# Right Answers for the Right Reasons: Coherence-Enhanced Policy Optimization for Large Language Models

**Anonymous ACL submission**

## Abstract

Large language models (LLMs) trained with reinforcement objectives often achieve superficially correct answers via shortcut strategies, pairing correct outputs with spurious or unfaithful reasoning and degrading under small input perturbations. We introduce *Coherence-Enhanced Policy Optimization* (CE-PO), a model agnostic reward fusion framework that augments policy optimization with a differentiable *Jacobian coherence* signal along the generation pathway from prompt ($Z$) to rationale ($X$) to answer ($Y$). CE-PO estimates internal gradient influence through Jacobian-based sensitivities, uses a reshuffling-based residualization procedure to reduce shortcut-driven reward hacking, and fuses the resulting coherence score with task-accuracy via an $L_p$ combiner that controls the strength of coherence regularization. The unified reward integrates with PPO/GRPO without architectural changes. Across multiple reasoning benchmarks, CE-PO improves accuracy over baselines by 5.49% on average (up to 9.58%), while increasing the model's faithful reasoning capability (12.14 % on average across five backbones).

## 1 Introduction

Outcome centric reinforcement training of large language models (LLMs), whether via standard policy optimization or preference-based tuning, (Ouyang et al., 2022; Schulman et al., 2017; Shao et al., 2024) optimizes *what* answer is produced but places little pressure on *how* that answer is obtained (Sim et al., 2025). As a result, models routinely learn to "game" reward signals (e.g., by exploiting length, format, or lexical overlap) and to pair correct answers with unfaithful or spurious reasoning traces (Wen et al., 2024; Arjovsky et al., 2019). This failure mode is not merely theoretical: without intervention, we observe models can game both accuracy proxies and gradient-based sensitivity surrogates (e.g., by guessing final answer or maximizing output length as reported in section 3). This creates a tension between optimizing for outcome-based rewards and enforcing reasoning-level robustness, leading to severe hallucination risks.

We argue that robust reasoning cannot be certified by output accuracy alone. Instead, robust reasoning requires *coherence* along the generation pathway, echoing the broader principle of being *right for the right reasons* (Ross et al., 2017), but extending it from feature-attribution constraints to generative reasoning traces. A *coherence* reasoning trace should satisfy three conditions: (i) the prompt genuinely informs the rationale, (ii) the prompt along with rationale in turn genuinely informs the final answer, and (iii) small, semantically relevant edits to the prompt or rationale induce correspondingly appropriate changes downstream, whereas irrelevant edits do not. Concretely, let $(Z \to X \to Y)$ denote the intended chain from prompt ($Z$), to rationale ($X$), to answer ($Y$)[1], if the model's internal influence respects this order, chain-of-thought (CoT) traces are more likely to be *right for the right reasons* and to generalize under distribution shift.

However, current RL objectives offer no signal that rewards such coherence. Related efforts to promote faithful reasoning typically rely on external supervision via invariance/IRM approaches (Kamath et al., 2021; Koh and Liang, 2017) or post-hoc decoding-time control (Chi et al., 2024). However, they often require external rationale supervision or only control surface-level generations. In contrast, we seek a *differentiable, annotation-free, training-time* internal signal that explicitly rewards influence coherence across $Z \to X \to Y$ and integrates the coherence-enhanced reward shaping into standard policy optimization. *Differentiability* lets us com-

[1]Note that $X$ and $Y$ can be unambiguously segmented in LLM outputs via special tokens, enabled by the model's instruction-following ability.

pute pathway coherence automatically for each rollout from model gradients; *annotation-free* scoring avoids human rationale labels or external verifiers; and *training-time* integration turns this signal into a reward that directly shapes PPO/GRPO updates rather than merely evaluating generations post hoc.

However, reaching the above goal raises three challenges. (i) Measurability: existing faithfulness metrics (saliency, raw input-gradient attributions) are fragile, may fail sanity checks, and are poorly suited as training signals (Adebayo et al., 2018); Prior "right-for-the-right-reasons" regularizers typically require human rationales and constrain input-to-output links in discriminative models (Ross et al., 2017), rather than the generative $Z \to X \to Y$ pathway. We therefore need a differentiable internal proxy for directed influence along $Z \to X$, $X \to Y$, and $Z \to Y$. (ii) Robustness: raw gradient signals are easily inflated by non-semantic factors (token frequency, position, format), encouraging shortcuts (Geirhos et al., 2020) and reward hacking (Amodei et al., 2016); meanwhile, CoT traces may look plausible while remaining unfaithful under perturbations (Chi et al., 2024). Hence the signal must denoise nuisance directions rather than trust raw sensitivities. (iii) Objective balance: coherence-oriented regularization can trade off with task accuracy when mis-specified (Arjovsky et al., 2019; Rosenfeld et al., 2020; Kamath et al., 2021), whereas optimizing only task reward leaves shortcuts unchecked (Amodei et al., 2016). We therefore need a tunable mechanism that controls the strength of coherence regularization during task optimization. Due to the length limitation, the thorough investigation of related works are place at Appendix A

**Our approach.** We introduce Coherence-Enhanced Policy Optimization (CE-PO)[2], a drop-in reward-shaping scheme for PPO/GRPO. CE-PO measures pathway coherence by using differentiable, model-internal Jacobians to estimate how upstream spans affect downstream generation over the $Z \to X$ and $X \to Y$ links; improves robustness through reshuffling-based residualization that filters superficial cues such as length and format; and aggregates the resulting coherence score with task accuracy by a tunable $L_p$ combiner. This yields a single training reward that controls the strength of coherence regularization during PPO/GRPO training, without any model architecture changes.

[2]Code is available at https://anonymous.4open.science/r/causalrl-CC6B/

Our proposed CE-PO offers three advantages:

- A differentiable, model-internal *Jacobian coherence* reward over the $Z \to X \to Y$ pathway, coupled with *Residualization* to suppresses shortcut sensitivities that enable *reward hacking*.
- A tunable $L_p$ *combiner* that controls the strength of coherence regularization relative to task accuracy, yielding a drop-in objective for PPO/GRPO with efficient reward computation.
- Consistent empirical gains across tasks, supported by ablation analyses and out-of-distribution evaluations (see section 3).

## 2 The Proposed CE-PO Framework

**Overview.** In this section, we introduce our *CE-PO* framework, with each rollout consisting of prompt $Z$, rationale $X$, and answer $Y$. As shown in Figure 1, our framework processes each rollout through three stages: ① **Pathway-coherence estimation**: computing Jacobian-based sensitivities that measure how perturbations to an earlier span's embeddings affect the likelihood of a later span. These sensitivities provide differentiable, model-internal proxies for $Z \to X$, $X \to Y$, and $Z \to Y$. ② **Robustness via residualization**: using Jacobian sensitivities directly as rewards can invite *reward hacking*, where superficial cues such as length or format dominate the signal Figure 2. To improve robustness, we use reshuffling-based residualization to remove spurious effects associated with these cues. ③ **Normalization and fusion**: We normalize the residualized Jacobian coherence into a stable scalar *coherence score* and then fuse it with task accuracy using an $L_p$ combiner, yielding a unified reward with a tunable weight on coherence regularization, optimized directly with PPO/GRPO.

### 2.1 Jacobian Coherence Estimation

For a differentiable language model $\Theta$, we aim to reward not only task accuracy on the final answer $Y$ but also coherence along the generated pathway. Specifically, the prompt $Z$ should shape the rationale $X$, and the rationale $X$ together with $Z$ should support the answer $Y$. We operationalize this notion with Jacobians, using them to measure how changes in an earlier span affect the model's likelihood of a later span. For two spans $A$ and $B$ in a rollout, the intuition is: if $A$ genuinely informs $B$, then perturbing the representations of $A$ should change the model's likelihood of generating $B$.

Let $E(A) \in \mathbb{R}^{T_A \times d}$ denote the token-embedding matrix of span $A$, where $T_A$ is its length and $d$ is

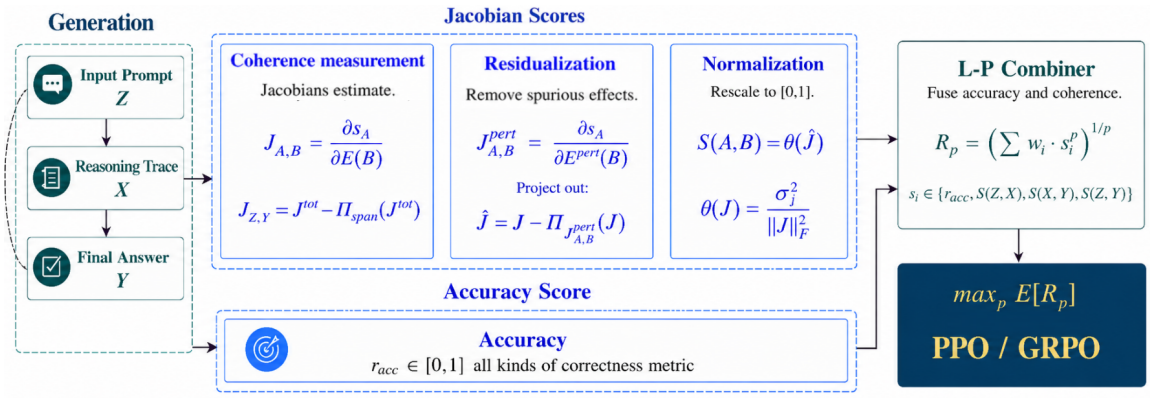



Figure 1: **CE-PO pipeline (left → right).** *Generation:* $Z \to X \to Y$ (X/Y split by a special token; see Fig. 15). *Coherence measurement:* Jacobians estimate pathway coherence over $Z \to X$, $X \to Y$, $Z \to Y$. *Residualization:* reshuffling removes spurious length/format directions. *Normalization:* spectral energy maps Jacobians to [0,1] scalars. *Fusion:* an $L_p$ combiner merges $S(A, B)$ with accuracy $r_{\text{acc}}$ for PPO/GRPO optimization.

the hidden size. We first define the log-likelihood scores of the rationale $X$ and answer $Y$ as

$$s_X = \sum_{t \in X} \log p_\Theta(x_t \mid Z, x_{<t}),$$
$$s_Y = \sum_{t \in Y} \log p_\Theta(y_t \mid Z, X, y_{<t}). \tag{1}$$

Here, $x_{<t}$ and $y_{<t}$ denote the generated prefixes before token $t$ within the rationale and answer spans, respectively.

The *Jacobian influence* from span $A$ to span $B$ is defined as the sensitivity of $B$'s log-likelihood score to the embeddings of $A$:

$$J_{AB} \triangleq \frac{\partial s_B}{\partial E(A)}. \tag{2}$$

For the intended chain $Z \to X \to Y$, we compute three raw Jacobian-based influence scores:

$$\{J_{ZX}, J_{XY}, J_{ZY}^{\text{total}}\} = \{\frac{\partial s_X}{\partial E(Z)}, \frac{\partial s_Y}{\partial E(X)}, \frac{\partial s_Y}{\partial E(Z)}\}. \tag{3}$$

In addition, to isolate the direct $Z \to Y$ contribution from $J_{ZY}^{\text{total}}$, we need to remove the component routed through $X$. We do so via projection: forming a via-$X$ template from $J_{ZX}$ ($Z \to X$) and the principal direction of $J_{XY}$ ($X \to Y$), then subtracting its projection from the total effect to obtain the residual direct influence:

$$J_{ZY} = J_{ZY}^{\text{total}} - \Pi_{J_{ZX} \odot \bar{u}}\left(J_{ZY}^{\text{total}}\right), \tag{4}$$

where $\bar{u} = \frac{1}{T_X} \sum_{t \in X} \frac{J_{XY}[t,:]}{\|J_{XY}[t,:]\|_2}$, and $\Pi_A(B) = \frac{\langle B, A \rangle_F}{\langle A, A \rangle_F} A$, with $\langle B, A \rangle_F$ indicating the Frobenius inner product and $\odot$ be Hadamard product with column-wise scaling. This is feasible based on the sublinear feature of trasnformer revealed in previous literature (Feng et al., 2022).

These Jacobian scores provide a first-order Taylor approximate, local proxy for how span $A$ influences the likelihood of span $B$ (i.e., Direct Causal Effect), while avoiding the cost and instability of higher-order estimators in LLM training (Pearlmutter, 1994; Martens, 2010; Koh and Liang, 2017). For $Z \to X \to Y$, $J_{ZX}$, $J_{XY}$, and $J_{ZY}$ are the raw components of the eventual coherence score, capturing prompt-to-rationale, rationale-to-answer, and direct prompt-to-answer support, respectively. We next residualize these raw scores to filter superficial effects before normalizing and aggregating them into a scalar reward component.

## 2.2 Residualization for Robust Coherence

The raw Jacobian scores defined above are computed on observed rollouts and may therefore conflate semantic dependence with superficial factors such as style, formatting, length, and token-frequency statistics. Theoretically, we show in Appendix C.2 that raw Jacobian scores can be length-sensitive. Empirically, Figure 2 shows that Non-CF-GRPO, a variant without coherence filtering, exhibits clear length hacking, with longer generations receiving inflated Jacobian rewards. We therefore introduce residualization to filter such superficial effects before reward construction.

For each link $(A, B) \in \mathcal{L} := \{(Z, X), (X, Y), (Z, Y)\}$, let $M(A, B)$ denote the intermediate span on the $A \to B$ path, with $M(Z, Y) = \{X\}$ and $M(A, B) = \emptyset$ otherwise. To isolate superficial effects, we construct source-side reshuffled rollouts by permuting the tokens of $A$ and, when present, $M$. This breaks the semantic alignment with $B$ while preserving low-level statistics such as length and unigram

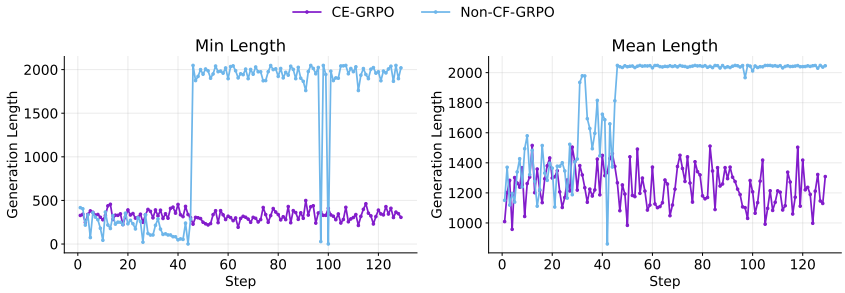



Figure 2: Training trajectories comparison: **Non-CF-GRPO** (raw $J_{AB}$ score) vs **CE-GRPO** ($\widehat{J}_{AB}$ from Equation 6) on Qwen-3-4B-Thinking with `max_tokens`=2048, in terms of minimum (left) and mean (right) generation length. Non-CF-GRPO curves plateau as generations approach the token limit, indicating length-driven reward hacking, while CE-GRPO doesn't.

frequency. Compared with deletion or random replacement, reshuffling cleanly preserves these surface statistics; Appendix E further shows that the filtered component has near-zero correlation with correctness, suggesting that it primarily captures nuisance rather than semantic signal.

For each reshuffled rollout $(\bar{A}, \bar{M})$, we compute the counterfactual Jacobian

$$J^{\mathrm{cf}}_{AB} = \left.\frac{\partial s_B}{\partial E(A)}\right|_{A, M = \bar{A}, \bar{M}}. \tag{5}$$

We sample multiple reshufflings and average their Jacobians to obtain $\bar{J}^{\mathrm{cf}}_{AB}$. In our experiments, we use four reshufflings per link; larger values can improve stability at additional computational cost.

We then estimate the dominant spurious subspace from the counterfactual Jacobians. Specifically, for each link $(A, B)$, we take the top-$k$ left singular vectors $U_{AB}$ of the averaged counterfactual Jacobian $\bar{J}^{\mathrm{cf}}_{AB}$ and define the projection operator $\Pi_{AB}(J) = U_{AB}U_{AB}^{\top}J$. Following the projection-based removal idea of Ravfogel et al. (2020), we obtain the residualized Jacobian by subtracting the component of the raw Jacobian that lies in this subspace:

$$\widehat{J}_{AB} = J_{AB} - \Pi_{AB}(J_{AB}). \tag{6}$$

For the direct $Z \to Y$ link, we further subtract the component mediated through $X$ before projection.

Figure 2 illustrates why this residualization step is necessary. Non-CF-GRPO uses the raw Jacobian scores $J_{AB}$ directly as the coherence reward, while CE-GRPO replaces them with the residualized scores $\widehat{J}_{AB}$. Without residualization, the reward becomes strongly coupled with generation length: Non-CF-GRPO drives outputs toward the token cap. In contrast, CE-GRPO filters this length effect and keeps generation length controlled while optimizing the reward.

### 2.3 Normalization and Fusion

Finally, we normalize each residualized Jacobian into a scalar coherence score. Specifically, we use the scale-free spectral energy ratio $\phi(J) = \frac{\sigma_1(J)^2}{\|J\|_F^2} \in [0, 1]$, where $\sigma_1(J)$ is the largest singular value (Vershynin, 2018). This maps each residualized Jacobian to a bounded score:

$$S(Z, X) = \phi(\widehat{J}_{ZX}),\ S(X, Y) = \phi(\widehat{J}_{XY}),\ S(Z, Y) = \phi(\widehat{J}_{ZY}). \tag{7}$$

Let $r_{\mathrm{acc}} \in [0, 1]$ be a task accuracy score on $Y$. We combine this task accuracy with the three coherence scores using a weighted $L_p$ combiner:

$$R_p = \left(w_0\, r_{\mathrm{acc}}^p + w_1\, S(Z, X)^p + w_2\, S(X, Y)^p + w_3\, S(Z, Y)^p\right)^{1/p}, \tag{8}$$

with $p \in \mathbb{R}$, weights $w_i \geq 0$, and $\sum_{i=0}^{3} w_i = 1$. The exponent $p$ controls the aggregation geometry: larger $p$ emphasizes high-scoring channels, smaller $p$ emphasizes low-scoring bottlenecks, and $p \to 0$ recovers a weighted geometric mean. Thus, $p$ determines whether optimization rewards already strong components or pushes weaker task/coherence components upward.

We use $R_p$ as the unified reward for PPO/GRPO and maximize $\mathbb{E}[R_p]$. Its channels capture complementary aspects of reasoning: task reward evaluates the final answer, while residualized Jacobian scores assess internal support along $Z \to X$, $X \to Y$, and $Z \to Y$. As shown in Figure 6, optimizing a single channel can improve one aspect while degrading others. The $L_p$ combiner jointly aggregates these channels, with weights setting component importance and $p$ controlling whether optimization emphasizes high-scoring channels or low-scoring bottlenecks. Figure 13 illustrates that CE-PO improves reasoning without simply inducing template-following.

As a practical guardrail, the $L_p$ exponent lets end-users adjust which channel the optimization

emphasizes: $p \to +\infty$ emphasizes the strongest channel (typically correctness), while $p \to -\infty$ emphasizes the weakest channel (typically coherence). A single scalar thus controls the optimization target without retraining or re-weighting.

Figure 3: Tunable emphasis enabled by $p$-norm interpolation in $R_p$.

**Choices of $r_{\text{acc}}$.** We study different choices of base-reward $r_{\text{acc}}$, including exact match, DeBERTa-based scoring, and other task metrics. The results in Table 2 validate that CE-PO's gains are robust across base-reward choices rather than tied to a specific accuracy metric.

**Theoretical Guarantee, Storage, and Runtime.** Theoretically, Appendix B derives a TRPO-style lower bound for CE-PO: residualization shrinks the surrogate-objective penalty by projecting out nuisance directions, and policy updates confined to the nuisance subspace incur zero penalty. CE-PO thus inherits PPO/GRPO's monotonic-improvement guarantee while suppressing shortcut-driven updates. Computationally, end-to-end CE-GRPO training on Llama-3-8B-Instruct adds only $\sim$13% wall-clock time and $\sim$7% peak GPU memory over vanilla GRPO, functioning as an efficient plug-in reward shaping method (Appendix C). CE-PO is thus practically deployable at scale without architectural changes.

## 3 Experiments

**Experiment Setting.** To avoid random guesses, all models are prompted to generate CoT reasoning (Wei et al., 2022), producing both a reasoning trace $X$ and a final answer $Y$. Response correctness is determined via an LLM-as-Judge protocol (Zheng et al., 2023) using GPT-4o-mini (see prompt in Figure 16, and validity analysis in Appendix D). Beyond accuracy, we evaluate reasoning faithfulness via **Answer Flip Rate**—the fraction of correctly answered instances whose answer changes when the CoT is removed. A higher AFR indicates greater reliance on the reasoning trace, a simple proxy to reasoning faithfulness (Matton et al., 2025).

We evaluate both standard instruction-tuned LLMs and "thinking" variants[3] that emit explicit intermediate reasoning and instruction following. Although compute constraints preclude experiments on 70B-scale models, our evaluation covers compact to mid-sized systems that are widely used in research: Llama-3-8B-Instruct, Llama-3.2-3B-Instruct, Phi-3.5-mini-Instruct, Qwen3-4B-Thinking, and Qwen3-1.7B-Thinking (Dubey et al., 2024; Yang et al., 2025; Abdin et al., 2024). Unless otherwise specified, we use $p$=0.2 and $\mathbf{w}=(0.7, 0.1, 0.1, 0.1)$ in $R_p$ defined in Eq. 8, prioritizing task reward while retaining pathway-coherence signals. Other hyperparameters, dataset details, and overhead are reported in Appendices C, I, and H.

**Training and Validation Dataset.** We train and tune on four reasoning-heavy sets: *BBEHCausal*, a reasoning split from the BIG-Bench Extra Hard dataset, including counterfactual deduction questions; (shoaa kazemi et al., 2025)); *CaseHOLD*, tasks centering on multiple-choice legal holdings given case citation context (Zheng et al., 2021); *MATHHARD*, the level-5 subset targeting multi-step mathematical reasoning with multiple-choice questions (Hendrycks et al., 2021); and *IfQA*, an open-domain QA under counterfactual *if*-clause presuppositions requiring hypothetical reasoning (Yu et al., 2023). Note that all datasets are formatted as question answer pairs, examples of them are presented respectively in Table 9.

**Testing Dataset.** For same-field evaluation, we use *BBEHMATH* (challenging multistep arithmetic), where we filter shorter and more trivial questions and normalize alphabetic numerals to digits with unfolded calculation for comparability on this dataset. We further assess generalization with *CLadder*, covering association/intervention/counterfactual queries (Jin et al., 2023), *LegalBench* (case-understanding split) (Guha et al., 2023) analyzing the reasoning of legal cases, and *LogiQA*, a multiple-choice dataset emphasizing deductive logical reasoning sourcing from officer entrance exams (Liu et al., 2023a).

**Main Experiment Results.** We compare CE-PPO and CE-GRPO against vanilla (zero-shot CoT), Naive PPO and Naive GRPO, Self-Consistency (Wang et al., 2022), and Math-Shepherd (Wang et al., 2024). Both are implemented atop our GRPO infrastructure: Self-Consistency selects the majority-voted answer

[3]We use Instruct/Thinking models to reliably separate $X$ and $Y$; their resistance to RL improvement makes CE-PO's gains more meaningful.

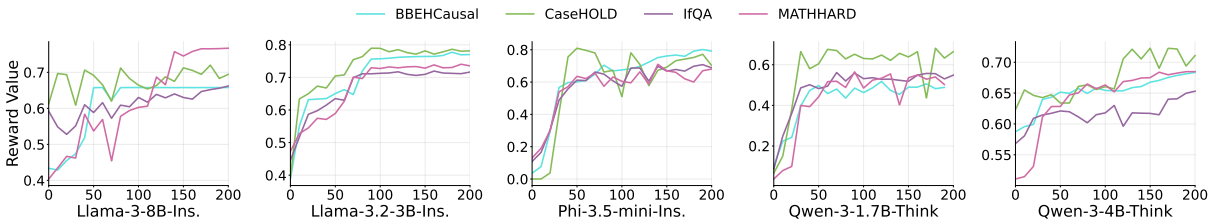



Figure 4: Reward curves of CE-PPO training on validation datasets across five models.

across the rollout group, while Math-Shepherd trains a PRM that estimates the probability of reaching a correct final answer from each intermediate step, and averages these step-level scores as the GRPO reward. As shown in Table 1, CE-PO variants consistently outperform all baselines across five models and four benchmarks, achieving average accuracy gains of 5.49% over vanilla with substantially higher AFR.

**Robustness of Base Reward $r_{\text{acc}}$.** CE-PO's $L_p$ combiner accepts a wide range of $r_{\text{acc}} \in [0, 1]$ as feasible base reward. We evaluate three candidate base rewards: BERTScore (Zhang et al., 2019) (token-level semantic similarity to the reference), DeBERTa-v3-large RM (He et al., 2021) (a pre-trained reward model producing a calibrated continuous score), and Exact Match (EM) (Rajpurkar et al., 2016) (string/numeric equality, yielding a graded signal in $[0, 1]$ when averaged over the rollout group without any external model). As Table 2 shows, CE-GRPO consistently improves over standalone GRPO across all three variants, confirming that the Coherence-Enhanced reward shaping is agnostic to the specific choice of $r_{\text{acc}}$.

**Out of Distribution (OOD) Evaluation.** Trained exclusively on reasoning corpora (including both general and math-oriented reasoning), our CE-PO models are stress-tested for OOD generalization on TruthfulQA (Lin et al., 2021) (misconception resistant factuality), CodeMMLU (Manh et al., 2024), and SuperGPQA (Du et al., 2025) (expert level STEM QA), evaluating 200 uniformly random samples from each benchmark. As shown in Fig.5, they deliver consistent, but task-dependent improvements over baselines: effects are strongest on structure-heavy code reasoning, moderate on STEM-oriented SuperGPQA, and smaller yet steady on TruthfulQA, suggesting that coherence regularization chiefly benefits multi-step reasoning while preserving overall robustness.

**Generation Pattern Analysis.** CE-GRPO does not induce uniform gains across all tokens but selectively amplifies reasoning-related words, suggesting that coherence regularization guides the model toward tokens central to stepwise inference rather than surface templates (per-word frequency gaps in Appendix F, Fig. 9). The pairwise structure of the three signals (Fig. 10) further shows that $S(Z, X)$ is nearly independent of $S(X, Y)$ and $S(Z, Y)$, while the latter two are moderately correlated: "evidence infusion" and "answer stability" provide orthogonal yet complementary signals, jointly supporting coherence across the full $Z \to X \to Y$ pathway.

Training reward trajectories in Fig. 6 highlight three regimes: optimizing correctness alone inflates accuracy while collapsing coherence channels; optimizing coherence alone boosts sensitivities but destabilizes training and harms task accuracy (Stiennon et al., 2020; Yuan et al., 2025). CE-GRPO achieves smooth improvements across all three coherence channels while keeping correctness competitive, sustaining the full $Z \to X \to Y$ pathway.

**Ablation Studies.** We ablate the $L_p$ exponent $p$, the component weights $w$ (dropping each individual signal in turn), and inject Gaussian noise into $r_{\text{base}}$. Results in Table 3 yield:

**Insight 1:** Reward hacking emerges whenever any one of the three coherence signals $S(Z, X)$, $S(X, Y)$, $S(Z, Y)$ is dropped. Removing the prompt→rationale link, the rationale→answer link, or the prompt→answer link each causes the ablated variant to underperform even the original (vanilla) model: the coherence loop $Z \to X \to Y$ is no longer closed, so the optimization ceases to be strictly benign. The supporting plot in Fig. 12 shows that initial removal first breaks the prior balance (reward dips) and subsequent updates then discover a length shortcut.

**Insight 2:** Sweeping $p$ recovers two limits: "Correctness-only" ($p$ large) drops the coherence signals and matches a correctness-only objective; "Coherence-only" ($p$ small) collapses to the smaller (residualized Jacobian) channel and emphasizes Jacobian coherence. The arithmetic mean ($p = 1$) sits

Table 1: **Main results across models, RL methods, and datasets.** We report Accuracy (%) and Answer Flip Rate (AFR, %). AFR measures the fraction of correctly answered instances whose answer changes when the CoT is removed; higher AFR indicates greater reliance on reasoning. We repeatedly generate 4 times for each trained model, reporting mean $\pm$ std.

| Model | Method | Accuracy (%) | | | | Avg. | AFR (%) |
|---|---|---|---|---|---|---|---|
| | | BBEHMATH | CLadder | LegalBench | LogiQA | | |
| **Qwen-3-1.7B** | Vanilla | $1.36 \pm 1.96$ | $34.77 \pm 0.68$ | $57.80 \pm 2.59$ | $50.00 \pm 0.45$ | 35.98 | 43.02 |
| | Self-Consistency | $3.64 \pm 1.59$ | $37.50 \pm 0.91$ | $59.63 \pm 1.95$ | $53.41 \pm 0.68$ | 38.55 | 44.85 |
| | Math-Shepherd | $8.09 \pm 1.14$ | $44.86 \pm 0.64$ | $65.47 \pm 1.30$ | $54.77 \pm 0.45$ | 43.30 | 45.61 |
| | Naive PPO | $5.91 \pm 1.29$ | $37.27 \pm 0.91$ | $58.26 \pm 1.95$ | $52.50 \pm 0.45$ | 38.48 | 42.27 |
| | Naive GRPO | $3.64 \pm 1.29$ | $35.00 \pm 0.52$ | $60.67 \pm 0.94$ | $57.04 \pm 0.68$ | 39.09 | 47.50 |
| | CE-PPO | $\mathbf{10.45 \pm 0.52}$ | $\mathbf{49.32 \pm 0.68}$ | $60.67 \pm 1.05$ | $\mathbf{58.05 \pm 1.14}$ | 44.62 | 59.18 |
| | CE-GRPO | $10.00 \pm 1.29$ | $44.77 \pm 0.91$ | $\mathbf{70.64 \pm 1.30}$ | $56.82 \pm 0.45$ | **45.56** | **63.21** |
| **Qwen-3-4B** | Vanilla | $6.82 \pm 0.45$ | $42.95 \pm 0.91$ | $74.77 \pm 1.95$ | $80.45 \pm 0.68$ | 51.25 | 38.4 |
| | Self-Consistency | $7.95 \pm 0.68$ | $45.23 \pm 0.64$ | $76.42 \pm 1.30$ | $82.27 \pm 0.91$ | 52.97 | 39.8 |
| | Math-Shepherd | $8.41 \pm 0.52$ | $46.82 \pm 0.45$ | $77.98 \pm 0.94$ | $81.36 \pm 0.45$ | 53.64 | 40.3 |
| | Naive PPO | $5.00 \pm 1.05$ | $45.00 \pm 0.64$ | $73.64 \pm 0.47$ | $80.00 \pm 0.45$ | 50.91 | 37.9 |
| | Naive GRPO | $6.82 \pm 0.45$ | $40.68 \pm 0.68$ | $76.61 \pm 0.65$ | $83.86 \pm 0.91$ | 51.99 | 39.2 |
| | CE-PPO | $\mathbf{9.55 \pm 0.52}$ | $45.45 \pm 0.68$ | $\mathbf{79.34 \pm 1.05}$ | $79.77 \pm 0.91$ | 53.53 | **50.6** |
| | CE-GRPO | $7.27 \pm 0.52$ | $\mathbf{49.09 \pm 0.64}$ | $76.90 \pm 0.47$ | $\mathbf{84.09 \pm 0.45}$ | **54.34** | 44.6 |
| **Phi-3.5-mini** | Vanilla | $6.13 \pm 0.52$ | $40.23 \pm 0.68$ | $68.35 \pm 1.95$ | $54.09 \pm 0.45$ | 42.20 | 47.8 |
| | Self-Consistency | $7.05 \pm 0.45$ | $\mathbf{46.50 \pm 0.64}$ | $68.81 \pm 1.30$ | $55.68 \pm 0.68$ | 44.51 | 48.7 |
| | Math-Shepherd | $7.50 \pm 0.68$ | $43.18 \pm 0.91$ | $\mathbf{70.64 \pm 1.05}$ | $54.55 \pm 0.45$ | 43.97 | 49.4 |
| | Naive PPO | $6.59 \pm 0.91$ | $43.40 \pm 0.87$ | $66.97 \pm 1.95$ | $54.77 \pm 0.68$ | 42.93 | 49.2 |
| | Naive GRPO | $5.45 \pm 0.87$ | $41.59 \pm 0.64$ | $68.81 \pm 1.30$ | $52.73 \pm 0.45$ | 42.15 | 48.5 |
| | CE-PPO | $\mathbf{8.41 \pm 0.64}$ | $42.95 \pm 0.91$ | $70.18 \pm 1.95$ | $55.90 \pm 0.68$ | 44.36 | 57.3 |
| | CE-GRPO | $6.81 \pm 0.64$ | $\mathbf{45.45 \pm 0.45}$ | $70.18 \pm 0.65$ | $\mathbf{57.95 \pm 0.45}$ | **45.10** | **58.6** |
| **Llama-3.2-3B** | Vanilla | $3.63 \pm 0.91$ | $30.00 \pm 0.68$ | $45.41 \pm 0.47$ | $41.59 \pm 0.45$ | 30.16 | 3.99 |
| | Self-Consistency | $5.45 \pm 0.68$ | $32.27 \pm 0.45$ | $47.25 \pm 0.94$ | $43.41 \pm 0.91$ | 32.10 | 4.15 |
| | Math-Shepherd | $7.50 \pm 0.52$ | $34.09 \pm 0.64$ | $50.00 \pm 1.30$ | $44.55 \pm 0.68$ | 34.04 | 4.52 |
| | Naive PPO | $6.59 \pm 0.68$ | $33.63 \pm 0.64$ | $45.87 \pm 1.95$ | $38.41 \pm 0.45$ | 31.12 | 4.39 |
| | Naive GRPO | $6.50 \pm 0.45$ | $31.36 \pm 0.91$ | $42.73 \pm 2.57$ | $44.09 \pm 0.68$ | 31.17 | 5.17 |
| | CE-PPO | $\mathbf{10.67 \pm 1.14}$ | $\mathbf{43.64 \pm 0.68}$ | $\mathbf{55.05 \pm 1.30}$ | $\mathbf{48.18 \pm 0.91}$ | **39.38** | 19.44 |
| | CE-GRPO | $7.95 \pm 0.91$ | $35.23 \pm 0.64$ | $54.13 \pm 2.59$ | $47.27 \pm 0.45$ | 36.15 | **23.23** |
| **Llama-3-8B** | Vanilla | $7.05 \pm 0.45$ | $39.54 \pm 0.91$ | $61.93 \pm 1.95$ | $65.68 \pm 0.64$ | 43.55 | 14.10 |
| | Self-Consistency | $8.86 \pm 0.68$ | $41.36 \pm 0.64$ | $64.22 \pm 1.30$ | $67.05 \pm 0.91$ | 45.37 | 14.72 |
| | Math-Shepherd | $11.18 \pm 0.52$ | $42.73 \pm 0.45$ | $66.14 \pm 0.94$ | $68.64 \pm 0.45$ | 47.17 | 15.38 |
| | Naive PPO | $5.27 \pm 0.64$ | $37.95 \pm 0.45$ | $61.93 \pm 0.65$ | $60.91 \pm 0.45$ | 41.52 | 18.37 |
| | Naive GRPO | $10.91 \pm 1.14$ | $39.09 \pm 0.91$ | $57.80 \pm 1.05$ | $67.50 \pm 0.68$ | 43.83 | 13.41 |
| | CE-PPO | $\mathbf{14.86 \pm 1.59}$ | $44.77 \pm 0.45$ | $\mathbf{67.35 \pm 0.79}$ | $68.40 \pm 0.91$ | 48.84 | 23.11 |
| | CE-GRPO | $13.64 \pm 0.91$ | $\mathbf{46.00 \pm 0.64}$ | $66.97 \pm 0.65$ | $\mathbf{71.82 \pm 0.45}$ | **49.61** | **18.37** |

in between. Under a perturbed reward (Gaussian noise $\sigma = 0.05$) the model still maintains satisfactory performance, indicating robustness of the training design.

**Jacobian Signals and Accuracy** Figure 7 shows the distributions of Jacobian-based scores for samples annotated by correctness with LLM-as-Judge with strict judging criteria provided in Figure 16 conducted by GPT-4o. Annotations were collected on 1000 samples generated by Llama-3-8B-Ins. In total, 371 cases were labeled as consistent and 629 as inconsistent.

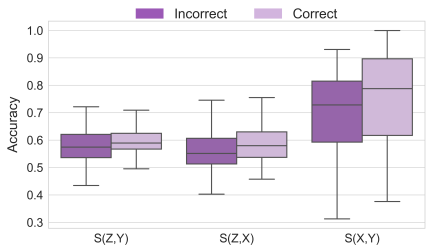



Figure 7: Boxplots of $S(Z, X)$, $S(Z, Y)$, and $S(X, Y)$ for LLM-as-Judge accuracy regarding their causal trajectory.

Table 2: Robustness to base-reward choices on Llama-3-8B-Instruct. CE-GRPO improves over standalone GRPO across all three $r_{\text{acc}}$ variants.

| Base-reward | Method | BBEHMATH | CLadder | LegalBench | LogiQA | Avg. |
|---|---|---|---|---|---|---|
| BERTScore | GRPO | 10.91 | 39.09 | 57.80 | 67.50 | 43.83 |
| | CE-GRPO | **13.64** | **45.00** | **66.97** | **71.82** | **49.36** |
| Exact Match | GRPO | 12.27 | 40.45 | 60.18 | 69.09 | 45.50 |
| | CE-GRPO | **15.45** | **46.36** | **68.35** | **73.41** | **50.89** |
| RM (DeBERTa) | GRPO | 8.39 | 39.54 | 63.00 | 69.58 | 45.13 |
| | CE-GRPO | **14.79** | **45.00** | **67.35** | **72.62** | **49.94** |

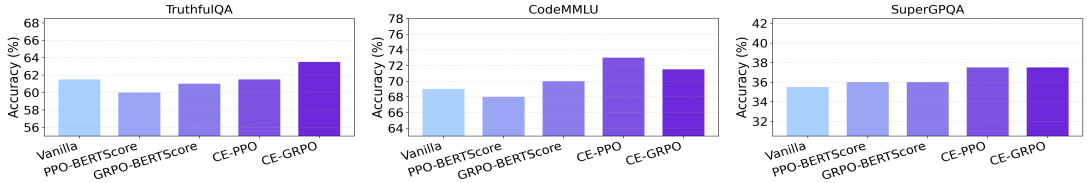



Figure 5: Qwen-3-4B-Thinking accuracy in OOD evaluation scenarios.

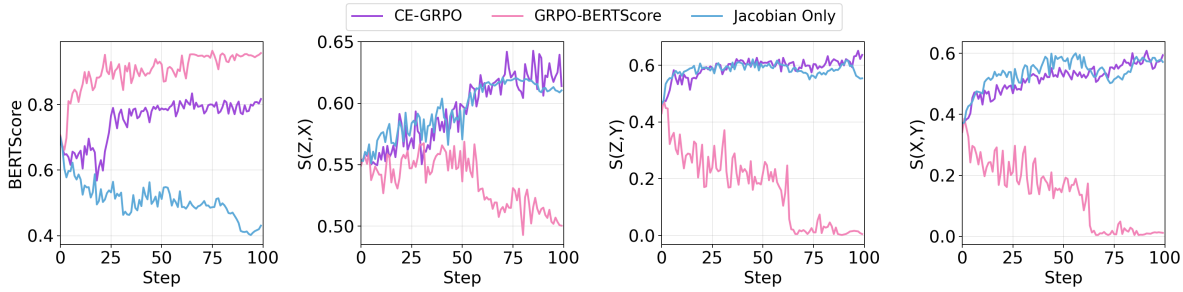



Figure 6: Training trajectories on Llama-3-8B-Instruct: CE-GRPO vs. optimization with only the Jacobian coherence signal or only the accuracy signal (BERTScore). Optimizing correctness alone inflates accuracy while collapsing coherence channels; optimizing coherence alone destabilizes training; CE-GRPO sustains all three coherence channels while keeping correctness competitive.

Table 3: Ablation studies on Llama-3-8B-Instruct. "w/o $S(\cdot,\cdot)$" drops the corresponding coherence link; "Coherence-only" / "Correctness-only" are the $p \to -\infty$ / $p \to +\infty$ limits of the $L_p$ combiner. Avg. is the mean across the four benchmarks.

| **Ablation** | **BBEH** | **CLad.** | **Legal** | **Logi** | **Avg.** |
|---|---|---|---|---|---|
| Correctness-only ($p \to +\infty$) | 6.36 | 39.09 | 56.90 | 67.27 | 42.91 |
| Coherence-only ($p \to -\infty$) | 12.73 | 43.63 | 65.52 | 70.00 | 47.97 |
| Arithmetic mean ($p = 1$) | 10.00 | 50.00 | 60.34 | 64.55 | 46.22 |
| w/o Correctness | 10.91 | 44.54 | 61.93 | 68.18 | 46.89 |
| w/o $S(X,Y)$ (rat.→ans.) | 5.45 | 23.63 | 40.37 | 41.82 | 27.82 |
| w/o $S(Z,Y)$ (pr.→ans.) | 3.63 | 30.00 | 45.87 | 49.09 | 32.15 |
| w/o $S(Z,X)$ (pr.→rat.) | 5.45 | 31.82 | 43.12 | 40.90 | 30.32 |
| Equal weights | 8.18 | 41.82 | 67.24 | 66.36 | 45.90 |
| Perturbed reward ($\sigma = 0.05$) | 7.27 | 40.00 | 65.52 | 69.09 | 45.97 |
| **CE-GRPO (full)** | **13.64** | **47.27** | **66.97** | **72.73** | **50.65** |

To assess differences between the two groups, we conducted t-test with results reported in Table 8. We evaluate whether higher $S(A,B)$ scores align with factual correctness, as validity of these causal signals requires separation under strict accuracy criteria. As in Table 8, all three metrics show significantly higher means for the correct group ($p < 0.01$), confirming that $S(A,B)$ faithfully tracks reasoning–answer validity beyond surface outputs, and by statistically aligns with cognitive accuracy.

## 4 Conclusion

We propose Coherence-Enhanced Policy Optimization (CE-PO), which couples task accuracy with continuous, model-internal Jacobian coherence signals via an $L_p$ combiner. CE-PO is a drop-in modification for PPO/GRPO, reduces reward-hacking behaviors, and yields consistent gains across models and datasets—especially on Thinking backbones. Future work includes richer coherence signals beyond local Jacobians (e.g., Hessian-based) and adaptive scheduling of the accuracy–coherence trade-off.

## Ethics and Reproducibility

CE-PO operates entirely on model-internal signals, requires no human-annotated rationales, and uses only publicly available benchmarks; LLM-as-Judge reliability is validated against human experts (Appendix D), and broader impacts are discussed in Appendix G. All code, training/evaluation pipelines, and prompt templates are released anonymously (link in §1).

## Limitations

Our study has several limitations that suggest directions for future work. First, the Jacobian coherence signal in CE-PO is built from *local* Jacobian sensitivities together with simple source-side Residualization perturbations; this is a tractable but heuristic approximation of long-range influence in the generation pathway, and may miss higher-order dependencies that more expressive sensitivity estimators could capture. Second, the Residualization perturbation procedure relies on reshuffling/mismatch as the perturbation operator, which is effective for suppressing length and positional shortcuts but does not exhaust the space of nuisance directions that could in principle inflate the Jacobian signal. Third, the $L_p$ combiner exposes a tunable accuracy–coherence trade-off via the exponent $p$ and the component weights; we report a default configuration that works well on the benchmarks we study, but adaptively tuning this trade-off during training remains open. Finally, our evaluation focuses on small- to mid-scale instruction-tuned LLMs and on reasoning benchmarks; scaling CE-PO to larger models and to a broader range of downstream tasks is an important next step.

## References


Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Hassan Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Singh Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Caio C'esar Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Parul Chopra, and 70 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *ArXiv*, abs/2404.14219.

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. 2018. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*.

Guangsheng Bao, Hongbo Zhang, Linyi Yang, Cunxiang Wang, and Yue Zhang. 2024. Llms with chain-of-thought are non-causal reasoners. *arXiv preprint arXiv:2402.16048*.

Atilim Gunes Baydin, Barak A. Pearlmutter, Alexey A. Radul, and Jeffrey Mark Siskind. 2018. Automatic differentiation in machine learning: A survey. *Journal of Machine Learning Research*, 18(153):1–43.

Haoang Chi, He Li, Wenjing Yang, Feng Liu, Long Lan, Xiaoguang Ren, Tongliang Liu, and Bo Han. 2024. Unveiling causal reasoning in large language models: Reality or mirage? In *Advances in Neural Information Processing Systems (NeurIPS)*.

Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, King Zhu, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, and 1 others. 2025. Supergpqa: Scaling llm evaluation across 285 graduate disciplines. *arXiv preprint arXiv:2502.14739*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony S. Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 510 others. 2024. The llama 3 herd of models.

Tom Everitt, Victoria Krakovna, Laurent Orseau, Marcus Hutter, and Shane Legg. 2017. Reinforcement learning with a corrupted reward channel. *arXiv preprint arXiv:1705.08417*.

Ruili Feng, Kecheng Zheng, Yukun Huang, Deli Zhao, Michael Jordan, and Zheng-Jun Zha. 2022. Rank diminishing in deep neural networks. *Advances in Neural Information Processing Systems*, 35:33054–33065.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

Gene H. Golub and Charles F. Van Loan. 2013. *Matrix Computations*, 4 edition. Johns Hopkins University Press.

Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N. Rockmore, Diego Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit

Sarfaty, Gregory M. Dickinson, Haggai Porat, Jason Hegland, and 21 others. 2023. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *Preprint*, arXiv:2308.11462.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Xiaodong Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *ArXiv*, abs/2103.03874.

Zhijing Jin, Yuen Chen, Felix Leeb, Luigi Gresele, Ojasv Kamal, Zhiheng Lyu, Kevin Blin, Fernando Gonzalez Adauto, Max Kleiman-Weiner, Mrinmaya Sachan, and Bernhard Scholkopf. 2023. Cladder: Assessing causal reasoning in language models.

Pritish Kamath, Akilesh Tangella, Danica Sutherland, and Nathan Srebro. 2021. Does invariant risk minimization capture invariance? In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 4069–4077. PMLR.

Divyansh Kaushik, Eduard Hovy, and Zachary C. Lipton. 2020. Explaining the efficacy of counterfactually augmented data. *arXiv preprint arXiv:2010.02114*.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *Proceedings of ICML*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

Hanmeng Liu, Jian Liu, Leyang Cui, Zhiyang Teng, Nan Duan, Ming Zhou, and Yue Zhang. 2023a. Logiqa 2.0—an improved dataset for logical reasoning in natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2947–2962.

Haotian Liu and 1 others. 2023b. Llms as narcissistic evaluators: When ego inflates evaluation scores. *arXiv preprint arXiv:2311.01964*.

Dung Nguyen Manh, Thang Phan Chau, Nam Le Hai, Thong T Doan, Nam V Nguyen, Quang Pham, and Nghi DQ Bui. 2024. Codemmlu: A multi-task benchmark for assessing code understanding capabilities of codellms. *CoRR*.

James Martens. 2010. Deep learning via hessian-free optimization. In *Proceedings of ICML*, pages 735–742.

Katie Matton, Robert Osazuwa Ness, John Guttag, and Emre Kıcıman. 2025. Walk the talk? measuring the faithfulness of large language model explanations. *arXiv preprint arXiv:2504.14150*.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. 2018. Mixed precision training. In *International Conference on Learning Representations (ICLR)*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Long Pan and 1 others. 2023. Spontaneous reward hacking in iterative self-refinement. *arXiv preprint arXiv:2307.04549*.

Long Pan and 1 others. 2024. Feedback loops with language models drive in-context reward hacking. *arXiv preprint arXiv:2402.06627*.

Barak A. Pearlmutter. 1994. Fast exact multiplication by the hessian. *Neural Computation*, 6(1):147–160.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 2383–2392.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. 2020. Null it out: Guarding protected attributes by iterative nullspace projection. *arXiv preprint arXiv:2004.07667*.

Elan Rosenfeld, Pradeep Ravikumar, and Andrej Risteski. 2020. The risks of invariant risk minimization. *arXiv preprint arXiv:2010.05761*.

Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. 2017. Right for the right reasons: Training differentiable models by constraining their explanations. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *ArXiv*, abs/1707.06347.

Zhen Shao and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*. Introduces GRPO.

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Kristjanson Duvenaud, Amanda Askell, Samuel R. Bowman, Newton Cheng, Esin Durmus, Zac Hatfield-Dodds, Scott Johnston, Shauna Kravec, Tim Maxwell, Sam McCandlish, Kamal Ndousse, Oliver Rausch, Nicholas Schiefer, Da Yan, Miranda

Zhang, and Ethan Perez. 2023. Towards understanding sycophancy in language models. *ArXiv*, abs/2310.13548.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*.

Mehrangiz shoaa kazemi, Bahare Fatemi, Hritik Bansal, John Palowitch, Chrysovalantis Anastasiou, Sanket Vaibhav Mehta, Lalit K. Jain, Virginia Aglietti, Disha Jindal, Peter Chen, Nishanth Dikkala, Gladys Tyen, Xin Liu, Uri Shalit, Silvia Chiappa, Kate Olszewska, Yi Tay, Vinh Q. Tran, Quoc V. Le, and Orhan Firat. 2025. Big-bench extra hard. In *Annual Meeting of the Association for Computational Linguistics*.

Shang Hong Sim, Tej Deep Pala, Vernon Toh, Hai Leong Chieu, Amir Zadeh, Chuan Li, Navonil Majumder, and Soujanya Poria. 2025. Lessons from training grounded llms with verifiable rewards. *arXiv preprint arXiv:2506.15522*.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel M. Ziegler, Ryan Lowe, and et al. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Sree Harsha Tanneru, Dan Ley, Chirag Agarwal, and Himabindu Lakkaraju. 2024. On the hardness of faithful chain-of-thought reasoning in large language models. *ArXiv*, abs/2406.10625.

Roman Vershynin. 2018. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, F. Xia, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903.

Yilun Wen and 1 others. 2024. Language models learn to mislead humans via rlhf. *arXiv preprint arXiv:2409.12822*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *ArXiv*, abs/2505.09388.

W. Yu, Meng Jiang, Peter Clark, and Ashish Sabharwal. 2023. Ifqa: A dataset for open-domain question answering under counterfactual presuppositions. *ArXiv*, abs/2305.14010.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2025. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Lucia Zheng, Neel Guha, Brandon R. Anderson, Peter Henderson, and Daniel E. Ho. 2021. When does pretraining help?: assessing self-supervised learning for law and the casehold dataset of 53,000+ legal holdings. *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*.

Yujun Zhou, Zhenwen Liang, Haolin Liu, Wenhao Yu, Kishan Panaganti, Linfeng Song, Dian Yu, Xiangliang Zhang, Haitao Mi, and Dong Yu. 2025. Evolving language models without labels: Majority drives selection, novelty promotes variation. *arXiv preprint arXiv:2509.15194*.

## Appendix Contents


**A Related Works** **12**
- A.1 Reward hacking in LLM . . . . . 12
- A.2 Faithful reasoning and shortcut learning in LLMs . . . . . 12

**B Theoretical Guarantees** **13**

**C Additional Analysis** **14**
- C.1 Wall-clocks, storage and complexity of CE-PO. . . . . 14
- C.2 Proof of length-induced bias in base Jacobian . . . . . 15
- C.3 Training objective . . . . . 15
- C.4 Detailed Statistics . . . . . 15

**D Justification of LLM-as-Judge over Exact Match** **16**

**E Ablation of Counterfactual Perturbation** **16**

**F Additional Plots** **16**

**G Broader Impact** **18**

**H Prompt Templates** **18**

**I Dataset Features** **19**

**J The Use of Large Language Models (LLMs)** **19**


# A Related Works

## A.1 Reward hacking in LLM

Reward hacking occurs when models exploit imperfections in reward functions rather than achieving the intended objective (Amodei et al., 2016; Everitt et al., 2017; Zhou et al., 2025). In large language models, this problem emerges prominently under alignment: proxy reward models can be gamed, leading to behaviors such as generating superficially persuasive but incorrect answers (Wen et al., 2024), sycophancy (Sharma et al., 2023), and exploiting evaluator biases in LLM-as-judge settings (Liu et al., 2023b). More recently, iterative refinement has revealed *in-context reward hacking*, where models adapt outputs to maximize evaluation scores rather than genuinely improving reasoning quality (Pan et al., 2023, 2024). Mitigation strategies typically involve improved supervision (e.g., rationale-level feedback (Ross et al., 2017)), evaluator debiasing, or structural interventions such as counterfactual augmentation (Kaushik et al., 2020) and invariance penalties (Arjovsky et al., 2019). However, these often rely on external annotations nor emphasized robustness–accuracy trade-offs. Our method introduces differentiable, counterfactual-enhanced Jacobian coherence signals as internal rewards, jointly optimizing accuracy and coherence to directly curb shortcut exploitation.

## A.2 Faithful reasoning and shortcut learning in LLMs

There have been several studies on the faithfulness of LLM reasoning. CoT often fails to be genuinely responsible for the answer it precedes (Bao et al., 2024), and faithful CoT remains challenging even with fine-tuning or activation edits (Tanneru et al., 2024). Methods that explicitly impose structure on reasoning are emerging (e.g., $G^2$-Reasoner combines knowledge and goal prompts to improve multi-step reasoning) (Chi et al., 2024). A related line of work intervenes directly in internal representations, editing factual circuits and testing hypotheses via activation patching, to probe and modify reasoning pathways. Beyond reasoning faithfulness, shortcut learning—where models exploit spurious correlations rather than the intended task structure—has been widely documented and mitigated via invariance penalties and counterfactual augmentation (Geirhos et al., 2020; Arjovsky et al., 2019). Yet most of these methods either require

external annotations or operate *post-hoc*. We build on this direction by using internal, pathway-aware signals during post-training, and by explicitly aggregating them with task rewards to control the accuracy–coherence trade-off.

## B Theoretical Guarantees

**Setup.** Let $\pi, \pi_{\text{old}}$ be stationary policies, $\gamma \in (0,1)$ the discount factor, and $d^\pi$ the normalized discounted state distribution. Fix the baseline policy $\pi_{\text{old}}$ and let $A(s,a) = A_{\pi_{\text{old}}}(s,a)$ be its advantage under the CE-PO reward $r_{\text{CE}}$ (the $L_p$ combiner of all channels). For each state $s$, denote by $\mathcal{N}_s \subset \mathbb{R}^{|\mathcal{A}|}$ the subspace spanned by *nuisance directions* (e.g., verbosity/length) estimated via counterfactual perturbations, and let $P_s$ be the Euclidean orthogonal projector onto $\mathcal{N}_s$. Write the action simplex $\Delta^{|\mathcal{A}|} = \{p \in \mathbb{R}^{|\mathcal{A}|} : p \geq 0,\ \mathbf{1}^\top p = 1\}$ with tangent space $\mathcal{T}_s = \{v \in \mathbb{R}^{|\mathcal{A}|} : \mathbf{1}^\top v = 0\}$. Define the *orthogonalized advantage*

$$A_\perp(s,\cdot) := (I - P_s)\, A(s,\cdot),$$
$$\varepsilon_\perp := \max_s \|A_\perp(s,\cdot)\|_\infty.$$

and let

$$\alpha := \max_s D_{\text{TV}}\big(\pi(\cdot|s), \pi_{\text{old}}(\cdot|s)\big) = \tfrac{1}{2} \max_s \|\pi(\cdot|s) - \pi_{\text{old}}(\cdot|s)\|_1.$$

The TRPO-style surrogate is

$$L^{\text{CE}}_{\pi_{\text{old}}}(\pi) = \eta_{\text{CE}}(\pi_{\text{old}}) + \frac{1}{1-\gamma} \mathbb{E}_{s\sim d^{\pi_{\text{old}}}}\big[\langle A(s,\cdot),\, \pi(\cdot|s) - \pi_{\text{old}}(\cdot|s)\rangle\big],$$

noting that $\langle A(s,\cdot), \pi_{\text{old}}(\cdot|s)\rangle = 0$ for all $s$.

**Assumptions and propositions**

**P0 (Fixed reward at $\pi_{\text{old}}$)** All CE components used to construct $r_{\text{CE}}$ (Jacobians, counterfactual subspaces $\mathcal{N}_s$, projectors $P_s$, and channel scores) are computed at $\pi_{\text{old}}$ and held fixed while evaluating $\eta_{\text{CE}}(\pi)$ over one update step.

**A0 (Residualization on the action-simplex tangent)** By construction of $r_{\text{CE}}$ via counterfactual projection, the CE advantage is locally invariant along nuisance directions at $\pi_{\text{old}}$:

$$\langle A(s,\cdot),\, v\rangle \;=\; 0 \quad \text{for all } v \in \mathcal{N}_s \cap \mathcal{T}_s.$$

Equivalently, $P_s A(s,\cdot) = 0$ and hence $A(s,\cdot) = A_\perp(s,\cdot) \in \mathcal{N}_s^\perp$.

**A1 (Well-posedness & bounded rewards)** Base rewards and channel scores are in $[0,1]$ and CE weights sum to one, so $r_{\text{CE}} \in [0,1]$. The MDP is standard (finite or $\sigma$-finite), ensuring existence of $V^\pi, Q^\pi, A_\pi$.

**Theorem B.1** (Orthogonalized TRPO Lower Bound for CE-PO)**.** *For any $\pi$, with $\alpha = \max_s D_{\text{TV}}\big(\pi(\cdot|s), \pi_{\text{old}}(\cdot|s)\big)$, the CE-PO performance satisfies*

$$\eta_{\text{CE}}(\pi) \;\geq\; L^{\text{CE}}_{\pi_{\text{old}}}(\pi) \;-\; \frac{4\gamma}{(1-\gamma)^2}\, \varepsilon_\perp\, \alpha^2.$$

*Proof.* **Step 1: Performance-difference decomposition.** The performance-difference lemma gives

$$\eta_{\text{CE}}(\pi) - \eta_{\text{CE}}(\pi_{\text{old}}) = \frac{1}{1-\gamma}\, \mathbb{E}_{s\sim d^\pi}\big[\langle A(s,\cdot),\, \pi(\cdot|s)\rangle\big].$$

Add and subtract the surrogate:

$$\eta_{\text{CE}}(\pi) = L^{\text{CE}}_{\pi_{\text{old}}}(\pi) + \frac{1}{1-\gamma}\, \mathbb{E}_s\Big[(d^\pi - d^{\pi_{\text{old}}})(s)\, \langle A(s,\cdot),\, \pi(\cdot|s)\rangle\Big]$$
$$=: L^{\text{CE}}_{\pi_{\text{old}}}(\pi) - \frac{1}{1-\gamma} T,$$

so it suffices to bound $|T|$.

**Step 2: Bounding the occupancy shift.** By a standard coupling argument, $\|d^\pi - d^{\pi_{\text{old}}}\|_1 \leq \frac{2\gamma}{1-\gamma}\, \alpha$. Thus

$$|T| \leq \big(\max_s |\langle A(s,\cdot),\, \pi(\cdot|s)\rangle|\big)\, \|d^\pi - d^{\pi_{\text{old}}}\|_1$$
$$\leq \frac{2\gamma}{1-\gamma}\, \alpha\, \max_s |\langle A(s,\cdot),\, \pi(\cdot|s)\rangle|.$$

**Step 3: Using CE-PO orthogonalization (A1).** For each $s$, by A1 we have $A(s,\cdot) = A_\perp(s,\cdot) \in \mathcal{N}_s^\perp$, and $\langle A(s,\cdot), \pi_{\text{old}}(\cdot|s)\rangle = 0$ (definition of advantage). Hence

$$\langle A(s,\cdot),\, \pi(\cdot|s)\rangle = \langle A_\perp(s,\cdot),\, \pi(\cdot|s) - \pi_{\text{old}}(\cdot|s)\rangle$$
$$\leq \|A_\perp(s,\cdot)\|_\infty\, \|\pi(\cdot|s) - \pi_{\text{old}}(\cdot|s)\|_1$$
$$\leq 2\, \varepsilon_\perp\, \alpha.$$

Taking the supremum over $s$ yields $\max_s |\langle A(s,\cdot),\, \pi(\cdot|s)\rangle| \leq 2\, \varepsilon_\perp\, \alpha$, and therefore

$$|T| \leq \frac{2\gamma}{1-\gamma}\, \alpha\, (2\, \varepsilon_\perp\, \alpha) = \frac{4\gamma}{1-\gamma}\, \varepsilon_\perp\, \alpha^2.$$

**Step 4: Final bound.** Plugging into Step 1,

$$\eta_{\text{CE}}(\pi) \geq L^{\text{CE}}_{\pi_{\text{old}}}(\pi) - \frac{4\gamma}{(1-\gamma)^2}\, \varepsilon_\perp\, \alpha^2,$$

which is inequality (B.1). □

**Properties.** *(Tightening via nuisance removal, $\ell_2$ statement).* Let $\varepsilon_{\text{base}}^{(2)} := \max_s \|A_{\text{base}}(s,\cdot)\|_2$ be the corresponding constant under a non-residualized baseline reward, and define $\varepsilon_{\perp}^{(2)} := \max_s \|A_{\perp}(s,\cdot)\|_2$. If there exists $\rho \in (0,1]$ such that $\|P_s A_{\text{base}}(s,\cdot)\|_2 \geq \rho \|A_{\text{base}}(s,\cdot)\|_2$ for all $s$ (*nontrivial nuisance energy*), then by Pythagoras for orthogonal projectors,

$$\varepsilon_{\perp}^{(2)} \;\leq\; \sqrt{1-\rho^2}\,\varepsilon_{\text{base}}^{(2)}.$$

Since $\|\cdot\|_\infty \leq \|\cdot\|_2$, we also have $\varepsilon_\perp \leq \varepsilon_\perp^{(2)}$, yielding a strictly smaller penalty constant in the $\ell_2$ analogue of equation B.1 (and a looser but still improved constant in $\ell_\infty$ up to norm equivalence).

*(Null penalty for hacking-only updates).* If for some $s$ the per-state policy change lies entirely in $\mathcal{N}_s$, i.e., $\pi(\cdot|s) - \pi_{\text{old}}(\cdot|s) \in \mathcal{N}_s$, then $(I - P_s)(\pi - \pi_{\text{old}}) = 0$ and hence $\langle A_\perp(s,\cdot), \pi(\cdot|s) - \pi_{\text{old}}(\cdot|s)\rangle = 0$, so that state contributes nothing to the penalty term in equation B.1.

**KL-based corollaries.** By Pinsker's inequality $D_{\text{TV}}(p,q) \leq \sqrt{\frac{1}{2}\text{KL}(p\|q)}$, if $\delta := \max_s \text{KL}\big(\pi_{\text{old}}(\cdot|s)\|\pi(\cdot|s)\big)$, then

$$\eta_{\text{CE}}(\pi) \;\geq\; L_{\pi_{\text{old}}}^{\text{CE}}(\pi) \;-\; \frac{2\gamma}{(1-\gamma)^2}\,\varepsilon_\perp\,\delta.$$

If only an *expected* per-state KL is enforced, $\mathbb{E}_{s\sim d^{\pi_{\text{old}}}}\big[\text{KL}(\pi_{\text{old}}\|\pi)\big] \leq \bar{\delta}$, then equation B holds with $\delta = C\,\bar{\delta}$ for a method-dependent constant $C \geq 1$ (e.g., via ratio clipping or per-state caps used in TRPO/PPO).

**From counterfactual Jacobians to action-space nuisance.** Let $\mathcal{U}_s$ be a finite set of representation-space directions obtained from counterfactual Jacobians (e.g., top singular vectors of counterfactual $J_{AB}$ blocks), and let $g_s$ be a smooth channel-score functional (e.g., spectral energy share). Write $G_s := \nabla_{\pi(\cdot|s)}\, g_s$ for its Jacobian with respect to action probabilities. Define

$$\mathcal{N}_s \;:=\; \text{span}\big\{\, G_s^\top u \;:\; u \in \mathcal{U}_s \,\big\} \;\cap\; \mathcal{T}_s.$$

If each channel score is *residualized* by projecting $J_{AB}$ onto $\mathcal{U}_s^\perp$, then, at $\pi_{\text{old}}$, the resulting CE reward satisfies $\langle \nabla_\pi r_{\text{CE}}(s,\cdot), v\rangle = 0$ for all $v \in \mathcal{N}_s \cap \mathcal{T}_s$, implying $P_s A(s,\cdot) = 0$ (Assumption **A0**).

# C Additional Analysis

## C.1 Wall-clocks, storage and complexity of CE-PO.

We design the Jacobian-based coherence reward to be *practical at scale*: it reuses forward activations, avoids any explicit Jacobian/Hessian materialization, and replaces costly subspace operations with a constant-time projection. Concretely, we rely on three tricks. *(i) Local Jacobian via vJP (vector–Jacobian Projection).* Sensitivities are computed with one forward and two backward passes using vector–Jacobian products (reverse-mode autodiff), i.e., for a test direction $U$ we use $\text{vJP}(s_B, E(A); U) = \langle J_{AB}, U\rangle_F = J_{AB}^\top \text{vec}(U)$ and the scalar sensitivity $\|J_{AB}^\top \text{vec}(U)\|_2$; so no dense $J \in \mathbb{R}^{Td\times Td}$ (or Hessian) is ever formed; working memory stays at $\mathcal{O}(Td)$ for token embeddings (Baydin et al., 2018; Pearlmutter, 1994). *(ii) Residual projection.* The counterfactual direction is removed with a single inner product: if $g$ is the local Jacobian direction and $\hat{c}$ the (reshuffled) counterfactual direction, we use $r = g - (g^\top \hat{c})\hat{c}$, avoiding SVD/subspace construction and extra buffers (Golub and Van Loan, 2013). *(iii) Mixed precision.* Forward/backward in FP16/bfloat16 with dynamic loss scaling provides wall-clock gains while roughly halving activation memory, without changing the algorithmic interface (Micikevicius et al., 2018).

***Throughput and footprint***. On a single modern GPU (e.g., A100 80 GB) and Llama-3-8B-Instruct, our implementation returns a reward in $\approx 2$ s, with negligible persistent memory beyond base activations. Simple Jacobian takes $O(Td)$ space complexity and would take $O(C_{fwd})$ time complexity, where $C_{fwd}$ means the time for one single forward in LLM reasoning. Yet for Hessian score, there needs to be continuous iteration of simple Jacobian to converge for a stable score and space complexity would also be $O((Td)^2)$, which is not practical. To provide a more quantitative analysis, we benchmarked our Jacobian computation against a BERTScore baseline on Qwen-3-14B (on a single node of 4 A100 GPU) using 100 randomly generated samples. We found the baseline took 1.20s and peaked at 28.00 GB, while our Jacobian step took 6.98s and peaked at 30.18 GB. This represents a 5.82x wall-clock time overhead and, more importantly, only a 1.078x GPU memory usage multiplier. Our CE-GRPO implementation introduces minimal overhead compared to original GRPO. As

shown in Table 4, the end-to-end wall-clock time and peak GPU memory over the **whole training pipeline** remain comparable between the two methods. Although the Jacobian-based reward computation incurs a $5.82\times$ overhead *in isolation*, this cost is largely amortized in practice: the dominant bottleneck in multi-GPU GRPO training is **inter-GPU communication** (e.g., rollout synchronization and all-reduce operations), not reward computation. Consequently, the overall training throughput degradation of CE-GRPO is modest.

Table 4: End-to-end wall-clock time and peak GPU memory for the whole training pipeline on Llama-3-8B-Instruct (single node, 4× A100 80 GB).

| Method | Time (h) | Mem (GB) |
|---|---|---|
| Original GRPO | 7.8 | 144.6 |
| CE-GRPO (Ours) | 8.8 | 155.2 |
| **Overhead** | **1.13×** | **1.07×** |

### C.2 Proof of length-induced bias in base Jacobian

We provide a simple theoretical argument showing that base Jacobian-based rewards are inherently sensitive to sequence length when used without normalization or counterfactual projection.

**Case 1: Length inflation on the source side (dimensional hacking).** By definition,

$$J_{AB} = \frac{\partial s_B}{\partial E(A)},$$

where $E(A) \in \mathbb{R}^{T_A \times d}$ is the embedding matrix of segment $A$. Therefore $J_{AB} \in \mathbb{R}^{T_A \times d}$. Consider the raw reward defined as the Frobenius norm:

$$R_{\text{raw}} = \|J_{AB}\|_F = \sqrt{\sum_{i=1}^{T_A} \sum_{j=1}^{d} J_{ij}^2}.$$

As $T_A$ increases, the number of additive terms in the outer summation grows sublinearly $O(\sqrt{T_A})$, which mechanically inflates $R_{\text{raw}}$ even if per-token sensitivities remain unchanged.

**Case 2: Length inflation on the target side (additive hacking).** The downstream score is a sum over tokens of $B$:

$$s_B = \sum_{t=1}^{T_B} \log p_\theta(b_t).$$

Applying the sum rule,

$$J_{AB} = \frac{\partial s_B}{\partial E(A)} = \sum_{t=1}^{T_B} \frac{\partial \log p_\theta(b_t)}{\partial E(A)}.$$

Thus increasing $T_B$ causes the final Jacobian to accumulate more matrices additively, enlarging its overall norm irrespective of semantic contribution.

Together, these two effects show that base Jacobian rewards scale with both upstream and downstream length, making them vulnerable to trivial reward hacking via token expansion. This motivates the counterfactual denoising and spectral normalization steps in CE-PO, which explicitly remove length-correlated nuisance directions and rescale Jacobian energy into a bounded coherence signal.

### C.3 Training objective

Let $R_p(Z, X, Y) \in [0, 1]$ denote the unified reward from Eq. (7). We treat $R_p$ as a *scalar terminal reward* for each rollout $(Z, X, Y)$: $r_T = R_p,\ r_t = 0$ for $t < T$, and **do not backpropagate through** $R_p$ (standard policy-gradient; the Jacobian signals are used only to compute $R_p$).

**CE-PPO.** With behavior policy $\pi_{\text{old}}$, importance ratio $w_t(\theta) = \pi_\theta(a_t|s_t)/\pi_{\text{old}}(a_t|s_t)$, and GAE advantages $\hat{A}_t$ computed from the sparse reward, the clipped objective is

$$\mathcal{J}_{\text{CE-PPO}}(\theta) = \mathbb{E}\Big[\min\big(w_t(\theta)\hat{A}_t,\ \text{clip}(w_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\big) - \beta\,\text{KL}\big(\pi_\theta \,\|\, \pi_{\text{ref}}\big)\Big]. \tag{8}$$

We optionally standardize $R_p$ per batch to stabilize advantages.

**CE-GRPO.** For each prompt we sample $K$ candidates, compute $R_{p,k}$, and form group-relative advantages $\tilde{A}_k = \big(R_{p,k} - \mu_{\text{grp}}\big)/(\sigma_{\text{grp}} + \varepsilon)$, with $\mu_{\text{grp}}, \sigma_{\text{grp}}$ the group mean/std. The GRPO-style objective is

$$\mathcal{J}_{\text{CE-GRPO}}(\theta) = \mathbb{E}\Big[\tfrac{1}{K}\sum_{k=1}^{K} \min\big(w_k(\theta)\tilde{A}_k,\ \text{clip}(w_k(\theta), 1-\epsilon, 1+\epsilon)\tilde{A}_k\big) - \beta\,\text{KL}\big(\pi_\theta \,\|\, \pi_{\text{ref}}\big)\Big]. \tag{9}$$

### C.4 Detailed Statistics

Training is performed on a single node with four A100 or H100 GPUs. Unless stated otherwise, we use a rollout group size of 6 per prompt, KL penalty of $\lambda_{\text{KL}} = 0.001$ to stabilize optimization, and cap decoder outputs at 2048 tokens and input within 512 tokens. In determining $R_p$, extensive experiments indicate that a good configuration could be $p = 0.2$ and correctness and coherence scores weights, respectively, as $(w_0, w_1, w_2, w_3) =$

Table 5: Guess-rate analysis. *Guess Rate*: fraction of guess-style completions (rationale $< 20$ tokens). $\Delta$: over-count from EM relative to LLM-as-Judge.

| **Benchmark** | **Guess Rate (%)** | **$\Delta$** (EM $-$ LLM-Judge) |
|---|---|---|
| BBEHMATH | 8.3 | +0.7 |
| CLadder | 44.7 | +10.7 |
| LegalBench | 62.7 | +1.9 |
| LogiQA | 64.5 | +12.0 |
| **Avg.** | 53.7 | +6.3 |

Table 6: Inter-rater reliability. $\kappa > 0.80$ denotes "almost perfect" agreement (**?**).

| **Rater Pair** | **Agreement (%)** | **Cohen's $\kappa$** |
|---|---|---|
| GPT-4o-mini vs. Human | 94.0 | 0.88 |
| GPT-4o vs. Human | 96.5 | 0.93 |
| GPT-4o-mini vs. GPT-4o | 98.5 | 0.97 |

$(0.7, 0.1, 0.1, 0.1)$, which is set as default for comparison unless otherwise specified in ablation studies. We set learning rate of $2 \times 10^{-6}$, batch size 32, and training for 10 epochs with temperature 0.6. Validation interval is set as 10 and checkpoints are saved per 100 steps. Experiments are based on Verl (Sheng et al., 2024) framework with mild modification for separate thread for reward calculation, and we also set warm-up step = 10 for critic.

## D Justification of LLM-as-Judge over Exact Match

EM accepts guessed answers whenever the surface token matches by chance, and cannot detect internally inconsistent reasoning. To quantify this gap, we measure the *guess rate*—the fraction of responses where rationale is less than 20 tokens—and report what fraction each protocol accepts. We sample 500 held-out examples per benchmark from the Llama-3-8B-Instruct checkpoint trained with CE-GRPO.

We additionally validate GPT-4o-mini reliability via inter-rater reliability (IRR) on 200 randomly sampled examples annotated by human experts (4 graduate students, 50 examples each), GPT-4o-mini, and GPT-4o.

GPT-4o-mini achieves $\kappa$=0.88 against human labels and deviates from GPT-4o in under 2% of cases, confirming it as a reliable, cost-efficient judge.

## E Ablation of Counterfactual Perturbation

To isolate semantic signal in our reward computation, we construct a counterfactual input that removes meaning while preserving surface-level properties such as sentence length and word choice. We compare two strategies: random replacement (RR) and token shuffle (TS).

RR replaces each token with a randomly drawn word from the vocabulary, which removes not only meaning but also the original word distribution. As a result, the difference between the original and counterfactual inputs mixes semantic signal with noise from swapping in unrelated words. TS instead permutes tokens in place, keeping the same words in a different order. This destroys sentence meaning while leaving word-level statistics unchanged, so the difference reflects semantics alone. We therefore adopt TS as our counterfactual construction strategy.

**Why token shuffle is a sufficient counterfactual.** We acknowledge that token shuffling also alters positional patterns and local syntax. However, our goal is not to construct a perfect intervention, but to produce a contrast where semantic content is disrupted while surface statistics (token identity, length, unigram frequency) are maximally preserved. TS achieves this better than RR, which additionally distorts the word distribution and conflates semantic with distributional differences. Table 7 confirms this: TS yields +1.66% average accuracy and +0.75% AFR over RR.

**Projection removes nuisance directions, not useful signal.** If projecting out counterfactual singular vectors were merely a generic regularizer, removing individual coherence channels should degrade performance gracefully. Instead, Table 3 (rows 5–7) shows that dropping any single term among $S(Z, X)$, $S(Z, Y)$, $S(X, Y)$ re-opens reward hacking: generation length inflates toward the token cap (Figure 12) and accuracy falls *below* the vanilla baseline. This collapse is confirming that the projected directions carry fundamental nuisance energy rather than semantic content.

## F Additional Plots

This section provides supplementary visual evidence supporting the core claims of the paper.

We further include additional ablation plots demonstrating reward degradation and length inflation when individual Jacobian coherence terms

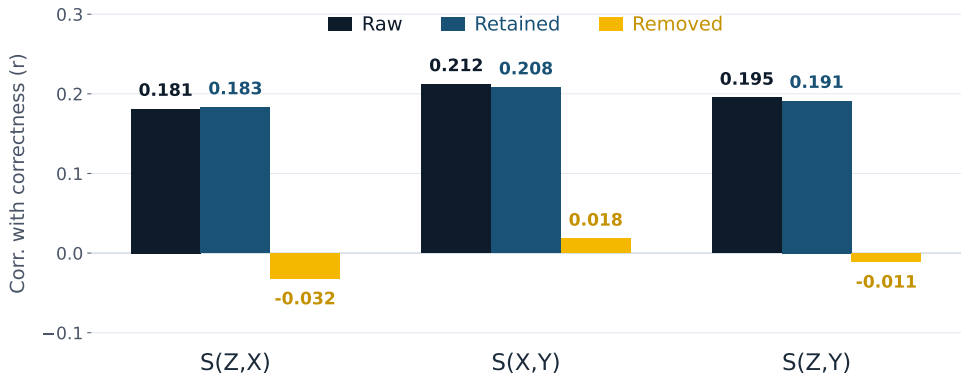



Figure 8: **Semantic content of projected subspaces.** For each Jacobian link, we decompose the raw signal into the *retained* component (after counterfactual projection) and the *removed* component (the projected-out subspace), then compute the point-biserial correlation of each with answer correctness ($r_{\text{base}}$). Across all three links, the retained component preserves virtually all correlation with correctness ($r \approx 0.18$–$0.21$), while the removed component drops to near zero ($|r| < 0.04$), confirming that the counterfactual perturbation discards semantically inert directions without sacrificing task-relevant signal.

Table 7: Counterfactual perturbation ablation on Llama-3-8B-Instruct (CE-GRPO). *Length*: mean generation tokens at convergence. Best in **bold**.

| Strategy | BBEHMATH | CLadder | LegalBench | LogiQA | Avg. | AFR (%) |
|---|---|---|---|---|---|---|
| RR (random replace) | 12.38 | 43.52 | 65.14 | 69.75 | 47.70 | 17.62 |
| TS (token shuffle) | **13.64** | **45.00** | **66.97** | **71.82** | **49.36** | **18.37** |

are removed, generation case comparisons illustrating how different reward formulations lead to distinct reasoning behaviors, and diagnostic analyses revealing the correlation between base Jacobian magnitude and output length.

Figure 11 reports the response lengths of CE-GRPO, GRPO-BERTScore, and the vanilla model on the training sets. The means are closely matched, indicating our optimization is largely *length-indifferent*. CE-GRPO achieves better task performance without increasing the token budget beyond the baseline, suggesting the efficiency of CE-GRPO.

**Question.** *Context: For husbands that don't set the alarm, the probability of ringing alarm is 90%. For husbands that set the alarm, the probability of ringing alarm is 25%. Question: For husbands that set the alarm, would it be less likely to see ringing alarm if the husband had not set the alarm?*

**Ground Truth:** No

| Method | Prediction (Reasoning) |
|---|---|
| BERTScore | Predicts "Yes"; relies on surface overlap (*...compares 25% vs 90% directly but ignores that the subject of "less likely" is "not set"*). |
| Jacobian Only | Predicts "Yes"; observes 25% < 90% but *misplaces the subject, treating "set" as the focus of "less likely" instead of "not set"*. |
| CE-GRPO | Predicts "No"; correctly checks whether 90% < 25% (*false*) and concludes "not set" actually makes ringing more likely, matching ground truth. |

**Takeaways.**

- **Jacobian Only** tends to *template-follow*: latches onto "less likely ⇒ smaller %" and ignores subject/polarity flips, leading to a misread of the conditional structure.
- **BERTScore** measures overlap, not conditional structure; it cannot reason about which condition the "less likely" modifies.

Figure 13: Comparison of methods on the alarm-setting counterfactual example on Qwen-3-4B-Thinking.

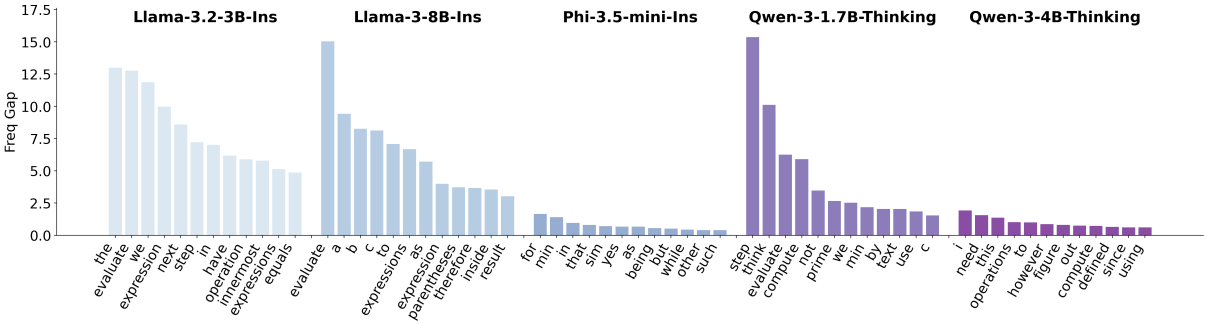



Figure 9: Frequency gap of generation per word where CE-GRPO exceeds the baseline models (Top-12 per model).

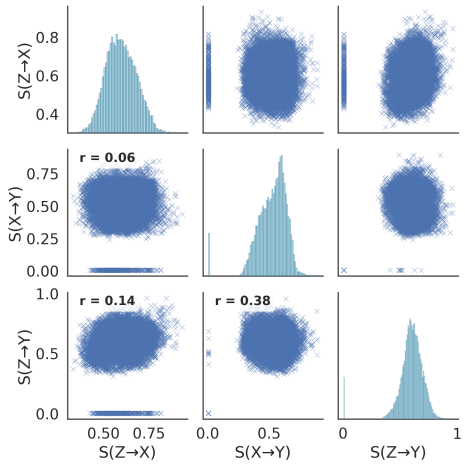



Figure 10: Pairwise scatter matrix of Jacobian coherence metrics on Qwen-3-4B.

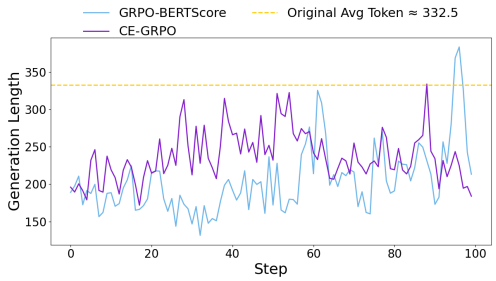



Figure 11: Response length comparison on Llama-3-8B-Ins.

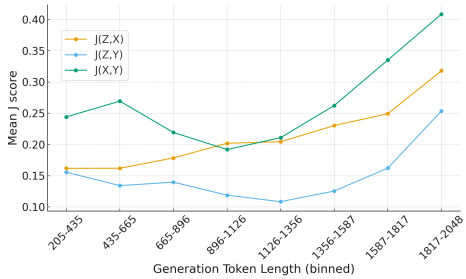



Figure 14: This figure illustrates the binned base Jacobian scores distribution among different generation token length, echoing the reward hacking by length statement in Figure 2.

## G Broader Impact

The proposed CE-PO framework has the potential to substantially improve the reliability and trustworthiness of large language models. By explicitly rewarding Jacobian coherence in addition to task accuracy, the method reduces the tendency of models to exploit superficial cues or engage in reward hacking, which can lead to misleading or unfaithful reasoning. In practice, this enables LLMs to generate explanations and rationales that are not only correct in outcome but also grounded in meaningful reasoning processes—an essential property for high-stakes domains such as law, science, medicine, and education.

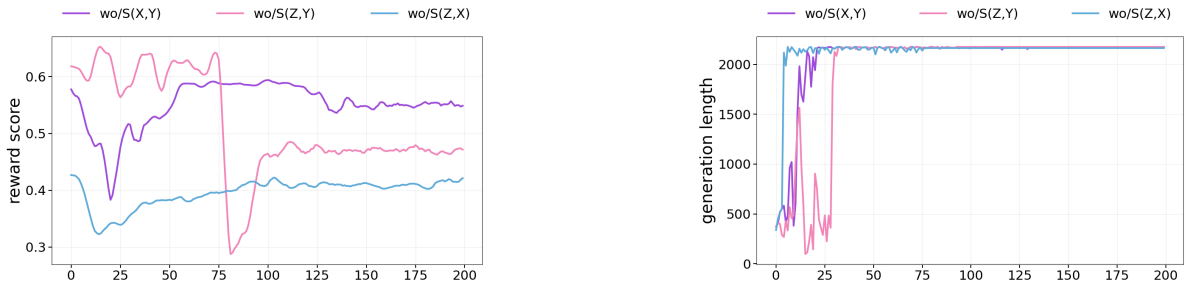
Beyond improving alignment, CE-PO can serve as a general methodology for building AI systems that are "right for the right reasons," supporting transparency, auditability, and robustness under distribution shifts. This contributes to safer deployment of generative models in real-world decision-making pipelines. At the same time, emphasizing coherence regularization may help mitigate harmful biases that arise when models rely on spurious correlations, thereby supporting more equitable AI systems.

## H Prompt Templates

We presented the prompt template for clear $Z$, $X$, $Y$ separation in Figure 15 and detailed instruction for LLM-as-Judge in Figure 16.

Table 8: Welch's $t$-tests comparing $S(A, B)$ distributions between Incorrect and Correct groups.

| Metric | $t$ | $p$-value |
|---|---|---|
| $S(Z, X)$ | 3.55 | 0.0005 |
| $S(X, Y)$ | 3.23 | 0.0014 |
| $S(Z, Y)$ | 3.15 | 0.0018 |




(a) Reward score

(b) Mean length

Figure 12: Supplement to the ablation analysis: omitting a single Jacobian term induces reward degradation accompanied by length inflation.

**Prompt Instructions (Final Answer Tagging)**

```
"Solve the question and generate in assigned format."
"Wrap your final answer between <finalanswer> </finalanswer> tags. "
"It's mandatory and required to not include anything after </finalanswer> tag. Don't let your
generation process to redundant and lengthy."
"Be very clear in your explanation, and ensure the final answer is presented separately."
"Example of output: We analyze A, solve B and answer is C. So final answer is <finalanswer> C is
correct </finalanswer>"
```

Figure 15: Verbatim instructions for final-answer tagging.

# I Dataset Features

# J The Use of Large Language Models (LLMs)

LLMs are the primary subject of our reinforcement learning (RL) experiments: we fine-tune pretrained models under CE-PO and baseline objectives to study stability, reward hacking, and reasoning quality. Separately, we use LLMs as utilities for controlled text transformations—rephrasing, grammar correction, and during writing phase.

**Prompt Template (Strict Judge)**

**Task**
Decide if the final answer **matches** the ground truth.

**Rules (both must hold)**
1) **Final-answer equivalence**: extract the last explicit final answer and compare it to the ground truth *after trimming whitespace and ignoring case*.
2) **Reasoning validity** : Validity ($Z \rightarrow X \rightarrow Y$): Z must substantiate X, and Z,X must substantiate Y. Accept only reasoning that is logically sound, evidence-backed, and contradiction-free; apply near-adversarial scrutiny—any gaps, guesses, or inconsistencies $\rightarrow$ 0.

**Question Z**
`{question}`

**Ground Truth Answer**
`{correct_answer}`

**Provided Reasoning X+Y**
`{reasoning_process}`

**Output (plain-text JSON, no code fences)**

```
{
  "judge_explanation": "<brief explanation>",
  "result": "<Yes or No>"
}
```

Figure 16: Prompt template for strict binary judgment requiring answer equivalence and valid reasoning.

Table 9: Training Dataset

| Dataset | Question | Answer |
|---|---|---|
| **BBEHCausal** | Question: Alice (AF) and Bob (BF) each fire a bullet at a window, simultaneously striking the window. The window only shatters (WS) if it is hit by two bullets. Is Alice firing the bullet a sufficient cause for the window shattering? Reply based on the answer a logician would give. | No |
| **CasHOLD** | Context: limitations was equitably tolled fails because he did not raise this argument before the district court. See Hinton v. Pac. Enters., 5 F.3d 391, 395 (9th Cir.1993) (HOLDING). Grace's remaining contentions lack merit. Select the correct holding (1 to 4): "holding that the burden to allege facts sufficient to establish jurisdiction resides with plaintiff", "holding that the plaintiff bears the burden when relying on the discovery rule", "recognizing the validity of the doctrine but holding no equitable tolling on the facts presented", "holding that plaintiff bears the burden to timely allege facts supporting equitable tolling", "holding that the burden is on the plaintiff to allege facts sufficient to establish jurisdiction" | 3: "holding that plaintiff bears the burden to timely allege facts supporting equitable tolling" |
| **MATHHARD** | There are thirty-five red, yellow, orange, and white marbles in a bag. If half the number of red marbles equals two less than the number of yellow marbles, equals a third the number of orange marbles, and equals a third of three more than the number of white marbles, how many red marbles are there? | 8 |
| **IfQA** | If Caroline Flack's mother had felt the name suited her, what would have been Caroline's name? | Caroline |