FROM GENERATING REASONING TO RELYING ON IT: DIRECT COUNTERFACTUAL OPTIMIZATION FOR CHAIN-OF-THOUGHT

Anonymous authors
Paper under double-blind review

ABSTRACT

A model can give a correct answer and a plausible chain-of-thought (CoT) even when that CoT has little causal influence on the answer. We examine whether training to improve answer accuracy or favor low-redundancy rationales reduces the CoT's causal redundancy with respect to the answer. We assess training effects through complementary behavioral and activation-based probes: CORD and all-layer activation patching, respectively. Outcome-based RLVR improves accuracy, and DPO changes generated CoTs, but neither consistently lowers held-out CORD. This leaves open whether causal redundancy is difficult to change or these training signals fail to reach the relevant answer behavior. We compare trajectory-level scalar rewards with direct gradients on counterfactual answer likelihood. Under the tested protocols, direct optimization substantially lowers CORD, whereas scalar credit does not consistently do so. Naive direct optimization, however, harms normal-task accuracy. We propose ***D**irect **C**ounterfactual **O**ptimization (**DCO**)*, a training objective that discourages premature recovery of the original answer from insufficient reasoning while protecting the model's behavior on intact question–CoT paths. It uses a reference-relative target to limit suppression under early, question-masked CoT prefixes and a KL budget to allow limited changes on intact inputs. Across three held-out mathematical-reasoning sets, DCO achieves matched $\Delta\mathrm{CORD}$ of -0.347 to -0.197, with accuracy changes of -0.6 to $+0.4$ percentage points. All-layer patching shows that DCO and naive direct optimization reduce sensitivity to question replacement and increase sensitivity to CoT replacement, unlike DPO and auxiliary-reward RLVR.

1 INTRODUCTION

Chain-of-thought (CoT) prompting elicits multistep reasoning and has improved performance on a range of reasoning tasks (Wei et al., 2022). Beyond improving accuracy, the generated CoT now serves as an interface for understanding and supervising model decisions (Korbak et al., 2025). Yet answer correctness and the apparent quality of the text do not establish CoT faithfulness. A CoT may contain hallucinations, include unverifiable steps, or fail to reflect the computation that produced the final answer. We focus on the last of these, a phenomenon called *shortcut reasoning*.

In shortcut reasoning, a model reaches an answer through information paths that bypass the visible CoT. A fluent, correct rationale may therefore differ from the reasoning on which the answer actually relies (Lanham et al., 2023; Turpin et al., 2023; Chen et al., 2025). Prior work has examined this gap by disrupting the CoT and observing the answer (Lanham et al., 2023; Aravindan & Kejriwal, 2026; Yu et al., 2026; Kim & Lee, 2026), or by replacing internal states to localize causal contributions (Geiger et al., 2025; Mehrafarin et al., 2026). We measure it with CORD, which quantifies the CoT's *causal redundancy* with respect to the answer by testing whether the original answer survives or can be recovered under two disruptions (Kim & Lee, 2026). All-layer activation patching serves as a complementary probe of how question or CoT replacement changes the answer's likelihood.

Recent work has sought to train desirable reasoning properties by adding auxiliary rewards (Yu et al., 2026) to outcome-based RLVR (which rewards verifiable final-answer correctness) (Lambert et al., 2024; Guo et al., 2025), intervening on update paths (Jia et al., 2026) during RLVR, or learning

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.

trajectory preferences (Paul et al., 2024; Swaroop et al., 2025) with DPO (Rafailov et al., 2023). Yet it remains unclear whether these training interventions also change the answer's reliance on the CoT. *Does improving outcome accuracy or generating more desirable reasoning trajectories also make answers rely more on those trajectories?* If so, outcome- or trajectory-level supervision could reduce shortcut reasoning without directly targeting reliance on the CoT.

Section 4 first tests whether outcome improvements and rationale-level changes make answers rely more on the CoT. Outcome RLVR raises Qwen2.5-7B accuracy by 10.1 and 13.5 percentage points (%p) on GSM8K and MATH500, respectively, but does not consistently lower held-out matched $\Delta\mathrm{CORD}$ on items both policies answer correctly. DPO, trained to prefer correct rationales with low proxy CORD, substantially changes the generated reasoning, yet matched $\Delta\mathrm{CORD}$ remains near zero across evaluation sets. These results show that outcome improvement, rationale redistribution, and the answer's reliance on the CoT can come apart. They leave open whether causal redundancy is difficult to change or the tested training interfaces provide insufficient access to the target behavior.

Section 5 tests whether causal redundancy is difficult to change or whether the earlier training signals fail to reach the relevant answer behavior. We train toward the same class of counterfactual targets through two routes: a scalar reward on sampled trajectories or a direct gradient on answer likelihood under disrupted CoT contexts. Under the tested protocols, the reward route does not consistently reduce held-out CORD, whereas both direct variants produce large reductions. On GSM8K, for example, the premature-recovery variant yields matched $\Delta\mathrm{CORD} = +0.002$ with scalar credit versus -0.209 with direct optimization. The target can therefore move, but naive direct optimization also harms normal-task accuracy.

Section 6 addresses this trade-off with *Direct Counterfactual Optimization* (DCO), which bounds premature answer recovery from insufficient reasoning while protecting behavior on normal inputs. Across the three held-out evaluation sets, DCO achieves matched $\Delta\mathrm{CORD}$ of -0.347 to -0.197, with accuracy changes of -0.6 to $+0.4$%p. Section 7 uses all-layer activation patching to test whether this route-level difference extends beyond the disruptions in CORD by substituting donor content for the question or generated CoT and scoring the original answer. Across all held-out evaluation sets, DCO is less sensitive to question substitution and more sensitive to CoT substitution than outcome-only RLVR. CoT substitution has a 3.120 to 3.522 larger mean absolute effect on answer log-likelihood. Naive direct optimization shows the same directional pattern, unlike DPO and auxiliary-reward RLVR.

Our contributions are fourfold. (1) We show at the policy level that neither outcome-RLVR accuracy gains nor DPO-induced changes in generated rationales guarantee lower causal redundancy. (2) By comparing reward-mediated and direct conditional routes for matched counterfactual targets, we separate the target's movability from the access supplied by the training objective. (3) We introduce DCO, which combines bounded counterfactual suppression with normal-behavior protection and achieves matched $\Delta\mathrm{CORD}$ of -0.347 to -0.197, with accuracy changes of -0.6 to $+0.4$%p, across GSM8K, MATH500, and DAPO-test. (4) All-layer activation patching shows that naive direct optimization and DCO produce smaller answer-score changes when the question is replaced and larger changes when the CoT is replaced; DPO and auxiliary-reward RLVR do not show this pattern.

2 Related Work

2.1 Measuring CoT Faithfulness and Causal Reliance

Although CoT provides a useful interface for interpreting and supervising model decisions (Wei et al., 2022; Korbak et al., 2025), the generated text need not reflect how the model arrived at its answer (Lanham et al., 2023; Turpin et al., 2023; Chen et al., 2025). We focus on shortcut reasoning, where the answer bypasses the visible CoT. Related measures take two broad approaches.

Behavioral measures truncate, corrupt, or paraphrase the CoT and measure changes in the answer (Lanham et al., 2023; Aravindan & Kejriwal, 2026). Causal Redundancy of Reasoning via Disruption (CORD) defines shortcut tendency as the CoT's causal redundancy with respect to the answer and combines two disruptions to reduce the ambiguity of a single-axis test (Kim & Lee, 2026). Causal Importance of Reasoning (CIR) measures the effect of CoT truncation on the answer

distribution (Yu et al., 2026). Other behavioral studies examine whether the CoT reflects input biases or hints that affect the answer (Turpin et al., 2023; Chen et al., 2025).

Activation-based measures instead intervene on internal representations. Activation patching can localize causal contributions by replacing selected hidden states and measuring the effect on output (Geiger et al., 2025). For example, Mehrafarin et al. (2026) patched CoT-token states into direct-answer runs to study answer recovery and token–layer effects. Our all-layer activation-patching experiment asks a different question: how replacing question or CoT content changes the answer score across training routes.

2.2 Training for CoT Faithfulness: Targets and Optimization Access

Outcome-based RLVR rewards verifiable final-answer correctness but leaves the path to that answer unspecified (Lambert et al., 2024; Guo et al., 2025). To shape that path, Yu et al. (2026) added auxiliary rewards based on reasoning importance and sufficiency, and Jia et al. (2026) constrained attention and gradient paths. DPO (Rafailov et al., 2023) offers another training interface for preferences between reasoning trajectories (Paul et al., 2024; Swaroop et al., 2025).

These studies target different aspects of reasoning, from final-answer correctness to preferred trajectories and training pathways. Improvements in those objectives do not necessarily establish that an answer relies on its generated CoT: a policy could produce a correct answer and a desirable rationale while reaching that answer through another path. Whether these interventions change the answer's causal reliance on the CoT therefore needs to be tested directly.

3 Problem Setup and Evaluation

This section defines the policies and interventions, introduces the behavioral measures and an all-layer activation-patching analysis, and specifies the evaluation protocol shared by Sections 4–7.

Notation. Let X, M, and Y denote the input, generated CoT, and answer, with realizations x, m, and y; θ denotes model parameters. The ground-truth answer is y^*, and y_0 is the policy's answer before intervention. Thus $y_0 = y^*$ on correctly answered items in paired analyses. We denote a question-masked input by $X_\varnothing$ and its realization by $x_\varnothing$. The first $p\%$ of a realized CoT is $m_{\leq p}$, while $\tilde{m}_p$ has its suffix after $p\%$ perturbed. Finally, π_{base} and π_{RLVR} denote the base and outcome-RLVR policies.

3.1 Measuring Causal Redundancy

Neither answer correctness nor the appearance of a generated rationale reveals whether the answer relies on that rationale (Lanham et al., 2023; Turpin et al., 2023; Chen et al., 2025). CORD is our primary behavioral outcome, with CIR as a secondary measure. We also use all-layer activation patching to compare how question and CoT substitutions change the original answer's likelihood. Full definitions and settings for the behavioral measures appear in Appendix A.

Primary behavioral measure: CORD. CORD measures the CoT's causal redundancy with respect to the final answer using two disruptions (Kim & Lee, 2026). The A-axis asks whether the original answer survives corruption of the CoT suffix when the question remains visible. The B-axis asks how early that answer can be recovered from a CoT prefix when the question is masked. A single disruption cannot readily distinguish bypass paths through information retained in the question or the CoT. Their harmonic mean is therefore high only when both bypass paths are strong. Let $a(p; x)$ and $b(p; x)$ indicate whether the original answer is produced under the A- and B-axis interventions, respectively. If $p^*(x) = \min(\{p \in \mathcal{P}_B : b(p; x) = 1\} \cup \{100\})$ is the first B-axis recovery position, then

$$\bar{a}(x) = |\mathcal{P}_A|^{-1} \sum\nolimits_{p \in \mathcal{P}_A} a(p; x), \quad \bar{b}(x) = 1 - p^*(x)/100, \quad \text{CORD}(x) = H(\bar{a}(x), \bar{b}(x)). \quad (1)$$

Here $H(u, v) = 2uv/(u + v)$, with $H(0, 0) = 0$. For full evaluation, the intervention positions, as percentages of the CoT, are $\mathcal{P}_A = \{0, 20, 40, \ldots, 100\}$ and $\mathcal{P}_B = \{0, 5, 10, \ldots, 100\}$. The reward-training proxy uses a coarser B-axis grid. Higher CORD means that the original answer survives

or is recovered readily under both disruptions; lower CORD is the desired direction. Throughout, *causal redundancy* refers to the construct measured by CORD. We use *causal reliance on the CoT* for the intuitive opposite direction, with the answer as the subject of that reliance.

Low-cost proxy measurements support DPO pair construction and reward training; all held-out ΔCORD results use the full protocol. Appendix A Table 6 reports a DPO control that replaces proxy-based pair scoring with full evaluation.

Secondary behavioral measure: CIR. CIR measures how CoT truncation changes the ground-truth answer probability relative to the full-CoT context, with the question left intact (Yu et al., 2026). Higher values indicate greater sensitivity to missing reasoning. Writing $r_p = p_\theta(y^* \mid x, m_{\le p})$ and $r_{100} = p_\theta(y^* \mid x, m)$ for the correct-answer probabilities under the prefix and full CoT, respectively, gives

$$\mathrm{CIR}(x) = |\mathcal{P}_{\mathrm{CIR}}|^{-1} \sum\nolimits_{p \in \mathcal{P}_{\mathrm{CIR}}} \mathrm{JS}(\mathrm{Ber}(r_p) \,\|\, \mathrm{Ber}(r_{100})). \tag{2}$$

Here $\mathcal{P}_{\mathrm{CIR}} = \{10, 20, \ldots, 90\}$ contains the nine truncated CoT percentages. $\mathrm{Ber}(r)$ is the Bernoulli distribution for whether the answer equals y^*, and JS is the Jensen–Shannon divergence. The nine-prefix grid follows the coarse percentile-based approximation used for CIR rewards by Yu et al. (2026), rather than their token-level definition.

All-layer activation patching. To test whether training effects extend beyond the disruptions in CORD, we replace either the question or the generated CoT with a donor span and measure the change in the original answer's log-likelihood, $|\Delta_P|$ or $|\Delta_C|$. The all-layer patching implementation is equivalent to evaluating the donor-substituted sequence: it measures sensitivity to each source of content, not the location of an internal computation. Appendix E.1 gives the protocol and controls.

3.2 Experimental Setup and Comparisons

Models and data. The main experiments use Qwen2.5-7B (Qwen et al., 2024). Starting from π_{base}, we obtain π_{RLVR} by applying GRPO with an outcome reward (Shao et al., 2024) on data from DAPO-Math-17K (Yu et al., 2025). Evaluation uses 1,000 held-out DAPO validation items, 1,000 DAPO test items, 1,319 GSM8K items (Cobbe et al., 2021), and 500 MATH500 items (Hendrycks et al., 2021; Lightman et al., 2023). Generation and answer-extraction settings are fixed across arms.

Evaluation estimands. For a comparison $\pi_a \to \pi_b$, we compute $\Delta\mathrm{acc} = \mathrm{acc}(\pi_b) - \mathrm{acc}(\pi_a)$ over all evaluation items. Matched ΔCORD averages itemwise policy differences only where both policies answer correctly and both measurements are valid. It therefore measures redundancy changes among questions both policies solve, while $\Delta\mathrm{acc}$ captures changes in the fraction solved. Arrows in the tables indicate the desired directions: higher accuracy and lower redundancy. Main-text results are means $\pm$ sample standard deviations over three training seeds. A * marks a paired item-bootstrap 95% interval excluding zero; in three-seed rows, all three intervals must exclude zero in the same direction. Appendix A.3 details the estimands and bootstrap procedure.

Comparison design. Each intervention is assessed relative to its training-stage control. Outcome RLVR measures the transition $\pi_{\mathrm{base}} \to \pi_{\mathrm{RLVR}}$; DPO and direct/DCO measure changes from π_{RLVR}. Auxiliary-reward arms are compared with outcome-only π_{RLVR}, sharing the same π_{base} initialization and GRPO protocol. Table 4 in Appendix A summarizes the design.

4 Outcome and Rationale Optimization Do Not Determine Causal Reliance

We first ask: *Do improvements in outcomes or changes in generated rationales determine answer-level causal reliance?* If so, outcome-based RLVR or DPO with preference supervision should change causal reliance.

Table 1: Effects of outcome RLVR and DPO (Qwen2.5-7B). Each Δ is the right policy minus the left policy in the group heading. Ident.(%) is the percentage of items on which the greedy CoTs of π_{RLVR} and DPO are token-identical. Δacc is in percentage points (%p); matched $\Delta\mathrm{CORD}$ is the paired difference on items both policies answer correctly.

	Outcome RLVR ($\pi_{\mathrm{base}} \to \pi_{\mathrm{RLVR}}$)		DPO ($\pi_{\mathrm{RLVR}} \to \pi_{\mathrm{DPO}}$)		
Task	Δacc $\uparrow$	$\Delta\mathrm{CORD}\downarrow$	Δacc $\uparrow$	$\Delta\mathrm{CORD}\downarrow$	Ident.
GSM8K	$+10.1_{\pm 0.2}{}^{*}$	$+0.003_{\pm 0.005}$	$-0.2_{\pm 0.2}$	$-0.005_{\pm 0.0004}{}^{*}$	$31.8_{\pm 2.2}$
MATH500	$+13.5_{\pm 0.5}{}^{*}$	$+0.022_{\pm 0.004}$	$-0.4_{\pm 1.1}$	$-0.001_{\pm 0.007}$	$24.5_{\pm 0.4}$
DAPO-val	$+3.6_{\pm 0.6}{}^{*}$	$+0.012_{\pm 0.009}$	$-0.4_{\pm 0.7}$	$-0.004_{\pm 0.005}$	$10.2_{\pm 0.5}$
DAPO-test	$+5.9_{\pm 1.5}{}^{*}$	$+0.004_{\pm 0.010}$	$0.0_{\pm 0.3}$	$+0.004_{\pm 0.006}$	$10.2_{\pm 0.2}$

4.1 Outcome RLVR Does Not Reliably Reduce Causal Redundancy

The left columns of Table 1 compare the base policy π_{base} with the outcome-trained policy π_{RLVR}. Accuracy increases substantially, but matched $\Delta\mathrm{CORD}$ does not consistently fall on items both policies answer correctly: it is $+0.003_{\pm 0.005}$ on GSM8K and $+0.022_{\pm 0.004}$ on MATH500. Better task performance therefore does not reliably translate into greater reliance on the CoT. (Appendix A.4 reports secondary behavioral measures for the same RLVR and DPO comparisons.) This raises a question: *Why does successful outcome optimization not increase causal reliance on the reasoning?*

4.2 Outcome Optimization Underdetermines the Reasoning Pathway

The probability of a correct answer y^* given X marginalizes over reasoning paths M[1]:

$$P_\theta(Y = y^* \mid X) = \sum\nolimits_m P_\theta(M = m \mid X)\, P_\theta(Y = y^* \mid X, M = m). \tag{3}$$

Outcome RLVR rewards correct answers, but does not specify whether an increase in the marginal success probability comes from (i) shifting probability mass toward reasoning paths favorable to the correct answer, (ii) changing the answer conditional given a reasoning context, or (iii) both. The outcome reward alone therefore does not tell us which of these changes would reduce causal redundancy.

We first probe the rationale-distribution route: if generating low-redundancy rationales more often is sufficient, directly preferring such rationales should reduce policy-level CORD.

4.3 Changing Generated Rationales Does Not Guarantee Causal Reliance

Preference pairs and DPO training. We construct within-problem preference pairs that differ in rationale redundancy while holding the question and correctness fixed. For each of 3,000 DAPO training problems, we sample 16 traces from π_{RLVR} (temperature 1.0, top-p 0.95). Among correct traces, those with proxy $\mathrm{CORD} < 0.2$ are chosen and those with $\mathrm{CORD} \geq 0.2$ are rejected. Requiring a gap of at least 0.2 and retaining at most four pairs per problem yields 816 pairs from 506 problems, split into 649 training and 167 evaluation pairs. The chosen and rejected traces have mean proxy CORD values of 0.061 and 0.481, respectively, but similar mean lengths (665 and 652 tokens). The pairs thus provide a preference signal aligned with redundancy rather than correctness or a large length difference. We train DPO on these pairs and evaluate whether favoring low-redundancy rationales reduces held-out policy-level CORD. Training details appear in Appendix B.1.

DPO changes reasoning text without consistently reducing CORD. DPO substantially changes the generated reasoning: the greedy CoT remains token-identical to that of π_{RLVR} on only 31.8% of GSM8K items, 24.5% of MATH500 items, and 10.2% of DAPO-test items (Table 1). Despite this textual change, matched $\Delta\mathrm{CORD}$ is only $-0.005_{\pm 0.0004}$ on GSM8K, $-0.001_{\pm 0.007}$ on MATH500, $-0.004_{\pm 0.005}$ on DAPO-val, and $+0.004_{\pm 0.006}$ on DAPO-test. The separation between training pairs in proxy CORD does not yield a consistent change in held-out policy-level CORD.

[1]This is not a parameter decomposition. The two factors share θ, so one training objective can change both.

The policy does respond to DPO training: preference accuracy on training pairs reaches 73–77%, and proxy CORD on DPO rollouts shifts. Neither held-out pair preference nor causal reliance on the four evaluation sets shows a corresponding consistent change (Appendix B.1). The pattern extends to Qwen 3B/14B, OLMo-7B/13B, and Llama-8B (Appendix B.3; Table 11). For Qwen 14B, the three-seed outcome-RLVR effect is $-0.028_{\pm 0.005}$ on the full matched set but reverses to $+0.008_{\pm 0.004}$ after length matching; subsequent DPO remains near zero at $-0.000_{\pm 0.001}$. Starting from band-filtered RLVR checkpoints, DPO yields GSM8K matched ΔCORD of $+0.004_{\pm 0.001}$ for OLMo-7B and $+0.000_{\pm 0.003}$ for OLMo-13B. No tested family shows a consistent decrease.

These results establish a narrower distinction: outcome improvement and rationale redistribution do not determine answer-level causal reliance. In the tested setting, trace-level preference supervision changes generated rationales substantially without reliably reducing causal redundancy.

5 Testing Target Movability Through Gradient Routes

Section 4 leaves an open question: *Is causal redundancy itself difficult to change, or did the preceding objectives provide insufficient optimization access to the target behavior?* We use *optimization access* to describe how target information enters the gradient.

We compare two ways of optimizing the same counterfactual behavior. The **reward-mediated route** assigns trajectory-level scalar credit based on that behavior. The **direct conditional route** differentiates answer likelihood in the intervention context. If the target itself resists optimization, neither route should move it substantially. If optimization access is the bottleneck, directly differentiating the target conditional should move the behavior even when scalar credit does not produce a consistent change.

5.1 A Diagnostic Comparison of Two Gradient Routes

Both routes address the A-axis answer retention and B-axis premature recovery defined in Section 3. The routes enter at different training stages. Reward-mediated arms start from π_{base} and add counterfactual credit during outcome RLVR; direct arms start from π_{RLVR} and optimize the counterfactual target after outcome training. Accordingly, we measure each route's held-out change against its own training-stage control.

Reward-mediated route: trajectory-level scalar credit. The reward-mediated route adds counterfactual utility to the outcome reward and uses the resulting detached advantage to weight each sampled trajectory. It tests whether target-aligned scalar credit changes policy formation during RLVR. For $v \in \{A, B, A+B\}$, let $c(\tau)$ indicate correctness and hats denote in-loop proxy measurements. The reward is

$$R_v(\tau) = c(\tau)\big[1 + \lambda_{\text{rew}} u_v(\tau)\big]. \tag{4}$$

Here $u_A = 1 - \widehat{a}$, $u_B = \widehat{p}^*/100$, and $u_{A+B} = 1 - \widehat{\text{CORD}}$. The counterfactual quantity affects the policy update through scalar credit for the entire trajectory.

Direct route: target-conditional gradient. The direct route uses fixed correct CoT–answer triples from π_{RLVR} to optimize the original answer's likelihood under A- and B-axis interventions. For a triple (x_i, m_i, y_i) and $v \in \{A, B\}$, let p denote the percentage of the CoT retained before suffix corruption (A) or supplied as a prefix (B). The intervention contexts and token-normalized answer score are

$$h^A_{i,p} := (x_i, \tilde{m}_{i,p}),\ h^B_{i,p} := (x_{i,\varnothing}, m_{i,\le p}),\ s^v_{\theta,i,p} := |y_i|^{-1} \sum_{j=1}^{|y_i|} \log p_\theta(y_{i,j} \mid h^v_{i,p}, y_{i,<j}). \tag{5}$$

For a minibatch $\mathcal{B}$, let $\langle\cdot\rangle_{\mathcal{B}}$ denote the item mean, and write $s_\theta(y_i \mid h)$ for the same score in a general context h. Define $S^{(v)}_{\text{cf}} = \langle s^v_{\theta,i,p_i}\rangle_{\mathcal{B}}$, $S_{\text{clean}} = \langle s_\theta(y_i \mid x_i, m_i)\rangle_{\mathcal{B}}$, and $S_{\text{rec}} = \langle s_\theta(y_i \mid x_{i,\varnothing}, m_i)\rangle_{\mathcal{B}}$. The first scores the intervened context; the latter two score the full CoT with the question visible and masked, respectively. We minimize

$$\mathcal{L}^{(v)}_{\text{dir}} = \lambda_{\text{cf}}[S^{(v)}_{\text{cf}} - f]_+ - S_{\text{clean}} - \mathbb{1}[v = B] S_{\text{rec}}. \tag{6}$$

Table 2: Effects relative to π_{RLVR} by training stage and gradient route. A and B target answer retention and premature answer recovery, respectively; DCO uses the B-axis. Δacc is in percentage points (%p), and $\Delta\mathrm{CORD}$ is the paired difference on items answered correctly by both the policy and π_{RLVR}.

	GSM8K		MATH500		DAPO-test	
Method	Δacc ↑	ΔCORD ↓	Δacc ↑	ΔCORD ↓	Δacc ↑	ΔCORD ↓
① *Post-training: rationale-level preference*						
DPO	$-0.2_{\pm 0.2}$	$-0.005_{\pm 0.0004}{}^{*}$	$-0.4_{\pm 1.1}$	$-0.001_{\pm 0.007}$	$0.0_{\pm 0.3}$	$+0.004_{\pm 0.006}$
② *Policy formation: trajectory-level scalar credit*						
Scalar credit (A)	$-0.1_{\pm 0.4}$	$+0.001_{\pm 0.006}$	$+0.7_{\pm 0.8}$	$+0.002_{\pm 0.001}$	$+1.6_{\pm 1.1}$	$+0.009_{\pm 0.004}$
Scalar credit (B)	$-0.3_{\pm 0.5}$	$+0.002_{\pm 0.012}$	$+0.3_{\pm 0.7}$	$-0.007_{\pm 0.013}$	$+0.2_{\pm 0.8}$	$+0.009_{\pm 0.009}$
Scalar credit (A+B)	$-0.3_{\pm 0.4}$	$+0.000_{\pm 0.014}$	$-0.5_{\pm 2.4}$	$-0.010_{\pm 0.014}$	$+0.9_{\pm 0.2}$	$+0.009_{\pm 0.011}$
③ *Post-training: target-conditional gradient*						
Naive direct (A)	$-6.4_{\pm 2.4}{}^{*}$	$-0.222_{\pm 0.005}{}^{*}$	$-5.4_{\pm 2.5}$	$-0.291_{\pm 0.006}{}^{*}$	$-4.9_{\pm 1.0}{}^{*}$	$-0.170_{\pm 0.009}{}^{*}$
Naive direct (B)	$-3.7_{\pm 0.6}{}^{*}$	$-0.209_{\pm 0.006}{}^{*}$	$-2.6_{\pm 1.0}$	$-0.236_{\pm 0.002}{}^{*}$	$-2.3_{\pm 0.2}$	$-0.141_{\pm 0.011}{}^{*}$
DCO (B)	$-0.6_{\pm 0.3}$	$-0.300_{\pm 0.002}{}^{*}$	$+0.3_{\pm 0.4}$	$-0.347_{\pm 0.007}{}^{*}$	$+0.4_{\pm 0.4}$	$-0.197_{\pm 0.018}{}^{*}$

Here $[z]_+ = \max(z, 0)$. The hinge suppresses the original answer only while its minibatch-mean counterfactual score exceeds the fixed floor f. The other terms encourage that answer when the full CoT is available. Appendix C.1 expands the loss.

5.2 Direct Gradients Move the Target Behavior

Training data. Reward-mediated arms generate new CoT–answer trajectories from the current policy on DAPO training prompts during GRPO and compute counterfactual bonuses on those rollouts. For the direct arms, we greedily generate one trace per problem from π_{RLVR} on 3,000 DAPO training problems. Retaining correct traces that do not reach the generation cap yields 767 fixed CoT–answer triples, on which we optimize answer likelihood by teacher forcing. Appendix C.1 gives the intervention grids and training hyperparameters.

Direct gradients consistently reduce CORD; scalar credit does not. Table 2 contrasts routes ② and ③ for the same A/B target variations. For the B-axis variation, three-seed mean matched $\Delta\mathrm{CORD}$ ranges from $-0.007_{\pm 0.013}$ to $+0.009_{\pm 0.009}$ under scalar credit, versus $-0.236_{\pm 0.002}$ to $-0.141_{\pm 0.011}$ under direct optimization across the three tasks in the table. The A-axis shows the same pattern. Appendix C checks the result against several alternatives: learning trajectories for the reward bonuses, a detached scalar analogue of the direct B loss, CIR/SR auxiliary rewards, and structural interventions. None yields a consistent reduction in held-out CORD across evaluation tasks. Causal redundancy is therefore movable, and under these protocols the naive direct objectives produce much larger held-out changes than trajectory-level scalar credit, albeit with lower normal-task accuracy.

6 Direct Counterfactual Optimization

Section 5 leaves a design question: *How can we retain direct access to the target while limiting damage to normal-task accuracy?* Direct Counterfactual Optimization (DCO) treats this as bounded policy editing. Its **counterfactual suppression** term changes answer behavior when the CoT is insufficient, while **normal-behavior protection** limits acceptable policy drift on the original question.

6.1 A Bounded Direct Objective

The naive B-axis loss reduces premature recovery at a lower accuracy cost than the A-axis loss (Table 2). DCO therefore uses the B-axis context $h^B_{i,p}$ and answer score $s^B_{\theta,i,p}$ from Equation 5, restricted to early prefixes $p \in \{0, 20, 40, 60\}$. For each fixed correct triple (x_i, m_i, y_i), the reference

score $s^B_{\mathrm{ref},i,p}$ evaluates the same quantity under π_{RLVR}. We replace the shared floor f with an item- and prefix-specific target $s^B_{\mathrm{ref},i,p} - \delta$.

To protect behavior on intact inputs, DCO limits drift from π_{RLVR} along the original question, unmodified reference CoT, and reference answer. Let $\mathcal{W}_i$ contain positions in a sampled reference-CoT window and all answer content-token positions, and let $c_{i,t}$ be the intact, teacher-forced context before position t. We measure drift by averaging the next-token KL over these positions in minibatch $\mathcal{B}$:

$$D_{\mathrm{normal}}(\theta;\mathcal{B}) := \Big(\sum_{i\in\mathcal{B}}|\mathcal{W}_i|\Big)^{-1}\sum_{i\in\mathcal{B}}\sum_{t\in\mathcal{W}_i} D_{\mathrm{KL}}(p_{\mathrm{ref}}(\cdot \mid c_{i,t}) \,\|\, p_\theta(\cdot \mid c_{i,t}))\,. \tag{7}$$

The DCO objective combines the shifted answer-score target with a budget on this normal-path KL:

$$\mathcal{L}_{\mathrm{DCO}}(\theta) = \underbrace{\frac{1}{2}\mathbb{E}_{i,p}\big[\rho_\kappa\big(s^B_{\theta,i,p} - (s^B_{\mathrm{ref},i,p} - \delta)\big)\big]}_{\text{counterfactual suppression}} + \underbrace{\lambda\,[D_{\mathrm{normal}}(\theta;\mathcal{B}) - \epsilon]_+}_{\text{normal-behavior protection}} \tag{8}$$

The Huber term moves the answer score toward $s^B_{\mathrm{ref},i,p} - \delta$ and opposes further suppression once that target is passed. The KL term activates only when the minibatch-average drift exceeds ϵ. Both are teacher-forced quantities computed on fixed training triples; neither CORD nor p^* nor another generation-based proxy is computed during training. Appendix D.1 gives the Huber penalty and token-sampling details.

Training triples and selection. DCO uses the same 767 fixed triples as the direct arms in Section 5. It starts from π_{RLVR} and trains for three LoRA epochs. Appendix D.2 details the search and validation-set selection analysis.

DCO improves both accuracy and CORD over naive direct optimization. In Table 2 (route ③), DCO outperforms naive direct optimization on both accuracy and CORD reduction across the three final evaluation tasks. Relative to π_{RLVR}, observed accuracy changes range from $-0.6_{\pm 0.3}$ to $+0.4_{\pm 0.4}$%p, while matched $\Delta\mathrm{CORD}$ ranges from $-0.347_{\pm 0.007}$ to $-0.197_{\pm 0.018}$. Appendix F illustrates this contrast with two held-out examples in which all five policies give the same correct answer and nearly identical CoTs, yet their CORD scores differ.

Removing either safeguard causes double-digit accuracy losses. We compare DCO with three variants (Appendix D, Table 13). *No protection* removes the normal-path KL term; *Unbounded* replaces shifted Huber suppression with linear suppression while retaining the KL budget; and *Plain KL* retains shifted suppression but applies the KL penalty without a budget. While DCO changes accuracy by only -0.6 to $+0.4$%p, No protection reduces it by 10.6–27.5%p and Unbounded by 14.3–32.9%p across the three held-out sets. Under Unbounded, no MATH500 items remain eligible for matched CORD. Plain KL keeps accuracy near the reference, but its mean matched CORD reductions are 0.017–0.028 smaller than DCO's on variant-specific common-correct cohorts. The ablations reveal two distinct failures: bounded suppression without KL and KL protection with unbounded suppression both damage accuracy, while KL without a budget preserves accuracy but limits the measured CORD reduction. DCO avoids these failures by pairing a bounded counterfactual target with a KL budget that tolerates limited change on the normal path.

7 All-Layer Activation Patching of Question and CoT Spans

Section 6 leaves a further question: *Does the change in* CORD *extend beyond its CoT-corruption and question-masking interventions?* A reduction in CORD alone does not tell us how the original answer responds to other changes in question or CoT content. We therefore use all-layer activation patching: starting from the original question and generated CoT, we replace one span at a time with content from another problem and measure the effect on the original answer's likelihood.

For each policy, we retain its own generated CoT on the original question. We substitute either the question span or CoT span with a length-matched donor span and keep the other span's tokens and teacher-forced gold answer fixed. The implementation patches donor-induced states at the embedding and every decoder block; for this all-layer intervention, the score equals that of the donor-substituted sequence. For $S \in \{P, C\}$, let $|\Delta_S|$ be the absolute change in the original answer's

Table 3: All-layer activation patching by training route. Values are paired differences on items answered correctly by each arm and the fixed seed-42 π_{RLVR}. The RLVR seed control uses two separately trained seeds.

Method	GSM8K	MATH500	DAPO-val	DAPO-test
(a) $\Delta^{P}_{\mathrm{patch}}$; lower indicates less sensitivity to question substitution				
RLVR seed control	$-0.0016_{\pm 0.0102}$	$-0.0017_{\pm 0.0007}$	$-0.0051_{\pm 0.0044}$	$-0.0044_{\pm 0.0040}$
DPO	$+0.0048_{\pm 0.0021}$	$+0.0038_{\pm 0.0024}$	$+0.0027_{\pm 0.0044}$	$+0.0048_{\pm 0.0022}$
Auxiliary-reward RLVR (B)	$-0.0051_{\pm 0.0103}$	$-0.0051_{\pm 0.0059}$	$-0.0066_{\pm 0.0068}$	$+0.0081_{\pm 0.0057}$
Naive direct (B)	$-0.1057_{\pm 0.0007}^{*}$	$-0.0658_{\pm 0.0058}^{*}$	$-0.0815_{\pm 0.0033}^{*}$	$-0.0720_{\pm 0.0021}^{*}$
DCO	$-0.0373_{\pm 0.0240}^{*}$	$-0.0382_{\pm 0.0110}^{*}$	$-0.0516_{\pm 0.0156}^{*}$	$-0.0267_{\pm 0.0105}$
(b) $\Delta^{C}_{\mathrm{patch}}$; higher indicates more sensitivity to CoT substitution				
RLVR seed control	$+0.063_{\pm 0.051}$	$+0.087_{\pm 0.032}$	$+0.068_{\pm 0.018}$	$+0.033_{\pm 0.013}$
DPO	$-0.063_{\pm 0.008}^{*}$	$+0.010_{\pm 0.010}$	$+0.011_{\pm 0.043}$	$-0.072_{\pm 0.012}$
Auxiliary-reward RLVR (B)	$+0.046_{\pm 0.045}$	$+0.082_{\pm 0.067}$	$+0.102_{\pm 0.042}$	$+0.029_{\pm 0.011}$
Naive direct (B)	$+3.098_{\pm 0.165}^{*}$	$+3.091_{\pm 0.061}^{*}$	$+3.254_{\pm 0.194}^{*}$	$+3.578_{\pm 0.158}^{*}$
DCO	$+3.522_{\pm 0.316}^{*}$	$+3.120_{\pm 0.395}^{*}$	$+3.351_{\pm 0.388}^{*}$	$+3.365_{\pm 0.312}^{*}$

token-averaged log-likelihood after substituting span S. For policy π_a, we report

$$\Delta^{S}_{\mathrm{patch}}(a) := E_{\pi_a}[|\Delta_S|] - E_{\pi_{\mathrm{RLVR}}}[|\Delta_S|], \qquad S \in \{P, C\} \tag{9}$$

where each expectation uses the same items answered correctly by both π_a and π_{RLVR}. Each policy begins with its own generated CoT, and the paired difference compares its response to the same type of substitution on those items. Appendix E.1 details the donors, all-layer implementation, and validation controls. Thus $\Delta^{P}_{\mathrm{patch}} < 0$ means a smaller answer-score change under question substitution than for π_{RLVR}, while $\Delta^{C}_{\mathrm{patch}} > 0$ means a larger change under CoT substitution. Table 3 reports these effects on four evaluation sets; matched item sets differ by arm.

Direct optimization changes sensitivity to question and CoT substitution. Table 3 shows an opposite-direction change under both direct objectives. Naive direct B reduces question-span sensitivity on every task ($\Delta^{P}_{\mathrm{patch}} = -0.106$ to -0.066) while increasing CoT-span sensitivity ($\Delta^{C}_{\mathrm{patch}} = +3.091$ to $+3.578$). DCO shows the same pattern: a smaller question-span change (-0.052 to -0.027) and a comparable CoT-span increase ($+3.120$ to $+3.522$) despite its normal-behavior protection. The RLVR seed control, DPO, and auxiliary-reward RLVR show much smaller changes on both axes. Direct optimization therefore changes the answer's response to these two sources in different directions, rather than making it uniformly more sensitive to replacement.

The same-problem CoT control tests whether the larger CoT-span effect is a uniform response to replacing a trace. DCO's increase relative to π_{RLVR} is $+0.276$ to $+0.417$ when the replacement CoT addresses the same question, much smaller than the between-problem increase of $+3.120$ to $+3.522$ (Appendix E.2). Thus, the routes that substantially reduce CORD also increase CIR (Appendix A.4) and alter answer-score sensitivity under distinct donor substitutions, including a same-problem control (Appendix E.2). DCO also requires substantially less training compute than reward-based GRPO in our implementation (Appendix A.5).

8 Conclusion

Correct answers and changed reasoning traces did not, by themselves, imply lower causal redundancy in our experiments. Outcome RLVR improved accuracy and DPO changed generated CoTs, yet neither consistently lowered held-out CORD. Direct optimization showed that the measured behavior can change, while also exposing a cost to normal-task accuracy. DCO reduced that cost and retained substantial CORD reductions. Under a separate question-and-CoT replacement intervention, all-layer activation patching found a related difference between the direct objectives and the other training routes. Together, these results show why a training objective's desired target and its access to the answer behavior should be examined separately.

AI USE STATEMENT

We used generative AI tools to polish the manuscript's language and to write and revise code. We supervised and verified all AI-assisted work and all reported results, and take responsibility for the final manuscript and artifacts.

REPRODUCIBILITY STATEMENT

The main text and appendix document the data construction, training settings, and evaluation protocols. We plan to release the training, evaluation, and analysis code and configurations upon acceptance.

REFERENCES

Ashwath Vaithinathan Aravindan and Mayank Kejriwal. Fragile thoughts: How large language models handle chain-of-thought perturbations, 2026. URL https://arxiv.org/abs/2603.03332.

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost, 2016. URL https://arxiv.org/abs/1604.06174.

Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger, Vlad Mikulik, Samuel R. Bowman, Jan Leike, Jared Kaplan, and Ethan Perez. Reasoning models don't always say what they think, 2025. URL https://arxiv.org/abs/2505.05410.

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.

Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah Goodman, Christopher Potts, and Thomas Icard. Causal abstraction: A theoretical foundation for mechanistic interpretability. *Journal of Machine Learning Research*, 26(83):1–64, 2025. URL https://www.jmlr.org/papers/v26/23-0058.html.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You,

Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025. doi: 10.1038/s41586-025-09422-z. URL https://www.nature.com/articles/s41586-025-09422-z.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Jinghan Jia, Joe Benton, and Eric Easley. Faithfulness as information flow: Evaluating and training faithful chain-of-thought reasoning, 2026. URL https://arxiv.org/abs/2605.24286.

Dhiraj Kalamkar, Dheevatsa Mudigere, Naveen Mellempudi, Dipankar Das, Kunal Banerjee, Sasikanth Avancha, Dharma Teja Vooturi, Nataraj Jammalamadaka, Jianyu Huang, Hector Yuen, Jiyan Yang, Jongsoo Park, Alexander Heinecke, Evangelos Georganas, Sudarshan Srinivasan, Abhisek Kundu, Misha Smelyanskiy, Bharat Kaul, and Pradeep Dubey. A study of BFLOAT16 for deep learning training, 2019. URL https://arxiv.org/abs/1905.12322.

Jiwoo Kim and Jongwuk Lee. Causally redundant reasoning: A dual-disruption test framework for measuring shortcut tendencies in chain-of-thought. In *Findings of the Association for Computational Linguistics: EMNLP 2026*, 2026. To appear.

Tomek Korbak, Mikita Balesni, Elizabeth Barnes, Yoshua Bengio, Joe Benton, Joseph Bloom, Mark Chen, Alan Cooney, Allan Dafoe, Anca Dragan, Scott Emmons, Owain Evans, David Farhi, Ryan Greenblatt, Dan Hendrycks, Marius Hobbhahn, Evan Hubinger, Geoffrey Irving, Erik Jenner, Daniel Kokotajlo, Victoria Krakovna, Shane Legg, David Lindner, David Luan, Aleksander Madry, Julian Michael, Neel Nanda, Dave Orr, Jakub Pachocki, Ethan Perez, Mary Phuong, Fabien Roger, Joshua Saxe, Buck Shlegeris, Martín Soto, Eric Steinberger, Jasmine Wang, Wojciech Zaremba, Bowen Baker, Rohin Shah, and Vlad Mikulik. Chain of thought monitorability: A new and fragile opportunity for AI safety, 2025. URL https://arxiv.org/abs/2507.11473.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023. doi: 10.1145/3600006.3613165. URL https://doi.org/10.1145/3600006.3613165.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tülu 3: Pushing frontiers in open language model post-training, 2024. URL https://arxiv.org/abs/2411.15124.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, Saurav Kadavath, Shannon Yang, Thomas Henighan, Timothy Maxwell, Timothy Telleen-Lawton, Tristan Hume, Zac Hatfield-Dodds, Jared Kaplan, Jan Brauner, Samuel R. Bowman, and Ethan Perez. Measuring faithfulness in chain-of-thought reasoning, 2023. URL https://arxiv.org/abs/2307.13702.

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, 2023. URL https://arxiv.org/abs/2305.20050.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, Benjamin Bossan, and Marian Tietz. PEFT: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft, 2022.

Houman Mehrafarin, Amit Parekh, and Ioannis Konstas. When chain-of-thought fails, the solution hides in the hidden states, 2026. URL https://arxiv.org/abs/2604.23351.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

Debjit Paul, Robert West, Antoine Bosselut, and Boi Faltings. Making reasoning matter: Measuring and improving faithfulness of chain-of-thought reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 15012–15032. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.882. URL https://aclanthology.org/2024.findings-emnlp.882/.

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, 2024. URL https://arxiv.org/abs/2412.15115.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741. Curran Associates, Inc., 2023. doi: 10.52202/075280-2338. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/a85b405ed65c6477a4fe8302b5e06ce7-Abstract-Conference.html.

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, 2024. URL https://arxiv.org/abs/2402.03300.

Anand Swaroop, Akshat Nallani, Saksham Uboweja, Adiliia Uzdenova, Michael Nguyen, Kevin Zhu, Sunishchal Dev, Ashwinee Panda, Vasu Sharma, and Maheep Chaudhary. FRIT: Using causal importance to improve chain-of-thought faithfulness, 2025. URL https://arxiv.org/abs/2509.13334.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Advances in Neural Information Processing Systems*, volume 36, pp. 74952–74965. Curran Associates, Inc., 2023. doi: 10.52202/075280-3275. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/ed3fea9033a80fea1376299fa7863f4a-Abstract-Conference.html.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. TRL: Transformers reinforcement learning. https://github.com/huggingface/trl, 2020.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022. doi: 10.52202/068431-1800. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Qinan Yu, Alexa R. Tartaglini, Peter Hase, Carlos Guestrin, and Christopher Potts. Outcome rewards do not guarantee verifiable or causally important reasoning. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=VrY5x2smAd.

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, 2025. doi: 10.52202/085713-3775. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/a4277440d50f1f15d2cb4c14f7e0c0d2-Abstract-Conference.html.

Appendix organization. The appendix follows the main text: Appendix A details measurement, evaluation, reproducibility, and training costs (Section 3), Appendix B supports the RLVR and rationale-preference analyses (Section 4), and Appendix C specifies the optimization-access comparisons (Section 5). Appendices D and E expand the DCO design (Section 6) and all-layer activation-patching analysis (Section 7), respectively. Appendix F shows qualitative examples in which policies generate nearly identical reasoning but differ in CORD.

A Measurement, Evaluation, and Reproducibility

A.1 Evaluation Protocol and Comparison Design

We use a common generation and evaluation protocol across interventions.

Implementation details. We implement outcome and auxiliary-reward RLVR with TRL (von Werra et al., 2020), using its GRPO trainer (Shao et al., 2024) and colocated vLLM generation (Kwon et al., 2023), with 16 rollouts per prompt. DPO (Rafailov et al., 2023) uses TRL's DPO trainer, whereas the direct objectives and DCO use custom PyTorch (Paszke et al., 2019) and PEFT (Mangrulkar et al., 2022) code that differentiates teacher-forced answer likelihoods. Unless stated otherwise, policy updates use bfloat16 model weights (Kalamkar et al., 2019), gradient checkpointing (Chen et al., 2016), and rank-16 LoRA adapters (Hu et al., 2022) ($\alpha = 32$, dropout 0.05) on the attention and MLP projection layers. Training runs use NVIDIA RTX A6000 or A100 GPUs. Appendices B–D provide method-specific optimization settings, and Table 9 reports training compute.

Answer extraction. All policy comparisons use the same answer extractor. It returns the last nonempty `\boxed{...}` expression when present, then checks `####` and explicit final-answer phrases; if none is found, it uses the last numeric expression in the completion. We compare the extracted answer with the gold answer using the same numeric and LaTeX normalization across policies. For CORD interventions, the model completes a final-answer cue ending in `\boxed{`, and the extracted response is compared with the policy's unperturbed answer y_0 rather than the gold answer.

Generation and decoding variability. Reference CoTs are generated greedily with a 2,048-token cap. Because batched vLLM decoding can vary with batch composition and execution mode even at temperature 0, comparisons fix the software stack, GPU family, tensor-parallel size, and decoding mode. Regenerating the same π_{RLVR} checkpoint under a different execution mode yields

Table 4: Comparison design for the main text. Each effect compares the resulting policy with the comparator.

Intervention	Start	Resulting policy	Comparator	Role
Outcome improvement	π_{base}	π_{RLVR}	π_{base}	change in accuracy and redundancy
Rationale preference	π_{RLVR}	π_{DPO}	π_{RLVR}	change in reasoning distribution
Auxiliary reward	π_{base}	π_{aux}	π_{RLVR}	reward-mediated access
Direct access	π_{RLVR}	π_{direct}	π_{RLVR}	target-conditional gradient
DCO	π_{RLVR}	π_{DCO}	π_{RLVR}	constrained direct access

Table 5: Summary of evaluation measures. CORD is the primary behavioral outcome, CIR is a secondary behavioral measure, SR measures text sufficiency, and all-layer activation patching probes answer-score sensitivity.

Measure	Level	Intervention and readout
CORD $\downarrow$	Behavioral (primary)	Corrupt CoT suffix or mask question with CoT prefix $\rightarrow$ original-answer survival/recovery
CIR $\uparrow$	Behavioral (secondary)	Truncate CoT with question retained $\rightarrow$ answer-distribution change
Activation patching	Counterfactual	Patch the question or CoT span at all layers $\rightarrow$ original-answer log-likelihood change
SR $\uparrow$	Text/verifier	Compare answer-removed CoT with/without question $\rightarrow$ verifier-answer agreement

token-identical CoTs on 51.3% of GSM8K, 38.2% of MATH500, and 23.3% of DAPO-test items. Appendix B.2 uses this variation as a decoding baseline rather than interpreting every token-level difference as a policy change.

Comparison design Table 4 identifies the control for each intervention. Each row defines a separate policy contrast: reward arms are compared with outcome-only π_{RLVR} under the shared π_{base} initialization and GRPO protocol, whereas DPO and direct objectives are evaluated relative to their π_{RLVR} initialization.

A.2 Overview of Evaluation Measures

A.2.1 Primary Metric: CORD

Let $y_A(p;x)$ be the answer obtained from a perturbed CoT at position $p \in \mathcal{P}_A$, and $y_B(p;x)$ the answer obtained from a CoT prefix with the question masked. The A-axis component averages original-answer survival across intervention positions:

$$a(p;x) := \mathbb{1}[y_A(p;x) = y_0(x)], \qquad \bar{a}(x) := \frac{1}{|\mathcal{P}_A|} \sum_{p \in \mathcal{P}_A} a(p;x). \tag{10}$$

The B-axis component is determined by the earliest prefix that recovers the original answer:

$$\begin{aligned} b(p;x) &:= \mathbb{1}[y_B(p;x) = y_0(x)], \\ p^*(x) &:= \min(\{p \in \mathcal{P}_B : b(p;x) = 1\} \cup \{100\}), \qquad \bar{b}(x) := 1 - \frac{p^*(x)}{100}. \end{aligned} \tag{11}$$

CORD combines the two components through their harmonic mean:

$$\text{CORD}(x) := H\big(\bar{a}(x), \bar{b}(x)\big), \qquad H(u,v) := \begin{cases} \dfrac{2uv}{u+v}, & u+v>0, \\ 0, & u=v=0. \end{cases} \tag{12}$$

Table 6: DPO with proxy-scored versus full-CORD-scored preference pairs. Values are held-out matched $\Delta\mathrm{CORD}$ relative to the shared seed-42 π_{RLVR} initialization. Both conditions are evaluated with the full battery on each policy pair's both-correct valid items.

Task	Proxy-scored DPO	Full-CORD-scored DPO
GSM8K	$-0.0053_{\pm 0.0004}{}^{*}$	$-0.0024_{\pm 0.0001}$
MATH500	$-0.0013_{\pm 0.0069}$	$-0.0004_{\pm 0.0058}$
DAPO-val	$-0.0041_{\pm 0.0048}$	$-0.0067_{\pm 0.0047}$
DAPO-test	$+0.0040_{\pm 0.0064}$	$+0.0067_{\pm 0.0079}$

Full evaluation. Full evaluation replaces the CoT suffix with bottom-1 tokens at $\mathcal{P}_A = \{0, 20, 40, \ldots, 100\}$ and supplies token-level CoT prefixes at $\mathcal{P}_B = \{0, 5, 10, \ldots, 100\}$. A-axis replacement candidates exclude special and non-Latin tokens. Each policy's reference CoT is generated at temperature 0 with a cap of 2,048 tokens; intervention responses have caps of 256 tokens on the A-axis and 64 on the B-axis. All final $\Delta\mathrm{CORD}$ tables use this six-position A-axis and 21-position B-axis full evaluation battery. (An anonymized copy of the CORD paper is included as supplementary material for reference.)

Sensitivity to proxy scoring. We rebuild the DPO preference pairs from the same candidate rationales using full CORD scores, then retrain DPO from the same π_{RLVR} checkpoint for three epochs with the original hyperparameters. Neither scoring pipeline produces a consistent reduction in held-out CORD across tasks (Table 6). Thus, the DPO result does not depend on the lower-cost pair-scoring pipeline.

A.2.2 Auxiliary Metrics

The main text reports CIR and SR (Yu et al., 2026) as complementary measures.

CIR. For $p \in \mathcal{P}_{\mathrm{CIR}} := \{10, 20, \ldots, 90\}$, the percentage of CoT retained, let $r_p := p_\theta(y^* \mid x, m_{\le p})$ and $r_{100} := p_\theta(y^* \mid x, m)$. Define

$$\mathrm{CIR}(x) := \frac{1}{|\mathcal{P}_{\mathrm{CIR}}|} \sum_{p \in \mathcal{P}_{\mathrm{CIR}}} \mathrm{JS}(\mathrm{Ber}(r_p) \,\|\, \mathrm{Ber}(r_{100}))\,. \tag{13}$$

CIR truncates the CoT with the question visible.

SR. Rather than intervening on the policy's answer behavior, SR uses an external verifier to assess whether the generated reasoning text explicitly contains enough information to determine the answer. We remove exact matches to the policy's generated answer and `boxed{...}` expressions from each CoT without truncating it. We then present the answer-removed CoT to the verifier (i) with and (ii) without the original question. We set $\mathrm{SR} = 1$ if the verifier returns valid, equivalent answers in both conditions, and 0 otherwise. We use the SR verifier prompts of Yu et al. (2026) with `gpt-4o-mini` (temperature 0; seed 1234; 1,024-token completion cap). We read the final `<answer>` tag and treat missing, empty, or `no answer found` responses as unanswered. As with other behavioral measures, policy-level $\Delta\mathrm{SR}$ is a paired difference on items both policies answer correctly.

A.2.3 All-Layer Activation Patching

All-layer activation patching complements CORD by testing the original answer's sensitivity to replacing question or CoT content. On items answered correctly by both policies, each policy is evaluated with its own generated CoT and a teacher-forced gold answer. We separately patch the question span ($S = P$) or complete CoT span ($S = C$) with donor-induced states at the embedding and every decoder block; this all-layer intervention yields the same answer score as the donor-substituted sequence.

Let $\ell_{\text{clean}}(x)$ be the gold answer's token-averaged log-likelihood without patching and $\ell_S(x)$ the corresponding score after patching span S. We measure

$$|\Delta_S(x)| := \big|\ell_{\text{clean}}(x) - \ell_S(x)\big|, \qquad S \in \{P, C\}. \tag{14}$$

A decrease in mean $|\Delta_P|$ together with an increase in mean $|\Delta_C|$ indicates smaller answer-score changes under question substitution and larger changes under CoT substitution. Appendix E.1 gives the donor construction, two-pass implementation, numerical example, same-item control, and validation checks.

A.3 Estimands and Uncertainty

For policy π, let $c_i(\pi)$ indicate answer correctness; $v_i(\pi) = 1$ requires an intervention record consistent with the baseline, a defined CORD value, and $a_i(100) = 1$ at the unperturbed A-axis endpoint. For policies π and π', the matched paired estimand is

$$\begin{aligned} \mathcal{S}_{\pi\to\pi'} &:= \{i : c_i(\pi) = c_i(\pi') = 1,\ v_i(\pi) = v_i(\pi') = 1\}, \\ \Delta_{\pi\to\pi'}\text{CORD} &:= \frac{1}{|\mathcal{S}_{\pi\to\pi'}|} \sum_{i\in\mathcal{S}_{\pi\to\pi'}} \big[\text{CORD}_i(\pi') - \text{CORD}_i(\pi)\big] \end{aligned} \tag{15}$$

For behavioral measures and activation patching, each seed's 95% interval uses 2,000 same-item paired-bootstrap draws. Multi-seed results additionally report the mean ± sample standard deviation across training seeds; the item bootstrap quantifies within-policy sampling uncertainty and does not include training-seed variation.

A.4 Secondary Behavioral Results

Tables 7 and 8 report CIR and SR for the policy comparisons in main-text Tables 1 and 2, respectively. Each value is a paired difference on the same items that both policies answer correctly and for which the measure is defined.

Table 7: Secondary behavioral results corresponding to Table 1. Each Δ is the right policy minus the left policy in the group heading.

	Outcome RLVR ($\pi_{\text{base}} \to \pi_{\text{RLVR}}$)		DPO ($\pi_{\text{RLVR}} \to \pi_{\text{DPO}}$)	
Task	ΔCIR ↑	ΔSR ↑	ΔCIR ↑	ΔSR ↑
GSM8K	$+0.023_{\pm 0.006}{}^{*}$	$+0.016_{\pm 0.012}$	$-0.003_{\pm 0.0002}{}^{*}$	$-0.005_{\pm 0.003}$
MATH500	$-0.001_{\pm 0.001}$	$+0.014_{\pm 0.013}$	$+0.000_{\pm 0.002}$	$+0.004_{\pm 0.009}$
DAPO-val	$-0.001_{\pm 0.004}$	$+0.056_{\pm 0.026}$	$+0.001_{\pm 0.002}$	$+0.006_{\pm 0.017}$
DAPO-test	$+0.015_{\pm 0.001}{}^{*}$	$-0.002_{\pm 0.003}$	$-0.002_{\pm 0.001}$	$+0.003_{\pm 0.014}$

Table 8: Secondary behavioral results corresponding to Table 2. All values are matched paired differences relative to π_{RLVR}.

	GSM8K		MATH500		DAPO-test	
Method	ΔCIR ↑	ΔSR ↑	ΔCIR ↑	ΔSR ↑	ΔCIR ↑	ΔSR ↑
① *Post-training: rationale-level preference*						
DPO	$-0.003_{\pm 0.0002}{}^{*}$	$-0.005_{\pm 0.003}$	$+0.000_{\pm 0.002}$	$+0.004_{\pm 0.009}$	$-0.002_{\pm 0.001}$	$+0.003_{\pm 0.014}$
② *Policy formation: trajectory-level scalar credit*						
Scalar credit (A)	$+0.011_{\pm 0.004}{}^{*}$	$-0.008_{\pm 0.009}$	$+0.001_{\pm 0.002}$	$+0.005_{\pm 0.007}$	$+0.005_{\pm 0.001}$	$-0.008_{\pm 0.016}$
Scalar credit (B)	$+0.009_{\pm 0.004}{}^{*}$	$+0.001_{\pm 0.008}$	$+0.002_{\pm 0.001}$	$+0.006_{\pm 0.020}$	$+0.003_{\pm 0.002}$	$-0.016_{\pm 0.013}$
Scalar credit (A+B)	$+0.011_{\pm 0.002}{}^{*}$	$-0.007_{\pm 0.009}$	$+0.001_{\pm 0.001}$	$+0.015_{\pm 0.024}$	$+0.003_{\pm 0.007}$	$-0.022_{\pm 0.028}$
③ *Post-training: target-conditional gradient*						
Naive direct (A)	$+0.229_{\pm 0.015}{}^{*}$	$+0.002_{\pm 0.004}$	$+0.186_{\pm 0.020}{}^{*}$	$+0.010_{\pm 0.010}$	$+0.221_{\pm 0.016}{}^{*}$	$-0.013_{\pm 0.008}$
Naive direct (B)	$+0.186_{\pm 0.013}{}^{*}$	$+0.002_{\pm 0.003}$	$+0.150_{\pm 0.015}{}^{*}$	$+0.033_{\pm 0.007}$	$+0.143_{\pm 0.016}{}^{*}$	$-0.058_{\pm 0.029}$
DCO (B)	$+0.227_{\pm 0.025}{}^{*}$	$-0.006_{\pm 0.007}$	$+0.180_{\pm 0.017}{}^{*}$	$+0.029_{\pm 0.002}$	$+0.163_{\pm 0.025}{}^{*}$	$-0.016_{\pm 0.020}$

Table 9: Training cost by intervention family. GPU-hours equal wall-clock time times allocated GPU count; evaluation jobs are excluded. DPO and DCO elapsed times are three-seed means. For the corrected A-axis and A+B arms, elapsed time includes both the initial and resumed training jobs.

Class	Method	Hardware × elapsed	GPU-hours
Outcome	RLVR (GRPO)	4×A6000 × 11.8h	47.2
Reward	A-axis reward	4×A6000 × 23.9h	95.4
	B-axis reward	4×A100 × 10.3h	41.2
	A+B reward	4×A6000 × 24.2h	96.8
	CIR reward Yu et al. (2026)	4×A6000 × 23.4h	93.6
	SR reward Yu et al. (2026)	4×A6000 × 13.8h	55.1
	Direct-objective scalar (detached)	4×A6000 × 14.4h	57.7
Preference	DPO	1×A6000 × 1.5h	1.5
Direct	Naive B loss	1×A6000 × 1.5h	1.5
	Naive A loss	1×A6000 × 2.1h	2.1
	DCO	1×A6000 × 1.3h	1.3

Under DCO, CIR rises on all three tasks alongside substantial CORD reductions, indicating greater answer reliance on the CoT under a second, truncation-based behavioral measure. This gain is not accompanied by a consistent increase in SR, the verifier-based measure of trace sufficiency. This pattern suggests that greater answer reliance does not require the generated trace to become more sufficient for an external verifier.

A.5 Training Cost

Direct objectives use teacher-forced computation on fixed triples and require over 20 times fewer GPU-hours than reward-based GRPO in these implementations (Table 9). This difference follows from optimizing the relevant likelihood directly rather than regenerating rollouts to recompute a counterfactual reward.

B Supporting Analyses for Outcome RLVR and Rationale-Side Supervision

This section examines whether the small held-out effects in Section 4 arise from failed preference optimization, decoding variability, or dependence on a particular model family.

B.1 DPO Training Details and Dynamics

DPO uses the TRL implementation with $\beta = 0.1$, learning rate 5×10^{-6}, three epochs, and LoRA rank 16. All main-text DPO results use three training seeds under this configuration.

Training-pair preference accuracy reaches 73–77%, while held-out-pair accuracy remains near chance (Figure 1(a)). Proxy CORD on DPO rollouts decreases by $0.010_{\pm 0.003}$ at the final checkpoint, but no consistent reduction appears on the four held-out evaluations (Figure 1(b)). DPO thus fits the training preference signal without transferring it reliably to held-out preference accuracy or causal reliance.

B.2 Reasoning-Text Change Beyond Decoding Variability

Table 10 separates policy-induced text change from greedy-decoding variability. For paired generated token-ID sequences a and b, we compute similarity as $2M/(|a| + |b|)$ using `difflib.SequenceMatcher` with `autojunk=False`, where M is the total length of its matched contiguous blocks. Ident. is the percentage of traces that exactly match the fixed seed-42 π_{RLVR} trace, and Rewr. is the percentage with similarity below 0.5. DPO exceeds the decoding

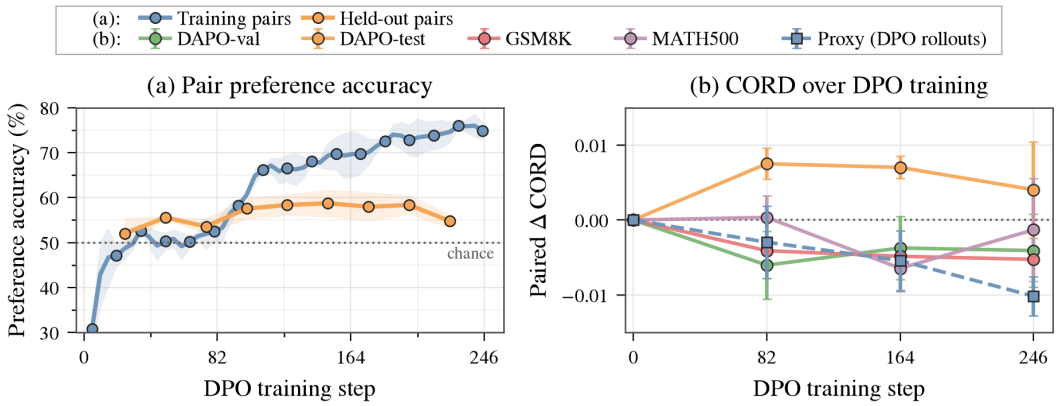

Figure 1: DPO training dynamics. (a) Preference accuracy on training and held-out pairs. (b) Proxy CORD on DPO rollouts and paired ΔCORD on four held-out sets. Curves show the three-seed mean $\pm$ sample standard deviation; the training curve uses a six-point trailing mean, and step 0 is the shared π_{RLVR} initialization.

Table 10: Reasoning-text change and matched ΔCORD relative to seed-42 π_{RLVR}. Ident. is the percentage of token-identical CoTs, and Rewr. the percentage with sequence similarity below 0.5. The RLVR control uses two seeds; the decoding baseline uses one.

	GSM8K			MATH500			DAPO-test		
vs π_{RLVR}	Ident.	Rewr.	ΔCORD $\downarrow$	Ident.	Rewr.	ΔCORD $\downarrow$	Ident.	Rewr.	ΔCORD $\downarrow$
Decoding floor	51.3	4.6	-0.000	38.2	8.8	$+0.001$	23.3	24.2	$+0.005$
π_{RLVR} (2 seeds)	16.6$\pm$14.8	13.0$\pm$5.5	-0.001 ± 0.007	17.4$\pm$5.7	19.3$\pm$4.7	-0.005 ± 0.004	8.4$\pm$6.9	39.3$\pm$6.6	$+0.014\pm0.010$
DPO	31.8$\pm$2.2	7.7$\pm$0.1	-0.005 ± 0.0004	24.5$\pm$0.4	13.8$\pm$0.4	-0.001 ± 0.007	10.2$\pm$0.2	35.0$\pm$0.8	$+0.004\pm0.006$
B-axis reward	14.8$\pm$9.6	12.7$\pm$3.2	$+0.002\pm0.012$	17.5$\pm$5.7	18.7$\pm$3.2	-0.007 ± 0.013	7.2$\pm$4.8	42.1$\pm$7.3	$+0.009\pm0.009$
Naive B loss	1.4$\pm$0.0	26.2$\pm$1.3	-0.209 ± 0.006	1.9$\pm$0.6	34.2$\pm$1.5	-0.236 ± 0.002	1.3$\pm$0.3	60.8$\pm$1.4	-0.141 ± 0.011
DCO	8.6$\pm$0.6	9.8$\pm$0.4	-0.300 ± 0.002	11.5$\pm$1.0	18.7$\pm$1.3	-0.347 ± 0.007	3.4$\pm$0.4	41.1$\pm$0.7	-0.197 ± 0.018

baseline on all three benchmarks, confirming that the policy changes its generated reasoning even when held-out CORD changes little.

Changing the decoding mode alters many CoTs without materially changing CORD. Conversely, DCO lowers CORD more than naive B loss while rewriting fewer traces across all three benchmarks.

B.3 Model-Family Extension

To test whether the lack of consistent CORD reductions under outcome RLVR and DPO extends beyond Qwen2.5-7B, we evaluate Qwen2.5-3B, Qwen2.5-14B, OLMo-2-7B, OLMo-2-13B, and Llama-3.1-8B on GSM8K (Table 11).

Band-filtered RLVR and an expanded DPO pool. Standard dynamic sampling makes any mixed 16-rollout group eligible, even when only one rollout is correct. Such groups are common for low-accuracy non-Qwen models, limiting outcome learning and leaving few correct traces for DPO pair construction. Band-filtered RLVR changes the group-selection rule: a group is eligible only if 2–14 of its 16 rollouts are correct, while the final-answer correctness reward remains the same. To obtain more correct candidates for DPO, we also expand the generation pool from 3,000 DAPO training problems in the Qwen experiments to 9,000 for the non-Qwen models. Pair construction from the resulting π_{RLVR} checkpoints yields 302, 135, and 21 pairs for OLMo-2-7B, OLMo-2-13B, and Llama-3.1-8B, respectively; DPO uses the same hyperparameters as in the Qwen experiments.

Table 11: GSM8K results across model families. LM restricts matched $\Delta\mathrm{CORD}$ to items for which both CoTs contain at most 1,024 tokens. Entries without $\pm$ use one seed. DPO seeds share the seed-42 π_{RLVR} initialization. †Base accuracy is 89.9%. §Band-filtered Llama DPO uses 21 preference pairs.

	$\pi_{\mathrm{base}} \to \pi_{\mathrm{RLVR}}$			$\pi_{\mathrm{RLVR}} \to$ DPO
Model	Δacc ↑ (%p)	$\Delta\mathrm{CORD}\downarrow$	LM $\Delta\mathrm{CORD}\downarrow$	$\Delta\mathrm{CORD}\downarrow$
Standard RLVR				
Qwen 3B	$+10.5_{\pm 0.3}{}^{*}$	$+0.014_{\pm 0.004}{}^{*}$	$+0.015_{\pm 0.004}{}^{*}$	$-0.004_{\pm 0.0004}$
Qwen 7B	$+10.1_{\pm 0.2}{}^{*}$	$+0.003_{\pm 0.005}$	$+0.006_{\pm 0.006}$	$-0.005_{\pm 0.0004}{}^{*}$
Qwen 14B	$+1.4_{\pm 0.4}{}^{\dagger}$	$-0.028_{\pm 0.005}{}^{*}$	$+0.008_{\pm 0.004}$	$-0.000_{\pm 0.001}$
Band-filtered RLVR				
OLMo 7B	$+30.6_{\pm 8.6}{}^{*}$	$+0.022_{\pm 0.004}$	$+0.042_{\pm 0.009}{}^{*}$	$+0.004_{\pm 0.001}$
OLMo 13B	$+15.5_{\pm 2.2}{}^{*}$	$-0.000_{\pm 0.003}$	$-0.000_{\pm 0.003}$	$+0.000_{\pm 0.003}$
Llama 8B	-2.7^{*}	-0.018^{*}	-0.017^{*}	$-0.010^{*\,\S}$
Standard RLVR, non-Qwen (reference)				
OLMo 7B	$+3.9^{*}$	$+0.003$	$+0.005$	—
OLMo 13B	$+12.1^{*}$	$+0.006$	$+0.006$	—
Llama 8B	$+4.4^{*}$	-0.016	-0.009	—

Accuracy gains do not consistently reduce CORD across model families. Band-filtered RLVR raises GSM8K accuracy by 30.6 and 15.5%p for OLMo-2-7B and -13B, respectively (three-seed means), but neither length-matched CORD nor subsequent DPO shows a consistent reduction. Qwen2.5-3B follows the same pattern; the apparent Qwen2.5-14B reduction disappears after matching CoT length ($\Delta\mathrm{CORD} = -0.028$ to $+0.008$). (The negative Llama-3.1-8B estimates are less informative because band filtering lowers accuracy and DPO uses only 21 pairs.) These results extend the Qwen2.5-7B finding beyond a single checkpoint.

C Supporting Analyses for Optimization Access

This section specifies the reward-mediated and direct objectives in Section 5, then tests whether the weak reward-mediated effects arise from an unlearned bonus or from the choice of optimization route.

Reward arms share the π_{base} initialization and GRPO protocol of outcome-only RLVR, varying only the auxiliary bonus. Direct arms start from π_{RLVR} and optimize fixed triples after outcome training. Results are therefore interpreted as intervention effects relative to the control appropriate to each training stage.

C.1 Exact Access-Path Objectives

Reward-mediated access. For a policy rollout τ, let $y_0(\tau)$ be the original answer and $c(\tau) = \mathbb{1}[y_0(\tau) = y^*]$ indicate correctness. Hats denote in-loop proxies computed during training. For target variation $v \in \{A, B, A+B\}$, the reward is

$$R_v(\tau) = c(\tau)\big[1 + \lambda_{\mathrm{rew}} u_v(\tau)\big], \qquad \begin{cases} u_A(\tau) = 1 - \widehat{a}(\tau), \\ u_B(\tau) = \widehat{p}^{\,*}(\tau)/100, \\ u_{A+B}(\tau) = 1 - \widehat{\mathrm{CORD}}(\tau) \end{cases} \tag{16}$$

with $\lambda_{\mathrm{rew}} = 1$. Each arm starts from π_{base}, generates 16 rollouts per prompt, and runs 300 steps of the same dynamic-sampling GRPO used for outcome-only π_{RLVR}. The in-loop reward uses six A-axis positions and six B-axis positions $(0, 20, \ldots, 100)$, with an answer cap of 24 tokens.

Direct access. For each correct training triple (x_i, m_i, y_i), let p be the percentage of the CoT retained before A-axis suffix corruption or supplied as a B-axis prefix. The intervention contexts are

$$h^A_{i,p} := (x_i, \tilde{m}_{i,p}), \qquad h^B_{i,p} := (x_{i,\varnothing}, m_{i,\le p}). \tag{17}$$

For $v \in \{A, B\}$, the token-normalized answer score is

$$s^{v}_{\theta,i,p} := \frac{1}{|y_i|} \sum_{j=1}^{|y_i|} \log p_\theta(y_{i,j} \mid h^{v}_{i,p}, y_{i,<j}). \tag{18}$$

For a minibatch $\mathcal{B}$, let $\langle \cdot \rangle_{\mathcal{B}}$ denote the item mean, and write $s_\theta(y_i \mid h)$ for the same answer score in a general context h. The counterfactual, clean, and full-CoT recovery scores are

$$S^{(v)}_{\mathrm{cf}} := \langle s^{v}_{\theta,i,p_i} \rangle_{\mathcal{B}}, \quad S_{\mathrm{clean}} := \langle s_\theta(y_i \mid x_i, m_i) \rangle_{\mathcal{B}}, \quad S_{\mathrm{rec}} := \langle s_\theta(y_i \mid x_{i,\varnothing}, m_i) \rangle_{\mathcal{B}}. \tag{19}$$

The direct objective is

$$\mathcal{L}^{(v)}_{\mathrm{dir}} = \lambda_{\mathrm{cf}} [S^{(v)}_{\mathrm{cf}} - f]_+ - S_{\mathrm{clean}} - \mathbb{1}[v = B] S_{\mathrm{rec}}. \tag{20}$$

Here $[z]_+ = \max(z, 0)$, $\lambda_{\mathrm{cf}} = 0.5$, and $f = -6$. For each item, the A-axis variation samples $p_i \sim \mathrm{Unif}(\mathcal{Q}_A)$ with $\mathcal{Q}_A = \{0, 20, 40, 60, 80, 100\}$, while the B-axis variation samples $p_i \sim \mathrm{Unif}(\mathcal{Q}_{\mathrm{early}})$ with $\mathcal{Q}_{\mathrm{early}} = \{0, 20, 40, 60\}$. Positions 80 and 100 are excluded from B-axis suppression because a late prefix may already suffice to recover the answer legitimately. The full CoT is retained as the recovery anchor in the final term, not targeted for suppression.

Direct arms use 767 fixed triples from correct greedy generations of π_{RLVR} that do not hit the generation cap. We train for three epochs with LoRA rank 16, learning rate 10^{-5}, four triples per minibatch, and gradient accumulation of two, giving 576 minibatch steps and 288 optimizer updates.

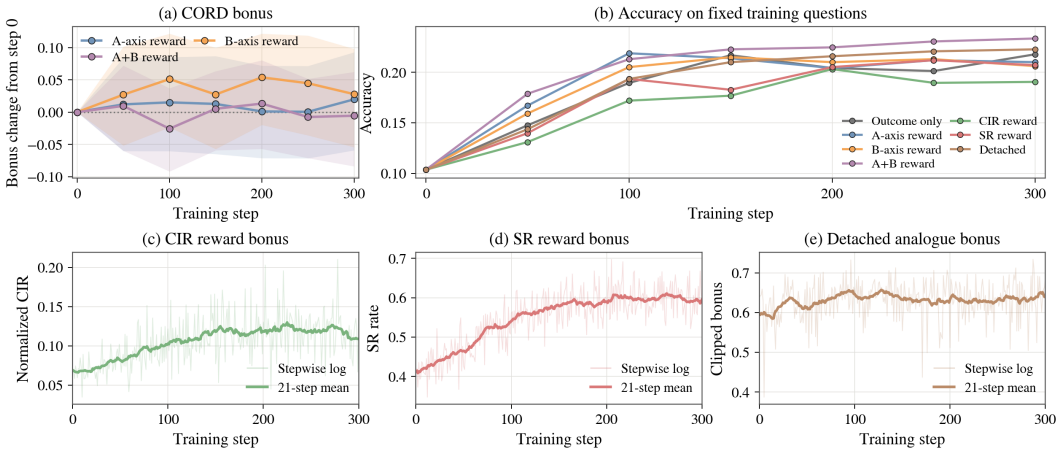

Figure 2: Training trajectories of reward-mediated controls (seed 42). (a) Change in the A-, B-, and A+B-based bonuses from step 0, with 95% question-bootstrap intervals; (b) accuracy. Both are measured every 50 steps on the same 128 training questions, with eight rollouts per question. (c–e) In-loop CIR, SR, and detached-analogue bonuses, showing step values (faint) and 21-step means (solid). The two groups use different measurement protocols and bonus scales. A and A+B use corrected replacement runs.

C.2 Reward-Signal Learning and Transfer

The reward controls distinguish two possibilities: the auxiliary signal may fail to increase during training, or an increased signal may fail to reduce held-out redundancy. For the A-, B-, and A+B-based rewards, the bonus shows no significant increase across checkpoints (Figure 2(a)), while accuracy rises in these arms and the outcome-only control (Figure 2(b)). Their total reward gains therefore come mainly from correctness.

To test whether a learnable auxiliary signal transfers, we train CIR and SR reward arms following Yu et al. (2026). For training, $\mathrm{CIR}_{\mathrm{rew}} = \min\{1, \mathrm{CIR}/\log 2\}$ normalizes the CIR bonus to $[0, 1]$; evaluation reports CIR on its original scale. The rewards are $R = c + \mathrm{CIR}_{\mathrm{rew}}$ and $R = c + \mathrm{SR}$, respectively. Both in-loop bonuses increase during training (Figure 2(c,d)).

Detached reward analogue of the direct loss. To test scalar credit for a closer analogue of the direct B-axis objective, we convert its answer-likelihood terms into a detached rollout-level bonus. For each correct rollout $\tau = (x, m, y_0)$, define

$$q(\tau) = s_\theta(y_0 \mid x, m) - \frac{\lambda_B}{|\mathcal{Q}_{\text{early}}|} \sum_{p \in \mathcal{Q}_{\text{early}}} [s_\theta(y_0 \mid x_\varnothing, m_{\le p}) - f]_+ + s_\theta(y_0 \mid x_\varnothing, m), \qquad R = \mathbb{1}[c]\,(1 + \lambda\,\text{clip}(1 + q/Q, 0, 1)).$$

Detaching q leaves only the score-function gradient. This control retains the direct loss's contexts, likelihood terms, candidate positions, and floor, but averages positions and applies the hinge per rollout rather than sampling one position and applying the hinge to the minibatch mean. It follows the reward-arm protocol from π_{base} for 300 steps. Its in-loop bonus rises only from 0.59 to 0.64 (Figure 2(e)), while the targeted score changes from -2.38 to -2.49.

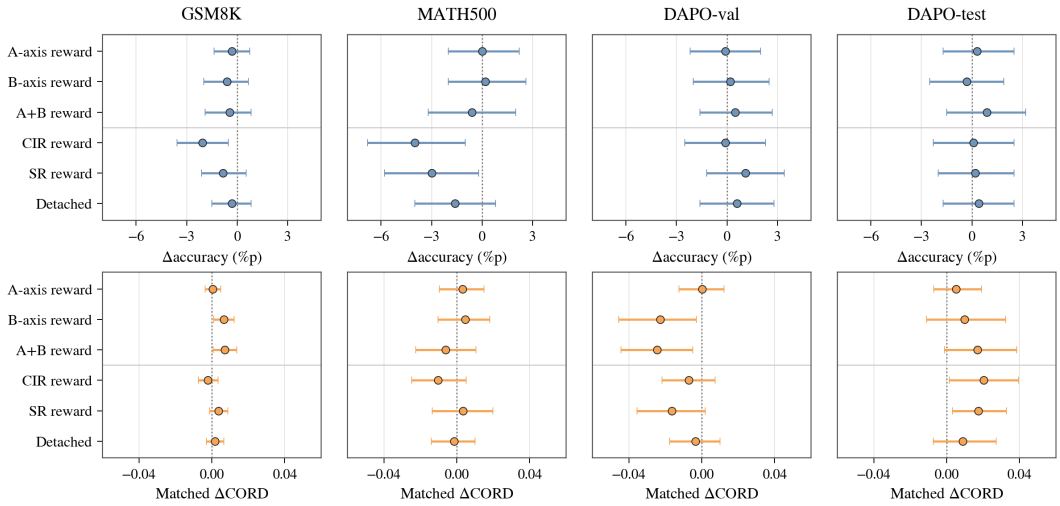

Figure 3: Final held-out effects of reward-mediated arms relative to outcome-only RLVR (seed 42). The upper row shows paired Δaccuracy in percentage points over all questions; the lower row shows matched ΔCORD on questions answered correctly by both policies with valid full-prefix recovery. Points and whiskers are paired means and 95% item-bootstrap intervals. A-axis and A+B use the corrected reward runs; negative ΔCORD indicates lower causal redundancy.

Figure 3 shows that neither CIR nor SR reward produces a consistent held-out CORD reduction despite increasing during training. CIR lowers accuracy on GSM8K and MATH500, SR lowers it on MATH500, and held-out SR does not increase consistently. The detached analogue likewise has no significant effect: matched ΔCORD ranges from -0.004 to $+0.009$ and accuracy from -1.6 to $+0.6$%p across four tasks. By comparison, direct B-axis optimization yields GSM8K $\Delta\text{CORD} = -0.209$ (Table 2). Although the routes start at different training stages, the learnable CIR/SR bonuses and the closer detached control show that the weak reward-mediated result is not explained solely by failure to increase the original A/B bonuses.

C.3 Structural Optimization Channels

Alongside reward-mediated access, we reproduce the structural interventions proposed by Jia et al. (2026) under the same evaluation. All four channels train from π_{base} for 300 GRPO steps; other settings match outcome RLVR.

Table 12: Info-Flow structural channels relative to π_{RLVR}. Δacc is in percentage points, and matched $\Delta\mathrm{CORD}$ uses items answered correctly by both policies.

	GSM8K		MATH500		DAPO-val		DAPO-test	
Channel	Δacc ↑	ΔCORD ↓	Δacc ↑	ΔCORD ↓	Δacc ↑	ΔCORD ↓	Δacc ↑	ΔCORD ↓
Update masking	$+0.0$	$+0.003$	$+1.0$	$+0.003$	$+2.0$	-0.007	$+1.4$	$+0.012$
Gradient masking	-1.1	-0.005	$+0.4$	-0.013	-0.5	-0.018^{*}	-0.1	$+0.007$
CoT-gradient	-5.1^{*}	-0.001	-2.4	-0.011	$+0.9$	-0.018	-2.8^{*}	$+0.004$
FACT ($\epsilon = 0.01$)	-1.0	-0.001	$+0.0$	-0.013	$+0.6$	-0.009	$+1.0$	$+0.007$

The structural channels yield no consistent CORD reduction across tasks. The two significant DAPO-val estimates, -0.018 for gradient masking and -0.020 for length-matched CoT-gradient, are about one tenth the effects of direct optimization. FACT is similarly small.

D Supporting Analyses for Direct Counterfactual Optimization

This section gives the full DCO objective, hyperparameter selection analysis, and component ablations supporting Section 6.

D.1 Objective Details

Equation 5 defines the question-masked B-axis context and answer score $s^{B}_{\theta,i,p}$. The corresponding quantity under the fixed π_{RLVR} is $s^{B}_{\mathrm{ref},i,p}$.

DCO samples $p \sim \mathrm{Unif}(\mathcal{Q}_{\mathrm{early}})$, where $\mathcal{Q}_{\mathrm{early}} = \{0, 20, 40, 60\}$, and moves the original answer score toward a target δ below the reference score:

$$\mathcal{L}_{\mathrm{sup}}(\theta) := \frac{1}{2}\,\mathbb{E}_{i,p}\left[\rho_{\kappa}\big(s^{B}_{\theta,i,p} - s^{B}_{\mathrm{ref},i,p} + \delta\big)\right], \tag{21}$$

Here

$$\rho_{\kappa}(u) = \begin{cases} \frac{1}{2}u^{2}, & |u| \le \kappa, \\ \kappa\left(|u| - \frac{1}{2}\kappa\right), & |u| > \kappa. \end{cases} \tag{22}$$

At initialization, $s^{B}_{\theta,i,p} = s^{B}_{\mathrm{ref},i,p}$, so this term lowers the original answer's likelihood. Its gradient vanishes when the score reaches $s^{B}_{\mathrm{ref},i,p} - \delta$ and reverses if the score falls below the target.

Normal-behavior protection keeps the policy close to the fixed π_{RLVR} on intact inputs: the original question, unmodified reference CoT, and reference answer. For each item, we sample a starting percentage $q_i \in \{0, 20, 40, 60, 80\}$ of the reference CoT. We select up to 32 CoT token positions from that point onward, together with all content-token positions in the reference answer. Let $\mathcal{W}_i$ denote these positions and $c_{i,t}$ the intact, teacher-forced context preceding position t. For minibatch $\mathcal{B}$, the protection quantity is

$$D_{\mathrm{normal}}(\theta; \mathcal{B}) := \frac{1}{\sum_{i\in\mathcal{B}} |\mathcal{W}_i|} \sum_{i\in\mathcal{B}} \sum_{t\in\mathcal{W}_i} D_{\mathrm{KL}}(p_{\mathrm{ref}}(\cdot \mid c_{i,t}) \,\|\, p_{\theta}(\cdot \mid c_{i,t}))\,. \tag{23}$$

This averages the next-token KL over all selected CoT and answer positions in the minibatch. The budget ϵ applies to this average, rather than to each token separately. The protection term is

$$\mathcal{L}_{\mathrm{prot}}(\theta) := \lambda\,[D_{\mathrm{normal}}(\theta; \mathcal{B}) - \epsilon]_{+}\,, \qquad [z]_{+} := \max(z, 0) \tag{24}$$

While $D_{\mathrm{normal}} \le \epsilon$, this term has no gradient; above the budget, it pushes the policy's normal-context distributions toward those of π_{RLVR}. Gradients do not pass through the fixed reference distribution.

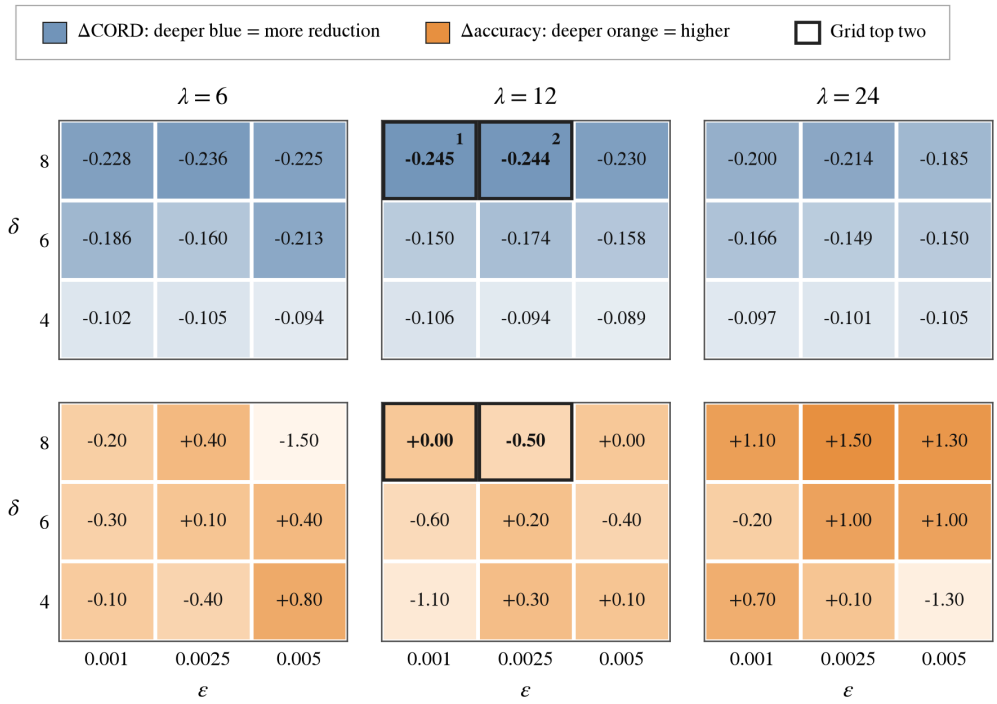

Figure 4: DAPO-val seed-1 results for the complete $3 \times 3 \times 3$ grid at $\kappa = 1$. Panels fix λ; rows vary δ and columns vary ϵ. Cells show conservative paired $\Delta\mathrm{CORD}$ (blue) and $\Delta\mathrm{acc}$ in percentage points (orange). Numbered outlines mark the top two settings under DAPO-val ranking.

D.2 Hyperparameter Selection for DCO

With the two-term objective fixed, we search the suppression distance δ, normal-path KL budget ϵ, and protection coefficient λ at $\kappa = 1$:

$$\delta \in \{4, 6, 8\}, \qquad \epsilon \in \{0.001, 0.0025, 0.005\}, \qquad \lambda \in \{6, 12, 24\}. \tag{25}$$

All 27 settings start from the same π_{RLVR} checkpoint and use the fixed triples and three-epoch LoRA protocol of the direct arms (Appendix C.1). Re-ranking the completed grid on DAPO-val, we require $\Delta\mathrm{acc} \geq -1.0\%\mathrm{p}$, then rank feasible settings by lower conservative $\Delta\mathrm{CORD}$, breaking ties by higher $\Delta\mathrm{acc}$. The top two have $(\delta, \lambda) = (8, 12)$ and $\epsilon \in \{0.001, 0.0025\}$ (Figure 4).

For each of these two settings, we evaluate $\kappa \in \{0.5, 1, 2\}$. We retain candidates with DAPO-val $\Delta\mathrm{acc} \geq -2.0\%\mathrm{p}$ and rank them by lower conservative DAPO-val $\Delta\mathrm{CORD}$, breaking ties by accuracy. The three finalists, $(\epsilon, \kappa) \in \{(0.0025, 2), (0.001, 1), (0.0025, 1)\}$, are additionally trained with seeds 2 and 3. Figure 5 shows their DAPO-val results. Applying the DAPO-val grid criterion to the three-seed finalist means selects $(\delta, \epsilon, \lambda, \kappa) = (8, 0.0025, 12, 1)$.

D.3 Ablation for DCO

All ablations use the selected setting $(\delta, \epsilon, \lambda, \kappa) = (8, 0.0025, 12, 1)$. We isolate the bounded suppression target, normal-path protection, and KL budget with three matched-seed ablations. Define

$$\begin{aligned} S_{\mathrm{shift}}(\theta) &:= \tfrac{1}{2}\mathbb{E}_{i,p}\left[\rho_\kappa\left(s^B_{\theta,i,p} - (s^B_{\mathrm{ref},i,p} - \delta)\right)\right], \\ S_{\mathrm{unbounded}}(\theta) &:= \tfrac{1}{2}\mathbb{E}_{i,p}\left[s^B_{\theta,i,p}\right], \end{aligned} \tag{26}$$

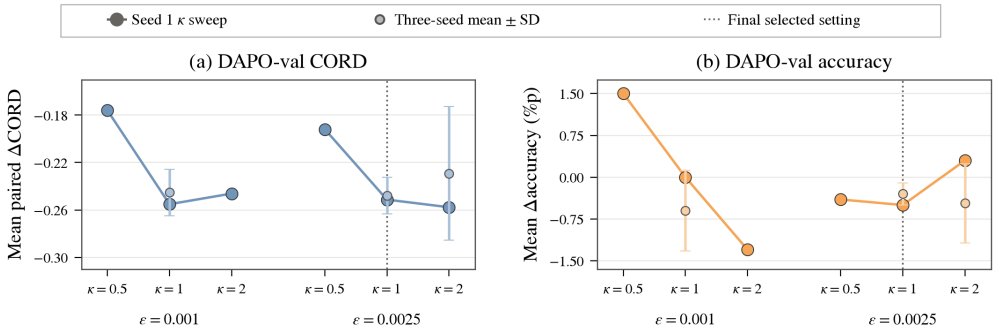

Figure 5: DAPO-val κ follow-up and three-seed finalist evaluation. Lines connect seed-1 results for $\kappa \in \{0.5, 1, 2\}$ within each ϵ group; lighter circles and whiskers show finalist means $\pm$ sample standard deviations across three seeds. Panels show (a) conservative paired ΔCORD and (b) Δacc in percentage points. The dotted line marks the final setting, $(\epsilon, \kappa) = (0.0025, 1)$.

Table 13: DCO ablations relative to fixed seed-42 π_{RLVR} at the selected setting. Δacc is in percentage points, and ΔCORD uses variant-specific common-correct valid cohorts. Significance is omitted if any seed has fewer than 10 matched items. N/A denotes no eligible item; † denotes one defined seed.

	GSM8K		MATH500		DAPO-test	
Variant	Δacc ↑	ΔCORD ↓	Δacc ↑	ΔCORD ↓	Δacc ↑	ΔCORD ↓
DCO	$-0.6_{\pm 0.3}$	$-0.300_{\pm 0.002}^{*}$	$+0.3_{\pm 0.4}$	$-0.347_{\pm 0.007}^{*}$	$+0.4_{\pm 0.4}$	$-0.197_{\pm 0.018}^{*}$
No protection	$-24.9_{\pm 11.9}^{*}$	$-0.322_{\pm 0.046}$	$-27.5_{\pm 8.5}^{*}$	$-0.386_{\pm 0.043}^{*}$	$-10.6_{\pm 2.8}^{*}$	$-0.206_{\pm 0.013}^{*}$
Plain KL	$-0.5_{\pm 0.4}$	$-0.280_{\pm 0.008}^{*}$	$+0.7_{\pm 0.9}$	$-0.319_{\pm 0.004}^{*}$	$+1.0_{\pm 1.4}$	$-0.180_{\pm 0.004}^{*}$
Unbounded	$-15.8_{\pm 3.2}^{*}$	$-0.220_{\pm 0.125}$	$-32.9_{\pm 5.5}^{*}$	N/A	$-14.3_{\pm 1.9}^{*}$	$-0.087^{\dagger}$

where p is sampled from $\{0, 20, 40, 60\}$ as in the final objective. With $D_{\text{normal}} = D_{\text{normal}}(\theta; \mathcal{B})$, the four losses are

$$\begin{aligned} \mathcal{L}_{\text{DCO}} &= S_{\text{shift}} + \lambda[D_{\text{normal}} - \epsilon]_+, \\ \mathcal{L}_{\text{suppression only}} &= S_{\text{shift}}, \\ \mathcal{L}_{\text{plain KL}} &= S_{\text{shift}} + \lambda D_{\text{normal}}, \\ \mathcal{L}_{\text{unbounded}} &= S_{\text{unbounded}} + \lambda[D_{\text{normal}} - \epsilon]_+. \end{aligned} \tag{27}$$

No protection removes the normal-path term, Plain KL removes its ϵ budget, and Unbounded replaces the shifted Huber target with linear suppression, making δ and κ inactive. Every variant is evaluated at its final checkpoint with the full battery.

Removing normal-path protection lowers accuracy by 10.6–27.5%p across the three tasks. Its mean CORD reductions are numerically large, but they are measured only on the items that remain correct under each variant. They therefore do not offset the loss of normal-task performance.

Retaining protection while making suppression unbounded still lowers accuracy by 14.3–32.9%p. MATH500 has no eligible matched CORD items, and DAPO-test has only one defined seed. This pattern is consistent with the shifted target's role in stopping further suppression once the target score is reached.

Plain KL avoids these accuracy losses, but its mean CORD reduction is smaller than DCO's by 0.020, 0.028, and 0.017 on GSM8K, MATH500, and DAPO-test, respectively. The KL budget thus permits larger CORD reductions while keeping accuracy changes small in these experiments.

E SUPPORTING ANALYSES FOR ALL-LAYER ACTIVATION PATCHING

E.1 ALL-LAYER ACTIVATION-PATCHING PROTOCOL

Motivation: source sensitivity rather than layer localization. Activation patching commonly replaces selected internal states to identify where a computation affects an output (Geiger et al., 2025; Mehrafarin et al., 2026). Here we ask how the original answer responds separately to a change in question content or CoT content, and whether these responses differ by training route. CORD tests CoT-suffix corruption and question masking; donor-span replacement tests the answer conditional under another pair of interventions. We patch an entire question or CoT span at the embedding and every decoder block to measure its total substitution effect, rather than to localize a layer. For this all-layer intervention, the patched run is equivalent to evaluating the donor-substituted token sequence, as detailed below.

Protocol overview. For each policy comparison, we retain items answered correctly by both policies. Each policy is evaluated with its own generated CoT, so the sequence for an item is

$$z = [\, p(x);\ m;\ c_{\mathrm{ans}};\ y^* \,],$$

where $p(x)$ is the rendered prompt, m is that policy's CoT, c_{ans} is a fixed final-answer cue, and y^* is the gold answer. The answer is teacher-forced: the model does not generate a new answer during patching; instead, we score the gold-answer tokens. We designate the question tokens inside $p(x)$ as span P and the complete generated CoT as span C. The clean score is the answer's token-averaged log-likelihood,

$$\ell_{\mathrm{clean}}(x) := \frac{1}{|y^*|} \sum_{t=1}^{|y^*|} \log p_\theta(y_t^* \mid p(x), m, c_{\mathrm{ans}}, y_{<t}^*)\,. \tag{28}$$

We call z the recipient sequence. A donor supplies replacement tokens for the selected span: another matched item's question or CoT in the main intervention, or the other policy's CoT for the same correctly answered item in the control. Question and CoT substitutions are run separately. The other span's tokens, answer cue, and teacher-forced gold answer remain those of the recipient, although their later representations can respond to the substituted content.

Formal intervention. For the between-item conditions, the next example in the fixed matched-item ordering provides the donor question or CoT. Let d^S denote its tokens for span $S \in \{P, C\}$. Because recipient and donor spans generally have different lengths, $\mathrm{Fit}(d^S, |S|)$ truncates a longer donor or cyclically repeats a shorter one to fill exactly $|S|$ positions. This keeps the total sequence length and all downstream positional indices fixed. The substituted sequence is

$$\widetilde{z}^S := \mathrm{Replace}\big(z, S, \mathrm{Fit}(d^S, |S|)\big)\,. \tag{29}$$

In the implementation, a *capture pass* runs $\widetilde{z}^S$ and saves the states at span S after the embedding and every decoder block. A *patched pass* runs z but replaces those states with the captured values at each site. Since z and $\widetilde{z}^S$ differ only at S, the embedding output after this replacement is the same as that of $\widetilde{z}^S$ at every position. Subsequent block outputs therefore also coincide. Up to numerical precision, the patched answer score $\ell_S(x)$ equals the token-averaged log-likelihood of y^* under $\widetilde{z}^S$.

Readout. We define

$$\Delta_S(x) := \ell_{\mathrm{clean}}(x) - \ell_S(x), \qquad S \in \{P, C\}. \tag{30}$$

The donor can occasionally increase rather than decrease the answer likelihood, so the primary sensitivity measure is $|\Delta_S|$. A large $|\Delta_P|$ means that substituting question content while keeping the CoT tokens fixed substantially changes the original answer score; $|\Delta_C|$ has the analogous meaning for CoT substitution with question tokens fixed. Policy-level effects in Section 7 compare mean absolute sensitivity with that of π_{RLVR} on the same common-correct items. We compare each substitution effect across policies. Because question and CoT substitutions differ in length and content, we do not compare their raw effect sizes with each other.

Controls and scope. Donors are assigned mechanically as the next matched item, rather than selected by their effect. Whole-span substitution measures how replacing question or CoT content changes the original answer score. Because the entire CoT is replaced, including its concluding answer expression, this analysis does not isolate individual reasoning steps or localize the effect to a particular layer.

E.2 Same-Problem CoT Control

Between-item CoT substitution replaces a trace with one written for a different problem. To examine whether the policy-level effect persists when the problem is held fixed, we also substitute the other policy's CoT for the same question on items both policies answer correctly. We score the original gold answer and compare each policy's change in answer likelihood, as in the main analysis.

Across the four datasets, DCO's increase relative to π_{RLVR} is $+3.120$ to $+3.522$ under between-item substitution, but only $+0.276$ to $+0.417$ under same-problem substitution. Thus, the large between-item change is not explained solely by replacing one CoT with another. The same-problem comparison does not isolate wording, since the two CoTs can also differ in reasoning content.

Table 14: Same-item CoT control on correctly answered items. Values are paired DCO-minus-π_{RLVR} differences.

Task	$\Delta^{C}_{\mathrm{patch}}$	$\Delta^{C,\mathrm{same}}_{\mathrm{patch}}$
GSM8K	$+3.522_{\pm 0.316}{}^{*}$	$+0.341_{\pm 0.047}{}^{*}$
MATH500	$+3.120_{\pm 0.395}{}^{*}$	$+0.276_{\pm 0.082}{}^{*}$
DAPO-val	$+3.351_{\pm 0.388}{}^{*}$	$+0.374_{\pm 0.206}{}^{*}$
DAPO-test	$+3.365_{\pm 0.312}{}^{*}$	$+0.417_{\pm 0.129}{}^{*}$

F Qualitative Examples

We show two held-out problems on which all five policies generate nearly identical reasoning and the same correct answer, yet CORD differs depending on how the policy was trained. Each policy's greedy CoT is shown in full; relative to π_{RLVR}, **bold** marks inserted or replaced words and ~~struck-out~~ words are omitted.

Example 1: MATH500 **Answer: 4**

If $2^8 = 4^x$, what is the value of x?

All five policies reach the correct answer with the same solution steps; the other policies' CoTs differ from π_{RLVR}'s in at most 5 words. Nevertheless, π_{RLVR}, DPO and Scalar credit (A+B) all have CORD 0.947, whereas Naive direct (B) and DCO lower it to 0.231 and 0.125, respectively.

π_{RLVR} **(outcome RLVR)** CORD **= 0.947**

To solve the equation $2^8 = 4^x$, we need to express both sides of the equation with the same base. Notice that 4 can be written as 2^2. Therefore, we can rewrite the equation as follows:

$$2^8 = (2^2)^x$$

Next, we use the property of exponents $(a^m)^n = a^{mn}$ to simplify the right side of the equation:

$$2^8 = 2^{2x}$$

Since the bases are the same, we can set the exponents equal to each other:

$$8 = 2x$$

Now, we solve for x by dividing both sides of the equation by 2:

$$x = \frac{8}{2} = 4$$

Thus, the value of x is $\boxed{4}$.

DPO CORD = **0.947**

To solve the equation $2^8 = 4^x$, we need to express both sides of the equation with the same base. Notice that 4 can be written as 2^2. Therefore, we can rewrite the equation as follows:

$$2^8 = (2^2)^x$$

Next, we use the property of exponents $(a^m)^n = a^{mn}$ to simplify the right side of the equation:

$$2^8 = 2^{2x}$$

Since the bases are the same, we can set the exponents equal to each other:

$$8 = 2x$$

Now, ~~we~~ solve for x by dividing both sides ~~of the equation~~ by 2:

$$x = \frac{8}{2} = 4$$

Thus, the value of x is $\boxed{4}$.

Scalar credit (A+B) CORD = **0.947**

To solve the equation $2^8 = 4^x$, we need to express both sides of the equation with the same base. Notice that 4 can be written as 2^2. Therefore, we can rewrite the equation as follows:

$$2^8 = (2^2)^x$$

Next, we use the property of exponents $(a^m)^n = a^{mn}$ to simplify the right side of the equation:

$$2^8 = 2^{2x}$$

Since the bases are the same, we can set the exponents equal to each other:

$$8 = 2x$$

Now, we solve for x by dividing both sides of the equation by 2:

$$x = \frac{8}{2} = 4$$

Thus, the value of x is $\boxed{4}$.

Naive direct (B) CORD = **0.231**

To solve the equation $2^8 = 4^x$, we need to express both sides of the equation with the same base. **We know** that 4 can be written as 2^2. **So,** we can rewrite the equation as follows:

$$2^8 = (2^2)^x$$

Next, we use the property of exponents $(a^m)^n = a^{mn}$ to simplify the right side of the equation:

$$2^8 = 2^{2x}$$

Since the bases are the same, we can set the exponents equal to each other:

$$8 = 2x$$

Now, we solve for x by dividing both sides of the equation by 2:

$$x = \frac{8}{2} = 4$$

So, the value of x is $\boxed{4}$.

DCO CORD = **0.125**

To solve the equation $2^8 = 4^x$, we need to express both sides of the equation with the same base. Notice that 4 can be written as 2^2. Therefore, we can rewrite the equation as follows:

$$2^8 = (2^2)^x$$

Next, we use the property of exponents $(a^m)^n = a^{mn}$ to simplify the right side of the equation:

$$2^8 = 2^{2x}$$

Since the bases are the same, we can set the exponents equal to each other:

$$8 = 2x$$

Now, we solve for x by dividing both sides of the equation by 2:

$$x = \frac{8}{2} = 4$$

Thus, the value of x is $\boxed{4}$.

Example 2: GSM8K **Answer: 8**

A candle melts by 2 centimeters every hour that it burns. How many centimeters shorter will a candle be after burning from 1:00 PM to 5:00 PM?

All five policies reach the correct answer with the same solution steps; the other policies' CoTs differ from π_{RLVR}'s in at most 2 words. Nevertheless, π_{RLVR}, DPO and Scalar credit (A+B) all have CORD 0.462, whereas Naive direct (B) and DCO lower it to 0.158 and 0.077, respectively.

π_{RLVR} (outcome RLVR) CORD = **0.462**

To determine how many centimeters shorter the candle will be after burning from 1:00 PM to 5:00 PM, we need to follow these steps:
1. Calculate the total time the candle burns.
2. Determine how much the candle melts in that time.
Step 1: Calculate the total time the candle burns.
The candle burns from 1:00 PM to 5:00 PM. The total time is:

$$5:00\text{ PM} - 1:00\text{ PM} = 4\text{ hours}$$

Step 2: Determine how much the candle melts in that time.
The candle melts by 2 centimeters every hour. So, in 4 hours, the candle will melt:

$$2\text{ cm/hour} \times 4\text{ hours} = 8\text{ cm}$$

Therefore, the candle will be $\boxed{8}$ centimeters shorter after burning from 1:00 PM to 5:00 PM.

DPO CORD = **0.462**

To determine how many centimeters shorter the candle will be after burning from 1:00 PM to 5:00 PM, we need to follow these steps:
1. Calculate the total time the candle burns.
2. Determine how much the candle melts in that time.
Step 1: Calculate the total time the candle burns.
The candle burns from 1:00 PM to 5:00 PM. The total time **it burns** is:

$$5:00\text{ PM} - 1:00\text{ PM} = 4\text{ hours}$$

Step 2: Determine how much the candle melts in that time.
The candle melts by 2 centimeters every hour. So, in 4 hours, the candle will melt:

$$2\text{ cm/hour} \times 4\text{ hours} = 8\text{ cm}$$

Therefore, the candle will be $\boxed{8}$ centimeters shorter after burning from 1:00 PM to 5:00 PM.

Scalar credit (A+B) CORD = **0.462**

To determine how many centimeters shorter the candle will be after burning from 1:00 PM to 5:00 PM, we need to follow these steps:
1. Calculate the total time the candle burns.
2. Determine how much the candle melts in that time.
Step 1: Calculate the total time the candle burns.
The candle burns from 1:00 PM to 5:00 PM. The total time is:

$$5:00\text{ PM} - 1:00\text{ PM} = 4\text{ hours}$$

Step 2: Determine how much the candle melts in that time.
The candle melts by 2 centimeters every hour. So, in 4 hours, the candle will melt:

$$2\text{ cm/hour} \times 4\text{ hours} = 8\text{ cm}$$

Therefore, the candle will be $\boxed{8}$ centimeters shorter after burning from 1:00 PM to 5:00 PM.

Naive direct (B) CORD = **0.158**

To determine how many centimeters shorter the candle will be after burning from 1:00 PM to 5:00 PM, we need to follow these steps:
1. Calculate the total time the candle burns.
2. Determine how much the candle melts in that time.
Step 1: Calculate the total time the candle burns.
The candle burns from 1:00 PM to 5:00 PM. The total time **it burns** is:

$$5:00\text{ PM} - 1:00\text{ PM} = 4\text{ hours}$$

Step 2: Determine how much the candle melts in that time.
The candle melts by 2 centimeters every hour. So, in 4 hours, the candle will melt:

$$2\text{ cm/hour} \times 4\text{ hours} = 8\text{ cm}$$

Therefore, the candle will be $\boxed{8}$ centimeters shorter after burning from 1:00 PM to 5:00 PM.

DCO CORD = **0.077**

To determine how many centimeters shorter the candle will be after burning from 1:00 PM to 5:00 PM, we need to follow these steps:
1. Calculate the total time the candle burns.
2. Determine how much the candle melts in that time.
Step 1: Calculate the total time the candle burns.
The candle burns from 1:00 PM to 5:00 PM. The total time is:

$$5:00\text{ PM} - 1:00\text{ PM} = 4\text{ hours}$$

Step 2: Determine how much the candle melts in that time.
The candle melts ~~by~~ 2 centimeters every hour. So, in 4 hours, the candle will melt:

$$2\text{ cm/hour} \times 4\text{ hours} = 8\text{ cm}$$

Therefore, the candle will be $\boxed{8}$ centimeters shorter after burning from 1:00 PM to 5:00 PM.