal bound.

Empirically, the gauge's advantage transfers across ranks. Relative to identity-gauge (plain SVD) truncation in the same bipartition, the optimized gauge reduces the discarded tail energy to 0.60× (ρ=0.1) and 0.58× (ρ=0.25) on `layers.0.down_proj`, and to 0.026× and 0.025× on `layers.31.up_proj`. The advantage is large, heterogeneous across layers, and essentially independent of the bake rank, supporting the use of the ρ=0.1-optimized gauge for Novices baked at ρ=0.25 (Section 5.1).

D Statistical tests

This appendix explains the units and tests used in Section 5.2 and why each test matches its design. A *nat* is the natural-logarithm unit of log-probability: a drop of one nat makes a continuation a factor of e less likely. The *Wilson* 95% confidence interval is the range of true rates consistent with the observed counts; it is preferred over the textbook (Wald) interval because it stays well-behaved for proportions near 0 or 1 and for small samples. *McNemar's exact test* suits E1's paired design: it looks only at the pairs whose two arms gave different outcomes (the discordant pairs) and asks whether one arm wins more often than a coin flip would predict. *Fisher's exact test* suits E2 and E3, whose free-run arms are independent groups: it operates on the two-by-two table of arm versus outcome, summarizes the association as an odds ratio, and returns an exact p-value that remains valid at small counts. For regression coefficients, the standard error (SE) quantifies the estimate's uncertainty, and the t-value (estimate divided by SE) counts how many standard errors the estimate lies from zero.

Secondary statistics. Entropy terciles split the perturbed positions into thirds from lowest to highest Master entropy; that the flip>control ordering holds within each tercile is a check that does not rely on the regression's linearity. In E2's free-run subset, Fisher's exact test on the two `rank2` arms gives an odds ratio of 0.46 ($p = 0.47$), numerically opposite to H1's direction, and no pairwise comparison approaches significance. In E3, the Novice's version of a step produced the larger answer drop in only 10% of step pairs.

Power and minimum detectable effects. The power of a test is the probability that it returns $p < 0.05$ when an effect of a given size truly exists; 80% is the conventional reporting benchmark, and its complement is the

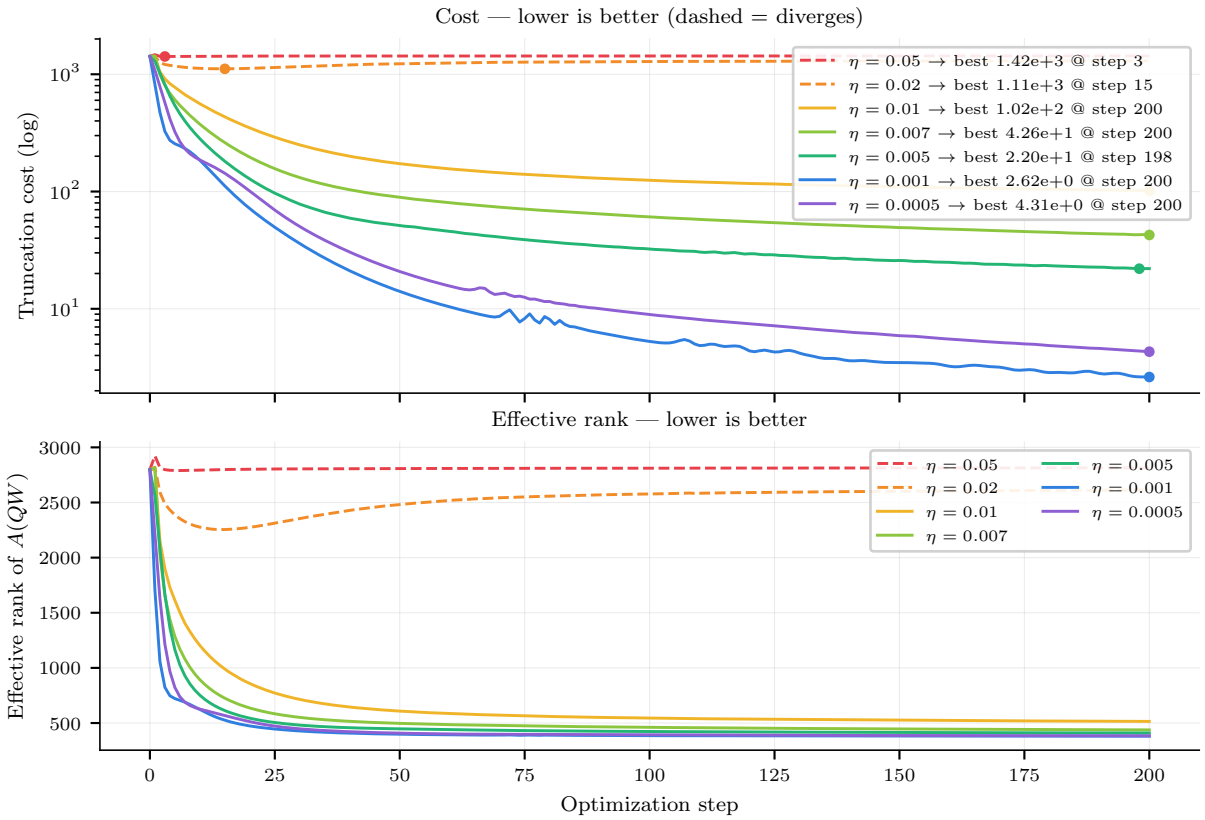

Figure 6: Learning-rate sweep for tensor disentanglement (Qwen3.5-0.8B MLP layer, 200 steps). Mid-range rates balance convergence speed and stability; large rates diverge.

false-negative rate (a test with 80% power misses the stated effect in 20% of repeated experiments). The minimum detectable effect (MDE) quoted in Section 7 is the smallest true effect our sample sizes would detect at least 80% of the time. An observed null is therefore strong evidence against effects at or above the MDE, which would very likely have been seen, and weak evidence against smaller ones, which the test would usually miss. Our MDEs are computed at two-sided $\alpha = 0.05$ with the observed control rates as baseline: for E2 and E3 via the standard two-proportion normal approximation, and for E1's paired design via the exact McNemar test, holding the flip-better pair probability at its observed value (3/115) while the control-better probability grows. Reasonable alternative assumptions shift these numbers modestly; they are calibrated estimates, not exact constants.